5-29-2018

Co-Training of Audio and Video Representations from Self-Supervised Temporal Synchronization

Bruno Korbar
Dartmouth College

Co-Training of Audio and Video Representations from Self-Supervised Temporal Synchronization

Undergraduate Thesis

written by

Bruno Korbar

under the supervision of **Professor Lorenzo Torresani** and **Du Tran**, and submitted to the Committee as a culminating experience for the degree of

Bachelor of Arts in Computer Science

at *Dartmouth College.*

Date of the public presentation:
May 29, 2018

Members of the Thesis Committee:
Prof Lorenzo Torresani
Prof Saeed Hassanpour
Prof Venkatramanan Siva Subrahmanian

Dartmouth Computer Science Technical Report **TR2018-849**

Acknowledgments

First and foremost, I would like to thank my advisor, Professor Lorenzo Torresani, for his undivided attention and mentorship despite geographical differences, and continuous supply of fine Italian espresso.

I would also like to thank my mentor Du Tran, lab mate Karim Ahmed, and former lab mate Naofumi Tomita for numerous brainstorming and debugging sessions over the last couple of months.

Furthermore, this thesis could not have been finished without the support from Dartmouth Computer Science Department, undergraduate advisers Professors Bailey-Kellog and Grigoryan, and system administrator extraordinaire William Ang, all of whom have my deepest gratitude.

I would also like to thank Professors Saeed Hassanpour and VS Subrahmanian for joining Professor Torresani on my thesis committee, and taking the time to read and discuss my thesis.

Last but not least, I'm thankful for my family and close friends, who were incredibly understanding of my absence during the deadlines and kept me motivated in moments of doubt.

Co-Training of Audio and Video Representations from Self-Supervised Temporal Synchronization

Bruno Korbar '18
Dartmouth College
bruno.18@dartmouth.edu

Lorenzo Torresani
Dartmouth College
Primary Advisor
lt@dartmouth.edu

Du Tran
Facebook AI Research
trandu@fb.com

Abstract

There is a natural correlation between the visual and auditive elements of a video. In this work, we use this correlation in order to learn strong and general features via cross-modal self-supervision with carefully chosen neural network architectures and calibrated curriculum learning. We suggest that this type of training is an effective way of pretraining models for further pursuits in video understanding, as they achieve on average 14.8% improvement over models trained from scratch. Furthermore, we demonstrate that these general features can be used for audio classification and perform on par with state-of-the-art results. Lastly, our work shows that using cross-modal self-supervision for pretraining is a good starting point for the development of multi-sensory models.

1 Introduction

Image recognition has been at the forefront of deep learning efforts since the breakthrough of AlexNet [1] and the widespread availability of increasingly large datasets such as Imagenet [2]. Models pretrained on Imagenet [2] allowed for development of very strong feature extractors that succeed at every task thrown at them. Similar efforts on video-understanding tasks have been significantly less successful, as the state-of-the-art models trained on videos [3] barely outperform models trained on still images. This small margin in performance can be explained by a lack of datasets that are challenging for video understanding in the same way the Imagenet [2] is challenging for still images. One could argue that many datasets in the video domain are "easy" in the sense that even humans could guess the action depicted based on one frame alone — for example, a person holding a baseball bat on a field is clearly playing baseball or a person on a bicycle is clearly cycling. In other words, actions tend to be better described by appearance rather than motion. Even though a commendable effort has been put into developing a new generation of larger and more diverse datasets [4, 5, 6], they often come with a large cost in terms of time and money.

One way to overcome the lack of challenging video datasets is to extrapolate information learned from the still images trained on the challenging datasets to a video-understanding setting (e.g. using inflation as a method of pretraining video models [7]). Another promising approach is to use multi-sensory aspects of videos in order to increase the amount of information that we can learn and use from a single video and therefore enrich the existing datasets with information that is already provided in the video.

In this paper, we take advantage of the multi-sensory aspect of videos to propose a different approach to pretraining neural networks for tasks of video understanding that is able to leverage additional information contained in different modalities. We train our networks in a self-supervised fashion that does not require any manual labelling of the videos, and could theoretically be applied to infinitely large training sets.

Preprint. Work in progress.

Specifically, we propose a form of co-training where the two modalities provide correlated but distinct information about the video. We use the audio to supervise the training of video features, and vice versa to create a balanced co-training scheme. We name this task audio-visual temporal synchronization (AVTS) and define it as the binary classification problem describing whether a given audio sample and video clip are synchronised.

