Gökce, Zeynep; Pehlivan, Selen

Temporal modelling of first-person actions using hand-centric verb and object streams

Published in:
SIGNAL PROCESSING: IMAGE COMMUNICATION

DOI:
10.1016/j.image.2021.116436

Published: 01/11/2021

Document Version
Peer-reviewed accepted author manuscript, also known as Final accepted manuscript or Post-print

Please cite the original version:
Gökce, Z., & Pehlivan, S. (2021). Temporal modelling of first-person actions using hand-centric verb and object streams. *SIGNAL PROCESSING: IMAGE COMMUNICATION*, *99*, Article 116436. https://doi.org/10.1016/j.image.2021.116436

Temporal Modelling of First-Person Actions using Hand-Centric Verb and Object Streams

Zeynep Gökce[a,c], Selen Pehlivan[b]

[a] *Computer Engineering Department, TED University, Ankara, Turkey*
[b] *Department of Computer Science, Aalto University, Espoo, Finland*
[c] *TUBITAK Space Tech. Research Institute, Ankara, Turkey*

Abstract

Analysis of first-person (egocentric) videos involving human actions could help in the solutions of many problems. These videos include a large number of fine-grained action categories with hand-object interactions. In this paper, a compositional verb-noun model including two complementary temporal streams is proposed with various fusion strategies to recognize egocentric actions. The first step is based on construction of verb and object video models as decomposition of actions with a special attention on hands. Particularly, the verb video model that is the spatial-temporal encoding of hand actions and the object video model that is the object scores with hand-object layout are represented as two separate pathways. The second step is the fusion stage to identify action category, where distinct verb and object models are combined to give their action judgments. We propose fusion strategies with recurrent steps collecting verb and object label judgments along a temporal video sequence. We evaluate recognition performances for individual verb and object models; and we present extensive experimental evaluations for action recognition over recurrent-based fusion approaches on the EGTEA Gaze+ dataset.

Keywords: first-person vision, egocentric vision, action recognition, temporal models, RNN

1. Introduction

With the increasing availability and popularity of the wearable cameras, first-person (egocentric) vision offers an interesting scenario to study action recognition problem. Recordings with these cameras have become a part of daily life and evaluation of actions on these recordings gains decisive importance for applications, in particular, for health monitoring, autonomous driving, robotics and entertainment. For instance, these videos can be analyzed for monitoring the patient activities to detect early signs of dementia [1, 2], or for monitoring the driver's behavioral status to provide necessary assistance for safe and comfortable driving [3]. In robotic, these kinds of videos are useful to make the robot learn the human motion structure from the first-person view [4]. Besides, tracking and understanding human actions in first-person videos are important for developing feasible virtual reality applications [5].

Unlike third-person setting with fixed camera view, the first-person videos are recorded from the perspective of camera wearer, and usually hands and objects bear the most significant clues to determine action in various scenarios [6]. Due to similar hand-movements and hand-object interactions, these videos include large number of action categories with high inter-class similarities (e.g., *take tomato*, *put tomato*, *mix salad*). In addition to these characteristics, these videos have new challenges such as uneven camera transitions, frequent illumination changes, and limited camera vision. Thus, action recognition task on these videos will be better succeeded with fine-grained evaluation.

Many previous studies focus on modeling egocentric actions as composition of appearance and motion-based features. In this paper, we first decompose the egocentric actions into semantically meaningful and complementary components, verbs and objects [7]. Then, we target the appearance and motion-based features of each component for the purpose of fine-grained analysis and we model their temporal dynamics. Our model aims to process large number of distinct action categories through decomposition and fine-grained analysis while guaranteeing the recognition performance.

Our proposed model is demonstrated in Figure 1 and composes of three main parts, Verb, Object and Fusion models. Particularly, the verb model corresponding to the hand action representation and the object model corresponding to the interactions are represented as two separate pathways which are decomposition of actions. Finally, fusion is the action model employing various fusion strategies based on recurrent neural networks (RNN) on individual verb and object model judgments. Verb model is a verb classifier that takes successive C clips, each consisting of N successive frames (N=16 for C3D), and returns the verb scores per clip as an output. Object model is an object detection network taking C video frames, and returns object

