

# Multitask Learning to Improve Egocentric Action Recognition

Georgios Kapidis<sup>1,2</sup>, Ronald Poppe<sup>2</sup>, Elsbeth van Dam<sup>1</sup>, Lucas Noldus<sup>1</sup>, Remco Veltkamp<sup>2</sup>

<sup>1</sup> Noldus Information Technology, Wageningen, The Netherlands

{georgios.kapidis, elsbeth.vandam, lucas.noldus}@noldus.nl

<sup>2</sup> Department of Information and Computing Sciences, Utrecht University, Utrecht, The Netherlands

{g.kapidis, r.w.poppe, r.c.veltkamp}@uu.nl

## Abstract

*In this work we employ multitask learning to capitalize on the structure that exists in related supervised tasks to train complex neural networks. It allows training a network for multiple objectives in parallel, in order to improve performance on at least one of them by capitalizing on a shared representation that is developed to accommodate more information than it otherwise would for a single task. We employ this idea to tackle action recognition in egocentric videos by introducing additional supervised tasks. We consider learning the verbs and nouns from which action labels consist of and predict coordinates that capture the hand locations and the gaze-based visual saliency for all the frames of the input video segments. This forces the network to explicitly focus on cues from secondary tasks that it might otherwise have missed resulting in improved inference. Our experiments on EPIC-Kitchens and EGTEA Gaze+ show consistent improvements when training with multiple tasks over the single-task baseline. Furthermore, in EGTEA Gaze+ we outperform the state-of-the-art in action recognition by 3.84%. Apart from actions, our method produces accurate hand and gaze estimations as side tasks, without requiring any additional input at test time other than the RGB video clips.*

## 1. Introduction

Human activity recognition from video is a growing field of computer vision that promises real-time and large-scale behavior recognition and automated analysis. Activity recognition applies to both the third and first-person vision domains, incorporating the distinct visual characteristics of each case. Third-person videos tend to capture the full range of motions of the human body from a static point of view. The viewing angle in egocentric videos matches that of the human performing the activity, providing a unique, moving perspective of the scene [20]. At the same time egocentric videos usually offer a clear view of the camera

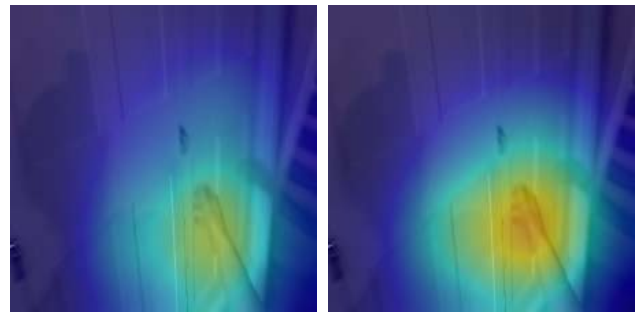


Figure 1. Visualizing the class activation maps [45] for an instance of class ‘open’ from EPIC-Kitchens [7]. Left: Multi-Fiber Network (MFNet) [6] trained end-to-end for the single task of classifying short clips into actions. Right: MFNet trained to additionally predict one (x,y) coordinate for each hand. Training with the hand coordinates as the extra task leads to a greater inclusion of the right hand area into the class activation map.

wearer’s hands [9], which in many cases are essential for the execution of an activity. An outlook of the objects manipulated by human hands promises additional cues about the performed activity culminating to improved recognition performance [9, 26, 1].

Recent methods for video activity recognition employ convolutional (CNN) and recurrent (RNN) deep neural network structures to capture the information from RGB images or video frames, regardless of the viewing perspective [22, 8]. More recent approaches use 3D convolutions [4] to incorporate the temporal information that resides in frame sequences. Occasionally, they are enhanced with attention schemes [12] to select specific features or frames as more informative for an activity. In order to expand the feature space, the network input can be augmented with additional data modalities. These can be optical flow [40], depth [49] or input segmentation masks around interesting areas [52]. The information from them aims to guide the network towards learning more activity-specific features that it might otherwise have missed. In order to incorporate the supplementary inputs, networks comprise multiple streams, i.e.

parts of the original structure are copied in the number of modalities, trained individually (e.g. [54]) and their results are combined at a later stage. The multi-stream approach is associated with the combination of the individual feature sets towards an extended and more expressive representation from which activities are inferred.

A related but fundamentally different concept that we employ for this work is that of Multitask Learning (MTL) [5]. The idea behind MTL is to train a neural network with multiple related objectives (tasks) while sharing as much as possible of a common network structure [5]. Branch diversification occurs only for the task-specific output layers and there are as many output layers (branches) as there are learnable tasks spawning from the main network block, increasing the dimensionality of the output.

MTL is conceptually the opposite from multi-stream approaches, since the additional information is not used as input but is expressed as the output of the network and is only required for supervision. The significant merit of MTL over multi-stream methods is that the additional information is only needed during training and what would otherwise come from the additional input modalities is already incorporated in the network weights at test time. For example, in the video domain, the input of a network remains the same set of RGB images regardless of the number of tasks.

The premise of MTL is that by combining the objectives of related tasks in the same network, we can benefit from their structural commonalities. This is the case because the weights of the shared network block aim to jointly encapsulate each task's representation requirements. When these are complementary, they enhance the inputs of the task-specific output layers. Then, inference can be improved for all or some of them or just the one that we focus on the most, by saving its best performing weights [5].

In this work, we utilize MTL to improve action classification performance in egocentric videos. We are motivated by the idea that hands are critical for the comprehension of egocentric actions but it remains difficult for networks to capture this delicate motion information. In Figure 1, we show that by having the network learn hand regions explicitly as an extra task in addition to the actions, we steer it to produce activation maps that cover the corresponding hand areas to a greater extent. Eventually, incorporating these areas also improves the action classification results.

We experiment with egocentric datasets EPIC-Kitchens [7] and EGTEA Gaze+ [25] by explicitly utilizing the location of hands, gaze and other signals towards actions. We leverage the motion and visual attention information that is present in the hand movements and the gaze of the camera wearer, respectively, which have proven descriptive for predicting egocentric actions on their own [9, 26, 21]. In addition, we show that when complementary classification tasks are added during training, performance improves further.