Features learned in this way can be used for different video and audio related applications, as a pretraining for video-understanding tasks, feature extractors for audio related tasks and as a starting point for further development of multisensory models. Sample applications are illustrated in Figure 1.

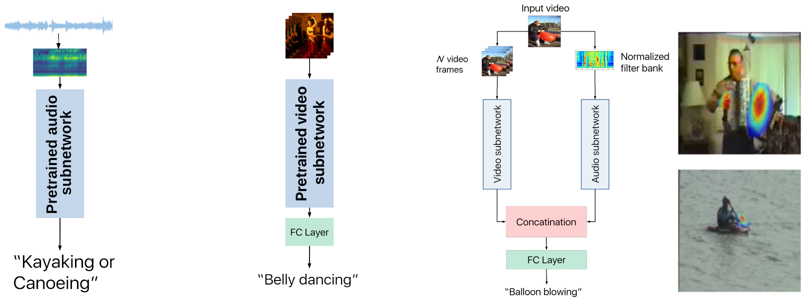
1.1 Related work

We are not the first to propose the idea of using the correlation between audio and video as a mean of developing self-supervised feature extractors. Aytar et al. [8] pioneered the idea of using video as a form of supervision in order to learn salient sound representations. This idea was taken a step further by Arandjelovic and Zisserman that devised a way to jointly learn both auditive and visual features from the same video [9], and later developed and optimised their approach for cross-modal retrieval and sound localization [10]. Our work differs from these in several important ways. While their approach used only a single frame of a video and therefore focuses on exploiting the *semantic* aspect of the audio-visual correlation, our method uses a video clip as an input, thus taking the temporal aspect of videos into account as well. Similarly, we argue that our AVTS task is different from their task of "audio-visual correspondence", as training with AVTS forces the networks to exploit the *correlation of sound and motion* within the video in addition to semantics.

In that sense, our work is similar to the problem of synchronization such as the work of Chung and Zisserman [11], where the correlation of sound and motion was used in order to correlate movements of the mouth with the words spoken. Here, we expand the scope of that study and show that applications of this way of training are far greater than just synchronization.

Our work is concurrent with those of Owens and Efros [12] and Zhao et al.[13], however, they differ in key applications and approach to the problems. The former focuses on the task of audio-localization within the video domain, while we use the self-supervision for feature learning. The later is similar in spirit to ours, but we present stronger experimental results on comparable benchmarks, with a substantially different technical approach.

Figure 1: Sample applications of our system



(a) Pretrained audio subnetwork as a feature extractor.

(b) Pretraining for video-based action recognition.

(c) Cross-modal action recognition

(d) Sound and motion localization.

2 Technical approach

Our training procedure is the biggest differentiator between our method and those of Arandjelovic et al. [9], and Owens et al. [12] as we train with a different objective in mind and receive benefit from curriculum learning. In order to do so, we propose a training task of audio-visual temporal synchronization (Sec 2.1), contrastive loss as an objective (Sec 2.2), and curriculum learning scheme [14] with carefully chosen negative examples (Sec 2.3 and 2.4). Finally, we outline the details of our architecture in Sec 2.5.

2.1 Audio visual temporal synchronization (AVTS)

Assume a given dataset consisting of N audio-video pairs $(a^{(i)}, v^{(i)})$, where $(a^{(i)}$ is an i-th audio clip, and $v^{(i)}$ is an i-th video clip. For i-th pair at training time, we have a label $y^{(i)} \in \{0, 1\}$ which indicates if the pair is "in-sync" or not. If $y^{(i)} = 0$, then pair $(a^{(i)}, v^{(i)})$ is picked either from different videos, or are misaligned within the video. If $y^{(i)} = 1$, the pair is in perfect sync.

At a very high level, the goal is to find a classification function that minimizes the error on as many pairs as possible. We define our function in terms of two processing streams that provide feature representation for each modality, and the function fuses the representations and perform the final prediction. The detailed explanation of how we evaluate AVTS task is provided in the experiments section 3.1.

2.2 Learning objective

Our objective is to make the output of the audio similar to the output of the video for the corresponding pairs, and different for the mismatched ones. It is still completely self-supervised and no action labels are seen during training time.