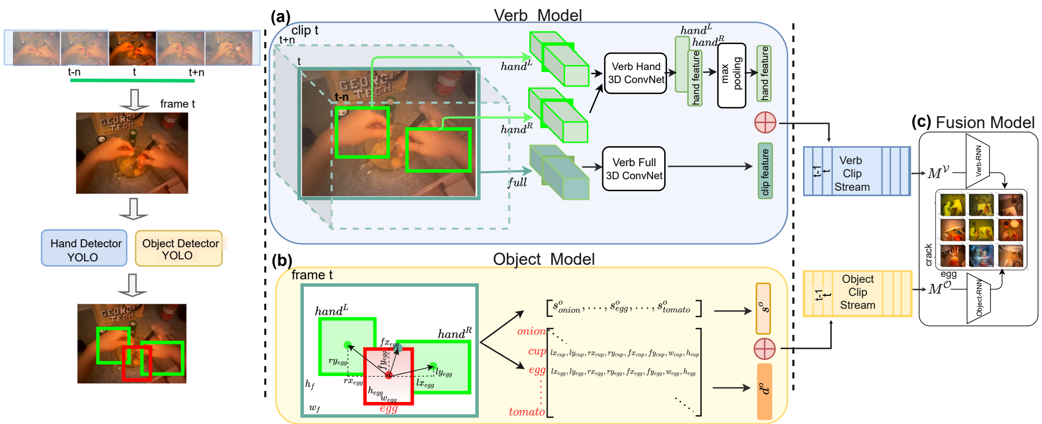

Figure 1: Overview of the proposed first-person action recognition model. Our model proposes a compositional model with two pathways, verb and object streams, respectively, and fusion strategies with recurrent structures. (a) Verb stream possesses the hand-centric action information and (b) the object stream possesses the appearance information of nearby objects with hand-interactions. Then (c) streams are fused using proposed fusion strategies to recognize actions.

proposals.

One of our motivation in the proposed model is to define a hand-centric action decomposition model as hands perform the main action and they are the target of attention in egocentric videos. First, we believe that verbs are strongly related to hand appearance and motion. Thus, our verb model prioritizes the fine-grained analysis of hand actions with spatial-temporal and multiscale modeling in a fully-supervised setting. Model encodes not only hand appearance using RGB-based 3D convolutional neural network (3D ConvNet), but also implicitly hand motion along tubes due to 3D ConvNet (a second stream can be added to encode Flow-based 3D ConvNet). Second, we similarly follow previous studies in mapping nouns to objects, but we believe hand-object interactions is also important to reveal target object. Thus, our noun component encodes two information: object proposals and the spatial layout of hands and object categories in frames.

Other motivation of our work is to contribute to action decomposition with support of recurrent models. Following this, we focus more deeply on modelling the temporal dynamics of hand-actions (verbs) and hand-object interactions (objects) individually and jointly using various fusion strategies. In our verb component, temporal model encodes the hand-centric information across clips. This means that it performs a smoothing over spatial-temporal multiscale features. In noun component, recurrent model encodes not only the temporal dynamics of object detection scores but also that of spatial layout across frames. This is also a kind of motion-based feature where the model tracks the layout of hands and objects.

Although video-based action decomposition is achieved priory in [7], our design of components is different and specialized over hands. [7] associates verbs with motion-based low-level features without any notion of hands, and represents frames with global dense trajectories [8]. Besides, nouns are associated with appearance-based features and encoded similarly over object proposals. Unlike using all objects, they encode only the objects found near hands. In short, our design is different in how we define the verb and noun components of the decomposition modelling and RNN-based fusion strategies that are proposed to combine complementary components while jointly modelling their temporal dynamics.

We are not the first using hand-centric approach for egocentric models. Hand is the main object and its manipulation is a strong clue for many studies [9, 10, 11, 12, 6]. Although these models are not based on decomposition, compared to these studies our architectural components are designed in a different way, where our model strongly relies on 3D ConvNet architecture for fine-grained evaluation of hand motion, appearance, and its interaction. Unlike attention based models using gaze estimation [13, 14], we examine regions around hand proposals as the target of attention.

The main contribution of our study is that (i) we propose a new first-person action decomposition model with hand-centric verb and noun components. Because hands serve as the target of attention, hand-centric spatial-temporal verb model and interaction feature are proposed. It has been shown that the combination of detectors in multiple scales and interaction feature increase the performance. As another contribution, (ii) we propose various fusion strategies with extension to recurrent structures. We observe that recurrent late fusion strategies outperform early fusion with many architectural advantages. We also show that our decomposition model performs comparable in recognition to conventional action recognition models, but with many architectural advantages. Then, (iii) we populate the recognition model with full supervision, since the action space is significantly large and contains extremely similar action categories. For that purpose, we train the background models of our system on a recent dataset, called EGTEA Gaze+ [13] with additional annotations. In addition to the available version, we gather on videos (1) the actual frames with the action label, and on these frames (2) the bounding boxes of hands with verb labels (annotations consist of additional verb labels such as *hold*), and (3) the bounding boxes of target objects. Finally, (iv) we present extensive experimental evaluation over fusion approaches on the EGTEA Gaze+ dataset with comparable results.