The contributions of this paper are the following:

- An MTL scheme that extends 3D CNNs [6] and functions with an arbitrary number of output tasks.
- Experiments demonstrating that MTL improves on egocentric action classification over single-task learning (STL) baselines without requiring any additional information at test time, other than the input video.
- Experiments generalizing our MTL scheme to a number of related classification and coordinate estimation tasks that improve egocentric action classification.

In Section 2 we review recent work on action classification and MTL. In Section 3 we develop our MTL pipeline for an arbitrary number of tasks. In Section 4 we document our experiments on two egocentric video datasets. In Section 5 we discuss our findings and in Section 6 we conclude.

## 2. Related work

In Section 2.1 we discuss feature-based egocentric action recognition approaches. We continue with the more recent deep network advances for activity recognition from the third person perspective in Section 2.2 and their expansion into egocentric in Section 2.3. In Section 2.4 we discuss related work from the perspective of MTL.

### 2.1. Feature-based egocentric action recognition

The hands, the manipulated objects, ego-motion and their interrelationships have been established as some of the most prominent characteristics for egocentric action recognition [9, 44, 20]. In this observation lies the origin of the hand-crafted feature approaches that prevail in earlier works in egocentric vision.

Fathi *et al.* [9] use hand and object segmentations to infer actions and based on feedback from the latter improve the initial hand and object detections. The importance of the detected objects and the interactions between them for activities is highlighted in [32], where activity recognition improves using additional information about objects being either passive or actively engaged with in the scene. In [10], the gaze of the camera wearer is used to define the salient areas in first-person views, recognizing that egocentric actions are further correlated with modalities that describe human attention in the video. Global and local motion are considered in [36] to produce features that describe egocentric actions. Feature-based egocentric action recognition is analyzed in [26] where the importance of motion and object cues, hand, head movements and gaze are evaluated in various combinations. A review of these approaches appears in [30]. In our work, we do not use an explicit feature-based representation of modalities for the input, but use data from these modalities as supervision to learn actions.

## 2.2. Advances in third-person activity recognition

Recent work in third-person vision [33, 46] has seen the successful employment of deep network approaches. We highlight the work of Karpathy *et al.* [22] who use 2D CNN architectures to classify video frames and in order to incorporate information from multiple frames, explore various fusion schemes to enhance the classification output. Other approaches include the use of two-stream networks [40, 54, 13] that capture appearance and motion in images with spatial and motion streams trained on single or multiple frames concurrently. Further attempts to take advantage of the temporal consistency in videos consider recurrent units following frame-wise feature-extracting CNNs [8, 27]. More recent approaches in video activity recognition use 3D CNNs [50, 4, 51, 6]. Here, video frames are modelled as a result of convolutional kernels being learned not only on the spatial dimension of images, but also on the changing pixel values in frame sequences. We find that using a 3D CNN to capture patterns in the temporal dimension also works well for egocentric action recognition.

## 2.3. Advances in first-person activity recognition

**Spatial networks** A large volume of work in egocentric vision already incorporates these advances. A CNN appearance feature extractor from images is used to categorize egocentric actions in [37]. In [55], CNN features from RGB images are used to produce embeddings that are semantically linked among videos and are used as the basis to model relationships between objects and actions and classify them. In [28], a two-stream network is trained to capture appearance and motion features. The appearance stream is pre-trained for hand segmentation and is finetuned for object localization, to find the interesting regions of the image. The motion stream is trained on optical flow and both feature representations are combined with late fusion to predict short-term actions, objects and activities. Furthermore, in [41], a three-stream architecture trained on egocentric features from hand masks, head motion and saliency maps for the first stream, and raw RGB images and optical flow maps for the other two is adapted to recognize actions. We also highlight the improvements from specific egocentric cues such as hand movements and gaze-based visual saliency on top of the image-based features, but from a contrasting perspective. Instead of using them as input, we consider them as additional learnable tasks with the advantage of not requiring the extra information at test time.

**Spatio-temporal with recurrence** Approaches utilizing recurrent networks following 2D CNNs include [53, 15, 43, 24]. CNN features are propagated to recurrent units for temporal action proposal generation [15], action related scene identification [53] and action recognition [43, 24].

**Recurrent with attention** The temporal aspect of videos is further studied with recurrent attention mechanisms [3, 39, 48, 47, 25, 11, 56] that act to find the most informative parts in images (spatial attention) or the most informative frames throughout videos (temporal attention). An encoder-decoder scheme is described in [3] for textual description of videos. Here, the current event's frames are encoded into CNN features and modelled temporally with LSTM. From the current and previous step's embedding, an attention mechanism selects the features that will be decoded as the optimal textual description of the current activity. The attention mechanism in [39] focuses on the frames that carry the action specific information by learning the associations between the input gaze, the detected objects and the segmented hands. The combined focus on these regions allows the network to discard redundant frames of the input video segment that would otherwise obfuscate the prediction task. Spatial attention is considered in [48] where the important regions from every frame are given as input to an LSTM for action recognition, whereas in [47] spatial attention is further correlated sequentially through continuous frames. In [56], attention is based on video specific spatial features (such as person detections in third-person videos and objects with motion for egocentric), which are calculated intermittently over the course of a video. These temporally examined spatial features introduce past information to predict the current action. In [11], the attention mechanism weighs the importance of the input modalities to select optimal features for action anticipation and recognition. In contrast to explicit attention mechanisms, we are using the additional supervisory tasks to enhance the representation in deeper layers by incorporating information from all and learning them together, thus inciting the network to acquire universally useful features.

## 2.4. Multitask learning

Training a network for multiple tasks jointly has been shown to improve the performance on all of them or at least the main task, as long as they share a conceptual similarity [5] or they are not competing [38]. In [38], the problem of loss function weighing is analyzed, to train for multiple tasks efficiently with an optimization scheme that searches for the optimal set of network parameters for the best trade-off among tasks. Instead, we weigh all tasks equally with our focus on properly selecting related tasks in advance.

MTL has been applied to computer vision problems such as joint object and action detection [19], object detection and segmentation [14] and boundary, surface normal and saliency estimation together with object segmentation and detection [23]. In [57], the relationship between tasks is modelled in a latent space to transfer knowledge between them and reduce the number of required training samples. MTL in egocentric vision appears in [1, 28, 25, 18, 29, 47].