Following procedure in Chung et al [11], originally developed for training of Siamese networks [15], we attempt to minimize (or maximize) the contrastive loss function. If the distance between the n-th audio and video vector is $d_n = ||v_n - a_n||_2$ then contrastive loss is given by

$$E = \frac{1}{2N} \sum_{n=1}^{N} (y_n) d_n^2 + (1 - y_n) max(margin - d_n, 0)^2 \tag{1}$$

where N is the number of examples and y_n is the indicator variable which is 0 if n-th video corresponds to n-th audio and 1 otherwise, and $margin$ is a hyperparameter set to 0.99. In principle, another possible approach would be to train the network as a classification problem using cross-entropy loss function, however, in our set-up, we were unable to achieve convergence. With higher-dimensional output and architecture modifications (three fully connected layers added after concatenation), we were able to achieve convergence, but there was no benefit to the performance on AVTS task (67% in 90 epochs, as opposed to 69% with our system), nor feature quality (71% on ESC50, compared to 82% of our system).

2.3 Selection of negative examples

For each example, we choose whether it is a positive or a negative example with 50% probability. We consider *positive* examples to be those where video frames correspond in time with the audio example. We divide *negative* examples in two categories; *easy negatives* are those where the frames and sound come from different videos, and *hard negatives* which are those where the pair is taken from the same video, but there is at least half-a-second time gap between the two. Additionally, we have tried extending this idea and using *super-hard negatives* which we define as audio examples from the same video that has a fixed overlap with the timing of the image frames, but such negatives were not beneficial for neither training nor feature quality.

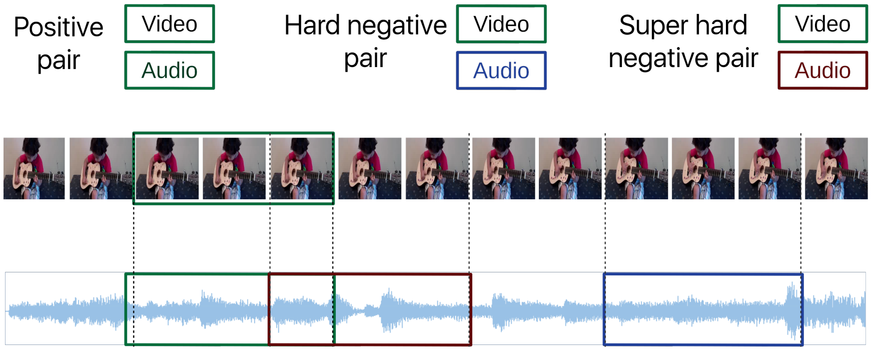


Figure 2: Example of a positive and a "hard" negative taken from the same video. Positive audio example is one that corresponds in time to the video frames. A "hard" negative audio sample is one that is misaligned with the frames by more than half a second, and "super-hard" negative is misaligned, but has a fixed 0.2s overlap with the positive example.

2.4 Curriculum learning

As a baseline, we trained the system from scratch with easy negatives alone, with hard negatives alone, as well as with the fixed proportion of hard negatives. We have found that when hard negatives were introduced from the beginning — either fully or as a proportion — not only was the initial loss significantly higher (which is to be expected), but the objective was close to impossible to optimize. Similarly, they produced poor results on AVTS, as well as other downstream tasks. However, if we introduced the hard negatives after the loss stagnated (in our case between 40th and 50th epoch), the optimization continued, and resulting features yielded better results on AVTS task and in every quantitative aspect that we have evaluated them by. Empirically, we have found the best results when 25% of all negatives were hard negatives. Table 1 reflects the difference in performance on AVC task between curriculum learning and learning with easy or hard negatives from scratch.

Table 1: Impact of curriculum learning on the performance of the AVC task on Kinetics. Performance is shown next to the benchmark L3 net [9]. Clearly, the model trained sequentially achieves significant boost in performance.

Method	Negative type	Epochs	AVTS Accuracy
Single learning stage	easy	1 - 90	69%
	easy + 25% hard	1 - 90	59%
	hard	1-90	52%
Curriculum learning	easy + 25% hard	51 - 90	**78%**
	hard	51-90	66%

2.5 Architecture details

Our training architecture is composed of two main parts: audio subnetwork and video subnetwork, each taking the respective input, and pre-processing. The overview of our architecture can be seen in Figure 3. The in-depth description of the video and audio subnetwork follows in section 2.5.1 and 2.5.2 respectfully.