2. Related Work

Within the development of wearable cameras, a wide range of studies are proposed for action recognition in first-person videos [15, 16, 17, 18, 19, 20, 21, 13]. We explore traditional and deep learning based models including appearance-based, motion-based and hybrid models.

2.1. Appearance Based Models

Appearance cues related to objects, hands and gaze are informative for first-person videos and they are used in many studies.

Objects have crucial knowledge to describe first-person actions by revealing human-object interactions. In many studies [16, 9], hand-object interactions are modelled over region of interests (ROIs) to understand egocentric activities. According to Fathi et al.[9], the target object is generally visible in the center of video frames and a new model with two steps is proposed over the GTEA dataset. The first step is segmenting videos into foreground and background regions using optical flow, SIFT, color histograms. Foreground segments are further decomposed into hands and active objects. The second step performs recognition using Multiple Instance Learning (MIL) over object segments. According to Fathi et al. [16], fine-grained actions are categorized using hand interaction features (such as optical flow of both hand and object, hand pose, hand location, hand size, and left/right-hand relative location). As their previous work [9], hands, foreground objects and background are segmented and Adaboost [22] classifier is used for recognition. The proposed model results in an accuracy of 45% on the GTEA dataset. Another object-based model by Pirsiavash and Ramanan [17] is a temporal pyramid based model provided to define the usage of the objects in videos for action recognition. HOG [23] features are used for object modelling. With linear SVM classifier, performance is achieved up to 77% using object information over the ADL dataset. Recently, Cartas et al. [24] present another object-based model with two steps over the GTEA dataset. First, the hand region is segmented to get object region in frames using Multiscale Combinational Grouping method [25]. Then, a star-structured region model, R*CNN [26], is used for more than one region classification. Last, the output of R*CNN as contextual cue is given to LSTM to predict action category.

Gaze information is another cue for first-person action recognition, since camera wearer generally focuses on the point where the action is performed. Visual features extracted around the gaze regions are more informative compared to features extracted on other regions. Fathi et al. [27] extend object-based model of their previous work [16] with addition of gaze appearance. The SVM classifier is used for action categorization using object-based, gaze-based appearance features. Object-based features are extracted from object classifiers including object context around the gaze point. Using gaze information that is given with the GTEA Gaze+ dataset, the unrelated background objects are eliminated and 47% performance is achieved compared to 27% of [16] on the same dataset without using gaze information. Similarly, Li et al. [28] develop a model for gaze prediction in first-person videos using hand/head movement, hand location and hand pose.

2.2. Motion Based Models

First-person videos capture motion information from camera wearer's head, hand and eye movements. Besides object-based models, which is known as appearance-based models, motion-based models are also proposed to recognize the first-person actions in the literature. Kitani et al. [29] model motion in first-person sport activities using motion histograms. The motion histograms are based on optical flow of the scene. Due to the unsupervised scenario, Drichlet process mixture models are proposed to get action categories using the motion histograms. Li et al. [30] model motion information using Dense Trajectories [31] as a baseline descriptor.

2.3. Hybrid Models

Appearance and motion domains are composed (i.e., stream-based models [32]), since fusing them is more informative for first-person action recognition [6, 33, 34, 35].

Ma et al. [6] model object appearance and motion information as a two-stream network. The first stream analyzes appearance in three steps; segmentation, localization and object recognition successively, and the second stream analyzes motion using optical flow features. Yansong Tang et al. [33] propose a tri-stream network that integrates depth knowledge besides appearance and motion information and test over RGB-D egocentric dataset (THU-READ). Action prediction is calculated by taking the average score of three streams. Hahn et al. [34] propose a model using visual information from videos and textual information from recipe of these videos as well. The proposed model has three steps which are action proposal, object recognition, and recipe alignment steps. In action proposal step, video frames are localized with Bidirectional LSTM with two classes, action or not-action. In object recognition step, ResNet101 network [36] is trained for object classification along frames having actions according to the action proposal step. Finally, in the recipe alignment step, the action category is predicted using NLP model.

Recently, G. Kapidis et al. [35] introduce a multi-modal approach based on sequential learning to recognize egocentric actions on EPIC-Kitchens dataset [37]. LSTM is trained over feature sequence which consists of frame-based hand coordinates as motion knowledge and presence of object as appearance knowledge. Hand and objects are detected by YOLOv3 [38]. In contrast, our action model is trained over clip-based verb scores based on hand regions as motion information and detected object scores as appearance information using recurrent models.

