Audio-Visual Egocentric Action Recognition

Evangelos Kazakos

A dissertation submitted to the University of Bristol in accordance with the requirements for the degree of Doctor of Philosophy in the Faculty of Engineering

September 2, 2022

53518 words

***And those who were seen dancing were thought to be insane by those who could not hear the music*.**

Friedrich Nietzsche

Abstract

Egocentric actions generate distinctive and varied sounds from the interactions between hands and objects. Yet, egocentric vision methods had dismissed the auditory signal and were focused on understanding object manipulations through visual reasoning solely. This thesis enhances egocentric action understanding with audio recognition capabilities by capitalising on the close proximity of the wearable sensor to the ongoing action that enables capturing crisp audio recordings.

This thesis leverages the natural synergy of vision and audio and focuses on audio-visual integration for egocentric action recognition. As actions progress at different speeds for each modality, traditional synchronous fusion approaches cannot associate misaligned discriminative moments of each modality. To unlock this potential, this thesis proposes an asynchronous fusion approach by randomly binding appearance, motion and auditory inputs within temporal windows.

The next endeavour of this thesis is to improve the audio understanding capacity of visually impaired action recognition models. Inspired by the two-stream hypothesis of the human auditory system, this thesis introduces a novel two-stream auditory architecture, where a slow stream focuses on harmonic sounds and a fast stream captures percussive sounds. This thesis also offers an investigation on four-stream architectures that fuse slow and fast visual and auditory streams and showcases the vital importance of audio-visual regularisation for training such architectures.

Finally, this thesis brings a new perspective to action recognition from untrimmed videos by showing that the current paradigm of treating each action in isolation is inefficient. Using the key insight that untrimmed videos offer well-defined sequences of actions, this thesis proposes to strengthen the understanding of actions by exploiting the temporal progression of the activity that takes place within their temporal context. To this end, this thesis introduces the notion of multimodal temporal context and proposes a model to capture the inductive biases of untrimmed videos using vision, audio and language.

To My Family.

Acknowledgements

I would like to offer thanks to my supervisor, Dima Damen, for her continuous support and patience during my PhD. Her commitment to science is inspiring and I feel grateful for having the opportunity of working alongside her these four years, and learning from her how to do research. Dima is not only an incredible advisor but she is also compassionate, which made me feel safe during personal challenges that I faced while I was doing my PhD.

Thank you to everyone in the VILab: Jonny, Fae, Richard, Jian, Zeynel, Miltos, Davide, Mike, Will, Hazel, Toby, Adriano, Jacob, Kevin, Alex, Dena, Ahmad, Dan, Bin, Eduardo, Obed, Young, Ramon, Xingrui, Abel, Perla, Yanan and Farnoosh, for making the lab a pleasurable and inclusive working environment. A special thanks to my close teammates for their strong work ethic and their willingness to help me overcome difficulties I faced in my research. Thank you to Toby for proofreading my thesis.

A big thank you to Andrew Zisserman, Arsha Nagrani and Jaesung Huh for this amazing collaboration. Working with them was a truly unique experience. The discussions about research during our meetings were enlightening, and I'm grateful to them for making my PhD journey splendid. A particular thanks goes to Andrew; his kindness and wisdom are truly rare, and I feel really fortunate for getting to work with him, I learned so much from him.

I would like to thank my parents and my sister for their selfless support and love throughout my life. They always believe in me and that gives me strength to move on. Without them I would not have made it here. Thank you for giving light and meaning to my life.

Last but not least, thank you to my friends, Kostas, Mitsos, Pablo, Hugo and Alexia, for being there when I needed them, but also for sharing together amazing times of joy and laughter.

List of Publications

The work described in this thesis has been presented in the following publications:

1. E. Kazakos, A. Nagrani, A. Zisserman, D. Damen. EPIC-Fusion: Audio-Visual Temporal Binding for Egocentric Action Recognition. In *International Conference on Computer Vision (ICCV)*, 2019. (Ch. 3)

2. E. Kazakos, A. Nagrani, A. Zisserman, D. Damen. Slow-Fast Auditory Streams for Audio Recognition. In *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2021. (Ch. 4)

3. E. Kazakos, J. Huh, A. Nagrani, A. Zisserman, D. Damen. With a Little Help from my Temporal Context: Multimodal Egocentric Action Recognition. In *British Machine Vision Conference (BMVC)*, 2021. (Ch. 5)

Additionally, during my PhD I have contributed to the following papers:

1. D. Damen, H. Doughty, G. M. Farinella, S. Fidler, A. Furnari, E. Kazakos, D. Moltisanti, J. Munro, T. Perrett, W. Price, M. Wray. Scaling Egocentric Vision: The EPIC-KITCHENS Dataset. In *European Conference on Computer Vision (ECCV)*, 2018.

2. D. Damen, H. Doughty, G. M. Farinella, S. Fidler, A. Furnari, E. Kazakos, D. Moltisanti, J. Munro, T. Perrett, W. Price, M. Wray. The EPIC-KITCHENS Dataset: Collection, Challenges, and Baselines. In *Transaction on Pattern Analysis and Machine Intelligence (TPAMI)*, 2021.

3. D. Damen, H. Doughty, G. M. Farinella, A. Furnari, E. Kazakos, J. Ma, D. Moltisanti, J. Munro, T. Perrett, W. Price, M. Wray. Rescaling Egocentric Vision: Collection, Pipeline and Challenges for EPIC-KITCHENS-100. In *International Journal of Computer Vision (IJCV)*, 2021.

Author's Declaration

I declare that the work in this dissertation was carried out in accordance with the requirements of the University's Regulations and Code of Practice for Research Degree Programmes and that it has not been submitted for any other academic award. Except where indicated by specific reference in the text, the work is the candidate's own work. Work done in collaboration with, or with the assistance of, others, is indicated as such. Any views expressed in the dissertation are those of the author.

SIGNED: DATE:

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER

ONE

INTRODUCTION

With the availability of multi-sensor wearable devices (*e.g.* GoPro, Vuzix Blade, Pupil Labs, Google Glass, Microsoft Hololens), egocentric audio-video recordings have become popular in many areas such as extreme sports, health monitoring, life logging, home automation and augmented reality. As a result, there has been a renewed interest from the computer vision community on collecting large-scale datasets [36, 63, 151, 167] as well as developing new or adapting existing methods to the first-person point-of-view scenario [13, 108, 109, 120, 179]. Egocentric videos offer a unique perspective of the wearer's hands and how these grasp and interact with objects. Thus, at the core of egocentric perception is the ability to reason about hand-object interactions where researchers have proposed various approaches including hand and object detection [13, 120], attention-based mechanisms to focus on salient regions around the hands [108, 179] as well as distilling egocentric signals (hand-object interaction maps and interactive object scores) from third-person datasets into first-person action recognition models [109]. Unreasonably, there has been no works exploring the role of audio in understanding object interactions despite being available at no extra cost in egocentric footage.

One reason might be that the tale of audio for third-person actions was not compelling. While audio has been successfully used for video representation learning [4, 5, 6, 7, 94, 137, 142], in third-person action recognition with datasets like Kinetics [90] the contribution of audio has been limited [115, 116, 133, 196, 207]. One cause is that these datasets are downloaded from YouTube and the visual action cannot always be heard due to irrelevant audio, such as speech from a narrator or background music. Moreover, the camera is often far from the performer and sounds relevant to the action might not be captured.

Nevertheless, the egocentric domain is vastly different as it offers rich sounds resulting from the interactions between hands and objects, as well as the close proximity of the wearable microphone to the undergoing action. Audio is a prime discriminator for some actions (*e.g.* 'wash', 'fry') as well as objects within actions (*e.g.* 'put plate' vs 'put bag'). Moreover, there are sound-emitting objects with distinct sound signatures (*e.g.* 'coffee machine' vs 'microwave oven'). At times, the temporal progression (or change) of sounds can separate visually ambiguous actions (*e.g.* 'open tap' vs 'close tap'). Audio can also capture actions that are out of the wearable camera's field of view, but audible (*e.g.* 'eat' can be heard but not seen). Conversely, other actions are *sound-less* (*e.g.* 'wipe hands') but can be seen. One challenge is that the wearable sensor might capture irrelevant sounds, such as talking or music playing in the background. The opportunities and challenges of incorporating audio in egocentric action recognition allow the exploration of novel multi-sensory fusion approaches but also pave the path towards enabling action recognition solely from the audio stream. Using these key insights, this thesis unlocks the potential of models that can both *listen and look* to objects, actions, and interactions. Towards this goal, the thesis proposes novel solutions to four problems that concern egocentric vision and hearing.

Problem 1: Audio-visual fusion

The central importance of appearance and motion for understanding actions has made two-stream architectures with spatial and temporal streams the prominent approach in third-person action recognition [24, 46, 110, 168, 187, 194, 217]. Research in egocentric action recognition has adapted the two-stream paradigm to encode hand-object interactions by introducing specialised appearance and motion streams [13, 120, 178, 179]. Other works extended the two-stream architecture to multiple streams with additional ego-specific inputs [170, 172]. Uniquely, this thesis explores audio as a prime modality to provide complementary information to visual modalities (appearance and motion) in egocentric action recognition.

However, fusing multi-modal data streams is not trivial and there are certain caveats that require consideration when combining inputs as a function of time. Works on multi-modal fusion within convolutional networks for action recognition commonly take either a holistic approach of combining temporally aggregated representations of each modality [115, 116, 194] or combine multi-modal temporal inputs using synchronised samples across modalities [46]. Each of these approaches are problematic in different ways. Fusing global representations disregards the importance of time for multi-modal integration whereas associating the most salient moments from each modality can bolster the learning of stronger representations and enable modelling the temporal progression of actions in a *cross-modal* rather than *within-modal* setting. Importantly, actions unfold at a different pace for each modality and at times modalities are

temporally misaligned in the semantic level, *i.e*. the discriminative moments of each modality might occur at different temporal positions. Therefore, a synchronous fusion approach cannot account for such modality misalignments. Moreover, the large disparities of sampling rates of different modalities make synchronous fusion further laborious and the effect becomes more pronounced with more modalities.

To address these challenges, this thesis draws inspiration from neuroscience where there is evidence about the existence of a Temporal Binding Window in humans that allows individuals to perceptually link multi-sensory inputs within a range of temporal offsets. Accordingly, the first milestone of the thesis is an *asynchronous* fusion approach that associates appearance, motion, and auditory samples using *random* modality offsets within temporal windows *before* temporal aggregation. Random offsets allow scaling up to many modalities with different sampling rates and also act as a regulariser during training.

Problem 2: Visually impaired action recognition

A visually impaired action recognition model should be able to resort to audio to comprehend egocentric actions. Yet, such a model needs enhanced audio recognition capabilities to compensate for the loss of visual information. To this end, the next endeavour of this thesis is to tackle auditory activity recognition – classifying objects, interactions and activities solely from the audio stream of videos depicting activities. This is a novel area of research as work on action recognition considers audio along with vision but not on its own right. Furthermore, the need for perceiving the sounds produced upon interacting with objects makes the problem unique from any prior work in audio recognition and particularly challenging. Differently from acoustic scene classification [124] that focuses on environmental background sounds, in auditory activity recognition there is the need for recognising sound-emitting and interactive objects. Compared to datasets for environmental sound classification [146] that do contain acoustic objects with distinct sound characteristics (*e.g*. 'washing machine' vs 'mouse click'), the fine-grained nature of egocentric actions entails intra-class variations (*e.g*. 'close cupboard' vs 'close fridge') necessitating more sophisticated approaches for discriminating between such subtle differences.

Inspired by Harmonic/Percussive Source Separation, this thesis considers two types of sounds in activity-based datasets, harmonic sounds that span in time at different frequencies (*e.g*. 'wash hands', 'mosquito buzzing') and percussive sounds that are momentary or temporally repetitive (*e.g*. 'put glass', 'playing tympani'). To model harmonic and percussive sounds, the thesis capitalises on evidence from neuroscience that points to the existence of two streams in the human auditory system, a ventral stream for acoustic object recognition and a dorsal stream for object localisation. Accordingly, the thesis proposes a two-stream architecture for audio

with a Slow stream that focuses on learning frequency semantics using a low sampling rate and increased channel capacity and a Fast stream that models temporal patterns with a high sampling rate and less channels. The architecture learns corresponding Slow-Fast concepts at multiple representation levels through multi-level fusion.

Problem 3: Multi-stream audio-visual fusion

The proposed auditory Slow-Fast network is in fact inspired from a similar vision-based architecture with a Slow stream for capturing appearance and a Fast stream for motion modelling [47]. This opens the opportunity of exploring the integration of corresponding streams across modalities, forming audio-visual Slow and audio-visual Fast pathways, a previously unexplored topic in multi-modal fusion for action recognition. The intuition behind this design is that it is advantageous to associate the appearance of an object with its spectral auditory properties (audio-visual Slow), as well as the motions that constitute an action with its auditory temporal patterns (audio-visual Fast).

This thesis proposes a cross-attention fusion approach for audio-visual Slow-Fast fusion, aiming to attend to the visual object that produces the sound and vice versa to select the acoustic components of the sounding object depicted in the video. However, training such architectures is challenging for two reasons. First and foremost, the network can memorise the noise in egocentric audio recordings due to irrelevant background sounds or silent actions, causing overfitting. This is more pronounced for the proposed architecture due to its increased capacity. Second, it has been shown that audio-visual networks overfit due to the discrepancy of learning dynamics between the two modalities – audio overfits faster causing joint optimisation schemes to underperform [196, 207]. This thesis shows that an integral component for improving the training of the proposed architecture is an audio-visual regularisation scheme that randomly drops the audio stream in training, inspired by [207]. Randomly dropping audio in training has two advantages: i) the network learns to rely less on audio, accounting for the noise in the audio recordings, and ii) the learning pace of the audio stream is adapted to that of its visual analogue.

Problem 4: Going beyond single actions

Most action recognition approaches take as input a single clip to predict an action. While this is reasonable for action recognition with trimmed videos, it is incompetent for capturing the prior structure of untrimmed videos that contain activities organised into well-defined sequences of actions. Actions are visually correlated within neighbourhoods of a given extent. For example consider the following sequence: take knife $\rightarrow$ cut pizza $\rightarrow$ put knife $\rightarrow$ take pizza slice $\rightarrow$ put pizza slice. It becomes easier to predict that the pizza is being cut if you watched the knife

being taken before and put down afterwards. Yet, approaches that treat actions in isolation cannot exploit this information and are prone to errors due to object occlusions (*e.g*. the knife being covered by the other hand), a problem that can be mitigated by looking at neighbouring action clips.

Based on this motivation, this thesis proposes a model that can capture the inductive biases of untrimmed videos by leveraging the action's temporal context. Interestingly, audio carries also important information about the temporal progression of the activity that takes place within the temporal context; in the example above, the sound of cutting the pizza may be followed by the sound of putting the knife down. Moreover, in addition to the temporal context in the data stream, the semantic temporal context from the labels of neighbouring actions is predictive for the activity too, *i.e*. an action can be predicted solely based on language. Taking these into consideration, this thesis proposes an attention-based architecture for capturing multi-modal temporal context by attending to neighbouring actions using vision and audio as input modality context and a language model acting as a prior to filter out improbable action sequences.

Contributions

The main contributions of this thesis are summarised as follows:

- The thesis offers an empirical study of the importance of audio for understanding object interactions.
- The thesis proposes a novel end-to-end trainable three-stream convolutional network that combines appearance, motion and audio with mid-level fusion within temporal windows along with an analysis of the length of the window. The ability of the proposed network to handle videos that contain audio with irrelevant sounds is also studied.
- The thesis introduces the problem of activity recognition using only audio. To address the problem, a novel bio-inspired two-stream network with multi-level fusion and convolutional separation over frequency and time is proposed.
- A four-stream architecture that combines two-stream auditory and visual networks with cross-modal attention is proposed, and the significance of audio-visual regularisation for training efficiently the proposed architecture is investigated.
- The thesis defines the temporal context of an action as the sequence of past and future actions from the untrimmed video. The thesis proposes a multi-modal transformer-based architecture to attend to multi-modal temporal context using vision, audio and language. The effect of the temporal context extent is analysed along with the significance of using multi-modal context. The upper bound performance of the incorporated language model

is also analysed to stress the potential of language models for action recognition.

Thesis structure

The thesis is organised as follows. Ch. 2 reviews a broad scope of works relevant to the research presented in this thesis. Ch. 3 offers a study on the significance of audio in comprehending object manipulations and recognising egocentric actions, and proposes a novel approach for asynchronous fusion of appearance, motion and audio. Ch. 4 delves into recognising egocentric actions using only audio and introduces a two-stream architecture that captures the harmonic and percussive structure of audio. Then, it investigates techniques for fusing and regularising visual and auditory two-stream architectures. Ch. 5 introduces temporal context from action sequences in untrimmed videos and proposes a model that attends to multi-modal temporal context using vision and audio as input modality context and a language model to capture prior temporal context from sequences of action labels. Finally, conclusions are provided in Ch. 6 which also discusses directions for future work.

CHAPTER

TWO

RELATED WORK

This chapter offers a review of a wide spectrum of the literature that is relevant to the work presented in this thesis. The structure of the chapter can be conceptually divided in three parts. The first part is concerned with action recognition, starting with Sec. 2.1 that analyses the field of visual action recognition from third-person videos, from convolutional methods to the more recent Transformer-based architectures. Then, Sec. 2.2 sets the grounds of audio processing and recognition of auditory scenes and events. Sec. 2.3 delves into audio-visual action recognition discussing three major aspects: architecture design, audio-visual regularisation and self-supervised audio-visual learning. Sec. 2.4 reviews the most dominant fields of studies that have arised in egocentric action recognition.

At the second part, Sec. 2.5 examines methods that leverage temporal context for video recognition. This is followed by Sec. 2.6 that delineates common paradigms in multi-modal Transformer architecture design and Sec. 2.7 that provides an introduction to language modelling and its use for capturing action context, as these can be used for modelling multi-modal temporal context as will be shown in Ch. 5.

Finally, Sec. 2.8 and Sec. 2.9 present audio-visual datasets for third-person and egocentric action recognition, respectively, and Sec. 2.10 reviews large-scale audio-visual datasets for scene and event recognition.

It should be noted that in its majority, this chapter focuses on modern deep learning approaches and does not survey traditional methods of the pre-deep learning era. For action recognition, two seminal works of that period include dense trajectories [193] and its follow up, Improved Dense Trajectories (IDT) [192]. While these works have pointed out the importance of mo-

tion for action recognition, they have some common characteristics that make them inferior compared to deep learning approaches. Firstly, they focus on extracting hand-crafted features from videos as opposed to learnable ones. Consequently, the extracted features do not adapt to the distribution of the dataset of interest, potentially missing important domain information. Secondly, the extracted features are shallow in that they encode low-level statistics of the video such as motion cues and cannot capture higher-level semantic information about the actors and the actions they perform. On the contrary, deep learning methods learn semantic information on the higher layers of the feature hierarchy.

2.1 Third-Person Action Recognition using only Vision

Action recognition is the problem of training a model to predict what a person (or several) is doing in a video using pairs of videos and action labels. A visual model should learn a function that maps sequences of video frames to a predefined set of semantic classes. Third-person videos are recorded from a camera that is placed somewhere in the scene and 'observes' the human from a *third-person viewpoint*, and thus capturing the body of the actor. Therefore, a third-person action recognition model should focus on the human body to comprehend the action being performed. While at time actions can be recognised from still images, for example when the posture of the actor has low variance across frames, *e.g.* 'drinking', motion is imperative for disambiguating between actions such as 'open' and 'close'. Moreover, for complex actions that are composed of multiple steps, like 'assembling furniture', utilising a sequence of frames is a necessity for modelling the progression of the action. Nevertheless, visual appearance is important too and complementary to motion. The background of a scene can provide contextual information for recognising an action, for example videos of the action 'swimming' can only be observed around the sea or water surroundings. In addition, visual attributes of the objects that are part of the action entail semantic information as well, *e.g.* for discriminating 'playing basketball' from 'playing football' the colour of the ball is a useful hint. Thus, action recognition models need to be able to perform both temporal and spatial reasoning.

The efforts of the action recognition community to address these challenges have resulted in a rich literature of seminal works innovating temporal modelling [110, 194, 217], spatio-temporal modelling with convolutional networks [24, 56, 187] and transformers [10, 18, 22, 57, 134, 143] as well as two-stream fusion [46, 47, 112, 168]. These are expanded below.

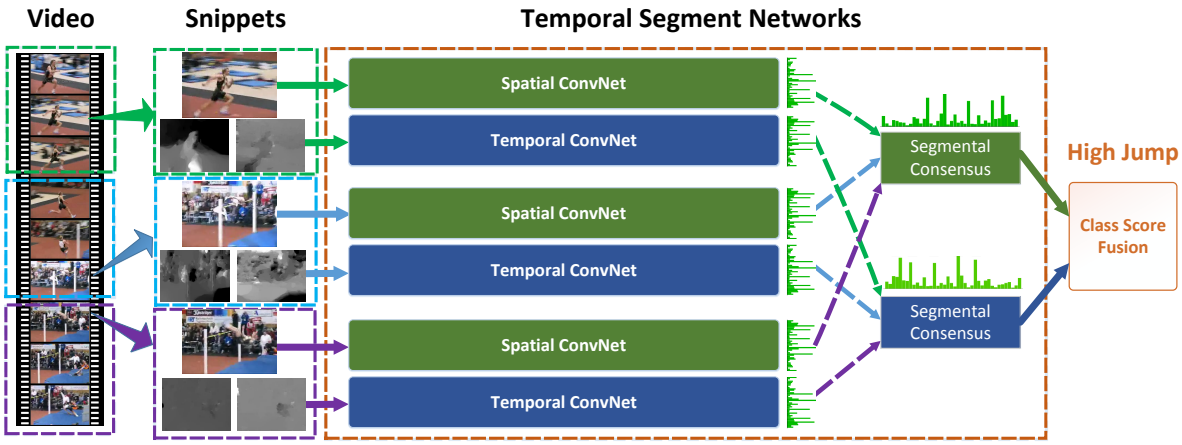


Figure 2.1: *Temporal Segment Networks (TSN) [194]. A video is split in three segments of equal length and a snippet is randomly sampled within each segment. A convolutional network, shared across snippets, makes snippet-level action predictions which are aggregated with a consensus function to provide a video-level prediction. Per-class averaging is used as the consensus function. An appearance TSN and a motion TSN are trained independently and fused in inference with averaging after temporal aggregation within each modality. The figure is from [194].*

2.1.1 Temporal modelling with 2D ConvNets

Wang *et al*. [194] tackled the problem of long-range temporal modelling, *i.e*. capturing temporal relationships in videos that span over long periods, effectively utilising the whole video clip. They indicated that prior attempts to long-range temporal modelling [41, 189, 212] fail as they employ dense temporal sampling making it computationally infeasible to cover more than only a handful of seconds of the video clip. To address this issue, they introduced Temporal Segment Networks (TSN), depicted in Fig. 2.1. The main novelty of TSN is a sparse temporal sampling strategy that allows to obtain video-level predictions (as opposed to frame-level or clip-level predictions) at a low computational budget. In particular, the video is divided in K segments of equal length, and within each segment a snippet is sampled randomly, formulating a sequence of snippets. Then, a 2D ConvNet operating on frames produces K snippet-level action predictions. A segmental consensus function $\mathcal{G}$ aggregates the snippet-level predictions into predictions over the entire video, where the authors applied per-class averaging as the consensus function. TSN maintains a low computational complexity as K is set to 3, significantly lower than the number of frames in videos. Importantly, the loss function operates on $\mathcal{G}$, and thus video-level dynamics are baked into the parameter gradients:

$$\frac{\nabla \mathcal{L}(y, \mathcal{G})}{\nabla W} = \frac{\nabla \mathcal{L}}{\nabla \mathcal{G}} \sum_{k=1}^{K} \frac{\nabla \mathcal{G}}{\nabla \mathcal{F}(T_k)} \frac{\nabla \mathcal{F}(T_k)}{\nabla W}, \tag{2.1}$$

where $\mathcal{L}$ is the loss function, y is the ground-truth, W are the model's parameters and $\mathcal{F}(T_k)$ are the predictions for each snippet. Interestingly, by performing class visualisations of the learnt models, the authors showed that a model trained on single frames tends to focus on background contextual information, such as the scenery, while TSN successfully attends to the actor performing the action, showcasing the benefits of backpropagating the error of the consensus.

The success of TSN triggered the interest of researchers in action recognition and some follow-up works adopted the sparse temporal sampling paradigm and proposed improvements for enhanced temporal modelling capabilities. Zhou *et al*. [217] argue that temporal relational reasoning, that is the ability to reason about temporal relations and transformations of entities over time is critical for action recognition and central for describing the progression of events. Accordingly they propose Temporal Relational Networks (TRN) to model temporal relations between frames at multiple temporal scales for capturing both short-term and long-term dependencies. A 2D ConvNet initially extracts N features sparsely sampled. The relational module then models d-frame relations, $\forall d = 2, .., N$, using two MLPs for each temporal scale. The first MLP associates d ordered frames for each respective d. As k ordered sets of d frames are sampled for each relation d, the second MLP links the k different associated sets of frames. TRN is end-to-end trainable allowing the features of the ConvNet to encode the relations learnt from the relational module. Differently from TSN that simply averages the predictions of the sparsely sampled frames, ignoring the temporal ordering of frames, TRN is aware of their order as the MLPs in the relational module operate on ordered frames, and thus the order is encoded in their weights.

Lin *et al*. introduced the Temporal Shift Module (TSM) [110], that is able to enhance 2D ConvNets with temporal modelling capabilities at no additional computational cost by simply shifting part of the channels of the spatial features maps over the temporal dimension. More particularly, a 2D ConvNet extracts features of size $C \times H \times W$ for T sparsely sampled frames, effectively forming features of size $C \times T \times H \times W$. Part of the C channels is shifted by $+1$ and others by -1 in the temporal dimension T, *i.e*. channels are shifted both to the future and past timesteps, respectively. This allows temporal convolutions to be approximated within the spatial convolutions, as parts of features from different timesteps are associated and jointly modelled in the subsequent 2D convolutions. The authors propose an online shift operation for online recognition too, by shifting channels only from the past. Two important finds are that, firstly the amount of channels is critical as a small number would not enable sufficient temporal modelling and a large number would degrade the spatial modelling capacity of the network. Secondly, to further allow for both spatial and temporal learning capacity, the authors introduce a residual shift operation, where the shift module is inserted within a residual branch.

Therefore, even after temporal shifting particularly a larger amount of channels, the original spatial features are preserved through identity mapping.

Acknowledging the advantages of sparse temporal sampling, Ch. 3 delves into effective ways of temporally associating multiple modalities and proposes multi-modal temporal binding.

2.1.2 Spatio-temporal modelling with 3D ConvNets

The recognition of the significance of time for understanding actions has spurred researchers to naturally extend 2D convolutional networks into 3D ones by convolving over the temporal dimension too using three dimensional kernels. Early efforts include [87, 186] which underperformed due to the lack of large-scale datasets. Here, the discussion is focused on more modern approaches that had access to large (pre) training data volumes and trained with more effective optimisation strategies as well as more carefully crafted architectures.

Inflated 3D ConvNets (I3D) were introduced in [24] with main motivation to leverage pre-trained well-designed architectures from image classification for action recognition, instead of designing architectures from scratch. To this end, the authors introduced inflating 2D ConvNets and their parameters into 3D ones. A 2D ConvNet is inflated by simply adding an extra dimension representing time in both the filters and the pooling kernels of an existing network design, where the authors inflated the Inception-V1 architecture [183]. Moreover, the convolutional filters / pooling kernels are symmetric in the spatial and temporal dimensions after inflation. The parameters of the 2D pre-trained model are inflated by simply replicating the 2D weights along the temporal dimension, and rescaling them by dividing by the size of the temporal dimension, such that the response of the filters remain the same.

[187] examines various 3D residual architectures showcasing their benefits over their 2D counterparts. Eventually, an R(2+1)D architecture is proposed that factorises spatio-temporal convolutions into a spatial convolution followed by a temporal convolution, yet keeping the number of parameters approximately equal to the full 3D network (R3D) to assess the importance of the factorisation and disentangle it from the effect of increasing/decreasing the number of parameters in the network. It is demonstrated that R(2+1)D outperforms R3D, where the authors attribute it to two factors: i) R(2+1)D has effectively double the number of nonlinearities comparing to R3D as there is an additional activation function between the factorised spatial and temporal convolutions, and thus R(2+1)D is more expressive, ii) the factorisation eases the optimisation of the network as the authors observed a lower training loss in addition to a lower validation loss.

In a similar spirit, and motivated by the good performance of depthwise separable convolutions for image classification (*e.g.* [32]), [188] introduced Channel-Separated Convolutional

Networks that provide a good trade-off between accuracy gains and computational complexity. Channel-Separated Convolutional Networks employ 3D residual networks with depthwise convolutions, *i.e*. spatio-temporal convolutions where each channel receives input only from the corresponding channel of the previous layer rather than from all channels as in vanilla convolutions. This allows for significant reduction in memory requirements and computational cost while maintaining the accuracy of the network. Importantly, the authors have shown that preserving the channel interactions in the convolutions is beneficial and propose factorised 3D convolutions by disentangling channel interactions from spatio-temporal interactions using a $1 \times 1 \times 1$ convolutional layer that allows for communication between channels followed by a $3 \times 3 \times 3$ depthwise convolutional layer solely for spatio-temporal modelling within channels. This strategy enables channel interactions while at the same time maintains a lower computational cost than original 3D convolutions.

2.1.3 Spatio-temporal modelling with Transformers

Over the last couple of years, Transformers [190] overwhelmed the domain of Nature Language Processing (NLP), which was until then grounded on recurrent and convolutional architectures, by showcasing that attention-only architectures have strong modelling capabilities in NLP tasks. The central idea is that sentences can be modelled by relating a word to all other words in the sequence of words. This is achieved with a dot-product self-attention mechanism where the input sequence $X \in \mathbb{R}^{N \times D}$ is mapped into queries Q, keys K and values V, and attention weights are calculated based on the dot-product similarity of queries with all keys. The attention is then computed as a weighted sum of the values using the attention weights:

$$\text{SA}(X) = \text{Softmax}\left(\frac{QK^T}{\sqrt{D_h}}\right)V, \tag{2.2}$$

where $Q = W_Q X$, $K = W_K X$, $V = W_V X$ and $W_Q, W_K, W_V \in \mathbb{R}^{D \times D_h}$. Multi-head Self-Attention (MSA) extends self-attention by performing k self-attention operations in parallel, which are concatenated and projected:

$$\text{MSA}(X) = [\text{SA}_1(X); \text{SA}_2(X); ...; \text{SA}_k(X)]W_{\text{MSA}}, \tag{2.3}$$

where for each self-attention operation $D_h = D/k$ and $W_{\text{MSA}} \in \mathbb{R}^{kD_h \times D}$. A Transformer encoder layer, l, then consists of Multi-head Self-Attention and an MLP block, with Layer Normalisation (LN) [12] being applied before each block and residual connections [71] after

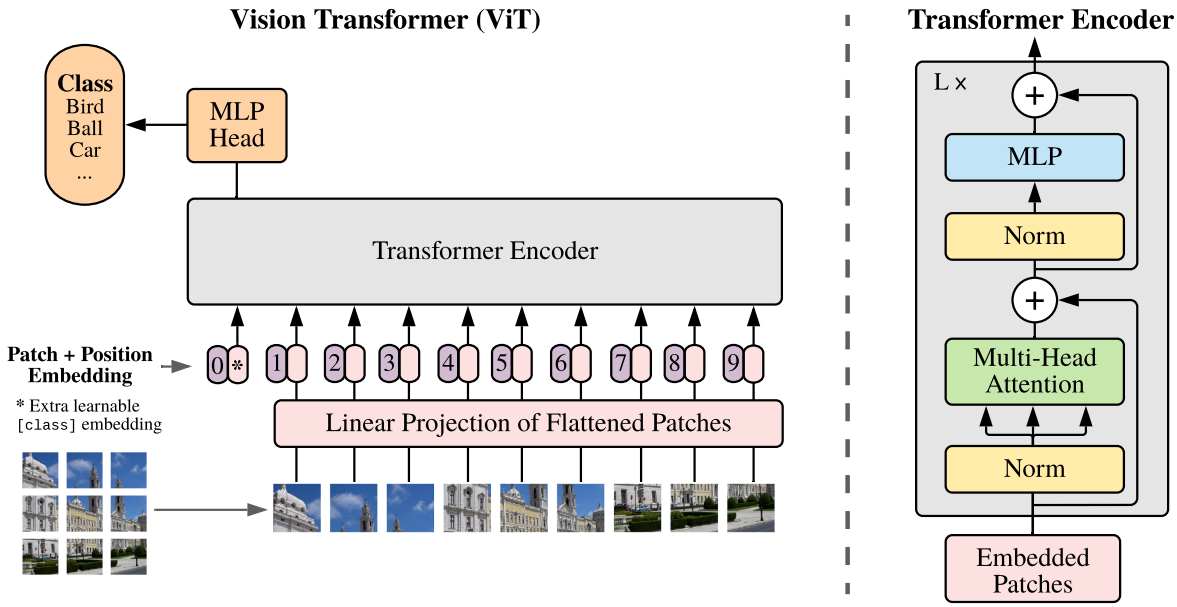


Figure 2.2: *Vision Transformer (ViT) [43]. An image is tokenised into patches of size* 16×16*, each. Each patch is embedded with a linear projection layer and augmented with positional information. A special learnable classification token is added at the beginning of the sequence. The sequence is encoded with a Transformer encoder, and a classification head is applied in the summary embedding,* i.e. *the encoded classification token. The figure is from [43].*

each block:

$$y_\ell = \text{MSA}(\text{LN}(z_{\ell-1})) + z_{\ell-1} \tag{2.4}$$

$$z_\ell = \text{MLP}(\text{LN}(y_\ell)) + y_\ell, \tag{2.5}$$

where MLP comprises of two linear layers with a GELU activation function [74] interleaved. Importantly, dot-product self-attention is permutation invariant and therefore disregards the position of elements in the sequence. To address this, positional encodings, fixed or learnt, are added to the inputs to enhance them with positional information which is fundamental for tasks that operate on sequences.

It was not long before Transformers made their appearance in computer vision where the concept from text can naturally adapt to the image domain: images can be modelled by relating a spatial location to all other spatial locations in the image. The Vision Transformer (ViT) [43] was of definitive importance. The idea in ViT is simple: the image is split into a grid of non-overlapping patches (visual words), each of size 16×16, a projection layer embeds each patch which are subsequently tagged with positional encodings. A special learnable clasification token is appended in the sequence which is fed to a transformer encoder. Finally, a classifier makes predictions using the summary embedding, *i.e.* the output from the classification token.

ViT is trained end-to-end from raw pixels. An overview of ViT is depicted in Fig. 2.2. Some benefits of Transformers over ConvNets are: i) Transformers perform attention with *dynamic* weights that depend on the data vs ConvNets performing convolutions with learnt but *fixed* weights, ii) Transformers are more flexible architectures as they impose less restrictive inductive biases than ConvNets, and iii) they are better at capturing long-range dependencies by means of global attention in the input sequence whereas ConvNets have a restricted receptive field and its enlargement is upper-bounded by the depth of the network.

While ViT can flexibly learn effective image representations with good downstream performance, it requires vast amounts of data to achieve this. Particularly, the authors of [43] pre-train ViT on ImageNet-21k [39] with 14M images and 21k classes as well as on JFT [181] with 300M images and 18k classes. Moreover, despite the ability to model long-term information, Transformers have quadratic complexity to the number of inputs as dot-product attention requires comparing each element with all other elements in the sequence. Based on these observations, several follow-up works proposed hybrid approaches that bake convolutions into ViT [204, 211] and preserve the benefits of Transformers while at the same time reduce their computational complexity and do not require large-scale pre-training by leveraging the inductive biases of convolutional networks, *i.e.* grid structure and translation invariance. Furthermore, other works proposed multi-scale ViT variants [114, 198] to learn hierarchical representations in Vision Transformers similar to ConvNets, with main motivation to enhance the performance of dense prediction tasks such as object detection and semantic segmentation. For a detailed review of Transformers in computer vision, please refer to the survey of [91].

As expected, the transition happened in action recognition too [10, 18, 22, 45, 57, 134, 143]. Girdhar *et al.* [57] were the first to employ transformers for action recognition and detection. They argue that one of the abilities required to understand human actions is to be able to reason about other humans and objects in the proximity of the person of interest. To this end, they introduce the Action Transformer, an end-to-end trainable architecture that employs an I3D network [24] to extract features, a Region Proposal Network [153] to generate bounding box proposals of detected people which are then used as queries to a Transformer to aggregate relevant context from other people and objects around each detected person.

Driven by the motivation to model long-term information in videos, [134] proposes to perform temporal attention over the entire sequence of frames from a video. To accomplish that, the authors employ ViT for per-frame spatial modelling, and a Transformer-based architecture that attends temporally to the sequence of frame-level features where the Longformer [15] was chosen for this purpose due to its ability to model longer sequences more effectively than the vanilla Transformer. Similarly to ViT, the 2D pooled features are tagged with temporal

positional information and a classification token is concatenated in the sequence as well, prior to self-attention from the Longformer. Finally, a classification head operating on the encoded classification token is used to predict actions.

[18] introduces TimeSFormer that extends ViT for video modelling by performing spatio-temporal attention over 3D volumes. TimeSFormer parses a video as a sequence of patches extracted from each individual frame, and similarly to ViT embeds each patch and enhances it with spatio-temporal positional information. Then, a classification token is appended to the sequence. This strategy of tokenising 3D volumes $\rightarrow$ embedding $\rightarrow$ positional encoding $\rightarrow$ adding classification token, is adopted by all the video transformer architectures described next. As the quadratic complexity of Transformers renders them computationally prohibitive particularly for videos where the number of spatio-temporal patches can be extremely large, the authors investigated efficient spatio-temporal attention schemes to alleviate the computational burden. They found out that a 'Divided Space-Time Attention' where temporal and spatial attention are applied independently performed the best while at the same time being computationally cheaper than joint space-time attention. More concretely, 'Divided Space-Time Attention' first performs temporal attention over all patches of the same spatial location across frames followed by spatial attention within each frame.

In a similar fashion, the work of [10] focused on different variants of factorising spatial and temporal attention and proposed the ViViT architecture. The authors found that while for larger datasets such as Kinetics [90] an unfactorised spatio-temporal transformer performed the best, for smaller datasets, including for egocentric ones likes EPIC-KITCHENS [36], a 'Factorised Encoder' outperformed all other variants while requiring fewer floating point operations (FLOPs). The Factorised Encoder consists of a spatial transformer encoder followed by a temporal transformer encoder. The spatial encoder is essentially a ViT applied on a per-frame basis. The encoded classification tokens of each temporal index then form the inputs to the temporal encoder. Interestingly, this strategy resembles the convolutional approaches [194, 217] where first a 2D convolutional network extracts frame-level features which are then fed to a temporal modelling module. It is also almost identical to the architecture of [134]. Other contributions of [10] include a tubelet embedding for videos as opposed to embedding each frame independently, good practices for leveraging the pre-trained ViT image model, and an ablation analysis of augmentation techniques / regularisers showcasing their importance in a small-scale data regime.

Bulat *et al*. [22] approach the problem of reducing the computational complexity of the full space-time attention, which is $O(T^2S^2)$ for T frames and S tokens per frame, from a different perspective. Different from [10, 18] that investigate spatio-temporal attention factorisation, [22] proposes full spatio-temporal attention within a temporal window. To further reduce the

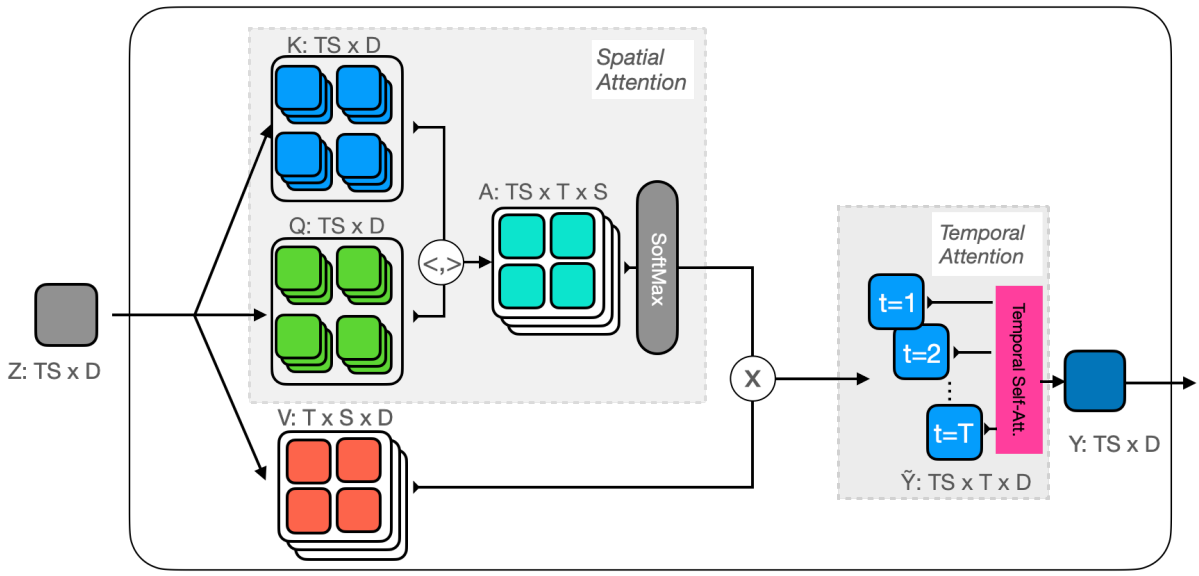


Figure 2.3: *Trajectory attention [143]. A query from Q is compared with keys K from each frame, and the learnt attention A is used to pool values V independently within each frame, producing trajectory tokens for each spatio-temporal location that span over the length of the video, T. This can be seen as locating the trajectory at a given timestep by comparing the the trajectory query to the keys at that timestep. Then, temporal attention operating on the trajectory tokens aggregates information across trajectories. The figure is from [143].*

computational complexity, channel shifting from neighbouring timesteps within the window is employed, similar to TSM [110]. Concretely, while the query representation remains intact, key and value representations of a given patch are mixed with temporal information from patches at the same spatial location within the window; part of the channels from neighbouring patches are shifted to the current patch resulting in spatio-temporal information encoded in the patches of a single frame. Then, full spatio-temporal attention within the window can be approximated by simply performing spatial attention within the frame, which recuces the computational complexity to $O(TS^2)$ compared to $O(T^2S + TS^2)$ in [10, 18].

Patrick *et al*. [143] claim that pooling temporally over motion trajectories is a fruitful inductive bias for videos, where objects move and may appear in different positions across different frames. Approaches that pool axially [18] or over the entire space-time volume [10] overlook these trajectories. Accordingly, the authors introduce a self-attention mechanism that enables attention along trajectories in two steps. Similarly to the other approaches [10, 18, 22] the video is parsed into spatio-temporal tokens and each token, st, represents the trajectory 'reference point'. The goal of the first step then is to locate the trajectory at a given timestep by

comparing the trajectory query to the keys at that timestep, which is implemented as:

$$\tilde{y}_{stt'} = \sum_{s'} v_{s't'} \frac{\exp(q_{st} \cdot k_{s't'})}{\sum_{\bar{s}} \exp(q_{st} \cdot k_{\bar{s}t'})}, \tag{2.6}$$

where $\tilde{y}_{stt'}$ represent the trajectory tokens at different times t', and q_{st}, k_{st}, v_{st} are the query, key and value respectively. Thus, there is a trajectory per spatio-temporal token with temporal extent same as the number of frames in the video. Note that, trajectory attention is different than both spatial-only attention and joint space-time attention in that, in trajectory attention a query is compared with keys from every frame but information is pooled independently within each frame, whereas in spatial-only attention queries and keys from the same frame only are compared, and in joint space-time attention a query is compared with keys from every frame and information is pooled across all frames. In the second step, information is aggregated along trajectories using temporal attention over the trajectory tokens. Although, this approach has the same computational complexity as joint space-time attention, it outperforms both joint space-time attention and divided space-time attention [18]. The proposed model is called Motionformer. An overview of the proposed attention is depicted in Fig. 2.3. Ch. 5 employs visual features extracted from Motionformer and demonstrates that they provide enhanced object recognition capabilities compared to convolutional features.

Finally, the concurrent work of [45] introduced Multiscale Vision Transformers (MViT), a multi-scale transformer architecture for videos that shares an analogous objective as the works of [114, 198] for images, *i.e.* to employ hierarchical representations in video Transformers. The paradigm of multi-scale feature hierarchies in computer vision and convolutional networks dictates that lower levels in the pyramid of features operate at high spatial resolution and a small number of channels to model low-level features and progressively the channel dimension is increased while decreasing the spatial resolution where higher levels in the hierarchy model semantics. In MViT, this is realised with a Multi Head Pooling Attention (MHPA). MHPA employs a pooling kernel and a stride, applied on the query of a self-attention operation to reduce the sequence length, effectively reducing the spatial dimension of the underlying visual representation. MViT has multiple scales and each scale reduces the spatial resolution of the previous scale using MHPA while increasing the channel capacity with an MLP. Interestingly, the authors show that by leveraging such prior for videos, MViT can reach or outperform other video transformers described above [10, 18, 134], without resorting to large-scale pre-training in contrast to the rest ViT-based video architectures [10, 18, 22, 134, 143].

Concluding remarks

The landscape of (single-stream) neural architectures for action recognition has undergone significant changes over the last years; starting from ConvNet approaches that pursued sparse

temporal sampling to model long-term dependencies and 3D convolutions for spatio-temporal modelling till the more recent Transformer-based methods that can tackle both problems more efficiently. Yet, the price to be paid is the enormous size of datasets required to allow Transformers to generalise well. When there is ample availability of data, the inductive priors of ConvNets limit the scope of functions they can model. Transformers are more suitable for big data regimes as they can more flexibly learn spatio-temporal relationships from scratch without any encoded structure in their architecture. On the other hand, the inductive biases integrated in ConvNets are particularly useful for labs and researchers with limited resources. Combining the benefits of both worlds by building some of the priors of ConvNets in Transformers is a promising direction for action recognition amongst other computer vision areas. Ch. 5 reasserts the usefulness of priors in Transformers for action recognition, by means of modelling the prior temporal context of actions using a language model.

2.1.4 Two-stream fusion

By observing the importance of spatial and temporal features for action recognition, fusion of appearance and motion has become a standard technique [24, 46, 47, 56, 88, 110, 112, 168, 187, 194, 217]. The objective is to harness the complementary information in the two modalities, where appearance can reason about scenes and objects while motion can model the movements of the actor. Typically, fusion is realised with two-stream architectures that consist of a convolutional network for spatial modelling and a second for temporal modelling, coupled with a mechanism for integrating the two streams. Different works suggested approaches of fusing the modalities at different levels of the architecture, namely *late fusion*, *mid-level fusion* and *multi-level fusion*, which are analysed next.

Late fusion. The pioneering work of Simonyan and Zisserman [168] was the first to propose the integration of appearance and motion streams for action recognition, inspired by the two stream hypothesis of the human visual cortex, where the ventral stream (also known as the 'what' stream) is responsible for object recognition and the dorsal stream (also known as the 'where' stream) models motion and is involved with spatio-temporal localisation. In the proposed architecture, the spatial ConvNet takes as input single RGB images and the temporal ConvNet takes as input a stack of horizontal and vertical optical flow components to model the motion across consecutive video frames. The authors explored different optical flow representations including i) simply stacking optical flow components of consecutive frames where at each point a vector represents its displacement to the corresponding point in the next frame, ii) stacking trajectories of optical flow where at each point a vector represents the displacement of the trajectory of the point and iii) a bi-directional optical flow representation that accounts for the motion in both directions. Interestingly, simple optical flow stacking outper-

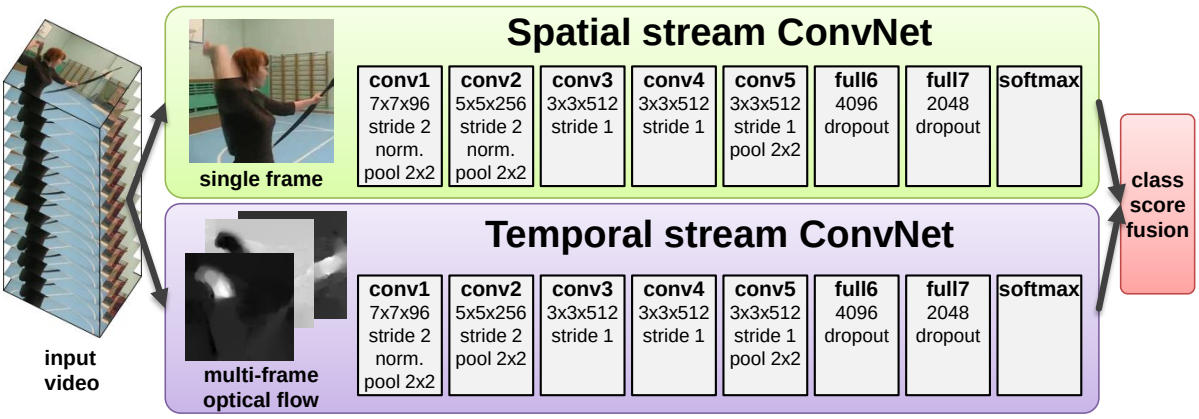


Figure 2.4: *Two-stream architecture with late fusion [168]. The architecture is comprised of a spatial stream that operates on single RGB frames for modelling appearance and a temporal stream operating on stacks of vertical and horizontal optical flow components of consecutive frames for modelling motion. The streams are trained independently and combined by late fusion of their softmax predictions where both averaging and a multi-class SVM that operates on the concatenation of the softmax predictions from each stream were considered. The figure is from [168].*

formed trajectory stacking and the bi-directional flow provided only negligible performance improvements. [168] proposed a late fusion approach by combining the predictions of both streams, where two strategies were considered: averaging the softmax scores of each stream as well as a multi-class SVM that operates on the concatenation of the scores. Each stream was trained independently, where the RGB stream was pre-trained on ImageNet while optical flow was randomly initialised, and the streams were only fused in test time. The architecture is depicted in Fig. 2.4. Furthermore, to alleviate overfitting caused by the small-scale video datasets that where available at that time (UCF-101 [173] and HMDB-51 [97]) the authors employed a multi-task learning scheme by training on both datasets simultaneously with a separate classification head for each dataset. Their results demonstrated the significance of ImageNet pre-training, the benefits of stacking optical flow (as opposed to using only one between two frames) that allows the network to model long-term information to an extent, and last but not least that, although optical flow was sufficient to recognise most actions, fusion provided notable boosts in performance capitalising on the complementarity of appearance and motion.

Following [168], all the temporal modelling approaches described above [110, 194, 217] as well as the spatio-temporal works of [24] and [187], adopted the two-stream paradigm with late fusion of appearance and motion streams. Moreover, late fusion of appearance and motion has been adopted by [212] that explored temporal modelling through different pooling strategies in convolutional networks as well as with LSTMs and also by [56] that introduced an extension of NetVLAD [8] for spatio-temporal aggregation of video representations.

Mid-level fusion. Feichtenhofer *et al*. [46] observed that the late fusion approach of [168] is not capable of modelling pixel-wise correspondences between appearance and motion features as each stream is trained independently and fused at the prediction level. Accordingly, they pose the questions of: i) what is the optimal way of fusing the streams, and ii) where to fuse the streams. To answer the first question, the authors explored different functions of fusing the appearance and motion convolutional feature maps to learn pixel-wise correspondences across the channels of each modality. Namely, they considered a) *sum fusion* which computes a summation of the feature maps of each modality at the same spatial locations and channels, b) *max fusion* that calculates the maximum of the feature maps position-wise and within each channel, c) *concatenation fusion* that simply stacks the feature maps in the channel dimension, d) *convolutional fusion* that first concatenates the feature maps and then inserts a convolutional layer to fuse them which also performs dimensionality reduction, and e) *bilinear fusion* that computes the outer product of the feature maps at each spatial location followed by summing over all locations. As discussed in the paper, convolutional, max and sum fusion reduce the number of parameters significantly as the network continues as a single stream after fusion, whereas concatenation fusion involves significantly more parameters as the dimensionality of the fused features is not decreased. While bilinear fusion enables interactions between all channels of each stream at each spatial location, its disadvantages are that it marginalises the spatial information and induces high dimensional fused features.

The authors have shown that convolutional fusion provided the best recognition accuracy, where the performance was comparable with late fusion [168] but with nearly half parameters. Another fundamental difference with [168] is that streams are trained simultaneously end-to-end to learn the correspondences between appearance and motion features. Importantly, regarding the second question, optimal results were achieved by combining the streams after the last convolutional layer, and accordingly *mid-level fusion* of the spatial and temporal streams was proposed. Mid-level fusion was also combined with late fusion but this option didn't provide any additional benefits. The proposed architecture is demonstrated in Fig. 2.5. An extension of the architecture to fuse the streams spatio-temporally with 3D convolutions was also proposed but this is not analysed here as the advantages of 3D convolutions have already been discussed in Subsec. 2.1.2.

An intriguing problem of fusing streams of data from different modalities concerns potential asynchronies between the modalities. Such asynchronies arise when the discriminative patterns of each modality occur at different timesteps. For example, in the action 'open cupboard' the discriminative motion of pulling the cupboard's handle takes place before the distinctive appearance of the cupboard being open. Approaches that can leverage such asynchronies by associating the most discriminative timesteps of each modality could learn stronger representa-

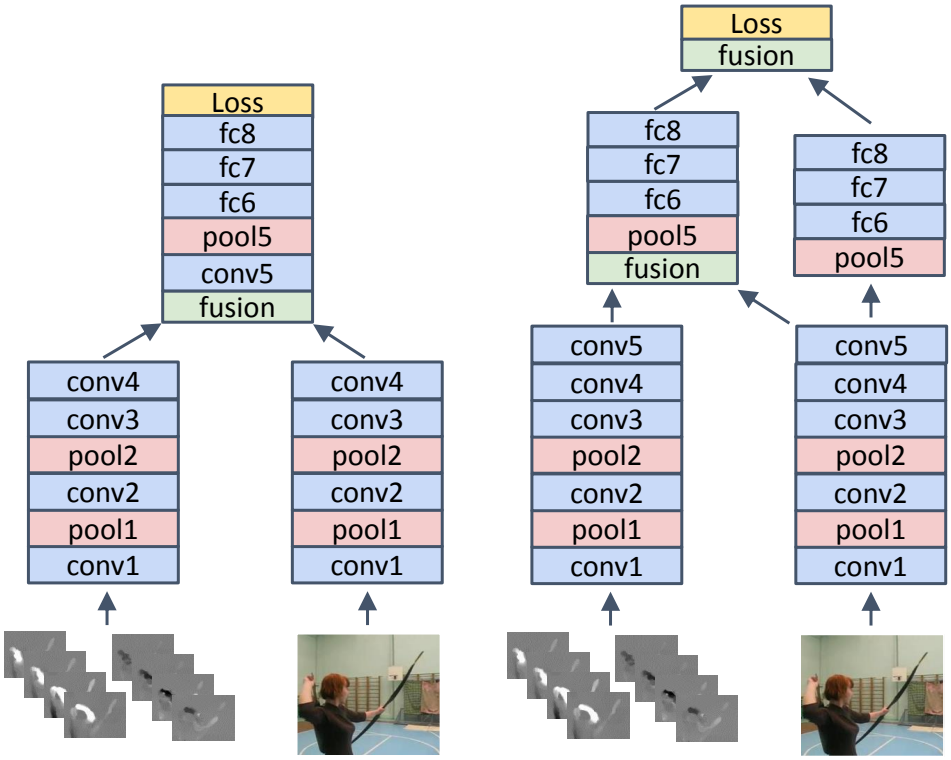


Figure 2.5: *Mid-level fusion of spatial and temporal streams [46]. (Left) The streams are combined at the fourth convolutional layer after which the architecture continues as a single ConvNet tower. (Right) The streams are combined with mid-level fusion at the fifth convolutional layer, the spatial stream is not truncated and the streams are fused with late fusion of predictions too. While [46] proposes fusion at the fifth convolutional layer, the combination of mid-level and late fusion did not advance the model's performance. The streams are trained simultaneously end-to-end. The figure is from [46].*

tions for those actions. The works described so far that employ late fusion or mid-level fusion along with temporal aggregation are not capable of capturing potential asynchronies between the appearance and motion streams. The sparse temporal sampling approaches [110, 194, 217] perform temporal aggregation / temporal modelling of each modality independently prior to late fusion of modalities. Similarly for the spatio-temporal convolutional approaches of [24] and [187]. Thus, any asynchronies are marginalised by temporal aggregation and the fusion can only combine global representations as opposed to individual timesteps. Conversely, [46] fused modalities first followed by temporal pooling of the fused representations, yet the streams were fused at corresponding temporal locations rather than asynchronously. Asynchronous modality fusion before temporal aggregation was proposed in [112], where the appearance of the current frame is fused with 5 uniformly sampled motion frames, and vice versa, using two LSTMs [77]. Therefore, the approach of [112] focused on using predefined asynchrony offsets between two modalities.