In [1], from an RGB video input, multiple network branches learn activities, object proposals and segmentations, but with large parts of the network trained independently. In [28], an object and an action learning task are combined to produce an activity prediction (as a combination of the two), but the network layers except the last one are not sharing parameters. In [25], the network learns a gaze map which is used to pool from the activations of the final feature map for actions, thus the two tasks affect each other. Similarly, in [18], a prior gaze estimation is used to influence the action prediction, which in turn affects the final gaze prediction. Training takes place jointly, but internally, each network part is deployed for a specific task, without parameter sharing. Another example of multitask learning in egocentric vision is from [29] with joint learning of activities and energy expenditures from video. Still, the input to the network is multimodal (video and accelerometer signals) and each stream is trained individually with parameter sharing only during a late fusion stage. Finally, in [47] the action task is augmented by a verb and a noun learning task similar to ours, however, the bias of the action classifier is applied to the secondary tasks and alters their output explicitly. In this work, we do not attempt to influence task outputs explicitly, but employ parameter sharing in every network layer, down to the last step before the final output, as a way to induce implicit information sharing between them.

### 3. Methodology

In Section 3.1, we describe the process to adapt a network from single to multitask and in Section 3.2 the output layers with their individual loss functions for the tasks we consider. In Section 3.3, we discuss the details of the coordinate prediction layer and its application to 3D CNNs to model the progression of movements through time.

#### 3.1. Multitask network structure

In Figure 2 we visualize our network architecture for multitask training. The backbone of the network is a feature extractor (the shared network block), upon which the task-specific output layers are attached. We represent the feature extracting network block with  $g(x; \theta)$ , where  $x$  is an input data point from input space  $\mathcal{X}$  and  $\theta$  are the parameters of  $g$ . For each task  $t$ , we define an output function  $f_t(g(x); \theta_t)$ , where  $\theta_t$  are the parameters of the task-specific layer and  $t \in \mathcal{T}$ , with  $\mathcal{T}$  being the set of tasks. For this work, function  $g$  is approximated with a 3D Convolutional Neural Network and the input space  $\mathcal{X}$  is defined as a set of RGB images sampled from a video clip.

#### 3.2. Task-specific output layers

In order to train the network, we formulate the loss function based on the number and types of tasks it encapsulates. We perform MTL with two types of tasks: classification and

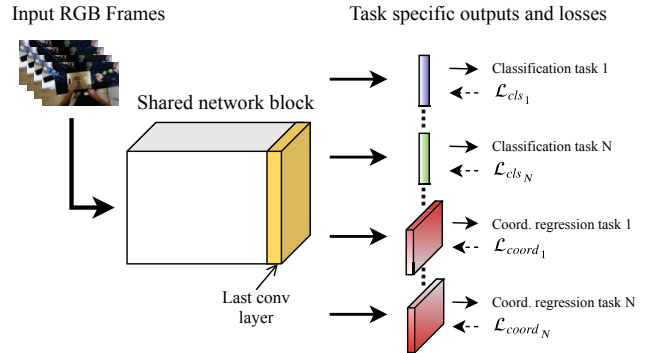


Figure 2. MTL Network Structure. The shared network block can be any convolutional network that extracts features from the input. Each task-specific output layer is plugged to the ‘last conv layer’. They are independent from the others and their parameters are trained individually. However, all output layers use the feature representation produced by the shared part of the network as input.

coordinate regression. For a classification task  $i$ ,  $\mathcal{L}_{cls_i}$  is the categorical cross-entropy loss. For a coordinate regression task  $j$ ,  $\mathcal{L}_{coord_j}$  is defined as the Differentiable Spatial to Numerical Transform (DSNT) loss from [31], explained in Section 3.3. The full loss function  $\mathcal{L}$  is defined as

$$\mathcal{L} = \sum_i \mathcal{L}_{cls_i} + \sum_j \mathcal{L}_{coord_j}. \quad (1)$$

During training the value of  $\mathcal{L}$  is not propagated through each task-specific layer, but each task layer  $t$  produces gradients with respect to its individual loss, hence its parameters  $\theta_t$  are not affected by the remaining tasks. Finally, the gradients from all the output layers are summed and back-propagated through  $g$ .

#### 3.3. Coordinate prediction

Our approach for predicting coordinates stems from the numerical coordinate regression layer introduced in [31]. It enables a 2D CNN to output an  $(x, y)$  coordinate without using a fully connected layer, thus ensuring spatial invariance in the predicted coordinate [31]. Instead, it relies on an additional convolutional layer that predicts a heatmap  $Z$  of shape  $m \times n$ . The softmax activation is applied on  $Z$ , such as  $\hat{Z} = \sigma(Z)$ , to create a 2D probability distribution, which is passed through the Differentiable Spatial to Numerical Transform (DSNT) layer to become a coordinate.

In DSNT,  $\hat{Z}$  is discretized by calculating its Frobenius inner product for each dimension, against two uniformly distributed vectors with values in  $[-1, 1]$ , shaped  $m \times 1$  and  $1 \times n$  respectively and copied over their singular dimension ( $n$  and  $m$  times) to become matrices the shape of  $\hat{Z}$ . The output value of the Frobenius inner product for each matrix is the respective coordinate value with sub-pixel precision in the range  $[-1, 1]$ . This process preserves differentiability through the layer and allows gradient flow from a

loss function directly associated with the error in coordinate space instead of the error in heatmap space.

The coordinate loss  $\mathcal{L}_{coord}$  is the Euclidean distance between the predicted ( $c_p$ ) and the expected ( $c_{gt}$ ) coordinate with an added regularization factor  $\lambda = 0.5$  to smooth the gradients around the prediction, i.e.

$$\mathcal{L}_{coord} = \lambda \mathcal{L}_{euc}(c_p, c_{gt}) + (1 - \lambda) \mathcal{L}_{reg}(\hat{Z}), \quad (2)$$

where the Euclidean loss is

$$\mathcal{L}_{euc} = \|(c_p, c_{gt})\|_2 \quad (3)$$

with  $c_p = DSNT(\hat{Z})$  and the regularization loss is

$$\mathcal{L}_{reg}(\hat{Z}) = \mathcal{L}_{JS}(\hat{Z}, c_{gt}) = JS(\hat{Z} \parallel \mathcal{N}(c_{gt}, \sigma^2)) \quad (4)$$

based on the Jensen-Shannon divergence.