3. Proposed Approach

Many studies emphasize the importance of appearance and motion-based lower-level modelling in action video understanding either with simple concatenation [34, 35] or with stream-based structures [32, 6, 33]. In this study, we aim at stream-based structures, but in semantic-level as decomposing actions into two complementary pathways that are the verb and the object components. Our model targets temporal modelling on each pathway with a strong attention to hands that perform actions to manipulate surrounding objects. The verb component is defined as a hand-centric temporal model. The model consists of short-term temporal modelling of hand regions within each video clip in multiple-scales using spatial-temporal 3D ConvNet models [39, 40]; and long-term temporal modelling of hands over clips of videos using RNN model. Complementary to this, the object component is defined as based on a temporal modelling of objects and hand interactions. The model consists of objects extracted by YOLOv2 object detector [41] and interaction-based spatial layout features, and further relies on long-term modelling of temporal dynamics of hand-object interactions over frames of videos using RNN model. Our aim is to perform action recognition by combining pretrained verb and object models using various fusion strategies.

3.1. Temporal Modelling of Verbs

Assuming the camera wearer's attention is on the hands in first-person videos, the verb model is hand-centric that it models short term temporal dynamics of hands performing action (hands perform *take* during *take tomato*) in multiple scale. Our verb model is fully-supervised and trained using clips including verb-labelled mid-frames and hands on these frames. It composes of two sub-models in different scales, namely full-scale and hand-scale verb models. With clips including hands, while the full-scale verb model represents coarse-grained verb description of the clips by utilizing whole frame region (covering hands, objects and scene); the hand-scale verb model represents fine-grained verb description by utilizing zoomed regions around hands. Given a video, each sub-model extracts stream of clip features per video. Particularly, these streams correspond to $V \times C$-dimensional verb matrices with V-dimensional features and C clips (When softmax outputs are used as clip representation, V corresponds to verb categories. Otherwise, V corresponds to the dimension of intermediate layer). These matrices are further combined over scales into a single verb matrix as the video verb representation. Figure 1 (a) shows an overview of the verb stream.

3.1.1. Full-Scale Verb Representation

The purpose of the full-scale verb model is to encode coarse-level verb category details of video clips. Model is based on 3D ConvNet architecture C3D [39] which takes the video clips as inputs and produces the category score vectors as outputs. Given a video with C successive clips extracted using temporal stride of two (i.e., dropping every other clip), each clip is embedded into a V-dimensional feature by the full-scale verb model. As the video full-scale hand representation, the model returns $V \times C$-dimensional verb matrix.

Following the original setting of [39], the 16-frame video clips are extracted and each is resized to $112 \times 112 \times 3 \times 16$ before fed into C3D [1]. Unlike the original C3D model [39] trained over randomly selected video clips, our model is trained over ground truth video clips that include verb-labelled mid-frame with hand performing verb (see Section 4.1 for the dataset details).

[1]https://github.com/hx173149/C3D-tensorflow

3.1.2. Hand-Scale Verb Representation

As part of a hand-centric approach, the purpose of the hand-scale verb model is to encode fine-level verb category details of video clips. Focusing on hand actions, this model utilizes hand regions instead of looking at a video in full-scale mode. It consists of two parts: hand detector and verb classifier. Given a video with clips, first a hand detector localizes hand regions in mid-frames of these clips, and then a spatial-temporal verb classifier takes the hand-volumes around these regions to identify verb categories.

To localize hands in video frames, we train a hand detector using the state-of-the art object detector YOLOv2 [41]. The original YOLO architecture is fine-tuned for binary classification of hands (hand/not-hand) on our hand dataset gathered from the EGTEA Gaze+ dataset (see Section 4.1 for the dataset details). During training of the hand-scale verb model, the YOLO hand detector is used to obtain hand-volumes, where volumes are particularly tubes cropped around hand proposals containing the hand-region in its mid-frame and lasts 16 frames. Hand proposals having 0.5 overlap with verb-labelled hands on the ground truth frames are used to train our hand-scale verb model.

Following the full-scale model, the hand-scale one is also based on 3D ConvNet architecture C3D [39] which takes hand-volumes as inputs and produces category scores as outputs. Given a video with the same set of C clips, hand-volumes are computed from hand proposals on mid-frames of these clips and fed into 3D ConvNet model to get volume features. As the video hand-scale verb representation, the model returns $V \times C$-dimensional verb matrix. Here, the cropped hand-volume is also resized into $112 \times 112 \times 3 \times 16$ before fed into the 3D ConvNet. Detection of multiple hand-volumes on the same video frame is possible, since some action categories are performed by both hands (e.g., while *open* and *take* are performed by one hand, *cut* and *mix* are performed by two hands). In that case, the second hand acts as an auxiliary hand to help the main hand performing action. For example, in verb *cut*, one hand cuts the object while the other holds the object. Following this, hand-scaled verb model is trained with one extra verb category, verb *hold*, that is also a ground truth label of hand regions in our dataset. In case of having multiple hands on a frame, such as *cut*, we apply max-pooling over features of hands to reduce into a single feature vector.