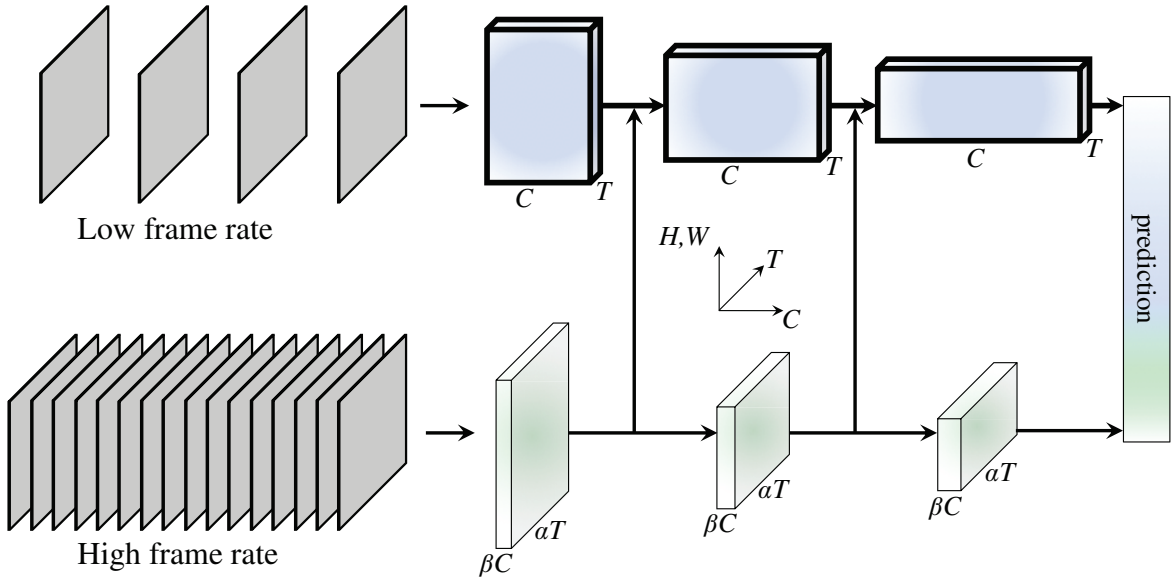


Figure 2.6: *SlowFast networks [47]. The architecture consists of a Slow and a Fast pathway. The Slow pathway operates on a low sampling rate with a high number of channels to focus on learning appearance semantics, while the Fast pathway has a high sampling rate and less channels to focus on modelling motion. The streams are fused at multiple representation levels with lateral connections from the Fast to the Slow stream. Finally, the pooled features of each stream are concatenated and fed to the classification layer. The figure is from [47].*

Multi-level fusion. Appearance and motion streams can also be combined at multiple representations levels [47, 88] to form hierarchically fused features where the earlier layers are responsible for integrating low-level information across modalities, *e.g.* edges with short-term movements, while the role of the upper layers is to associate semantics across modalities, *e.g.* an object with its motion signature. A preeminent architecture of this line of work is the SlowFast Networks [47]. The authors of [47] point out that the spatial semantics of videos are characterised by slow temporal changes whereas the motion of the actors progresses faster, and thus a low sampling rate should be sufficient to model appearance, although a higher temporal resolution is a necessity for capturing fine-grained movements. Based on this motivation, a two-stream architecture is proposed with a Slow pathway that operates on a low sampling rate to capture appearance and Fast pathway with a higher sampling rate to model motion. A breakthrough of [47] is that differently than all the two-stream approaches described until now, both streams operate on the raw video and the Fast stream does not depend on optical flow which is computationally expensive, and as a result a bottleneck for employment in real-world settings. In contrast, appearance and motion are modelled through the different design of the Slow and Fast streams while both digest RGB video frames. An overview of the SlowFast architecture is demonstrated in Fig. 2.6. The Slow stream operates on a low frame rate with a large channel

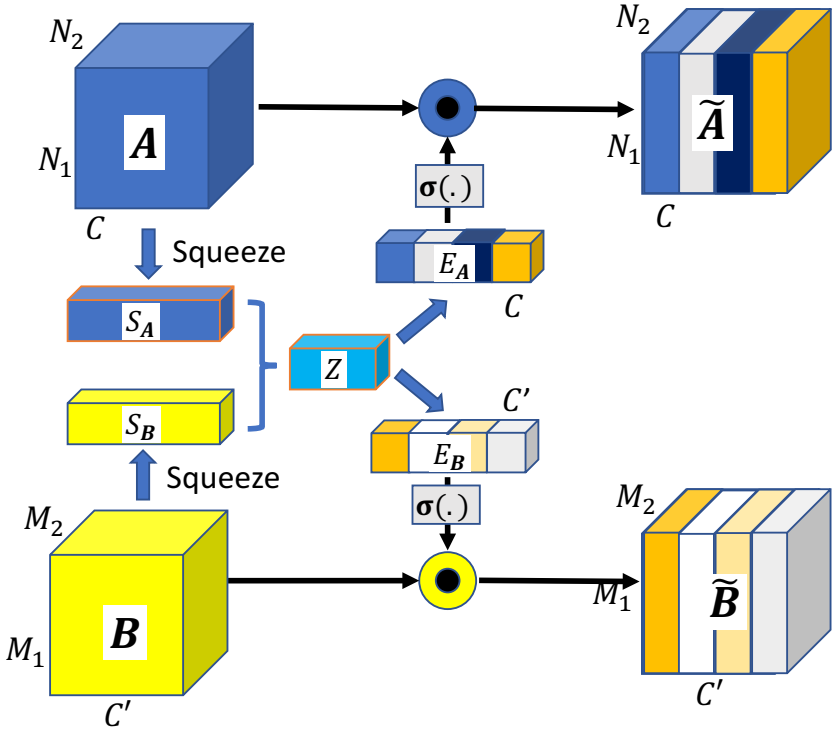


Figure 2.7: *Multimodal Transfer Module (MMTM) [88]. The squeeze operations, S_A and S_B, aggregate the feature maps of modalities A and B respectively, with global average pooling producing a channel descriptor per modality. These are concatenated and passed to a linear layer that generates a multi-modal feature, Z. Then an excitation operation per modality, E_A and E_B, that is composed of a linear layer and a sigmoid activation are employed to rescale the feature maps of each modality channel-wise. The figure is from [88].*

capacity to focus on modelling appearance. The architecture employs separable convolutions over space and time and the Slow stream has temporal convolutions only from intermediate layers while the earlier layers convolve only spatially to compensate for the low temporal resolution. The Fast stream has a higher frame rate and fewer channels while utilising temporal convolutions throughout to focus on learning temporal patterns. The frame rate and channel ratio between the two streams are controlled by α and β, respectively, which are set to $\alpha = 8$ and $\beta = 1/8$ in the Fast stream. The streams are fused with lateral connections from the Fast to the Slow stream at multiple levels of the network, which are realised as strided temporal convolutions such that the frame rate of Fast matches that of Slow, and the Slow and Fast features are concatenated in the Slow pathway. Finally, the pooled representations of the two pathways are concatenated and fed to the classification layer. Results demonstrate that the two streams learn complementary information which benefits action recognition when these are fused with lateral connections which is shown by ablating the lateral connections.

Joze *et al.* [88] expose the main impediments of fusing modalities at intermediate convolu-

tional layers: i) the feature maps of each stream at a given layer might be of different dimensionality and therefore combining them is not straightforward, and ii) fusion layers typically induce significant changes in the backbone architecture complicating the use of pre-trained models for each modality. The latter is one of the reasons that late fusion has become so popular. To address these, they propose the Multimodal Transfer Module (MMTM) that incorporates a multi-modal Squeeze and Excitation mechanism. Squeeze and Excitation networks [79] enhance the representations of *uni-modal* convolutional networks by recalibrating their feature map activations channel-wise through modelling interdependencies between channels with a gating mechanism. Correspondingly, an MMTM block recalibrates the features of each modality with *multi-modal* gating, in two steps. First, a squeeze operation aggregates the features maps of each modality with global average pooling producing a global representation per modality. These are concatenated and projected with a linear layer that generates a multi-modal descriptor. Then, an excitation operation per modality that is composed of a linear layer and a sigmoid activation rescales the feature maps of each modality channel-wise, emphasising important features and suppressing less important ones. A summary of an MMTM block can be seen in Fig. 2.7. It becomes clear that the squeeze operation facilitates integration of modalities with inconsistent number and size of dimensions. For example, it is suitable for fusing visual streams with spatio-temporal (3D) representations and auditory streams with frequency-temporal (2D) representations, where additionally their sampling rates differ substantially. The authors employ MMTM blocks in different domains including gesture recognition, audio-visual speech enhancement and action recognition. The action recognition architecture is comprised of an RGB stream that employs an I3D network [24] and a skeleton based network [100] modelling skeletal data and their motion. A noteworthy finding is that multi-level fusion with MMTM blocks is advantageous when applied at intermediate and higher levels of the architecture to learn corresponding semantics across modalities, while there are no benefits of fusing earlier layers as the inter-modality low-level features are less correlated.

2.2 Auditory Scene and Event Recognition

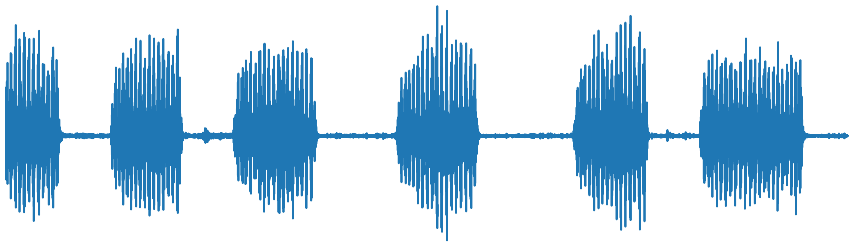
Audio is an important modality in its own right. We perform various tasks daily using audio alone. The predominant one is communicating with others through speech. Nevertheless, it extends beyond that; when visual scenes or events are out of sight, we resort to the sounds they produce to comprehend them. This inspiration has led researchers in developing methods for auditory scene and event recognition which are reviewed in this section. First, approaches that employ single-stream architectures are presented along with data augmentation and pre-training techniques, followed by a discussion on the importance of non-symmetric filtering of

time and frequency. Lastly, the section reviews works that introduced multi-stream auditory architectures.

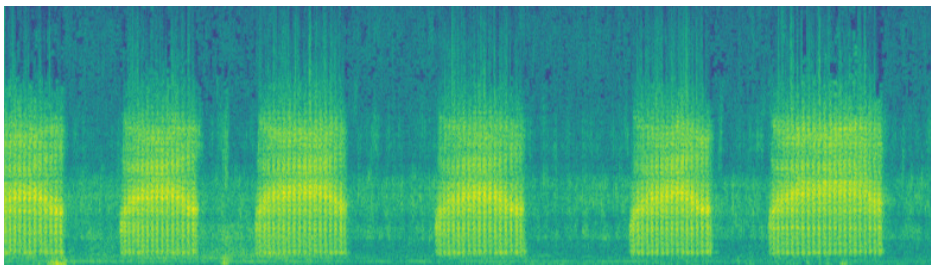
The prevailing paradigm in audio recognition of scenes and events is to convert the raw audio waveform into a 2D representation, typically a spectrogram, and use it to train 2D convolutional networks, which is the main focus of this section. Before delving in the relevant literature, it is important defining what is a spectrogram and what is the motivation of using it for training 2D ConvNets.

An audio signal consists of samples of air pressure over time, each representing the amplitude of the signal at a given temporal location. Audio signals are composed of multiple waves representing different frequencies. To capture those frequencies, one can apply the Discrete Fourier Transform (DFT) that converts a signal from the time domain into the frequency domain, resulting in an energy distribution of the signal over different frequencies, *i.e*. a vector representing the amplitude of different frequencies. In practice, the Fast Fourier Transform (FFT) is used which is more computationally efficient. However, commonly audio signals are non-stationary necessitating a representation that varies over time. This can be computed using the Short-Time Fourier Transform (STFT) that applies a series of FFTs over successive short time intervals. It requires as input the length of the window over which FFT will be applied and the length of the hop between successive FFTs. Typically, overlapping windows are used to account for correlations between neighbouring windows. This results in a 2D matrix representing the energy distribution of the signal over both frequency and time, the so called spectrogram. In spectrograms, time is represented in the x-axis while frequencies in the y-axis, resulting in a matrix $S \in \mathbb{R}^{T \times F}$ where $S(t, f)$ represents the magnitude of the f-th frequency at the t-th time. As humans perceive amplitudes in a logarithmic scale, *i.e*. we are more sensitive to small changes in amplitude for high amplitudes than low ones, usually log-spectrograms are calculated by taking the log of the spectrogram, amplifying low amplitudes while compressing higher ones.

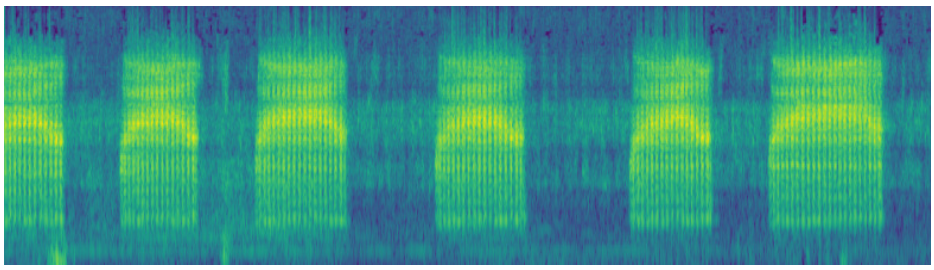
But, apart from amplitudes, humans perceive frequencies in a logarithmic scale too, *i.e*. we are more sensitive to low than high frequencies. This can be addressed by transforming the frequency scale to a Mel Scale. A Mel is a unit of pitch such that a pair of sounds equally distant to each other on the Mel Scale are also perceptually equally distant. The transformation takes the number of Mel bins as input and partitions the frequency range in equidistant bins expressed in Mels and corresponding logarithmically spaced bins expressed in Hz. Then, it creates a Mel filter bank using overlapping triangular filters that map frequency bins in Hz to Mel bins. Finally, the frequency scale is transformed through a matrix multiplication between the Mel filter bank and the log-spectrogam, resulting in a log-mel-spectrogram. Fig. 2.8 displays the extraction of a log-spectrogram and log-mel-spectrogram from an audio signal.



(a) *Raw waveform*



(b) *Log-spectrogram*



(c) *Log-Mel-spectrogram*

Figure 2.8: *Audio feature extraction. The example represents the class 'frog croaking' and is taken from the VGGSound [30] dataset. To compute the log-spectrogram in (b) using the audio signal in (a), a window of length 20ms and a hop of 10ms were used. The log-mel-spectrogram in (c) is calculated from (b) using 128 mel bins. The x-axis in all three figures represents time. The y-axis in (a) represents amplitude in the time domain while in (b) and (c) it represents frequencies in linear scale and mel scale, respectively, where the intensity of each pixel represents amplitude in the frequency domain. Note that in (c) the signal is more spread out than (b) in the frequency axis as the frequencies have been transformed from a linear scale to a logarithmic (mel) scale.*

Note that the log-mel-sprectrogram in Fig. 2.8c is more spread out in the frequency axis than the log-spectrogram in Fig. 2.8b as the frequency scale has been transformed from linear to logarithmic (mel). Both log-spectrograms and log-mel-spectrograms have been used widely

for audio recognition.

An incentive of training neural networks with spectrograms instead of the raw audio waveform is to facilitate training by providing an easier representation to the model; a model trained with raw waveforms would have to infer frequencies itself in the earlier layers of the architecture. This argument is supported by the findings of [93] which showed that 2D ConvNets operating on spectrograms outperformed 1D ConvNets operating on the raw waveform with a similar number of parameters. Furthermore, [93] proposed a 1D ConvNet for learning a spectrogram-like representation termed as a wavegram which was subsequently fed to a 2D network, further backing the argument that the earlier layers of an 1D network operating on raw waveforms would waste capacity in learning frequency representations. Based on this, in what follows we focus on 2D convolutional networks with spectrogram inputs for audio recognition.

2.2.1 Single-stream architectures

A common approach in auditory scene and event recognition is to use single-stream 2D convolutional architectures [2, 67, 75, 93, 138, 145, 156]. [145] and [156] proposed shallow architectures consisting of 2 and 3 convolutional layers respectively, for environmental sound classification using small-scale datasets. Hershey *et al*. [75] were one of the first to evaluate deep convolutional networks for large-scale audio classification using their own YouTube-100M dataset that consists of 100 million videos, while [75] utilised only the audio stream. They investigated slightly modified versions of AlexNet [96], VGG [169], InceptionV3 [184] and ResNet50 [71] with Inception and ResNet providing the best performance. They also offered a study on the effect of training set size in test performance, with boosts in performance by incorporating more data, although the benefits after 700K audio clips were negligible. A similar investigation with comparable findings was presented in [93], where the authors trained various deep architectures on AudioSet [55] and found that ResNets were the best performing models and that although utilising the full dataset improves the performance, the accuracy degraded only slightly with 50% of the training data. Single-stream 2D ConvNets have been extensively used by high-ranked entries of DCASE audio recognition challenges [29, 42, 78, 95, 113, 180] too, for acoustic scene classification.

Similarly to works on video recognition, attention modelling has been explored for audio classification as well [59, 60]. [59] enhanced the output feature maps of a ConvNet backbone with a simple temporal gating mechanism composed of parallel gating attention heads where each head performs temporal aggregation on the gated feature map, and the output of all heads are combined with learnable weighted averaging. The same authors introduced the Audio Spectrogram Transformer (AST) [60] that is essentially the ViT architecture [43] operating on spectrograms, *i.e*. a spectrogram is split into patches of size 16×16 which are embedded

and enhanced with positional encodings, and a learnable classification token is attached to the sequence which is fed to a transformer encoder. AST demonstrated the benefits of ViT-based architectures on the audio domain by establishing state-of-the-art performance against a top-performing convolutional approach.

Data augmentation. The small-sized audio datasets necessitate the utilisation of data augmentation techniques for combating overfiting while training high-capacity audio recognition architectures. Some works considered time stretching and pitch shifting [78, 138, 145, 156] as well as dynamic range compression [78, 156] and adding background noise from other audio samples [156]. [156] offered an analysis of the impact of each individual augmentation per class as well as for all classes and by combining all augmentations and showed that pitch shifting was the most beneficial augmentation and that combining augmentations provides additional boosts. [78] also showcased that combining augmentations has a positive impact in performance. Moreover, a popular temporal augmentation technique is to randomly crop an audio segment out of the full audio clip [42, 78, 156, 180].

Many works [42, 59, 60, 78, 93, 95, 180] have utilised mixup [214] for audio augmentation. Mixup [214] generates new training samples by randomly selecting two examples from the training set and computing a linear combination of both their data and labels using coefficients sampled from a Beta distribution. Although it was originally tested on image classification, it has proven a strong method for preventing overfitting in audio recognition. Note, that mixup can be applied either in the audio signal before extracting the spectrogram or on the spectrogram itself.

A simple but powerful data augmentation technique is SpecAugment [140] which has also been adopted by various works [59, 60, 78, 93]. As the name reveals, SpecAugment is applied directly on the spectrogram treating it as an image. It is composed of three deformations: (1) time warping, where a point sampled along the horizontal line that passes through the centre of the image is warped either to the left or right, using a warping parameter that defines the range for sampling the point and the distance that it is warped, (2) time masking that masks a block of consecutive timesteps using a time mask parameter to sample the size and the location of the mask, and (3) frequency masking that masks a block of frequency channels using a frequency mask parameter to sample the size and the location of the mask. Furthermore, two additional parameters are used to define the number of frequency and time masks. An illustration of SpecAugment can be seen in Fig. 2.9. Although SpecAugment has been proposed for Automatic Speech Recognition, it has provided solid performance improvements in other audio recognition areas too, including scene and event recognition.

Pre-training. Just as with image and video recognition, pre-training a model using a large-

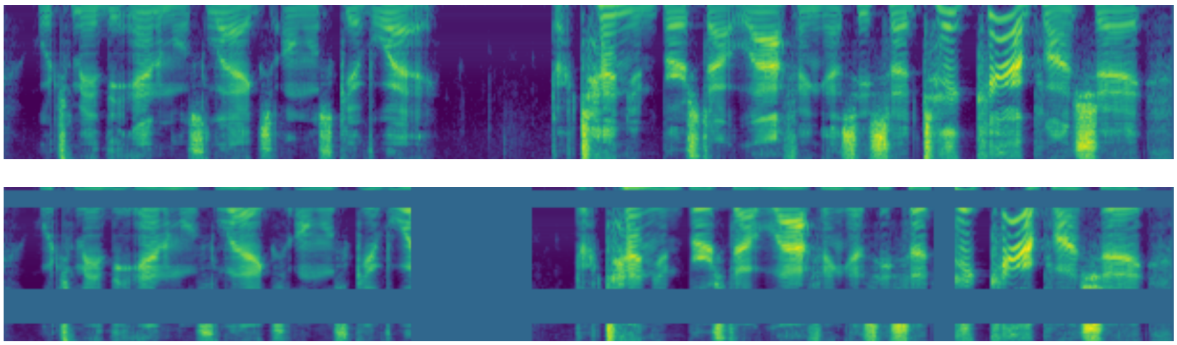


Figure 2.9: *SpecAugment [140]. (Top) Log-mel-spectrogram. (Bottom) Log-mel-spectrogram augmented with SpecAugment using two time and frequency masks. Also note how the spectrogram has been temporally warped to the right. The images are from [140].*

scale database and fine-tuning it for downstream tasks has been explored in audio recognition as well for transferring knowledge from large training sets to smaller ones [2, 59, 60, 67, 68, 93, 138]. Two methodologies have emerged: i) transferring within domain knowledge from large-scale audio databases [93] and ii) transferring knowledge from large-scale image databases, namely ImageNet [39], to the audio domain [2, 59, 60, 67, 68, 138].

Kong *et al*. [93] transferred a model pre-trained on AudioSet to 6 different downstream tasks including environmental sound classification using the ESC-50 dataset [146] and acoustic scene classification and audio tagging tasks from the DCASE challenge. For transferring, they explored two popular approaches: i) extracting features from the pre-trained model and training a linear classifier on top, aka linear probing, and ii) initialising the model with the pre-trained weights and fine-tuning it for the task of interest. Overall, they demonstrated that utilising the pre-trained model enhanced the performance of all downstream tasks and that fine-tuning outperforms linear probing. Nevertheless, two useful findings are that, first, in some cases (*e.g*. the DCASE tasks) linear probing is even harmful compared to training from scratch attributing it to the distributional shift between AudioSet and the task of interest, and secondly, that in some few-shot settings (*i.e*. using only a few samples per class for the downstream task) linear probing provides better results than fine-tuning.

While at a first glance exploiting image pre-trained models for audio recognition may seems counter-intuitive, it is advantageous as shown by [2, 59, 60, 67, 68, 138]. [68] was one of the first works that utilised a pre-trained ImageNet model for music genre classification while later [2] and [67] fine-tuned ImageNet pre-trained models for urban sound classification and environmental sound classification respectively, where in [67] pre-training provided considerable performance improvements.

Palanisamy *et al*. [138] also showed that ImageNet pre-training results in significant performance boosts in different audio classification benchmarks. More importantly, they studied systematically the impact of ImageNet pre-training in training auditory networks by posing the question of how much of the pre-trained ImageNet weights are useful for the considered audio recognition tasks. To answer this question they performed a number of different experiments. First, they measured how much the pre-trained weights had changed at different layers of the network after fine-tuning, where smaller changes were observed in initial layers while the changes in later layers were more significant. Secondly, they evaluated the impact on performance by progressively initialising more layers of the network with the pre-trained weights while keeping the rest randomly initialised and observed that pre-trained weights had a big positive impact up to intermediate layers of the network while after that pre-trained weights offered smaller benefits. Lastly, they've shown that freezing the early layers had a minor impact on the performance, while by freezing the upper layers the accuracy dropped significantly. From these experiments, the authors concluded that prior knowledge from the pre-trained model is particularly helpful for the earlier layers of the network as it is maintained after fine-tuning, and that the weights of the early layers are more transferable between between the vision and audio domains. Through feature map visualisation, they noted that the model learnt to perform event boundary detection by detecting edges in the spectrogram, further emphasising the importance of the early layers and the corresponding pre-trained weights for the auditory tasks. Interestingly, this confirms the hypothesis of [59] that ImageNet pre-training is beneficial because the pre-trained model learnt to detect edges which may assist the auditory model to detect 'acoustic edges', *i.e*. event boundaries. More interestingly, these findings are in contrast with works on audio-visual action recognition [207] as well as two-stream visual action recognition [88] that fuse the modalities after intermediate layers explicitly based on the intuition that earlier layers are modality specific and do not generalise across modalities.

2.2.2 Non-symmetric filtering of time and frequency

So far, we have seen approaches that utilise vanilla 2D convolutional networks for audio modelling. This has certain benefits. First, it enables training the networks with spectrograms that has shown to be more powerful than training with the raw audio [93]. Second, it allows leveraging ImageNet pre-trained models [2, 59, 60, 67, 68, 138]. Third, the increased receptive field of deep networks equips the models with long-term temporal modelling capabilities. Nevertheless, employing vanilla 2D convolutional networks entails the danger of not capturing adequately the prior structure of audio signals as spectrograms have some fundamental differences from natural images, stemming from the fact that both axes of natural images contain

spatial information whereas the time and frequency axes of spectrograms have different properties. The translation invariance of ConvNets is beneficial for the time dimension as a model should be able to recognise a sounding object independently of the temporal moment that it sounds, while it may be harmful for the frequency dimension where nearby frequency channels might encode entirely different entities. Moreover, the locality of convolutions is suitable for modelling the sequential nature of time but not for frequencies which can be non-local due to harmonics. Empirical evidence to support these claims is drawn from [215], where by looking at the frequency statistics of natural images and spectrograms, a heuristic analysis has shown that that the energy distribution of natural images is homogeneous across both spatial dimensions, while for spectrograms the energy is distributed differently across frequency and time.

These necessitate non-symmetric filtering over the time and frequency dimensions of spectrograms, which can be achieved by various means [78, 81, 122, 180, 207]. A common approach is to design architectures with different receptive fields in the temporal and frequency dimensions which is affected by different factors including the convolutional filter size, dilation and downsampling factors, *i.e*. convolutional stride and pooling layers. [81] employed rectangular $k \times m$ convolutional kernels as opposed to square $k \times k$ ones which are typically used in the image domain. [122] applied downsampling only in time and not in frequencies using a convolutional stride of 2 in time and a stride of 1 in the frequency dimension, resulting in a convolutional network that has a larger receptive field in frequencies. Similarly, [78] maintained the temporal downsampling while reducing the frequency downsampling, yet via the max-pooling layers instead of the stride. [180] designed a network with a larger receptive field over the frequency axis too, by combining convolutional layers with temporal-wise striding followed by frequency-wise dilated convolutional layers. A more principled way of modelling the structure of harmonic series was proposed in [215] that introduced harmonic convolutions but the model was applied only in audio restoration and musical source separation tasks but not in audio recognition. Lastly, [207] explored a different approach. To treat frequency and time independently, the authors used separable convolutions layers in parallel with $1 \times k$ and $k \times 1$ filters for the early layers of the network while they employed $k \times k$ filters for the later layers. However, the receptive field was kept identical over both frequency and time.

Each of these methods tackle the shortcomings of vanilla 2D architectures from different perspectives. While [81] and [207] allow modelling the non-homogeneous energy distribution of spectrograms via using non-square kernels and separable convolutions, respectively, they do not explicitly tackle the translation invariance and the locality of convolutions in the frequency axis. On the other hand, [122] and [78] explicitly address the translation invariance in the frequency axis by reducing the pooling over frequencies. [180] is the only work that in addition

to reducing the translation invariance over the frequency axis via reducing frequency-pooling, it also employs dilated convolutions over the frequency axis which can account for the non-local structure of harmonics. Finally, while the approach of [215] for modelling the harmonic structure of frequencies is more sophisticated, its effectiveness in audio classification has not yet been evaluated.

2.2.3 Multi-stream architectures

Leveraging complementary information from multiple information streams has not been explored only for videos, but it is a viable learning paradigm for audio too where multi-stream architectures have been proposed for environmental sound classification [106] and acoustic scene classification [19, 122, 205]. These can be divided in two categories: i) approaches that utilise modality-specific streams [19, 106] and ii) approaches where all streams digest the same input [122, 205].

[106] introduced a three-stream architecture operating on the raw audio, concatenated spectrograms of different frequency resolutions, and delta features for each spectrogram, respectively. The raw audio stream was composed by 1D convolutional layers followed by 2D convolutions whereas the spectrogram/delta streams incorporated 2D convolutions throughout. The streams were fused with late fusion of their output feature maps and further integrated with multi-level fusion in the form of attention using a temporal gating mechanism. [19] considered different input representations including mel-spectrograms, log-mel-spectrograms, and Mel Frequency Cepstral Coefficients (MFCC) and combined them in a multi-stream architecture by sharing the weights across streams. Similarly to [106], the streams were integrated with both multi-level attentional fusion and late fusion.

The second line of works investigate capturing complementary auditory information from a different standpoint; instead of incorporating different modalities, the streams observe different 'views' of the same input. In [122], the inputs were split in half along the frequency axis and the first half was fed to one stream while the second half to another, aiming to learn frequency-specific streams that can learn to reason independently about low frequency and high frequency scenes/entities. The output feature maps of each stream were concatenated on the frequency axis and their correspondences were learnt with two additional convolutional layers with 1×1 kernels. [205] incorporated three streams and applied median filtering with different kernels at the input of each stream to model long duration sound events, medium, and short duration impulses separately. The streams were fused by concatenating their outputs and feeding them to fully-connected layers akin to [122].

While in [122, 205] each stream acts on complementary views of the same input, the architec-

ture of each stream remains identical. In other works, complementary information is sought through stream-specific network designs and streams operate on identical inputs. One example is that of [69], where 1D convolutions were used with different dilation rates at each stream to model convolutional streams that operate on different temporal resolutions, and the approach was tested on speech recognition. Ch. 4 proposes a two-stream auditory architecture that combines both of these approaches. That is, the streams consider different views of the same input where one of them uses a subsampled version of the original spectrogram, while at the same time the architecture adopts a stream-specific design with varying number of channels and temporal resolution at each stream to facilitate one stream at focusing in frequency while the other in temporal patterns. Moreover, based on the observations of the previous subsection the architecture employs convolutional separation for non-symmetric filtering in the frequency and temporal axes.

2.3 Audio-Visual Action Recognition

While most action recognition approaches focus on spatio-temporal modelling with visual modalities, audio is a commodity that needs to be leveraged for alleviating the complexity of understanding human actions. Audio can be helpful for recognising visually occluded actions but audible, complementing the visual stream. Moreover, the simultaneity of vision and audio can advance action understanding by learning corresponding patterns across modalities, *e.g.* for the action 'clap hands' it would be useful to associate the movement of the hands with the sound of clapping. This necessitates models able to correlate visual and auditory signals to capture both the redundant and complementary information in the two modalities. Typically, these models take the form of multi-stream architectures with one or more visual streams and an audio stream operating on spectrograms. One question that naturally arises is, how to encode each modality, and what is the optimal fusion and training strategy for audio-visual integration? In the attempt to answer these questions, three subfields of research have emerged. The first focuses on the design of audio-visual action recognition architectures, with relevant approaches being reviewed in Subsec. 2.3.1. The second is concerned with the development of audio-visual regularisation techniques, presented in Subsec. 2.3.2. Lastly, Subsec. 2.3.3 discusses the third area that consists of methods aimed at learning audio-visual representations without human annotations and fine-tuning them with action recognition training objectives.

2.3.1 Audio-visual architectures

Inspired by the two-stream fusion paradigm of [168], and also acknowledging the limitation of existing deep learning architectures in employing audio for action recognition, [206] proposed an audio-visual fusion approach using three modalities, appearance, motion and audio. Three independent convolutional networks were used for modelling short-term information within each modality, and two LSTMs for capturing long-term temporal information by aggregating the outputs of the frame-level appearance and motion ConvNets, yielding a total of five streams. For combining all five streams, the authors proposed a late fusion scheme acting on the streams' predictions. They noted that the traditional late fusion with averaging assumes that each stream contributes equally to each class, and thus does not consider the relationships between modalities and actions, where the appearance stream might be more efficient in describing classes associated with objects whereas the motion stream can reason better for classes that entail the movement of the actor. To tackle this, a fusion scheme was devised to learn the contributions of each stream to the final action scores where at a first stage each stream was trained independently, and at the second stage the predictions of each stream were concatenated and a fusion weight matrix mapping from the stacked multi-modal predictions to the final action predictions was trained with logistic regression.

[116] and [115] also considered the integration of appearance, motion and audio using attention-based mechanisms and operating on pre-extracted features for each modality. In [116], a 'Keyless Attention' mechanism was proposed to aggregate the temporal features from the output of an LSTM into a global representation by attending on salient video parts. 'Keyless Attention' simply projects the temporal inputs followed by a softmax that produces temporal attention weights which are used to pool the features through a weighted linear combination. The authors adopted one LSTM per modality and investigated fusion at different levels of the architecture, where they found that best results were achieved with mid-level fusion after temporal aggregation within each modality, *i.e.* by concatenating the pooled LSTM features of each modality. They attributed this to the ability of the model to attend to different parts of the video for each modality, in contrast to fusing before the attention layer where the model is restrained to attend to the same temporal moments for all modalities. Nevertheless, the gains of one fusion technique over the other were negligible.

[115] adopted a mid-level fusion approach as well and applied attention directly on the outputs of RGB, Flow and Audio convolutional networks, based on the observation that long-term information (*i.e.* utilising an LSTM like [116]) is superfluous for short trimmed videos. The proposed architecture is similar to [116], in that there is an attention-module within each modality exploiting simple temporal attention functions that learn attention weights using a softmax over projected temporal features. Moreover, the aggregated multi-modal representa-

tions were concatenated and fed to a linear classifier. The novelty in [115] is the introduction of Attention Clusters that consist of multiple attention units with independent parameters that attend to the video in parallel to enable more diverse attention signals by allowing each attention unit to focus at different parts of the video (similar in motivation to multi-head attention in Transformers). The output of an Attention Cluster is the concatenation of the pooled representation from each attention unit. To encourage diversity amongst attention units, [115] introduced a shifting operation that scales and shifts the output of each attention unit with learnable parameters.

Xiao *et al*. [207] considered the problem of fusing vision with audio at multiple representation levels to form hierarchical audio-visual concepts. To this end, the SlowFast architecture [47] for visual action recognition was enhanced with an additional ResNet-based audio stream tailored to perform audio processing. That is to say, the audio stream incorporates parallel separable convolutions up to intermediate layers for non-symmetric filtering in frequency and time while using square filters for the rest of the architecure. In addition, the first convolutional layer does not subsample the auditory features and there is no pooling layer afterwards to maintain high-resolution time-frequency representations at the beginning of the architecture. Symmetric pooling over frequency and time is applied progressively, once at each residual stage with strided convolutions. The concept of lateral connections was adopted from [47] to integrate audio features to the Slow and Fast streams at different layers of the architecture. An overview of the architecture can be seen in Fig. 2.10. The authors found that is more beneficial to incorporate lateral connections from audio to vision only from intermediate layers of the architecture and above, attributing it to the fact that low-level features are not transferable between modalities, *e.g*. edges do not have a distinctive sound pattern. Three different types of audio-visual lateral connections were investigated which are illustrated in Fig. 2.11: i) first audio is integrated with Fast followed by temporal downsampling and concatenation with Slow (A$\rightarrow$F$\rightarrow$S) imposing a strict temporal alignment of the modalities as audio and Fast are fused with a high temporal precision, ii) SlowFast fusion precedes audio-visual integration (A$\rightarrow$FS), relaxing the temporal alignment requirement between the two modalities, as audio is downsampled to match the coarse temporal resolution of the fused SlowFast representation, and iii) audio-visual Non-Local blocks [199] to attend spatio-temporally to the event using audio as a query. The authors concluded that the fusion technique that allows loose temporal alignment of modalities is beneficial as it can account for temporally displaced semantics in the two modalities, *e.g*. the sound of a door closing is heard after the motion generated by pushing the door.

[53] leveraged the congruence of audio and video from a novel point of view, where the goal was to reduce the clip-level and video-level visual redundancies in untrimmed videos for ef-

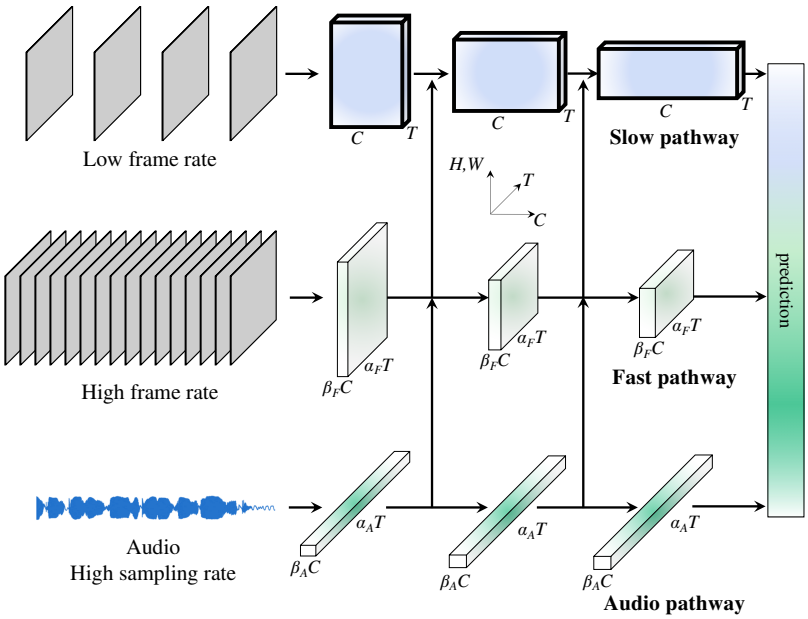


Figure 2.10: *The Audio-Visual SlowFast architecture. The visual SlowFast architecture from [47] is complemented with an additional ResNet-based audio stream. The streams are fused at multiple representation levels to learn hierarchical audio-visual features with lateral connections from the audio stream to the Fast and Slow streams. Finally, the pooled features of all three streams are concantenated and fed to the classifier. The figure is from [207].*

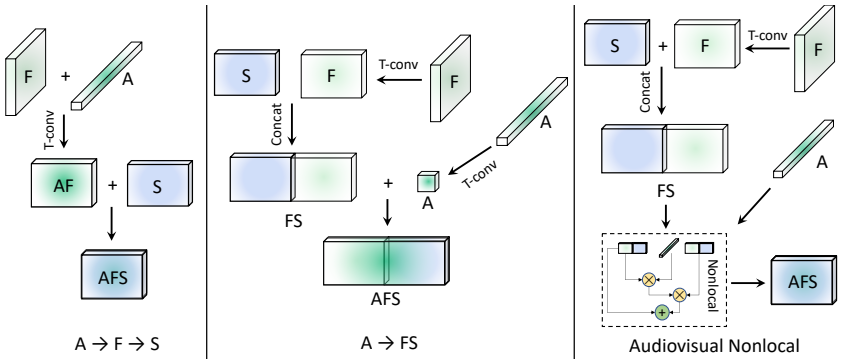


Figure 2.11: *Types of lateral connections of AVSlowFast. (Left) The audio stream is first combined with the Fast stream which are then downsampled with a strided convolution and integrated with Slow. This scheme enforces a stringent audio-visual temporal alignment as features from audio and Fast have a high temporal resolution. (Middle) Fast is subsampled and combined with Slow (as in [47]) which are then fused with a downsampled version of the audio representation. This approach allows a coarser temporal alignment between modalities. (Right) Non-Local blocks [199] are used to perform audio-visual cross-attention using audio as the query and the visual SlowFast features as keys/values. The figure is from [207].*

ficient action recognition, by exploiting audio as an efficient preview of video. Short-term redundancies arise from temporally adjacent frames being visually similar, while long-term

redundancies can occur by events that are repetitive or temporally localised within a video, making the utilisation of the full video unnecessary. For clip-level redundancies, the expensive processing of all frames within a clip was replaced with a cheaper module that processes a single frame and the accompanying audio, based on the observation that a single frame contains sufficient appearance semantics while audio captures the temporal dynamics of the clip. Concretely, a teacher-student distillation was proposed, where the student is a two-stream network that takes the first frame of a clip and the corresponding audio and learns to approximate the distribution of a 3D teacher ConvNet that operates on clips. Then, an attention-based LSTM was introduced to iteratively attend to informative moments in the video, reducing video-level redundancies by skipping irrelevant video parts. Using the learnt two-stream student model as the backbone, the LSTM makes predictions for a significantly smaller number of steps than the total number of frame-audio pairs, where at each step two independent attention mechanisms softly select the relevant frames and audio, respectively, to form the inputs for the next LSTM step. The overall method offers a good trade-off between accuracy and speed at a significant lower computational cost than using a clip-based model, while audio provides considerable accuracy gains, particularly for dynamic scenes. Interestingly, similarly to [116], in this paper there is also one attention module per modality rather than a single attention module operating on concatenated audio-visual features to allow the model to attend to different moments within each modality.

After the wide adoption of Transformers in action recognition, a natural question that arises is how could these models be extended to the audio-visual setting and what are the best practices considering the idiosyncrasy of visual and auditory information. Nagrani *et al*. [133] offered an attempt to answer these questions and proposed an audio-visual transformer-based architecture for action and event recognition. The authors explored end-to-end trainable architectures from raw videos and spectrograms which are tokenised, adopting the paradigm of ViT and AST [60] (discussed in Sec. 2.2), respectively. They argue that attention across all visual and auditory tokens and at all layers of the model is unnecessary due to the redundancies in visual and auditory inputs. Furthermore, the quadratic complexity of self-attention would make infeasible the applicability of such model in long videos. To tackle these, they introduced two strategies for restricting the attention flow across modalities while allowing full attention within modalities. The first follows the standard approach in multi-modal fusion, where modalities are fused at a mid-level with uni-modal blocks at the earlier layers of the architecture that focus on learning modality specific representations. Accordingly, [133] adapted this strategy by employing uni-modal Transformers in the earlier layers, namely a ViT-based architecture for video frames and an AST-based architecture for spectrograms, after which cross-modal attention layers are added to restrict audio-visual attention to later layers of the architecture. The second strategy, the key insight of [133], is to restrict cross-modal communication within

a layer by introducing a set of latent units, the fusion bottleneck tokens, that form an attention bottleneck through which cross-modal attention must flow. A fusion bottleneck layer is composed of two transformer layers, one per modality, that operate on the concatenation of the modality's tokens and a small set of learnable bottleneck tokens shared across modalities. Each transformer layer is producing intermediate bottleneck tokens per modality which are averaged to form the final fusion tokens fed to the next layer. Thus, the bottleneck tokens encode audio-visual information and cross-modal attention is achieved by comparing one modality's tokens to the bottleneck tokens rather than to all tokens of the other modality. Intuitively, by incorporating a smaller number of fusion tokens than the number of per-modality tokens, the model is forced to compress and distribute the most discriminant information through the bottleneck tokens while at the same time reducing the computational complexity comparing to full cross-attention between modalities. The proposed model is called Multimodal Bottleneck Transformer (MBT). The authors compared MBT with vanilla cross-attention that employs unique cross-attention blocks per modality yet without fusion bottleneck tokens. It was shown that for both strategies mid-level fusion outperformed early or late fusion, with MBT providing a better performance overall comparing to vanilla cross-attention. Nonetheless, the benefits of MBT over vanilla cross-attention were preeminent particularly when fusion was applied at earlier layers of the architecture, where the gap in both the accuracy and the computational cost was larger. Finally, it is worth noting that only four bottleneck tokens were sufficient to achieve good performance.

2.3.2 How to regularise audio-visual networks?

The ever larger and data-hungry deep learning architectures come at the price of overfitting, where researchers strived to propose solutions, such as Dropout [174], to regularise training. A reason for overfitting in neural nets is that there are complex co-adaptations between units in hidden layers of a network, *i.e.* a unit learns to correct the mistakes made by other hidden units which results in complex relationships between hidden units by memorising the noise in the training set.

The multi-modal regime is more convoluted, with additional sources of overfitting:

1. In addition to co-adaptations of units within modalities, cross-modal co-adaptations occur as well, resulting in networks that do not perform well in the absence of one or more modalities [135]

2. The training dynamics of different modalities are not compatible, where different modalities train and overfit at different rates hurting performance [196, 207]

3. Higher uncertainty in the predictions of one of the modalities [177]

From the works presented in this subsection only [177, 196, 207] are on regularisation for audio-visual activity recognition. Although [135] proposes a method for gesture recognition, it is reviewed in this subsection for a comparison with [207] as they both propose an approach for multi-modal regularisation that belongs to the same family of methods.

[177] highlights that existing audio-visual action recognition frameworks are prone to errors in the presence of noise in individual modalities as they do not consider uncertainty in the modalities' parameters which leads to higher uncertainty in the predictions of the noisier/less informative modality. To tackle this issue, the authors propose a hybrid Bayesian audio-visual ConvNet with stochastic variational inference for action recognition, that combines well-established deterministic layers with variational layers. Namely, the backbones of both the visual and audio networks are deterministic convolutional layers, while the final fully-connected layers are variational layers. Variational layers are parameterised with a mean μ and a variance σ^2 to approximate the complex posterior over the parameters $p(w|D)$ (D is the training set) with a simpler distribution $q_\theta(w) = \mathcal{N}(w|\mu, \sigma^2)$. This procedure allows to measure the uncertainty in the predictions, and the authors propose to average the predictions of audio and vision if the uncertainty measures are below a threshold, otherwise only the predictions of the modality with lower uncertainty are considered. Furthermore, Bayesian neural networks regularise training by capturing uncertainty in the network's parameters, and therefore in [177] the more uncertain modalities are more heavily regularised. Although the method has appealing attributes regarding its ability to regularise audio-visual networks as well as to omit uncertain modalities from the final predictions, it has a basic drawback: it is computationally costly as it requires multiple Monte Carlo passes through the variational layers by sampling from $q_\theta(w)$ to obtain the final predictions.

The authors of [196] recognise two main reasons of overfitting in multi-modal networks. First, different modalities have different generalisation/overfitting rates and such discrepancies can cause joint optimisation schemes to underperform. Second, the increased capacity of multi-modal architectures make them further prone to overfitting. Demonstrated by the experiments that compare an audio-visual action recognition model with the video-only and audio-only models, there is indeed a discrepancy in the overfitting rates between the two modalities where the audio model overfits more than video. Importantly, the audio-visual network, trained without multi-modal regularisation, has lower training error and higher validation error than the video stream alone, causing even a drop in the accuracy of the audio-visual fusion model compared to using only video.

To tackle these training difficulties, the authors first propose a measure, namely the overfitting-to-generalisation ratio (OGR), to quantify the quality of training between model checkpoints. More precisely, between two versions of a model (at epoch N and $N + n$, respectively) OGR

is calculated as the change in overfitting divided by the change in generalisation, where overfitting is defined as the gap between the training loss and a held-out validation loss, while generalisation is simply measured by the validation loss. Low OGR means that overfitting has dropped while generalisation has increased, comparing two checkpoints of the model, and as a result training has arrived at a better solution. Having defined OGR, the authors then set the objective to minimise it, by blending the gradients of different modalities such that each optimisation step performs at least as good as the best individual modality. To this end, a closed-form solution of the optimal blending weights is provided which incorporates OGR. These weights are then used to blend the losses of individual modalities (and thus their gradients) as a weighted sum. This strategy essentially adjusts the generalisation/overfitting rates of different modalities by making each modality aware of the total OGR of all modalities, through a normalisation term in the weight computation. Practically, to calculate one loss per modality, classifiers are attached at each modality. The authors propose both an offline version of Gradient-Blending where the optimal weights are calculated only once, and an online version where the weights are periodically updated. Both versions improve the performance of the audio-visual network with online gradient blending providing some extra boosts over the offline version. The method is tested in various benchmarks including egocentric action recognition in EPIC-KITCHENS improving over the visual-only networks as well as the audio-visual networks trained without Gradient-Blending. One drawback of the approach similarly to the Bayesian DNN in [177] is its increased computational overhead resulting from the computation of the weights particularly in the online version. Below, more computationally efficient audio-visual regularisation schemes are presented.

Dropout can be applied at any layer of a neural network, and during training it randomly removes subsets of units of a layer by masking them out. Multi-modal networks and their input samples are grouped in modalities, and there are high correlations within modalities. By applying vanilla dropout at each modality, such correlations are overlooked [101, 135]. The works of [135] and [207] capitalise on the prior information of modality groupings and the within-modal correlations proposing dropout-variants compliant with the multi-modal setting. Differently than classical dropout, these approaches either drop or retain the entire set of features within a modality simultaneously, at a given part of the architecture. These approaches differentiate according to whether a modality is dropped and at which part, from which different regularisation properties emerge. Similar ideas of leveraging the structuring of inputs have also been explored for object localisation [185], where contiguous units in feature maps are either all dropped or retained.

The basic motivation of [135] is to hinder co-adaptations between modality representations, *i.e.* learn robust representations in one modality that do not depend on the features of other

modalities with the aim to handle missing or noisy modalities in testing. While vanilla dropout can be viewed as training an exponential number of neural networks, where each network is sampled from the original neural network by dropping out units, the authors exploit the grouping of modalities, casting the problem as training all the possible combinations of modalities, where each combination is sampled by dropping out modalities. Accordingly, the proposed objective function constitutes of the losses of all possible modality combinations. As it is computationally expensive to train the network with this objective function, it is approximated by randomly training one term per iteration. This is realised by randomly dropping the modalities *inputs*. That is, given multi-modal inputs x_m, $\forall m \in [1, .., M]$, where M is the number of modalities, each modality input is masked as $\delta_m x_m$, where δ_m is a Bernoulli random variable, and dropped with probability $q_m = 1 - p_m$ ($\delta_m = 0$). This method is termed as ModDrop. The regularisation properties of ModDrop are studied using a single-layer fully-connected network with binary cross-entropy loss. It is shown that the error of the network trained with ModDrop is proportional to the error of the full network (*i.e.* no dropping of modalities) minus a term that expresses cross-modality correlations. Consequently, when modalities inputs are uncorrelated the second term tends to vanish and ModDrop does not have a significant impact in training. As analysed by the authors, when modalities are correlated this second term in the error performs cross-modality regularisation by adjusting the network's weights to align the multi-modal signals. Aligning modalities can make the model robust to missing signals as multi-modal inputs are embedded in the same space, and the model can infer redundant information from all modalities, even in the absence of some of them. As the sole purpose of ModDrop is to align modalities, it might not be able to tackle other important sources of overfitting, discussed earlier. This is manifested in the experiments, where although ModDrop successfully alleviates the effect of missing modalities fulfilling the authors' motivation, it does not improve the performance when all input signals are present.

AVSlowFast exploits additional prior information to perform audio-visual regularisation. As shown in Fig. 2.12, the audio stream converges/starts overfitting significantly faster than the visual network. The authors claim that this incompatibility in training speed leads to overfitting in the audio-visual network. Note that, this is very similar with the observation made by [196] but with a focus on the *momentum* of overfitting, while [196] is more concerned with the *amount* of overfitting in each modality. The overfitting induced by such discrepancy is showcased in the experiments where similar to [196] the audio-visual network performs worse than the visual-only model when audio-visual regularisation is not applied. Accordingly, in addition to the prior knowledge of grouping of modalities, the authors take it to the next step by also leveraging the observation that audio overfits earlier, and introduce the idea of dropping *only audio* to adapt its learning pace to that of its visual analogue, calling this strategy as DropPathway. In other words, the main goal of DropPathway is to slow down the training of the

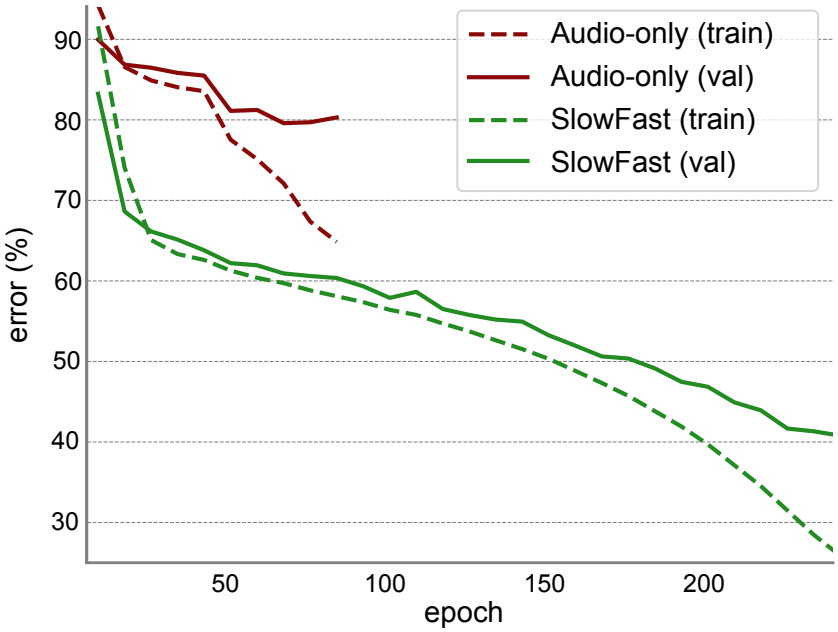


Figure 2.12: *Training/validation curves on Kinetics comparing SlowFast with an Audio-only network. Top-1 errors are demonstrated. The audio network starts to overfit after approximately* $3\times$ *fewer training iterations than the visual, demonstrating the discrepancy in learning pace of the two modalities. The figure is from [207].*

audio pathway by not updating its parameters, randomly when the audio network is dropped. More concretely, DropPathway drops randomly the *auditory lateral connections* of the AVSlowFast architecture, altogether, by multiplying them with a Bernoulli random variable, δ_A. Thus, when $\delta_A = 0$ AVSlowFast does not perform multi-level audio-visual fusion and the audio stream is connected with the visual streams only via late fusion in the prediction layer (see Fig. 2.10). This has a positive impact on generalisation in two ways: 1) it decreases the training momentum of the auditory lateral connection parameters, adjusting the learning dynamics of the two modalities to a certain degree and 2) it encourages the learning of robust visual and auditory intermediate representations that do not rely on each other, similar to the motivation of [135], as randomly removing their links prevents them from co-adapting. Results verify the effectiveness of the method improving audio-visual classification compared to the visual-only model, demonstrating that asymmetrically dropping only audio information is beneficial for audio-visual action recognition. Nevertheless, DropPathway has some weaknesses which are discussed in Ch. 4 where a simple solution to address them is proposed.

Comparing all the regularisation schemes presented in this subsection, DropPathway is the most efficient in terms of both accuracy and computational cost. [177] and Gradient-Blending are both computationally demanding, however Gradient-Blending requires increased computational budget only to compute the blending weights in training, whereas [177] requires multiple Monte Carlo runs in inference as well, and thus is the most computationally inefficient.

Moreover, Gradient-Blending increases the accuracy adequately in popular audio-visual action recognition benchmarks. ModDrop and DropPathway are both in the family of dropping modalities in training, and both are cheaper than Gradient-Blending. Nevertheless, ModDrop can only handle missing modality signals while it does not improve performance when all modalities are present. A possible cause is that ModDrop does not address the incompatibility in learning dynamics of different modalities. DropPathway both increases the action recognition accuracy and has a minimal computational budget, addressing the disadvantages of both ModDrop and Gradient-Blending, respectively. Finally, it is interesting to compare the online version of Gradient-Blending and DropPathway. In a similar fashion to DropPathway that slows down the learning of the audio stream, Gradient-Blending maintains a lower weight for the audio loss than the visual one throughout training. Interestingly, at the second half of training, where audio starts to overfit, Gradient-Blending further increases the visual loss weight. Hence, while DropPathway consistently trains the audio network at a slower pace, online Gradient-Blending tackles the problem from the opposite direction and adaptively speeds up the learning pace of the video network.

2.3.3 Going beyond action supervision

The advent of self-supervised learning brought the significance of learning effective representations without human-label supervision at the forefront of the machine learning and computer vision communities. The trend was adopted in audio-visual learning too when it came to the attention of researchers that the natural synergy between vision and audio can provide a strong supervisory signal for learning powerful audio-visual representations useful for many downstream tasks including action recognition. Numerous works proposed various forms of audio-visual self-supervision tasks building on the intuition that by leveraging the co-occurrence of audio and vision, a model can be trained to comprehend the common underlying causal factors that produce the auditory and visual events. Although self-supervised audio-visual representation learning is out of the scope of this thesis, below some of the most influential advances in the field are reviewed as they paved the path for learning to recognise actions audio-visually without human supervision.

The works of [6, 7] capitalised on the natural correspondence between audio and vision and introduced the audio-visual correspondence task, *i.e.* training a network to recognise whether a video frame and an audio clip correspond, a binary classification problem. Positive pairs were sampled from the same moment in a video while negative pairs were sampled from different videos. The authors argue that a model can solve this problem only by learning semantic concepts in both modalities. The learnt model in [6] was tested for downstream classification tasks including audio and image classification with competitive performance compared to previous

self-supervised approaches. The follow-up work of [7] showcased that specially designed architectures for the tasks of intra-modal and cross-modal retrieval as well as for the task of sound source localisation, that is to visually detect the sounding object, can be effectively learnt simply by employing the audio-visual correspondence task as the objective function. Although [6, 7] did not test the trained models for action classification, they sparked the interest of a series of works that utilised tasks of predicting whether visual and auditory signals coincide and evaluated the learnt representations in action recognition. These are described next.

Owens and Efros [137] explored the task of training a neural network to predict whether visual and auditory inputs are synchronised a problem that has been studied before as a goal in its own right [33, 121] whereas in [137] it is utilised as a proxy task for learning audio-visual representations. The task is formulated by sampling video clips, where in half of them the visual and auditory streams are temporally aligned while in the rest audio is temporally shifted by a few seconds. Note that this is a more subtle task than predicting audio-visual correspondence [6, 7]. In [6, 7], negative pairs are from different videos and the visual inputs are single frames. Hence, the problem can be solved by associating semantics in the two modalities, *e.g.* the appearance of a car with the sound that it makes while distinguishing from the sound of other objects. On the other hand, for detecting temporal misalignment, where the negative pairs always come from the same video, the model should be able to reason about motion. Therefore, [137] used a 3D convolutional network operating on video clips. Moreover, the authors suggested that solving this task requires associating low-level features between the two modalities and accordingly proposed an early fusion scheme.

Korbar *et al.* [94] proposed a curriculum learning strategy, effectively combining the tasks of [6, 7] and [137]. At the first stage of the curriculum, the audio-visual architecture was trained only with easy negatives where the visual and audio samples are from different videos, *i.e.* solving for audio-visual correspondence. After a number of training epochs, the second stage introduces both easy and hard negatives in training, where hard negatives are temporally misaligned samples from the same video, *i.e.* predicting temporal synchrony akin to [137]. Similarly to [137], a 3D ConvNet was employed for the visual stream. Furthermore, the network was trained with a contrastive loss acting on the representations of the visual and auditory streams (ergo no fusion layer), as opposed to [6, 7, 137] that incorporated a cross-entropy loss and fused the streams.

[4] introduced MultiModal Versatile networks (MMV) to align vision, audio and language through self-supervised pre-trainin, where modalities were embedded in a common space with projection heads implemented as MLPs. As text was obtained using Automatic Speech Recognition (ASR), the authors avoided constructing an audio-text embedding space and did not

enforce their alignment through the loss either, to deter the network from learning ASR and encourage it associating words with the sounds produced from the relevant objects. Thus, audio was implicitly associated with text through its common embedding with vision. Furthermore, motivated by the observation that vision and audio have a more fine-grained structure than text, they proposed a modality embedding graph, namely Fine and coarse spaces (FAC). In FAC, visual and auditory modalities are commonly embedded in a fine-grained space, whereas a coarse-grained space embeds text, audio and vision using a fine-to-coarse projection to embed vision and audio from the fine to the coarse-grained space. MMV is trained with a contrastive loss that is composed of two components, an audio-visual contrastive loss and a visual-text contrastive loss. Similar to [6, 7], to for the positive pairs the modalities are sampled from the same temporal moment in the video while negative pairs are sampled from different videos.

Patrick *et al*. [142] noted that proposed methods for image representation learning like [23, 31, 73] are aimed at learning representations invariant to various transformations (*e.g*. rotation) which might not be optimal when considering video transformations, for example video representations should be distinct rather than invariant to time reversing a video as the semantics of the video might change (*e.g*. open vs close). To address this, they proposed a framework for composing data transformations to form training batches using a hierarchical sampling strategy. In the proposed framework all types of operations used for generating a batch for training, including sampling an example from the dataset, were expressed in terms of data transformations. For videos, five types of transformations were considered: (1) selecting a video from the training set, (2) temporal shift, (3) modality slicing, *i.e*. splitting the video in visual and audio features, (4) time reversal, (5) visual and auditory data augmentations. This formulation allowed the authors to express invariance and distinctiveness to transformations through an additional term in the contrastive loss function. For example, a representation should be distinctive to (1) to push representations from different videos apart (*i.e*. negative sampling). Moreover, the authors assumed invariance to (3) to embed modalities in a common space similarly to [4, 7, 94] and to (5) as is generally the case in contrastive learning. For (2) they explored both invariance [6, 7] and distinctiveness [94, 137], as well as for (4), and showed that a mix of distinctiveness and invariance provides optimal results for these transformations.

Alwassel *et al*. [5] introduced cross-modal clustering as a self-supervision task, where features extracted from each modality were clustered independently with k-means and the cluster assignments from one modality were used as pseudo-labels to train the other modality. The feature encoders for each modality were initialised randomly and the training process consisted of alternating between generating pseudo-labels with clustering and training the modality en-

coders. Asano *et al*. [11] considered audio-visual clustering in a self-supervised manner too, with the main goal being to assign semantically meaningful labels to unlabelled videos, yet the learnt representations were also evaluated for action recognition. The training procedure in [11] amounts to alternating between clustering and training the modality encoders using the clusters as labels similar to [5]. Different from [5], each modality network is trained independently with modality-invariant labels. This is implemented by adopting the formulation of [142] that considers modality slicing as a data transformation and marginalising over the transformations, *i.e*. averaging the visual and auditory representations. The averaged representation is used for clustering that produces the same pseudo-labels for both modalities.

Concluding remarks

In the realm of architecture design, the early attempts of [115, 116, 206] were not end-to-end trainable but relied on pre-extracted features which was prohibiting the errors from the multi-modal network from backpropagating in the uni-modal backbone architectures, limiting their representation capabilities. Nevertheless, this was addressed by the more modern approaches of [133, 207] which were trained end-to-end. Furthermore, the problem of semantic-level audio-visual asynchronies has been acknowledged among different works [53, 116, 207], an issue that was also discussed in Subsec. 2.1.4 for visual modalities, where [112] introduced asynchronous fusion with predefined asynchrony offsets between two modalities. Ch. 3 builds on these findings and proposes an end-to-end trainable architecture with asynchronous fusion of appearance, motion and audio streams where the constraint of [112] is relaxed and fusion is allowed from any random offset within a temporal window, which is more suitable for scaling up to many modalities.