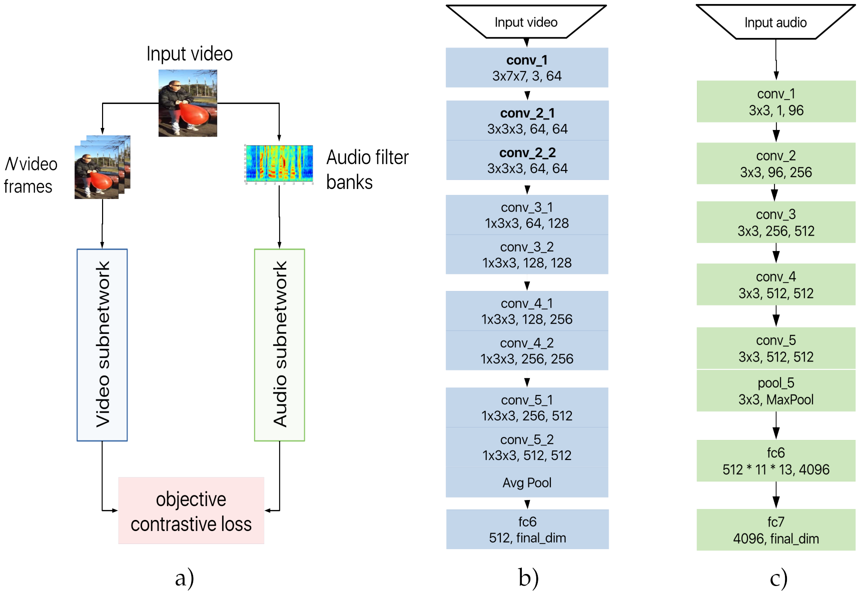


Figure 3: Architecture design and overview can be seen in a), video subnetwork based on MCx family of architectures in b), and audio subnetwork based on VGG-like model used by Chung et al. [11] in c).

2.5.1 Video

Our video subnetwork is based on a mixed convolution (MCx) family of architectures [16, 17]. We chose this particular family of architectures because it allows for a precise learning of "motion" descriptors in early 3D layers and achieves good performance on video-understanding benchmarks. We found that within our system MC3 yields the best performance overall and is used whenever not specified otherwise. In the evaluation of the video features, we also show results for MC2 in order to asses the importance of that additional 3D layer block. Full details of MC3 architecture are discussed in Tran et al [17]. Note that our architecture differs from the original in the dimensionality of the final FC layer and the lack of residual connections.

The input in our video subnetwork are video clips of size $(3 \times t \times w \times h)$, where t is number of frames, 3 is number of channels (RGB), and h, w are height and width respectfully. In order to keep pooling sizes and final dimensionality the same, we keep the number of frames $t = 25$ regardless of the different sampling rate for different datasets.

2.5.2 Audio

An mp3 audio is extracted from each video, FFT-filterbank features are extracted and normalized, and then passed through our audio subnetwork. The design of audio subnetwork is based on VGG-like architecture with batch normalization used for audio-visual synchronisation in Chung and Zisserman [11].

2.6 Implementation details

Input alignment and pre-processing: The starting frame of each clip is chosen at random within a video. The length of each clip is set to $t = 25$ frames. This results in a clip duration of 1 second on all the datasets considered here except for HMDB51, which uses a frame rate different from 25 fps (clip duration on HMDB51 is roughly 1.2 seconds). Standard spatial transformations (multi-scale random crop, random horizontal flip, and **Z** normalization) are applied to each frame with consistent parameters across all frames of the clip at training time. FFT- filterbank features are extracted from the aligned clip, and **Z** normalization is applied. The filterbank parameters are set as follows: window length to 0.02, window step to 0.01, FFT size to 1024, and number of filters to 40. Mean and standard deviation for normalization are extracted over a random 20% subset of the training dataset.

Training details: We train both video and sound networks using stochastic gradient descent with initial learning rate determined via grid search, and both networks streams are trained simultaneously. Training is done on four GPUs with mini-batches of size 64 per GPU. Learning rate is multiplied by 0.1 each time the loss value fails to decrease for more than 5 epochs.

3 Experiments

3.1 Evaluation on Audio-Visual Temporal Synchronization

The first benchmark we look at is how well the system performs on the AVTS task — is the audio clip "in-sync" with the video clip. We use two methods of evaluation: setting a threshold on d_n, and fine-tuning FC layers on top of the learned feature extractors.