3.1.3. Video Verb Recognition

Given a video, our model extracts multiple verb matrices in multiple scales, full-scale and hand-scale, as verb representations of the video. These matrices are combined into a single verb matrix $M^{\mathcal{V}}$ using max-pooling. Verb matrix is particularly a sequence of clip features. Having training videos with various number of clips, we introduce a count-based verb model (see Section 4.3.1) using histograms and RNN-based verb model (see Section 4.4.1) to recognize the verb category of videos.

3.1.4. Discussion on 3D ConvNet Architectures

In this study we use two variants of 3D ConvNet architectures, C3D network [39] and I3D RGB network [40] to encode verbs in spatial-temporal domain.

C3D network has 5 convolutional and 5 pooling layers, where each convolution layer is immediately followed by a pooling layer. Then, network includes 2 fully connected layers and a linear classifier to predict action categories [39]. Compared to C3D, I3D is a denser network of inception-v1 with 3 convolution layers, 9 inception modules and 7×7 average-pooling layer preceding the last linear classifier [40]. Two-stream 3D ConvNet extension is available with I3D RGB and I3D Flow pathways, but we use RGB modality to encode our verbs.

C3D network is shallower than I3D network. But the fact that it is shallower makes the network appealing to train for each scale. Following this, we first concentrate on C3D network and train two models for hand-scale and full-scale keeping the original setting. These models are used to extract low level clip and hand features (see Section 3.1.1 and Section 3.1.2). Later, I3D RGB network is used in final experiments to evaluate the performance improvement. Instead of training two I3D models for each scale, we fine-tune the model with a simple network over our dataset. Selected intermediate layers of pre-trained I3D RGB network are used to extract coarse-level features for full-scale model and fine-level features for hand-scale model. Later, a simple shallow network is trained for classification over concatenated full-scale and hand-scale verb representations (please see Section 4.4.4 for network details).

3.2. Temporal Modelling of Objects

Objects manipulated by hands is used to model noun component of our decomposition model. First, we aim to find which objects appear in the video, and then we encode object information to reveal their interactions with hands. Figure 1 (b) indicates an overview of the object stream. Our object model is trained over object bounding boxes that are annotated on verb-labelled frames with hand annotations (see Section 4.1 for the dataset details). Given a video, object model extracts stream of frame features per video. Particularly, these streams correspond to $O' \times C$-dimensional verb matrices with O'-dimensional features and C clips.

3.2.1. Object Representation

To localize objects in video frames, the object detector YOLOv2 [41] is fine-tuned with object categories gathered from the EGTEA Gaze+ dataset (see Section 4.1 for details). During training, the YOLO object detector is used to obtain object proposals on video frames extracted using temporal stride of two frames. Later, each frame is encoded with a O'-dimensional $[\mathbf{s^o}, \mathbf{d^o}]$ feature vector, where $\mathbf{s^o}$ is an O-dimensional object score vector, and $\mathbf{d^o}$ is a $8 \times O$-dimensional object distance vector computed to represent hand-object interactions.

Proposals having 0.4 overlap with objects on the ground truth verb-labelled frames are extracted and used for training our object model. Having detected proposals, max-pooling is applied to pool over the confidence scores of the detected objects of the same category, and this results in a score vector $\mathbf{s^o} = [s^o_1, ..., s^o_i, ..., s^o_O]$, where s^o_i is the maximum score over all detected proposals belonging to category i and O is the number of object categories.

In addition to score vector, interactions of detected objects with detected hands (see Section 3.1.2 for detected hands) are encoded using a distance-based representation $\mathbf{d^o} = [\mathbf{d^o_1}, ..., \mathbf{d^o_i}, ..., \mathbf{d^o_O}]$ to encode spatial layout, where O is the number of object categories and $\mathbf{d^o_i}$ is the distance vector belonging to category i as follows,

$$
\begin{aligned}
&\mathbf{d^o_i} = [lx_i, ly_i, rx_i, ry_i, fx_i, fy_i, w_i, h_i] \\
&lx_i = (cx_i - cx_{lhand})/w_f, \quad ly_i = (cy_i - cy_{lhand})/h_f \\
&rx_i = (cx_i - cx_{rhand})/w_f, \quad ry_i = (cy_i - cy_{rhand})/h_f \\
&fx_i = (cx_i - cx_f)/w_f, \qquad fy_i = (cy_i - cy_f)/h_f \\
&w_i = w/w_f, \quad h_i = h/h_f
\end{aligned}
\tag{1}
$$

where lx_i and ly_i show the scaled x-distance and y-distance between the center of the left-hand and the center of the object i, respectively. rx_i and ry_i represent the scaled distances between the center of the right-hand and the center of the object i. fx_i and fy_i show x-distance and y-distance of object center i to frame center. Variables w_f, h_f, w and h are for the frame-width, frame-height, width and height of detected object i, respectively.