The works on audio-visual regularisation and self-supervision signify that in addition to designing domain-specific networks, multi-modal inductive priors can be exploited for training the architectures too, where for instance the regularisation technique of [207] utilises the prior knowledge that inputs are grouped in modalities and that audio overfits faster, while the audio-visual self-supervised methods take advantage of the simultaneity of audio-visual inputs.

2.4 Egocentric Action Recognition

Egocentric videos are recorded with cameras, such as GoPro, attached to the forehead of the person that performs the action, *i.e*. the wearer, offering an 'egocentric viewpoint' that is similar to what the person sees. Different from third-person vision, egocentric videos mostly capture the hands of the wearer. At the core of egocentric vision is the interaction of the actor with the world and the manipulation of objects towards achieving a goal. Concretely, egocentric actions encapsulate interactions of hands with objects, and the first-person receptive field

focuses on the vicinity of the hands which are often around the centre of the image. Egocentric actions are commonly modelled as combinations of verbs and nouns, where the verb denotes what the person is doing, *i.e*. the action, and the noun corresponds to the object that the action is applied. Some examples are 'chop garlic', 'wash plate', 'open fridge' and 'stir pasta'. Appearance and motion information are critical for the egocentric domain too, to reason about the action and the object, respectively.

A major challenge of egocentric actions is that they are particularly fine-grained. First and foremost, in contrast to third-person actions that have a distinct motion signature ('brush teeth' versus 'walk'), egocentric actions consist of subtle hand movements. Second, applying the same action to different objects results in two different actions ('cut tomato' versus 'cut cucumber'). Third, the same action can be performed in a completely different manner depending on the object of interaction ('open cupboard' versus 'open bag'). Thus, undoubtedly an egocentric action recognition model should be able to focus on fine-grained hand motion and objects as well as to model their interactions. Another challenge of egocentric action recognition is that hands may occlude the manipulated object making arduous its prediction or that one hand may occlude the other resulting in ambiguity about the action being performed. Finally, abrupt head movements make the motion modelling demanding as the model has to learn to ignore camera motion and focus on hand motion.

Accordingly, researchers have proposed specialised architectures to capture the intricacies of the egocentric domain, by adapting the two-stream paradigm to encode hand-object interactions or by enhancing it with additional egocentric streams. Another prevalent approach is to employ attention mechanisms to focus around hands and objects using gaze supervision or without. This section analyses these topics as well as emerging ones deemed to be promising for pushing the frontiers of egocentric vision.

2.4.1 Hand-object interactions

The central role of hand-object interactions in understanding egocentric actions prompted researchers to investigate methods capable of reasoning about hands, objects and their interactions. Ma *et al*. [120] introduced a model in that direction that is also one of the first deep learning approaches for egocentric action recognition. A specialised network design for egocentric vision was proposed operating on interacted objects, by first localising and then recognising the object of interest. Based on the importance of hand pose for understanding interactions and on the observation that objects are typically in the proximity of the hands, the authors first trained a hand segmentation network to enable attention around the hands. Subsequently, an object localisation network was initialised from the hand segmentation network and fine-tuned by regressing the 2D heatmap of the location of the object of interest. Finally, objects were

cropped from the images using the detections from the previous step, and the cropped images were fed to an object recognition network. Feature map visualisation showed that particular neurons were firing at hand regions and object attributes, such as shape and texture, enabling the modelling of hand-object configurations.

In [13], the problem of modelling hand-object and object-object interactions is investigated under the scope of object-level relational reasoning, *i.e*. learning relations between pairs of objects. To this end, Object Relation Network (ORN) was introduced to relate detected objects through space and time. ORN resembles the model of [120] in that both employ object detectors to infer interactions. Nevertheless, ORN utilises Mask-RCNN [72] which is more modern and accurate than regressing for the location heatmap. In more detail, a backbone architecture with 3D convolutions extracts spatio-temporal features. In parallel, Mask-RCNN generates masks of predicted objects, and afterwards the spatio-temporal representation is pooled over the object proposals with RoI pooling to provide appearance features for each detected object. The final object representation passed to the relational module is composed of the appearance features, shape features, and class distribution, where the shape features are estimated by projecting the binary object mask. To leverage the causality of time and model causal object relations (*i.e*. that past actions affect current actions but not the opposite), for a given frame a past frame is sampled and the relational module operates on objects detected between pairs of frames, namely a random past frame and the current frame. The architecture of the relational module is simple; an MLP models inter-frame interactions between pairs of objects, followed by a summation over all pairs. To model long-range interactions, a recurrent network aggregates the interaction features from all frames. The output from the recurrent network is finally used to classify activities. ORN improved the performance of baselines in both object interaction classification and egocentric action recognition, and importantly the authors demonstrated through visualisation that the learnt object interactions are correlated to the predicted activities.

The merits of modelling hand-object interactions with object detectors are also supported by evidence from the EPIC-KITCHENS action recognition challenge, where for two consecutive years (2019 and 2020) the winning entries (from the same team) utilised the active object bounding-box annotations to enable attention on the interacted objects [200, 201]. [200] incorporated Faster-RCNN and ROIAlign [72] to extract top-K features from the top-K bounding boxes with the highest scores. These were max-pooled and used to gate the spatio-temporal CovnNet features. [201] adopted a more complicated architecture, which consists of two branches, one for recognising objects and the other for actions. The top-K RoI features were utilised to enhance both branches with location-aware object information via global and local alignment respectively, providing "object-centric" features. Then, a cross-stream gating

mechanism enables interaction across the object and action branches. Finally, an attention mechanism predicts attention weights between the ConvNet outputs and the object-centric features which is used to aggregate the object-centric features based on their relevance.

2.4.2 Modelling attention with gaze supervision

In egocentric vision, gaze conveys significant information about the action being carried out as eyes need to be coordinated with the hands for grasping and manipulating objects, and thus fixation points can serve as a proxy for attention to discriminant spatio-temporal locations in videos. Accordingly, various works capitalised on gaze supervision to model attention by jointly estimating gaze and predicting egocentric actions [80, 108, 118, 128].

Li *et al*. [108] formulated gaze as a latent variable and modelled its distribution with stochastic units in a ConvNet. Modelling gaze as a probabilistic variable helps to account for its uncertainty resulting from errors in the gaze measurements. The goal of [108] is to optimise the conditional probability of an action class given an input video, where gaze takes the form of a latent distribution of attention, resulting in a term that represents the posterior of gaze which is intractable to learn. An approximation of the posterior is proposed parameterised with a 2D feature map from the convolutional network. The proposed network is optimised with variational learning, where the loss function consists of two terms: i) the cross-entropy between the class prediction and the ground-truth, and ii) the Kulback-Leibler divergence between the ground-truth and the predicted gaze distributions. As the gaze distribution is discrete, it is reparameterised with the Gumbel-Softmax distribution [85] to make the model fully differentiable. Attention maps, sampled from the learnt attention distribution, are used to aggregate visual features spatio-temporally.

Min *et al*. [128] adopted the framework of [108] and introduced three main modifications to improve its performance. First, while [108] simply uses the outputs of a ConvNet as inputs to the gaze modelling network, [128] added 3D convolutional layers within the gaze network to account for temporal modelling in the gaze distribution. Second, [108] directly utilised the samples of the gaze distribution as attention maps for spatio-temporal pooling, whereas [128] added a fully-connected layer followed by a sigmoid on top to enhance the attention mechanism by suppressing uninformative attention locations. Finally, [128] replaces the Gumble-Softmax estimator with a direct optimisation method, originally introduced for training variational auto-encoders with discrete random variables, that introduces an unbiased gradient estimator.

Lu *et al*. [118] introduced a simpler deterministic approach where I3D networks [24] were augmented with a spatio-temporal attention module (STAM). STAM consists of a 3D Incep-

tion module [24] followed by an additional 3D convolutional layer and operates on the visual feature maps of I3D and outputs an attention map supervised with gaze. Ground-truth attention maps are generated by placing a 2D Gaussian around the gaze points which also serves as a simple way of handling the uncertainty of gaze measurements. STAM is trained by minimising the MSE loss between its output and the ground-truth attention, and the overall architecture is optimised jointly for attention prediction and action supervision. Attention is realised as element-wise multiplication between the maps produced by STAM and the feature map of the convolutional layer that it operates, and is placed only after the last convolutional layer.

Huang *et al*. [80] proposed the Mutual Context Network (MCN) for jointly predicting egocentric actions and estimating gaze with a gaze-guided action recognition module and an action-dependent gaze prediction module. While their motivation for the gaze-guided action recognition module is similar to other works [108, 118, 128], *i.e*. to visually guide the learnt representations by focusing on the attended regions, they offer a novel perspective for the action-dependent gaze prediction module. They capitalise on the connection between gaze and the performed action, where they observe that different actions are associated with different gaze patterns, particularly because the objects involved in actions convey information about the region of attention. Based on this motivation, they propose to condition the gaze prediction module with the predicted action distribution from the action recognition module to estimate gaze based on semantic information. To this end, an 'action kernel generator' consisting of a fully-connected layer and two convolutional layers takes as input the predicted action likelihood and generates a set of convolutional kernels that encode information about the performed action. The generated action kernels are then convolved with the input features (extracted from an I3D backbone) and passed to a 3D decoder that estimates a gaze probability map per frame.

2.4.3 Modelling attention without gaze supervision

The works of [178, 179] showcased that spatial attention mechanisms can locate the manipulated objects in a scene without explicit streams for learning to localise hands and objects and without the need of hand segmentation, object bounding box or gaze supervision. To achieve this, [178] exploited Class Activation Mapping (CAM) [216] to facilitate a ConvNet in focusing on the object of interest. CAM operates on convolutional feature maps and produces class-specific activation maps, by learning a weighted average of the feature map across channels for each class. In [178], the outputs of the last convolutional layer of a ConvNet as well as the predicted object class are provided to CAM for obtaining an activation map relevant to the interacted object which is converted to an attention map by passing it through a softmax activation. The object-specific attention map is then multiplied with the output feature map

of the ConvNet to attend to the object. Finally, a convolutional LSTM [163] operating on the attended feature maps of all frames in the video, is utilised for spatio-temporal aggregation. Ablations highlighted that attention enables more efficient learning of activities and visualisations verified that the model focuses on the relevant objects and interaction points.

The follow-up work of [179] introduces Long Short-Term Attention (LSTA) that integrates attention into the ConvLSTM architecture, as opposed to modelling attention and spatio-temporal aggregation in two distinct steps as its predecessor [178]. LSTA employs class-specific mappings akin to CAM to pool the input features over the channel dimension and generate attention maps, namely attention pooling. To enhance the attention mechanism, a second ConvLSTM operating on the attention maps of each frame models long-range attention. An attention map at the current frame is calculated by summing the class-specific activation map of attention pooling and the current hidden state of the attention LSTM followed by a softmax activation. This is then used to attend the inputs of the video LSTM at each timestep. To further enhance the forget-update mechanism of the cell state of the video LSTM, the authors introduce output pooling that similar to attention pooling is used to identify the most discriminative features of the cell state of the LSTM.

2.4.4 Appearance and motion streams and beyond

There has been a rich stream of works in egocentric action recognition that: i) employ fusion of vanilla appearance and motion streams [35, 36, 80, 108, 118, 128, 129], ii) adapt them to the egocentric paradigm by introducing specialised appearance and / or motion streams capable of capturing egocentric signals [13, 120, 178, 179, 200, 201], or iii) extend the two-stream architecture to multiple streams with additional sensor measurements [172] or hand-crafted egocentric features [170].

Various works integrated the original appearance and motion streams for egocentric action recognition, as capturing spatial and temporal information is imperative for modelling egocentric videos too, where appearance is essential for reasoning about objects while motion for understanding the wearer's actions. Moltisanti *et al*. [129] assessed the robustness of the mid-level fusion two-stream architecture of [46] to variations in the temporal bounds of egocentric actions. Moreover, in both the EPIC-KITCHENS-55 [35] and EPIC-KITCHENS-100 [36] datasets, late fusion of appearance and motion streams has been employed to obtain baseline performance on each respective dataset. Finally, all the works in joint action recognition and gaze estimation [80, 108, 118, 128] adopted the fusion of appearance and motion I3D streams; [80, 108, 128] applied mid-level fusion by summing the feature maps of each stream, whereas [118] performed late fusion.

While plain appearance and motion streams have proven to be advantageous for first-person action recognition, adapting them to the egocentric domain can enable harnessing of useful egocentric priors, a topic of research that concerned the works of [13, 120, 178, 179, 200, 201]. In addition to a dedicated appearance stream for learning object interactions through hand segmentation and object localisation, [120] employed a conventional motion stream as well, and fused the two streams with late fusion. Interestingly, visualisation of the motion stream's feature maps showed that camera motion compensation automatically emerged through traning the network, where it learnt to ignore background motion and focus on the actor's movements. To complement the Object Relation Network with motion information, the authors of [13] incorporated an 'activity head' too, operating on the same backbone as ORN, followed by global spatial pooling and an GRU for long-range temporal modelling. Furthermore, the winners of the 2019 and 2020 EPIC-KITCHENS action recognition challenges [200, 201] enhanced both appearance and motion streams with the proposed gating mechanisms and combined them with late fusion. Likewise, [178] and [179] applied the attention-augmented ConvLSTM architectures to both appearance and motion inputs; while [178] simply merged the two streams with late fusion, [179] took a more principled cross-modal fusion approach, where the output convolutional features of each stream where used as biases to the LSTM gates of the other stream.

Apart from appearance and motion, egocentric-relevant information resides in other modalities too. That was the main motivation of Song *et al*. [172] that lead them to explore multiple sensor data including accelerometer, gyroscope, magnetic field and rotation. Both gyroscope and rotation could provide hints of gaze, while accelerometer could inform a model about the type of action being performed, as different actions require different speed of movement. [172] trained an LSTM for each type of sensor data and combined them with late fusion. Additionally the authors trained appearance and motion streams by incorporating RGB, optical flow as well as stabilised optical flow to account for camera motion. The visual streams are also combined with late fusion. A second fusion stage was applied to fuse the visual and sensor streams. An interesting find is that max fusion, *i.e*. taking the maximum prediction across the modalities was found to outperform averaging the multi-modal predictions.

In the realm of multi-stream architectures, [170] augmented appearance and motion streams with an additional Ego Stream with the aim to capture egocentric cues, particularly related to the coordination between hands, head and eyes. Different from works that task the network to predict hand masks [120] or gaze [80, 108, 118, 128], the authors of [170] proposed to feed the network with stacks of analogous features. To eliminate the need for annotation, the authors resorted to computer vision techniques to generate hand masks, head motion as well as saliency maps as a proxy for gaze. Ego Stream operates on the concatenation of those features

and is composed of a 2D ConvNet and a 3D ConvNet. By design, the 2D ConvNet performs early fusion of the features in the first convolutional layer while the 3D ConvNet fuses them progressively throughout the architecture. The ego networks are combined with late fusion independently from the spatial and temporal streams, and a second stage combines the fused Ego Stream with the fused appearance and motion streams.

2.4.5 Multi-task learning

The motivation of multi-task learning is that training simultaneously for multiple tasks can enhance the performance of individual tasks, particularly when the tasks are correlated. It can also be seen as a form of regularisation that constrains a model to learn representations by exploiting the relationships between the tasks. In egocentric datasets like EPIC-KITCHENS and EGTEA [108], where actions are decomposed into verbs and nouns, there is a natural correlation between verbs and nouns resulting from the ways that an object can be manipulated that restrains the number of possible verb-noun combinations, *e.g.* you can chop a garlic or an onion but not a cupboard whereas a cupboard can be opened or closed. [120] leveraged the relationship of objects and actions and trained the egocentric-specific appearance stream to classify objects while the motion stream to classify actions, individually as a first step. Then, the two streams were combined with late fusion by concatenating their last fully-connected layers and feeding them to a new fully-connected layer. On top, three classification heads were added, one for verbs, one for nouns and the other for actions, *i.e.* verb-noun combinations, and there was one loss function per head. The joint network was fine-tuned with a weighted combination of the three losses. In a similar manner, [201] trained a VerbNet for actions and a NounNet for objects, but differently than [120] the streams were fused with cross-gating and trained simultaneously from the beginning. In EPIC-KITCHENS, all the baseline models were trained in a multi-task setting too, with a verb head and a noun head. [179] evaluated LSTA in EPIC-KITCHENS by also adding a head for verb-noun combinations in addition to the verb and the noun heads, similarly to [120]. The multi-task learning paradigm of verbs and nouns is adopted throughout this thesis too.

Multi-task learning was also employed by the approaches that jointly learn to predict gaze and actions [80, 108, 118, 128] where the results of [108] and [80] demonstrated that when training for gaze in addition to actions, action recognition performance improved. While multi-task learning is a byproduct of the aforementioned approaches, Kapidis *et al.* [89] introduced an explicit multi-task learning method, and provided a more systematic evaluation of the importance of training for multiple tasks in egocentric action recognition. A shared 3D backbone was utilised to simultaneously train for verb, noun, action recognition, gaze estimation and hand localisation, resulting in five task-specific heads. Differently than the works of

[80, 108, 118, 128] on gaze estimation, gaze was only used as a supervisory signal and the estimated gaze was not utilised for attending the visual features. Consistent improvements were observed by training for multiple tasks over a single-task baseline.

2.4.6 Domain adaptation

Leveraging knowledge from a learnt source domain to train a target domain is a desirable property of machine learning methods, which is particularly advantageous when there is ample data in the source that can be utilised to enhance the performance of the target that lacks sufficient data. Yet, the distributional shift between domains hinders the adoption of simple approaches, such as pre-training a model in the source and fine-tuning it in the target. This problem concerns egocentric action recognition too, where there are various types of domain gaps. Two central types of domain gap are: 1) employing third-person data to improve models aimed to recognise first-person actions, where during pre-training the third-person video model ignores key egocentric properties such as the camera viewpoint and hand-object interactions, and 2) in egocentric datasets like EPIC-KITCHENS, where data is recorded in different environments (kitchens), adapting a model trained in a source environment to a target one is hard, due to variability in appliances, cookware, room layout, as well as the motions of the wearer.

The goal of Li *et al*. [109] is to tackle the first type of domain gap by augmenting the traditional pre-training of third-person video models that uses an action classification objective with additional tasks able to discover egocentric signals in third-person videos and distil them into the video backbone. The motivation for this is to leverage large-scale labelled third-person video datasets while at the same facilitating the model into learning egocentric-specific video features that can narrow the domain gap when fine-tuning for downstream egocentric tasks. To this end, the authors utilised pre-trained models from three egocentric tasks to assign different types of pseudo-labels in third-person videos. The tasks are: i) determining the 'egocentricity' of a video, by training a model in Charades-Ego [167] that contains paired egocentric and third-person videos, to solve a binary classification task of whether a video is egocentric or not, ii) recognising interactive objects by employing a model pre-trained on ImageNet, and iii) detecting hand-object interaction regions by adopting an off-the-shelf hand-object detector. Each model was ran on a third-person video dataset (Kinetics-400 [90]) producing a set of pseudo-labels for each video instance in the form of soft labels, *i.e*. the class probability distribution for the egocentricity and interactive object tasks, while for the hand-object detection task the generated bounding boxes were converted into hand and object score-maps. After obtaining the pseudo-labels, the video backbone was augmented with a head per task forming a knowledge-distillation loss that trained each head to approximate the responses of the egocentric models. Qualitative results demonstrated that the model successfully identified egocentric signals in

third-person videos as the scores of each head were particularly high when hand-object interactions were present. Moreover, ablations demonstrated that the proposed approach learned representations from third-person datasets that transfer better to egocentric action recognition comparing to pre-training solely for third-person action classification, and that the proposed tasks provided complementary benefits. The work of [167] also proposed a method to transfer representations from third-person to first-person videos, yet focused on learning viewpoint invariant representations by exploiting corresponding third-person and egocentric videos.

To tackle the second type of domain gap, Munro and Damen [132] adopted the Unsupervised Domain Adaptation (UDA) paradigm, and formulated the problem as transferring fine-grained representations from a labelled environment to an unlabelled one using the EPIC-KITCHENS dataset. [132] introduced a novel multi-modal domain adaptation method applied to RGB and optical flow modalities, in contrast to previous approaches that operate on single modalities (RGB). Inspired by multi-modal self-supervised approaches, the authors exploited the natural congruence between RGB and optical flow and employed a self-supervised correspondence task that trains a binary classifier to identify whether RGB and flow samples are from the same video. The self-supervised correspondence task was applied to both source and unlabelled target domains where it was shown that this strategy enables learning common features in both domains encouraging adaptation. To further align the source and target domain distributions, an adversarial alignment method [52] was utilised within each modality, that trains a domain discriminator and reverses its gradients to maximise its loss and remove domain specific information from the feature representations. Finally, an action classifier was trained on source-only videos with late fusion by summing the modalities' predictions. The architecture was trained by combining each individual loss.

2.4.7 Combining audio and vision

In this section, we have witnessed the importance of learning representations that can encode hand-object interactions for recognising fine-grained actions where a broad range of approaches for tackling the problem have been discussed. Yet, none of these approaches explored the role of audio in reasoning about interacted objects. The audio-visual methods of [133, 196, 207] were evaluated on EPIC-KITCHENS and verified that audio is beneficial for egocentric action recognition, although these are general-purpose action recognition approaches and not particularly designed for egocentric vision. The work presented in this thesis fills this gap by introducing audio-visual action recognition approaches with the main objective being to recognise egocentric actions, by leveraging the discriminant sounds produced from object interactions. In particular, Ch. 3 shows that audio is a competitive modality for egocentric action recognition, achieving comparable performance to appearance. This is vastly

different than third-person audio-visual action recognition approaches. For example, the proposed method of [206] was evaluated on UCF101 were it was shown that the ability of audio to recognise third person actions was severely limited (16% accuracy for audio compared to 80% and 78% for appearance and motion), with only minor improvements comparing to utilising only vision. A similar conclusion can be made for the Kinetics dataset from the results of [115, 116, 196, 207].

2.5 Temporal Context for Video Recognition Tasks

Contextual information is valuable. For example, image recognition models frequently identify objects by resorting to their surroundings, such as other object instances or the background. In the same way, video recognition models can benefit from contextual cues. The methods that have been reviewed thus far, may implicitly do so by leveraging context within the clip, *e.g.* to recognise the action 'kick ball' in a video that shows people playing football, the model might learn to exploit the grass in the field in addition to the players' movements. Nevertheless, the temporal dimension of videos offers the opportunity of explicitly capitalising on temporal context in more sophisticated ways. In this thesis, we would like to discriminate the notion of temporal context from long-term temporal modelling which are typically used interchangeably. We refer to long-term temporal modelling for approaches that utilise a large temporal horizon *within* the clip with the goal to model the progression of long events, whereas we use the term temporal context for information *outside* the clip which is particularly useful when untrimmed video is available. For instance, an action recognition method can take advantage of the temporal context of surrounding actions or background frames outside the action boundaries [203], while an object recognition method can benefit from detections in neighbouring frames [14]. Intuitively, a model that can relate the present with the past and the future (when available) should develop better reasoning capabilities. Thereupon, this section reviews relevant methods that make use of temporal context for various video recognition tasks.

Action anticipation, that is the task of predicting the action that will happen next given a video segment, utilises temporal context by definition as the model operates on past context to make a prediction and the context can include both background segments and other actions [49, 159]. Furnari *et al*. [49] introduced Rolling-Unrolling LSTMs (RU), where a Rolling LSTM encodes the observed sequence of frames and an Unrolling LSTM provides predictions at sequential anticipation times. Moreover, RU is multi-modal where appearance, motion and object-centric RU branches are fused by linearly combining their predictions with modality attention weights. Nevetheless, RU operates on short temporal horizons of 3.5s and an analysis on the input length demonstrated that the model's performance drops when using longer horizons.

Sener *et al*. [159] proposed Temporal Aggregation Blocks (TAB) that operate on multi-scale past temporal context for action anticipation and integrate recent observations with long-range past context. The authors considered recent features that last up to a minute in the past and spanning features from the longer past up to ten minutes before the current frame. Recent and spanning feature banks were introduced that incorporate multi-scale features which were assembled by splitting the video into snippets and max-pooling the features within each snippet, where the scale is controlled by the number of snippets. A TAB is a bottom-up architecture where at its core there are Non Local Blocks (NLB) [199] that perform dot-product attention inspired by Transformers. A TAB is composed of multiple Coupling Block (CB) that use NLBs to relate a recent feature with spanning features from all scales and the outputs of all CBs are aggregated within a TAB. [159] made an important observation about the role of temporal context. It was shown that while for the Breakfast dataset [98] where actions are sparsely annotated within the videos the performance was increasing by incorporating more temporal context up to 10 minutes, for EPIC-KITCHENS where actions are much denser more temporal context was beneficial up to a minute of video. The authors' take on this behaviour was that given the size of the EPIC-KITCHENS dataset the model was not able to capture the large variability of long-range dependencies at longer temporal context. Nevertheless, the proposed model can handle significantly longer temporal context than [49]. Finally, the model was tested on action recognition as well by modifying the architecture to consider both past and future context around the action to be classified.

Ng and Fernando [136] tackled the problem of human action sequence classification, where given a video that contains multiple actions, the goal is to train a model to predict the sequence of actions in the order they appeared in the video. The task, as defined by [136], is difficult in that the model is trained with weak supervision using only the ground-truth sequence of actions but not their temporal boundaries, and thus it should implicitly learn the action boundaries. To address this task, [136] proposed an autoregressive encoder-decoder LSTM with attention, where after encoding the sequence of video features with the encoder, at each step the decoder takes as input the predicted action from the previous step and predicts the current action. Therefore, the model learns relationships between the current action and previous actions, effectively attending to the relevant temporal context that consists of other actions in the video.

Zhang *et al*. [213] focused on the problem of fine-grained action classification in untrimmed videos that consist of short events spanning a few frames, where multiple possibly overlapping events can occur in a video. Similarly to [136], they investigated weak supervision for solving the problem, *i.e*. no temporal location of events was provided, but differently than [136] they did not consider the temporal ordering of actions; their goal was to identify which

actions occur in a video by learning to answer to a predefined set of questions. To this end, they introduced the Temporal Query Network (TQN) that operates on densely extracted clip-level visual features and employs a set of learnable permutation-invariant query vectors that are transformed into response vectors by attending over the visual features, where the query vectors correspond to events and their attributes. More specifically, a Transformer decoder performs cross-attention between the query vectors and the visual features and each query vector aggregates information relevant to the corresponding event. The output of the Transformer is a set of response vectors that are linearly classified into attributes with individual classifiers per response. An analysis of the temporal context length in training showed that optimal results were achieved by using the whole extent of the video, indicating that although the query vectors are intended to pick out the segments relevant to a particular event, the temporal context from other events is beneficial.

Wu *et al*. [203] augmented 3D ConvNets that typically operate on short video segments with a Long-Term Feature Bank (LFB) that stores features of the entire untrimmed video. LFB is coupled with a Feature Bank Operator (FBO), namely an attention mechanism that enhances the clip-level representation of the ongoing action by relating it with information about the past and the future using LFB. In more detail, FBO is realised with Non-Local Blocks and uses the short-term representation of the ConvNet as the query and a subset of features from LFB using a window centered at the current clip as keys and values and outputs an encoded representation about the current clip by attending to past and future features from LFB. The classifier then takes as input the concatenation of the short-term features and the enhanced representation from FBO. The proposed model was evaluated for a range of different tasks including spatio-temporal action localisation using the AVA dataset [65] and egocentric action recognition using EPIC-KITCHENS. Depending on the needs of each task, different types of features were stored in LFB. For AVA, a person detector was run for the entire video and the features from a parallel ConvNet were RoI pooled using the detection boxes. For EPIC-KITCHENS, two models were trained, one for verbs and one for nouns. While for the verb model LFB was constructed by simply pooling the features of a 3D network, the noun model utilised an object detector storing RoI pooled object-level features in LFB. Ablations on AVA demonstrated that results improved by increasing the temporal support of LFB (maximum was 60 seconds) and that although the model can be used in a causal setting, *i.e*. leveraging only past context to predict the current action, exploiting both past and future context is more beneficial. For EPIC-KITCHENS, the authors found that best performance was obtained using a window length of 40 seconds for verbs and 12 for nouns, corroborating the findings of [159] to some extent, that densely annotated datasets like EPIC-KITCHENS require shorter context.

Cartas *et al*. [27] leveraged egocentric-specific temporal context by modelling sequences of hand-object interactions using a hierarchical LSTM. As a first stage, they adapted the method of [58] for the egocentric domain, where hands and objects were detected for each frame and actions were predicted at the frame level using a late fusion of the predictions from the pooled hand and object features. The frame-level action predictions were used as input to an LSTM to perform temporal modelling within each action clip and produce temporally consistent frame-level predictions. At the top stage of the hierarchy, a second LSTM was utilised to capture long-term temporal interactions across consecutive actions by exploiting the action boundaries. That is, the hidden unit of the bottom LSTM corresponding to the last frame of each action was fed to the top LSTM, effectively connecting past actions to the current action. The top LSTM performed action-level classification. After training, the top LSTM was discarded to remove the need for utilising the action boundaries in testing. Ablations demonstrated that the temporal context of past actions was beneficial to a certain extent but bigger performance boosts were due to temporal modelling within the action.

This section established that there is already a variety of different techniques for capturing temporal context for video recognition. Nonetheless, all the presented approaches are targeted at modelling temporal context in the visual domain. Ch. 5 argues that leveraging the temporal context of multiple modalities has complementary benefits and fills the gap between multi-modal methods and temporal context approaches by proposing a multi-modal temporal context framework that capitalises on the temporal context in the data stream using vision and audio as well as prior temporal context using a language model that operates on the action labels. To facilitate better understanding of the framework, the next two sections describe relevant concepts.

2.6 Multi-modal Transformer Architectures

The self-attention mechanism of transformers provides a natural bridge to connect multi-modal signals by comparing multi-modal inputs and aggregating representations from different domains using multi-modal attention weights. This property inspired the development of multi-modal transformer architectures for a range of different applications, including vision and language representation learning [104, 117, 182], image/audio/video classification [84, 99, 182], image/video captioning [82, 102, 117, 182], image segmentation [210], cross-modal sequence generation [51, 103, 105] and video retrieval [50]. Rather than delving into the details of each approach and task, this section highlights some of the most common paradigms for multi-modal transformer design.

The most basic distinctions between these architectures lie in the mechanism that integrates

signals from different modalities, where two primary categories have emerged: 1) multi-stream architectures [82, 99, 102, 103, 105, 117] and 2) single-stream architectures [50, 51, 84, 104, 182, 210]. The distinction is inspired by [91] which however covers mostly works on vision and language, whereas here the categorisation is extended to other multi-modal areas described above.

Early fusion has been broadly adopted among single-stream architectures, where a Transformer encoder simply ingests concatenated sequences from different modalities [50, 104, 182] and performs self-attention to the multi-modal inputs. In such case, inputs from each modality act as queries and keys/values at the same time and interaction is allowed between all queries and keys/values. Therefore, the Transformer encoder performs simultaneously within-modal and cross-modal attention by comparing each query with keys from both the same and other modalities. This produces multi-modal attention weights which then fuse the modalities by aggregating the multi-modal values. In other problems, such as generating music from videos [51], a single-stream autoregressive decoder is a reasonable approach as it provides a straightforward mechanism for conditioning the audio generation on the input video. The Transformer decoder enforces cross-modal attention by using the audio as query and the video representation as key/value, therefore preventing interaction within the modalities.

A popular strategy in multi-stream architectures is to restrict the early layers to perform uni-modal modelling and enable multi-modal communication only at the later layers of the architecture, *i.e*. mid-level fusion. In multi-stream Transformers, this is realised with modality-specific Transformer streams that perform self-attention within each modality for the first layers of the architecture followed by concatenating the uni-modal Transformer outputs and feeding them to another Transformer that performs multi-modal self-attention [99, 105]. The method of [133] (discussed in Sec. 2.3) is also in this category, where the concatenated outputs of modality-specific Transformers are fed to attention bottleneck layers for multi-modal processing. Others do not utilise a Transformer to fuse the uni-modal representations and apply merely late fusion of the predictions of each Transformer stream [103]. Finally, cross-modal attention has been employed in a two-stream setting too, where each modality's cross-modal Transformer uses inputs from that modality as queries and inputs from the other modality as keys/values [102, 117].

In addition to being permutation invariant, Transformers are modality invariant too, *i.e*. they cannot discriminate from which modality the inputs are coming from. This should not be an issue for cross-attention as modality information is encoded in the query/key/value projection weights because one modality is always used as they query and the other as key/value. However, it becomes a problem for approaches that concatenate modalities and encode them with self-attention. A common strategy to alleviate this is to add learnable modality encodings to

the input in addition to the positional encodings [50, 84, 99, 104]. Another common paradigm is to extract features for each modality from pre-trained uni-modal convolutional networks and use them as inputs to the Transformer [50, 82, 99]. Similarly, models that operate on vision and language use visual features from pre-trained ConvNets while for language word embeddings are learnt within the Transformer [102, 104, 117, 182, 210].

Lastly, a major breakthrough in multi-modal Transformers was the Perceiver architecture [84]. While it belongs in the single-stream architectures it's unique in its kind. Differently than other multi-modal Transformers that operate on extracted convolutional features and more similarly to [133], the Perceiver is trained end to end from raw multi-modal signals. Yet, while [133] leverages the prior knowledge about the modality from which the inputs are coming from by incorporating modality-specific streams before fusion, the Perceiver makes no assumptions about its inputs. Modality inputs are simply flattened and concatenated and treated as a large byte array. As the quadratic complexity of vanilla Transformers would make quite inefficient the processing of such enormous byte arrays, they key contribution of the [84] is the introduction of a learnable latent array of significantly smaller length than that of the input byte array. The Perceiver projects the high-dimensional byte array into a low-dimensional bottleneck using cross-attention between the byte array and the latent array followed by vanilla self-attention and it alternates between cross-attention and self-attention layers. Following the discussion on Subsec. 2.1.3, as the Perceiver does not exploit any inductive biases about the inputs, it is mostly suitable for big data regimes.

2.7 Language Models and Their Usage for Capturing Action Context

2.7.1 A brief introduction to language modelling

Language models estimate the probability distribution of a sequence of words or the probability of a word given other words in the sequence. That is, a language model assigns a probability to a sequence of words that expresses how likely the sequence is. For example, a language model trained with a corpus of human written text would assign a high probability to a natural sentence and a low probability to a random sequence of words. Thus, language models encode the prior structure of word sequences. Based on this capability they have been used in various tasks such as speech recognition [28, 70, 127] and machine translation [66] to re-score the predictions of the model and filter out improbable sequences. Typically these approaches perform a search (often a beam search) to find the most probable sequences of words according to their model. To filter out improbable sequences, the language model is incorpo-

rated in the search algorithm such that the most probable sequences are computed according to both the model for the task in hand and the language model. A standard approach is a shallow fusion that computes a weighted average of the probability of a sequence using the model for the task in hand and the probability of the same sequence using the language model.

Traditionally, statistical language models have been extensively used where n-gram is a typical statistical language model. An n-gram model approximates the probability of a sequence of words as the product of the conditional probabilities of each word in the sequence given $n-1$ preceding words, *i.e.* the context, based on the n^{th} order Markov property. The conditional probability is simply calculated by counting the frequency of occurrence of the word given the $n-1$ previous words among all sequences of length n. Nevertheless, a major weakness of statistical language models is related to the curse of dimensionality; as the context size increases for a given vocabulary size (*i.e.* the number of unique words), the number of possible sequences increases exponentially which causes a data sparsity problem, where many sequences do not occur and their probability cannot be calculated resulting in a sparse probability distribution.

Neural Language Models (NLM) employ neural networks to model the probability distribution of word sequences [17, 127]. Similarly to an n-gram, they predict a word by taking as input the context of previous words, and model the probability of a sequence as the product of the conditional probabilities of the words in the sequence. A fundamental difference is that words are represented as continuous vectors, namely word embeddings, and modelled as non-linear combinations of weights in neural networks. An important property of word embeddings is that semantically similar words are close in the learnt embedding space. Moreover, NLMs learn jointly the word embedding and the probability distribution of words. In their simplest form, NLMs adopt a feed-forward fully-connected neural network design [17]. At the input level, the word embedding layer takes as input the context of preceding words as one-hot encodings over the vocabulary size and projects each one independently. A hidden layer projects the concatenation of the word embeddings of the context to a latent space. Subsequently a linear layer with a softmax activation is used to predict the current word. The network is trained to maximise the sum of the log-likelihood for each word in the sequence.

NLMs has certain advantages over n-grams. First, they scale better with the increase in the vocabulary and the context size that cause only a linear increase in the parameters of the word embedding layer and the hidden layer, respectively. Conversely, the representation of n-grams grows exponentially with the context size for a given vocabulary size and becomes sparser. Second, the learnt similarities between word embeddings foster generalisation to sequences of words that have not been observed in training in contrast to n-grams that can reason only about observed sequences. Furthermore, the strengths of NLMs over statistical language models

have been validated experimentally [17, 44, 92]. It is also common to employ a recurrent neural network instead of a feed-forward network as recurrent nets can handle both longer-term and variable length context [127].

Recently, high-capacity transformer-based language models have shown outstanding performance in various downstream tasks by pre-training them with language modelling objectives using massive-scale datasets [21, 40, 149, 150, 209]. GPT [149] employed an autoregressive Transformer decoder and pre-trained it using the traditional training objective of NLMs, *i.e*. the cross-entropy of the conditional probability of a word given the context of previous words, summed for each word in the sequence. BERT [40] proposed to leverage both past and future context using a Transformer encoder that is bidirectional by default. Differently than unidirectional language models that read text sequentially left-to-right or right-to-left, BERT observes the whole sequence at once. Therefore, the standard language modelling objective of predicting the next word would not be suitable for training BERT as the model could trivially predict the target word because it would have observed it. To address this, [40] proposed a Masked Language Modelling training objective for bidirectional models where a percentage of the inputs are randomly masked and predicted.

2.7.2 Language modelling for sequences of actions

The ability of language models to capture the context of word sequences makes them appropriate candidates for capturing the contextual structure of action sequences in untrimmed videos, which can be achieved by treating the action labels as words and using them as inputs to a language model. Action context can inform models about prior relationships in sequences of actions, *i.e*. which actions are more likely to co-occur, and steer the learning and inference process by constraining the space of possible solutions. This is a promising but underexplored research direction and current applications include the use of language models for action detection [154] and action sequence classification [111].

Richard *et al*. [154] considered the duration of actions using a length model and the temporal context of actions using a language model and combined them with an action recognition model for detecting and classifying actions. The problem was modelled under a probabilistic framework where the goal was to segment a video into a sequence of actions by maximising the conditional probability of the sequence of action boundaries and action labels given a sequence of features for each frame in the video. Using the Bayes theorem this was decomposed into a product of three terms that include: i) the action recognition model, ii) the length model that specifies the length of each segment given the predicted actions, *i.e*. it learns the prior duration of each action class, and iii) the language model that computes the prior probability of a sequence of actions and penalises improbable sequences. For the language model, [154]

utilised an n-gram where the conditional probability of an action given the context of previous actions is estimated by counting the frequency of action sequences in the training set. The length model was employed to compensate for the preference of the language model towards longer and fewer segments as large sequences of actions are less likely, where different choices of length models were considered including a class-dependent Poisson distribution. The action recognition model was a linear softmax classifier operating on Fisher vector features, and the framework was optimised with dynamic programming. It was demonstrated experimentally that the language model provided consistent performance boosts.

Lin *et al*. [111] investigated the problem of recognising a sequence of actions from an untrimmed video in a weakly supervised setting, *i.e*. without utilising action boundaries annotations but only the temporal ordering of actions, a problem that was also considered by [136] which however did not use a language model. To address this problem, [111] combined a sequential action recognition model with an n-gram for modelling the prior context of actions and a prior about the length of actions. Evidently, this formulation draws similarities with [154]. Nevertheless, while [154] both detects and recognises actions, the aim of [111] was to predict only the sequence of actions but not their temporal boundaries. The posterior probability of the action sequence given the video was decomposed into two terms, the action recognition model and the prior probability of the action sequence. [111] modelled the prior of the action sequence as a product of the probability of the language model for the sequence of actions and the prior length of the sequence. The action recognition model was a bidirectional LSTM that takes as input Fisher vector features for each frame in the video. The proposed model was trained with the Connectionist Temporal Classification (CTC) [64] to learn the mapping between the inputs and the target output sequence. Similarly to [154], it was shown that both the language model and the length model improved the efficacy of the proposed method.

While both of these works rely on a statistical language model, Ch. 5 adopts a transformer-based language model trained with the MLM objective to take advantage of both past and future action context and empirically demonstrates its benefits over an n-gram model.

2.8 Third-Person Action Recognition Datasets

There are third-person action recognition datasets that are relevant to the work presented in this thesis as they capture hand-object interactions, like Breakfast [98], MPII Cooking [155], 50 salads [175] and Something-Something [62] but these do not contain audio. Moreover, YouCook2 [218] and HowTo100M [126] exhibit object manipulations but the audio is mostly speech of narrated instructions. This section focuses on two popular third-person action recognition benchmarks which contain audio that mostly originates from the action and discusses

possible reasons that the benefits of audio-visual recognition approaches on these have been limited.

2.8.1 UCF101

UCF101 [173] is a medium-scale third-person action recognition dataset with over 13K action clips downloaded from YouTube that amount to 27 hours of video and 101 actions. The dataset is trimmed in that each clip contain a single action as opposed to untrimmed videos with multiple actions occurring at different parts of the video. The approach of downloading content from an online platform resulted in videos with camera motion, background clutter and diverse lighting conditions amongst other challenges, as the videos have been captured in the wild in naturalistic environments. UCF101 extended UCF50 [152] that had 50 classes with 51 new classes but only the videos corresponding to the new classes have audio. The action classes can be divided in five groups: i) human-object interactions, *e.g.* 'apply lipstick', ii) body-motion only, *e.g.* 'push ups', iii) human-human interactions, *e.g.* 'head massage', iv) playing musical instruments, *e.g.* 'playing violing' and v) sports, *e.g.* skiing. The dataset was annotated manually by downloading a pool of candidate clips from YouTube and the irrelevant ones were removed after visual inspection. The mean clip length is 7.21s while the minimum is 1.06s and the max 71.04s.

2.8.2 Kinetics

The Kinetics dataset [90] (also known as Kinetics-400) contains 400 human action classes with at least 400 clips per action resulting in ~300K trimmed videos clips with audio. Each clip lasts 10s and the clips were collected from unique YouTube videos rather than extracting many clips from the same video. As the videos are amassed from YouTube they contain many challenging conditions as UCF101, but this is more amplified in Kinetics due to its scale. Moreover, by collecting each clip from a different YouTube video the data contain a large number of performers and variance in camera viewpoint. Kinetics has a mixture of coarse-grained and fine-grained classes and the types of actions can be divided in three categories: i) singular human actions, *i.e.* actions that do not involve interactions such as singing and drinking, ii) human-human actions where humans interact with each other like hugging and kissing and iii) human-object actions where humans interact with objects, for example 'cleaning shoes' and 'washing dishes'. An important contribution of [90] is the partial automation of the dataset curation process by resorting to image classifiers for selecting candidate clips and to action classifiers for filtering out ambiguous classes and videos with noisy labels.

Kinetics-600 [25] is an extension of Kinetics-400 and was collected by refining the annotation

process of Kinetics-400. The dataset contains 600 classes with at least 600 videos each and ~500K video clips. Further fine-tuning of the data collection pipeline resulted in Kinetics-700 [26] with ~650K clips and 700 classes with at least 600 clips each, and subsequently in Kinetics-700-2020 [171] with 700 classes and at least 700 clips per class.

Concluding remarks

As discussed in Subsec. 2.4.7, audio classification performance on Kinetics and UCF101 has been low, inhibiting advances using audio-visual fusion. This might be due to a few different reasons. First, these datasets were not curated with an audio-visual correspondence objective, that is it was not guaranteed that the visual action can also be heard, which can lead to videos with irrelevant audio, such as speech from a narrator or background music. This is more pronounced in Kinetics because of its large scale. Second, many classes do not have distinct sound signatures, *e.g.* 'tying tie' in Kinetics and 'apply eye makeup' in UCF101. Lastly, in third-person videos the microphone is often far from the action and important sounds might be missed.

2.9 Egocentric Action Recognition Datasets

Differently than third-person datasets that often contain irrelevant sounds and classes with indiscriminative audio, most egocentric actions are inherently audio-visual due to object interactions and sound-emitting objects. Nevertheless, the collection of egocentric datasets sometimes results in audio-visual artifacts (which will be discussed in this section) limiting their use for audio-visual egocentric action recognition. Another factor impeding the adoption of modern deep learning approaches for audio-visual fusion is the small scale of some of the existing egocentric datasets that contain audio like ADL [147], CMU [38] and BEOID [34]. In addition, an issue of ADL in particular is that the wearable cameras were chest-mounted rather than head-mounted occluding the sound of many actions from the friction of the microphone with the wearers' clothings. This section details medium to large scale audio-visual egocentric datasets (with the exception of [108] that is visual-only) and also discusses possible issues with their audio recordings.

2.9.1 EGTEA Gaze+

EGTEA Gaze+ [108] is a visual-only dataset for egocentric action recognition and gaze estimation. It consists of 32 subjects preparing 7 different meals, such as American breakfast, Greek salad and pizza, and it was collected in a single kitchen environment. The dataset contains 86 untrimmed egocentric videos with a total duration of 29 hours recorded using SMI

eye-tracking glasses, and each untrimmed video contains a single meal preparation activity that is composed of multiple actions. Gaze location was also recorded at every frame. Actions are combinations of verbs and nouns and there are 19 verb classes, 53 noun classes and 106 action classes. The start and end times of actions were annotated for each untrimmed video, resulting in a total of 10K annotated action segments with an average duration of 3.2s. The action classes follow a long-tailed distribution. EGTEA provides hand mask annotations too. While audio was not released with the dataset, EGTEA accommodates sequences of short actions within untrimmed videos making it a suitable candidate for evaluating part of the method proposed in Ch. 5 for capturing visual temporal context from sequences of actions.

2.9.2 Charades-Ego

Charades-Ego [167] features paired first-person and third-person videos to facilitate research in transfer learning from third- to first-person vision. The videos were collected following the approach of [166], where Amazon Mechanical Turk (AMT) workers were asked to record videos of themselves performing certain activities. For Charades-Ego, the authors employed crowd-sourced scripts from the Charades dataset [166], and asked workers to record two videos enacting the same script: one in a third person setting, and the second in first-person using a head-mounted camera. The dataset consists of 112 subjects performing actions in different rooms at their homes. All the 157 actions from the Charades dataset are included which are verb-noun pairs and some examples include 'holding a laptop', 'making a sandwich' and 'closing a window'. There are 4000 pairs of third/first-person untrimmed videos with audio with an average length of 31.2s and 43.5K trimmed action clips in total. While Charades-Ego is seemingly a large scale dataset (in terms of number of action segments), this is not exactly true in practice. That is because the start/end times annotations of actions are noisy, where many annotated segments do not actually contain the labelled action, making it impractical for training and evaluating egocentric action recognition models. Moreover, the egocentric video recording procedure is questionable as there are activities recorded by holding the camera in one hand while performing the activity with the other restricting the freedom of the performer, which resulted in cases where the recording does not contain the performed activity.

2.9.3 Home Action Genome

Home Action Genome (HOMAGE) [151] is a multi-modal multi-view dataset for action recognition with hierarchical activity labels and spatio-temporal scene graph annotations. Concretely, the dataset offers videos from both first-person and third-person viewpoints and the annotations capture video-level activity labels as well as segment-level action labels. Similarly with the other egocentric datasets described thus far, HOMAGE is scripted as participants were

given specific instructions for performing tasks. The dataset contains recordings of 27 participants performing activities in different rooms of two different houses. Multi-modal data were composed using 12 different sensors amongst cameras, microphones, infrared, acceleration, gyro, humidity and temperature. For the third-person views the sensors were placed in various locations in the room whereas for the egocentric view it was mounted to the participants' heads and the cameras for all views were synchronised.

About the annotation process, each video was assigned with a single activity label and multiple often overlapping atomic action labels using verb-object pairs. Moreover, the scene graph labels are bounding boxes of the person performing the action and the interacted objects from the third-person views. HOMAGE comprises of 75 activities labels, 453 actions labels and 1752 untrimmed multi-modal sequences. Labelling of start/end times of action resulted in 24.5K action segments of short duration (60% is under 2s). It should be noted that the egocentric views of one participant are not provided and therefore the actual scale of the dataset in terms of egocentric videos is smaller. Furthermore, the annotation of actions is redundant in that there are many overlapping actions that carry similar semantics (*e.g.* 'take cup from somewhere'-'hold cup'-'close cabinet'-'hold cabinet' happening simultaneously). Finally, while providing multiple sensor measurements can facilitate research in sensor fusion for action recognition, for purely audio-visual purposes the dataset is not ideal because audio is not always in sync with the video possibly because the two modalities were not captured from the same device.

2.9.4 EPIC-KITCHENS

In the previous decade, a major bottleneck in egocentric action recognition research was the lack of sufficiently large and diverse datasets. Existing egocentric datasets [34, 38, 108, 147, 167] were significantly smaller than their third-person analogues. One factor limiting the volume of ego-datasets is the scarce availability of footage capturing interactions from wearable cameras in YouTube, making their collection burdensome. Apropos the restricted data diversity, a central cause is that all egocentric datasets were scripted, that is participants were asked to act out a script or to perform a predetermined set of actions. This approach does not conform with real-life scenarios where people perform multiple tasks in parallel and each individual completes a task in their own way and order; instead, it reduces the heterogeneity of videos as the subjects perform the steps of an activity in a particular order. Another issue was that most first-person datasets were captured in a single environment [34, 38, 108]. This was addressed by ADL [147] and Charades-Ego [167] where the recordings took place in the participants native environments. Nevertheless, these datasets are scripted as well.

EPIC-KITCHENS [35, 36, 37] addressed the shortcomings of previous first-person datasets by allowing participants to perform non-scripted daily activities in their native kitchen environ-

ments. As participants did not follow instructions, the data are more natural and reflect real-life challenges such as multi-tasking and interleaving tasks, *i.e*. pausing a task to perform another and then resuming the former. Until recently, EPIC-KITCHENS was the largest dataset in egocentric vision. The first version of the dataset, namely EPIC-KITCHENS-55 [35, 37], contains 55 hours of untrimmed video corresponding to 432 sequences and a total of ~11.5M frames that were temporally annotated resulting in ~39.6K action segments. The videos were recorded by 32 participants (thus 32 different environments) from 10 different nationalities across 4 different cities. Participants were asked to be alone in their kitchen while recording, and thus the dataset contains single-person activities and focuses on interactions with objects. The data was captured with a head-mounted GoPro at 60fps and a resolution of 1920×1440. The audio was recorded from the GoPro's built-in microphone, with 2 channels, a sampling rate of 48000kHz and a bit rate of 128kb/s. The proximity of the microphone to the ongoing action offered audio recordings of high amplitude with a small signal-to-noise ratio, making the dataset ideal for audio-visual research. Ch. 3 returns to the importance of audio in EPIC-KITCHENS and discusses sources of ambient noise in the dataset and how these affect the models' performance. The annotation pipeline is described next.

To obtain initial coarse annotations, participants were asked to watch their videos and narrate their actions live (*i.e*. without pausing the video) using a recording device. Participants were instructed to use verb-object pairs for narrating an action (*e.g*. 'peel carrot') and they were allowed to narrate in their native language resulting in narrations in five different languages: English, Italian, Spanish, Greek and Chinese. The narrations were transcribed using Amazon Mechanical Turk (AMT) and the narration timestamps were also extracted from the audio recordings and used as rough locations of the actions. The transcribed narrations along with narration timestamps were utilised for crowd-sourcing accurate action segmentations using AMT too. In particular, AMT workers annotated the start and end time of actions by watching the videos subtitled with the narrations. For robustness, each action was annotated by four different workers and the annotation of the worker with the maximum agreement with other workers was used as the ground-truth temporal bound of the action. As the participants narrated their actions using free text, a large variety of verbs and nouns have been collected. These were manually clustered to reduce the number of classes by merging synonyms. This resulted in a final set of 125 verb classes and 331 noun classes, though these are heavily imbalanced. Finally, object bounding boxes were annotated too. The test set of EPIC-KITCHENS is divided in two splits: Seen Kitchens (S1) where sequences from the same environment are in both training and testing, and Unseen Kitchens (S2) where the complete sequences for 4 participants are held out for testing.

One important aspect of EPIC-KITCHENS is that actions are densely and accurately annotated

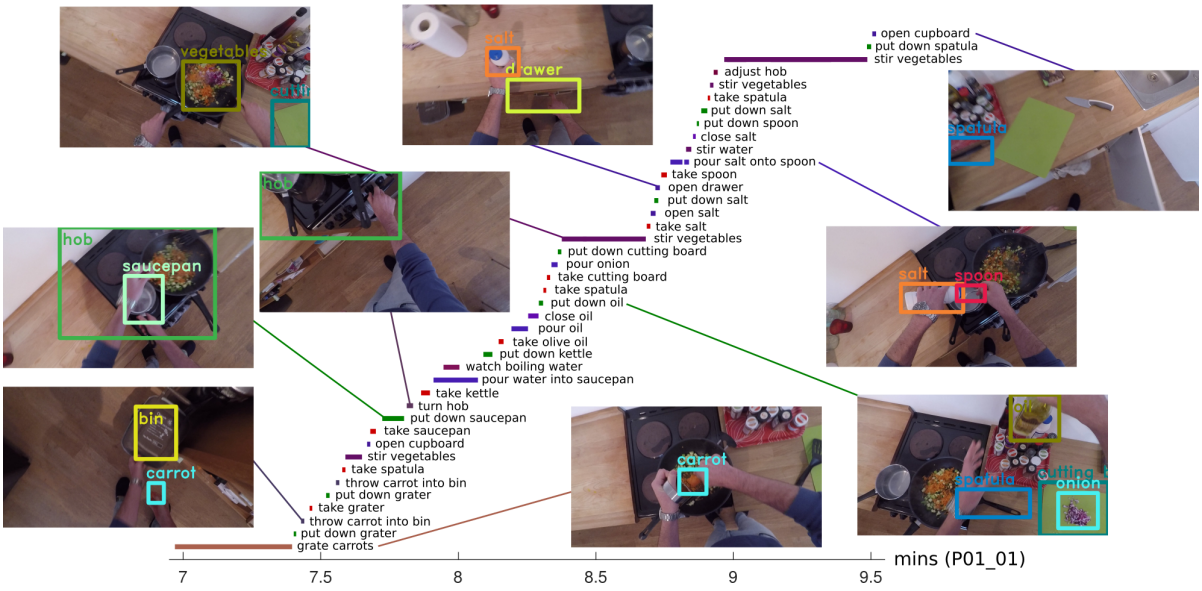


Figure 2.13: *Consecutive action segments in EPIC-KITCHENS [35] with object bounding box annotations. Actions are densely annotated resulting in well-defined sequences of actions that are correlated within large temporal neighbourhoods.*

(the error of the start/end time annotations was 5.7%) resulting in well-defined sequences of actions within untrimmed videos, as depicted in Fig. 2.13. Evidently, actions are correlated within a temporal neighbourhood (*e.g.* open cupboard → take saucepan → put saucepan) and the density of annotations enables correlations within large neighbourhoods. This is in stark contrast with other egocentric datasets [108, 151, 167]. Charades-Ego [167] contains temporal annotations with high amount of noise and HOMAGE [151] has many redundantly overlapping actions both resulting in ill-defined sequences of actions. In EGTEA [108] on the other hand, although the sequences of actions are semantically meaningful the annotations are sparse, limiting correlations between actions within smaller neighbourhoods comparing to EPIC-KITCHENS. Evaluations of the temporal context approach in Ch. 5 using both datasets verify that while the action recognition performance in EPIC-KITCHENS increases with larger context windows, EGTEA enjoys benefits from shorter context.

EPIC-KITCHENS-100 [36] is an extension of EPIC-KITCHENS-55 [35, 37]. The extension contains additional footage from 16 out of 32 participants from [35, 37] as well as data from 5 new subjects. 8 out of the 16 preexisting participants had changed homes, and thus the number of subjects has increased to 37 while the total number of kitchens to 45 in the extended dataset. The extension data were captured using a GoPro Hero7 black at 50fps. EPIC-KITCHENS-100 features an improved annotation pipeline; a fundamental difference is the 'pause-and-talk' narration approach (as opposed to non-stop narration in [35]), where participants narrate the actions they perform in videos, using an interface that allows them to

pause the video and speak. This resulted in denser annotations, *i.e.* more actions per video with smaller gaps between actions. Furthermore, there is a higher ratio of labelled frames, as well as shorter actions (avg length 2.6s and min length 0.25s versus avg length 3.7s and min length 0.5s in EPIC-KITCHENS-55) and higher amount of overlapping actions, comparing to the previous version. The narration timestamps are closer to their related actions which resulted in more accurate start/end time annotations in the temporal annotation step. It also worth noting that while in EPIC-KITCHENS-55 workers annotated the temporal segments of actions using only vision, in EPIC-KITCHENS-100 the audio was included too to improve the annotation of actions that are out of the camera's field-of-view but also to accommodate more precise temporal localisation of the action by leveraging the synchrony of audio and vision. All transcribed narrations, including the ones from EPIC-KITCHENS-55, were re-parsed into verbs and nouns using a more complete parser and re-clustered to reduce ambiguities in verb and noun clusters. The dataset was also enriched with automatic hand and object segmentations using Mask R-CNN [72] and hand-object interaction detections from [161]. Last but not least, to facilitate progress in various areas of video understanding, [36] introduced six tasks with baselines and evaluation metrics: action recognition, weakly-supervised action recognition, action detection, action anticipation, cross-modal retrieval, and unsupervised domain adaptation for action recognition.

EPIC-KITCHENS-100 contains 100 hours of data, split across 700 untrimmed videos, and 90K trimmed action clips. There are 97 verb classes, 300 noun classes, and 4025 action classes. The dataset is split in Train/Val/Test splits with a ratio of $\sim$ 75/10/15, where each video with all its action segments is in one of the splits. The validation set contains the videos from the test sets of EPIC-KITCHENS-55 and the test set contains only newly collected videos. Validation and test set contain two subsets, each: i) unseen participants, *i.e.* participants not present in Train, where there are 2 in Val and 3 in Test and ii) tail classes, defined as the set of smallest classes which comprise 20% of the total number of examples in the training set.

2.9.5 Ego4D

Ego4D [63] is a recently proposed multi-modal egocentric vision dataset of enormous-scale with 3000 hours of video, the largest and most diverse of its kind to date. It was recorded by 855 different people with wearable cameras from 74 cities across 9 different countries. The geographic and demographic diversity of the participants resulted in a large variety of objects and activities that lack from any other egocentric dataset. The dataset captures both indoor and outdoor unscripted daily life activities of the camera wearers in various places and circumstances such as in the home, workplace as well as social interactions, covering a wide range of scenarios, like cooking, arts & crafts, construction and cleaning. To enable

cross-device generalisation, different types of wearable cameras were used for collecting the dataset, offering different camera viewpoints depending on the mounting and the field of view of each device. Moreover, the multi-type camera setting enabled multi-modal data collection, where in addition to video, subsets of the dataset contain audio, stereo, 3D meshes, gaze, synchronised multi-view cameras as well as other modalities. Audio in particular amounts for 73% of the dataset and contains conversations, acoustic scenes and events, and sounds from interacted objects. Inspired by the pause-and-talk narration approach of EPIC-KITCHENS-100, Ego4D was narrated by annotators that were pausing the video to write a sentence about every action performed by the wearers. The narrations contain 1772 verbs and 4336 nouns which were clustered into 117 verb and 556 noun classes. Along with the data, [63] offers five benchmark challenges to enable research in core egocentric areas and annotations for each benchmark. The tasks that incorporate vision and audition are: i) Audio-Visual Diarisation, that is to localise and recognise the active speakers and transcribe their speech (*i.e.* ASR), and ii) Social Interactions, where given a video and audio that portray people interacting, a model should predict whether each person is looking and talking to the camera wearer. Unfortunately, in this version of the dataset the authors did not consider an audio-visual event detection and recognition task, but I do hope they will do in the future to facilitate progress in audio-visual object interaction modelling at scale.