In order to directly compare the results of our training method with the one of Arandjelovic et al [9], we add two fully connected layers (512, 512; 512, 2) after feature concatenation after training, and fine-tune the best model for binary classification task of whether the video is synchronized to the audio. In order to have results completely comparable to those of Arandjelovic et al [9], for final evaluation, we only consider "easy" negatives. Results on Kinetics dataset can be found in Table 1.

While setting a threshold on d_n did get a comparable performance for easy negatives (76% on Kinetics), we found it to be unreliable and largely inferior when hard negatives were introduced, as the performance dropped to under 70% accuracy on Kinetics.

3.1.1 Results and discussion

At 78% at Kinetics [4] and 86% on Audioset [5] our model marginally outperforms the L3Net (74% and 81% respectfully) [9]. The inclusion of hard negatives during training allows us to maintain a reasonably high performance when hard negatives are included in the testing set (70% on Kinetics), which outperforms our implementation of L3 Net whose performance drops drastically when hard negatives are included (57% in our experiments). Performance on the AVTS task is, however, not our main objective, but only a proxy for learning rich representations. In the following sections, we present and discuss the results of the evaluation of auditive and visual features and the significance of our training procedure for each of them.

3.2 Evaluation of AVTS audio features

In this section we evaluate features learned through our approach. Namely, we train our system in a self-supervised fashion on different datasets, and evaluate results on standard sound classification datasets: ESC-50 [18] and DCASE2014 [19].

The **ESC-50 dataset** [18] is a labelled collection of environmental recordings suitable for benchmarking methods of sound classification. It consists of 2000 sound recordings, each of which being 5 seconds long and organized in 50 semantic classes. Each class consists of approximately 40 examples. Results on ESC-50 are summarised in Table 2.

Table 2: Evaluation of our audio features on ESC50 [18] dataset. "VGG-like" is a baseline performance of our sound extractor architecture trained on ESC50 without AVTS pretraining.

Method	Pretraining Dataset	# auxiliary training examples	Accuracy
SVM-MFCC [18]	none	none	39.6%
Random Forest [18]	none	none	44.3%
VGG-like	none	none	61.6%
SoundNet [8]	Flickr-SoundNet	2M+	74.2%
L3 Net [9]	Flickr-SoundNet	2M+	79.3%
AVTS features	Kinetics	230K	76.7%
AVTS features	AudioSet	1.8M	80.6%
AVTS features	Flickr-SoundNet	2M+	**82.3%**
Human performance [9]	n/a	n/a	*81.3%*
State-of-the-art (RBM) [20]	none	none	**86.5%**

Commonly used for the evaluation of audio representations, **DCASE2014** training dataset consists of 10 seconds long audio segments from 15 acoustic scenes; each acoustic scene has 312 segments totalling 52 minutes of audio for training. Evaluation dataset consists of 10 seconds long audio segments from 15 acoustic scenes, where each acoustic scene has 108 segments totalling 18 minutes of audio. Results of the classification on DCASE2014 can be seen in Table 3

Table 3: Evaluation of the audio features on DCASE2014 [19] dataset. "VGG-like" is a baseline performance of our sound extractor architecture trained on DCASE2014 without AVTS pretraining.

Method	Pretraining dataset	# of training examples	Accuracy
SoundNet [8]	Flickr-SoundNet	2M+	88%
L3 Net [9]	Flickr-SoundNet	2M+	93%
VGG-like	None	4.6K	72%
AVTS features	Kinetics	230K	91%
AVTS features	AudioSet	1.8M	93%
AVTS features	Flickr-SoundNet	2M+	**94%**

3.2.1 Experimental procedure

In order to evaluate our performance, we train a multiclass one-vs-all linear SVM on the features extracted from *conv_5* layer of our audio subnetwork without fine-tuning. For each five-second clip of the ESC-50 dataset, we extract 10 equally spaced two-second subclips, while for the 6× longer DCASE clips we extract 60 two-second subclips, and report the final clip score as the average over scores of its subclips.