Detected hands are categorized as a left-hand or a right-hand based on their relative distances. The detected hand whose center is closer to the top left corner of the frame is classified as the left-hand and the right-hand otherwise. If there is only one hand detected, we duplicate the values for both hands. If multiple proposals of the same object category are detected, the proposal having a minimum Euclidean distance with any hand is selected (If there is no proposal for an object category, we insert zero values.).

As the video representation, the model returns a $(O+8\times O)\times C$-dimensional object matrix $M^{\mathcal{O}}$ as stacked $[\mathbf{s^o}, \mathbf{d^o}]$ features over video frames.

3.2.2. Video Object Recognition

Given a video, our model extracts a matrix $M^{\mathcal{O}}$ as object representation of the video. Object matrix is particularly a sequence of frame features. Having training videos with various number of frames, we introduce a count-based object model (see Section 4.3.3) using histograms and RNN-based object model (see Section 4.4.2) to recognize the object category of videos.

3.3. Temporal Modelling of Actions as Fusion of Verb-Object Pairs

Fusion is the last step of our proposed model to combine verb and object streams for recognizing actions. Since action videos consist of sequence of short-term clips, modeling of temporal relations between consecutive clips are important for action recognition. Such temporal modelling is also critical to smooth information over clips. Given the decomposition of actions into verb and object streams per video, we introduce multiple strategies with early and late fusion techniques using recurrent neural network (RNN) models for encoding temporal dynamics of actions. In this section, we first introduce a new count-based baseline model, and then we describe five different fusion strategies over verb-object streams for recognizing actions. Figure 2 shows the proposed fusion strategies. The architectural details of the best performed neural network for the proposed fusion strategies are given in Table 4 and experimental evaluations are reported in Table 7 and Table 8 .

Count-based verb-object multiplication. This model has no recurrent step and no training for action recognition (see Figure 2 (a)). Verb and object category scores for videos are obtained using the convolutional neural network

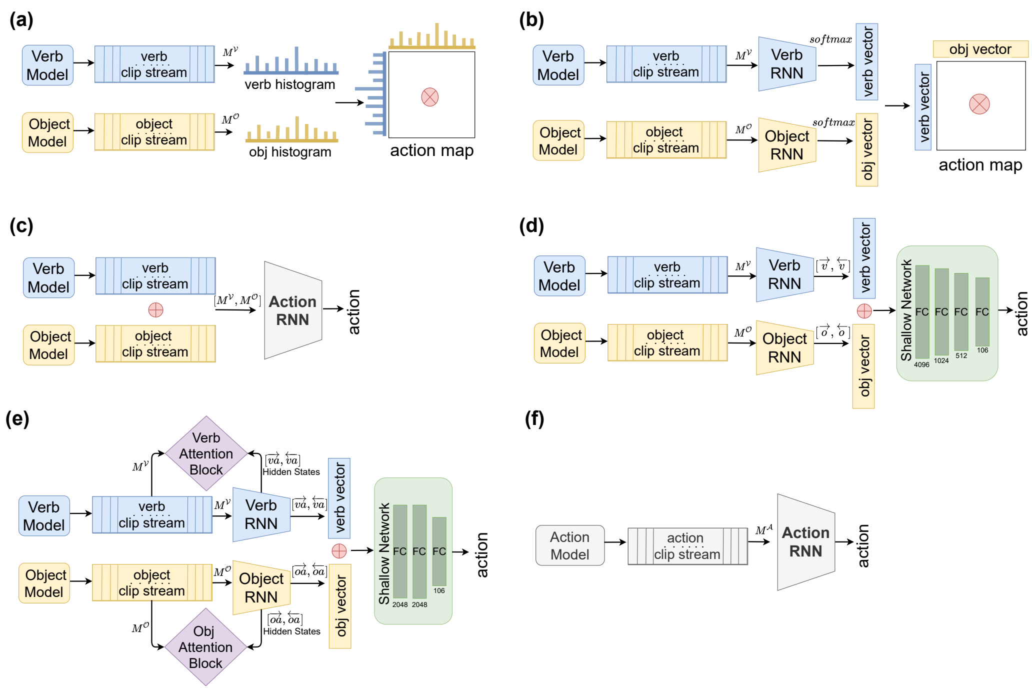

Figure 2: Fusion strategies for action recognition. (a) Count-Based Verb-Object Multiplication Baseline model, (b) Recurrent Verb-Object Multiplication model, (c) Recurrent Verb-Object Early Fusion model, (d) Recurrent Verb-Object Late Fusion model, (e) Recurrent Verb-Object Attention Late Fusion model, (f) Recurrent Action Baseline model (verb vector is either "scores" or "features", since softmax scores and intermediate level features are used interchangeably in the experiments).

softmax predictions per clip, and the model performs a simple multiplication for recognition.