In order to successfully apply this coordinate regression layer in our setup, we need to account for the output dimensions of the last convolutional layer of the 3D CNN. In the 2D case that is  $B \times C \times m \times n$  with  $B$  the batch size and  $C$  the channel dimension. In the 3D case the output shape extends to  $B \times C \times l \times m \times n$  with  $l$  the added temporal dimension due to the 3-dimensional input. This leads to having  $l$  heatmaps  $Z$  as well as  $l$  coordinate losses (instead of 1) for an RGB clip, which are averaged over the temporal dimension to avoid huge gradients.

## 4. Experiments

In this section we describe our experiments for various task combinations. Besides having improved performance on action recognition we produce accurate detectors for hands and gaze without complicated modules other than the DSNT layers for the coordinate prediction tasks. Finally, we compare against the state-of-the-art.

### 4.1. Datasets

We use EPIC-Kitchens [7] and EGTEA Gaze+ [25] for our experiments. EPIC is a collection of 432 videos by 32 participants, performing kitchen-related activities in their homes. We use the publicly available training split which consists of 272 videos by 28 participants (28,470 action segments). It is annotated with start and end times and 125 verb and 352 noun class labels, the valid combinations of which are 2,521 actions. We partition the fully annotated training split into custom training and validation splits with participants 1-29 used for training (26,375 clips) and 30-31 for validation (2,095 clips). EGTEA consists of 86 videos of 32 people in 7 scenarios of food preparation activities in kitchens. The videos are cropped into 10,321 clips based on action segment annotations. The dataset comes with three predefined training and testing splits, with the first one comprising 8,299 and 2,022 clips, respectively. We use this split

unless stated otherwise, to train and evaluate at the clip level [25]. Each clip is labeled from 19 verbs and 53 nouns and their 106 valid action combinations in the dataset. In addition, the dataset is complemented with a gaze annotation for every video frame, which consists of an  $(x, y)$  coordinate and its type (fixation, saccade or unknown).

**Hand locations** One of the tasks we consider for MTL is hand coordinate prediction. Similar to gaze annotations, this task requires a supervision signal for the hand locations on each frame. To accommodate our experiments we use the egocentric hand detection, tracking and identification algorithm from [21] to produce hand location information for every video frame. It uses [34] to detect hand bounding boxes, [2] to track them through time and hand-crafted priors to remove false-positives and identify between left and right hands. We further modify [21] to track the top right area of the left hand bounding box and the top left area of the right hand leading to coordinates that more accurately pinpoint the hands instead of the forearms.

### 4.2. Training and evaluation

For the shared network block of Figure 2 we employ the Multi-Fiber Network (MFNet) [6]. It contains 3D convolutional layers in its structure to capture spatio-temporal information from frame sequences and uses a relatively low number of parameters (8M) and computational resources (11.1 GFLOPs), which leads to an efficient training scheme for large video datasets. For all our experiments we use weights pretrained on Kinetics [4] and retrain the full network structure end-to-end on the respective dataset.

We train with a triangular cyclical learning rate [42] policy that shifts learning rate from  $5 \times 10^{-4}$  to  $5 \times 10^{-3}$  and back in 20 epochs. For optimization we use stochastic gradient descent with Nesterov momentum (0.9) and weight decay ( $5 \times 10^{-4}$ ). We input a sequence of 16 frames, randomly scaled to  $256 \times 256$  and cropped to  $224 \times 224$ . Frames are uniformly sampled from a 32-frame window that starts at a random point of an action video segment and does not exceed its last frame. Batch size is 32 for our setup with two Nvidia 1080Ti GPUs, training lasts for 60 epochs and results are reported for the best performing epoch for the main task (early stopping). To evaluate we sample uniformly 16 frames from a window of 32, centered around the temporal center of an action segment. We resize them to  $256 \times 256$  and use the  $224 \times 224$  center crop as the network input.

### 4.3. Results on EPIC-Kitchens

Our results on EPIC-Kitchens are summarized in Table 1. Initially, we train the STL baseline with verbs for supervision. Then, we combine verbs with nouns and hands as separate tasks. Training for hands together with verbs (V+H) increases Top1 accuracy to 49.31% (+0.75). Adding

Tasks	Top1 Acc. (%)			Top5 Acc. (%)			Avg class Prec. (%)			Avg class Rec. (%)		
	Actions	Verbs	Nouns	Actions	Verbs	Nouns	Actions	Verbs	Nouns	Actions	Verbs	Nouns
V	-	48.57	-	-	78.32	-	-	34.39	-	-	25.29	-
V + H	-	<b>49.31</b>	-	-	<b>78.80</b>	-	-	29.85	-	-	25.68	-
V + N + H	-	47.47	<b>27.6</b>	-	78.37	<b>51.19</b>	-	27.80	21.43	-	23.61	18.80
A	18.48	-	-	36.20	-	-	2.76	-	-	2.67	-	-
A + V + H	18.58	49.05	-	<b>38.82</b>	78.75	-	2.89	28.43	-	2.87	23.23	-
A + V + N + H	<b>19.29</b>	48.9	27.27	35.91	78.18	47.85	3.25	29.31	22.68	3.04	24.03	17.84

Table 1. Multitask learning results on EPIC-Kitchens. The first column shows the trained tasks for a model: Actions (A), Verbs (V), Nouns (N) and Hands (H). We report Top1 and Top5 accuracy on our validation set. Average class precision and recall are reported for many-hot verbs, nouns and actions. Many-hot verbs and nouns have more than 100 instances in our training set. Many-hot actions are the valid combinations of many-hot verbs and nouns with at least one instance in the training set, following [7].

nouns (V+N+H) harms verb Top1 by 1.1% but produces our best performing noun classifier.

In the EPIC-Kitchens literature [7] verb and noun predictions are combined following their individual inference stages and are later synthesized into an action prediction. In our MTL scheme we train for the action task explicitly, i.e. the 2,521 valid verb and noun combinations. Having actions, verbs and hands for supervision (A+V+H) leads to 49.05% for verbs, improving on the STL baseline by 0.48% and additionally using the nouns (A+V+N+H) still improves from STL (+0.33). However, both cases do not improve as much as with only the hands, implying a conflict between the extra tasks. On the other hand, if we consider actions as the main task, the addition of verb, noun and hand learning tasks will only improve on the action STL baseline reaching 19.29% (+0.81 from A and +0.71 from A+V+H).