3.2.2 Results and discussion

Our results in the audio domain show that our pretraining method is able to learn strong audio representations even on relatively small datasets, and that the representations learned with our method outperform our randomly initialized audio network trained on the training portion of ESC50 and DCASE2014 datasets (see "VGG-like" in Tables 2 and 3). Our best result not only outperforms other self-supervised benchmarks but also marginally surpasses human recognition benchmark on ESC50 [18]. Additionally, we have found that by using filter banks as an audio input improves our performance compared to MFCC features used by Chung [11] (71.9% on ESC50 when pretrained on Kinetics compared to 76.7% with filter banks). Furthermore, using filter bank features allows for a fairly consistent training of our model — something we were not able to achieve with raw audio signals as proposed by Aytar et al. [8].

3.3 Evaluation of AVTS video features

In this section, we evaluate the strength of the video features learned by our video subnetwork. We fine tune the network on smaller datasets and compare the results with the same network architecture trained from scratch, as well as with the models trained on a fully supervised action recognition task on Kinetics. Rather than set state-of-the-art performance, we aim to persuade the reader that our pretraining does not only beat random initialization, but it also tracks more than just the appearance, which is one way of explaining the on-average larger difference between the clip and video accuracy for the self-supervised system. The discussion of the experimental procedure can be found in section 3.3.1.

First, we evaluate our pretraining by fine-tuning the best video model on UCF101 action recognition dataset [21]. UCF101 is an “action recognition dataset of realistic action videos, collected from YouTube, having 101 action categories” [21]. The dataset contains 13,320 videos in 101 balanced classes with variable camera movement, object occlusions, background noise, etc. We use standard train/test splits as defined by the authors. Results can be found in Table 4.

We also evaluate the strength of our features on HMDB51 [22] dataset. It contains 6,849 labelled clips collected from various public databases, divided into 51 action categories, each category containing at least 101 clips. We use standard train/val/test split as defined by the authors. Results can be found in Table 5.

3.3.1 Experimental procedure

In order to obtain the following results, we load the weights of the video subnetwork belonging to the model that achieved the best performance on the AVTS task. Next, we truncate the fully connected layer of the MCx architecture with one with appropriate size for the dataset in question. The learning rate is reduced by a factor of 10 from the learning rate that is optimal for the action-recognition task and the network is fine-tuned for 30 epochs.

Table 4: Evaluation of spatiotemporal features learned through AVTS after fine tuning on UCF101 [21] dataset. We report clip and video level accuracies of the baseline models trained from scratch, models pretrained on Kinetics[4] dataset with action categorization labels as an upper bound, and models using our proposed method. Additionally, we compare them to the current state of the art.

Video Network Architecture	Pretraining Dataset	Pretraining Supervision Method	Clip@1	Video@1
MC2	None	NA	42.60%	67.23 %
MC3	None	NA	44.28%	69.11 %
MC2	Kinetics	self-supervised	56.11%	83.57%
MC3	Kinetics	self-supervised	58.62%	85.84%
MC2	Kinetics	fully supervised	67.48%	87.85%
MC3	Kinetics	fully supervised	71.58%	90.53%
I3D - two stream [7]	Imagenet + Kinetics	fully-supervised	/	98.00%

Table 5: Evaluation of spatiotemporal features learned through AVTS after fine tuning on HMDB51 [22] dataset. We report clip and video level accuracies of the baseline models trained from scratch, models pretrained on Kinetics[4] dataset with action categorization labels as an upper bound, and models using our proposed method. Additionally, we compare them to the current state of the art.

Video Network Architecture	Pretraining Dataset	Pretraining Supervision Method	Clip@1	Video@1
MC2	None	NA	18.21%	41.21%
MC3	None	NA	18.77%	43.86%
MC2	Kinetics	self-supervised	25.21%	54.26%
MC3	Kinetics	self-supervised	26.89%	56.91%
MC2	Kinetics	fully-supervised	40.32%	61.99%
MC3	Kinetics	fully-supervised	45.81%	66.84%
I3D - two stream [7]	Imagenet + Kinetics	fully-supervised	/	80.70%

3.3.2 Results and discussion

Using self-supervision in this way, we have been able to learn very strong visual representations using our pretraining method, which our results clearly convey. The highlights of our results, however, are the increase in performance between clip and video level predictions, as well as the boost we receive from self-supervised pretraining.

On both datasets, our system is closer to the fully supervised upper bound in the video level predictions than it is on the clip level. We attribute this performance gain to the fact that our network seemed to learn features that are more related to the relationship of sound and motion in addition to semantics. This intuition arises from looking at numerous visualizations, such as CAM activations in Figure 4 — for instance, the player's head only moves when it turns marginally, or swimmer's body is only lit up while he is actively splashing. While only qualitative, this type of patterns, where the activations are higher around the moving objects that make a sound, appear time and time again.