To compute verb stream, count-based multiplication model uses 3D ConvNet softmax prediction scores over clips of video samples. Given a video, two $V \times C$-dimensional verb matrices with V verb categories and C clips are extracted using full-scale and hand-scale 3D ConvNet verb models, and these matrices are combined into a single score matrix by max-pooling (see Section 3.1). Computing verb stream, each clip is assigned to a verb category with the maximum score over V categories. Later, a histogram showing the distribution of verb categories over C video clips is computed and L1 normalization is applied to eliminate the effect of video length. To sum up, the video is represented as a V-dimensional verb category score vector $\mathbf{v}$. Similarly, to compute object stream, objects are detected in frames using the object detection model (this fusion strategy does not use spatial layout feature, $\mathbf{d^o}$, see Section 3.2.1). Then, each frame is assigned to an object category with the maximum detection score. Later, a histogram showing the distribution of object categories over C video frames is extracted and L1 normalization is applied to eliminate the effect of video length. To sum up, the video is represented as a O-dimensional object category score vector $\mathbf{o}$.

Inspiring from a recent study on human-object interactions in still images [42], we combine verb vector $\mathbf{v}$ and object vector $\mathbf{o}$ using a simple multiplication as follows,

$$\begin{aligned} A &= \mathbf{v} \cdot \mathbf{o}^T \\ A' &= A \odot B \end{aligned} \tag{2}$$

where A is a $V \times O$-dimensional estimation map with scores for all action categories corresponding to combinations of all *verb-object* category pairs. B is a $V \times O$-dimensional ground truth binary mask where 1 shows the existence of a *verb-object* category pair, 0 shows the nonexistence of the pair in the dataset (e.g., $cut\text{-}fridge$ pair is 0 since it is not an action in our dataset). In order to evaluate the scores of the subset of *verb-object* pairs existing in the dataset, the estimation map A is masked by binary mask B and the final result is matrix A'.

Finally, the *verb-object* pair with the maximum value over matrix A' is assigned as the predicted category of the given test sample. Particularly, this fusion strategy returns the prediction of action categories without training.

Recurrent verb-object multiplication. This model uses the score vectors of RNN-based verb and object models as inputs (see Section 3.1.3 and Section 3.2.2), and action recognition stage is performed with a simple multiplication over these score vectors without training (see Figure 2 (b)). First, two individual RNN models are trained over verb matrix $M^{\mathcal{V}}$ and object matrix $M^{\mathcal{O}}$ of training videos, namely verb-RNN and object-RNN models. During testing, the verb-RNN returns the V-dimensional verb category score vector $\mathbf{v}$ (V is the number of verb categories) and the object-RNN returns O-dimensional object category score vector $\mathbf{o}$ per video sample (O is the number of verb categories). Then, we similarly apply multiplication and masking using Eq. 2. Please note that this fusion strategy does not use spatial layout feature, $\boldsymbol{d^o}$, and object-RNN is trained over stream of score vectors $\boldsymbol{s^o}$ (see Section 3.2)

Recurrent verb-object early fusion. This model includes a single recurrent model to recognize the action category of the video (see Figure 2 (c)). Extracting verb (see Section 3.1) and object (see Section 3.2.1) matrices, $M^{\mathcal{V}}$ and $M^{\mathcal{O}}$, we concatenate them as a video representation. Then, a single RNN model, action-RNN, is trained to predict action categories.

Recurrent verb-object late fusion. Similar to second fusion strategy, Recurrent Verb-Object Multiplication, two individual recurrent neural network models, verb-RNN and object-RNN, are trained over verb and object streams on training video samples (see Figure 2 (d)). Then, the score vectors from the RNN models, $\mathbf{v}$ and $\mathbf{o}$, are concatenated as a video representation. As an additional training stage, a shallow network with a set of fully connected layers is trained to predict action categories over concatenated representation.

We also extend this model with forward and backward feature vectors when BiLSTM recurrent models are available. Let $[\overrightarrow{\mathbf{v}}, \overleftarrow{\mathbf{v}}]$ be the concatenation of forward and backward-direction BiLSTM recurrent function outputs of the verb network, and $[\overrightarrow{\mathbf{o}}, \overleftarrow{\mathbf{o}}]$ be forward and backward-direction BiLSTM recurrent function outputs of the object network. Then, these vectors from the verb and object recurrent models are concatenated and $[\overrightarrow{\mathbf{v}}, \overleftarrow{\mathbf{v}}, \overrightarrow{\mathbf{o}}, \overleftarrow{\mathbf{o}}]$ is used as a video representation. Later, a shallow network with a set of fully connected layers is trained to predict action categories.