2.10 Large-Scale Datasets for Auditory Model Pre-training

While it has been shown that leveraging ImageNet pre-trained models is beneficial for audio recognition [59, 60, 67, 138], it permits utilising only image recognition architectures like ResNets. Yet, for developing a specialised architecture for audio, where pre-trained models for images do not exist, one should resort to germane auditory databases for pre-training the model. This section examines datasets for auditory scene and event recognition, as they are the most pertinent for pre-training auditory action recognition models. A pre-training dataset should enable learning discriminant features and generalisable to downstream tasks. This can be achieved using large-scale datasets with accurate annotations. Thus, small-scale datasets for audio scene and event recognition like DCASE 2018 [124], UrbanSound8k [157] or ESC-50 [146] are not appropriate as they would overfit deep architectures. AudioSet [55] is the largest scale audio-visual dataset of audio events with 2.1M clips of 10s and 527 annotated classes. While it has been a milestone for audio recognition, it has high labelling noise and highly unbalanced data, hence not ideal for supervised pre-training. A dataset that meets better the criteria for supervised pre-training of audio architectures is presented next.

2.10.1 VGG-Sound

VGG-Sound [30] is an audio-visual dataset collected from YouTube with over 200K video clips that last 10s. There are 300 audio classes with at least 300 clips each. The classes are grouped into the following categories: people, animals, music, sports, nature, vehicle, home, tools, and others. VGG-Sound was curated with a semi-automated pipeline based on computer vision techniques that ensures a low number of labelling errors as well as audio-visual correspondence, *i.e*. that the sound-emitting object is present in the visual clip. The data collection process consisted of multiple stages. In a similar spirit to Kinetics, it started by downloading a large number of candidate videos from YouTube using a list of potential classes as queries followed by progressively filtering out videos and classes to arrive at a subset with accurate annotations. The initial list of classes was created by considering classes of sounds that occur naturally in real world scenarios and that can be visually grounded. The filtering stages consist of visual verification, audio verification, and iterative noise filtering. The visual verification stage employed image classifiers to detect frames in the videos that were most likely to contain the visual signature of the sound class (*e.g*. that a video with the class 'cat meowing' contains a cat). These frames were then used as the centre of the 10s candidate clips and videos that did not pass the test were discarded. The goal of the audio verification step was to ensure that videos do not contain speech or background music but sound emitted from the visual object. This was achieved using an audio classifier to reject clips, which was trained with three classes: music, speech and others. Finally, iterative noise filtering trained audio classifiers for each of the remaining classes to select clips that their class can be easily guessed from audio, and visual features to retrieve clips that is hard to distinguish their category from audio because they contain multiple sound sources.

2.11 Summary

This chapter reviewed the literature that is relevant to the contributions of this thesis. The chapter started by scrutinising third-person visual action recognition approaches – the basis of both egocentric and audio-visual action recognition. Special emphasis was given on state-of-the-art convolutional and transformer-based approaches for spatio-temporal modelling as well as two-stream visual action recognition methods as this thesis builds on these works.

Next, the chapter introduced the preliminaries of audio processing for audio recognition and reviewed methods and techniques for auditory scene and event recognition that this thesis is based on. Audio-visual action recognition approaches were then reviewed in detail as all the chapters in this thesis build upon relevant audio-visual architectures. After having discussed the advances in third-person action recognition, the chapter provided an overview of the most

central works in egocentric action recognition several of which are used for comparison with the method presented in Ch. 5.

As Ch. 5 proposes a method for capturing multi-modal temporal context from vision, audio and language using a multi-modal Transformer architecture, this chapter presented methods for modelling visual temporal context for video recognition, followed by an overview of the most common paradigms in multi-modal Transformers as well as a discussion on language models for capturing action context.

Finally, the chapter reviewed audio-visual datasets for third-person and egocentric action recognition as well as for scene and event recognition, some of which are used for the experimental analysis of the proposed methods.

CHAPTER

THREE

AUDIO-VISUAL TEMPORAL BINDING

This chapter explores the potential *temporal asynchrony* between the action's appearance and the discriminative audio signal. In Fig. 3.1, we show an example of 'breaking an egg into a pan' from the EPIC-KITCHENS dataset. The distinct sound of cracking the egg, the motion of separating the egg and the change in appearance of the egg occur at different frames/temporal positions within the video. Approaches that fuse modalities with synchronised inputs (*e.g.* [46]) would thus be limited in their ability to learn such actions. Another challenge in integrating visual frames with audio segments, resulting from the large discrepancy in visual and auditory sampling rates, is the correspondence problem. That is, the ambiguity about which visual frame should be combined with an audio segment amongst the sequence of frames within the audio segment's temporal range. It is unclear whether it should be the middle, first, last or any frame in between. This chapter explores fusing inputs within a Temporal Binding Window (TBW) (Fig. 3.1), allowing the model to train using asynchronous inputs from the various modalities. Evidence in neuroscience and behavioural sciences points at the presence of such a TBW in humans [139, 191]. This is typically assessed by presenting to subjects audio-visual stimuli at varying temporal delays across modalities and asking them to determine whether the stimuli are synchronised. Even when the audio-visual signals are delayed by hundreds of milliseconds the subject still perceives them as synchronous at high rates, suggesting that there is a window of time, the so called TBW, that offers a "range of temporal offsets within which an individual is able to perceptually bind inputs across sensory modalities" [176, 191]. This is triggered by the gap in the biophysical time to process different senses [123]. Interestingly, the width of the TBW in humans is heavily task-dependent, shorter for simple stimuli such as flashes and beeps and intermediate for complex stimuli such as a hammer hitting a

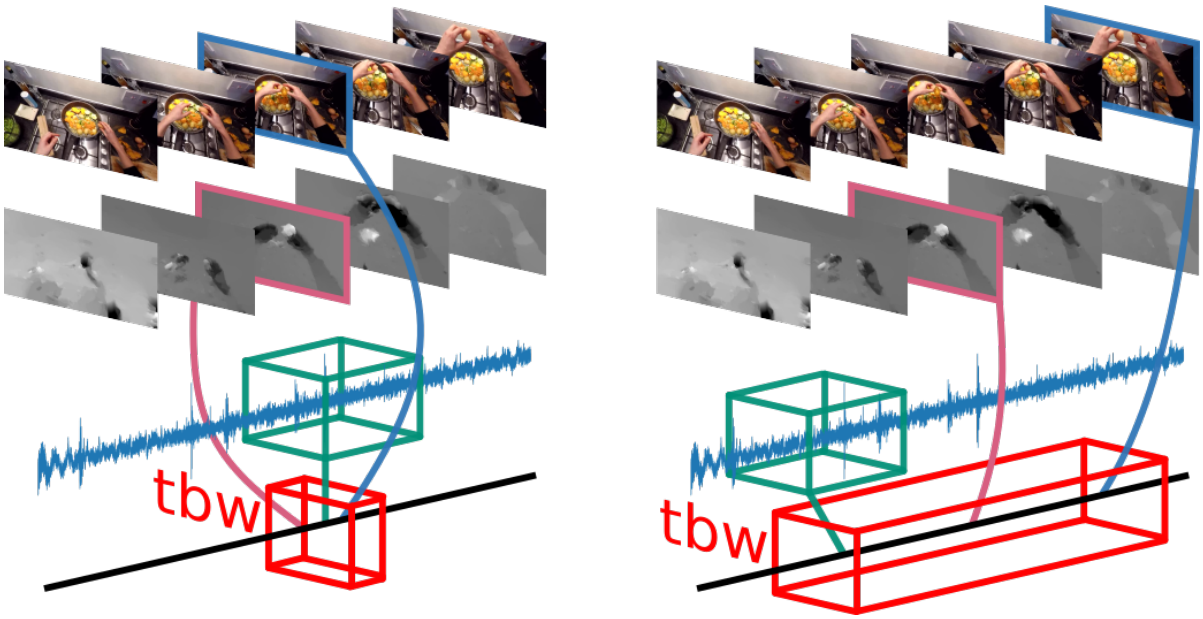


Figure 3.1: *A video of 'breaking an egg into a pan' with samples of appearance, motion and audio modalities. The Temporal Binding Window (TBW) allows the fusion of modalities within a range of temporal shifts, where the range is bounded by the width of the window. As the width of TBW increases (left to right), modalities are fused with increasingly varied temporal shifts, enabling asynchronous fusion, which is useful in this example as the sound of cracking the egg, the motion of separating it, and the appearance of the broken egg take place at different temporal positions in the video.*

nail [191]. It is also intriguing that infants have a very large TBW which narrows as they grow older [191].

Combining our explorations into audio for egocentric action recognition, and using a TBW for asynchronous modality fusion, this chapter's contributions are summarised as follows. First, an end-to-end trainable mid-level fusion Temporal Binding Network (TBN) is proposed. Second, we present the first audio-visual fusion attempt in egocentric action recognition. Third, we achieve state-of-the-art results on the EPIC-KITCHENS public leaderboards on both seen and unseen test sets. Our results show (i) the efficacy of audio for egocentric action recognition, (ii) the advantage of mid-level fusion within a TBW over late fusion, (iii) that multi-modal fusion can alleviate the class imbalance problem, and (iv) the robustness of our model to background or irrelevant sounds.[1]

[1]Python code of our TBN model and pre-trained model on EPIC-Kitchens are available at http://github.com/ekazakos/temporal-binding-network.

3.1 The Temporal Binding Network

Our goal is to find the optimal way to fuse multiple modality inputs while modelling temporal progression through sampling. We first explain the general notion of temporal binding of multiple modalities in Subsec. 3.1.1, then detail our architecture in Subsec. 3.1.2.

3.1.1 Multi-modal temporal binding

Consider a sequence of samples from one modality in a video stream, $m_i = (m_{i1}, m_{i2}, \cdots, m_{iT/r_i})$ where T is the video's duration and r_i is the modality's frame-rate (or frequency of sampling). For simplicity, we will be using T_i in the place of T/r_i. Input samples are first passed through uni-modal feature extraction functions f_i. To account for varying representation sizes and frame-rates, most multi-modal architectures apply pooling functions G to each modality in the form of average pooling or other temporal pooling functions (*e.g.* maximum or VLAD [86]), before attempting multi-modal fusion.

Given a pair of sequences of samples m_1 and m_2 from two modalities corresponding to the same video, the final class predictions for the video are hence obtained as follows:

$$y = h\big(G(z(f_1(m_1))), G(z(f_2(m_2)))\big) \tag{3.1}$$

where $f_i \in \mathbb{R}^{T_i \times S^1_{m_i} \times \ldots \times S^{L_i}_{m_i}} \rightarrow \mathbb{R}^{T_i \times D}$ are uni-modal feature extraction functions ($S^1_{m_i} \times \ldots \times S^{L_i}_{m_i}$: shape of modality m_i, D: dimension of extracted feature), $z \in \mathbb{R}^{T_i \times D} \rightarrow \mathbb{R}^{T_i \times C}$ is a uni-modal classifier (C: number of classes), $G \in \mathbb{R}^{T_i \times C} \rightarrow \mathbb{R}^{C}$ is a temporal aggregation function, $h \in \mathbb{R}^{C \times C} \rightarrow \mathbb{R}^{C}$ is the multi-modal fusion function and y is the output label for the video. In such architectures (*e.g.* TSN [194]), modalities' predictions are temporally aggregated before different modalities are fused; this is typically referred to as 'late fusion'.

Conversely, multi-modal fusion can be performed at *each* time step as in [46]. One way to do this would be to synchronise modalities and perform a multi-modal prediction at *each* time-step. For modalities with matching frame-rates, synchronised multi-modal samples can be selected as (m_{1k}, m_{2k}), and fused according to the following equation:

$$y = G(z(f_{sync}(m_{1k}, m_{2k}))) \tag{3.2}$$

where $f_{sync} \in \mathbb{R}^{T_{sync} \times (S^1_{m_1} \times \ldots \times S^{L_1}_{m_1}) \times (S^1_{m_2} \times \ldots \times S^{L_2}_{m_2})} \rightarrow \mathbb{R}^{T_{sync} \times D}$ is a multi-modal feature extractor that produces a representation for each time step k, $z \in \mathbb{R}^{T_{sync} \times D} \rightarrow \mathbb{R}^{T_{sync} \times C}$ is a multi-modal classifier, and $G \in \mathbb{R}^{T_{sync} \times C} \rightarrow \mathbb{R}^{C}$ then performs temporal aggregation over all time steps. Note that here $T_1 = T_2 = T_{sync}$. Basic differences with (3.1) are: i) here multi-modal fusion takes place within f_{sync} followed by temporal aggregation of *multi-modal*

predictions, therefore there is no h, while in (3.1) fusion takes place after temporal aggregation of *uni-modal predictions*, therefore feature extraction and fusion should be performed in two separate steps, ii) (3.2) requires modalities with matching frame-rates while in (3.1) frame-rates can differ as predictions are temporally aggregated before fused. The later is a limitation of (3.2) which can be tackled as explained next.

When frame-rates vary, and more importantly so do representation sizes, only approximate synchronisation can be attempted,

$$y = G(z(f_{sync}(m_{1k}, m_{2l}))) \quad : l = \lceil \frac{kr_2}{r_1} \rceil \tag{3.3}$$

by mapping samples of one modality to the samples of the other using their frame-rate ratio $\frac{r_2}{r_1}$. We refer to (3.2) and (3.3) as 'synchronous fusion' where synchronisation is achieved or approximated. Differently than (3.2), the temporal dimension of each function in (3.3) is T_1 as for each sample of m_1 a corresponding sample of m_2 is calculated.

A disadvantage of (3.3) is that the approximation is prone to errors. Approximation errors result from rounding up to the next integer to estimate l, which is necessary as we have access to discrete modalities' samples. These errors increase with the addition of more modalities. More importantly, a disadvantage of 'synchronous fusion', *i.e.* both (3.2) and (3.3), is that it is not able to model examples where the discriminative samples of each modality are at different temporal positions, as shown in Fig. 3.1, as modalities are fused with synchronous inputs. We provide a solution to both of these issues, as shown next in our proposal.

In this work, we propose fusing modalities *asynchronously* within temporal windows. Here, modalities are fused within a range of temporal offsets, with all offsets constrained to lie within a finite time window, which we henceforth refer to as a temporal binding window (TBW). Formally,

$$y = G(z(f_{tbw}(m_{1k}, m_{2l}))) \quad : l \in [\lceil \frac{kr_2}{r_1} - \frac{b}{2} \rceil, \lceil \frac{kr_2}{r_1} + \frac{b}{2} \rceil] \tag{3.4}$$

where f_{tbw} is a multi-modal feature extractor that combines inputs within a binding window of width b. Concretely, first m_{1k} is sampled from modality m_1. Then, a TBW of width b is placed around m_{1k} and m_{2l} is sampled within the window. Therefore, similarly to (3.3), the temporal dimension of each function in (3.4) is T_1 as for each sample of m_1 a sample of m_2 is randomly selected within the TBW. Note that (3.4) is a generalisation of both (3.2) and (3.3); when $b = 0$ it corresponds to (3.3), and when in addition $r_2 = r_1$ and therefore $l = k$, it corresponds to (3.2) as well. Hence, it is apparent that (3.4) is more flexible as it encapsulates 'synchronous fusion' but also extends to asynchronous sample combination. Interestingly, as the number of modalities increases, say from two to three modalities, the TBW representation

allows fusion of modalities each with different temporal offsets, yet within the same binding window of width b:

$$\begin{aligned} y = G(z(f_{tbw}(m_{1k}, m_{2l}, m_{3n}))) : l \in [\lceil \frac{kr_2}{r_1} - \frac{b}{2} \rceil, \lceil \frac{kr_2}{r_1} + \frac{b}{2} \rceil] \\ : n \in [\lceil \frac{kr_3}{r_1} - \frac{b}{2} \rceil, \lceil \frac{kr_3}{r_1} + \frac{b}{2} \rceil] \end{aligned} \tag{3.5}$$

This formulation hence allows a large number of different input combinations to be fused. This is different from proposals that fuse inputs over predefined temporal differences (*e.g.* [112]). Sampling within a temporal window allows fusing modalities with various temporal shifts, *up to* the temporal window width b. This: 1) enables straightforward scaling to multiple modalities with different frame-rates, removing the need for synchronisation approximation and therefore making the model robust to approximation errors, 2) allows training with a variety of temporal shifts, accommodating, say, different speeds of action performance and 3) provides a natural form of multi-modal data augmentation.

With the basic concept of a TBW in place, we now describe our proposed audio-visual fusion model, TBN.

3.1.2 TBN with sparse temporal sampling

Our proposed TBN architecture is shown in Fig. 3.2 (left), consisting of RGB, Flow and Audio modalities. First, the action video is divided into K segments of equal width. Within each segment, we select a random sample of the first modality $\forall k \in K : m_{1k}$. This ensures the temporal progression of the action is captured by sparse temporal sampling of this modality, as with previous works [194, 217], while random sampling within the segment offers further data for training. The sampled m_{1k} is then used as the centre of a TBW of width b. The other modalities are selected randomly from within each TBW (Eq. 3.5). In total, the input to our architecture in both training and testing is $K \times M$ samples from M modalities.

Within each of the K TBWs, we argue that the complementary information in audio and vision can be better exploited by combining the internal representations of each modality before temporal aggregation, and hence we propose a *mid-level* fusion. A ConvNet (per modality) extracts *mid-level* features, which are then fused through *concatenating* the modality features and feeding them to a non-linear fully-connected layer, making multi-modal predictions per TBW. Hence, we model f_{tbw} from Eq. 3.5 as:

$$f_{tbw}(m_{1k}, m_{2l}, m_{3n}) = \sigma(W[f_1(m_{1k}); f_2(m_{2l}); f_3(m_{3n})] + c), \tag{3.6}$$

where σ is a non-linear activation function, $[;]$ denotes input concatenation, and $f_1(m_{1k})$,

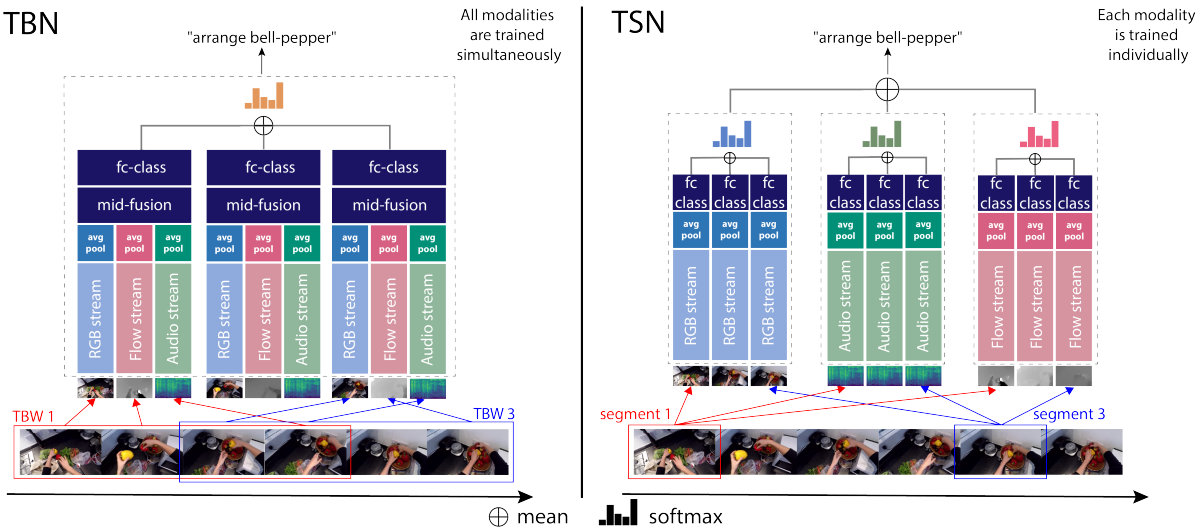


Figure 3.2: ***Left:*** *our proposed Temporal Binding Network (TBN). Modalities are sampled within a TBW, where average pooled convolutional features are extracted per modality, and fused with mid-level fusion. Modalities are trained jointly. A classifier takes as input the fused multi-modal features and performs a prediction per TBW. Predictions from multiple TBWs, possibly overlapping, are averaged. Modality-specific weights (same colour), as well as fusion and classification weights are shared amongst different TBWs.* ***Right:*** *TSN [194] with an additional audio stream performing late fusion. Here, predictions of each modality are first temporally aggregated independently, followed by modality fusion via averaging. Modalities are trained independently. Note that while in TSN a prediction is made for each modality, TBN produces a single prediction per TBW after fusing all modality representations. Best viewed in colour.*

$f_2(m_{2l})$ and $f_3(m_{3n})$ correspond to the mid-level features extracted from the RGB, Flow and Audio ConvNets, denoted as f_1, f_2 and f_3, respectively. Gradients are backpropagated all the way to the inputs of the ConvNets. Fig. 3.3 details the proposed TBN block. The predictions, for each of the K unified multi-modal representations (one per TBW), are obtained using z, and are finally temporally aggregated using G for video-level predictions. In the proposed architecture, we train all modalities simultaneously. The convolutional weights for each modality are shared over the K TBWs. Additionally, mid-level fusion weights and class prediction weights are also shared across the TBWs.

To avoid biasing the fusion towards longer or shorter action lengths, we calculate the window width b relative to the action video length. Our TBW is thus of variable width, where the width is a function of the length of the action. We note again that b can be set independently of the number of segments K, allowing the temporal windows to overlap. This is detailed in Subsec. 3.2.1.

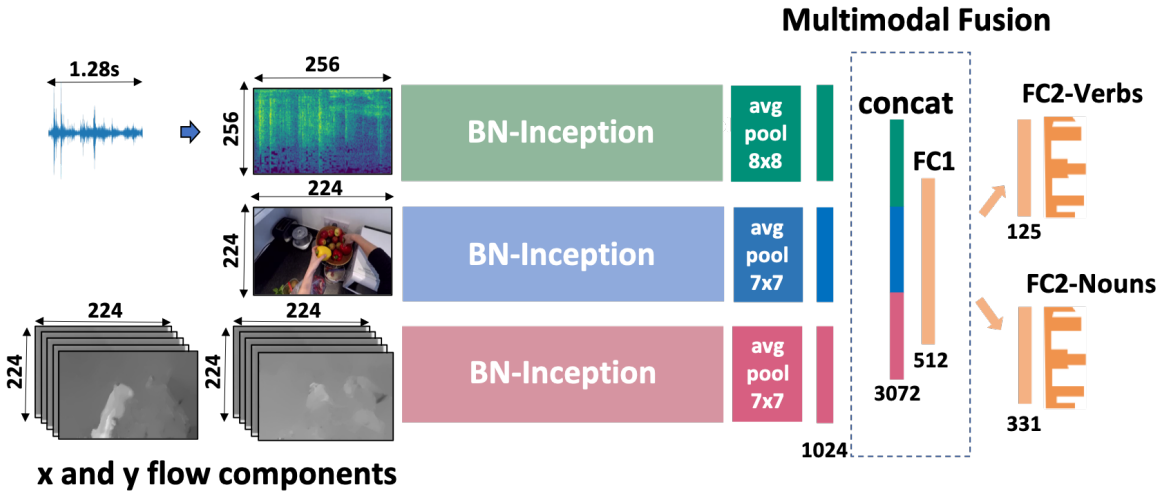


Figure 3.3: *A single TBN block showing architectural details and feature sizes. Audio is converted into a spectrogram, a single RGB frame is used as input to the appearance stream, while for motion, stacks of five consecutive horizontal and vertical optical flow frames are used. A ConvNet per modality extracts convolutional feature maps which are average pooled, and the resulting feature vectors are used as input to the fusion layer. Fusion takes the form of concatenation of modality features followed by a fully-connected layer that learns multi-modal relationships. We model the problem of learning both verbs and nouns as a multi-task learning problem, by adding two output fully-connected layers, one that predicts verbs and the other nouns (as in [35]). Outputs from multiple TBN blocks are averaged as shown in Fig. 3.2. Best viewed in colour.*

Relation to TSN. In Fig. 3.2, we contrast the TBN architecture (left) to an extended version of the TSN architecture (right). The extension is to include the audio modality, since the original TSN only utilises appearance and motion streams. There are two key differences: first, in TSN each modality is temporally aggregated independently (across segments), and the modalities are only combined by late fusion (*e.g.* the RGB scores of each segment are temporally aggregated, and the flow scores of each segment are temporally aggregated, individually), as in Eq. 3.1. Hence, it is not possible to benefit from combining modalities *within* a segment which is the case for TBN. Second, in TSN, each modality is trained independently first after which predictions are combined in inference. In the TBN model instead, all modalities are trained simultaneously, and their combination is also learnt.

3.2 Experimental Setup

We evaluate the TBN architecture on EPIC-KITCHENS-55 [35, 37].

3.2.1 Implementation details

RGB and Flow. We use the publicly available RGB and computed optical flow with the dataset [35]. RGB is extracted at 60fps whereas Flow at 30fps.

Audio processing. We extract 1.28s of audio, convert it to single-channel, and resample it to 24kHz. 1.28s of audio is a reasonable choice in our setup because it is $\approx \frac{1}{3} \times$avg_action_length in EPIC-KITCHENS-55, and we train TBN using $K = 3$ segments (more details on training hyperparameters can be found below).[2] The sampling rate is halved (original is 48kHz) to reduce the input size for tackling overfitting. The subsampled audio segment is then converted to a log-spectrogram representation using an STFT of window length 10ms, hop length 5ms and 256 frequency bands. This results in a 2D spectrogram matrix of size 256×256, after which we compute the logarithm. Preliminary experiments have shown that the window and hop length are important hyperparameters as our method is sensitive to those (this happens to be the case for most methods). The reason is that using overly large windows results in temporally coarse representations while very short windows can induce overfitting due to very large representation sizes (which imply a parameter increase in the model). An empirical rule for setting these values is that the window and hop length should scale linearly with the input length, where longer inputs require longer windows (to capture sufficiently global statistics of the input) while for shorter inputs shorter windows are adequate.[3] Finally, since many egocentric actions are very short ($<$ 1.28s), we extract 1.28s of audio from the untrimmed video, allowing the audio segment to extend beyond the action boundaries.

Architectural details. We implement our model in PyTorch [141]. We use Inception with Batch Normalisation (BN-Inception) [83] as a base architecture, and fuse the modalities after the average pooling layer, corresponding to the outputs of f_1, f_2 and f_3 in Eq. 3.6. We chose BN-Inception as it offers a good compromise between performance and model-size, critical for our proposed TBN that trains all modalities simultaneously, and hence is memory-intensive. Compared to TSN, the three modalities have 10.78M, 10.4M and 10.4M parameters, with only one modality in memory during training. In contrast, TBN has 32.64M parameters. In Eq. 3.6, we use a ReLU for the non-linearity, σ. For the temporal aggregation function, G, we use average. Finally, as shown in Fig. 3.3, we learn verbs and nouns simultaneously as in multi-task learning, and therefore the classifier z consists of two output fully-connected layers, for predicting verbs and nouns, respectively (as in [35]).

[2] Note that we don't set the audio input length to exactly $\frac{1}{3} \times$ avg_action_length $= 1.23$s as the backbone architecture [83] in TBN cannot operate on any arbitrary input size.

[3] In addition to the input length, the input domain is another important factor for choosing an appropriate window (and hop) length; for example, for instantaneous sounds like hitting a drum short windows are sufficient whereas for ambient sounds like the sound of the rain long windows work better.

Train / Val details. We train using SGD with momentum [148] and cross-entropy loss, a batch size of 128, a momentum of 0.9 and a learning rate of 0.01. Networks are trained for 80 epochs, and the learning rate is decayed by a factor of 10 at epoch 60. For regularisation, we employ dropout on the output of the fusion layer with a probability of 0.5, as well as a weight decay of $5 \cdot 10^{-4}$. Furthermore, we clip the parameter gradients using the value of 20. In the visual modalities, we apply the same data augmentations as the ones used in TSN, while we do not augment audio. We initialise the RGB and the Audio streams from ImageNet. While for the Flow stream, we use stacks of 10 interleaved horizontal and vertical optical flow frames, and use the pre-trained Kinetics [90] model, provided by the authors of [194]. We fine-tune on EPIC-KITCHENS by freezing the Batch-Normalisation layers except the first one, as done in TSN. Note that our network is trained end-to-end for all modalities and TBWs. We train with $K = 3$ segments over the $M = 3$ modalities. We allow TBWs to be as large as the action segment by setting $b/2 = T$ and trimming parts of the window that fall outside the action boundaries. Formally, in Eq. 3.5 we select the TBW boundaries as:

$$\begin{aligned} l &\in \left[\max(0, \lceil \frac{kr_2}{r_1} - T \rceil), \min(T, \lceil \frac{kr_2}{r_1} + T \rceil)\right] \\ n &\in \left[\max(0, \lceil \frac{kr_3}{r_1} - T \rceil), \min(T, \lceil \frac{kr_3}{r_1} + T \rceil)\right] \end{aligned} \tag{3.7}$$

Thus, there is full overlapping between all TBWs and the action boundaries. We test using 25 evenly spaced samples for each modality, and a single centre crop per sample for the visual modalities (spectrograms are not cropped). Note that TSN uses 10 crops per sample.

3.2.2 Evaluation metrics

We follow the evaluation protocol that we introduced in [35], and report aggregate and per-class metrics. For aggregate metrics, we compute top-1 accuracy and top-5 accuracy. Top-1 accuracy measures the ratio of correctly classified examples, yet it assigns the same weight to all examples disregarding the dataset's class distribution. Top-5 accuracy counts an instance as correctly classified if it is correctly predicted by any of the top-5 classifier's scores. It is particularly useful for fine-grained datasets with a large number of classes as a wrong top-1 prediction might still be meaningful in cases where it carries similar semantics with the ground-truth (*e.g.* cut onion vs cut garlic). For per-class metrics, we compute per-class precision and recall for classes with more than 100 examples in the training set, and average the results across classes. We call these average class precision and average class recall.[4] Average

[4] Average class precision should not be confused with the well known Average Precision, as the former calculates an overall precision score for each class while the latter calculates precision values at different recall ranks and then computes the area under the Precision-Recall curve.

		Top-1 Accuracy			Top-5 Accuracy			Avg Class Precision			Avg Class Recall		
	Modality	Verb	Noun	Action	Verb	Noun	Action	Verb	Noun	Action	Verb	Noun	Action
S1	RGB	45.68	36.80	19.86	85.56	64.19	41.89	61.64	34.32	09.96	23.81	31.62	08.81
	Flow	55.65	31.17	20.10	85.99	56.00	39.30	48.83	26.84	09.02	27.58	24.15	07.89
	Audio	43.56	22.35	14.21	79.66	43.68	27.82	32.28	19.10	07.27	25.33	18.16	06.17
	TBN (RGB+Flow)	60.87	42.93	30.31	89.68	68.63	51.81	**61.93**	39.68	18.11	39.99	38.37	16.90
	TBN (All)	**64.75**	**46.03**	**34.80**	**90.70**	**71.34**	**56.65**	55.67	**43.65**	**22.07**	**45.55**	**42.30**	**21.31**
S2	RGB	34.89	21.82	10.11	74.56	45.34	25.33	19.48	14.67	04.77	11.22	17.24	05.67
	Flow	48.21	22.98	14.48	77.85	45.55	29.33	23.00	13.29	05.63	19.61	16.09	07.61
	Audio	35.43	11.98	06.45	69.20	29.49	16.18	22.46	09.41	04.59	18.02	09.79	04.19
	TBN (RGB+Flow)	49.61	25.68	16.80	78.36	50.94	32.61	30.54	20.56	09.89	21.90	20.62	11.21
	TBN (All)	**52.69**	**27.86**	**19.06**	**79.93**	**53.78**	**36.54**	**31.44**	**21.48**	**12.00**	**28.21**	**23.53**	**12.69**

Table 3.1: *Comparison of our fusion method to single modality performance. For both splits, the fusion outperforms single modalities. TBN trained on audio-visual modalities (All) outperforms TBN trained on visual-only modalities (RGB+Flow). For the seen split (**S1**), the RGB and Flow modalities perform comparatively for actions, while RGB perform better for nouns and Flow for verbs. For the unseen split (**S2**), the Flow modality outperforms RGB for all verbs, nouns and actions. Audio is comparable to RGB on top-1 verb accuracy for both splits, while it can better predict verbs than nouns.*

class precision assesses the classifier's purity of predictions for each class while average class recall evaluates the coverage of different classes by the classifier. Both average class precision and recall are better suited for imbalanced datasets than accuracy as they consider the size of each class. We calculate all metrics over verbs, nouns as well as their valid combinations (actions).

3.3 Results

This section is organised as follows. First, we show and discuss the performance of single modalities, and compare them with our proposed TBN, with a special focus on the efficacy of the audio stream. Second, we compare different mid-level fusion techniques. And finally, we investigate the effect of the TBW width on both training and testing.

3.3.1 Single-modality vs multi-modal fusion performance

We examine the overall performance of each modality individually in Table 3.1. Although it is clear that RGB and optical flow are stronger modalities than audio, an interesting find is that audio performs comparably to RGB on some of the metrics (*e.g.* top-1 verb accuracy), signifying the relevance of audio on recognising egocentric actions. While as expected Flow outperforms RGB in **S2**, interestingly for **S1**, the RGB and Flow modalities perform compar-

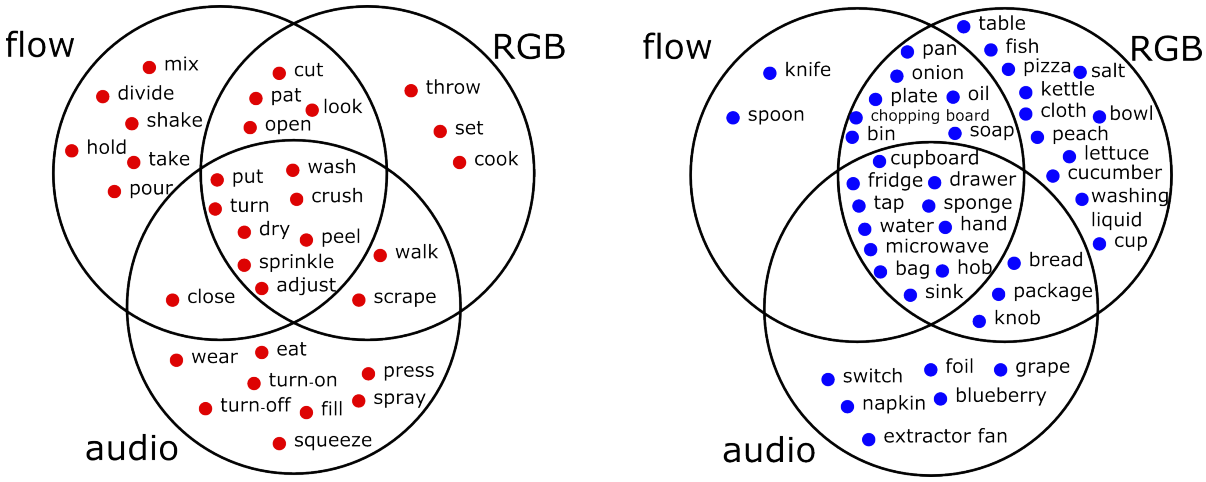


Figure 3.4: *Venn diagrams showing verb (left) and noun (right) classes' performances on the **S1** test set using single modalities for top-performing 32 verb and 41 noun classes, using single modality accuracy. For each class, we consider whether the accuracy is high for Flow, Audio or RGB, or for two or all of these modalities, and accordingly plot that class in the outer part of the modality's circle or in the intersection of two or all three circles, respectively. It can be clearly seen that noun classes can be predicted with high accuracy using RGB alone, whereas for verbs, Flow is more important. Audio is important for both verbs and nouns, but there is higher concentration of verb than noun classes in audio's circle which is also validated from Table 3.1 that shows that audio is more accurate for verbs.*

atively, and in some cases RGB performs better. This matches the expectation that optical flow is more invariant to the environment as models trained on it do not suffer from colour bias.

To obtain a better analysis of how these modalities perform, we examine the accuracy of *individual* verb and noun classes on **S1**, using single modalities. Fig. 3.4 plots top-performing verb and noun classes, into a Venn diagram. For each class, we consider the accuracy of individual modalities. If all modalities perform comparably (within 0.15), we plot that class in the intersection of the three circles. On the other hand, if one modality is clearly better than the others (more than 0.15), we plot the class in the outer part of the modality's circle. For example, for the verb 'close', we have per-modality accuracy of 0.23, 0.47 and 0.42 for RGB, Flow and Audio respectively. We thus note that this class performs best for two modalities: Flow and Audio, and plot it in the intersection of these two circles.

From this plot, many verb and noun classes perform comparably for all modalities (*e.g.* 'wash', 'peel' and 'fridge', 'sponge'). This suggests all three modalities contain useful information for these tasks. A distinctive difference, however, is observed in the importance of individual modalities for verbs and nouns. Verb classes are strongly related to the temporal progression of actions, or in other words the motion of the hands of the wearer, making Flow more important

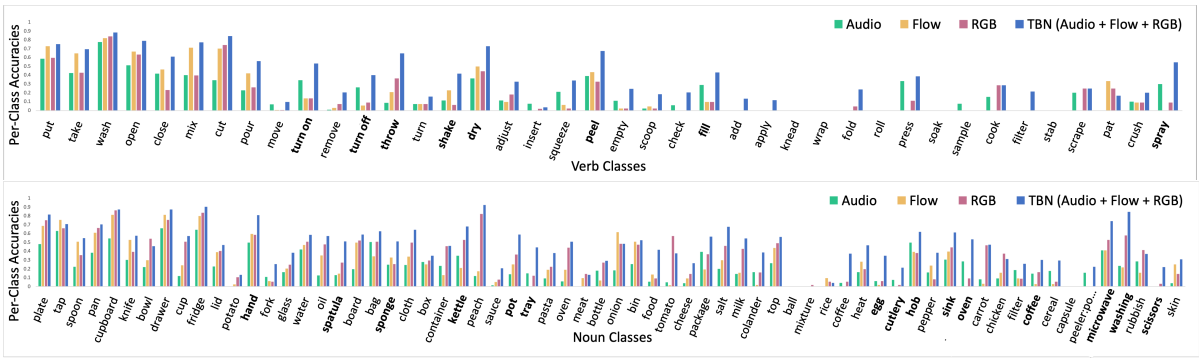


Figure 3.5: *Per-class accuracies on the **S1** test set for verbs (top) and nouns (bottom), for fusion and single modalities. We select verb classes with more than 10 samples, and noun classes with more than 30 samples. The classes are presented in decreasing order of number of samples per class, from left to right. For many classes, the fusion method provides significant performance gains over single modality classification (largest improvements shown in bold). Best viewed in colour.*

for verbs than nouns. Conversely, noun classes can be predicted with high accuracy using RGB alone showing the importance of appearance/colour in classifying objects. This is also manifested in Table 3.1, where in **S1** Flow has higher accuracy for verbs and RGB for nouns. Audio, on the other hand, is important for both nouns and verbs, particularly for some verbs such as 'turn-on', and 'spray'. For nouns, Audio tends to perform better for objects with distinctive sounds (*e.g.* 'switch', 'extractor fan') and materials that sound when manipulated (*e.g.* 'foil'). Nevertheless, as demonstrated in Table 3.1, audio is more informative for verbs than nouns, attributed to the fact that there are several soundless objects, as it will be further shown next that we look at per-class accuracies.

In Table 3.1, we compare single modality performance to the performance over the three modalities. Single modalities are trained as in TSN, as TBN is designed to bind multiple modalities. We find that the fusion method outperforms single modalities, and that audio is a significantly informative modality across the board. Per-class accuracies (on **S1**), for individual modalities as well as for TBN trained on all three modalities, can be seen in Fig. 3.5. The advantage of the fusion method is more pronounced for verbs (where we expect motion and audio to be more informative) than nouns, and more for particular noun classes than others, such as 'pot', 'kettle', 'microwave', and particular verb classes *e.g.* 'spray' (fusion 0.54, RGB 0.09, Flow 0, Audio 0.3). This suggests that the mixture of complementary and redundant information captured in a video is highly dependent on the action itself, yielding the fusion method to be more useful for some classes than for others. To further demonstrate that, in Tables 3.2a and 3.2b, we show per-class accuracies on **S1**, on selected verbs and nouns, respectively. We arrange the chosen set of verbs and nouns in three main categories: **top:** TBN outperforms the best individual modality, **mid:** TBN performs comparably with the best modality, and **bot-**

Verb	RGB	Flow	Audio	TBN
open	63.32	66.81	51.05	79.08
walk	55.56	11.11	55.56	88.89
turn-on	13.79	13.79	33.79	53.10
scoop	02.27	04.55	02.27	18.18
look	14.29	14.29	00.00	28.57
scrape	25.00	00.00	16.67	25.00
hold	00.00	20.00	00.00	20.00
set	33.33	00.00	16.67	33.33
cook	28.57	00.00	14.29	28.57
finish	00.00	00.00	16.67	16.67
insert	01.79	00.00	07.14	03.57
divide	00.00	40.00	00.00	20.00
sprinkle	00.00	00.00	11.11	00.00
sample	00.00	00.00	07.14	00.00
pat	25.00	33.33	00.00	16.67

(a) *Verbs*

Noun	RGB	Flow	Audio	TBN
paella	25.00	00.00	00.00	50.00
fridge	83.25	80.10	60.73	87.96
hand	56.38	59.73	43.62	76.51
sponge	25.27	32.97	23.08	48.35
salt	40.98	27.87	16.39	62.30
switch	50.00	00.00	75.00	75.00
knife	36.29	52.12	27.80	52.12
salad	14.29	19.05	04.76	19.05
tortilla	42.86	00.00	14.29	42.86
leaf	00.00	10.00	10.00	10.00
pizza	100.00	09.09	36.36	72.73
fish	90.00	00.00	00.00	50.00
bowl	51.49	29.79	19.57	42.98
chicken	31.58	15.79	07.89	26.32
paper	03.70	00.00	14.81	07.41

(b) *Nouns*

Table 3.2: *Top-1 accuracy of selected verbs and nouns on the **S1** test set for individual modalities and for TBN (Single Model). Shading of rows reflects grouping of verbs/nouns in three categories: **top:** TBN performs better than the best modality, **mid:** TBN performs on par with the best modality, and **bottom:** TBN performs worse than the best modality.*

tom: TBN performs worse than the best individual modality. We shade the rows reflecting these three groups in the order mentioned above.

A few conclusions could be made from these tables about properties of the proposed mid-level fusion:

1. Fusion can improve results when all modalities are individually performing well for both verb and noun classes (*e.g.* 'open', 'fridge'), as well as when all modalities are under-performing (*e.g.* 'scoop', 'sponge').
2. Fusion can though be difficult at times, particularly when two of the three modalities are uninformative (*e.g.* 'divide', 'fish').
3. All nouns for which audio is outperforming other modalities have distinct sounds (*e.g.* 'switch', 'paper').
4. Similarly, audio is least distinctive when the noun does not have a sound per se or its sound depends on the action (*e.g.* 'chicken', 'salt').

We also note that the fusion method helps to significantly boost the performance of the tail

		RGB	Flow	Audio	TBN
Verb	Top-10%	41.90	49.29	35.02	63.00
	Tail	02.85	02.26	04.21	11.00
Noun	Top-10%	39.52	34.23	22.36	46.50
	Tail	06.22	02.38	04.76	13.02

Table 3.3: *Comparison of mean class accuracy on the **S1** test set, for the top-10% of classes when ranked by class size and the remaining (tail) classes, for individual modalities and TBN. It can be seen that fusion, clearly, has a greater effect on the tail classes, where single modalities perform very poorly (rate of improvement of TBN on tail $\gg$ rate of improvement of TBN on top-10%).*

classes, where individual modality performance tends to suffer. This can be seen in Fig. 3.5, right, and further discussed in the next subsection.

3.3.2 Effect of fusion on tail classes

Table 3.3 shows a comparison of the performance of the top largest classes against the less represented classes on the **S1** test set, for individual modalities, and our proposed TBN. The classes are ranked by the number of examples in training, and the results are reported separately for the top-10% classes versus the rest which we refer to as *tail classes*. The effect of fusion (compared to the best individual modality in each case) is more evident on the tail classes – 161.2% improvement on tail vs. 27.8% improvement on top-10% for verbs, and 109.3% improvement on tail vs. 17.6% improvement on top-10% for nouns. This finding shows that fusion in TBN decreases the effect of the class-imbalance. Furthermore, it is important to note that audio outperforms RGB and flow on the tail verbs.

3.3.3 Efficacy of audio

We train TBN only with the visual modalities (RGB+Flow) and the results can be seen in Table 3.1. An increase of 5% (S1) and 4% (S2) in top-5 action recognition accuracy with the addition of audio demonstrates the importance of audio for egocentric action recognition. Fig. 3.6 shows the confusion matrices without and with the utilisation of audio for the largest-15 verb and noun classes (in **S1**). The confusion matrices on the left show TBN (RGB+Flow), and the ones on the middle show TBN (RGB+Flow+Audio). Studying the difference (Fig. 3.6 right) clearly (yet not uniformly) demonstrates an increase (blue) in confidence along the diagonal, and a decrease (red) in confusion elsewhere.

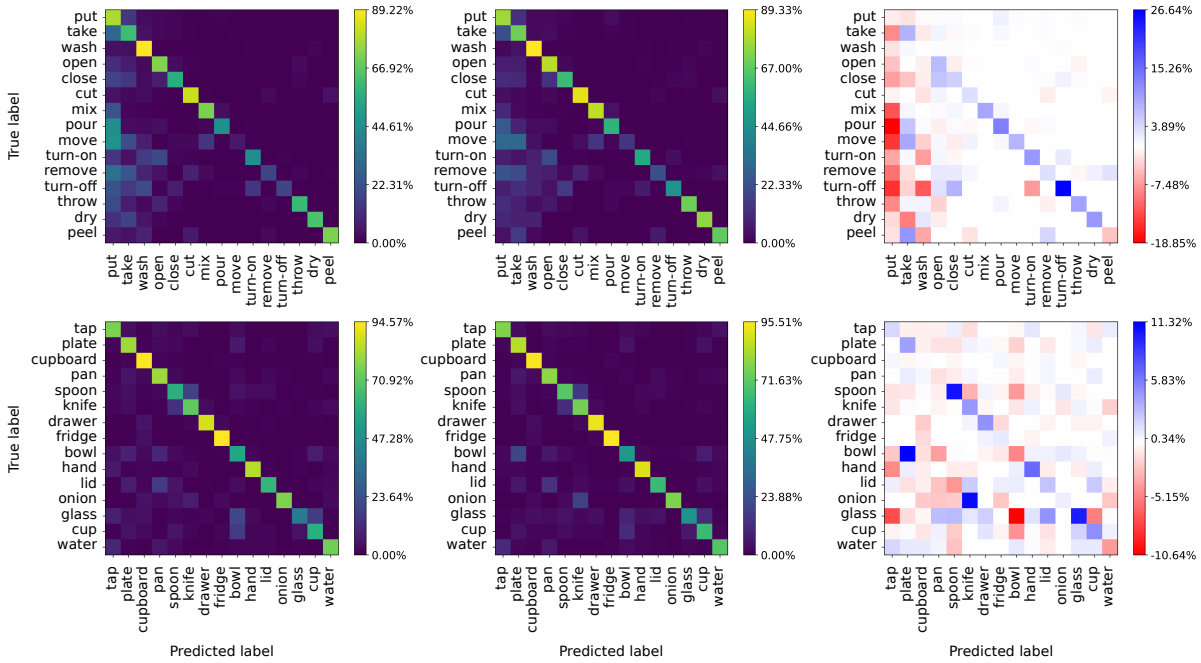


Figure 3.6: *Confusion matrices for largest-15 verb classes (top) and the largest-15 noun classes (bottom) in the **S1** test set, without (left) and with (middle) audio, as well as their difference (right). The difference shows increase (blue) in confidence in the main diagonal and decrease (red) in confusion elsewhere, when audio is incorporated.*

		All			RGB+Flow		
	TBN	Verb	Noun	Action	Verb	Noun	Action
S1	irrelevant	61.37	46.46	32.63	57.28	42.55	27.73
	rest	65.28	45.97	35.14	61.44	42.99	30.72
S2	irrelevant	47.32	23.36	15.30	44.41	20.45	12.39
	rest	57.21	31.66	22.22	54.00	30.09	20.52

Table 3.4: *Comparing top-1 accuracy of All modalities (left) to RGB+Flow (right). Actions are split in segments with 'irrelevant' background sounds, and the 'rest' of the test set. With the addition of audio, the model's accuracy increases consistently, even for the 'irrelevant' segments.*

3.3.4 Audio with irrelevant sounds

In the recorded videos of EPIC-Kitchens, background sounds irrelevant to the observed actions have been captured by the wearable sensor. These include music or TV playing in the background, ongoing washing machine, coffee machine or frying sounds while other actions take place. To quantify the effect of these sounds, we annotated the audio in the test set, and report that 14% of all action segments in S1, and 46% of all action segments in S2 contain other audio sources. We refer to these as actions containing 'irrelevant' sounds, and report

		Top-1 Accuracy			Top-5 Accuracy			Avg Class Precision			Avg Class Recall		
	Fusion technique	Verb	Noun	Action	Verb	Noun	Action	Verb	Noun	Action	Verb	Noun	Action
S1	Concatenation	**64.75**	**46.03**	**34.80**	**90.70**	**71.34**	**56.65**	55.67	**43.65**	**22.07**	45.55	**42.30**	**21.31**
S1	Context gating [125]	63.77	44.33	33.47	90.04	69.09	54.10	**57.31**	42.20	21.72	**45.63**	41.53	20.20
S1	Gating fusion [9]	61.52	43.54	31.61	89.54	68.42	52.57	52.07	39.62	18.39	42.55	39.77	18.66
S2	Concatenation	**52.69**	**27.86**	19.06	**79.93**	**53.78**	**36.54**	**31.44**	21.48	12.00	**28.21**	**23.53**	12.69
S2	Context gating [125]	52.65	27.35	**19.16**	79.25	52.00	36.40	30.82	**23.16**	11.72	23.39	25.03	12.58
S2	Gating fusion [9]	50.16	27.25	18.41	78.80	50.84	34.04	28.42	22.42	**12.34**	23.92	24.15	**13.14**

Table 3.5: *Comparison of mid-level fusion techniques for the TBN architecture. The fusion techniques are presented in increasing order of number of parameters, from top to bottom. Concatenation surpasses the more sophisticated approaches. Performance drops with the increase in parameters.*

results independently for the subset with irrelevant background audio, and the 'rest' of the test set, in Table 3.4. Results are shown for both TBN trained on all modalities (All) and for TBN trained on visual-only modalities (RGB+Flow). The table shows that when audio is incorporated, the model performs consistently better, even when 'irrelevant' sounds occur in the background. Both models (All and RGB+Flow) show a drop in performance for 'irrelevant' comparing to 'rest', validating that irrelevant sounds are not the source of confusion, but that this set of action segments is more challenging even in the visual modalities, where audio is absent and therefore irrelevant sounds cannot affect the model's performance. This demonstrates the robustness of our network to noisy and unconstrained audio sources.

3.3.5 Comparison of fusion strategies

As Fig. 3.2 indicates, TBN performs mid-level fusion on the modalities within the binding window, with concatenation, as shown in Eq. 3.6 and Fig. 3.3. Here, we describe two alternative mid-level fusion strategies to concatenation and then compare their performances.

(i) *Context gating* was used in [125], aiming to recalibrate the strength of the activations of different units with a self-gating mechanism:

$$f_{tbw}^{context} = \sigma(Wv + c) \circ v, \tag{3.8}$$

where σ is a sigmoid function and $\circ$ is element-wise multiplication. We apply context gating on top of our multi-modal fusion with concatenation, so v in (3.8) is equivalent to (3.6).

(ii) *Gating fusion* was introduced in [9], where a gate neuron, g, takes as input the features

from all modalities to learn the importance of one modality w.r.t. all modalities.

$$v_i = \sigma_1(W_i f_i(m_{ij}) + c_i) \quad \forall i \in [1,2,3] \quad \forall j \in [k,l,n] \tag{3.9}$$

$$g_i = \sigma_2(W_{g_i}[f_1(m_{1k}); f_2(m_{2l}); f_3(m_{3n})] + c_{g_i}) \quad \forall i \in [1,2,3] \tag{3.10}$$

$$f_{tbw}^{gating} = g_1 \circ v_1 + g_2 \circ v_2 + g_3 \circ v_3, \tag{3.11}$$

where σ_1 is a hyperbolic tangent function and σ_2 is a sigmoid function. v_i and g_i use different activation functions (σ_1 and σ_2), as the design of Eq. 3.9-3.11 in [9] was inspired by LSTMs that perform gating using sigmoid activation functions to model the contribution of past and current cell states to the new cell state, while the current state is non-linearly projected with a tanh activation function.

In Table 3.5, we compare the various fusion strategies. We find that the simplest method, concatenation (Eq. 3.6) generally outperforms more complex fusion approaches. Although the other two strategies perform comparably to concatenation, we observe that the performance drops with the increase in the fusion technique's parameters; gating fusion that has the most parameters performs the worst in most metrics. Therefore, these approaches might perform better if trained in larger datasets.

3.3.6 The effect of TBW width

Here, we investigate the effect of the TBW width in training and testing. We varied the TBW width with $b \in \{\frac{T}{60}, \frac{T}{30}, \frac{T}{24}, \frac{T}{15}, \frac{T}{9}, \frac{T}{6}, \frac{T}{3}, T\}$ (remember that the width of the TBW is variable with the action segment length, T, as opposed to using a fixed width). This corresponds, on average, to varying the width of TBW on the training set between 60ms and 3750ms while on the **S1** test set between 60ms and 3650ms.

To evaluate the effect of the window width in training, we trained eight TBN models for each respective window width as well as one with synchrony $b \sim 0$, while testing all with synchrony. [5] To test the effect of the window width during inference, we use the model with the optimal width in training and evaluate it in an asynchronous fashion for each width, as follows: within a window of a given width, we randomly sample 25 inputs per modality, pass them to TBN and average the predictions to marginalise the noise from random sampling. We also compare with synchrony in testing. We train and test the models using a *single TBW*, to solely assess the effect of the window size, and disentangle it from the effect of temporal aggregation

[5] By the time of writing the paper in which we presented TBN, we had trained all the models in the paper with different random seeds per optimisation run, including the experiments of varying the TBW width in training. We re-trained the models of different window widths in this subsection with a fixed seed for a more conclusive evaluation (in the rest of the chapter models are trained with random seeds).

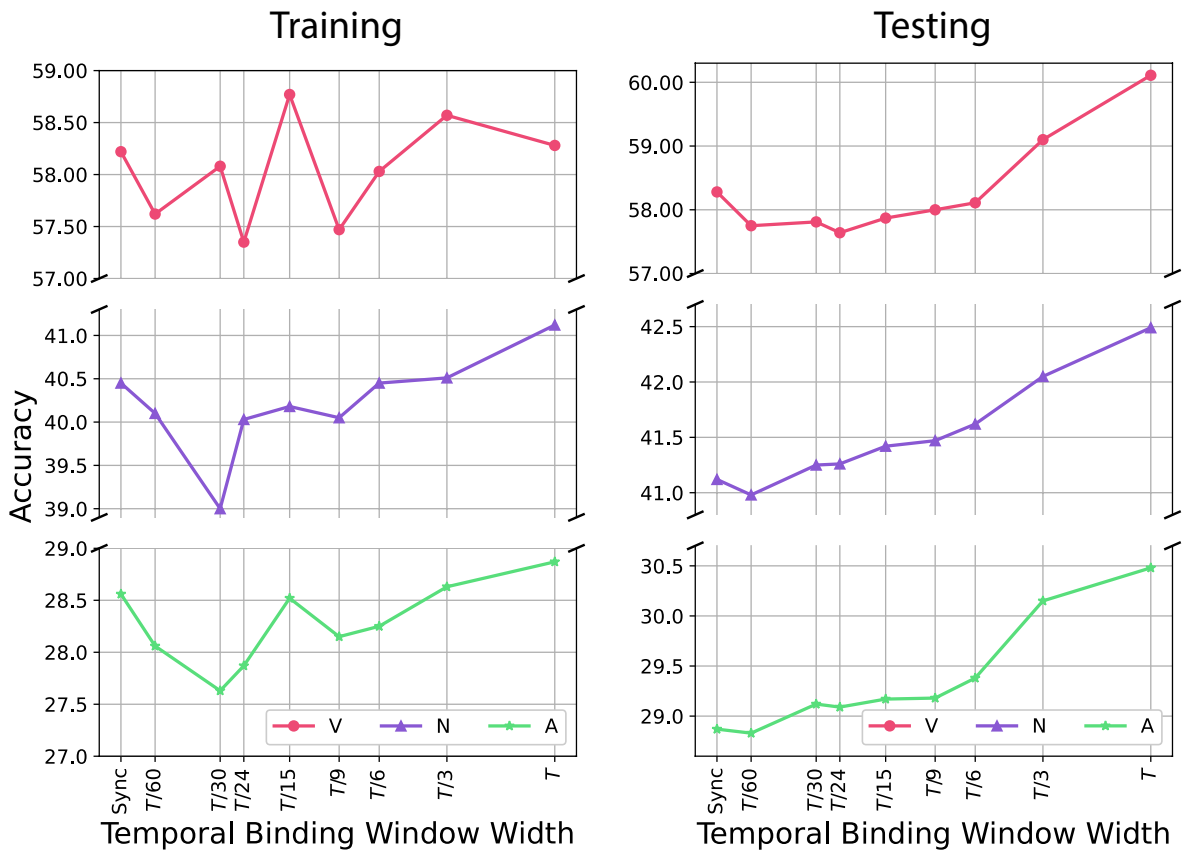


Figure 3.7: *Effect of the TBW width in training (left) and inference (right), on verb (V), noun (N) and action (A) top-1 accuracy, evaluated on the* ***S1*** *split. The different widths are multiples of the action length, T, as the width of the window is variable with T rather than fixed. To assess the effect of the width in training, eight TBNs are trained, one for each width, by sampling a single TBW per action segment, and each model is evaluated with synchrony ($b \sim 0$) by using the centre sample of the video for each modality. We also train a TBN with synchrony for comparison. For the effect of the width in testing, the model with the optimal width in training is utilised and evaluated asynchronously for each width as well as with synchrony (Sync); for asynchronous evaluation, the centre sample of the video is used for RGB, while 25 Flow and audio random samples are extracted within a TBW centred on the RGB sample and their predictions are averaged. The range in the y-axis is cut to emphasise the details. The best performance is obtained using the largest width $b = T$ in both training and testing, excluding verbs in training.*

of multiple TBWs. Accordingly, in testing, for synchrony we use the centre sample from the video for each modality, whereas for asynchronous evaluation we use the centre sample from the video for RGB, and sample Flow and audio inputs from a TBW centred at the RGB sample. All evaluations are performed on the **S1** test set.