#### 4.4. Results on EGTEA Gaze+

In Table 2 we delineate our results as the network moves from one to multiple tasks in the EGTEA Gaze+ dataset. We establish the action STL baseline (A) at 63.75% Top1 accuracy. Next, we train using additional supervision from verbs and nouns (A+V) and (A+V+N) and reach 67.80% (+4.05) and 68.00% (+4.25), respectively. For further experiments we utilize coordinate regression layers to train on gaze points and hand tracks. We see that with either task we improve in both Top1 and mean class accuracy over STL; A+G is 66.59% (+2.84) and A+H is 67.46% (+3.71). Further improvements stem from training for all classification tasks together with gaze or hand prediction. A+V+N+G reaches Top1 68.74% (+4.99) and A+V+N+H is 68.20% (+4.45). The attempt to combine gaze and hand coordinate regression tasks only with actions shows that the two coordinate tasks are competing to influence the shared representation and have the smallest improvement over the STL baseline with A+G+H Top1 at 66.12% (+2.37). However, when all the classification and coordinate prediction tasks are present in one model (A+V+N+G+H), we achieve our best Top1 accuracy at 68.99% (+5.24) and our best mean class accuracy 61.40% (+6.05 from STL at 55.35%).

#### 4.5. Comparison to the state-of-the-art

In Tables 3 and 4 we compare with the state-of-the-art in action recognition for EPIC-Kitchens and EGTEA Gaze+, respectively. For EPIC we demonstrate slightly lower but comparable performance to the top methods for the seen (s1) and unseen (s2) test splits, by requiring only a fraction of the input. For example [11] requires RGB and flow at test time and [56] utilizes knowledge from past video segments, in effect having a larger temporal view of the action. However, we still outperform the attention mechanism of [47]. For EGTEA, we test against several methods, for different metrics. Top1 recognition accuracy for the first split at the clip level is reported in [18] (55.63%) and in [48] (62.17%) where we improve by 13.36% and 6.82% respectively. Li *et al.* [25] report 47.71% mean class accuracy on the first split at the clip level (and 53.3% on the video level). Our method depending on the task combination reaches 58.91% up to 61.4% (+11.2 to +13.69 respectively).

For a more elaborate comparison on EGTEA Gaze+, we train the A+V+N+G+H model for splits 2 and 3 and average the Top1 accuracy over all splits. We achieve 65.7% Top1 accuracy which is the highest among the reported values by a margin of 3.84%. For future reference we also report the mean class accuracy averaged over the three splits (57.6%).

An additional interesting scope from EGTEA is gaze estimation. Since a number of our models are able to predict gaze on the input frames, we proceed to evaluate it with two standard metrics in the literature: Average Angle Error in degrees (AAE) and Area Under the Curve (AUC) [35] following [18]. For evaluation we use only the frames from the clips of the first test split for which after resizing and cropping to  $224 \times 224$  there is a valid ground truth gaze point in this area, regardless of the gaze type. This leads to the evaluation of 177,292/206,649 (85.79%) frames from 2,022 clips (the remaining frames are not considered). The results are shown in Table 5. We discover that the gaze estimation techniques which are explicitly designed to model gaze through elaborate attention mechanisms such as [18] achieve lower angular error ( $-3.11^\circ$ ) although our model

Tasks	Top1 Acc. (%)			Top5 Acc. (%)			Mean Cls Acc. (%)		
	Actions	Verbs	Nouns	Actions	Verbs	Nouns	Actions	Verbs	Nouns
A	63.75	-	-	91.05	-	-	55.35	-	-
A + V	67.80	79.03	-	91.89	99.41	-	59.15	<b>79.44</b>	-
A + V + N	68.00	78.98	78.93	91.94	99.31	96.24	59.67	78.24	72.06
A + G	66.59	-	-	91.54	-	-	59.44	-	-
A + H	67.46	-	-	91.99	-	-	59.78	-	-
A + G + H	66.12	-	-	90.54	-	-	58.91	-	-
A + V + N + G	68.74	78.14	<b>79.13</b>	91.59	99.41	<b>96.54</b>	60.34	79.29	72.03
A + V + N + H	68.20	<b>79.18</b>	77.94	<b>92.24</b>	<b>99.51</b>	96.34	60.13	79.34	71.1
A + V + N + H + G	<b>68.99</b>	79.08	79.03	91.74	99.26	96.39	<b>61.40</b>	77.40	<b>72.49</b>

Table 2. Multitask learning results on EGTEA Gaze+. The first column shows the names of the supervised tasks: Actions (A), Verbs (V), Nouns (N), Gaze (G) and Hands (H). We report Top1, Top5 and Mean class accuracy on the first split of the EGTEA Gaze+ test set.

(A+G+H) improves over [16] and is very close to [25]. Furthermore, considering AUC, our model is second best to [18] with a -0.06 margin. The two metrics imply that our method is able to produce gaze predictions that lie in the vicinity of the ground truth (high AUC) but with an angular offset with respect to the exact ground truth gaze. In Figure 3 we show and qualitatively assess gaze and hand predictions. The images show both the predicted saliency in heatmap form as well as its transformation into a single point per frame for gaze and each hand.

Method	Top1 Acc. (%)			Top5 Acc. (%)		
	Actions	Verbs	Nouns	Actions	Verbs	Nouns
<b>Test s1 (Seen kitchens)</b>						
TSN [7]	20.54	48.23	36.71	39.79	84.09	62.32
LSTA [47]	30.33	59.55	38.35	49.97	85.77	61.49
<i>Ours (all tasks)</i>	29.73	56.00	40.15	50.95	87.06	64.07
RU [11]	<b>33.06</b>	56.93	43.05	<b>55.32</b>	85.68	67.12
LFB [56]	32.70	<b>60.00</b>	<b>45.00</b>	55.30	<b>88.40</b>	<b>71.80</b>
<b>Test s2 (Unseen kitchens)</b>						
TSN [7]	10.89	39.40	22.70	25.26	74.29	45.72
LSTA [47]	16.63	47.32	22.16	30.39	77.02	43.15
<i>Ours (all tasks)</i>	17.86	45.99	26.25	35.68	<b>77.98</b>	50.19
RU [11]	19.49	43.67	26.77	37.15	73.30	48.28
LFB [56]	<b>21.20</b>	<b>50.90</b>	<b>31.50</b>	<b>39.40</b>	77.60	<b>57.80</b>

Table 3. Comparison on action recognition against state-of-the-art methods on EPIC-Kitchens. Our method is consistently close to the best performing, while requiring less information at test time.