On both datasets, our pretraining provides a significant boost over a network trained from scratch (+16.7% on UCF101 and +13.0% on HMDB51 for MC3). As expected, making use of Kinetics action labels yields further boost, but the performance gaps are not too large, and may potentially be bridged by making use of a larger pretraining dataset since no manual cost is involved for our procedure.

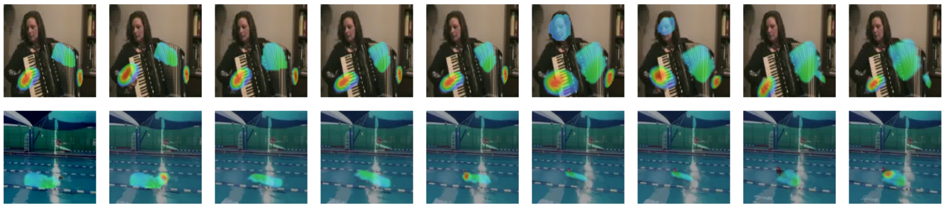


Figure 4: Sound and movement localization can be achieved by looking at grad-CAM activations overlay over video frames, with threshold applied to avoid random noise at low intensity. An interesting note is that the subject's head does not show up activated until it marginally moves in the frame.

3.4 Multisensory extension of action recognition

Following the example of Owens et al. [12] we combine our learned features and use them in conjunction in order to obtain a result on action recognition. Similar to AVC task, we concatenate audio and visual features and add two fully connected layers (512, 512; 512, *output class number*), and then fine-tune the entire system. In only 20 epochs we achieved the same level of performance as we did with our video stream only, and after fine-tuning for 30 epochs, our model was performing more than a random chance better (see Table 6 for comparison with other self-supervised and multisensory methods). We suspect that even better results can be achieved with more sophisticated fusion methods and larger pretraining datasets.

Table 6: Comparison of self-supervised models on UCF-101 [21]. Fully supervised state-of-the-art is at the bottom.

Model	Accuracy
Purushwalkam et al. [23]	55.4%
Owens et al. (vision only) [12]	77.6%
Ours (vision only)	85.8%
Owens et al. (multisensory) [12]	82.1%
Ours (multisensory)	**87.0%**

4 Conclusion

We have shown that optimizing the correspondence of audio-visual representations can be used to learn powerful representation in both audio and video domains. Our audio representations were powerful even when trained on a relatively small dataset, such as Kinetics, and surpassed all self-supervised methods on different scene classification benchmarks. Similarly, using additional information from audio allowed for a powerful initialization for action recognition task on videos, setting respectable results on both UCF101 and HMDB51 benchmarks and setting the state-of-the-art for self-supervised methods. Finally, we show that training with a temporal aspect is crucial for a success on downstream tasks with a temporal domain (such as video). This can be seen in our summary Table 7, where we can clearly evaluate the performance benefit of features trained with spatiotemporal features over the single-frame approach of L3Net [9]. Furthermore, the table shows the implications of AVTS performance on other downstream tasks, and the importance of curriculum learning.

Table 7: Implications of AVTS performance on the quality of the features. All models are pretrained on Kinetics unless otherwise noted.

Method	AVTS	ESC50[1]	DCASE2014[1]	HMDB51	UCF101
AVTS - curriculum	78%	82%	94%	57%	86%
AVTS - combined	69%	71%	89%	46%	77%
L3 Net [9]	74%	79%	93%	40.2%[2]	72.3%[2]

Our network seems to learn more than just semantics, and we argue that what it learns are aspects of motion, that are informed not by action labels but by the innate correspondence to audio. This, we believe is an inspiring new frontier of research, as it shows that we can augment the datasets that already exist, create larger datasets, and set a strong benchmark for what comes ahead. We hope that the ability to easily use this multisensory information in a classification setting indicates that we are on the right track in pursuit of improving video understanding.

References

[1] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[2] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 248–255. IEEE, 2009.

[1] Models pretrained on Kinetics-Sound [8] for fair comparison with L3 Net [9].

[2] Based on our own implementation of L3Net.