Results are shown in Fig. 3.7, where comparisons are performed using top-1 accuracy of verbs (V), nouns (N) and actions (A). The effect of the TBW width is more evident in testing, where

3.3 Results

	Model	Top-1 Accuracy			Top-5 Accuracy			Avg Class Precision			Avg Class Recall		
		Verb	Noun	Action	Verb	Noun	Action	Verb	Noun	Action	Verb	Noun	Action
S1	Attention Clusters [115]	40.39	19.37	11.09	78.13	41.73	24.36	21.17	09.65	02.50	14.89	11.50	03.41
S1	[35] (from leaderboard)	48.23	36.71	20.54	84.09	62.32	39.79	47.26	35.42	11.57	22.33	30.53	09.78
S1	Ours (TSN [194] w. Audio)	55.49	36.27	23.95	87.04	64.17	44.26	53.85	30.94	13.55	30.60	29.82	11.11
S1	**Ours (TBN, Single Model)**	64.75	46.03	34.80	90.70	71.34	56.65	55.67	43.65	22.07	45.55	42.30	21.31
S1	**Ours (TBN, Ensemble)**	**66.10**	**47.89**	**36.66**	**91.28**	**72.80**	**58.62**	**60.74**	**44.90**	**24.02**	**46.82**	**43.89**	**22.92**
S2	Attention Clusters [115]	32.37	11.95	05.60	69.89	31.82	15.74	17.21	03.86	01.84	11.59	07.94	02.64
S2	[35] (from leaderboard)	39.40	22.70	10.89	74.29	45.72	25.26	22.54	15.33	06.21	13.06	17.52	06.49
S2	Ours (TSN [194] w. Audio)	46.61	22.50	13.05	78.19	48.59	29.13	28.92	15.48	06.47	21.58	16.61	07.55
S2	**Ours (TBN, Single Model)**	52.69	27.86	19.06	79.93	53.78	36.54	31.44	21.48	**12.00**	**28.21**	23.53	12.69
S2	**Ours (TBN, Ensemble)**	**54.46**	**30.39**	**20.97**	**81.23**	**55.69**	**39.40**	**32.57**	**21.68**	10.96	27.60	**25.58**	**13.31**

Table 3.6: *Results on EPIC-Kitchens for S1 and S2 test sets. TBN (Single Model) with mid-level fusion significantly outperforms a late fusion of the same modalities (TSN w. Audio). An ensemble of five TBNs trained with different TBW widths provides further boost in performance.*

performance of nouns and actions consistently increases with the increase in the TBW width where $b = T$ obtains the highest accuracy, showcasing that asynchronous combination of modalities in test time is advantageous. Synchrony is performing only slightly better than $b = \frac{T}{60}$, which is expected due to noise in sampling. For verbs, inference-time asynchronous fusion is also helpful where long TBWs with $b \in \{\frac{T}{3}, T\}$ provide better accuracy than synchrony, while performance drops for widths in between.

In training, there is a fluctuation in performance comparing synchrony and the various TBW widths. Although most asynchronous models are performing comparably or worse with synchrony, optimal performance is attained for $b > 0$; the optimal width for nouns and actions is $b = T$ whereas for verbs $b = \frac{T}{15}$, demonstrating the benefits of asynchronous fusion in training as well. We attribute the variation in performance in training as well as the low performance of intermediate widths for verbs in testing to different widths providing complementary benefits to different types of classes, which is supported by the results in the next subsection where we show that an ensemble of TBNs trained with different TBW widths improves the performance.

Finally note that in Fig. 3.7, we compare widths on a single temporal window. When we temporally aggregate multiple TBWs, in both training and testing, the effect of the TBW width is smoothed and the model becomes robust to TBW widths. Nevertheless, the results of our ensemble in the next subsection convey the gains of exploiting TBWs of different widths, even with temporal aggregation.

3.3.7 Comparison with the state of the art

We compare our work to the baseline results reported in [35] in Table 3.6 on all metrics. The baseline model in [35] is a TSN with late fusion of RGB+Flow. First, we show that a late fusion with an additional audio stream (TSN w. Audio), outperforms the baseline on top-1 verb accuracy by 7% on S1 and also 7% on S2. This complements our finding about the usefulness of audio which is shown in Table 3.1 for mid-level fusion, as here it is also shown for late-fusion. Second, we show that our TBN (Single Model) improves these results significantly (9%, 10% and 11% on top-1 verb, noun and action accuracy on S1, and 6%, 5% and 6% on S2, respectively), validating our assumption that combining modalities before temporal aggregation with mid-level learnable fusion is more efficient than late fusion with averaging. Finally, we report results of an Ensemble of five TBNs, where each one is trained with a different TBW width. The ensemble shows additional improvement of up to 3% on top-1 metrics, demonstrating that TBWs of different widths lead to TBNs that learn to predict actions in a complementary way.

We compare TBN with Attention Clusters [115], a previous effort to utilise RGB, Flow, and Audio for action recognition, using *pre-extracted features*. We use the authors' available implementation, and fine-tuned features (TSN, BN-Inception) on EPIC-Kitchens, from the global avg pooling layer (1024D), to provide a fair comparison to TBN, and follow the implementation choices from [115]. The method from [115] performs significantly worse than the baseline, as pre-extracted video features are used to learn attention weights, signifying the need for end-to-end training.

In Fig. 3.8, we show results for **Ours (TBN, Single Model)** and **Ours (TBN, Ensemble)**, as they appeared on the public leaderboard of the EPIC-Kitchens - Action recognition challenge on CodaLab at the time of submission (March 22nd 2019). The single model TBN outperforms all other submissions on all metrics by a clear margin, on both test sets **S1** and **S2**, and the results are further improved using the ensemble of TBNs. In particular, our TBN Ensemble results demonstrated an overall improvement over all state-of-the-art, published or anonymous, by 11%, 11% and 13% on top-1 verb, noun and action for S1 and by 11%, 7% and 6% on top-1 verb, noun and action for S2, respectively. As the challenge concluded, our model (TBN_Ensemble) was ranked 2nd in the leaderboard. A snapshot of the leaderboard for the 2019 challenge is available at
https://epic-kitchens.github.io/2019#results.

3.4 Qualitative Results

Seen Kitchens (S1)

#	User	Entries	Date of Last Entry	Team Name	Top-1 Accuracy (%)			Top-5 Accuracy (%)			Precision (%)			Recall (%)		
					Verb ▲	Noun ▲	Action ▲	Verb ▲	Noun ▲	Action ▲	Verb ▲	Noun ▲	Action ▲	Verb ▲	Noun ▲	Action ▲
1	TBN_Ensemble	1	03/22/19		66.10 (1)	47.88 (1)	36.66 (1)	91.28 (1)	72.80 (1)	58.62 (1)	60.73 (2)	44.89 (1)	24.01 (1)	46.81 (1)	43.88 (1)	22.92 (1)
2	TBN_Single_Model	1	03/22/19		64.74 (2)	46.03 (2)	34.80 (2)	90.69 (2)	71.33 (2)	56.64 (2)	55.66 (3)	43.65 (2)	22.06 (2)	45.55 (2)	42.30 (2)	21.30 (2)
3	zolfagha	7	10/18/18		55.08 (3)	35.50 (5)	23.08 (3)	84.81 (4)	60.92 (6)	42.91 (4)	42.98 (6)	31.02 (5)	14.68 (3)	34.08 (3)	33.84 (3)	13.99 (3)
4	EPIC_TSN_Fusion	1	09/04/18		48.23 (4)	36.71 (4)	20.54 (4)	84.09 (6)	62.32 (4)	39.79 (6)	47.26 (5)	35.42 (3)	11.57 (5)	22.33 (7)	30.53 (5)	9.78 (5)
5	masterchef	1	10/02/18		43.51 (7)	32.94 (6)	20.19 (5)	84.38 (5)	61.66 (5)	43.57 (3)	28.42 (9)	27.99 (8)	7.62 (7)	24.18 (5)	26.83 (7)	8.85 (7)
6	EPIC_TSN_RGB	1	09/04/18		45.68 (5)	36.80 (3)	19.86 (6)	85.56 (3)	64.19 (3)	41.89 (5)	61.64 (1)	34.32 (4)	11.02 (6)	23.81 (6)	31.62 (4)	9.76 (6)
7	zhun	4	01/09/19		43.59 (6)	31.12 (7)	17.94 (7)	83.10 (7)	55.37 (8)	33.37 (7)	48.16 (4)	27.71 (9)	12.72 (4)	24.90 (4)	27.91 (6)	10.32 (4)
8	EPIC_2SCNN_RGB	2	09/06/18		40.44 (10)	30.46 (8)	13.67 (8)	83.04 (8)	57.05 (7)	33.25 (8)	34.74 (7)	28.23 (7)	6.66 (8)	15.90 (8)	23.23 (8)	5.47 (8)
9	EPIC_2SCNN_Fusion	1	09/06/18		42.16 (9)	29.14 (9)	13.23 (9)	80.58 (9)	53.70 (9)	30.36 (9)	29.39 (8)	30.73 (6)	5.92 (9)	14.83 (10)	21.10 (9)	4.93 (9)
10	EPIC_TSN_Flow	1	09/04/18		42.75 (8)	17.40 (11)	9.02 (10)	79.52 (10)	39.43 (11)	21.92 (10)	21.42 (10)	13.75 (11)	2.58 (10)	15.58 (9)	9.51 (11)	2.28 (10)
11	EPIC_2SCNN_Flow	1	09/06/18		39.00 (11)	13.93 (12)	6.01 (11)	77.03 (11)	33.98 (12)	16.90 (11)	16.33 (11)	5.59 (12)	1.10 (12)	12.19 (11)	6.47 (12)	1.32 (12)
12	szhang_	3	03/11/19		24.23 (12)	21.80 (10)	4.76 (12)	71.46 (12)	44.22 (10)	15.74 (12)	5.87 (12)	21.82 (10)	2.00 (11)	4.86 (12)	14.72 (10)	1.47 (11)
13	chinmayd	1	03/22/19		0.83 (13)	0.25 (13)	0.00 (13)	4.21 (13)	1.24 (13)	0.01 (13)	3.92 (13)	0.98 (13)	0.00 (13)	1.40 (13)	0.17 (13)	0.00 (13)

Unseen Kitchens (S2)

#	User	Entries	Date of Last Entry	Team Name	Top-1 Accuracy (%)			Top-5 Accuracy (%)			Precision (%)			Recall (%)		
					Verb ▲	Noun ▲	Action ▲	Verb ▲	Noun ▲	Action ▲	Verb ▲	Noun ▲	Action ▲	Verb ▲	Noun ▲	Action ▲
1	TBN_Ensemble	1	03/22/19		54.46 (1)	30.39 (1)	20.96 (1)	81.22 (1)	55.68 (1)	39.40 (1)	32.56 (1)	21.67 (1)	10.96 (2)	27.60 (2)	25.58 (1)	13.31 (1)
2	TBN_Single_Model	1	03/22/19		52.68 (2)	27.86 (2)	19.05 (2)	79.92 (2)	53.77 (2)	36.53 (2)	31.43 (2)	21.47 (2)	11.99 (1)	28.21 (1)	23.53 (2)	12.68 (2)
3	masterchef	1	10/02/18		39.30 (6)	22.43 (4)	14.10 (3)	76.41 (3)	47.35 (3)	32.43 (3)	20.42 (5)	15.96 (4)	4.83 (6)	16.95 (4)	17.72 (4)	8.46 (3)
4	zolfagha	7	10/18/18		43.77 (3)	21.92 (5)	12.73 (4)	74.84 (4)	44.86 (6)	26.32 (4)	24.45 (3)	19.12 (3)	8.37 (3)	19.47 (3)	18.41 (3)	7.83 (4)
5	EPIC_TSN_Fusion	1	09/04/18		39.40 (5)	22.70 (3)	10.89 (5)	74.29 (6)	45.72 (4)	25.26 (6)	22.54 (4)	15.33 (5)	6.21 (4)	13.06 (6)	17.52 (5)	6.49 (5)
6	EPIC_TSN_RGB	1	09/04/18		34.89 (9)	21.82 (6)	10.11 (6)	74.56 (5)	45.34 (5)	25.33 (5)	19.48 (7)	14.67 (7)	5.32 (5)	11.22 (8)	17.24 (6)	6.34 (6)
7	EPIC_2SCNN_Fusion	1	09/06/18		36.16 (8)	18.03 (7)	7.31 (7)	71.97 (9)	38.41 (8)	19.49 (8)	18.11 (8)	15.31 (6)	3.19 (9)	10.52 (9)	12.55 (7)	3.00 (9)
8	EPIC_2SCNN_RGB	2	09/06/18		33.12 (10)	17.58 (8)	6.79 (8)	73.23 (8)	40.46 (7)	20.42 (7)	16.06 (9)	11.97 (8)	3.39 (8)	9.44 (10)	12.53 (8)	3.01 (8)
9	EPIC_TSN_Flow	1	09/04/18		40.08 (4)	14.51 (10)	6.73 (9)	73.40 (7)	33.77 (9)	18.64 (9)	19.98 (6)	9.48 (9)	2.32 (10)	13.81 (5)	8.58 (11)	2.54 (10)
10	zhun	4	01/09/19		28.71 (11)	13.25 (11)	5.16 (10)	67.70 (11)	29.67 (11)	12.94 (11)	10.61 (11)	8.22 (11)	3.59 (7)	7.62 (11)	9.47 (10)	3.29 (7)
11	EPIC_2SCNN_Flow	1	09/06/18		37.28 (7)	11.85 (12)	4.75 (11)	71.56 (10)	28.41 (12)	14.82 (10)	14.93 (10)	2.93 (12)	1.17 (12)	11.60 (7)	6.26 (12)	2.05 (11)
12	szhang_	3	03/11/19		17.86 (12)	14.89 (9)	2.80 (12)	64.12 (12)	31.21 (10)	8.71 (12)	5.16 (12)	8.76 (10)	1.50 (11)	4.31 (12)	9.54 (9)	1.55 (12)
13	chinmayd	1	03/22/19		0.89 (13)	0.31 (13)	0.03 (13)	4.17 (13)	1.23 (13)	0.03 (13)	3.79 (13)	1.37 (13)	0.38 (13)	1.88 (13)	0.43 (13)	0.02 (13)

Figure 3.8: *Our submission on the 2019 EPIC-Kitchens - Action recognition challenge for S1 (top) and S2 (bottom) test sets. At the time of submission, the single model TBN outperformed all previous methods on all metrics by a significant margin on both S1 and S2, and the TBN Ensemble further increased the gap.*

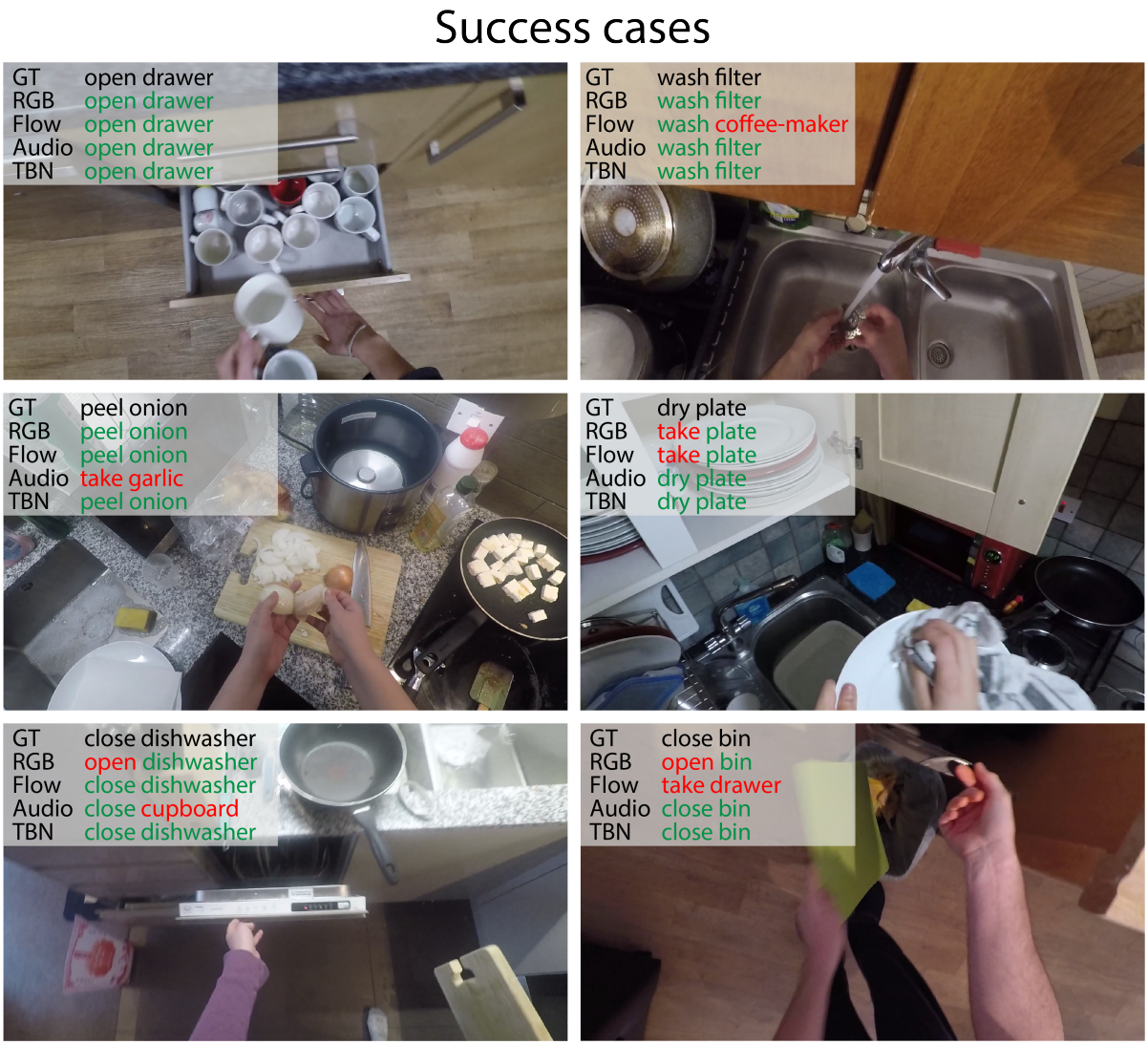


Figure 3.9: *Qualitative results on a held-out validation set, from the training videos. The ground truth and the predictions of RGB, Flow and Audio are compared with TBN (Single Model). We show success cases where TBN predicts both the verb and the noun correctly.*

3.4 Qualitative Results

In Fig. 3.9-3.11, we show selected qualitative results on a held-out validation set, from the publicly available training videos. We hold-out 14 (untrimmed) videos from the training set, for qualitative examples. Results are shown using a TBN trained on the rest of the training set. For each example, we show the ground truth, and the predictions of individual modalities (RGB, Flow, Audio) compared with our TBN (Single Model). In Fig. 3.9 and Fig. 3.10, we show success cases were TBN predicts both the verb and the noun correctly. One interesting example is 'close bin', where only audio amongst individual modalities predicts the action correctly, because of the characteristic sound of the bin's lid when it closes. In the 'soak

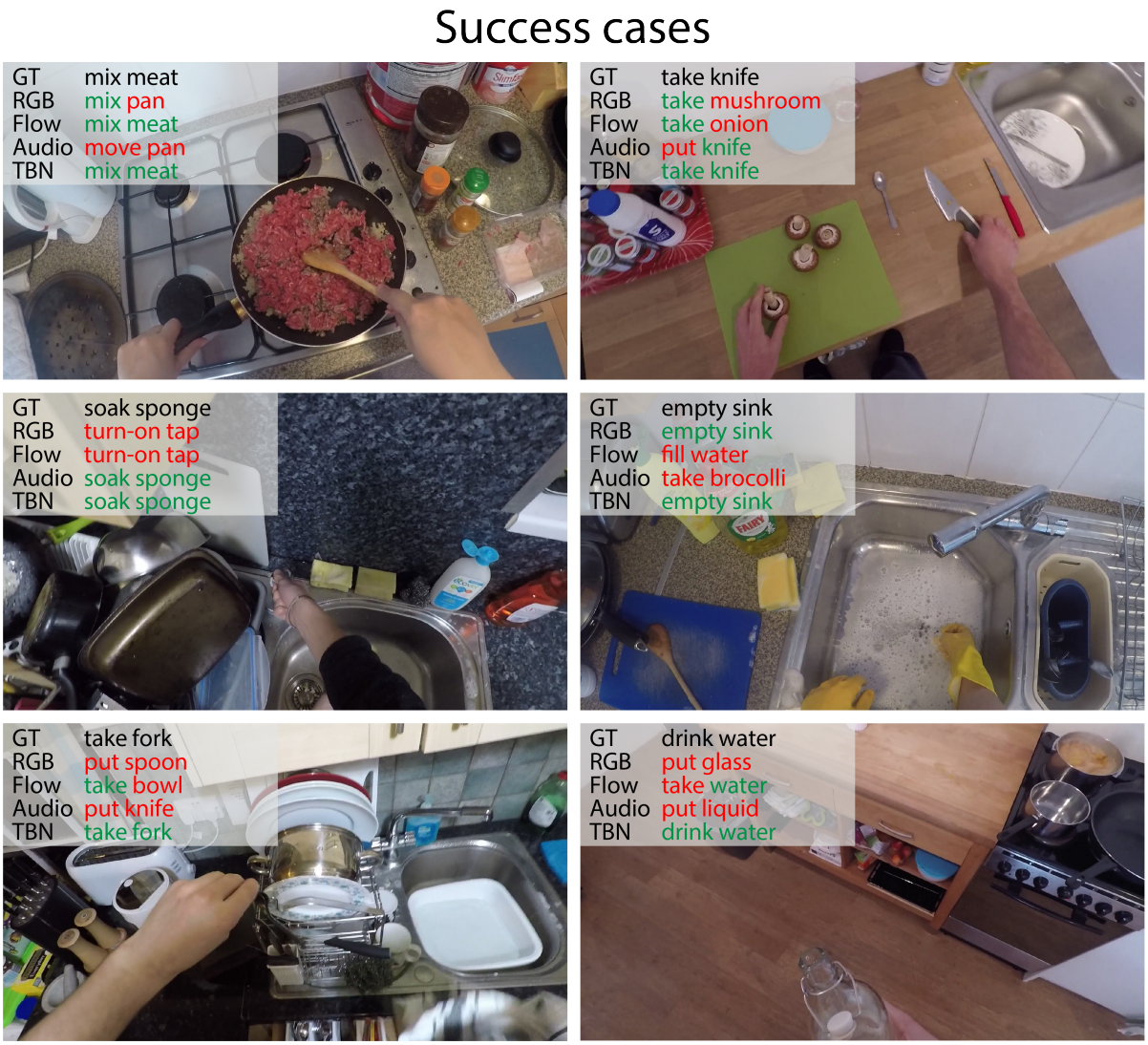


Figure 3.10: *More success cases.*

sponge' example, where the sponge and the hand that holds it are occluded by the other hand, comparing to single modalities only audio can predict the action correctly from the distinctive sound of the water when the sponge is squeezed. On the contrary, in 'peel onion', audio confuses the action with 'take garlic' as both actions generate similar sounds. These two examples showcase that multiple modalities help to resolve ambiguities arising from one of the modalities, where in the first case there is visual occlusion and audio can predict the action correctly, while in the second case the visual modalities can infer the action where audio is not discriminative. Another notable example is 'dry plate', where in addition to TBN, only audio predicts the verb correctly, demonstrating the clear and high amplitude audio in egocentric videos due to the proximity of the wearable camera to the action. Finally, there are examples, such as 'drink water', where although all modalities provide wrong predictions, TBN can still predict the action correctly, showcasing the potential of multi-modal fusion. In

Figure 3.11: *Failure cases where TBN predicts the verb and the noun, or only the noun, incorrectly.*

Fig. 3.11, there are failure cases where TBN mispredicts the action. In 'put bowl' and 'fill kettle' only audio makes correct prediction, whereas in 'take glass' and 'take bottle' RGB is sufficient, while TBN cannot recover from errors made from the other modalities. For a better understanding of the qualitative results, please watch the video with audio at https://www.youtube.com/watch?v=VzoaKsDvv1o.

3.5 Conclusion

We have shown that the TBN architecture is able to flexibly combine the RGB, Flow and Audio modalities with sparse temporal sampling and mid-level fusion within TBWs. TBN achieves an across the board performance improvement, compared to individual modalities. In particular, we have demonstrated how audio is complementary to appearance and motion for a number of classes; audio is more informative for verbs than nouns, while being discriminative for nouns when the associated objects make distinct sounds or for certain materials that sound upon interaction. The complementarity of audio to appearance and motion was also conveyed by comparing an audio-visual TBN to a visual-only TBN, as when audio was incorporated performance improved and class confidence increased, while confusion dropped. Moreover, we have illustrated the preeminence of appearance for noun classes, whereas for verbs, motion

is dominating. The performance of TBN significantly exceeds TSN trained on the same data showcasing the capabilities of mid-level learnable fusion compared to late fusion. TBN also provided state-of-the-art results on the public 2019 EPIC-Kitchens leaderboard. Another important find is the potential usefulness of multi-modal fusion for tackling the class imbalance problem; TBN significantly improved the performance on tail classes compared to individual modalities. Last but not least, we have illustrated that asynchronous fusion can be advantageous, over the more common synchronous approaches, in two ways. First, by showing that a long TBW provides optimal recognition accuracy and second by showing that an ensemble of TBNs trained with different TBW widths improves the performance.

In this work, we have shown the usefulness of multi-modal temporal binding in egocentric action recognition, but its applicability is not limited in this context. It could be adopted in any problem where multiple modalities with different sampling rates need to be fused, and where possibly the semantics in each modality progress at different speeds. Binding could also go beyond time, and extend to other dimensions as well. Some examples include: a) applications using video and text where text is particularly asynchronous to video and binding could be used to combine different parts of a video with different parts of a sentence, b) integration of multiple sensor measurements for robotic applications and c) fusion of multi-modal medical data such as medical images and electronic health records where a binding network could learn to associate parts of an image with subsets of health records of a patient.

The work presented in this chapter sheds light on challenges of binding vision with audio using neural networks, but also provides insights of how to address them. A limitation of our approach is that temporal aggregation of multiple TBWs diminishes the effect of the TBW width, therefore designing temporal aggregation techniques sensitive to the width of TBW would be an interesting exploration. A challenge not only for our approach, but for multi-modal problems in general, is the design of the optimal fusion function. We have shown that the simplest approach, concatenation, outperformed more sophisticated methods, where the performance decreased with increase in parameters. Thus, a fusion method should either be lightweight and subtle at the same time, or be (pre) trained in large-scale datasets when more complex. Further avenues for exploration include a model that learns to adjust TBWs over time, while also being able to learn the modalities' temporal offsets within the window. This could be enhanced by implementing class-specific temporal binding windows, since different actions progress at different speeds. Spatio-temporal binding would be an intriguing step forward as a way to perform visual grounding using binding windows. Finally, given the successful application of Transformers in many vision tasks, one should not disregard their feasibility to perform multi-modal attention within and across TBWs.

CHAPTER

FOUR

AUDITORY SLOW-FAST NETWORKS AND MULTI-STREAM AUDIO-VISUAL FUSION

Recognising objects, interactions and activities from audio is distinct from prior efforts for scene audio recognition, due to the need for recognising sound-emitting objects (*e.g.* alarm clock, coffee-machine), sounds generated from interactions with objects (*e.g.* put down a glass, close drawer), and activities (*e.g.* wash, fry). This introduces challenges related to variable-length audio associated with these activities, the primary motivation of our work. Some can be momentary (*e.g.* close) while others span over a longer period (*e.g.* fry). Moreover, other challenges include that many sounds exhibit intra-class variations (*e.g.* cut onion vs cut cheese) as well as that background or irrelevant sounds are often captured with these activities (also discussed in the previous chapter). Our first goal in this chapter is to perform activity recognition solely from the audio signal associated with videos from activity-based datasets, where we focus on VGG-Sound [30] and EPIC-KITCHENS [36], captured from YouTube and egocentric videos respectively.

In these datasets, an example of sounds that span over longer timescales are *harmonic* sounds whereas an example of shorter sounds are *percussive* sounds. These are demonstrated in Fig. 4.1. Harmonic sounds are defined as pitched sounds spanning in time, such as "rinse bell pepper" and "canary calling". The prototype of a harmonic sound is the acoustic realisation of a sinusoid [131]. Percussive sounds are defined as momentary or temporally repetitive sounds, *e.g.* "chop garlic cloves" and "typing on typewriter". The prototype of a percussive sound is the acoustic realisation of an impulse [131].

Therefore, we propose to use two-stream architectures, able to capture long-term sounds in

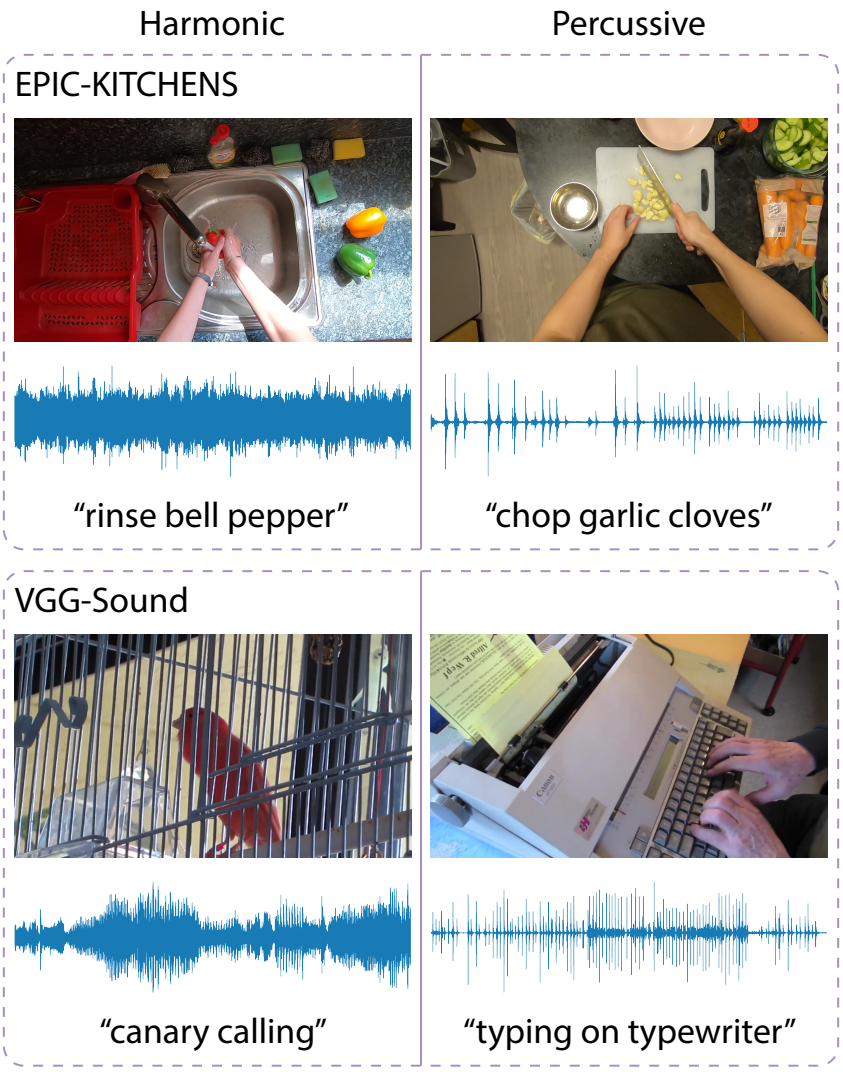


Figure 4.1: *Harmonic (left) and percussive (right) sounds on EPIC-KITCHENS (top) and VGG-Sound (bottom). Harmonic sounds are pitched sounds spanning in time, whereas percussive sounds are momentary or temporally repetitive sounds.*

one stream and short-term sounds in the other. We draw further inspiration from neuroscience, where there is strong evidence for the existence of two streams in the human auditory system, the ventral stream for identifying sound-emitting objects and the dorsal streams for locating these objects. Studies [158, 219] suggest the ventral stream accordingly exhibits high spectral resolution for object identification, while the dorsal stream has a high temporal resolution and operates at a higher sampling rate.

Using this evidence as the driving force for designing our architecture, and inspired by a similar vision-based architecture [47], we propose two streams for auditory recognition: a Slow and a Fast stream, that realise some of the properties of the ventral and dorsal auditory pathways respectively. Our streams are variants of residual networks and use 2D separable convolu-

tions that operate on frequency and time independently. The streams are fused in multiple representation levels with lateral connections from the Fast to the Slow stream, and the final representation is obtained by concatenating the global average pooled representations for action recognition.

Our second goal in this chapter is to fuse our proposed auditory Slow-Fast architecture with its visual analogue [47]. Visual Slow-Fast aims to learn appearance and motion in the two streams. We are motivated by the observation that appearance in vision and long-term sounds in audio are both characterised by slow temporal changes, while motion in vision and short-term sounds in audio are characterised by faster temporal changes, and propose the fusion of corresponding streams across modalities, *i.e.* visual Slow with auditory Slow and visual Fast with auditory Fast, with the aim to associate appearance with long-term sounds and motion with short-term sounds. To this end, we propose fusing the streams at multiple representation levels with cross-modal attention fusion that incorporates residual connections.

We also show that such architectures are difficult to train, where evidence points to two main causes of overfitting: 1) the discrepancy in training dynamics between vision and audio where audio overfits faster [196, 207], and 2) co-adaptations between modalities representations [135], where complex multi-modal relationships are learnt as a modality learns to correct the errors of other modalities, resulting in modalities intolerant to the noise and to the absence of other modalities. This is more likely to happen when one modality is weaker, as the weaker modality makes more errors which lead to more co-adaptations. In our case the weaker modality is audio, where by leveraging some of our observations from Ch. 3, we note that noise in the auditory data stream of egocentric actions originates from background sounds irrelevant to the action or from actions with overlapping audio. Moreover, similar to the complete absence of a modality in [135], there are silent egocentric actions, such as 'open cupboard' and 'spoon sugar'. The visual modality can falsely co-adapt to noisy audio and silent actions, affecting negatively the performance of the model. Motivated by these observations, our goal is twofold: 1) slow down the learning pace of the audio network to match that of its visual counterpart, 2) alleviate audio-visual co-adaptations that are caused by noisy audio/silent actions, to learn robust visual representations that are not affected by noise in the audio stream. To address these, we explore different variants of regularisation by dropping the audio modality during training, which we term as DropAudio.

The contributions of the first part of this chapter are the following: i) we propose a novel two-stream architecture for auditory recognition that respects evidence in neuroscience; ii) we achieve state-of-the-art results on both EPIC-KITCHENS and VGG-Sound; iii) we show the importance of fusing our specialised streams through an ablation analysis; and finally iv) we show that the Slow stream learns harmonic sounds while the Fast stream focuses on percussive

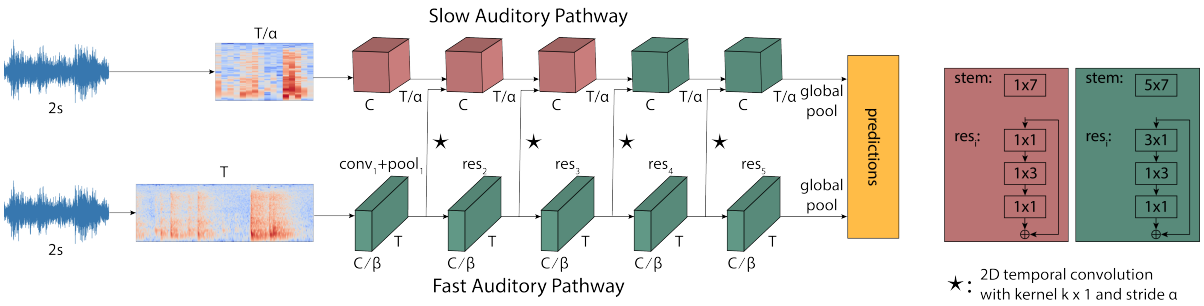


Figure 4.2: *Proposed Auditory Slow-Fast (ASF) architecture. The input to the Slow pathway is strided by α, whereas Fast takes as input the whole spectrogram. Slow has increased number of channels with the first layers performing convolutions only over the frequency dimension (brown blocks). The Fast pathway has less channels, by β, with convolutions over both frequency and time over the whole architecture (green blocks). The architecture employs separable convolutions over frequency and time, for both types of residual blocks (brown vs green), as shown on the right. Streams are fused with multi-level lateral connections from the Fast to the Slow stream, with strided temporal convolution on Fast and concatenation on Slow. Final representations, fed to the classifier, are globally average pooled and concatenated.*

sounds through class analysis and feature map visualisations.

The second part of this chapter offers the following contributions: i) a novel four-stream architecture for audio-visual action recognition with visual and auditory Slow and Fast streams which aims to learn corresponding concepts across modalities; ii) we achieve state-of-the-art results on EPIC-KITCHENS; iii) an investigation of fusion techniques at different levels of the architecture, namely late fusion, mid-level fusion and multi-level fusion showcasing the benefits of the proposed multi-level cross-attention fusion with residual connections; iv) an ablation analysis demonstrating the necessity of audio-visual regularisation through dropping audio, where we propose DropAudio building on existing techniques [207].[1]

4.1 ASF: Auditory Slow-Fast Network

Next, we describe the design principles of our architecture, depicted in Fig. 4.2, namely Auditory Slow-Fast Network (ASF). The Slow stream operates on a low sampling rate with high channel capacity to capture frequency semantics, while the Fast stream operates on a high sampling rate with more temporal convolutions and less channels to capture temporal patterns.

[1]Our pre-trained models and python code for auditory Slow-Fast are available at https://github.com/ekazakos/auditory-slow-fast. In the future, we will also release the code for the proposed audio-visual Slow-Fast.

stage	*Slow* pathway	*Fast* pathway	output sizes $T \times F$
spectrogram	-	-	400×128
data layer	stride 4, 1	stride 1, 1	*Slow* : 100×128 *Fast* : 400×128
conv_1	1×7, 64 stride 2, 2	5×7, 8 stride 2, 2	*Slow* : 50×64 *Fast* : 200×64
pool_1	3×3 max stride 2, 2	3×3 max stride 2, 2	*Slow* : 25×32 *Fast* : 100×32
res_2	$\begin{bmatrix} 1 \times 1, 64 \\ 1 \times 3, 64 \\ 1 \times 1, 256 \end{bmatrix} \times 3$	$\begin{bmatrix} 3 \times 1, 8 \\ 1 \times 3, 8 \\ 1 \times 1, 32 \end{bmatrix} \times 3$	*Slow* : 25×32 *Fast* : 100×32
res_3	$\begin{bmatrix} 1 \times 1, 128 \\ 1 \times 3, 128 \\ 1 \times 1, 512 \end{bmatrix} \times 4$	$\begin{bmatrix} 3 \times 1, 16 \\ 1 \times 3, 16 \\ 1 \times 1, 64 \end{bmatrix} \times 4$	*Slow* : 25×16 *Fast* : 100×16
res_4	$\begin{bmatrix} 3 \times 1, 256 \\ 1 \times 3, 256 \\ 1 \times 1, 1024 \end{bmatrix} \times 6$	$\begin{bmatrix} 3 \times 1, 32 \\ 1 \times 3, 32 \\ 1 \times 1, 128 \end{bmatrix} \times 6$	*Slow* : 25×8 *Fast* : 100×8
res_5	$\begin{bmatrix} 3 \times 1, 512 \\ 1 \times 3, 512 \\ 1 \times 1, 2048 \end{bmatrix} \times 3$	$\begin{bmatrix} 3 \times 1, 64 \\ 1 \times 3, 64 \\ 1 \times 1, 256 \end{bmatrix} \times 3$	*Slow* : 25×4 *Fast* : 100×4
global average pool, concatenate, fc			# classes

Table 4.1: *Architecture details of ASF (Fig. 4.2). Temporal downsampling is performed at conv_1 and pool_1 layers using a stride $= 2$. Frequency downsampling with a stride $= 2$ is performed at conv_1 and pool_1, as well as at the middle convolutional layer of the first residual block of each residual stage except res_2, following [47]. pool_1 performs max-pooling with a kernel of 3×3. In convolutional layers (conv_1 and res_i), first the kernel sizes are shown followed by the number of channels. In residual stages, the numbers after the brackets denote the number of residual blocks. Colors of kernels correspond to the different types of blocks from Fig. 4.2. The output sizes of the feature maps of each layer are also shown on the rightmost column.*

Input. Both streams operate on the same audio length, from which a log-mel-spectrogram is extracted. The Fast stream takes as input the whole log-mel-spectrogram without any striding, while the Slow stream uses a temporal stride of α on the input log-mel-spectrogram, where $a \geq 1$. Hence, for T frames in the Fast stream, the temporal input dimension to the Slow stream is T/α.

Slow and Fast streams. The two streams are variants of ResNet50 [71]. Each stream is comprised of an initial convolutional block with a pooling layer followed by 4 residual stages, where each stage contains multiple residual blocks. The two streams differ in their ability to capture frequency semantics and temporal patterns. The details of each stream including the number and type of blocks per stage, numbers of channels and output sizes can be seen in Table 4.1.

The Slow stream has a high channel capacity, with β times more channels than the Fast stream, while operating on a low sampling rate. As the input spectrogram is strided temporally by α, the intermediate feature maps have a lower temporal resolution. Moreover, the Slow stream has temporal convolutions only in res_4 and res_5, as shown in Fig. 4.2 right, where the brown residual block (left) utilises only frequency convolution whereas the green block (right) employs both frequency and temporal convolutions (Table 4.1 uses the same colours as Fig. 4.2 to denote frequency-only and frequency-temporal convolutional blocks). By restricting the temporal resolution and the temporal kernels of the Slow stream while keeping a high channel capacity, this stream can focus on learning frequency semantics, as its architecture is composed of high-capacity frequency-only convolutions up to intermediate layers (conv_1-res_3).

The Fast stream on the other hand uses no temporal striding in the input. Therefore, the intermediate feature maps have a higher temporal resolution, with temporal convolutions throughout the stream. With a high temporal resolution and more temporal kernels while having fewer channels, the design of the Fast stream facilitates the learning of temporal patterns while maintaining a reasonable computational budget and parameter increase.

Separable convolutions. We use separable convolutions in frequency and time as can be seen in the green block in Fig. 4.2 right. We break a 3×3 kernel in two kernels, 3×1 followed by 1×3. Separable convolutions have proven useful for video recognition [187] and have recently been used in audio as well [207]. Motivated by the necessity of non-symmetric filtering over the frequency and temporal dimensions of spectrograms (see Subsec. 2.2.2), we utilise separable convolutions to separately attend to time and frequency of the input signal. We contrast them to two-dimensional filters that convolve across both frequency and time.

Multi-level fusion. Following the approach in [47], we fuse the information from the Fast to the Slow stream with lateral connections, at multiple levels. We first apply a 2D temporal convolution with a kernel 7×1 and a stride of α to the output of the Fast stream to match the Slow stream sampling rate, and then we concatenate the downsampled feature map with the Slow stream feature map. Fusion is applied after pool_1 and each residual stage except res_5.

Output representation. The final representation fed to the classifier is obtained by applying time-frequency global average pooling after the last convolutional layer of both Slow and Fast streams and concantenating the pooled representations.

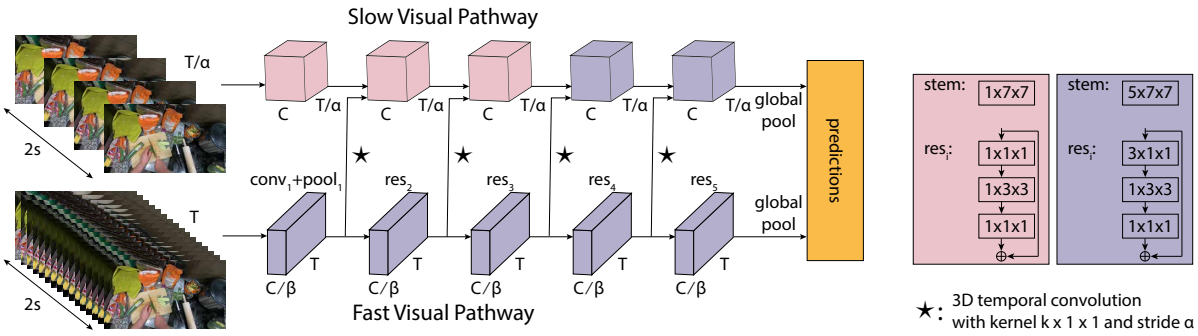


Figure 4.3: *The visual Slow-Fast architecture [47]. Differences to auditory Slow-Fast: Visual Slow-Fast operates on space-time, therefore the inputs are 3D rather than 2D. Accordingly, 3D separable convolutions are incorporated over space and time in the residual blocks (pink and purple blocks). Furthermore, the lateral connections employ 3D strided temporal convolutions. The first layers of Slow perform convolutions only over space (as opposed to frequencies in auditory Slow-Fast), whereas Fast has convolutions over both space and time over the whole architecture.*

4.2 AVSF^2: Audio-Visual Slow-Fast Stream Fusion

After having described our proposed ASF architecture for audio recognition, in this section we detail our approach for integrating visual and auditory Slow and Fast streams, which we dub Audio-Visual Slow-Fast Stream Fusion (AVSF^2). First, we demonstrate the visual Slow-Fast network [47] and discuss its differences to our proposed auditory Slow-Fast architecture. Then, we describe various audio-visual multi-stream fusion techniques we investigate. Finally, we propose different audio-visual regularisation schemes.

4.2.1 Visual vs auditory Slow-Fast networks

The visual Slow-Fast network [47] is shown in Fig. 4.3 and the details of its architecture can be seen in Table 4.2. Our two-stream auditory architecture is inspired by its visual counterpart [47], however, key differences are introduced: Our input is 2D rather than 3D, as we operate on time-frequency while the visual Slow-Fast operates on time-space. Hence, we use 2D separable convolutions decomposed as 3×1 and 1×3 filters, whereas [47] uses 3D separable convolutions decomposed as $3 \times 1 \times 1$ and $1 \times 3 \times 3$ filters. Similarly, we use 2D strided temporal convolutions in the lateral connections, while [47] uses 3D strided temporal convolutions. Additionally, the sampling rate for audio is naturally significantly higher than that of video, e.g. 24kHz vs 50fps in EPIC-KITCHENS-100, and the dimensionality in video is significantly higher. Accordingly, the approach in [47] only considers a few temporal samples (16 and 64 frames in the Slow and Fast streams respectively). In contrast, our audio spectrogram (see Subsec. 4.3.2) contains 100 and 400 temporal dimensions in the Slow and Fast streams,

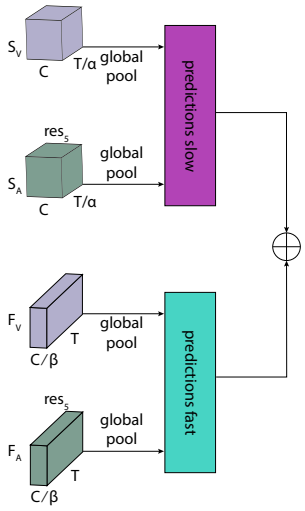
stage	*Slow* pathway	*Fast* pathway	output sizes $T \times S^2$
raw clip	-	-	128×224^2
data layer	stride 8, 1^2	stride 2, 1^2	*Slow* : 16×224^2 *Fast* : 64×224^2
conv_1	1×7^2, 64 stride 1, 2^2	5×7^2, 8 stride 1, 2^2	*Slow* : 16×112^2 *Fast* : 64×112^2
pool_1	1×3^2 max stride 1, 2^2	1×3^2 max stride 1, 2^2	*Slow* : 16×56^2 *Fast* : 64×56^2
res_2	$\begin{bmatrix} 1 \times 1^2, 64 \\ 1 \times 3^2, 64 \\ 1 \times 1^2, 256 \end{bmatrix} \times 3$	$\begin{bmatrix} 3 \times 1^2, 8 \\ 1 \times 3^2, 8 \\ 1 \times 1^2, 32 \end{bmatrix} \times 3$	*Slow* : 16×56^2 *Fast* : 64×56^2
res_3	$\begin{bmatrix} 1 \times 1^2, 128 \\ 1 \times 3^2, 128 \\ 1 \times 1^2, 512 \end{bmatrix} \times 4$	$\begin{bmatrix} 3 \times 1^2, 16 \\ 1 \times 3^2, 16 \\ 1 \times 1^2, 64 \end{bmatrix} \times 4$	*Slow* : 16×28^2 *Fast* : 64×28^2
res_4	$\begin{bmatrix} 3 \times 1^2, 256 \\ 1 \times 3^2, 256 \\ 1 \times 1^2, 1024 \end{bmatrix} \times 6$	$\begin{bmatrix} 3 \times 1^2, 32 \\ 1 \times 3^2, 32 \\ 1 \times 1^2, 128 \end{bmatrix} \times 6$	*Slow* : 16×14^2 *Fast* : 64×14^2
res_5	$\begin{bmatrix} 3 \times 1^2, 512 \\ 1 \times 3^2, 512 \\ 1 \times 1^2, 2048 \end{bmatrix} \times 3$	$\begin{bmatrix} 3 \times 1^2, 64 \\ 1 \times 3^2, 64 \\ 1 \times 1^2, 256 \end{bmatrix} \times 3$	*Slow* : 16×7^2 *Fast* : 64×7^2
global average pool, concatenate, fc			# classes

Table 4.2: *Architecture details of visual Slow-Fast [47] (Fig. 4.3). Temporal input dimension (16 in Slow and 64 in Fast) is smaller than the one used in auditory Slow-Fast (100 in Slow and 400 in Fast), due to the large difference in video vs audio sampling rates. Differently than auditory Slow-Fast, no temporal downsampling is performed. conv_1, pool_1 layers, and the middle convolutional layer of the first residual block of each residual stage except res_2 perform spatial downsampling with a stride $= 2$. Colors of kernels correspond to the different types of blocks from Fig. 4.3.*

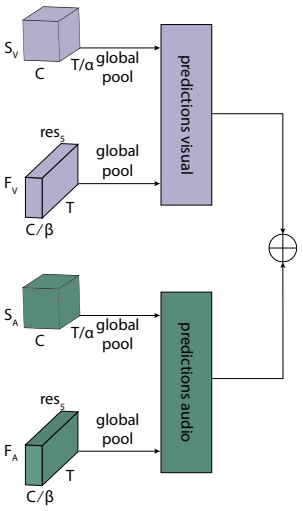
respectively. To compensate for the high sampling rate of audio, we temporally downsample the representations of both streams by a factor of 4, using a temporal stride $= 2$ in conv_1 and pool_1 of both streams, whereas [47] does not perform any temporal downsampling, retaining the initial temporal dimension size (as shown in Table 4.2 right). The remaining stages of our proposed auditory architecture do not perform any temporal downsampling.

4.2.2 Multi-stream fusion architectures

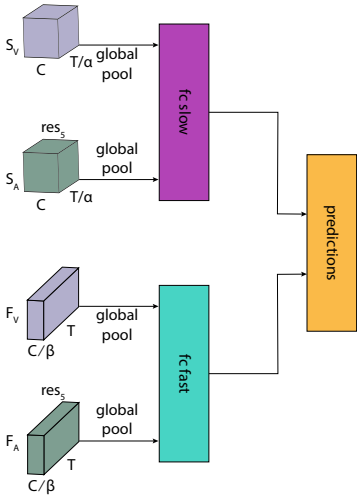
We explore techniques of integrating visual and auditory Slow and Fast streams. Our goal is to fuse and train all four streams simultaneously, end to end. The innovation of our proposal lies in that there are same types of streams across modalities, *i.e.* audio-visual Slow and audio-visual Fast. We leverage this to fuse visual and auditory Slow streams and learn correspondences between the appearance of an entity and its spectral auditory properties; *e.g.* associate the presence of an object such as a blender with the frequencies at which it sounds.



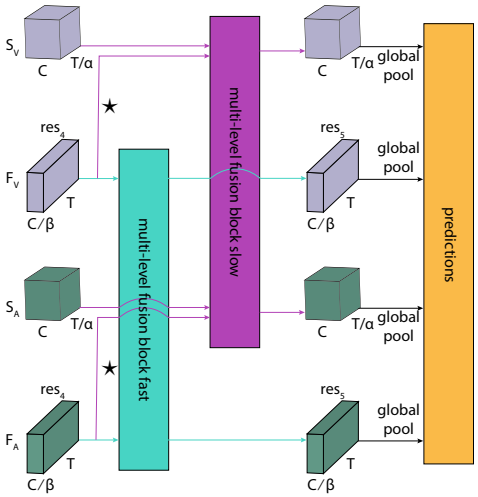
(a) *Late fusion (SS-FF)*



(b) *Late fusion (SF-SF)*



(c) *Mid-level fusion*



(d) *Multi-level fusion*

Figure 4.4: *Techniques for fusing visual and auditory Slow and Fast streams. In late fusion ((a) and (b)), streams are fused at the prediction layer by averaging ($\bigoplus$) their predictions. We propose combining same types of streams across modalities, shown on (a) for late fusion (SS-FF: Slow$_V$-Slow$_A$ & Fast$_V$-Fast$_A$), which is contrasted to combining different types of streams within modalities, shown on (b) (SF-SF: Slow$_V$-Fast$_V$ & Slow$_A$-Fast$_A$). Mid-level fusion (c) integrates visual and auditory features at the last convolutional layer of each stream (output of res$_5$), using two fully-connected layers for Slow and Fast streams, respectively, which are then concatenated and fed to the prediction layer. Multi-level fusion (d) is employed at multiple intermediate layers of the network with a Slow and a Fast fusion block, producing visual and auditory outputs of same size as their inputs. At the last convolutional layer, there is no multi-level fusion block; the globally pooled outputs of all four streams are concatenated and fed to the prediction layer. Note that the Slow multi-level fusion block takes as input the concatenation of Slow with the lateral connection from the Fast stream. The different multi-level fusion blocks that we consider are shown in Fig. 4.5.*

We also fuse the Fast streams of the two modalities to correlate the motion with temporal auditory properties; *e.g.* the movements of the hand when chopping an onion with the percussive sound of the knife knocking the chopping board. To this end, we explore late, mid-level, and multi-level fusion techniques, which are shown in Fig. 4.4 and detailed below.

Late fusion. In this scheme, fusion takes place at the prediction level. Before the prediction layers, the visual and auditory Slow-Fast architectures remain unchanged. We combine the same type of streams across modalities. We concatenate the globally pooled representations of the Slow streams and feed them in a prediction layer, and similarly we feed the concatenated representations of the Fast streams to a different prediction layer. Finally, the audio-visual Slow and Fast streams' predictions are averaged. These serve as the final predictions and their error is backpropagated. As shown in Fig. 4.4 top, we contrast our approach (Fig. 4.4a) to combining same modalities across streams (Fig. 4.4b).

Mid-level fusion. Here, fusion takes place in between the last convolutional layer and the prediction layer. Similarly to the mid-level fusion in TBN (Ch. 3), it is realised with fully-connected layers on top of the concatenation of the globally pooled representations. A fully-connected layer takes as input the concatenated Slow representations and projects them to a lower dimension, producing an audio-visual Slow representation, and likewise another fully-connected layer operating on the Fast representations produces an audio-visual Fast representation. The final Slow and Fast features are concatenated and fed to a prediction layer.

Multi-level fusion. We explore audio-visual fusion at multiple representation levels. Although in Fig. 4.4 we demonstrate multi-level fusion only at the output of res_4, we apply it at the output of each residual stage, where also Slow-Fast fusion within modalities with lateral connections takes place. The last residual stage (res_5) is excluded and there is no multi-level fusion block, akin to the uni-modal Slow-Fast architectures where there are no lateral connections; at res_5 the outputs of all four streams are simply concatenated and fed to a prediction layer. We fuse Slow streams across modalities after the concatenation of lateral connections from Fast to Slow within modalities. In preliminary experiments, we also tried fusing Slow streams before lateral connections and observed a drop in performance. Fast streams fusion takes as input solely the Fast output representations from the previous layer.

There are a few architectural constraints that need to be considered for designing a multi-level fusion function. To leverage pretrained networks within each modality, the original uni-modal architectures should not be modified. That means that the fusion block should output *two* representations of the *same* shapes as its inputs to be passed to the next layer of each modality (late and mid-level fusion produce a single output). Therefore, it should be able to associate tensors of different shapes. For example, video tensors are 4D whereas audio 3D; these need

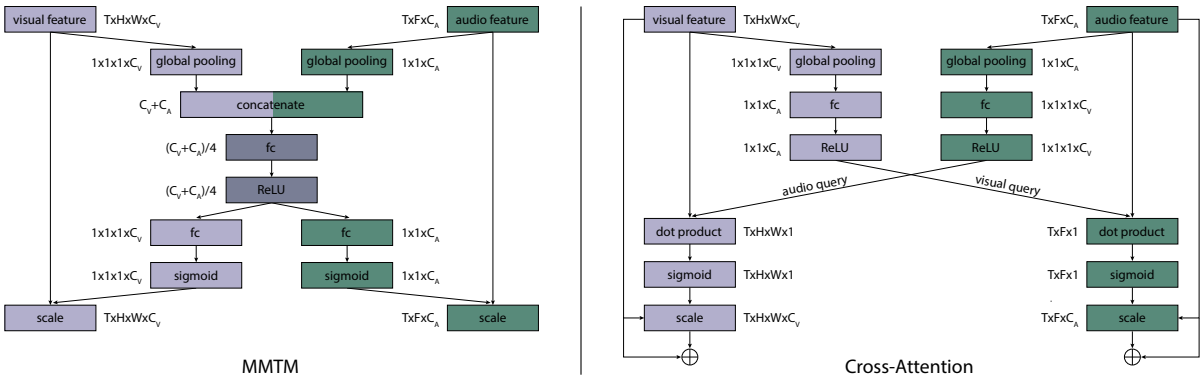


Figure 4.5: *Different multi-level fusion blocks. In MMTM fusion [88] (left), globally pooled visual and auditory representations are concatenated and passed to a non-linear layer producing an audio-visual feature. A multi-modal gating mechanism then, takes as input the audio-visual feature and uses two fully-connected layers, one for each modality, followed by a sigmoid activation, which are multiplied with the input representations of each modality channel-wise. In Cross-Attention fusion (right), the pooled representations of each modality are passed to two non-linear layers, one per modality, producing a visual query and an audio query. The dot-product of the query from one modality with the input feature from the other modality is calculated, followed by a sigmoid activation to form visual and auditory cross-attention masks. These are then multiplied position-wise with the input features of each modality. Cross-Attention employs residual connections ($\bigoplus$) from the input feature of each modality to the output of the fusion block for that modality.*

to be fused and the outputs should also be 4D and 3D audio-visual tensors, respectively. These constraints are also partially discussed in [88].

We investigate two fusion techniques that respect these constraints: i) the Multimodal Transfer Module (MMTM) [88], and ii) an approach that we term Cross-Attention fusion. Similar techniques to Cross-Attention fusion have been used for unsupervised audio-visual source localisation [3, 7, 160], while here we investigate its viability for supervised action recognition. Both types of multi-level fusion blocks are shown in Fig. 4.5.

MMTM [88] proposed multi-modal Squeeze-And-Excitation blocks [79]. For a more detailed description of [88], please see Subsec. 2.1.4. MMTM injects multi-modal knowledge in the feature maps of each modality using a multi-modal gating mechanism. In Fig. 4.5 (left), we illustrate an MMTM block for fusing our visual and auditory streams. First, global average pooling aggregates the feature maps of both modalities into visual and auditory channel descriptors. These are then concatenated and passed to a non-linear fully-connected layer to model dependencies between audio-visual channels by projecting them to a lower dimension. Finally, the gating mechanism is formed with two fully-connected layers (projecting back to

the original modalities dimensions), one for each modality, followed by sigmoid activation, which rescale the visual and auditory feature maps channel-wise, enhancing them with multi-modal information.

The motivation of our Cross-Attention block (Fig. 4.5, right) is to attend to spatio-temporal visual areas corresponding to sounding objects/actions and to frequency-temporal auditory locations corresponding to visually salient objects/actions. To this end, first a global average pooling layer is applied to both modalities followed by a non-linear fully-connected layer per modality. The fully-connected layers produce a learnt feature vector per modality that is used to query the feature map of the other modality about the presence of the learnt concept at each location of the feature map. This is achieved by calculating the dot-product between the audio query and each spatio-temporal location of the input visual feature map followed by a sigmoid activation function, producing the visual attention mask. Similarly, the auditory attention mask is produced from the dot-product of the visual query and each frequency-temporal location of the input auditory feature map, followed by a sigmoid. The attention masks are then multiplied position-wise with the input feature maps, producing cross-modally attended visual and auditory features.