## 5. Discussion

Our first aim with MTL is to drive the focus of the activation maps around hand regions and their movements. By training for the hand coordinate task we imply greater importance to them and introduce this requirement to the weights of the shared network block via gradient descent. An example of the expected behavior of the network is in Figure 1, where the class activation maps after the last convolution layer cover a larger area of the visible hands.

The task of gaze prediction is similar to hand detection

Method	Split 1		Avg. Splits 1-3	
	Top1	Mean Cls	Top1	Mean Cls
Li <i>et al.</i> [25]	-	47.71	-	-
MCN [18]	55.63	-	-	-
RU [11]	-	-	60.20	-
ego-rnn [48]	62.17	-	60.76	-
LSTA [47]	-	-	61.86	-
<i>Ours (all tasks)</i>	<b>68.99</b>	<b>61.40</b>	<b>65.70</b>	<b>57.60</b>

Table 4. Action recognition comparison on EGTEA Gaze+. We compare against the available values from each paper.

Method	AAE	AUC
SALICON [16]	11.17	0.881
<i>Ours (A+G+H)</i>	8.90	0.926
Li <i>et al.</i> [25]	8.58	0.87
DFG [58]	6.30	0.923
Huang <i>et al.</i> [17]	6.25	0.925
MCN [18]	<b>5.79</b>	<b>0.932</b>

Table 5. Gaze estimation comparison on EGTEA Gaze+ split 1. AAE lower is better, AUC higher is better. SALICON [16], Li *et al.* [25], DFG [58] and Huang *et al.* [17] are reported from [18].

in that it expects the network to focus on specific regions of the input frames. The difference is that these regions do not necessarily contain the well-structured form of hands, but the salient areas of a scene, which are not predetermined. This limits the ability of region-specific features to become significant making it a dataset- and class-specific quality.

In both datasets, we observe almost consistent improvements over STL with the introduction of hands and other tasks. However, the choice of tasks involves a significant amount of intuition as well as the weighing of their importance in the loss function. In this work, we use a naive weighing mechanism and consider all tasks equal regardless of the loss they incur. When training multitask models for EPIC-Kitchens we notice high values of loss in the classification tasks, which stem from the class imbalance and the large number of action, verb and noun classes. These losses initially affect their individual layers, but the back-

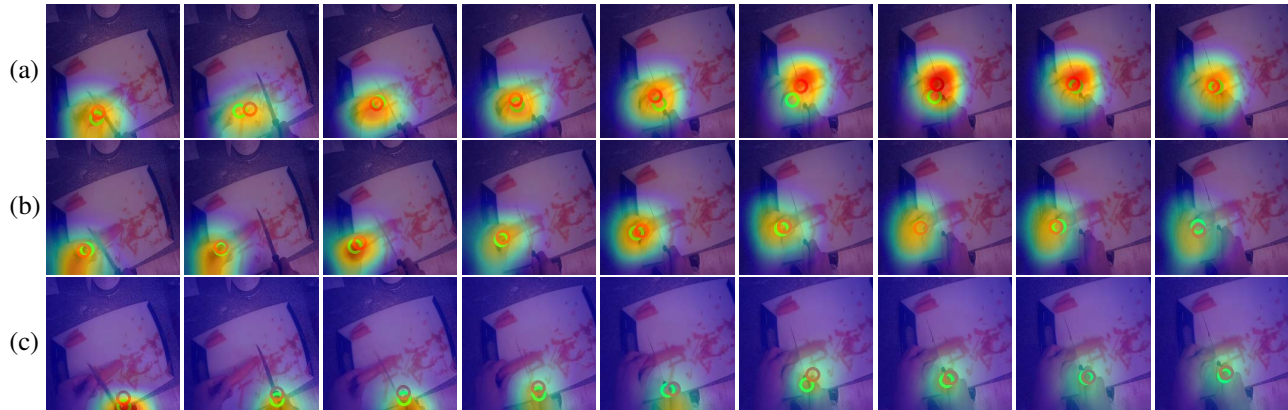


Figure 3. (a) Gaze, (b) left hand, (c) right hand coordinate prediction from the A+G+H model. The green circle represents the ground truth coordinate, the red circle the predicted coordinate and the underlying heatmap the 2D probability distribution  $\hat{Z}$ .

propagated gradients to the shared weights are also higher, affecting the representation in an unbalanced way. In the EPIC-Kitchens results we see that by adding classification tasks with more classes (such as N at V+N+H, or A at A+V+N+H), we get a worse verb classifier. This is caused by the high losses incurred from the added tasks. In certain cases, they act as regularization but when they are too high they can increase training times and even prevent convergence. We believe further research is needed in MTL for video recognition to establish weighing mechanisms such as [38] for a more optimized shared parameter space.

On the other hand, on EGTEA Gaze+, MTL consistently outperforms STL for every task combination. This shows that carefully designing the classification tasks (e.g. fewer classes, balanced dataset) can be mutually beneficial to all, but more importantly to the action task we are most interested in. Incorporating hands in training confirms our initial intuition that motions create a fitting side-task to actions, increasing performance. A possible reason for the higher improvement due to hands on EGTEA compared to EPIC is the presence of hand annotations from the former in the training set of the hand detector of [21]. This, could result in a more accurate synthetic hand dataset for EGTEA. Finally, improvements due to gaze validate the connection between actions and gaze [18, 25] also from the perspective of MTL.

A possible pitfall of MTL is competition among tasks with negative effects on performance. This is possible if tasks are incompatible or if the network is not large enough to create a representation that engulfs the different aspects of information required for each one. The former regards a (lack of) conceptual relevance, for example verbs and nouns on EPIC, or structural, for example classification layers operating differently from coordinate regression ones and possibly requiring a distinct representation in earlier layers. The concept of task compatibility has been studied for other domains in [5] concluding that the degree of assistance from an extra task in learning another cannot be clear

a priori without experimentation. This can be viewed as treating tasks as an additional hyperparameter. An example of task incompatibility with respect to actions is when both gaze and hands are used for action recognition (A+G+H) but lead to worse performance than training individually (A+G, A+H). Adding a task may not improve as much as another combination but can also reduce the expected baseline performance. Here, however, the trade-off is actions and hands contributing towards an optimal gaze detector.