[3] Du Tran, Lubomir Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. Learning spatiotemporal features with 3d convolutional networks. In *Computer Vision (ICCV), 2015 IEEE International Conference on*, pages 4489–4497. IEEE, 2015.

[4] Will Kay, Joao Carreira, Karen Simonyan, Brian Zhang, Chloe Hillier, Sudheendra Vijayanarasimhan, Fabio Viola, Tim Green, Trevor Back, Paul Natsev, Mustafa Suleyman, and Andrew Zisserman. The kinetics human action video dataset. *CoRR*, abs/1705.06950, 2017.

[5] Jort F. Gemmeke, Daniel P. W. Ellis, Dylan Freedman, Aren Jansen, Wade Lawrence, R. Channing Moore, Manoj Plakal, and Marvin Ritter. Audio set: An ontology and human-labeled dataset for audio events. In *Proc. IEEE ICASSP 2017*, New Orleans, LA, 2017.

[6] Bernard Ghanem Fabian Caba Heilbron, Victor Escorcia and Juan Carlos Niebles. Activitynet: A large-scale video benchmark for human activity understanding. In *CVPR*, 2015.

[7] Joao Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4724–4733. IEEE, 2017.

[8] Yusuf Aytar, Carl Vondrick, and Antonio Torralba. Soundnet: Learning sound representations from unlabeled video. In *Advances in Neural Information Processing Systems*, pages 892–900, 2016.

[9] Relja Arandjelović and Andrew Zisserman. Look, listen and learn. *arXiv preprint arXiv:1705.08168*, 2017.

[10] R Arandjelovic and A Zisserman. Objects that sound. arxiv preprint. *arXiv preprint arXiv:1712.06651*, 3(10), 2017.

[11] Joon Son Chung and Andrew Zisserman. Out of time: automated lip sync in the wild. In *Asian Conference on Computer Vision*, pages 251–263. Springer, 2016.

[12] Andrew Owens and Alexei A Efros. Audio-visual scene analysis with self-supervised multisensory features. *arXiv preprint arXiv:1804.03641*, 2018.

[13] Hang Zhao, Chuang Gan, Andrew Rouditchenko, Carl Vondrick, Josh McDermott, and Antonio Torralba. The sound of pixels. *arXiv preprint arXiv:1804.03160*, 2018.

[14] Yoshua Bengio, Jerome Louradour, Ronan Collobert, and Jason Weston. Curriculum learning. ICML, 2009.

[15] Gregory Koch, Richard Zemel, and Ruslan Salakhutdinov. Siamese neural networks for one-shot image recognition. In *ICML Deep Learning Workshop*, volume 2, 2015.

[16] Du Tran, Jamie Ray, Zheng Shou, Shih-Fu Chang, and Manohar Paluri. Convnet architecture search for spatiotemporal feature learning. *CoRR*, abs/1708.05038, 2017.

[17] Du Tran, Heng Wang, Lorenzo Torresani, Jamie Ray, Yann LeCun, and Manohar Paluri. A closer look at spatiotemporal convolutions for action recognition. *arXiv preprint arXiv:1711.11248*, 2017.

[18] Karol J. Piczak. ESC: Dataset for Environmental Sound Classification. In *Proceedings of the 23rd Annual ACM Conference on Multimedia*, pages 1015–1018. ACM Press.

[19] Dan Stowell, Dimitrios Giannoulis, Emmanouil Benetos, Mathieu Lagrange, and Mark D Plumbley. Detection and classification of acoustic scenes and events. *IEEE Transactions on Multimedia*, 17(10):1733–1746, 2015.

[20] Hardik B Sailor, Dharmesh M Agrawal, and Hemant A Patil. Unsupervised filterbank learning using convolutional restricted boltzmann machine for environmental sound classification. *Proc. Interspeech 2017*, pages 3107–3111, 2017.

[21] Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah. Ucf101: A dataset of 101 human actions classes from videos in the wild. *arXiv preprint arXiv:1212.0402*, 2012.

[22] Hilde Kuehne, Hueihan Jhuang, Rainer Stiefelhagen, and Thomas Serre. Hmdb51: A large video database for human motion recognition. In *High Performance Computing in Science and Engineering '12*, pages 571–582. Springer, 2013.

[23] Senthil Purushwalkam and Abhinav Gupta. Pose from action: Unsupervised learning of pose features based on motion. *arXiv preprint arXiv:1609.05420*, 2016.