A disadvantage of Cross-Attention is that, by focusing on regions that are relevant to the query from the other modality, it may disregard discriminant within-modal features. For example, for the action 'cut cheese' it may produce a visual feature map with strong activations in the area that corresponds to the hand and the knife as these are responsible for the generated sound, while suppressing the feature map locations that correspond to the cheese. To mitigate this, we employ residual connections from the input feature of each modality to the output of the fusion block for that modality. This can address the within-modal information loss as the identity mappings provide access to the original feature map before attention.

A fundamental difference between Cross-Attention and MMTM is that while MMTM performs *channel-wise attention* by sharing each channel's attention weight across all positions within that channel, Cross-Attention performs *position-wise attention* by sharing each position's attention weight across all channels. Furthermore, a key difference between our Cross-Attention block and the attention modules in [3, 7, 160] is that we employ both visual and auditory cross-attention masks as we want to leverage information from both modalities for action recognition, while [3, 7, 160] utilise only visual cross-attention masks as the problem they are targeting is to localise the sound source in images/videos.

4.2.3 Audio-visual regularisation

In this subsection, we investigate techniques to regularise the training of audio-visual networks. The regularisation methods we explore are inspired by [135] and [207], where modalities are randomly dropped in training. Following our motivation, our goal is to tackle overfitting from two different perspectives: i) adapt the learning pace of the audio streams to that of their visual analogues, addressing the discrepancy in training dynamics of the two modalities [196, 207], and ii) hinder false co-adaptations between modalities representations [135, 207], *i.e*. learn robust visual representations that do not rely on the auditory features, mitigating the effect of silent actions or noisy audio. In order to fulfil these objectives, we focus our attention on different schemes of randomly dropping audio, and exclude dropping out video from our study (which was considered in [135]). Both Slow and Fast auditory streams are either dropped or retained simultaneously, and we do not experiment with dropping one while preserving the other. An overview of the schemes that we explore is shown in Fig. 4.6, while they are described next in detail.

DropAudio-Pathways. Different variants of dropping modalities have been proposed in [135] and [207], where the pathway to be dropped is masked with 0 at different layers of its architecture, depending on the strategy. First, we explain why ModDrop [135] is inadequate to fulfil our goals. Then, we analyse why DropPathway [207] can *only partially* tackle both sources of overfitting that we are interested in, as it is not carefully crafted. Finally, we propose a simple but practically important modification to DropPathway to rectify its shortcomings.

ModDrop is not designed to drop only audio in audio-visual networks, but to drop any modality in multi-modal networks. Despite that it would be easy to modify it to drop only audio, it would still probably not boost the performance of the model. ModDrop's primary motivation is to learn robust modalities representations with the aim to handle missing modalities in inference. It achieves that by randomly dropping modalities *inputs* by masking them with 0. While it successfully aids the model to maintain a decent performance in the absence of a modality, it does not provide any gains when all modalities are present. This might happen because ModDrop cannot address the incompatibility in training dynamics between modalities. For a multi-modal network implemented with a single-layer fully-connected network and concatenation of the multi-modal inputs, it is easy to show that when a modality's input is dropped by setting it to zero, the gradients of the parameters of the dropped modality are also zero. Thus, for single-layered networks ModDrop can slow down the learning speed of a dropped modality by not updating its parameters. Nonetheless, this is not true for multi-layered networks; due to the addition of bias in each layer, the intermediate representations are non-zero even with zero inputs. In this scenario, zero inputs do not result in zero parameter gradients, and therefore dropping a modality's input does not decrease its training speed.

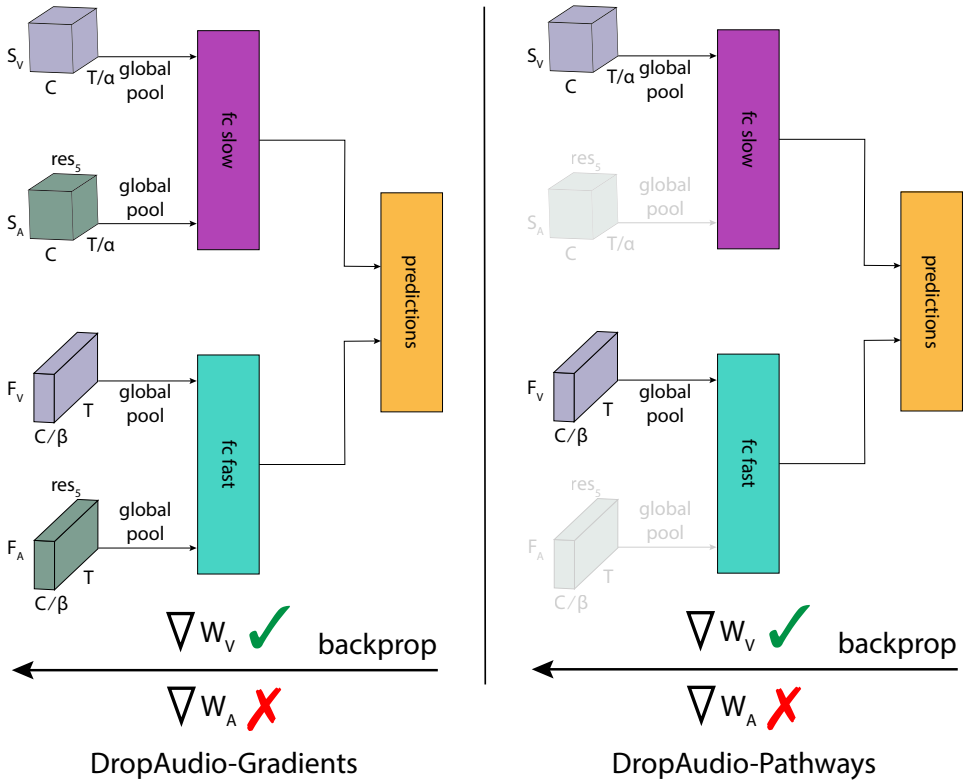


Figure 4.6: *DropAudio regularisation. DropAudio-Gradients (left) drops randomly the gradients of the auditory streams in the backward pass, effectively slowing down the training of the auditory network by not updating its parameters. DropAudio-Pathways drops randomly the auditory streams in the forward pass which has the effect of dropping their gradients in the backward pass too. Here, DropAudio is demonstrated for mid-level fusion for simplicity of illustration, while it can be applied in late and multi-level fusion too.*

DropPathway drops randomly the *auditory lateral connections* of the AVSlowFast architecture of [207]. This corresponds to simultaneously removing all the audio-visual multi-level fusion blocks, allowing the auditory stream to be combined with the visual streams only via late fusion in the classification layer (see Fig. 2.10). This exposes a weakness of DropPathway: it does not completely drop the audio pathway as it ignores dropping its connection with the classifier. This has two main consequences. First, DropPathway is not capable of slowing down the learning of the entire audio network but only the auditory lateral connections parameters. This stems from the fact that auditory layers with lateral connections to the visual streams are also connected to the next auditory layer, *i.e.* they employ two output heads. Therefore, they receive gradient contributions from both heads. As gradient contributions from multiple heads are additive (rather than multiplicative), even when auditory lateral connections are dropped, gradients are backpropagated to the audio network since it is linked to the prediction layer, and as a result its parameters keep updating. This could result in overfitting due to insufficient ad-

justment of the training dynamics of the two modalities. Second, DropPathway does not train the classifier to perform well in the absence of audio, as the classifier consistently depends on audio inputs in training, and thus its performance can still be undermined from noisy audio or silent actions. Moreover, co-adaptations between visual and auditory representations can only incompletely be alleviated through dropping the auditory lateral connections, as they can still co-adapt through their connection in the classifier, *i.e*. errors from the audio network can still be injected in the visual pathway and vice-versa.

We propose a straightforward solution which we term DropAudio-Pathways: drop the connection of the audio streams to the prediction layer along with audio in the multi-level fusion blocks (when present), effectively performing a full drop of the audio pathways. Concretely, for all fusion schemes that we examine, this corresponds to randomly masking with zeros the output of the last convolutional layer of both Slow and Fast auditory streams, *i.e*. the output of res_5. For late fusion and mid-level fusion, this is sufficient to drop the entire audio pathway, while for multi-level fusion we also mask with zeros the audio input features to the fusion blocks, adapted from DropPathway that masks the auditory lateral connections, as the multi-level fusion techniques we investigate do not employ lateral connections. Similarly to DropPathway and ModDrop and differently from vanilla Dropout, DropAudio-Pathways drops randomly an entire layer rather than a random subset of the units of a layer, *i.e*. it utilises a Bernoulli random variable per layer as opposed to using a Bernoulli random variable per unit. Also, note that in the case of multi-level fusion, for all DropAudio-Pathways layers in both the fusion blocks and before the prediction layer, the same Bernoulli random variable is used, *i.e*. all the layers to which DropAudio-Pathways is applied are either dropped or retained simultaneously to implement a full drop of the audio network.

So far, we have only described how DropAudio-Pathways eliminates all possible pathways through which audio can contribute in the predictions in the forward pass. Importantly, it also blocks any auditory parameter gradient computation in the backward pass. To understand this, let's first consider the case of late or mid-level fusion where there are no multi-level audio-visual links. When a DropAudio-Pathways layer that is inserted on top of the auditory res_5 layers sets their outputs to zero, all the auditory parameter gradients up to res_5 are also zero. That is because when a DropAudio-Pathways layer drops all its input features producing zero outputs, it backpropagates zero gradients as well. From the chain rule used to derive backpropagation, the gradients at a given layer are calculated as the product of intermediate gradients from the output layer down to that layer. Consequently, the gradients of any layer preceding the DropAudio-Pathways layer include a term corresponding to the gradients of the DropAudio-Pathways layer, and become zero when DropAudio-Pathways backpropagates zeros. Furthermore, when multi-level fusion comes into play, the additional DropAudio-Pathways layers on

the audio inputs to fusion prevent gradients from the visual pathways to leak into the auditory ones.

In conclusion, our proposed DropAudio-Pathways decreases the training momentum of all parameters of the audio streams, which can lead to better generalisation due to more adequate adaptation of the learning dynamics of audio and vision, compared to DropPathway. Finally, dropping audio from the classifier in addition to possibly existing multi-level fusion blocks has two benefits: 1) the classifier is trained to be robust to noisy audio or silent actions, and 2) features co-adaptations through late fusion are prevented.

DropAudio-Gradients. We also devise a novel approach of dropping audio that has not been explored before. This scheme aims solely to make the training speed of visual and auditory nets compatible, by randomly dropping the audio SGD parameter updates while always updating the visual net parameters. As depicted in Fig. 4.6, its main difference with DropAudio-Pathways is that while the latter drops audio in both the forward and the backward pass, DropAudio-Gradients drops only the gradients of the auditory parameters in the backward pass. Thus, DropAudio-Gradients cannot train the visual net to be robust to the absence of audio, as it allows the network's predictions and consequently the visual parameter gradients to rely on audio. Accordingly, we propose this scheme to disentangle the effect of the incompatibility of learning pace between audio and vision from the effect of audio-visual co-adaptations.

In more detail, at each training iteration, with a probability $q_A = 1 - p_A$, the error is not backpropagated in the audio network, and therefore gradients are not calculated and set as $\nabla W_A = 0$. This can be seen as multiplying the audio network gradients in an SGD update with a Bernoulli random variable $\delta_A \in \{0, 1\}$:

$$W_A^{t+1} = W_A^t - \lambda \delta_A \nabla W_A, \tag{4.1}$$

where λ is the common learning rate of visual and audio nets. Taking the expectation w.r.t. δ_A leads to:

$$\mathbb{E}_{\delta_A}(W_A^{t+1}) = W_A^t - \lambda p_A \nabla W_A. \tag{4.2}$$

Therefore, the learning rate of the audio network is expected to be scaled with the probability p_A of retaining ∇W_A. For any value of $p_A < 1$, the training of the audio streams is slowed down (given that $p_V = 1$, where p_V is the probability of retaining the video streams' gradients). Although this technique is equivalent in expectation with simply setting a lower learning rate for the audio network, randomly dropping the audio SGD updates provides extra regularisation as different audio clips are dropped in each epoch [207].

4.3 Experimental Setup

4.3.1 Datasets

VGG-Sound. We utilise VGG-Sound [30] as an audio recognition benchmark for evaluating ASF, as it captures human actions, sound-emitting objects as well as interactions, and therefore it is appropriate for auditory activity recognition. Audio is sampled at 16kHz. We do not employ this dataset in our experiments with AVSF^2, where we focus on egocentric action recognition.

EPIC-KITCHENS-100. We employ EPIC-KITCHENS-100 [36] to evaluate both ASF and AVSF^2. Remember that EPIC-KITCHENS-100 is an extension of EPIC-KITCHENS-55 [35, 37]. The extension videos are recorded at 50fps, different from EPIC-KITCHENS-55 at 60fps. Audio has a sampling rate of 48kHz.

4.3.2 Implementation details of ASF

Feature extraction. We extract log-mel-spectrograms with 128 Mel bands using the Librosa library. For both datasets, we first convert audio to single-channel. For VGG-Sound, we use 5.12s of audio with a window of 20ms and a hop of 10ms, resulting in spectrograms of size 512×128. For EPIC-KITCHENS-100, we resample audio at 24kHz, and we use 2s of audio with a 10ms window and a 5ms hop, resulting in spectrograms of size 400×128. For clips $<$ 2s in EPIC-KITCHENS-100, we duplicate the last time-frame of the log-mel-spectrogram. We found that these audio input lengths perform the best based on preliminary experiments. An explanation for the need of longer inputs in VGG-Sound is that it contains audio examples of 10s each while the average action length in EPIC-KITCHENS-100 is 3.1s. For setting the window and hop length for extracting spectrograms, we follow the empirical rule discussed in Subsec. 3.2.1, where the length of the window (and hop) should scale linearly with the input length. Accordingly, for EPIC-KITCHENS-100 we use the same values as the ones used for EPIC-KITCHENS-55 in Ch. 3, as the input length is not significantly different. For VGG-Sound, we double the window and hop size as the input length is $> \times 2$ the input length used for EPIC-KITCHENS-100.

Train / Val details. All models are trained using SGD with momentum set to 0.9 and cross-entropy loss. We train on EPIC-KITCHENS-100 as a multitask learning problem (similarly to the previous chapter), using two prediction heads, one for verbs and one for nouns. We train on VGG-Sound from random initialisation for 50 epochs using a learning rate of 0.01 and fine-tune on EPIC-KITCHENS-100 using the VGG-Sound pretrained models with a learning rate of 0.001 for 30 epochs. We drop the learning rate by 0.1 at epochs 30 and 40 for

VGG-Sound, and at epochs 20 and 25 for EPIC-KITCHENS-100. For fine-tuning, we freeze Batch-Normalisation layers except the first one (we also utilised this technique in TBN). For regularisation, we use dropout on the concatenation of Slow and Fast streams with probability 0.5, plus weight decay in all trainable layers using the value of 10^{-4}. For data augmentation during training, we use the implementation of SpecAugment [140] from [1] and set its parameters as follows: 2 frequency masks with F=27, 2 time masks with T=25, and time warp with W=5. During training we randomly extract one audio segment from each clip. During testing we average the predictions of 2 equally distanced segments for VGG-Sound, and 10 for EPIC-KITCHENS-100. We set $\alpha = 4$ and $\beta = 8$ in all our experiments.

4.3.3 Implementation details of AVSF^2

RGB. For the visual modality, we utilise the publicly available extracted RGB frames from the videos of EPIC-KITCHENS-100. As shown in Table 4.2, clips are formed from 128 consecutive frames, which correspond to 2.13s under 60fps. For the videos with 50fps, we adjust the number of frames per clip to 107, such that the same duration (2.13s) is covered (as done in SlowFast codebase [47] to cope with variable frame rates in videos). From these, 64 uniformly spaced frames are sampled (regardless of the video's frame rate), which form the final visual input. For clips < 64 frames, we replicate the last frame of the clip.

Audio feature extraction. We compute log-mel-spectrograms for EPIC-KITCHENS-100 with a slightly different procedure than the one used for ASF. Here, we desire audio to be in full sync with video, *i.e.* their start/end times to coincide, which is accomplished by sampling only the start/end frames of the video clip and mapping them to the start/end samples of the audio clip. Thus, audio clips are also 2.13s long. The same window size, hop size and number of Mel bands are used as for ASF, producing spectrograms of size 425×128

Train / Val details. We use the Kinetics-400 [90] pre-trained model for the visual pathways provided by the authors of [47], and our VGG-Sound pre-trained model for the auditory pathways. Differently from ASF, we fine-tune by keeping the Batch-Normalisation layers unfrozen (while we freeze them for training the visual Slow-Fast baseline). We also apply warm-up, by starting with a learning rate of 0.001 and linearly increase it to 0.01 during the first epoch. Dropout is always applied before the prediction layer. For late fusion, this corresponds to two separate dropout layers, one on the concatenation of Slow streams and the other on the concatenation of Fast streams. For mid-level fusion, dropout is applied on the concatenation of the outputs of Slow and Fast fusion layers. And finally, in multi-level fusion, dropout acts on the concatenation of all four streams. For DropAudio, we use a $p_A = 0.2$ (*i.e.* we drop audio in training 80% of the time, while it is fully utilised in testing). We employ the same visual data augmentations used in [47], while for audio we use SpecAugment. During training, we

randomly sample one audio-visual clip from each video. In inference, we sample 10 equally distanced clips per video, and a centre crop per visual clip (spectrograms are not cropped), and average their predictions. For the visual streams, we use $\alpha = 4$ and $\beta = 8$, same as for ASF. All unspecified hyperparameters remain identical as the ones used for ASF.

4.3.4 Evaluation and baselines

Evaluation metrics. For VGG-Sound, we follow the evaluation protocol of [30, 75] and report mAP, AUC, and d-prime. Additionally, we report top-1/5% accuracy. mAP is the mean Average Precision (AP) across all classes. AP is the area under the Precision-Recall (PR) curve. For classification, per-class AP is computed by taking the predictions of all examples for a given class, ranking them in descending order based on the predicted scores, and then computing the precision and recall at each rank. AP is then calculated by computing the weighted average of the precision values across ranks using the difference in recall between consecutive ranks as the weight. AUC is the area under the Receiver Operating Characteristic Curve (ROC). The ROC curve is the recall as a function of the false positive rate, *i.e*. the probability of incorrectly classifying a negative example as positive. [30, 75] calculated the average AUC across all classes. A perfect classification gives an AUC of 1.0. In signal detection theory, d-prime is the separation between the means of the signal and noise distributions. In classification, it expresses class separation, *i.e*. how well the classifier discriminates positives from negatives. The definition used in [30, 75] expresses d-prime as a function of AUC: $d' = \sqrt{2}F^{-1}(\text{AUC})$, where F^{-1} is the inverse cumulative distribution function for a unit Gaussian.

While we report mAP to be able to directly compare our approach with [30], there are certain methodological issues with using it [48]. The most notable one is that precision values between two precision points on the PR curve cannot be computed as a linear interpolation of the two points; thus it is not meaningful taking the arithmetic mean of precision points as it leads to calculation errors. ROC curves are more appropriate as they do not suffer from the issues of PR curves, and for imbalanced classification problems where PR analysis is more desirable, one could use Precision-Recall-Gain Curves [48] which addresses the issues of PR curves. Moreover, although we report d-prime for a full one-to-one comparison with [30], it is monotonically related with AUC, and therefore redundant.

For EPIC-KITCHENS-100, we follow the evaluation protocol of [36] and report top-1 and top-5 % accuracy for the validation and test sets separately, as well as top-1% accuracy for the subset of unseen participants within val/test. For evaluating AVSF^2, we also report top-1% accuracy of tail classes.

Baselines and ablation study for ASF. We compare to published state-of-the-art results in

each dataset, considering methods that utilise only audio. For VGG-Sound, we also compare against [122] using their publicly available code, which is the closest work to ours in motivation, as it uses two audio streams separating input into low/high frequencies.

Training the model in [122] using the publicly available code with the default hyperparameters on VGG-Sound provided poor results. We tuned the hyperparameters as follows: We set the maximum learning rate to 0.01, train the network for 62 epochs, with alpha $= 0.1$ for mixup. Lastly, we adjusted the number of FFT points to 682 for log-mel-spectrogram extraction, to apply a window and hop length similar to the ones in [122] (their datasets are sampled at 48kHz and 44.1kHz, while VGG-Sound is sampled at 16kHz).

We also perform an ablation study investigating the importance of the two streams as follows:

- Slow, Fast: We compare to each single stream individually.
- Enriched Slow stream: We combine two Slow streams with late fusion of predictions, as well as a deeper Slow stream (ResNet101 instead of ResNet50).
- Slow-Fast without multi-level fusion: Streams are fused with late fusion, by averaging their predictions, without lateral connections.

Finally, we provide an ablation of separable convolutions on VGG-Sound.

Baselines and ablation study for AVSF^2. We compare to state-of-the-art audio-visual as well as visual-only methods in EPIC-KITCHENS-100. Moreover, we carry out experiments and ablations to assess the significance of fusing visual and auditory Slow-Fast streams, to showcase the importance of DropAudio regularisation for effective fusion, as well as to identify the best technique of combining the streams. To this end, we first perform a modality ablation comparing the proposed audio-visual model, AVSF^2, to the visual-only and auditory-only Slow-Fast. Then, we compare the two different DropAudio techniques we introduced and we also ablate DropAudio from our method. And finally, we compare the four different fusion schemes that we presented.

4.4 Results for ASF

4.4.1 EPIC-KITCHENS-100

Our proposed network achieves state-of-the-art results as can be seen in Table 4.3 for both Val and Test. Our previous results [36] use a TSN with a single-stream BN-Inception architecture [194], *i.e.* the audio network from TBN in the previous chapter, initialised from ImageNet, while here we utilise pre-training from VGG-Sound. Moreover, the audio network of TBN was

Split	Model	Overall						Unseen Participants			# Param.
		Top-1 Accuracy (%)			Top-5 Accuracy (%)			Top-1 Accuracy (%)			
		Verb	Noun	Action	Verb	Noun	Action	Verb	Noun	Action	
Val	Damen *et al*. [36]	42.63	22.35	14.48	75.84	44.60	28.23	35.40	**16.34**	**9.20**	10.67M
	Slow	41.17	18.64	11.37	77.52	42.34	24.20	34.93	14.65	7.79	24.89M
	Fast	39.84	17.07	8.76	76.94	41.31	22.01	33.33	15.21	6.57	00.49M
	Two Slow Streams	41.41	19.06	11.41	77.87	43.05	24.73	34.37	14.27	6.85	49.78M
	Slow ResNet101	42.24	19.35	12.12	78.14	42.83	25.30	37.37	13.90	7.61	46.11M
	Slow-Fast (late fusion)	42.28	19.23	11.27	78.40	44.17	25.36	34.65	15.68	7.70	25.38M
	Slow-Fast (Proposed)	**46.05**	**22.95**	**15.22**	**80.01**	**47.98**	**30.24**	**37.56**	**16.34**	8.83	26.88M
Test	Damen *et al*. [36]	42.12	21.51	14.76	75.06	41.12	25.86	37.45	17.74	11.63	10.67M
	Slow-Fast (Proposed)	**46.47**	**22.77**	**15.44**	**78.30**	**44.91**	**28.56**	**42.48**	**20.12**	**12.92**	26.88M

Table 4.3: *Results on EPIC-KITCHENS-100. We provide an ablation study over the Val set on the importance of the two streams, as well as a comparison to published state-of-the-art [36], where our proposed model outperforms [36] in audio recognition on both Val and Test. Models in the ablation study are split in three groups: Top: Single streams, Middle: Enriched Slow Streams with i) late fusion of two Slow streams, and ii) a deeper Slow stream, Bottom: Slow-Fast streams with i) late fusion, and ii) the proposed architecture with multi-level fusion. Number of parameters per model is also shown on the rightmost column. Slow outperforms Fast. Our proposed architecture outperforms enriched Slow streams, crucially with less parameters. Multi-level fusion surpasses the performance of late fusion.*

trained in Ch. 3 and [36] without data augmentation, while here we train ASF using SpecAugment. Our proposed architecture outperforms [36] by a good margin, showcasing the benefits of combining Slow and Fast auditory streams for egocentric action recognition.

We report the ablation comparison using the published Val split. The significant improvement in our proposed Slow-Fast architecture when compared to Slow and Fast streams independently shows that there is complementary information in the two streams that benefits audio recognition. The Slow stream performs better than Fast, due to the increased channel capacity. When comparing to the enriched Slow architectures (see the last column of Table 4.3 for number of parameters), our proposed model still significantly outperforms these baselines, showcasing the need for the two different pathways. We conclude that the synergy of Slow and Fast streams is more important than simply increasing the number of parameters of the stronger Slow stream. Finally, our proposed architecture consistently outperforms late fusion, indicating the importance of multi-level fusion with lateral connections.

4.4.2 VGG-Sound

We report results in Table 4.4 comparing to state-of-the-art from [30], which uses a single-stream ResNet50 architecture, [122] which uses a ResNet variant with 19 layers as backbone

Model	Top-1	Top-5	mAP	AUC	d-prime
Chen *et al.* [30]	51.00	76.40	0.532	0.973	2.735
McDonnell & Gao [122]	39.74	71.65	0.403	0.963	2.532
Slow	45.20	72.53	0.472	0.967	2.607
Fast	41.44	70.68	0.442	0.966	2.576
Two Slow Streams	45.80	72.78	0.482	0.969	2.633
Slow ResNet101	45.60	72.27	0.476	0.968	2.615
Slow-Fast (late fusion)	46.75	73.90	0.498	0.971	2.671
Slow-Fast (Proposed)	**52.46**	**78.12**	**0.544**	**0.974**	**2.761**

Table 4.4: *Results on VGG-Sound. We compare to published results from [30] which is a ResNet50, and also against [122] (using their publicly available code) which is a two-stream ResNet variant. Our proposed model outperforms both. Furthermore, we perform the same ablation analysis as in EPIC-KITCHENS-100, where we come to the same conclusions. One noteworthy difference is that here, the performance gap between Slow, Fast and the proposed Slow-Fast is larger.*

Model	Top-1	Top-5	mAP	AUC	d-prime
Chen *et al.* [30]	51.00	76.40	0.532	0.973	2.735
ResNet50	52.23	78.08	0.542	0.974	2.747
ResNet50-separable	52.38	77.81	**0.544**	**0.975**	**2.777**
Slow-Fast (Proposed)	**52.46**	**78.12**	**0.544**	0.974	2.761

Table 4.5: *Ablation of separable convolutions on VGG-Sound. An original ResNet50 without separable convolutions (trained with our training setup) outperforms Chen* et al. *[30], which is also a ResNet50. ResNet50 with separable convolutions outperforms the original ResNet50 in all metrics except Top-5. Slow-Fast provides a further increase in performance in Top-1 and Top-5.*

for their two-stream architecture with significantly less parameters than our model at 3.2M parameters, as well as ablations of our model. We report the best performing model on the test set in each case. Our proposed Slow-Fast architecture outperforms [30] and [122]. The rest of our observations on the ablations from EPIC-KITCHENS-100 hold for VGG-Sound as well, with a key difference: the gap in performance between single streams and our proposed two-stream architecture is even bigger for VGG-Sound, indicating more complementary information in the two streams. The fact that Slow-Fast outperforms Slow by such a large accuracy gap with an insignificant increase in parameters indicates the efficient interaction between Slow and Fast streams.

4.4.3 Ablation of separable convolutions

We provide an ablation of separable convolutions in Table 4.5. We trained the ResNet50 architecture as proposed in [71] without separable convolutions, as well as a variant with separable convolutions. We compare this to the published results by Chen *et al.* [30] that also

uses a ResNet50 architecture. Our reproduced results (ResNet50) already outperform [30]. ResNet50-separable has separable convolutions as used in our Slow-Fast network (see Fig. 4.2 and Table 4.1).

Results show that ResNet50-separable achieves slightly better results than ResNet50 in all metrics except Top-5. Although accuracy is not significantly increased in this ablation, we employ separable convolutions in our proposed architecture, following our motivation to attend differently to frequency and time. These results also show that a single stream ResNet50 has comparable performance to our two stream proposal, however ours performs marginally better in accuracy, but more importantly the two streams accommodate different characteristics of audio classes, satisfying our expectations that Slow learns harmonic while Fast percussive sounds, as shown next.

4.5 What is Learnt from Each of the Auditory Streams?

Here, we provide further insight into what each of the Slow and Fast streams learn, through class analysis and visualising feature maps from each stream, on VGG-Sound.

4.5.1 Class performance of the two streams

In Fig. 4.7, we distinguish between VGG-Sound classes that are better predicted from the Slow stream to the left, and classes that are better predicted from the Fast stream to the right. To obtain these, we calculated per-class accuracy and retrieved classes for which the accuracy difference is above a threshold. Particularly, we used $\text{accuracy}_{\text{Slow}} - \text{accuracy}_{\text{Fast}} > 20\%$ to retrieve classes best predicted from Slow and $\text{accuracy}_{\text{Fast}} - \text{accuracy}_{\text{Slow}} > 10\%$ to retrieve classes best predicted from Fast. We used a higher threshold for the Slow stream as it more frequently outperforms the Fast stream, as shown in our earlier results.

As can be seen in Fig. 4.7, Slow predicts better animals and scenes. This matches our intuition that Slow focuses on learning frequency patterns as different animals make distinct harmonic sounds at different frequencies, e.g. mosquito buzzing vs whale calling, requiring a network with fine spectral resolution to distinguish between those. In Scenes, there are classes such as sea waves, airplane and wind chime, that contain harmonic sounds as well.

The Fast stream, in contrast, can better predict classes with percussive sounds like playing drum kit, tap dancing, woodpecker pecking tree, and popping popcorn. This also matches our design motivation that the Fast stream learns better temporal patterns as these classes contain temporally localised sounds that require a model with fine temporal resolution. Interestingly, Fast is better at human speech, laughter, singing, and other human voices, where we speculate

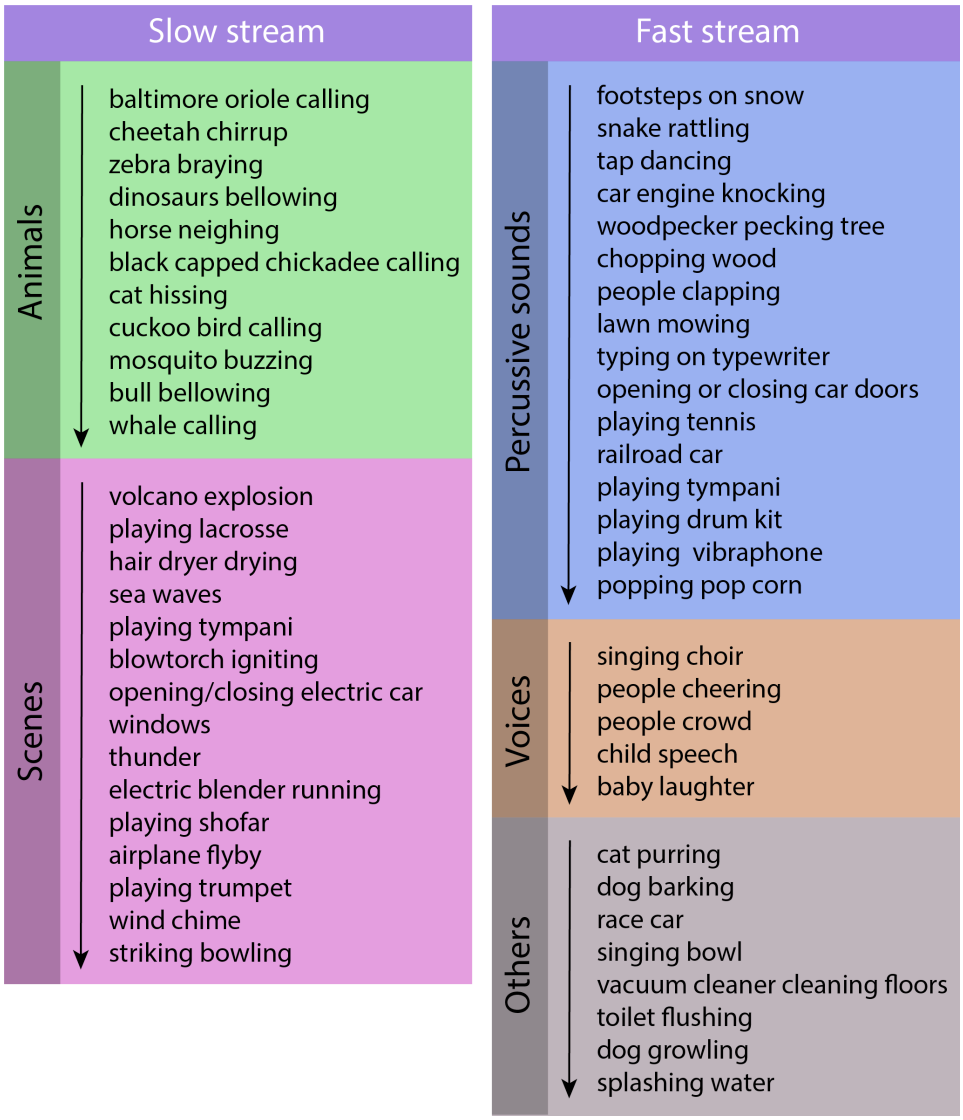


Figure 4.7: *Classes from VGG-Sound that are significantly better predicted from Slow (left) versus Fast (right) streams. These sets of classes are assembled by calculating class-wise accuracy and retrieving classes for which the accuracy difference between the two streams is above a threshold. A threshold of* 20% *is used to retrieve classes best predicted from Slow while a threshold of* 10% *for classes best predicted from Fast. Slow can better predict animals and scenes, containing harmonic sounds, while Fast is better at percussive sounds, as well as human voices.*

that it can better capture articulation.

4.5.2 Visualising feature maps

We show examples of feature maps from Slow and Fast streams, when trained independently (Fig. 4.8). In each case, we show two samples from classes that are better predicted from the corresponding stream. For Slow, these are sea waves and mosquito buzzing, compared to

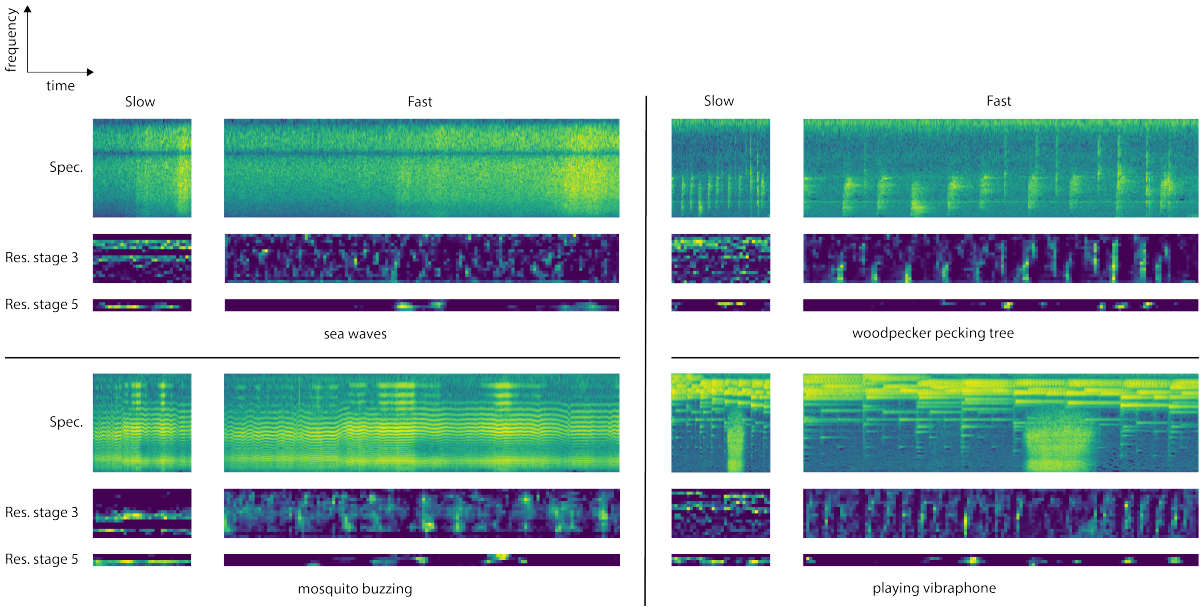


Figure 4.8: *Feature maps from classes that are better predicted from Slow (left) and Fast (right) streams. In each case, we show the input spectrogram and feature maps from residual stages 3 and 5. Time is represented horizontally while frequency vertically, and a single channel from the feature maps is shown. For 'sea waves' and 'mosquito buzzing' (where Slow performs better), Slow extracts frequency patterns over time while Fast is attempting to perform temporal localisation. For 'woodpecker pecking tree' and 'playing vibraphone' (which are better predicted from Fast), Fast temporally detects the hits of the woodpecker in the tree as well as the hits of the mallets on the vibraphone.*

woodpecker pecking tree and playing vibraphone for Fast. In each case, we show the input spectrogram as well as feature maps from residual stages 3 and 5. In each plot, the horizontal axis represents time while the vertical axis corresponds to frequency. We visualise a single channel from each feature map, manually chosen.

In Fig. 4.8, we demonstrate that Fast is capable of temporally detecting the hits of the woodpecker on the tree as well as the hits of the mallets on the vibraphone, as there are activations in both residual stages 3 and 5 that temporally coincide with the hits in the input spectrogram. In contrast, Slow extracts frequency patterns that do not seem to be useful for discriminating these classes that contain temporally localised sounds. For sea waves and mosquito buzzing, Slow extracts frequency patterns over time, where by comparing the input spectrogram and residual stage 3, it can be seen that the feature maps can capture the energy distribution of the input signal in different frequency bands. Fast aims to temporally localise events, which does not assist the discrimination of these classes that contain harmonic sounds.

These observations provide some further evidence that Slow models harmonic sounds by learning to attend at the relevant frequency bands, whereas Fast models percussive sounds by learn-

		Overall						Unseen Participants			Tail-classes		
		Top-1 Accuracy (%)			Top-5 Accuracy (%)			Top-1 Accuracy (%)			Top-1 Accuracy (%)		
V	A	Verb	Noun	Action	Verb	Noun	Action	Verb	Noun	Action	Verb	Noun	Action
✓	✗	67.54	51.54	40.26	90.92	75.60	60.03	58.22	43.85	31.27	**39.43**	24.47	20.71
✗	✓	44.97	21.80	14.32	77.81	45.99	28.89	38.78	14.84	08.26	17.67	09.32	05.54
✓	✓	**68.89**	**53.64**	**42.44**	**91.09**	**76.46**	**60.84**	**61.88**	**45.54**	**33.99**	39.32	**27.68**	**21.35**

Table 4.6: *Modality ablation on the validation set of EPIC-KITCHENS-100. With the addition of audio (A), the performance improves consistently over the visual-only network (V).*

ing to temporally localise them.[2]

4.6 Results for AVSF^2

We present an ablation analysis in Subsec. 4.6.1 and 4.6.2 as well as comparison of fusion strategies in Subsec. 4.6.3, performed on the validation set of EPIC-KITCHENS-100. In Subsec. 4.6.4, we compare AVSF^2 to the state of the art in both the validation and test sets of EPIC-KITCHENS-100.

4.6.1 Modality ablation

An ablation of modalities is shown in Table 4.6. Note that the performance of the audio modality alone, *i.e.* Auditory Slow-Fast, is slightly worse than the one reported in Table 4.3, as the feature extraction process differs here as described in Subsec. 4.3.3. Although the modification to the log-mel-spectrogram computation is introduced to synchronise visual and auditory inputs when modalities are trained jointly, for this ablation we follow the same procedure for single modalities too, for a fair comparison between individual modalities and the joint network; that is, for training the auditory network individually, we sample start/end video frames which are then converted to start/end audio samples to compute the log-mel-spectrogram. Consequently, the drop in performance of the audio network compared to Table 4.3 arises from reduced temporal augmentation of audio inputs, *i.e.* audio start/end times are sampled at larger intervals (video-frame basis vs audio-sample basis) allowing smaller temporal variation of auditory inputs.

One compelling observation is that, compared to the modality ablation in the previous chapter,

[2] Additionally, some qualitative results for both EPIC-KITCHENS and VGG-Sound where we play the audio and show the ground truth as well as the individual predictions of Slow and Fast streams compared to our proposed Slow-Fast architecture can be seen (and heard) at https://youtu.be/3q31N43Dr4k?t=331.

			Overall						Unseen Participants			Tail-classes		
			Top-1 Accuracy (%)			Top-5 Accuracy (%)			Top-1 Accuracy (%)			Top-1 Accuracy (%)		
Modality	Fusion	DropAudio	Verb	Noun	Action	Verb	Noun	Action	Verb	Noun	Action	Verb	Noun	Action
V	-	-	67.54	51.54	40.26	90.92	75.60	60.03	58.22	43.85	31.27	39.43	24.47	20.71
AV	**Late**	-	67.08	50.50	39.42	90.11	74.41	58.31	59.06	41.41	30.14	36.53	24.79	19.13
		Gradients	68.58	52.42	40.90	90.73	76.62	**61.58**	59.62	**44.79**	32.11	38.18	25.26	19.77
		Pathways	**68.89**	**53.00**	**41.81**	**91.25**	**76.65**	**61.58**	**61.03**	44.23	**32.39**	**40.23**	**26.16**	**20.71**
	Mul.-lev.	-	66.99	51.15	39.92	90.00	75.26	58.68	59.44	42.25	32.02	37.22	26.84	20.26
		Gradients	**69.00**	52.71	41.45	**91.26**	**76.48**	**61.36**	59.15	43.57	30.99	**40.11**	26.89	20.81
		Pathways	68.89	**53.64**	**42.44**	91.09	76.46	60.84	**61.88**	**45.54**	**33.99**	39.32	**27.68**	**21.35**

Table 4.7: *Ablation of DropAudio on the validation set of EPIC-KITCHENS-100 for both late and multi-level fusion. When DropAudio is not incorporated (-) in the audio-visual network (AV), the performance degrades with the addition of audio compared to the visual-only model (V) for both late and multi-level fusion. When DropAudio is employed, the inclusion of audio boosts the performance of the model. DropAudio-Gradients increases fairly the accuracy, and DropAudio-Pathways improves the results further in most metrics.*

here the gap in performance between the visual and auditory networks is larger, due to the enhanced performance of the visual Slow-Fast network. While in the previous chapter the audio stream has shown comparable performance to the RGB stream in top-1 verbs, here visual Slow-Fast that takes only RGB inputs as well performs significantly better than auditory Slow-Fast in all metrics including top-1 verbs. This is reasonable as the RGB network in the previous chapter is single-stream, modelling only appearance, while visual Slow-Fast enables both appearance (Slow) and motion (Fast) modelling.

Despite the restricted auditory temporal augmentation, and the strong performance of visual Slow-Fast alone, combining auditory Slow and Fast streams with their visual analogues boosts decently the performance of the model, outperforming the visual-only network. This showcases that visual and auditory Slow-Fast networks contain complementary information which can be leveraged for more accurate action recognition, when these are fused and trained properly as will be shown in the next subsections.

4.6.2 Audio-visual regularisation ablation

We demonstrate the necessity of DropAudio for training AVSF^2 through an ablation in Table 4.7 for both late and multi-level fusion. Similar to the findings of [207] and [196], when DropAudio is not incorporated in training, not only the performance does not improve with the addition of audio, but the audio-visual model performs worse compared to using only vision. Importantly, this finding is consistent for both late and multi-level fusion, validating that this is not a property of a particular type of fusion, but more probably an attribute of the particular architecture that we consider for each modality. In the previous chapter, TBN behaved

differently with the addition of audio, where performance improved without the need of any form of audio-visual regularisation. Here, one cause of deterioration of performance could be that both the audio architecture and the full audio-visual network are higher capacity models compared to the audio stream in TBN and the complete TBN architecture, respectively; TBN has a lightweight single auditory stream with 10.67M parameters while Auditory Slow-Fast has 26.88M parameters (comparison of #parameters in Table 4.3), and the full TBN model has 32.60M parameters vs 65.80M parameters in AVSF^2. The increased number of parameters in audio-visual networks (compared to uni-modal networks) is also discussed in [196] as a potential source of overfitting. A higher capacity audio-visual model is more prone to complex co-adaptations across modalities, and noisy audio as well as silent actions can easier ovefit the model.

When DropAudio is employed, performance improves allowing audio to benefit action recognition. DropAudio-Gradients obtains a good boost in performance compared to not using DropAudio, demonstrating that the incompatibility in learning dynamics between audio and vision is a significant factor of overfitting and that adapting their learning pace increases generalisation. For multi-level fusion, DropAudio-Gradients outperforms DropAudio-Pathways in top-5 accuracy for all verb, noun and action, as well as in top-1 and tail verbs. We conclude that DropAudio-Gradients is a reasonable baseline for audio-visual regularisers for action recognition. DropAudio-Pathways increases the accuracy of the model further in most metrics, showcasing that in addition to addressing the discrepancy in training speed between modalities, training an audio-visual network by also calibrating the degree it depends on audio (*i.e.* dropping the audio pathways in the forward pass along with their gradients in the backward pass) is important as well. One possible explanation is that DropAudio-Pathways inhibits co-adaptations between the modalities, by limiting the amount to which irrelevant noise in the audio data stream can contaminate the visual representations and the predictions of the model. Note that even for the metrics mentioned above where DropAudio-Gradients has better performance in multi-level fusion, DropAudio-Pathways achieves comparable performance. As the results show that DropAudio-Pathways is more effective overall, tackling both sources of overfitting we consider, we utilise it as a default choice in all our experiments.

4.6.3 Comparison of fusion techniques

We compare the efficacy of the different fusion schemes we consider in Table 4.8. First of all, our results demonstrate a case, commonly observed among different fusion studies: late fusion is a strong competitor over more complex fusion approaches, where for example it obtains second best top-1 action accuracy. In general, mid-level fusion as well as MMTM and Cross-Attention *without* residual connections show similar performance to late fusion.

4.6 Results for AVSF[2]

	Overall						Unseen Participants			Tail-classes		
	Top-1 Accuracy (%)			Top-5 Accuracy (%)			Top-1 Accuracy (%)			Top-1 Accuracy (%)		
Fusion	Verb	Noun	Action	Verb	Noun	Action	Verb	Noun	Action	Verb	Noun	Action
Late	68.89	53.00	41.81	**91.25**	76.65	61.58	61.03	44.23	32.39	**40.23**	26.16	20.71
Mid-level	**69.35**	52.01	41.39	91.21	76.61	**61.75**	60.56	44.88	33.43	38.52	23.95	19.58
MMTM w/o res. (multi-level)	68.91	52.58	41.69	91.04	77.15	61.16	59.81	43.19	32.21	38.24	26.58	20.58
MMTM w. res. (multi-level)	68.88	53.11	41.78	91.19	**77.35**	61.71	59.72	43.57	31.36	37.78	**28.05**	20.90
Cross-Attention w/o res. (multi-level)	68.57	52.71	41.51	91.21	76.65	61.42	59.72	44.41	32.11	40.11	27.21	20.97
Cross-Attention w. res. (multi-level)	68.89	**53.64**	**42.44**	91.09	76.46	60.84	**61.88**	**45.54**	**33.99**	39.32	27.68	**21.35**

Table 4.8: *Comparison of fusion strategies using the validation set of EPIC-KITCHENS-100. Late fusion is competing more complex fusion approaches, obtaining the second place in top-1 action accuracy. Mid-level fusion as well as MMTM and Cross-Attention without residual connections perform comparably to late fusion. Cross-Attention with residual connections obtains the best performance in most metrics excluding mainly top-5 accuracy.*

The best results in several metrics are obtained using Cross-Attention *with* residual connections, while the other fusion approaches obtain best performance in two metrics at most. In Cross-Attention, residual connections provide a non-negligible boost in performance, showcasing their importance. We also add residual connections in MMTM for a more thorough evaluation. The results show that while residuals provide small boosts in MMTM, they mostly benefit Cross-Attention. This finding is an indication that our assumption holds, *i.e.* that Cross-Attention *without* residuals disregards discriminant within-modal information by attending on features that are important to the other modality, and that residual connections alleviate that by providing access to the initial feature map before attention. MMTM does not suffer from this problem, but yet its slight increase in performance after the addition of residuals could be because residuals can also ease the training of deep networks allowing for better propagation of gradient signals [71]. Note that, adding residuals is only possible for multi-level fusion, which both takes as input and produces as output visual and auditory features, allowing to add visual and auditory residuals in the outputs, respectively. Adding residuals in late and mid-level fusion is not feasible, as they produce as output a single multi-modal representation, to which visual and auditory features cannot be added. It is interesting that the performance of Cross-Attention (with residuals) is particularly better in top-1 nouns compared to the rest fusion approaches, as it could signify its ability to attend to objects that sound. To properly assess whether Cross-Attention indeed possesses this property, we should visualise its attention maps and we leave this for future exploration. We employ Cross-Attention with residuals in all our experiments as it provides the best performance overall.

Split	Model	Overall						Unseen Participants			Tail-classes		
		Top-1 Accuracy (%)			Top-5 Accuracy (%)			Top-1 Accuracy (%)			Top-1 Accuracy (%)		
		Verb	Noun	Action	Verb	Noun	Action	Verb	Noun	Action	Verb	Noun	Action
Val	TSN [194]	60.18	46.03	33.19	89.59	72.90	55.13	47.42	38.03	23.47	30.45	19.37	13.88
	TRN [217]	65.88	45.43	35.34	90.42	71.88	56.74	55.96	37.75	27.70	34.66	17.58	14.07
	TBN (Ch. 3)	66.00	47.23	36.72	90.46	73.76	57.66	59.44	38.22	29.48	39.09	24.84	19.13
	TSM [110]	67.86	49.01	38.27	90.98	74.97	60.41	58.69	39.62	29.48	36.59	23.37	17.62
	SlowFast [47]	67.54	51.54	40.26	90.92	75.60	60.03	58.22	43.85	31.27	**39.43**	24.47	20.71
	AVSF^2 (Proposed)	**68.89**	**53.64**	**42.44**	**91.09**	**76.46**	**60.84**	**61.88**	**45.54**	**33.99**	39.32	**27.68**	**21.35**
Test	TSN [194]	59.03	46.78	33.57	87.55	72.10	53.89	53.11	42.02	27.37	26.23	14.73	11.43
	TRN [217]	63.28	46.16	35.28	88.33	72.32	55.26	57.54	41.36	29.68	28.17	13.98	12.18
	TBN (Ch. 3)	62.72	47.59	35.48	88.77	73.08	56.34	56.69	43.65	29.27	30.97	19.52	14.10
	TSM [110]	65.32	47.80	37.39	89.16	73.95	57.89	**59.68**	42.51	30.61	30.03	16.96	13.45
	SlowFast [47]	64.63	48.62	37.40	88.61	73.84	56.13	57.54	41.24	28.71	**33.64**	21.27	16.05
	AVSF^2 (Proposed)	**66.06**	**51.12**	**39.83**	**89.85**	**75.94**	**58.89**	59.49	**46.35**	**33.36**	31.94	**22.83**	**16.83**

Table 4.9: *Comparison with the state of the art on both the validation (top) and test set (bottom) of EPIC-Kitchens-100. AVSF^2 outperforms all other approaches where of particular interest is TBN which is also audio-visual and SlowFast which we utilise for the visual streams of AVSF^2.*

4.6.4 Comparison with the state of the art

A comparison of AVSF^2 with state-of-the-art convolutional approaches in both the validation and test sets of EPIC-KITCHENS-100 can be seen in Table 4.9. AVSF^2 outperforms all other methods by a noticeable margin. Of particular interest is TBN (which we proposed in the previous chapter) as it is audio-visual and SlowFast since it shares the same architecture as our visual streams. Note that the results for TBN are from [36] where TBN was trained with slightly different hyperparameters than those presented in Ch. 3; namely, a batch size of 64 was used, 6 segments, and the learning rate was dropped at epoch 40 and 60, while the rest hyperparameters were left unchanged. We have already discussed the improvement of AVSF^2 over SlowFast in the ablations in Subsec. 4.6.1 and 4.6.2. Using only the visual streams of AVSF^2 (*i.e.* SlowFast), we can already outperform TBN, showcasing that the visual Slow and Fast streams have enhanced appearance and motion modelling capabilities compared to the RGB and optical flow streams of TBN, and also that there is a subtle interplay between Slow and Fast streams (through multi-level lateral connections) which is more effective than TBN's mid-level fusion. Another benefit of the visual Fast stream of AVSF^2 compared to the motion stream of TBN is that it does not require costly computation of optical flow. Our AVSF^2 equipped with high-performing visual streams and state-of-the-art auditory streams as well as more effective fusion (we've already shown that Cross-Attention is better than mid-level fusion) outperforms TBN by an even wider margin.

4.7 Conclusion

The first part of this chapter introduced a two-stream architecture for audio recognition, inspired by the two pathways in the human auditory system, fusing Slow and Fast streams with multi-level lateral connections. We demonstrated the importance of our fusion architecture through ablations on two activity-based datasets, EPIC-KITCHENS-100 and VGG-Sound, achieving state-of-the-art performance. In particular, we have shown that multi-level is better than late fusion, that the two streams have useful complementary information and that their fusion is more beneficial than simply using higher capacity single stream architectures. Class analysis revealed that Slow focuses on learning frequency semantics and is able to capture harmonic sounds such as animals and scenes, while Fast models temporal patterns and predicts better percussive sounds and human voices (possibly capturing articulation). Feature map visualisation has further verified the spectral modelling capabilities of Slow, and importantly the temporal detection capabilities of the lightweight Fast stream.

At the second part, this chapter proposed a four-stream architecture for integrating visual and auditory Slow and Fast streams by associating corresponding streams across modalities using multi-level cross-modal attention fusion with residual connections. Ablation analysis showcased that auditory Slow-Fast streams contain complementary information to visual Slow-Fast streams, and therefore can assist in increasing their performance. More importantly, the chapter offered a thorough investigation of audio-visual regularisation by randomly dropping audio, namely DropAudio, and the results have demonstrated that audio-visual regularisation is of vital importance, as when it is removed from training the auditory modality cannot benefit action recognition. Dropping the auditory pathways in the forward pass (rather than merely their gradients in the backward pass) provided higher boosts, signifying that both the training dynamics of the two modalities should be adjusted and the network's dependence on audio should be calibrated to mitigate shortcut training of the network by memorising noise in audio. Lastly, a thorough comparison of fusion schemes has demonstrated the benefits of fusing the modalities at multiple levels with Cross-Attention fusion that incorporates residual connections, pointing out that attention can effectively enhance each modality's representations. The proposed model, AVSF^2, outperforms TBN which is introduced in Ch. 3 as well as other visual approaches, obtaining state-of-the-art performance.

Despite the intriguing findings of this chapter regarding four-stream fusion and audio-visual regularisation, there are various baselines, experiments and analysis that are crucial for a more thorough evaluation of AVSF^2, and yet these have not been performed in this chapter and are left for future exploration. These are:

- *Assessment of the basic idea*: First of all, as the chapter's main motivation is that it

is beneficial to fuse the same type of streams across modalities, it would be essential to demonstrate it by contrasting it to combining different type of streams within modalities, *i.e*. Slow_V-Slow_A and Fast_V-Fast_A versus Slow_V-Fast_V and Slow_A-Fast_A (see Fig. 4.4a vs Fig. 4.4b).

- *Comparison with additional audio-visual approaches*: AVSF^2 has been compared only with TBN from audio-visual approaches. A comparison with Audiovisual SlowFast [207] is important as it is the closest to AVSF^2. Moreover, it would be necessary to compare against [196] as it is an audio-visual method reporting on EPIC-KITCHENS. Both [207] and [196] are trained on EPIC-KITCHENS-55, and thus AVSF^2 needs to be trained on EPIC-KITCHENS-55 too, for a direct comparison.

- *Scrutinising DropAudio*: Although the chapter discussed how the proposed DropAudio-Pathways can more adequately tackle the sources of overfitting in audio-visual networks that have been considered in this chapter, compared to DropPathway [207], this still needs to be verified experimentally. In addition, a comparison with Gradient-Blending [196] by incorporating it in training AVSF^2 instead of DropAudio would also be fruitful, as its motivation is to address overfitting due to inconsistent training dynamics between modalities, similar to DropAudio and DropPathway. Another valuable experiment is to test whether DropAudio-Pathways improves the performance particularly on the subsets that contain noisy audio/silent actions, and compare it with ModDrop [135] that has shown good performance with noisy/absent modalities. Dropping the visual modality in addition to dropping audio can also be explored to assess the contribution of the stronger modality to overfitting.

- *Hyperparameter tuning*: In [207], the authors have shown that it is beneficial to fuse audio-visual features with multi-level lateral connections only starting from intermediate layers, attributing it to the fact that low-level features are not generalisable across modalities. As this chapter has utilised multi-level fusion blocks across the whole architecture of AVSF^2, tuning the layers to which multi-level fusion should be applied could provide further boosts in performance.

- *Qualitative analysis*: A qualitative evaluation, where the video clips are shown along with the ground truth, the predictions of the each modality and predictions of AVSF^2, can help to identify modes of failure of the proposed model. Finally, a visualisation of attention maps from 'Cross-Attention' fusion can reveal whether sound source localisation in videos emerges using action supervision.

The architectures proposed in this chapter can be employed out of the box in different research areas. A follow up of our work [195] showed that training Auditory Slow-Fast with

self-supervision leads to strong audio representations that are transferable across a wide range of tasks, including environmental sound classification tasks, as well as speech and music tasks. However, [195] demonstrated that the transferability of the Auditory Slow-Fast representations is affected by the architecture's normalisation scheme and proposed a normaliser-free variant of Auditory Slow-Fast by employing the design principles of [20] to avoid the shortcomings of normalisation. The success of Auditory Slow-Fast in speech tasks can be attributed to its ability to model slow temporal changes, such as prosody, in the Slow stream, and faster changes, such as phonemes, consonants and temporal pitch, in the Fast stream (see [219] for similar findings in the neuroscience domain). The applicability of Auditory Slow-Fast in music tasks may be related with the ability of Slow to model the harmonic sounds of musical instruments such as guitars and violins, and the capacity of Fast to learn the percussive sounds of musical instruments such as drums. This would also make the application of Auditory Slow-Fast straightforward to music source separation. For audio-visual music source separation, AVSF^2 could assist in learning correspondences of streams across modalities for better separation, *e.g.* associate the sound of the drums with the motion of hitting them using the Fast streams.

Another exciting avenue of research is to enhance the Slow-Fast streams with more properties of the dorsal and ventral streams from neuroscience to facilitate robots able to perceive the world audio-visually using the ventral streams as well as to speak and interact with objects using the dorsal streams, as research in visual and auditory dorsal-ventral streams has revealed that the auditory dorsal stream maps acoustic speech signals into articulatory motor representations which are essential for speech production [76], while the visual dorsal stream maps a visual signal to a motor command for visually guided reaching and grasping [61]. A step towards dorsal and ventral streams for robotic manipulation has already been made in [165].

This chapter has investigated a novel paradigm for designing efficient architectures that operate on temporal streams of data, *i.e.* leveraging complementary information within and across modalities using two streams per modality, one that learns slow and the other fast changing concepts, along with merging Slow streams independently from Fast. Interestingly, Slow and Fast modelling can go beyond audio-visual applications and be employed in any modality formed as a temporal stream of data, *e.g.* lidars, with several interesting applications including visual-lidar Slow and Fast streams fusion for self-driving cars.

By scrutinising Slow-Fast architectures in vision and audio, several directions of investigation have emerged that would enable more robust and efficient training and network design. First, a learnable and adaptive stride (α) parameter would allow to adjust the complementarity of information within each stream based on the example through balancing the 'slowness' of the Slow stream in relation to Fast. While there is sufficient supporting evidence and motiva-

tion for modelling frequency and time independently, separable convolutions did not boost the performance of Auditory Slow-Fast adequately, and possibly more sophisticated approaches are required, *e.g*. [215]. The building idea of AVSF^2, *i.e*. the combination of same type of streams across modalities, need to be assessed and advanced through further means. To this end, stream-specific fusion blocks could be designed as opposed to stream-agnostic fusion that has been used in this chapter (that is, the same fusion block has been used for either Slow or Fast stream fusion). Moreover, a loss function that encourages the representations of corresponding streams across modalities to be close in space could further assist in meeting the goal of leveraging corresponding audio-visual streams. Finally, adaptive DropAudio regularisation (rather than random) by deciding whether an audio input is informative or should be dropped is worth investigating.