## 6. Conclusions

In this work, we have developed a Multitask Learning scheme for egocentric action recognition that supports a variable number of tasks. We train for actions together with related classification tasks, such as verbs and nouns, and show that performance on one or all of them will improve over the single-task baseline. We further combine classification with coordinate regression tasks to learn the egocentric left and right hand and gaze locations, by predicting coordinate sequences for video segments exploiting the temporal dimension of 3D CNNs. We highlight that having a network estimate coordinates allows it to focus more on certain areas of the activation maps with higher correspondence to the hands or other salient objects in the original image. Our tests on the EPIC-Kitchens show improvements on action recognition performance over single-task. On EGTEA Gaze+ we achieve state-of-the-art performance in action recognition reaching 65.70% surpassing the previous best by more than 3.8%. In addition, with multitask learning we can produce accurate hand detectors, as well as gaze predictors with performance close to state-of-the-art.

## Acknowledgment

This project has received funding from the EU H2020 research and innovation program under the Marie Skłodowska Curie grant agreement No 676157 (ACROSSING).



## References

- [1] F. Baradel, N. Neverova, C. Wolf, J. Mille, and G. Mori. Object Level Visual Reasoning in Videos. In *European Conference on Computer Vision (ECCV)*, pages 106–122, 2018.
- [2] A. Bewley, Z. Ge, L. Ott, F. Ramos, and B. Upcroft. Simple Online and Realtime Tracking. In *IEEE International Conference on Image Processing (ICIP)*, pages 3464–3468, 2016.
- [3] M. Bolaños, Á. Peris, F. Casacuberta, S. Soler, and P. Radeva. Egocentric Video Description based on Temporally-Linked Sequences. *Journal of Visual Communication and Image Representation, Special Issue on Egocentric Vision and Lifelogging Tools*, 50:205–216, 2018.
- [4] J. Carreira and A. Zisserman. Quo Vadis, Action Recognition? A New Model and the Kinetics Dataset. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4724–4733, 2017.
- [5] R. Caruana. Multitask Learning. *Machine Learning*, 28(1):41–75, 1997.
- [6] Y. Chen, Y. Kalantidis, J. Li, S. Yan, and J. Feng. Multi-fiber Networks for Video Recognition. In *European Conference on Computer Vision (ECCV)*, pages 364–380, 2018.
- [7] D. Damen, H. Doughy, G. Maria 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.
- [8] J. Donahue, L. A. Hendricks, M. Rohrbach, S. Venugopalan, S. Guadarrama, K. Saenko, and T. Darrell. Long-Term Recurrent Convolutional Networks for Visual Recognition and Description. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(4):677–691, 2017.
- [9] A. Fathi, A. Farhadi, and J. M. Rehg. Understanding egocentric activities. In *IEEE International Conference on Computer Vision (ICCV)*, pages 407–414, 2011.
- [10] A. Fathi, Y. Li, and J. M. Rehg. Learning to Recognize Daily Actions Using Gaze. In *European Conference on Computer Vision (ECCV)*, pages 314–327, 2012.
- [11] A. Furnari and G. M. Farinella. What Would You Expect? Anticipating Egocentric Actions with Rolling-Unrolling LSTMs and Modality Attention. *CoRR abs/1905.09035*, 2019.
- [12] R. Girdhar and D. Ramanan. Attentional Pooling for Action Recognition. In *31st International Conference on Neural Information Processing Systems*, pages 33–44, 2017.
- [13] G. Gkioxari and J. Malik. Finding action tubes. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 759–768, 2015.
- [14] K. He, G. Gkioxari, P. Dollár, and R. Girshick. Mask R-CNN. In *IEEE International Conference on Computer Vision (ICCV)*, 2017.
- [15] S. Huang, W. Wang, S. He, and R. W. H. Lau. Egocentric Temporal Action Proposals. *IEEE Transactions on Image Processing*, 27(2):764–777, 2018.
- [16] X. Huang, C. Shen, X. Boix, and Q. Zhao. SALICON: Reducing the Semantic Gap in Saliency Prediction by Adapting Deep Neural Networks. In *IEEE International Conference on Computer Vision (ICCV)*, pages 262–270, 2015.
- [17] Y. Huang, M. Cai, Z. Li, and Y. Sato. Predicting Gaze in Egocentric Video by Learning Task-Dependent Attention Transition. In *European Conference on Computer Vision (ECCV)*, pages 789–804, 2018.
- [18] Y. Huang, Z. Li, M. Cai, and Y. Sato. Mutual Context Network for Jointly Estimating Egocentric Gaze and Actions. *CoRR abs/1901.01874*, 2019.
- [19] V. Kalogeiton, P. Weinzaepfel, V. Ferrari, and C. Schmid. Joint Learning of Object and Action Detectors. In *IEEE International Conference on Computer Vision (ICCV)*, pages 2001–2010, 2017.
- [20] T. Kanade and M. Hebert. First-Person Vision. *Proceedings of the IEEE*, 100(8):2442–2453, 2012.
- [21] G. Kapidis, R. Poppe, E. van Dam, L. P. J. J. Noldus, and R. C. Veltkamp. Egocentric Hand Track and Object-based Human Action Recognition. In *IEEE SmartWorld/SCALCOM/UIC/ATC/CBDCCom/IOP/SCI*, page (to appear), 2019.
- [22] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei. Large-Scale Video Classification with Convolutional Neural Networks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1725–1732, 2014.
- [23] I. Kokkinos. UberNet: Training a ‘Universal’ Convolutional Neural Network for Low-, Mid-, and High-Level Vision using Diverse Datasets and Limited Memory. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.
- [24] H. Kwon, Y. Kim, J. S. Lee, and M. Cho. First Person Action Recognition via Two-stream ConvNet with Long-term Fusion Pooling. *Pattern Recognition Letters*, 112:161–167, 2018.
- [25] Y. Li, M. Liu, and J. M. Rehg. In the Eye of Beholder: Joint Learning of Gaze and Actions in First Person Video. In *European Conference on Computer Vision (ECCV)*, pages 639–655, 2018.
- [26] Y. Li, Y. Zhefan, and J. M. Rehg. Delving into egocentric actions. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 287–295, 2015.
- [27] Z. Li, K. Gavriluk, E. Gavves, M. Jain, and C. G. M. Snoek. VideoLSTM convolves, attends and flows for action recognition. *Computer Vision and Image Understanding (CVIU)*, 166:41–50, 2018.
- [28] M. Ma, H. Fan, and K. M. Kitani. Going Deeper into First-Person Activity Recognition. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1894–1903, 2016.
- [29] K. Nakamura, S. Yeung, A. Alahi, and L. Fei-Fei. Jointly Learning Energy Expenditures and Activities Using Egocentric Multimodal Signals. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6817–6826, 2017.
- [30] T.-H.-C. Nguyen, J. C. Nebel, and F. Florez-Revuelta. Recognition of Activities of Daily Living with Egocentric Vision: A Review. *Sensors*, 16(1):72, 2016.

- [31] A. Nibali, Z. He, S. Morgan, and L. Prendergast. Numerical Coordinate Regression with Convolutional Neural Networks. *CoRR abs/1801.07372*, 2018.
- [32] H. Pirsiavash and D. Ramanan. Detecting activities of daily living in first-person camera views. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2847–2854, 2012.
- [33] R. Poppe. A survey on vision-based human action recognition. *Image and Vision Computing*, 28(6):976–990, 2010.
- [34] J. Redmon and A. Farhadi. YOLOv3: An Incremental Improvement. *CoRR abs/1804.02767*, 2018.
- [35] N. Riche, M. Duvinage, M. Mancas, B. Gosselin, and T. Dutoit. Saliency and Human Fixations: State-of-the-Art and Study of Comparison Metrics. In *IEEE International Conference on Computer Vision (ICCV)*, pages 1153–1160, 2013.
- [36] M. S. Ryoo and L. Matthies. First-Person Activity Recognition: What Are They Doing to Me? In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2730–2737, 2013.
- [37] M. S. Ryoo, B. Rothrock, and L. Matthies. Pooled motion features for first-person videos. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 896–904, 2015.
- [38] O. Sener and V. Koltun. Multi-task Learning As Multi-objective Optimization. In *32nd International Conference on Neural Information Processing Systems*, pages 525–536, 2018.
- [39] Y. Shen, B. Ni, Z. Li, and N. Zhuang. Egocentric Activity Prediction via Event Modulated Attention. In *European Conference on Computer Vision (ECCV)*, pages 202–217, 2018.
- [40] K. Simonyan and A. Zisserman. Two-stream convolutional networks for action recognition in videos. In *27th International Conference on Neural Information Processing Systems*, pages 568–576, 2014.
- [41] S. Singh, C. Arora, and C. V. Jawahar. First Person Action Recognition Using Deep Learned Descriptors. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2620–2628, 2016.
- [42] L. N. Smith. Cyclical Learning Rates for Training Neural Networks. In *IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 464–472, 2017.
- [43] S. Song, V. Chandrasekhar, B. Mandal, L. Li, J. H. Lim, G. S. Babu, P. P. San, and N. M. Cheung. Multimodal Multi-Stream Deep Learning for Egocentric Activity Recognition. In *IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 378–385, 2016.
- [44] E. H. Spriggs, F. De La Torre, and M. Hebert. Temporal segmentation and activity classification from first-person sensing. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, pages 17–24, 2009.
- [45] A. Stergiou, G. Kapidis, G. Kalliatakis, C. Chrysoulas, R. C. Veltkamp, and R. Poppe. Saliency Tubes: Visual Explanations for Spatio-Temporal Convolutions. In *IEEE International Conference on Image Processing (ICIP)*, page (to appear), 2019. [abs/1902.01078](#).
- [46] A. Stergiou and R. Poppe. Understanding human-human interactions: A survey. *CoRR abs/1808.00022*, 2018.
- [47] S. Sudhakaran, S. Escalera, and O. Lanz. LSTA: Long Short-Term Attention for Egocentric Action Recognition. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9954–9963, 2019.
- [48] S. Sudhakaran and O. Lanz. Attention is All We Need: Nailing Down Object-centric Attention for Egocentric Activity Recognition. In *British Machine Vision Conference (BMVC)*, page 12, 2018.
- [49] Y. Tang, Y. Tian, J. Lu, J. Feng, and J. Zhou. Action recognition in RGB-D egocentric videos. In *IEEE International Conference on Image Processing (ICIP)*, pages 3410–3414, 2017.
- [50] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri. Learning Spatiotemporal Features with 3D Convolutional Networks. In *IEEE International Conference on Computer Vision (ICCV)*, pages 4489–4497, 2015.
- [51] D. Tran, H. Wang, L. Torresani, J. Ray, Y. LeCun, and M. Paluri. A Closer Look at Spatiotemporal Convolutions for Action Recognition. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6450–6459, 2018.
- [52] Z. Tu, W. Xie, Q. Qin, R. Poppe, R. C. Veltkamp, B. Li, and J. Yuan. Multi-stream CNN: Learning representations based on human-related regions for action recognition. *Pattern Recognition*, 79:32–43, 2018.
- [53] G. Vaca-Castano, S. Das, J. P. Sousa, N. D. Lobo, and M. Shah. Improved scene identification and object detection on egocentric vision of daily activities. *Computer Vision and Image Understanding (CVIU)*, 156:92–103, 2017. Image and Video Understanding in Big Data.
- [54] 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 *European Conference on Computer Vision (ECCV)*, pages 20–36, 2016.
- [55] M. Wray, D. Moltisanti, W. Mayol-Cuevas, and D. Damen. SEMBED: Semantic Embedding of Egocentric Action Videos. In *European Conference on Computer Vision Workshops (ECCVW)*, pages 532–545, 2016.
- [56] C.-Y. Wu, C. Feichtenhofer, H. Fan, K. He, P. Krähenbühl, and R. Girshick. Long-Term Feature Banks for Detailed Video Understanding. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.
- [57] A. Zamir, A. Sax, W. Shen, L. Guibas, J. Malik, and S. Savarese. Taskonomy: Disentangling Task Transfer Learning. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.
- [58] M. Zhang, K. T. Ma, J. H. Lim, Q. Zhao, and J. Feng. Anticipating Where People will Look Using Adversarial Networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41(8):1783–1796, 2019.