CHAPTER

FIVE

LEVERAGING MULTI-MODAL TEMPORAL CONTEXT

Action recognition in egocentric video streams from sources like EPIC-KITCHENS poses a number of challenges that differ substantially from those of conventional third-person action recognition – where training and evaluation is on 10 second video clips and classes are quite high-level [90]. Actions are fine-grained (*e.g.* 'open bottle') and noticeably short, often one second or shorter. Along with the challenge, the footage offers an under-explored opportunity, as actions are captured in long untrimmed videos of well-defined and at-times predictable sequences. For example the action 'wash aubergine' can be part of the following sequence – you first 'take the aubergine', 'turn on the tap', 'wash the aubergine' and finally 'turn off the tap' (Fig. 5.1). Furthermore, the objects (the aubergine and tap in this case) are persistent over some of the neighbouring actions.

This chapter investigates utilising not only the action's temporal context in the data stream, but also the temporal context from the labels of neighbouring actions. A model that attends to neighbouring actions is proposed. Note that these action start/end times are readily available in labelled datasets for untrimmed videos and do not require additional labels. These are just leveraged. Concretely, the attention mechanism of a multi-modal transformer architecture is used to take account of the context in both the data and labels, using three modalities: vision, audio and language. Motivated by the significance of audio in recognising egocentric actions (shown in Ch. 3 and [196, 208]), we include the auditory temporal context in addition to the visual clips. We also utilise context further, by training a language model on the sequence of action labels, inspired by the success of language models [21, 40, 209] in re-scoring model outputs for speech recognition [28, 70, 127] and machine translation [66] (Fig. 5.2).

Figure 5.1: *Egocentric video demonstrating two temporal context windows (pink, green), centred around the action to be recognised (in border). The timestamps correspond to the start and end times of each action. We can infer 'wash aubergine' with higher accuracy if we know that the tap was turned on before and turned off afterwards. The example also exhibits the variable length of different temporal context windows, due to both variable length of actions and variable distance between actions.*

The main contributions of this chapter are summarised as follows: First, temporal context is formulated as a sequence of actions surrounding the action in a sliding window. Second, a novel framework able to model multi-modal temporal context is proposed. It consists of a transformer encoder that uses vision and audio as input context, and a language model as output context operating on the action labels. Third, state-of-the-art performance is obtained on two datasets: (i) the large-scale egocentric dataset EPIC-KITCHENS, outperforming high-capacity end-to-end transformer models; and (ii) the EGTEA dataset. Finally, an ablation study analysing the importance of the extent of the temporal context and of the various modalities is included. The results presented in this chapter signify: (i) the advantages of multiple temporal context, where audio and language provide complementary benefits to visual temporal context, (ii) the capacity of language models to improve the predictions of audio-visual networks by filtering out improbable sequences, and (iii) that additional supervision from the neighbouring actions in the temporal context can enhance the recognition of the action of interest.[1]

5.1 Multi-modal Temporal Context Network (MTCN)

Given a long video, we predict the action in a video segment by leveraging the *temporal context* around it. We define the temporal context as the sequence of neighbouring actions that precede and succeed the action, and aim to leverage that information, when useful, through learnt attention. We utilise multi-modal temporal context both at the input and the output of our model. An audio-visual transformer ingests a temporally-ordered sequence of visual inputs, along with the corresponding sequence of auditory inputs. We use modality-independent positional encodings as well as modality-specific encodings. The language model, acting on

[1]The Python code for the proposed model and the pre-trained model on EPIC-KITCHENS-100 as well as the extracted audio-visual features for all datasets are available at https://github.com/ekazakos/MTCN.

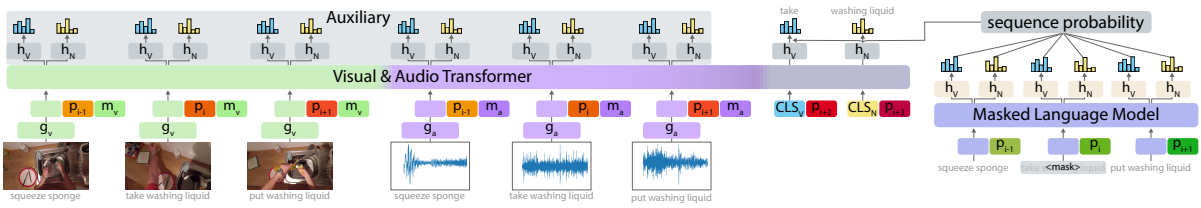


Figure 5.2: *The Multi-modal Temporal Context Network (MTCN). An encoding layer takes as input visual and auditory features and produces as output visual and auditory tokens by first projecting the features to a lower dimension using fully-connected layers, g_v and g_a, followed by tagging them with positional and modality encodings. Verb and noun learnable classification tokens with independent positional encodings are appended in the sequence. An audio-visual transformer encoder attends to the sequence. The verb and noun summary embeddings predict the action at the centre of the window ('take washing liquid') using a two-head classifier. The classifier also predicts the sequence of actions from the audio-visual tokens to train an auxiliary loss that enhances the prediction of the centre action using additional supervision from the temporal context. A Masked Language Model (also a transformer architecture) that takes as input the sequence of action labels, operates on the concatenation of verb and noun embeddings using two separate word-embedding layers. The language model is trained by randomly masking any ground-truth action and predicting it. Given the predictions of the audio-visual transformer for the whole sequence, a beam search computes the K most probable sequences. The language model filters out improbable sequences by computing the probability of each sequence.*

the output of the transformer learns the prior temporal context of actions, *i.e.* the probability of the sequence of actions, using a learnt text embedding space.

Inspired by similar approaches [202], and instead of using a single summary embedding as in prior works for image [43] and action classification [10], the audio-visual model utilises two separate summary embeddings to attend to the action (*i.e.* verb) class and the object (*i.e.* noun) class. This allows the model to attend independently to the temporal context of verbs vs objects. For example, the object is likely to be the same in neighbouring actions while the possible sequences of verbs can be independent of objects (*e.g.* 'take' $\rightarrow$ 'put'). Each summary embedding uses a different learnt classification token, and the classifier predicts a verb and a noun from the summary embeddings. The predictions of the audio-visual transformer are then enhanced by filtering out improbable sequences using the language model. We term the proposed model Multi-modal Temporal Context Network (MTCN).

In the next three subsections, we detail the architectural components of MTCN, as well as our training strategy and inference procedure. An overview of MTCN is depicted in Fig. 5.2.

5.1.1 Audio-visual Transformer

Let $X_v \in \mathbb{R}^{w \times d_v}$ be the sequence of visual inputs from a video, and $X_a \in \mathbb{R}^{w \times d_a}$ the corresponding audio inputs (video and audio inputs/features are synchronised, therefore they have the same length w), for w consecutive actions in the video (*i.e.* the temporal context window), with d_v, d_a being the input dimensions of the two modalities respectively. X_v and X_a correspond to features extracted from visual and auditory networks, respectively. Our temporal window is centred around an action b_i with surrounding action segments, excluding any background frames. That is, each action b_j within the window, $i - \frac{w-1}{2} \leq j \leq i + \frac{w-1}{2}$ is part of the transformer's input.

Encoding layer. Our model first projects the inputs X_v, X_a to a lower dimension D and tags each with positional and modality encodings. Then, an audio-visual encoder performs self-attention on the sequence to aggregate relevant audio-visual temporal context from neighbouring actions. Because all self-attention operations in a transformer are permutation invariant, we use positional encodings to retain information about the ordering of actions in the sequence. We use w learnt absolute positional encodings, shared between audio-visual features to model corresponding inputs from the two modalities. Modality encodings, m_v, $m_a \in \mathbb{R}^D$, are learnt vectors added to discriminate between audio and visual tokens.

A classifier predicts the action b_i, using two summary embeddings, acting on the learnt verb/noun tokens. We use the standard approach of appending learnable classification tokens to the end of the sequence but use two tokens, one for verbs and one for nouns, denoted as CLS_V, $\text{CLS}_\text{N} \in \mathbb{R}^D$, with unique positional encodings. To summarise, the encoding layer transforms the inputs X_v and X_a as follows:

$$X^e_{v_j} = g_v(X_{v_j}) + p_j + m_v \qquad X^e_{a_j} = g_a(X_{a_j}) + p_j + m_a \qquad \forall j \in [1, ..., w] \tag{5.1}$$

$$\text{CLS}^e_\text{V} = \text{CLS}_\text{V} + p_{w+1} \qquad \text{CLS}^e_\text{N} = \text{CLS}_\text{N} + p_{w+2} \tag{5.2}$$

$$X^e = [X^e_v; X^e_a; \text{CLS}^e_\text{V}; \text{CLS}^e_\text{N}], \tag{5.3}$$

where $[;]$ denotes input concatenation and $p \in \mathbb{R}^{(w+2) \times D}$ are the positional encodings. $g_v(\cdot) : \mathbb{R}^{d_v} \mapsto \mathbb{R}^D$ and $g_a(\cdot) : \mathbb{R}^{d_a} \mapsto \mathbb{R}^D$, are fully-connected layers projecting the visual and audio features, respectively, to a lower dimension D. The input to the transformer is $X^e \in \mathbb{R}^{(2w+2) \times D}$. In Subsec. 5.3.4, we compare the absolute positional encodings with relative [162] and Fourier feature positional encodings [84].

Transformer and classifier. We use a transformer encoder $f(\cdot)$ to process sequential audio-visual inputs, $Z = f(X^e)$. We share the weights of the transformer encoder layer-wise. In

Subsec. 5.3.4, we compare this to a version without weight sharing. Weight sharing uses $2.7\times$ fewer parameters with comparable results. A two-head classifier $h(\cdot)$ for verbs and nouns then predicts the sequence of w actions from both the transformed visual and audio tokens $\hat{Y} = h(Z_{1:2w})$, and the action b_i from the summary embeddings $\hat{y} = h(Z_{2w:2w+2})$.

Loss function. Recall that our goal is to classify b_i, the action localised at the centre of our temporal context. Nevertheless, we can leverage the ground-truth of neighbouring actions within w for additional supervision to train the audio-visual transformer. Our loss is composed of two terms, the main loss for training the model to classify the action at the centre of our temporal context $i = \frac{w}{2}$, and an auxiliary loss to predict all actions in the sequence:

$$L_{\text{m}} = CE(Y_i^{\text{V}}, \hat{y}^{\text{V}}) + CE(Y_i^{\text{N}}, \hat{y}^{\text{N}}) \tag{5.4}$$

$$L_{\text{a}} = \sum_{j=1}^{w} \left(CE(Y_j^{\text{V}}, \hat{Y}_j^{\text{V}}) + CE(Y_j^{\text{V}}, \hat{Y}_{|w|+j}^{\text{V}}) + CE(Y_j^{\text{N}}, \hat{Y}_j^{\text{N}}) + CE(Y_j^{\text{N}}, \hat{Y}_{|w|+j}^{\text{N}}) \right) \tag{5.5}$$

$$loss = \beta L_{\text{m}} + (1-\beta) L_{\text{a}}, \tag{5.6}$$

where $CE()$ is a cross-entropy loss, and $Y = (Y_1, ..., Y_w)$ is the ground-truth of the sequence, while $\hat{Y}_1, ..., \hat{Y}_w$ and $\hat{Y}_{w+1}, ..., \hat{Y}_{2w}$ correspond to predictions from the transformed visual and auditory inputs respectively. We use β to weight the importance of the auxiliary loss.

5.1.2 Language model

In addition to input context from the visual and audio domains, we introduce output context using a language model. Language modelling, commonly applied to predict the probability of a sequence of *words*, is a fundamental task in NLP research [21, 40, 144, 209]. Our language model predicts the probability of a sequence of *actions*. We use the language model to improve the predictions of the audio-visual transformer by filtering out improbable sequences.

We adopt the popular Masked Language Model (MLM), introduced in BERT [40]. Therefore, our language model employs a transformer architecture too. We train this model independently from the audio-visual transformer. Specifically, given a sequence of actions $Y = (Y_{i-\frac{w-1}{2}}, ..., Y_{i+\frac{w-1}{2}})$, we randomly mask any action Y_j and train the model to predict it. For example, an input sequence to the model (for $w = 5$) would be: ('turn on tap', 'wash hands', <MASK>, 'pick up towel', 'dry hands'). Without any visual or audio input, the language model is tasked to learn a high prior probability for 'turn off tap', which is masked. Note that the model is trained using the ground-truth sequence of actions. For input representation, we split the action into verb and noun tokens (*e.g.* 'dry hands' $\rightarrow$ 'dry' and 'hand'),

convert them to one-hot vectors and input them into separate trainable word-embedding layers. In preliminary experiments we found that learning a word embedding outperforms pretrained embeddings. MLM takes as input the concatenation of verb and noun embeddings. The outputs are scores for verb and noun classes using a two-head classifier, and the model is trained with a cross-entropy loss per output.

5.1.3 Inference

Given the scores of the sequence, $\hat{S} = (\hat{y}_{i-\frac{w-1}{2}}, ..., \hat{y}_{i+\frac{w-1}{2}})$, from the audio-visual model (these are temporally ordered predictions from the summary embeddings, and thus different from $\hat{Y}$), we apply a beam-search of size K to find the K most probable action sequences $\hat{S}_b$. Therefore, $\hat{S}_b$ is of size $K \times w$. In inference, the trained language model takes as input $\hat{S}_b$, *i.e.* it operates on sequences predicted from the audio-visual transformer.

For each sequence l, we calculate the probability of the sequence $p_{LM}(\hat{S}_b^l)$ from the language model by utilising the method introduced in [164]. We mask actions, one at a time, and predict their probability. $p_{LM}(\hat{S}_b^l)$ is the sum of log probabilities of all actions in l. We also calculate the probability $p_{AV}(\hat{S}_b^l)$ by summing the log probabilities of all predicted actions in l by the audio-visual model. Then, we combine the probabilities of sequences of the audio-visual and language models:

$$p(\hat{S}_b^l) = \lambda p_{LM}(\hat{S}_b^l) + (1 - \lambda) p_{AV}(\hat{S}_b^l). \tag{5.7}$$

Sequences are then sorted in descending order by $p(\hat{S}_b^l)$. The score of the centre action from the sequence with the highest probability is used as the final prediction.

5.2 Experimental Setup

5.2.1 Datasets

We evaluate MTCN on EPIC-KITCHENS-100 [36] and EGTEA [107]. As EPIC-KITCHENS-100 contains unscripted daily activities, it offers naturally variable sequences of actions. There are on average 129 actions per video (std 163 actions/video and maximum of 940 actions/video). This makes the dataset ideal for exploring temporal context. The length of sequences of $w = 9$ actions (this is our default window length) is 34.4 seconds of video on average with an std of 27.8 seconds (minimum of 3.4 seconds to a maximum of 720.2 seconds). For completeness, we also perform experiments separately on EPIC-KITCHENS-55 [35]. Regarding EGTEA,

although it does not contain audio, it has sequential actions annotated within long videos, and we use it to train part of our approach (vision and language).

5.2.2 Implementation details

Visual features. For EPIC-KITCHENS, we extract visual features with SlowFast [47], using the public model and code from [36]. We first train the model with slightly different hyperparameters, where we sample a clip of 2s from an action segment, do not freeze batch normalisation layers, and warm-up training during the first epoch starting from a learning rate of 0.001. We note that this gave us better performance. All unspecified hyperparameters remain unchanged. For feature extraction, 10 clips of 1s each are uniformly sampled for each action segment, with a center crop per clip. The resulting features have a dimensionality of $d_v = 2304$. The SlowFast visual features are used for all the results in this paper, apart for the comparison with the state of the art in Tables 5.10 and 5.11 where we additionally experiment with features from Mformer-HR [143]. These are extracted from the EPIC-KITCHENS pretrained model using a single crop per clip. The resulting features have a dimensionality of d_v=768.

For EGTEA, we train SlowFast [47] using the EPIC-KITCHENS pre-trained model, by sampling a clip of 2s from an action segment similar to EPIC-KITCHENS. We use a learning rate of 0.001, no warm-up, and we keep the batch normalisation layers frozen. All unspecified hyperparameters remain unchanged. For feature extraction, we follow the same procedure as EPIC-KITCHENS, except that we use clips of 2s rather than 1s.

Auditory features. We use Auditory SlowFast (Ch. 4) for audio feature extraction when present. Similarly to the visual features, we extract 10 clips of 1s each uniformly spaced for each action segment, with average pooling and concatenation of the features from the Slow and Fast streams, and the resulting features have the same dimensionality, $d_a = 2304$.

Architectural details. Both the audio-visual transformer encoder and the language model consist of 4 layers with shared weights, 8 attention heads and a hidden unit dimension of 512. In the audio-visual transformer, positional/modality encodings as well as verb/noun tokens have also dimensionality $D = 512$ and are initialised to $\mathcal{N}(0, 0.001)$. The layers $g_v(\cdot)$ and $g_a(\cdot)$ reduce the features to the common dimension $D = 512$. In the encoding layer, dropout is applied at the inputs of $g_v(\cdot)$ and $g_a(\cdot)$ as well as at X^e. In the language model, both verb and noun word-embedding layers have a dimension of 256, and positional encodings have a dimension of 512, while dropout is applied to its inputs.

Scheduled sampling. We modify the scheduled sampling from [16] to train the language model. At each training iteration, we randomly mask an action, predict it, and replace the

corresponding ground-truth with the prediction.

Train / Val details. For EPIC-KITCHENS-100, the audio-visual transformer is trained using SGD, a batch size of 32 and a learning rate of 0.01 for 100 epochs. Learning rate is decayed by a factor of 0.1 at epochs 50 and 75. In the loss function, we set $\beta = 0.9$. For regularisation, a weight decay of 0.0005 is used and a dropout 0.5 and 0.1 for the encoding layers and transformer layers respectively. We use mixup data augmentation [214] with $\alpha = 0.2$. The language model is trained for the same number of epochs with a batch size of 64, Adam optimiser with initial learning rate of 0.001 and the learning rate is decreased by a factor of 0.1 when validation accuracy saturates for over 10 epochs. The values of dropout and weight decay are the same as those of the audio-visual model. For inference, we tune λ in Eq. 5.7 on the validation set with grid-search from the set $\{0, 0.05, 0.1, 0.15, 0.2, 0.25, 0.3\}$, and we use a beam size of $K = 10$. For training the audio-visual transformer, we randomly sample 1 out of 10 features per action. For testing, we feed all 10 features per action to the transformer and we share the positional encoding corresponding to an action with all 10 features. We also tried single feature per action followed by averaging 10 predictions but observed no difference in performance.

For EPIC-KITCHENS-55, where there is no official validation set, we used the same hyperparameters as for EPIC-KITCHENS-100 for training both the audio-visual transformer and the language model. Accordingly, we trained the models for 64 epochs which was the epoch that the audio-visual transformer obtained the highest accuracy on EPIC-KITCHENS-100, and dropped the learning rate only at epoch 50. We also followed the same testing procedure used for EPIC-KITCHENS-100.

Remember that for EGTEA we train only vision and language as EGTEA does not contain audio. First, as there is no audio input to the transformer, we do not use modality encodings either. Second, following previous methods [80, 108, 118, 128, 178, 179] that train using a single head for actions and report only action accuracy, we use a single summary embedding for actions, rather than verb/noun embeddings. Accordingly, the language model utilises a single word-embedding for actions, with a dimension of 512. For training the visual-only transformer, we use a learning rate of 0.001, train the model for 50 epochs and decay the learning rate at epochs 25 and 38, while keeping all other hyperparameters unchanged. We use same hyperparameters for the language model. For evaluation, differently than EPIC-KITCHENS, we average the predictions of the 10 clips per action, rather than feeding all 10 clips in the transformer, as we experimentally found that this option provides better results for EGTEA.

5.2.3 Evaluation metrics

For EPIC-KITCHENS-100, similarly to Ch. 4, we follow [36] and report top-1 and top-5 accuracy for the validation and test sets separately. We also follow [36] and report results for two subsets within val/test: unseen participants and tail classes. For EPIC-KITCHENS-55, we report top-1 and top-5 accuracy solely for the Seen split (S1). For EGTEA, we follow [89, 118, 128] and report top-1 accuracy and mean class accuracy using the first train/test split. In contrast to top-1 accuracy, mean class accuracy accounts for the size of each class, and thus it's more suitable for imbalanced datasets.

5.3 Results

This section is organised as follows. First, we assess the importance of temporal context extent in Subsec. 5.3.1. Then, we perform an ablation analysis of modalities and the auxiliary loss in Subsec. 5.3.2. In Subsec. 5.3.3, we assess the statistical significance of the language model and compare the performance of MTCN to variants using baseline language models. In Subsec. 5.3.4, we analyse architectural components of MTCN, followed by testing the performance of MTCN in an online setting in Subsec. 5.3.5. Finally, we compare MTCN with SOTA in Subsec. 5.3.6. For the analysis of the model and ablations in Subsecs. 5.3.1-5.3.5, we use the validation set of EPIC-KITCHENS-100, except from Subsec. 5.3.1 where we additionally use EGTEA.

5.3.1 Analysis of temporal context length

We first analyse the importance of the temporal context extent by varying the size of w, *i.e.* the length of window of actions that our model observes, for both EPIC-KITCHENS-100 and EGTEA. For EPIC-KITCHENS, we perform this analysis both for MTCN containing all modalities as shown in Table 5.1, as well as for the language model as shown in Table 5.2. Varying the length of the window has a big impact on the model's accuracy showcasing that MTCN successfully utilises temporal context. Using temporal context outperforms $w = 1$, *i.e.* no temporal context. As the window length increases the performance also increases. Overall top-1 accuracy increases up to $w = 9$ while top-5 up to a window $w = 5$. Performance on unseen participants and tail classes also increases up to $w = 9$.

In Table 5.2, experiments are conducted by masking the centre action and measuring how well the model predicts it. We use ground-truth for the other actions, as in this experiment we are interested in the maximum possible performance of the language model, *i.e.* assuming correct predictions from the audio-visual model. Here $w = 1$ corresponds to a language model

	Overall						Unseen Participants			Tail-classes		
	Top-1 Accuracy (%)			Top-5 Accuracy (%)			Top-1 Accuracy (%)			Top-1 Accuracy (%)		
w	Verb	Noun	Action	Verb	Noun	Action	Verb	Noun	Action	Verb	Noun	Action
1	67.93	52.29	41.30	90.53	76.47	61.52	61.13	44.60	32.58	42.05	27.42	21.48
3	69.80	55.24	43.52	91.30	79.04	63.25	61.41	46.48	33.71	39.09	32.58	23.06
5	70.38	56.16	45.13	**91.67**	**79.47**	**64.14**	61.97	46.95	34.74	**43.12**	32.53	24.54
7	70.43	56.19	45.01	91.23	79.13	63.52	62.63	**47.14**	34.84	41.31	32.79	24.12
9	**70.60**	**56.26**	**45.48**	91.14	79.06	63.06	**63.76**	**47.14**	**35.87**	41.36	32.84	**24.70**
11	70.55	55.74	44.68	91.18	79.23	63.02	62.91	46.57	34.74	41.82	**33.58**	24.44

Table 5.1: *Analysis of temporal context extent for MTCN on the validation set of EPIC-KITCHENS-100. As the length of the window, w, increases the accuracy of the model increases as well. For top-1 accuracy best performance is obtained using a window of 9 actions, while for top-5 accuracy optimal results are achieved with $w = 5$. For unseen participants and tail classes a window of 9 actions provides the highest performance gains too.*

	Overall		
	Top-1 Accuracy (%)		
w	Verb	Noun	Action
1	19.32	3.74	0.82
3	38.08	45.56	23.85
5	42.15	**50.36**	29.48
7	42.93	50.35	**29.91**
9	**43.06**	50.22	29.41
11	41.89	49.96	29.14

Table 5.2: *Analysis of temporal context extent for the language model using the validation set of EPIC-KITCHENS-100. In this experiment, the centre action is masked and predicted, and the ground-truth is used for the rest of the actions in the temporal context. $w = 1$ corresponds to a random chance baseline. Verb performance improves up to $w = 9$, while for nouns up to $w = 5$ and $w = 7$ for actions.*

that randomly guesses the masked action without any context. When varying the length of the window for the language model, verb performance increases when enlarging the temporal context from $w = 3$ to $w = 9$ while for nouns optimal temporal context is $w = 5$, and $w = 7$ for actions. Interestingly, the language model is performing particularly well for nouns, since the same object is often used for the entire sequence.

Finally, Fig. 5.3 shows the effect of temporal context extent on verb and noun accuracy of the individual modalities as well as of MTCN for EPIC-KITCHENS-100. For verbs, visual modality performance increases up to 7 actions and then decreases while for nouns, it steadily increases up to 11 actions, possibly because larger context is able to resolve ambiguities due to occlusion. Audio can better capture temporal context for verbs than the language model,

showcasing that the progression of sounds conveys more useful information about the action. For nouns, the language model outperforms audio. Moreover, the language model performs better on nouns because of repetitions of the same object in the sequence. MTCN utilising all modalities (A-V-LM) benefits the most from larger temporal context, particularly for verbs. With this analysis completed, we fix w to 9 in all subsequent experiments concerning EPIC-KITCHENS.

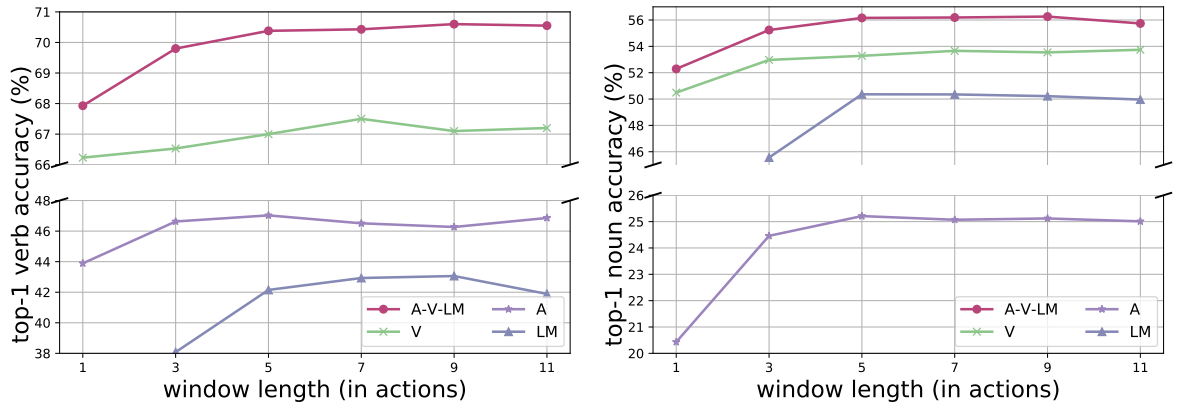


Figure 5.3: *Effect of temporal context on verb (left) and noun (right) accuracy of individual modalities and for MTCN on the validation set of EPIC-KITCHENS-100. V: Visual, A: Audio, LM: Language Model, A-V-LM: MTCN. The Y-axis is cut to emphasise details. Performance of the visual modality increases up to 7 actions for verbs while it continually increases for nouns. Audio performs better than the language model for verbs, while for nouns the opposite is true. Also, note that the language model perform better on nouns than verbs. Finally, MTCN attains the highest gains from larger temporal context, especially for verbs.*

	Visual		Visual + LM	
w	Top-1(%)	Mean Class (%)	Top-1(%)	Mean Class (%)
1	72.26	64.98	72.26	64.98
3	72.55	64.86	**73.59**	65.87
5	**73.10**	**65.42**	73.49	65.57
7	72.26	64.38	73.19	65.31
9	72.55	64.86	73.44	**66.02**

Table 5.3: *Analysis of temporal context extent and ablation of language model in the first test split of EGTEA. The visual-only model achieves optimal top-1 and mean class accuracy for $w = 5$, while when the language model is incorporated best top-1 accuracy is obtained at $w = 3$ and best mean class accuracy at $w = 9$. Note that the language model improves the results, where optimal performance is achieved at a shorter temporal context comparing to the visual-only model.*

We study the effect of the temporal context length both with and without the language model

				Overall						Unseen Participants			Tail-classes		
				Top-1 Accuracy (%)			Top-5 Accuracy (%)			Top-1 Accuracy (%)			Top-1 Accuracy (%)		
V	A	LM	Aux	Verb	Noun	Action	Verb	Noun	Action	Verb	Noun	Action	Verb	Noun	Action
✓	✗	✗	✓	67.10	53.54	41.49	90.62	78.22	62.32	58.87	43.29	30.99	40.97	30.47	22.35
✓	✗	✓	✓	67.84	54.08	42.05	90.63	78.20	60.68	59.06	44.23	31.36	39.77	31.32	22.38
✓	✓	✗	✓	70.23	55.82	45.00	91.13	79.06	**64.58**	63.29	46.38	35.02	**41.76**	32.26	24.41
✓	✓	✓	✗	69.31	55.46	43.81	**91.19**	**79.76**	62.35	61.13	46.01	33.90	39.55	30.74	22.74
✓	✓	✓	✓	**70.60**	**56.26**	**45.48**	91.14	79.06	63.06	**63.76**	**47.14**	**35.87**	41.36	**32.84**	**24.70**
✓	✓	†	✓	71.33	63.56	50.32	92.05	83.16	67.85	62.25	52.68	37.56	41.53	44.16	29.89

Table 5.4: *Ablation on multi-modal temporal context and auxiliary loss on the validation set of EPIC-KITCHENS-100. V: Visual, A: Audio, LM: Language Model, Aux: Auxiliary loss. †: upper bound performance from the addition of the language model using ground-truth temporal context actions as its input. Multi-modal temporal context (*i.e. *audio and language in addition to vision) provides a noticeable boost in performance. Audio is particularly helpful. The language model offer smaller gains; nevertheless it has complementary advantages to audio and the upper bound performance from its addition showcases the space for improvement.*

on the first test split of EGTEA. Results are shown in Table 5.3. For the visual-only model, top-1 accuracy increases when we increase the length of temporal context from $w = 1$ to 5, and optimal results for both top-1 and mean class accuracy are obtained for $w = 5$. When the language model is incorporated top-1 accuracy increases from $w = 1$ to 3 and then decreases while best mean class accuracy is obtained at $w = 9$. These findings showcase that our model successfully utilises context in this dataset as well. Furthermore, the language model is helpful for EGTEA, where it is worth noting that after its addition best performance is obtained at a shorter temporal context, showing that shorter sequences of actions provide a stronger prior in this dataset.

5.3.2 Ablation study

Multi-modality Ablation. We offer an ablation to identify the performance impact of the components of our MTCN. Results are shown in Table 5.4. We remove audio and language modalities from our model's input and output respectively to assess their importance. Multi-modal context is important, as our proposed model enjoys considerable margins compared with the model trained only with visual context (line 1 in Table 5.4). Audio is beneficial, confirming the findings of prior works and previous chapters of this thesis.

Although the language model provides smaller boost in performance than audio, it showcases that it is useful to model prior temporal context at the output level, and that its benefits are complementary to audio. We also include upper bound performance improvement from the addition of the language model, where it takes as input the ground-truth preceding/succeeding

	Top-1 Accuracy (%)			Top-5 Accuracy (%)		
LM	Verb	Noun	Action	Verb	Noun	Action
✗	70.26 ± 0.27	55.70 ± 0.22	44.90 ± 0.20	91.12 ± 0.13	$\mathbf{79.03 \pm 0.18}$	$\mathbf{64.79 \pm 0.17}$
✓	$\mathbf{70.52 \pm 0.25}$	$\mathbf{56.08 \pm 0.21}$	$\mathbf{45.25 \pm 0.18}$	$\mathbf{91.13 \pm 0.13}$	$\mathbf{79.03 \pm 0.18}$	64.58 ± 0.18

Table 5.5: *Mean and standard deviation of multiple runs both w. & w/o language model in the validation set of EPIC-KITCHENS-100. The experiment was conducted by training 10 audio-visual transformers and 10 language models with different random seeds. The employment of the language model improves the efficiency of the model on average with a low std.*

actions rather than the predictions from the audio-visual transformer (last line in Table 5.4), effectively a language infilling problem. These results demonstrate the potential margin for improvement, particularly for nouns and accordingly action accuracy as well as tail classes. In Subsec. 5.3.3, we report statistical significance over multiple runs for the language model, and compare it to alternative baselines.

Auxiliary loss. Also from Table 5.4, training MTCN with auxiliary loss boosts its performance almost in all metrics, confirming that utilising supervision from neighbouring actions can improve the performance of the action of interest, *i.e.* the action at the centre of the window.

5.3.3 Language model

In this section, we assess the statistical significance of our language model and compare the performance of our MTCN to variants using baseline language models.

Statistical significance of LM. We train 10 audio-visual transformers and 10 corresponding language models with different random seeds. Table 5.5 shows the mean and standard deviation top-1 and top-5 accuracy without and with the language model. Utilising the language model improves performance on average with a low std, demonstrating that the improvement from the language model is statistically significant. We further showcase that by conducting T-tests on verb, noun and action top-1 accuracies, obtaining a p-value of $3.6e-2$, $6.0e-4$, $9.7e-4$, respectively.

Baselines comparison. We compare our MTCN that uses a transformer based language model to two baselines, N-gram and Bi-directional LSTM (BiLSTM). For N-gram, we follow a similar procedure to natural language processing. In particular, from all action sequences of length 9 in the training set, we derive the heuristic probability of occurrence of the centre action given the preceding and succeeding actions. We train a BiLSTM with 3 layers and a hidden size of 512. The rest hyperparameters are the same as the transformer encoder.

	Overall						Unseen Participants			Tail-classes		
	Top-1 Accuracy (%)			Top-5 Accuracy (%)			Top-1 Accuracy (%)			Top-1 Accuracy (%)		
Model	Verb	Noun	Action	Verb	Noun	Action	Verb	Noun	Action	Verb	Noun	Action
No LM	70.23	55.82	45.00	91.13	**79.06**	**64.58**	63.29	46.38	35.02	**41.76**	32.26	24.41
N-gram	70.23	55.84	45.02	91.13	**79.06**	64.49	63.29	46.38	35.02	**41.76**	32.26	24.41
BiLSTM	70.57	55.97	45.09	**91.14**	**79.06**	**64.55**	63.29	46.76	35.31	40.68	32.47	24.15
Transformer enc. (proposed)	**70.60**	**56.26**	**45.48**	**91.14**	**79.06**	63.06	**63.76**	**47.14**	**35.87**	41.36	**32.84**	**24.70**

Table 5.6: *Performance of MTCN in the validation set of EPIC-KITCHENS-100 using different language models. For N-gram, we calculate the heuristic probability of occurrence of the centre action given the preceding and succeeding actions. The BiLSTM has 3 hidden layers of size 512 while the rest hyperparameters are identical with the transformer encoder. Adding an N-gram (comparing to not using a language model) does not affect the performance as only a handful of preceding-succeeding sequences from train also appear in val. The proposed transformer-based language model outperforms both N-gram and BiLSTM.*

Results are shown in Table 5.6. It turns out that only a few preceding-succeeding action sequences in the training set also appear in the validation set, resulting in no difference in performance when N-gram is added comparing to not using a language model. Our transformer-based Masked Language Model (MLM) outperforms both the N-gram and BiLSTM, showcasing that it is beneficial to use a deep neural network language model over a heuristic prior and that MLM with transformers outperforms recurrent architectures in this problem.

5.3.4 Analysis of architectural components

In Table 5.7, we explore different number of layers in MTCN, both without and with (layer-wise) weight sharing, and compare each case with a single layer. Note that we use the same number of layers and sharing strategy for both AV and LM. We use bold to indicate best performance within each group rather than overall. Best results are obtained using four layers in most metrics, both without & with weight sharing. These outperform a single layer, demonstrating that is beneficial to use a multi-layered transformer. Although MTCN without weight sharing performs slightly better, our proposed model (with weight sharing) has $2.7\times$ less parameters with only a minor drop in performance.

In Table 5.8, we compare the effect of different types of positional encodings. Specifically, we replace our chosen absolute learnt positional encoding with relative positional encodings [162] and Fourier feature positional encodings [84]. Fourier feature positional encodings replace our learnable absolute positional encodings with non-learnable ones represented as a vector of log-linearly spaced frequency bands up to a maximum frequency. Relative positional encodings replace our absolute positional encodings of the inputs, with positional encodings representing distances between tokens and placed within the self-attention layers. As shown

		Overall						Unseen Participants			Tail-classes		
		Top-1 Accuracy (%)			Top-5 Accuracy (%)			Top-1 Accuracy (%)			Top-1 Accuracy (%)		
Layers	Shared	Verb	Noun	Action	Verb	Noun	Action	Verb	Noun	Action	Verb	Noun	Action
1	-	69.58	55.04	43.71	91.27	79.02	63.96	61.03	46.01	33.33	42.10	32.42	24.09
2	✗	69.49	55.41	43.94	**91.13**	78.96	63.86	**62.72**	46.57	34.37	**41.42**	**32.32**	23.90
4	✗	**71.01**	**56.55**	**46.04**	90.98	**79.28**	**63.97**	62.35	**47.61**	**35.96**	39.94	31.89	**24.22**
6	✗	69.58	55.68	44.89	90.28	78.17	63.37	61.31	45.63	34.46	38.75	32.21	23.90
2	✓	69.82	55.37	43.81	91.09	78.97	**64.26**	61.50	44.32	32.68	**42.05**	32.58	23.74
4	✓	**70.60**	**56.26**	**45.48**	**91.14**	**79.06**	63.06	**63.76**	**47.14**	**35.87**	41.36	**32.84**	**24.70**
6	✓	69.58	55.68	44.89	90.28	78.17	63.37	61.31	45.63	34.46	38.75	32.21	23.90

Table 5.7: *Analysis of performance using different number of layers, both w. and w/o (layer-wise) weight sharing. Results are shown in the validation set of EPIC-KITCHENS-100. The same number of layers and sharing strategy are used both for AV and LM. Bold indicates best performance within each group rather than overall. Optimal performance in most metrics is achieved using 4 layers both with and without weight sharing which outperform using a single layer. MTCN without weight sharing performs only slightly better whereas the proposed model with weight sharing has* $2.7\times$ *less parameters with an insignificant decline in performance.*

	Overall						Unseen Participants			Tail-classes		
	Top-1 Accuracy (%)			Top-5 Accuracy (%)			Top-1 Accuracy (%)			Top-1 Accuracy (%)		
Pos. enc.	Verb	Noun	Action	Verb	Noun	Action	Verb	Noun	Action	Verb	Noun	Action
Fourier PE	69.60	56.13	44.63	90.65	78.86	63.31	63.29	45.63	35.12	38.86	32.53	23.09
Relative PE	70.32	**56.30**	45.37	91.01	**79.35**	**64.04**	61.41	46.67	33.99	**41.42**	**33.74**	**24.96**
Absolute PE (Proposed)	**70.60**	56.26	**45.48**	**91.14**	79.06	63.06	**63.76**	**47.14**	**35.87**	41.36	32.84	24.70

Table 5.8: *Comparison of different positional encodings (PE) using the validation set of EPIC-KITCHENS-100. Fourier feature encodings are non-learnable absolute positional encodings represented as vectors of log-linearly spaced frequency bands. Relative learnable positional encodings, as opposed to absolute ones, represent distances between tokens and are applied within the self-attention layers. The proposed absolute positional encodings outperform Fourier feature positional encodings, and in comparison to relative positional encodings, the proposed ones are better in top-1 accuracy and unseen participants while relative positional encodings perform better in top-5 accuracy and tail classes. In general, the differences between different types of positional encodings are insignificant.*

in the table our proposed absolute learnable positional encodings outperform Fourier feature positional encodings in all metrics (except top-5 action accuracy). Comparing to relative positional encodings, our positional encodings are slightly better in top-1 verbs and actions, as well as in unseen participants, while relative positional encodings perform slightly better in top-5 accuracy and tail classes. Overall, there are no notable differences between the different choices of positional encodings.

	Overall						Unseen Participants			Tail-classes		
	Top-1 Accuracy (%)			Top-5 Accuracy (%)			Top-1 Accuracy (%)			Top-1 Accuracy (%)		
w	Verb	Noun	Action	Verb	Noun	Action	Verb	Noun	Action	Verb	Noun	Action
1	67.93	52.29	41.30	90.53	76.47	61.52	61.13	44.60	**32.58**	**42.05**	27.42	21.48
3	68.42	54.15	42.59	91.20	78.52	61.10	**61.69**	44.41	32.11	40.11	31.26	22.58
5	68.58	54.04	42.75	**90.96**	78.27	62.04	59.81	43.94	32.11	39.49	31.11	22.48
7	**68.88**	**54.31**	**42.96**	90.89	77.87	62.39	61.41	43.38	32.02	40.51	32.00	23.61
9	68.77	54.28	42.77	90.66	77.72	**62.44**	60.38	**45.07**	31.83	40.80	**32.68**	**23.86**
11	67.83	54.04	42.13	90.63	**78.85**	62.10	57.46	43.94	31.46	36.88	30.74	21.96

Table 5.9: *Online action recognition results by varying temporal context length in the validation set of EPIC-KITCHENS-100. In the online setting, only the preceding actions are used to predict the current action. The audio-visual transformer is trained to predict the last action in the sequence instead of the centre one, whereas there is no need to train a new language model; the last action in the sequence is masked and predicted. The performance of the model improves for $w > 1$, with $w = 7$ performing the best for top-1 accuracy, while unseen participants benefit more from shorter temporal context, $w = 3$, and tail-classes from larger, $w = 9$. Compared to the original proposal that utilises both past and future context, performance degrades by 2.5%.*

5.3.5 Online action recognition

The focus of this work is to leverage both past and future context to predict an action, *i.e.* offline action recognition. In this subsection however, we explore the performance of our model in online recognition, *i.e.* using only the preceding actions as context to predict the current action. This approach can be used to recognise actions in an online fashion for streaming videos. For this setting, we train the audio-visual transformer to predict the last action in the sequence instead of the centre one. We do not train a new language model for this task; we simply mask and predict the last action in the sequence instead of the centre one.

Results are demonstrated in Table 5.9 by varying w. Our model can also utilise temporal context in this setting, as performance improves for $w > 1$ with optimal top-1 accuracy at $w = 7$ and optimal accuracy on tail-classes at $w = 9$. Compared to our original proposal that utilises also future context (see Table 5.1), the overall performance degrades – best top-1 action performance using past context solely is 42.96% compared to 45.48% using surrounding context, indicating that leveraging future context is beneficial.

5.3.6 Comparison with the state of the art

EPIC-KITCHENS-100. We compare our approach with the state-of-the-art (SOTA) approaches on the validation set of EPIC-KITCHENS-100 as shown in Table 5.10. We first focus

5.3 Results

	Overall						Unseen Participants			Tail-classes		
	Top-1 Accuracy (%)			Top-5 Accuracy (%)			Top-1 Accuracy (%)			Top-1 Accuracy (%)		
Model	Verb	Noun	Action	Verb	Noun	Action	Verb	Noun	Action	Verb	Noun	Action
TSN [194]	60.2	46.0	33.2	89.6	72.9	55.1	47.4	38.0	23.5	30.5	19.4	13.9
TBN (Ch. 3)	66.0	47.2	36.7	90.5	73.8	57.7	59.4	38.2	29.5	39.1	24.8	19.1
TSM [110]	67.9	49.0	38.3	91.0	75.0	60.4	58.7	39.6	29.5	36.6	23.4	17.6
SlowFast [47]	67.5	51.5	40.3	90.9	75.6	60.0	58.2	43.9	31.3	39.4	24.50	20.7
ViViT-L/16x2 [10]	66.4	56.8	44.0	-	-	-	-	-	-	-	-	-
X-ViT (16x) [22]	68.7	56.4	44.3	-	-	-	-	-	-	-	-	-
Mformer-HR [143]	67.0	58.5	44.5	-	-	-	-	-	-	-	-	-
MBT [133]	64.8	58.0	43.4	-	-	-	-	-	-	-	-	-
MTCN - v.f. SlowFast [47]	70.6	56.3	45.5	**91.1**	79.1	63.1	**63.8**	47.1	35.9	41.4	32.8	24.7
MTCN - v.f. Mformer-HR [143]	**70.7**	**62.1**	**49.6**	90.7	**83.1**	**68.6**	63.7	**50.9**	**38.9**	**41.9**	**39.2**	**27.7**

Table 5.10: *Comparison with SOTA on the validation set of EPIC-KITCHENS-100. For this comparison, MTCN is trained and evaluated using two different visual features ('v.f.'): from SlowFast [47] as well as from Mformer-HR [143]. The authors of [10, 22, 133, 143] published only top-1 accuracy. Using the SlowFast visual features, MTCN outperforms TBN that is also audio-visual and SlowFast that uses the same visual features. MTCN outperforms the transformer-based approaches [10, 22, 133, 143] in top-1 verbs and actions too, and performs comparably to [10, 22] in top-1 nouns. By employing the visual features of Mformer-HR, MTCN surpasses the performance of all transformer-based approaches, including the audio-visual method in [133], by considerable margins.*

	Overall						Unseen Participants			Tail-classes		
	Top-1 Accuracy (%)			Top-5 Accuracy (%)			Top-1 Accuracy (%)			Top-1 Accuracy (%)		
Model	Verb	Noun	Action	Verb	Noun	Action	Verb	Noun	Action	Verb	Noun	Action
TSN [194]	59.03	46.78	33.57	87.55	72.10	53.89	53.11	42.02	27.37	26.23	14.73	11.43
TBN (Ch. 3)	62.72	47.59	35.48	88.77	73.08	56.34	56.69	43.65	29.27	30.97	19.52	14.10
TSM [110]	65.32	47.80	37.39	89.16	73.95	57.89	59.68	42.51	30.61	30.03	16.96	13.45
SlowFast [47]	64.63	48.62	37.40	88.61	73.84	56.13	57.54	41.24	28.71	33.64	21.27	16.05
Ego-Exo [109]	66.07	51.51	39.98	**89.39**	76.31	60.68	59.83	45.50	32.63	33.92	22.91	16.96
MTCN - v.f. SlowFast [47]	**68.44**	55.41	44.10	88.74	78.04	61.69	**61.82**	47.62	34.94	34.77	28.60	20.45
MTCN - v.f. Mformer-HR [143]	67.88	**60.02**	**46.83**	88.69	**81.84**	**66.48**	61.07	**55.21**	**38.98**	**35.16**	**34.70**	**22.79**

Table 5.11: *Comparison with SOTA on the test set of EPIC-KITCHENS-100 using two different visual features ('v.f.'), similarly to Table 5.10. MTCN outperforms all approaches, including [109] that distills egocentric signals from larger-scale third-person datasets.*

on MTCN trained with SlowFast visual features. MTCN significantly outperforms convolutional approaches ([47, 110, 194] and TBN from Ch. 3). We outperform the audio-visual TBN by 8% on top-1 actions, and SlowFast [47] by 5% (using the same visual features). We also outperform very recent transformer-based approaches [10, 22, 133, 143] on top-1 verbs and actions, reporting published results (the authors of these approaches provided only top-1 accuracy). Note, that MTCN consists of a lightweight transformer that operates on pre-extracted features, while [10, 22, 133, 143] are high-capacity models trained end-to-end. Nevertheless, these approaches perform better than MTCN on top-1 nouns (when the latter is trained with

	Top-1 Accuracy (%)			Top-5 Accuracy (%)		
Model	Verb	Noun	Action	Verb	Noun	Action
LFB [203]	60.0	45.0	32.7	88.4	71.8	55.3
G-Blend [197]	66.7	48.5	37.1	88.9	71.4	56.2
AV-SlowFast [207]	65.7	46.4	35.9	89.5	71.7	57.8
Ego-Exo [109]	65.97	47.99	37.09	**90.32**	70.72	56.32
MTCN (Ours)	**69.12**	**51.30**	**40.77**	90.18	**73.53**	**59.15**

Table 5.12: *Comparison with SOTA on the Seen split (S1) of EPIC-KITCHENS-55. MTCN outperforms all other approaches, where [197] and [207] are audio-visual and [203] was one of the first works to employ temporal context.*

SlowFast visual features) due to the enhanced object recognition performance of their ViT backbone [43] and its large-scale pre-training. Interestingly, MTCN performs comparably to [10, 22] in top-1 nouns, demonstrating that it has competitive object recognition performance even when trained with visual features extracted from models that were pre-trained on smaller scale and less variable datasets.

When we employ visual features from Mformer-HR [143], MTCN improves over all transformer-based approaches, including audio-visual fusion [133], by 3.5% on top-1 nouns and 5% on actions. Although as explained above, the boost on nouns is due to the large-scale pre-trained backbone of Mformer-HR, these results further demonstrate the potential to boost other methods and features by utilising multi-modal temporal context which therefore has complementary benefits to training large transformer models end-to-end.

Unfortunately, most works do not report on the leaderboard test set. In Table 5.11, we provide results on the test set comparing our model to baselines from [36] (*i.e.* the convolutional approaches [47, 110, 194] and TBN from Ch. 3), as well as Ego-Exo [109] that distills knowledge from a much larger training set. MTCN outperforms all other methods, including the competitive method of [109], showcasing that multi-modal temporal context from consecutive actions is more beneficial than pretraining large models (ResNet101) using egocentric signals from third-person datasets.

EPIC-KITCHENS-55. We also compare our model to works that report on the earlier version of this dataset, namely EPIC-KITCHENS-55 [35] in Table 5.12. We compare MTCN with two audio-visual approaches [197] and [207], as well as [203] which was one of the first works to utilise temporal context. We also report the performance of [109] which evaluates their method on both EPIC-KITCHENS-55 and EPIC-KITCHENS-100. Our MTCN outperforms all approaches.

EGTEA. We compare MTCN without audio with the state of the art on EGTEA, in Table 5.13. This model uses $w = 3$ which provided the best top-1 accuracy (see Table 5.3). Our model

Method	Top-1 (%)	MC(%)
Li et al. [108]	-	53.30
Ego-RNN [178]	62.17	-
Kapidis et al. [89]	68.99	61.40
Lu et al. [118]	68.60	60.54
SlowFast [47]	70.43	61.92
MCN [80]	55.63	-
Min et al. [128]	69.58	62.84
MTCN (V) (Ours)	72.55	64.86
MTCN (V+LM) (Ours)	**73.59**	**65.87**

Table 5.13: *Comparative results on the first test split of EGTEA. MC: Mean Class. This model is trained using only vision and language as EGTEA does not contain audio and with $w = 3$. As there is no audio, modality encodings are not incorporated either, and a single summary embedding for actions is used to be directly comparable with the other methods that utilise a single classification head for actions. MTCN improves over all approaches by 3% in both metrics, and it outperforms SlowFast which was used to extract features. The addition of the language model provides a bigger boost in performance than in EPIC-KITCHENS ('V' vs 'V+LM').*

improves over previous approaches by 3% in both top-1 and mean class accuracy. Note that MTCN outperforms SlowFast [47] which we used to extract features. Importantly, the ablation of the language model in Table 5.13 ('V' vs 'V+LM') showcases that the language model provides a bigger boost in performance in EGTEA than in EPIC-KITCHENS (ablation shown in more detail in Table 5.3). The main reason is because priors (inductive biases) are more helpful for small datasets. This has also been witnessed with the shift to transformers from convolutional architectures, where transformers do not capture the inductive biases of convolutional networks, *i.e.* grid structure and translation invariance; yet when transformers are trained with massive scale data, they can outperform convolutional networks, whereas convolutional networks outperform transformers when both pre-trained on smaller datasets thanks to the priors they are able to model.

5.4 Qualitative Results

In Fig. 5.4 and 5.5, we demonstrate qualitative results selected from the validation set of EPIC-KITCHENS-100. We visualise the auditory (purple) and visual (green) temporal context attention, that is the attention weights from the noun embedding to auditory and visual tokens. One could likewise show the attention from the verb embedding, but we found it to be visually indistinguishable to the one from the noun embedding. As there are 10 clips per action, we average the 10 attention weights per action. In each case, we also show the ground-truth along with the predictions of the model.

Fig. 5.4 illustrates success cases, where the centre action in the sequence is predicted correctly.

5.4 Qualitative Results

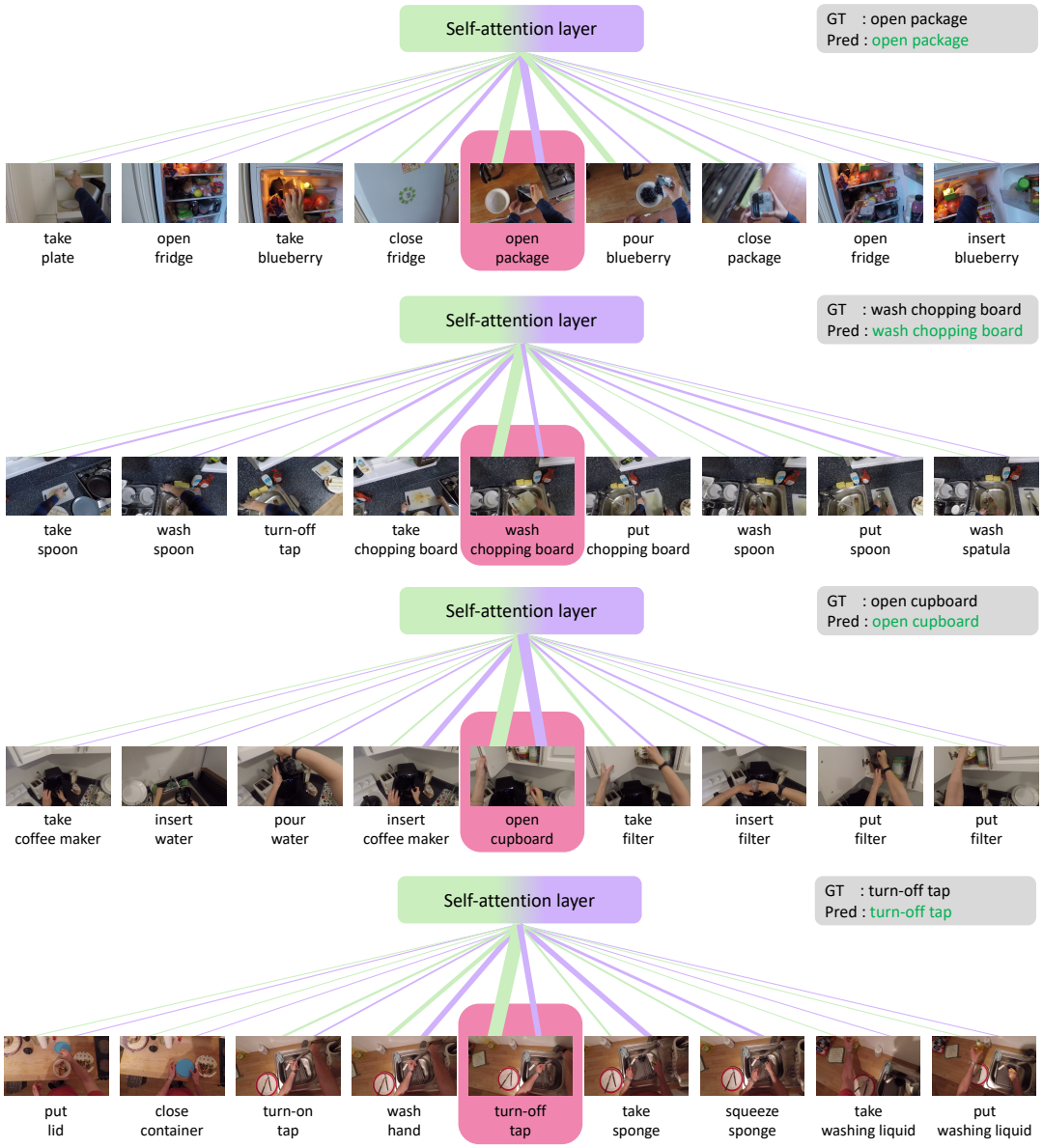


Figure 5.4: *Qualitative results of attention weights along with the ground-truth and the predictions of MTCN using selected examples from the validation set of EPIC-KITCHENS-100. Green and purple edges represent attention weights from the noun embedding to visual and auditory tokens, respectively. Thickness indicates attention weights magnitude to centre (bordered) and temporal context actions. There are 10 clips per action and the 10 attention weights per action are averaged. This figure demonstrates success cases where the model predicts the centre action correctly. For 'open package', actions that contain the blueberries package are more informative to the model while for 'wash chopping board' the model attends especially to actions that include the chopping board. The last two examples demonstrate the importance of auditory temporal context with some actions having high auditory attention in comparison to the visual one.*

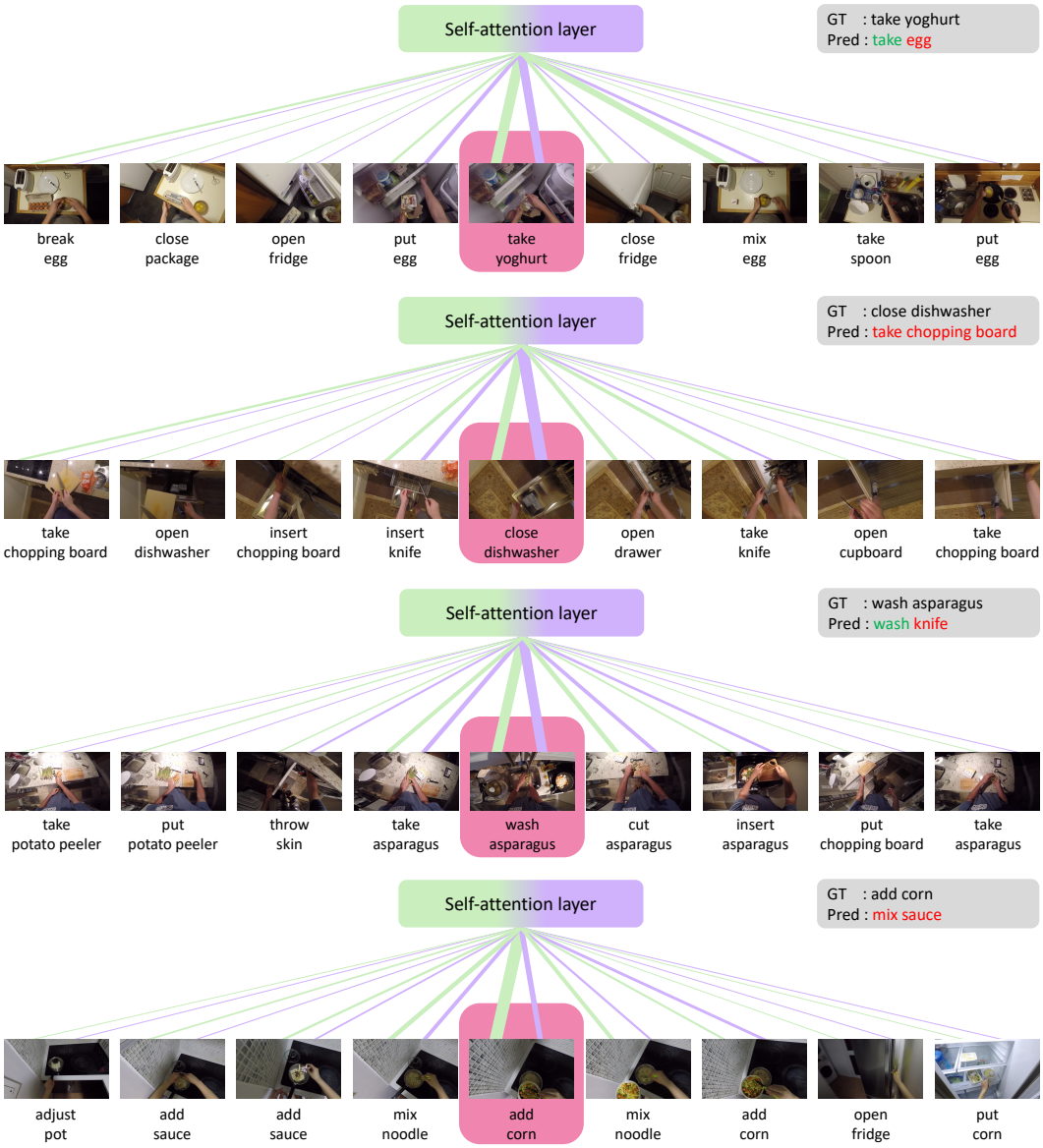


Figure 5.5: *Qualitative results of attention weights along with the ground-truth and the predictions of MTCN using selected examples from the validation set of EPIC-KITCHENS-100. This figure demonstrates failure cases where the model fails to provide a correct prediction of the centre action. For 'take yoghurt', the model has high attention on actions containing the egg and mispredict the noun as egg. 'close dishwasher' is predicted as 'take chopping board' corresponding to the first and last actions in the temporal context. In the third example the noun is wrongly predicted as knife, as a knife appears in several action clips in the sequence. In the fourth example, MTCN predicts the action as 'mix sauce', probably being distracted by the mixing in the fourth action and the sauce in the second and third actions.*

For 'open package', the preceding actions of taking the blueberries and closing the fridge, as well as the subsequent actions of pouring blueberries and closing the package are particularly informative. In the 'wash chopping board' example, the model particularly attends to actions

containing the chopping board. For 'open cupboard', the model has high audio-visual attention to the centre action, and high attention to the audio of the previous action ('insert coffee maker'), showing that at times audio provides useful temporal context. The importance of audio is also apparent in the fourth example.

Fig. 5.5 displays failure cases, where MTCN predicts the centre action incorrectly. For 'take yoghurt', both visual and auditory attention weights are high on actions containing the egg, causing the model to incorrectly predict the noun as egg. A similar source of error in the model results from confusing the centre action with another action in the sequence; in the second example 'close dishwasher' is predicted as 'take chopping board' which corresponds to the first and last actions in the temporal context. The third example is also interesting because while there is no action in the temporal context containing the knife, it is consistently visually present in most of the action clips, causing the model to mispredict the noun as knife. Finally, in the last example, MTCN incorrectly predicts the action as 'mix sauce' possibly by assembling the verb and the noun from different actions in the sequence, *i.e.* 'mix noodle' and 'add sauce'. Overall a common source of error in most of these cases arises from confusing the (centre) action with another action or part of another action (*i.e.* either the verb or the noun) in the sequence.

A more thorough demonstration of qualitative results where the change in attention is illustrated using a sliding window over a sequence of video clips with audio can be watched at https://youtu.be/AkvyX79RVvw?t=112.

5.5 Conclusion

Differently from the previous chapters that treated actions in isolation, this chapter explores a new dimension, that of the temporal context formulated as a sequence of actions, and utilises past and future context to enhance the prediction of the centre action in the sequence. We proposed MTCN, a model that attends to vision and audio as input modality context, and language as output modality context. MTCN is trained with additional supervision from neighbouring actions. The results demonstrated that MTCN successfully utilises temporal context, as when temporal windows of more than one actions were employed the performance improved noticeably. It was illustrated that the temporal context of each modality has a different effect on verbs and nouns. For the visual modality, object recognition accuracy was consistently increasing for longer windows, as larger context can possibly resolve ambiguities due to occlusions. In addition, the language model is more powerful for nouns probably due to repetitions of the same object in the sequence of action labels. MTCN incorporating all modalities benefited the most from larger temporal context particularly for verbs. Furthermore, the importance of audio

and language as additional modality context was demonstrated through an ablation analysis. Finally, MTCN obtained SOTA results on two egocentric video datasets: EPIC-KITCHENS and EGTEA. In EPIC-KITCHENS, MTCN outperformed high-capacity transformer models and its ability to improve over existing methods was demonstrated. A noteworthy find is that the language model provides higher boosts in EGTEA, since priors are more useful in small datasets as has been also shown in works that compare the inductive biases of ConvNets versus the prior-free transformers. This could further urge researchers working with small untrimmed action recognition datasets to incorporate language models as priors in their approaches.

The paradigm investigated in this chapter is that of a temporal horizon of actions that informs a model what took place in the past and what will occur in the future for a better understanding of what is happening in the present. This paradigm could generalise beyond action recognition, essentially in any area of research that is concerned with temporal streams of data where there is progression of events that depend on each other. Of course, temporal context can be considered within events by modelling their long-term temporal dependencies (*e.g.* [179, 194, 217]), but here we focus our discussion on the temporal dependencies across sequential events. First of all, similarly to how temporal context has been utilised for object detection by harnessing detected objects of nearby frames from static cameras [14], action detection could benefit from temporal context too, by leveraging nearby detected actions. An exciting application of temporal context, and importantly for social good, is in sign language recognition where signs in a given temporal proximity depend on each other. In the video domain, temporal context would be well suited in learning from instructional videos as well. Interestingly, multi-modal temporal context could be deployed in all the aforementioned problems; for action detection, auditory and language temporal context can be utilised in the same way as MTCN when available, while for sign language recognition and instructional videos language temporal context can be exploited from the subtitles. The instructional speech as well as the audio from the task being carried out can provide auditory temporal context in instructional videos too.

The endeavour of modelling multi-modal temporal context paved the path for fascinating avenues of future exploration but also exposed vulnerabilities of MTCN that require attention. First and foremost, MTCN depends on the start-end times annotations of the actions in the temporal context. While the annotated start-end times of actions is a common requirement in action recognition evaluation protocols, it is restricting and does not align with real-world settings where these timestamps are not available. An extension to MTCN to address this shortcoming would incorporate an actionness score of neighbouring frames, to distinguish background frames and learn from action sequences in untrimmed videos without temporal bounds during testing. This would bridge the problems of recognition and detection, utilising multi-modality and temporal context. Moreover, the qualitative results in this chapter

have shown that a common source of error in MTCN emerges from confusing the action to be recognised with another action in the sequence which signifies the need for more sophisticated positional encodings able to associate more efficiently the classification tokens with the action of interest.

Intuitively, joint training of vision, audio, and language could enhance the capabilities of the model which could be achieved with an autoregressive encoder-decoder architecture. Motivated by the dominance of Transformers over ConvNets in computer vision and NLP amongst other domains, where their flexibility has been extensively demonstrated in various works, this chapter changed tack from the previous chapters and adopted the transformer paradigm too. Nevertheless, MTCN operates on pre-extracted features and an end-to-end trainable transformer architecture that also leverages multi-modal temporal context would unlock the full potential of Transformers.

CHAPTER

SIX

CONCLUSION

This thesis sets the grounds towards perceiving egocentric actions visually and aurally. This endeavour was triggered by the finding that the audio signal in egocentric videos is strong due to the sounds produced upon interacting with objects and the proximity of the microphone to the ongoing action. The thesis was concerned with three major problems of multi-modal egocentric vision: audio-visual fusion, audio recognition and modelling multi-modal temporal context. While many questions within these areas remain unanswered, the conclusions of the research outcomes of this thesis will hopefully facilitate finding answers to those questions in the future. These are provided below.

1. Audio complements effectively visual modalities through audio-visual fusion but it also performs surprisingly well on its own for fine-grained analysis of objects, actions, and interactions. Thus, the inherent audio-visuality of egocentric videos suggests that audio modelling should be an integral part of fine-grained action recognition methods, particularly if we want to close the gap with the human paradigm.

2. Actions unfold at different speeds for each modality, meaning that modalities are temporally misaligned at the semantic level. Thus, to unlock the full potential of audio-visual integration for action recognition, it is important to employ an asynchronous fusion approach capable of associating the most discriminant moments across modalities, be it a ConvNet or a Transformer. Late fusion should be avoided because it marginalises temporal information before modality fusion and cannot perform asynchronous modality integration.

3. Egocentric action recognition using only audio is a viable avenue of research, but currently this topic has been explored only within this thesis. While there are many possible research

directions, disentangling audio modelling into multiple streams where each focuses on a different auditory characteristic, such as frequency and time, is a fruitful approach.

4. Vision and audio have different learning dynamics – audio overfits faster. Moreover, audio-visual models can memorise the noise in the audio signal that typically originates from irrelevant background audio and silent actions. These are causes of overfitting in audio-visual models which can be alleviated with audio-visual regularisation.

5. Untrimmed videos contain useful temporal context as actions are organised into well-defined sequences of actions that constitute an activity. Allowing a model to observe the actions that took place before and the ones that will happen after the action of interest leads to significant performance gains in comparison to models that treat each action in isolation as the model leverages the progression of the ongoing activity. Multi-modal temporal context offers additional benefits; auditory temporal context is complementary to visual temporal context, and importantly a language model can capture prior temporal context and filter out improbable sequences from the labels of actions.

Findings and limitations

The main findings of this thesis, limitations, as well as possible means of addressing them are presented below.

Ch. 3 offered the first attempt to audio-visual integration for egocentric action recognition. A thorough empirical analysis signified that audio provides complementary information to appearance and motion for various classes, while audio is more informative for actions than objects. Yet, the audio stream is incapable of predicting silent actions, objects with indistinct sounds or materials that do not sound when interacted, which amount to a considerable number of classes.

The study suggested that multi-modal data streams are semantically misaligned and questioned the ability of existing late fusion and synchronised mid-level fusion approaches to capture such asynchronous information. To address this, an end-to-end trainable mid-level fusion architecture of appearance, motion and audio was proposed, coupled with a Temporal Binding Window that allowed fusing modalities asynchronously. A limitation of the proposed approach is that it is incapable of binding the most discriminative temporal samples from each modality but simply uses random temporal modality offsets. A Transformer with early fusion would enable learnable (as opposed to random) modality binding as multi-modal self-attention naturally links multi-modal inputs with learnable attention weights. Importantly, such scheme would allow to systematically assess the significance of asynchronous versus synchronous fusion by evaluating the ratio of asynchronous to synchronous attention weight magnitude.

Ch. 4 took a step towards activity recognition using only audio and introduced a two-stream architecture with a Slow stream for capturing spectral patterns and a Fast stream for temporal modelling. The streams were fused with multi-level lateral connections which performed significantly better than late fusion, denoting the virtue of learning hierarchically integrated auditory representations. The effectiveness of the proposed approach was verified in two different domains, egocentric action recognition using EPIC-KITCHENS and activity recognition using VGG-Sound. Nevertheless, techniques that enable non-symmetric filtering in frequency and time were not investigated in depth. Instead, a simple approach of employing separable convolutions was considered, which though intuitive, didn't prove fruitful. In fact, this is an open problem in auditory recognition and there is ample room for improvements. Harmonic convolutions [215] could be a good starting point.

The second part of Ch. 4 revisited audio-visual fusion and investigated a four-stream architecture that integrates audio-visual Slow and audio-visual Fast streams. A concluding remark from that study is that sound-source localisation is likely to be important for audio-visual action recognition as cross-modal attention fusion performed optimally. Yet, the gap with the baselines was not sufficiently large and the problem necessitates deeper investigation, but it opens a new exciting horizon of research that could bridge audio-visual recognition and source localisation. An interesting question is, can video-level labels (*e.g.* ground-truth actions) be used as a weak supervisory signal to facilitate progress in sound-source localisation as opposed to pixel-level annotations or self-supervision that is the current paradigm [7, 160]?

Ch. 5 argued for the significance of relating actions within a temporal horizon in untrimmed videos and introduced for the first time the notion of multi-modal temporal context from vision, audio and language. A multi-modal Transformer architecture was proposed that attends to the action's temporal context in the data stream using vision and audio, as well as to the temporal context from the labels of neighbouring actions using a language model that estimates the prior probability of the action sequence and re-scores the model's outputs. The results of the investigation suggest that enlarging the temporal context enhances considerably the action recognition capabilities of the model, and that multi-modal context is important too, corroborating the previous chapters that audio is complementary to vision and other studies about the capacity of language models to model prior action context [111, 154].

Yet, the proposed model is far from perfect. While the study showcased the potential usefulness of the language model by estimating its upper bound performance, its contribution was small. Language models need abundant data to learn the prior structure of action sequences using only text, a possibility that can become a reality with the introduction of Ego4D [63], the largest scale egocentric dataset that contains action sequences from untrimmed videos. Moreover, the dependence of the proposed model on the temporal bounds of actions is impractical.

A model that operates on untrimmed video should be able to perform action detection or at least use weak temporal supervision [130]. The localisation capabilities of audio (Ch. 4) and the applicability of language models for action detection [154] could make this happen.

Finally, a limitation of this thesis as a whole is that it has focused on evaluating predicted classes (*e.g.* using top-1/top-5 accuracy), neglecting the possibility of evaluating predicted probabilities. Two popular metrics for predicted probability evaluation are the log loss and the Brier score, where the former is equivalent to the cross-entropy loss that is typically used for training deep networks for classification and the latter is the mean squared error between the predicted probabilities and the ground-truth class encoded as one-hot vector. Evaluating predicted probabilities can be particularly useful for imbalanced datasets, like EPIC-KITCHENS, as it allows calibrating the predicted probabilities and reducing the bias towards the majority classes.

From egocentric vision to egocentric multi-sensory perception

The findings of this thesis signify the multi-modality of egocentric reasoning and will hopefully serve as a starting point for reshaping the mindset of the community from egocentric vision to *egocentric multi-sensory perception*. This is an ambitious yet important goal as multi-modality is the essence of reasoning about interactions with objects and learning object-centric representations – humans interact with objects using all their five senses. Yet, this thesis provided only an initial exploration of this vast area of research, focusing on vision and audio and showcasing the potential benefits of language. Thus, the thesis left ample space for research and some directions for future investigation are provided below.

Unified multi-sensory perception. An exciting avenue of exploration is unified multi-sensory perception, *i.e.* a single architecture for handling all modalities, as it would allow efficiently scaling up to many modalities. One such architecture is the Perceiver [84] that treats modalities simply as concatenated byte arrays and processes them with a single Transformer architecture making no modality-specific assumptions, while tackling the quadratic complexity of Transformers with learnt latent representations. Leveraging the findings of this thesis, a more efficient and bio-plausible unified multi-sensory perception model would be a Perceiver that restricts its attention within Temporal Binding Windows, reducing its complexity further while respecting the neuroscience paradigm regarding modality binding. Towards unified multi-sensory egocentric perception, one could enhance such models with interacting units that reason about hand-object interactions [119] and train them with large-scale datasets [63].

Transfer learning. Apropos audio-visual learning, while the sounds produced from object interactions are strong and discriminative, a model would still benefit from pre-training in

audio-visual sources with a broader collection of audio, such as scenes, events and environmental sounds and transferring the acquired knowledge to audio-visual interaction recognition in egocentric videos. Maybe better transferability could be achieved by extending the paradigm of [109] to distill audio-visual egocentric signals from large-scale audio-visual databases.

Multi-tasking. To enable human-level egocentric perception a model should be able to do much more than simply recognising actions. It should be able to remember what happened in the past, to predict what will happen in the future as well as to understand intuitive physics, *e.g.* 'what will happen to the object if I poke it?'. These capabilities can be enabled by training models in various sophisticated egocentric tasks, and an ideal testbed for this is the Ego4D benchmark suite [63] that introduces tasks like episodic memory, object state change detection and predicting future actions, interacted objects, as well as camera and hand movements. An additional intriguing multi-modal forecasting task that is not considered in Ego4D is to predict the future gaze of the wearer from the present video clip.

More senses. Finally, a longer-term goal is to collect datasets that employ additional human-like sensory inputs such as touch, smell and taste. One such effort is [54] that proposes visual, auditory and tactile synthetic representations. While this is an important step forwards, eventually we need real (as opposed to synthetic) and in-the-wild tactile data recorded with wearable tactile sensors in egocentric datasets along with vision and audition. While this is an arduous procedure we ultimately need to undertake it, because if we desire more intelligent agents we must activate all their senses.

REFERENCES

[1] SpecAugment. https://github.com/zcaceres/spec_augment. 117

[2] S. Adapa. Urban sound tagging using convolutional neural networks. *CoRR*, abs/1909.12699, 2019. 27, 29, 30

[3] T. Afouras, Y. M. Asano, F. Fagan, A. Vedaldi, and F. Metze. Self-supervised object detection from audio-visual correspondence. *CoRR*, abs/2104.06401, 2021. 110, 111

[4] J.-B. Alayrac, A. Recasens, R. Schneider, R. Arandjelović, J. Ramapuram, J. De Fauw, L. Smaira, S. Dieleman, and A. Zisserman. Self-supervised multimodal versatile networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020. 1, 44, 45

[5] H. Alwassel, D. Mahajan, B. Korbar, L. Torresani, B. Ghanem, and D. Tran. Self-supervised learning by cross-modal audio-video clustering. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020. 1, 45, 46

[6] R. Arandjelovic and A. Zisserman. Look, listen and learn. In *Proceedings of International Conference on Computer Vision (ICCV)*, 2017. 1, 43, 44, 45

[7] R. Arandjelović and A. Zisserman. Objects that sound. In *Proceedings of European Conference on Computer Vision (ECCV)*, 2018. 1, 43, 44, 45, 110, 111, 160

[8] R. Arandjelovic, P. Gronat, A. Torii, T. Pajdla, and J. Sivic. Netvlad: Cnn architecture for weakly supervised place recognition. In *Proceedings of Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 19

[9] J. Arevalo, T. Solorio, M. Montes-y Gómez, and F. A. González. Gated multimodal units for information fusion. In *International Conference on Learning Representations Workshops (ICLRW)*, 2017. 90, 91

[10] A. Arnab, M. Dehghani, G. Heigold, C. Sun, M. Lucic, and C. Schmid. Vivit: A video vision transformer. In *Proceedings of International Conference on Computer Vision (ICCV)*, 2021. 8, 14, 15, 16, 17, 136, 150, 151

[11] Y. M. Asano, M. Patrick, C. Rupprecht, and A. Vedaldi. Labelling unlabelled videos from scratch with multi-modal self-supervision. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020. 46

[12] L. J. Ba, J. R. Kiros, and G. E. Hinton. Layer normalization. *CoRR*, abs/1607.06450, 2016. 12

[13] F. Baradel, N. Neverova, C. Wolf, J. Mille, and G. Mori. Object level visual reasoning in videos. In *Proceedings of European Conference on Computer Vision (ECCV)*, 2018. 1, 2, 48, 51, 52

[14] S. Beery, G. Wu, V. Rathod, R. Votel, and J. Huang. Context r-cnn: Long term temporal context for per-camera object detection. In *Proceedings of Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 56, 156

[15] I. Beltagy, M. E. Peters, and A. Cohan. Longformer: The long-document transformer. *CoRR*, abs/2004.05150, 2020. 14

[16] S. Bengio, O. Vinyals, N. Jaitly, and N. Shazeer. Scheduled sampling for sequence prediction with recurrent neural networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2015. 140

[17] Y. Bengio, R. Ducharme, P. Vincent, and C. Janvin. A neural probabilistic language model. *Journal of Machine Learning Research (JMLR)*, 3:1137–1155, 2003. 62, 63

[18] G. Bertasius, H. Wang, and L. Torresani. Is space-time attention all you need for video understanding? In *Proceedings of International Conference on Machine Learning (ICML)*, 2021. 8, 14, 15, 16, 17

[19] G. Bhatt, A. Gupta, A. Arora, and B. Raman. Acoustic features fusion using attentive multi-channel deep architecture . In *CHiME 2018 Workshop on Speech Processing in Everyday Environments*, 2018. 32

[20] A. Brock, S. De, S. L. Smith, and K. Simonyan. High-performance large-scale image recognition without normalization. *CoRR*, abs/2102.06171, 2021. 132

[21] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, and D. Amodei. Language models are few-shot learners. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020. 63, 134, 138

[22] A. Bulat, J. Perez-Rua, S. Sudhakaran, B. Martínez, and G. Tzimiropoulos. Space-time mixing attention for video transformer. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021. 8, 14, 15, 16, 17, 150, 151

[23] M. Caron, I. Misra, J. Mairal, P. Goyal, P. Bojanowski, and A. Joulin. Unsupervised learning of visual features by contrasting cluster assignments. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020. 45

[24] J. Carreira and A. Zisserman. Quo Vadis, action recognition? A new model and the Kinetics dataset. In *Proceedings of Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 2, 8, 11, 14, 18, 19, 21, 24, 49, 50

[25] J. Carreira, E. Noland, A. Banki-Horvath, C. Hillier, and A. Zisserman. A short note about kinetics-600. *CoRR*, abs/1808.01340, 2018. 65

[26] J. Carreira, E. Noland, C. Hillier, and A. Zisserman. A short note on the kinetics-700 human action dataset. *CoRR*, abs/1907.06987, 2019. 66

[27] A. Cartas, P. Radeva, and M. Dimiccoli. Modeling long-term interactions to enhance action recognition. In *Proceedings of International Conference on Pattern Recognition (ICPR)*, 2021. 59

[28] W. Chan, N. Jaitly, Q. Le, and O. Vinyals. Listen, attend and spell: A neural network for large vocabulary conversational speech recognition. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2016. 61, 134

[29] H. Chen, Z. Liu, Z. Liu, P. Zhang, and Y. Yan. Integrating the data augmentation scheme with various classifiers for acoustic scene modeling. Technical report, DCASE2019 Challenge, 2019. 27

[30] H. Chen, W. Xie, A. Vedaldi, and A. Zisserman. VGGSound: A large-scale audio-visual dataset. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020. 26, 73, 100, 116, 118, 120, 121, 122

[31] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton. A simple framework for contrastive learning of visual representations. In *Proceedings of International Conference on Machine Learning (ICML)*, 2020. 45

[32] F. Chollet. Xception: Deep learning with depthwise separable convolutions. In *Proceedings of Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 11

[33] J. S. Chung and A. Zisserman. Out of time: Automated lip sync in the wild. In C.-S. Chen, J. Lu, and K.-K. Ma, editors, *Asian Conference on Computer Vision Workshops (ACCVW)*, 2017. 44

[34] D. Damen, T. Leelasawassuk, O. Haines, A. Calway, and W. Mayol-Cuevas. You-do, i-learn: Discovering task relevant objects and their modes of interaction from multi-user egocentric video. In *Proceedings of British Machine Vision Conference (BMVC)*, 2014. 66, 68

[35] D. Damen, H. Doughty, G. M. Farinella, S. Fidler, A. Furnari, E. Kazakos, D. Moltisanti, J. Munro, T. Perrett, W. Price, and M. Wray. Scaling egocentric vision: The epic-kitchens dataset. In *European Conference on Computer Vision (ECCV)*, 2018. 51, 68, 69, 70, 81, 82, 83, 93, 94, 116, 139, 151

[36] D. Damen, H. Doughty, G. M. Farinella, , A. Furnari, J. Ma, E. Kazakos, D. Moltisanti, J. Munro, T. Perrett, W. Price, and M. Wray. Rescaling egocentric vision: Collection pipeline and challenges for epic-kitchens-100. *International Journal of Computer Vision (IJCV)*, 2021. 1, 15, 51, 68, 70, 71, 100, 116, 118, 119, 120, 129, 139, 140, 142, 151

[37] D. Damen, H. Doughty, G. M. Farinella, S. Fidler, A. Furnari, E. Kazakos, D. Moltisanti, J. Munro, T. Perrett, W. Price, and M. Wray. The epic-kitchens dataset: Collection, challenges and baselines. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 43(11):4125–4141, 2021. doi: 10.1109/TPAMI.2020.2991965. 68, 69, 70, 81, 116

[38] F. De La Torre, J. Hodgins, A. Bargteil, X. Martin, J. Macey, A. Collado, and P. Beltran. Guide to the Carnegie Mellon University Multimodal Activity (CMU-MMAC) database. In *Robotics Institute*, 2008. 66, 68

[39] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and F.-F. Li. Imagenet: A large-scale hierarchical image database. In *Proceedings of Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009. 14, 29

[40] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of North American Chapter of the Association for Computational Linguistics (NAACL)*, 2019. 63, 134, 138

[41] J. Donahue, L. Anne Hendricks, S. Guadarrama, M. Rohrbach, S. Venugopalan, K. Saenko, and T. Darrell. Long-term recurrent convolutional networks for visual recognition and description. In *Proceedings of Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015. 9

[42] M. Dorfer, B. Lehner, H. Eghbal-zadeh, H. Christop, P. Fabian, and W. Gerhard. Acoustic scene classification with fully convolutional neural networks and I-vectors. Technical report, DCASE2018 Challenge, 2018. 27, 28

[43] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *Proceedings of International Conference on Learning Representations (ICLR)*, 2021. 13, 14, 27, 136, 151

[44] Y. Doval and C. Gómez-Rodríguez. Comparing neural-and n-gram-based language models for word segmentation. *Journal of the Association for Information Science and Technology*, 70:187–197, 2019. 63

[45] H. Fan, B. Xiong, K. Mangalam, Y. Li, Z. Yan, J. Malik, and C. Feichtenhofer. Multiscale vision transformers. In *Proceedings of International Conference on Computer Vision (ICCV)*, 2021. 14, 17

[46] C. Feichtenhofer, A. Pinz, and A. Zisserman. Convolutional two-stream network fusion for video action recognition. In *Proceedings of Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 2, 8, 18, 20, 21, 51, 75, 77

[47] C. Feichtenhofer, H. Fan, J. Malik, and K. He. SlowFast Networks for Video Recognition. In *Proceedings of International Conference on Computer Vision (ICCV)*, 2019. 4, 8, 18, 22, 35, 36, 101, 102, 104, 105, 106, 107, 117, 129, 140, 150, 151, 152

[48] P. Flach and M. Kull. Precision-recall-gain curves: Pr analysis done right. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2015. 118

[49] A. Furnari and G. M. Farinella. Rolling-unrolling lstms for action anticipation from first-person video. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 43:4021–4036, 2021. 56, 57

[50] V. Gabeur, C. Sun, K. Alahari, and C. Schmid. Multi-modal transformer for video retrieval. In *Proceedings of European Conference on Computer Vision (ECCV)*, 2020. 59, 60, 61

[51] C. Gan, D. Huang, P. Chen, J. B. Tenenbaum, and A. Torralba. Foley music: Learning to generate music from videos. In *Proceedings of European Conference on Computer Vision (ECCV)*, 2020. 59, 60

[52] Y. Ganin, E. Ustinova, H. Ajakan, P. Germain, H. Larochelle, F. Laviolette, M. March, and V. Lempitsky. Domain-adversarial training of neural networks. *Journal of Machine Learning Research (JMLR)*, 17:1–35, 2016. 55

[53] R. Gao, T.-H. Oh, K. Grauman, and L. Torresani. Listen to look: Action recognition by previewing audio. In *Proceedings of Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 35, 46

[54] R. Gao, Y.-Y. Chang, S. Mall, L. Fei-Fei, and J. Wu. Objectfolder: A dataset of objects with implicit visual, auditory, and tactile representations. In *Proceedings of Conference on Robot Learning (CoRL)*, 2021. 162

[55] J. F. Gemmeke, D. P. W. Ellis, D. Freedman, A. Jansen, W. Lawrence, R. C. Moore, M. Plakal, and M. Ritter. Audio set: An ontology and human-labeled dataset for audio events. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2017. 27, 72

[56] R. Girdhar, D. Ramanan, A. Gupta, J. Sivic, and B. Russell. Actionvlad: Learning spatio-temporal aggregation for action classification. In *Proceedings of Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 8, 18, 19

[57] R. Girdhar, J. Carreira, C. Doersch, and A. Zisserman. Video action transformer network. In *Proceedings of Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 8, 14

[58] G. Gkioxari, R. Girshick, and J. Malik. Contextual action recognition with r*cnn. In *Proceedings of International Conference on Computer Vision (ICCV)*, 2015. 59

[59] Y. Gong, Y.-A. Chung, and J. Glass. Psla: Improving audio tagging with pretraining, sampling, labeling, and aggregation. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 29:3292–3306, 2021. 27, 28, 29, 30, 72

[60] Y. Gong, Y.-A. Chung, and J. Glass. AST: Audio Spectrogram Transformer. In *Proceedings of Interspeech*, 2021. 27, 28, 29, 30, 37, 72

[61] M. A. Goodale and A. D. Milner. Separate visual pathways for perception and action. *Trends in Neurosciences*, 15:20–25, 1992. 132

[62] R. Goyal, S. Ebrahimi Kahou, V. Michalski, J. Materzynska, S. Westphal, H. Kim, V. Haenel, I. Fruend, P. Yianilos, M. Mueller-Freitag, F. Hoppe, C. Thurau, I. Bax, and R. Memisevic. The "something something" video database for learning and evaluating visual common sense. In *Proceedings of International Conference on Computer Vision (ICCV)*, 2017. 64

[63] K. Grauman, A. Westbury, E. Byrne, Z. Chavis, A. Furnari, R. Girdhar, J. Hamburger, H. Jiang, M. Liu, X. Liu, M. Martin, T. Nagarajan, I. Radosavovic, S. K. Ramakrishnan, F. Ryan, J. Sharma, M. Wray, M. Xu, E. Z. Xu, C. Zhao, S. Bansal, D. Batra, V. Cartillier, S. Crane, T. Do, M. Doulaty, A. Erapalli, C. Feichtenhofer, A. Fragomeni, Q. Fu, C. Fuegen, A. Gebreselasie, C. Gonzalez, J. Hillis, X. Huang, Y. Huang, W. Jia, W. Khoo, J. Kolar, S. Kottur, A. Kumar, F. Landini, C. Li, Y. Li, Z. Li, K. Mangalam, R. Modhugu, J. Munro, T. Murrell, T. Nishiyasu, W. Price, P. R. Puentes, M. Ramazanova, L. Sari, K. Somasundaram, A. Southerland, Y. Sugano, R. Tao, M. Vo, Y. Wang, X. Wu, T. Yagi, Y. Zhu, P. Arbelaez, D. Crandall, D. Damen, G. M. Farinella, B. Ghanem, V. K. Ithapu, C. V. Jawahar, H. Joo, K. Kitani, H. Li, R. Newcombe, A. Oliva, H. S. Park, J. M. Rehg, Y. Sato, J. Shi, M. Z. Shou, A. Torralba, L. Torresani, M. Yan, and J. Malik. Ego4d: Around the world in 3, 000 hours of egocentric video. *CoRR*, abs/2110.07058, 2021. 1, 71, 72, 160, 161, 162

[64] A. Graves, S. Fernández, F. Gomez, and J. Schmidhuber. Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks. In *Proceedings of International Conference on Machine Learning (ICML)*, 2006. 64

[65] C. Gu, C. Sun, D. A. Ross, C. Vondrick, C. Pantofaru, Y. Li, S. Vijayanarasimhan, G. Toderici, S. Ricco, R. Sukthankar, C. Schmid, and J. Malik. Ava: A video dataset of spatio-temporally localized atomic visual actions. In *Proceedings of Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 58

[66] C. Gulcehre, O. Firat, K. Xu, K. Cho, and Y. Bengio. On integrating a language model into neural machine translation. *Computer Speech & Language*, 45:137–148, 2017. 61, 134

[67] A. Guzhov, F. Raue, J. Hees, and A. Dengel. Esresnet: Environmental sound classification based on visual domain models. In *Proceedings of International Conference on Pattern Recognition (ICPR)*, 2021. 27, 29, 30, 72

[68] G. Gwardys and D. Grzywczak. Deep image features in music information retrieval. *International Journal of Electronics and Telecommunications*, 60:321–326, 2014. 29, 30

[69] K. J. Han, R. Prieto, and T. Ma. State-of-the-art speech recognition using multi-stream self-attention with dilated 1d convolutions. In *IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, 2019. 33

[70] A. Y. Hannun, C. Case, J. Casper, B. Catanzaro, G. Diamos, E. Elsen, R. Prenger, S. Satheesh, S. Sengupta, A. Coates, and A. Y. Ng. Deep speech: Scaling up end-to-end speech recognition. *CoRR*, abs/1412.5567, 2014. 61, 134

[71] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 12, 27, 104, 121, 128

[72] K. He, G. Gkioxari, P. Dollar, and R. Girshick. Mask r-cnn. In *Proceedings of International Conference on Computer Vision (ICCV)*, 2017. 48, 71

[73] K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 45

[74] D. Hendrycks and K. Gimpel. Gaussian error linear units (gelus). *CoRR*, abs/1606.08415, 2016. 13

[75] S. Hershey, S. Chaudhuri, D. P. W. Ellis, J. F. Gemmeke, A. Jansen, R. C. Moore, M. Plakal, D. Platt, R. A. Saurous, B. Seybold, M. Slaney, R. J. Weiss, and K. Wilson. Cnn architectures for large-scale audio classification. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2017. 27, 118

[76] G. Hickok. The cortical organization of speech processing: Feedback control and predictive coding the context of a dual-stream model. *Journal of Communication Disorders*, 45:393–402, 2012. 132

[77] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural Computation*, 9: 1735–1780, 1997. 21

[78] H. Hu, C. H. Yang, X. Xia, X. Bai, X. Tang, Y. Wang, S. Niu, L. Chai, J. Li, H. Zhu, F. Bao, Y. Zhao, S. M. Siniscalchi, Y. Wang, J. Du, and C. Lee. Device-robust acoustic scene classification based on two-stage categorization and data augmentation. Technical report, DCASE2020 Challenge, 2020. 27, 28, 31

[79] J. Hu, L. Shen, and G. Sun. Squeeze-and-excitation networks. In *Proceedings of Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 24, 110

[80] Y. Huang, M. Cai, Z. Li, F. Lu, and Y. Sato. Mutual context network for jointly estimating egocentric gaze and action. *IEEE Transactions on Image Processing (TIP)*, 29: 7795–7806, 2020. 49, 50, 51, 52, 53, 54, 141, 152

[81] M. Huzaifah. Comparison of time-frequency representations for environmental sound classification using convolutional neural networks. *CoRR*, abs/1706.07156, 2017. 31

[82] V. Iashin and E. Rahtu. Multi-modal dense video captioning. In *Proceedings of Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2020. 59, 60, 61

[83] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *Proceedings of International Conference on Machine Learning (ICML)*, 2015. 82

[84] A. Jaegle, F. Gimeno, A. Brock, A. Zisserman, O. Vinyals, and J. Carreira. Perceiver: General perception with iterative attention. *CoRR*, abs/2103.03206, 2021. 59, 60, 61, 137, 147, 161

[85] E. Jang, S. Gu, and B. Poole. Categorical reparameterization with gumbel-softmax. In *Proceedings of International Conference on Learning Representations (ICLR)*, 2017. 49

[86] H. Jegou, M. Douze, C. Schmid, and P. Perez. Aggregating local descriptors into a compact image representation. In *Proceedings of Conference on Computer Vision and Pattern Recognition (CVPR)*, 2010. 77

[87] S. Ji, W. Xu, M. Yang, and K. Yu. 3d convolutional neural networks for human action recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 35:221–231, 2013. 11

[88] H. R. V. Joze, A. Shaban, M. L. Iuzzolino, and K. Koishida. Mmtm: Multimodal transfer module for cnn fusion. In *Proceedings of Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 18, 22, 23, 30, 110

[89] G. Kapidis, R. Poppe, E. van Dam, L. Noldus, and R. Veltkamp. Multitask learning to improve egocentric action recognition. In *Proceedings of International Conference on Computer Vision Workshops (ICCVW)*, 2019. 53, 142, 152

[90] W. Kay, J. Carreira, K. Simonyan, B. Zhang, C. Hillier, S. Vijayanarasimhan, F. Viola, T. Green, T. Back, P. Natsev, M. Suleyman, and A. Zisserman. The kinetics human action video dataset. *CoRR*, abs/1705.06950, 2017. 1, 15, 54, 65, 83, 117, 134

[91] S. H. Khan, M. Naseer, M. Hayat, S. W. Zamir, F. S. Khan, and M. Shah. Transformers in vision: A survey. *CoRR*, abs/2101.01169, 2021. 14, 60

[92] Y. Kim, Y. Jernite, D. Sontag, and A. Rush. Character-aware neural language models. In *Proceedings of AAAI Conference on Artificial Intelligence*, 2016. 63

[93] Q. Kong, Y. Cao, T. Iqbal, Y. Wang, W. Wang, and M. D. Plumbley. Panns: Large-scale pretrained audio neural networks for audio pattern recognition. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 28:2880–2894, 2020. 27, 28, 29, 30

[94] B. Korbar, D. Tran, and L. Torresani. Cooperative learning of audio and video models from self-supervised synchronization. In *Advances in Neural Information Processing Systems (NeurIPS)*. 2018. 1, 44, 45

[95] K. Koutini, H. Eghbal-zadeh, M. Dorfer, and G. Widmer. The receptive field as a regularizer in deep convolutional neural networks for acoustic scene classification. In *EUSIPCO*, 2019. 27, 28

[96] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2012. 27

[97] H. Kuehne, H. Jhuang, E. Garrote, T. Poggio, and T. Serre. HMDB: a large video database for human motion recognition. In *Proceedings of International Conference on Computer Vision (ICCV)*, 2011. 19

[98] H. Kuehne, A. B. Arslan, and T. Serre. The language of actions: Recovering the syntax and semantics of goal-directed human activities. In *Proceedings of Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014. 57, 64

[99] S. Lee, Y. Yu, G. Kim, T. Breuel, J. Kautz, and Y. Song. Parameter Efficient Multimodal Transformers for Video Representation Learning. In *Proceedings of International Conference on Learning Representations (ICLR)*, 2021. 59, 60, 61

[100] C. Li, Q. Zhong, D. Xie, and S. Pu. Co-occurrence feature learning from skeleton data for action recognition and detection with hierarchical aggregation. In *Proceedings of International Joint Conference on Artificial Intelligence (IJCAI)*, 2018. 24

[101] F. Li, N. Neverova, C. Wolf, and G. Taylor. Modout: Learning multi-modal architectures by stochastic regularization. In *IEEE International Conference on Automatic Face & Gesture Recognition (FG)*, pages 422–429, 2017. 40

[102] G. Li, L. Zhu, P. Liu, and Y. Yang. Entangled transformer for image captioning. In *Proceedings of International Conference on Computer Vision (ICCV)*, 2019. 59, 60, 61

[103] J. Li, Y. Yin, H. Chu, Y. Zhou, T. Wang, S. Fidler, and H. Li. Learning to generate diverse dance motions with transformer. *CoRR*, abs/2008.08171, 2020. 59, 60

[104] L. H. Li, M. Yatskar, D. Yin, C. Hsieh, and K. Chang. Visualbert: A simple and performant baseline for vision and language. *CoRR*, abs/1908.03557, 2019. 59, 60, 61

[105] R. Li, S. Yang, D. A. Ross, and A. Kanazawa. Ai choreographer: Music conditioned 3d dance generation with aist++. In *Proceedings of International Conference on Computer Vision (ICCV)*, 2021. 59, 60

[106] X. Li, V. Chebiyyam, and K. Kirchhoff. Multi-Stream Network with Temporal Attention for Environmental Sound Classification. In *Interspeech 2019*, 2019. 32

[107] Y. Li, Y. Li, and N. Vasconcelos. Resound: Towards action recognition without representation bias. In *Proceedings of European Conference on Computer Vision (ECCV)*, 2018. 139

[108] Y. Li, M. Liu, and J. M. Rehg. In the eye of beholder: Joint learning of gaze and actions in first person video. In *Proceedings of European Conference on Computer Vision (ECCV)*, 2018. 1, 49, 50, 51, 52, 53, 54, 66, 68, 70, 141, 152

[109] Y. Li, T. Nagarajan, B. Xiong, and K. Grauman. Ego-exo: Transferring visual representations from third-person to first-person videos. In *Proceedings of Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 1, 54, 150, 151, 162

[110] J. Lin, C. Gan, and S. Han. TSM: Temporal shift module for efficient video understanding. In *Proceedings of International Conference on Computer Vision (ICCV)*, 2019. 2, 8, 10, 16, 18, 19, 21, 129, 150, 151

[111] M. Lin, N. Inoue, and K. Shinoda. Ctc network with statistical language modeling for action sequence recognition in videos. In *Proceedings of the on Thematic Workshops of ACM Multimedia*, 2017. 63, 64, 160

[112] W. Lin, Y. Mi, J. Wu, K. Lu, and H. Xiong. Action recognition with coarse-to-fine deep feature integration and asynchronous fusion. *Proceedings of AAAI Conference on Artificial Intelligence*, 2018. 8, 18, 21, 46, 79

[113] Y. Liping, C. Xinxing, and T. Lianjie. Acoustic scene classification using multi-scale features. Technical report, DCASE2018 Challenge, 2018. 27

[114] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of International Conference on Computer Vision (ICCV)*, 2021. 14, 17

[115] X. Long, C. Gan, G. de Melo, J. Wu, X. Liu, and S. Wen. Attention clusters: Purely attention based local feature integration for video classification. In *Proceedings of Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 1, 2, 34, 35, 46, 56, 93, 94

[116] X. Long, C. Gan, G. Melo, X. Liu, Y. Li, F. Li, and S. Wen. Multimodal keyless attention fusion for video classification. In *Proceedings of AAAI Conference on Artificial Intelligence*, 2018. 1, 2, 34, 37, 46, 56

[117] J. Lu, D. Batra, D. Parikh, and S. Lee. Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019. 59, 60, 61

[118] M. Lu, D. Liao, and Z.-N. Li. Learning spatiotemporal attention for egocentric action recognition. In *Proceedings of International Conference on Computer Vision Workshops (ICCVW)*, 2019. 49, 50, 51, 52, 53, 54, 141, 142, 152

[119] J. Ma and D. Damen. Hand-object interaction reasoning, 2022. 161

[120] M. Ma, H. Fan, and K. M. Kitani. Going deeper into first-person activity recognition. In *Proceedings of Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 1, 2, 47, 48, 51, 52, 53

[121] E. Marcheret, G. Potamianos, J. Vopicka, and V. Goel. Detecting audio-visual synchrony using deep neural networks. In *Proceedings of INTERSPEECH*, 2015. 44

[122] M. McDonnell and W. Gao. Acoustic scene classification using deep residual networks with late fusion of separated high and low frequency paths. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020. 31, 32, 119, 120, 121

[123] P. Mégevand, S. Molholm, A. Nayak, and J. J. Foxe. Recalibration of the multisensory temporal window of integration results from changing task demands. *PLOS ONE*, 8, 2013. 75

[124] A. Mesaros, T. Heittola, and T. Virtanen. A multi-device dataset for urban acoustic scene classification. In *Proceedings of the Detection and Classification of Acoustic Scenes and Events 2018 Workshop (DCASE2018)*, 2018. 3, 72

[125] A. Miech, I. Laptev, and J. Sivic. Learnable pooling with context gating for video classification. *CoRR*, abs/1706.06905, 2017. 90

[126] A. Miech, D. Zhukov, J.-B. Alayrac, M. Tapaswi, I. Laptev, and J. Sivic. Howto100m: Learning a text-video embedding by watching hundred million narrated video clips. In *Proceedings of International Conference on Computer Vision (ICCV)*, 2019. 64

[127] T. Mikolov, M. Karafiát, L. Burget, J. Černockỳ, and S. Khudanpur. Recurrent neural network based language model. In *Proceedings of Annual Conference of the International Speech Communication Association (INTERSPEECH)*, 2010. 61, 62, 63, 134

[128] K. Min and J. J. Corso. Integrating human gaze into attention for egocentric activity recognition. In *Proceedings of Winter Conference on Applications of Computer Vision (WACV)*, 2021. 49, 50, 51, 52, 53, 54, 141, 142, 152

[129] D. Moltisanti, M. Wray, W. Mayol-Cuevas, and D. Damen. Trespassing the boundaries: Labeling temporal bounds for object interactions in egocentric video. In *Proceedings of International Conference on Computer Vision (ICCV)*, 2017. 51

[130] D. Moltisanti, S. Fidler, and D. Damen. Action recognition from single timestamp supervision in untrimmed videos. In *Proceedings of Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 161

[131] M. Müller. Harmonic–percussive separation (hps). https://www.audiolabs-erlangen.de/resources/MIR/FMP/C8/C8S1_HPS.html. 100

[132] J. Munro and D. Damen. Multi-modal domain adaptation for fine-grained action recognition. In *Proceedings of Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 55

[133] A. Nagrani, S. Yang, A. Arnab, C. Schmid, and C. Sun. Attention bottlenecks for multimodal fusion. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021. 1, 37, 46, 55, 60, 61, 150, 151

[134] D. Neimark, O. Bar, M. Zohar, and D. Asselmann. Video transformer network. In *Proceedings of International Conference on Computer Vision Workshops (ICCVW)*, 2021. 8, 14, 15, 17

[135] N. Neverova, C. Wolf, G. Taylor, and F. Nebout. Moddrop: Adaptive multi-modal gesture recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 38:1692–1706, 2016. 38, 39, 40, 42, 102, 112, 131

[136] Y. B. Ng and B. Fernando. Human action sequence classification. *CoRR*, abs/1910.02602, 2019. 57, 64

[137] A. Owens and A. A. Efros. Audio-visual scene analysis with self-supervised multisensory features. In *Proceedings of European Conference on Computer Vision (ECCV)*, 2018. 1, 44, 45

[138] K. Palanisamy, D. Singhania, and A. Yao. Rethinking CNN models for audio classification. *CoRR*, abs/2007.11154, 2020. 27, 28, 29, 30, 72

[139] C. Parise, C. Spence, and M. O. Ernst. When correlation implies causation in multisensory integration. *Current Biology*, 22:46 – 49, 2012. 75

[140] D. S. Park, W. Chan, Y. Zhang, C.-C. Chiu, B. Zoph, E. D. Cubuk, and Q. V. Le. SpecAugment: A Simple Data Augmentation Method for Automatic Speech Recognition. In *Proceedings of Interspeech*, 2019. 28, 29, 117

[141] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer. Automatic differentiation in pytorch. In *Advances in Neural Information Processing Systems Workshops (NeurIPSW)*, 2017. 82

[142] M. Patrick, Y. M. Asano, P. Kuznetsova, R. Fong, J. a. F. Henriques, G. Zweig, and A. Vedaldi. On compositions of transformations in contrastive self-supervised learning. In *Proceedings of International Conference on Computer Vision (ICCV)*, 2021. 1, 45, 46

[143] M. Patrick, D. Campbell, Y. M. Asano, I. M. F. Metze, C. Feichtenhofer, A. Vedaldi, and J. F. Henriques. Keeping your eye on the ball: Trajectory attention in video transformers. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021. 8, 14, 16, 17, 140, 150, 151

[144] M. E. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, and L. Zettlemoyer. Deep contextualized word representations. In *Proceedings of Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*, 2018. 138

[145] K. J. Piczak. Environmental sound classification with convolutional neural networks. In *International Workshop on Machine Learning for Signal Processing (MLSP)*, 2015. 27, 28

[146] K. J. Piczak. ESC: Dataset for Environmental Sound Classification. In *Proceedings of Annual ACM Conference on Multimedia (MM)*, 2015. 3, 29, 72

[147] H. Pirsiavash and D. Ramanan. Detecting activities of daily living in first-person camera views. In *Proceedings of Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012. 66, 68

[148] N. Qian. On the momentum term in gradient descent learning algorithms. *Neural Networks*, 12:145–151, 1999. 83

[149] A. Radford, K. Narasimhan, T. Salimans, and I. Sutskever. Improving language understanding by generative pre-training. 2018. 63

[150] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever. Language models are unsupervised multitask learners. 2019. 63

[151] N. Rai, H. Chen, J. Ji, R. Desai, K. Kozuka, S. Ishizaka, E. Adeli, and J. C. Niebles. Home action genome: Cooperative compositional action understanding. In *Proceedings of Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 1, 67, 70

[152] K. K. Reddy and M. Shah. Recognizing 50 human action categories of web videos. *Machine Vision and Applications (MVAP)*, 24:971–981, 2013. 65

[153] S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2015. 14

[154] A. Richard and J. Gall. Temporal action detection using a statistical language model. In *Proceedings of Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 63, 64, 160, 161

[155] M. Rohrbach, S. Amin, M. Andriluka, and B. Schiele. A database for fine grained activity detection of cooking activities. In *Proceedings of Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012. 64

[156] J. Salamon and J. P. Bello. Deep convolutional neural networks and data augmentation for environmental sound classification. *IEEE Signal Processing Letters*, 24:279–283, 2017. 27, 28

[157] J. Salamon, C. Jacoby, and J. P. Bello. A dataset and taxonomy for urban sound research. In *Proceedings of ACM International Conference on Multimedia (MM)*, 2014. 72

[158] R. Santoro, M. Moerel, F. D. Martino, R. Goebel, K. Ugurbil, E. Yacoub, and E. Formisano. Encoding of natural sounds at multiple spectral and temporal resolutions in the human auditory cortex. *PLOS Computational Biology*, 10:1–14, 2014. 101

[159] F. Sener, D. Singhania, and A. Yao. Temporal aggregate representations for long-range video understanding. In *Proceedings of European Conference on Computer Vision (ECCV)*, 2020. 56, 57, 58

[160] A. Senocak, T.-H. Oh, J. Kim, M.-H. Yang, and I. So Kweon. Learning to localize sound source in visual scenes. In *Proceedings of Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 110, 111, 160

[161] D. Shan, J. Geng, M. Shu, and D. F. Fouhey. Understanding human hands in contact at internet scale. In *Proceedings of Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 71

[162] P. Shaw, J. Uszkoreit, and A. Vaswani. Self-attention with relative position representations. In *Proceedings of Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*, 2018. 137, 147

[163] X. SHI, Z. Chen, H. Wang, D.-Y. Yeung, W.-k. Wong, and W.-c. WOO. Convolutional lstm network: A machine learning approach for precipitation nowcasting. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2015. 51

[164] J. Shin, Y. Lee, and K. Jung. Effective sentence scoring method using bert for speech recognition. In *Proceedings of Asian Conference on Machine Learning (ACML)*, 2019. 139

[165] M. Shridhar, L. Manuelli, and D. Fox. Cliport: What and where pathways for robotic manipulation. In *Proceedings of Conference on Robot Learning (CoRL)*, 2021. 132

[166] G. A. Sigurdsson, G. Varol, X. Wang, A. Farhadi, I. Laptev, and A. Gupta. Hollywood in homes: Crowdsourcing data collection for activity understanding. In *Proceedings of European Conference on Computer Vision (ECCV)*, 2016. 67

[167] G. A. Sigurdsson, A. Gupta, C. Schmid, A. Farhadi, and K. Alahari. Actor and observer: Joint modeling of first and third-person videos. In *Proceedings of Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 1, 54, 55, 67, 68, 70

[168] K. Simonyan and A. Zisserman. Two-stream convolutional networks for action recognition in videos. In *Advances in Neural Information Processing Systems (NeurIPS)*. 2014. 2, 8, 18, 19, 20, 34

[169] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In *Proceedings of International Conference on Learning Representations (ICLR)*, 2015. 27

[170] S. Singh, C. Arora, and C. V. Jawahar. First person action recognition using deep learned descriptors. In *Proceedings of Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 2, 51, 52

[171] L. Smaira, J. Carreira, E. Noland, E. Clancy, A. Wu, and A. Zisserman. A short note on the kinetics-700-2020 human action dataset. *CoRR*, abs/2010.10864, 2020. 66

[172] S. Song, V. Chandrasekhar, B. Mandal, L. Li, J.-H. Lim, G. Sateesh Babu, P. Phyo San, and N.-M. Cheung. Multimodal multi-stream deep learning for egocentric activity recognition. In *Proceedings of Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2016. 2, 51, 52

[173] K. Soomro, A. R. Zamir, and M. Shah. UCF101: A dataset of 101 human actions classes from videos in the wild. *CoRR*, abs/1212.0402, 2012. 19, 65

[174] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15:1929–1958, 2014. 38

[175] S. Stein and S. J. McKenna. Combining embedded accelerometers with computer vision for recognizing food preparation activities. In *Proceedings of the ACM International Joint Conference on Pervasive and Ubiquitous Computing (UbiComp)*, 2013. 64

[176] R. A. Stevenson, M. M. Wilson, A. R. Powers, and M. T. Wallace. The effects of visual training on multisensory temporal processing. *Experimental Brain Research*, 225:479–489, 2013. 75

[177] M. Subedar, R. Krishnan, P. L. Meyer, O. Tickoo, and J. Huang. Uncertainty-aware audiovisual activity recognition using deep bayesian variational inference. In *Proceedings of International Conference on Computer Vision (ICCV)*, 2019. 38, 39, 40, 42

[178] S. Sudhakaran and O. Lanz. Attention is all we need: Nailing down object-centric attention for egocentric activity recognition. In *Proceedings of British Machine Vision Conference (BMVC)*, 2018. 2, 50, 51, 52, 141, 152

[179] S. Sudhakaran, S. Escalera, and O. Lanz. Lsta: Long short-term attention for egocentric action recognition. In *Proceedings of Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 1, 2, 50, 51, 52, 53, 141, 156

[180] S. Suh, S. Park, Y. Jeong, and T. Lee. Designing acoustic scene classification models with CNN variants. Technical report, DCASE2020 Challenge, 2020. 27, 28, 31

[181] C. Sun, A. Shrivastava, S. Singh, and A. Gupta. Revisiting unreasonable effectiveness of data in deep learning era. In *Proceedings of International Conference on Computer Vision (ICCV)*, 2017. 14

[182] C. Sun, A. Myers, C. Vondrick, K. Murphy, and C. Schmid. Videobert: A joint model for video and language representation learning. In *Proceedings of International Conference on Computer Vision (ICCV)*, 2019. 59, 60, 61

[183] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In *Proceedings of Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015. 11

[184] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 27

[185] J. Tompson, R. Goroshin, A. Jain, Y. LeCun, and C. Bregler. Efficient object localization using convolutional networks. In *Proceedings of Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015. 40

[186] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri. Learning spatiotemporal features with 3d convolutional networks. In *Proceedings of International Conference on Computer Vision (ICCV)*, 2015. 11

[187] D. Tran, H. Wang, L. Torresani, J. Ray, Y. LeCun, and M. Paluri. A closer look at spatiotemporal convolutions for action recognition. In *Proceedings of Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 2, 8, 11, 18, 19, 21, 105

[188] D. Tran, H. Wang, L. Torresani, and M. Feiszli. Video classification with channel-separated convolutional networks. In *Proceedings of International Conference on Computer Vision (ICCV)*, 2019. 11

[189] G. Varol, I. Laptev, and C. Schmid. Long-term temporal convolutions for action recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 40: 1510–1517, 2018. 9

[190] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. u. Kaiser, and I. Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2017. 12

[191] M. T. Wallace and R. A. Stevenson. The construct of the multisensory temporal binding window and its dysregulation in developmental disabilities. *Neuropsychologia*, 64:105 – 123, 2014. 75, 76

[192] H. Wang and C. Schmid. Action recognition with improved trajectories. In *Proceedings of International Conference on Computer Vision (ICCV)*, 2013. 7

[193] H. Wang, A. Kläser, C. Schmid, and C.-L. Liu. Action recognition by dense trajectories. In *Proceedings of Conference on Computer Vision and Pattern Recognition (CVPR)*, 2011. 7

[194] L. Wang, Y. Xiong, Z. Wang, Y. Qiao, D. Lin, X. Tang, and L. Van Gool. Temporal segment networks: Towards good practices for deep action recognition. In *Proceedings of European Conference on Computer Vision (ECCV)*, 2016. 2, 8, 9, 15, 18, 19, 21, 77, 79, 80, 83, 93, 119, 129, 150, 151, 156

[195] L. Wang, P. Luc, Y. Wu, A. Recasens, L. Smaira, A. Brock, A. Jaegle, J. Alayrac, S. Dieleman, J. Carreira, and A. van den Oord. Towards learning universal audio representations. *CoRR*, abs/2111.12124, 2021. 131, 132

[196] W. Wang, D. Tran, and M. Feiszli. What makes training multi-modal classification networks hard? In *Proceedings of Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 1, 4, 38, 39, 41, 55, 56, 102, 112, 126, 127, 131, 134

[197] W. Wang, D. Tran, and M. Feiszli. What makes training multi-modal classification networks hard? In *Proceedings of Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 151

[198] W. Wang, E. Xie, X. Li, D.-P. Fan, K. Song, D. Liang, T. Lu, P. Luo, and L. Shao. Pyramid vision transformer: A versatile backbone for dense prediction without convolutions. In *Proceedings of International Conference on Computer Vision (ICCV)*, 2021. 14, 17

[199] X. Wang, R. Girshick, A. Gupta, and K. He. Non-local neural networks. In *Proceedings of Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 35, 36, 57

[200] X. Wang, Y. Wu, L. Zhu, and Y. Yang. Baidu-uts submission to the epic-kitchens action recognition challenge 2019. *CoRR*, abs/1906.09383, 2019. 48, 51, 52

[201] X. Wang, Y. Wu, L. Zhu, Y. Yang, and Y. Zhuang. Symbiotic attention: Uts-baidu submission to the epic-kitchens 2020 action recognition challenge. 2020. 48, 51, 52, 53

[202] M. Wray, D. Larlus, G. Csurka, and D. Damen. Fine-grained action retrieval through multiple parts-of-speech embeddings. In *Proceedings of International Conference on Computer Vision (ICCV)*, 2019. 136

[203] C.-Y. Wu, C. Feichtenhofer, H. Fan, K. He, P. Krahenbuhl, and R. Girshick. Long-term feature banks for detailed video understanding. In *Proceedings of Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 56, 58, 151

[204] H. Wu, B. Xiao, N. Codella, M. Liu, X. Dai, L. Yuan, and L. Zhang. Cvt: Introducing convolutions to vision transformers. In *Proceedings of International Conference on Computer Vision (ICCV)*, 2021. 14

[205] Y. Wu and T. Lee. Time-frequency feature decomposition based on sound duration for acoustic scene classification. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020. 32

[206] Z. Wu, Y.-G. Jiang, X. Wang, H. Ye, and X. Xue. Multi-stream multi-class fusion of deep networks for video classification. In *ACM International Conference on Multimedia*, 2016. 34, 46, 56

[207] F. Xiao, Y. J. Lee, K. Grauman, J. Malik, and C. Feichtenhofer. Audiovisual slowfast networks for video recognition, 2020. 1, 4, 30, 31, 35, 36, 38, 39, 40, 42, 46, 55, 56, 102, 103, 105, 112, 113, 115, 126, 131, 151

[208] F. Xiao, Y. J. Lee, K. Grauman, J. Malik, and C. Feichtenhofer. Audiovisual slowfast networks for video recognition. *CoRR*, abs/2001.08740, 2020. 134

[209] Z. Yang, Z. Dai, Y. Yang, J. Carbonell, R. R. Salakhutdinov, and Q. V. Le. XLNet: Generalized autoregressive pretraining for language understanding. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019. 63, 134, 138

[210] L. Ye, M. Rochan, Z. Liu, and Y. Wang. Cross-modal self-attention network for referring image segmentation. In *Proceedings of Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 59, 60, 61

[211] K. Yuan, S. Guo, Z. Liu, A. Zhou, F. Yu, and W. Wu. Incorporating convolution designs into visual transformers. In *Proceedings of International Conference on Computer Vision (ICCV)*, 2021. 14

[212] J. Yue-Hei Ng, M. Hausknecht, S. Vijayanarasimhan, O. Vinyals, R. Monga, and G. Toderici. Beyond short snippets: Deep networks for video classification. In *Proceedings of Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015. 9, 19

[213] C. Zhang, A. Gupta, and A. Zisserman. Temporal query networks for fine-grained video understanding. In *Proceedings of Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 57

[214] H. Zhang, M. Cisse, Y. N. Dauphin, and D. Lopez-Paz. mixup: Beyond empirical risk minimization. In *Proceedings of International Conference on Learning Representations (ICLR)*, 2018. 28, 141

[215] Z. Zhang, Y. Wang, C. Gan, J. Wu, J. B. Tenenbaum, A. Torralba, and W. T. Freeman. Deep audio priors emerge from harmonic convolutional networks. In *Proceedings of International Conference on Learning Representations (ICLR)*, 2020. 31, 32, 133, 160

[216] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba. Learning deep features for discriminative localization. In *Proceedings of Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 50

[217] B. Zhou, A. Andonian, A. Oliva, and A. Torralba. Temporal relational reasoning in videos. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018. 2, 8, 10, 15, 18, 19, 21, 79, 129, 156

[218] L. Zhou, C. Xu, and J. J. Corso. Towards automatic learning of procedures from web instructional videos. In *Proceedings of AAAI Conference on Artificial Intelligence*, 2018. 64

[219] I. Zulfiqar, M. Moerel, and E. Formisano. Spectro-temporal processing in a two-stream computational model of auditory cortex. *Frontiers in Computational Neuroscience*, 13: 95, 2020. 101, 132