Recurrent verb-object attention late fusion. This model includes an additional attention module to encode temporal information over RNN models (see Figure 2 (e)). On each component of our two-stream model, a recurrent neural network layer is trained with a self-attention module [43] as follows,

$$\boldsymbol{\alpha}^{\mathcal{V}} = softmax(\mathbf{w_2} tanh(W_1 M^{\mathcal{V}})), W_1 \in \mathbb{R}^{256\times V}, \mathbf{w_2} \in \mathbb{R}^{1\times 256} \tag{3}$$

where $M^{\mathcal{V}}$ is the verb stream matrix. The attention block takes $M^{\mathcal{V}}$ and BiLSTM outputs as input. The $M^{\mathcal{V}}$ is fed into a fully connected layer with a $256\times V$-dimensional weight matrix W_1, followed by tanh() function. Then, a score vector is produced by applying a vector of parameters $\mathbf{w_2}$ that is 1×256. Later, we sum up the RNN model hidden states according to the weight provided by $\boldsymbol{\alpha}^{\mathcal{V}}$ to get a vector representation $\mathbf{a}^{\mathcal{V}}$. Here, vector $\boldsymbol{\alpha}^{\mathcal{V}}$ represents

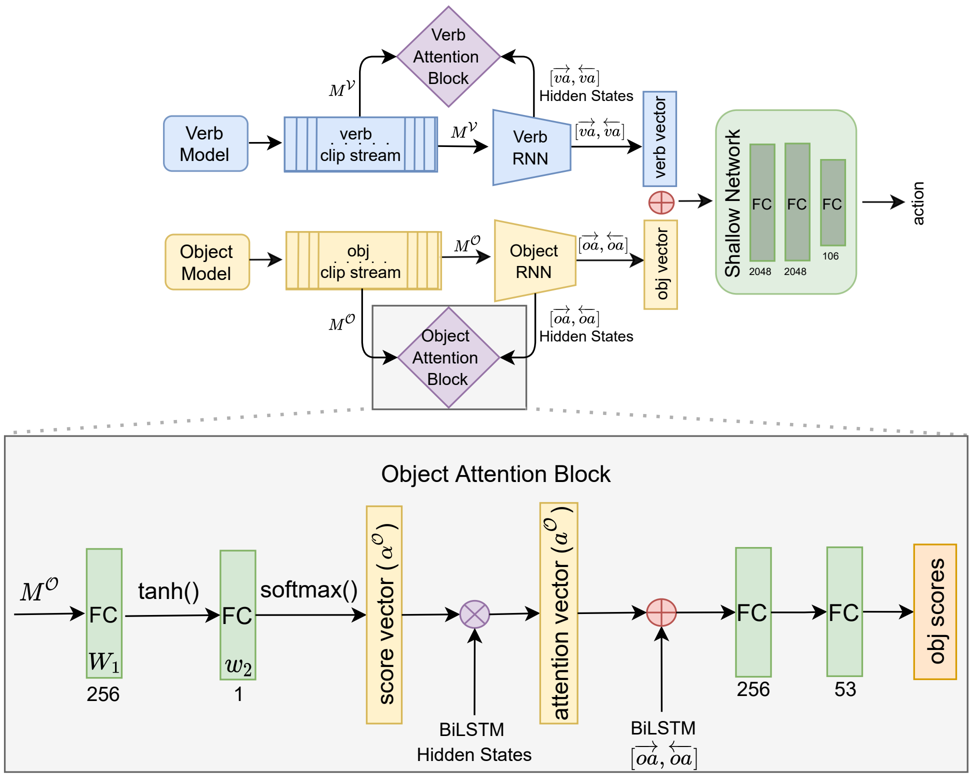

Figure 3: Recurrent Verb-Object Attention Late Fusion strategy. This model extends late fusion strategy with an additional attention block that can be applied both to verb and object streams. Attention block helps to encode temporal context within an attention vector.

the temporal attention of the video. The softmax() function ensures all the computed weights sum up to 1.

The object stream is similarly trained over $M^{\mathcal{O}}$ using RNN with attention module. Assuming BiLSTM as our RNN structure, forward and backward function outputs of verb-BiLSTM trained with attention block, $[\overrightarrow{\mathbf{va}}, \overleftarrow{\mathbf{va}}]$, and forward and backward function outputs of object-BiLSTM, $[\overrightarrow{\mathbf{oa}}, \overleftarrow{\mathbf{oa}}]$, are concatenated as a video representation. Later, a shallow network with a set of fully connected layers is trained to predict action categories. For the verb and object streams, we have added attention block as seen in Figure 3.

Recurrent action baseline. This model is a baseline model using 3D ConvNet architecture trained over action categories (see Figure 2 (f)). Given

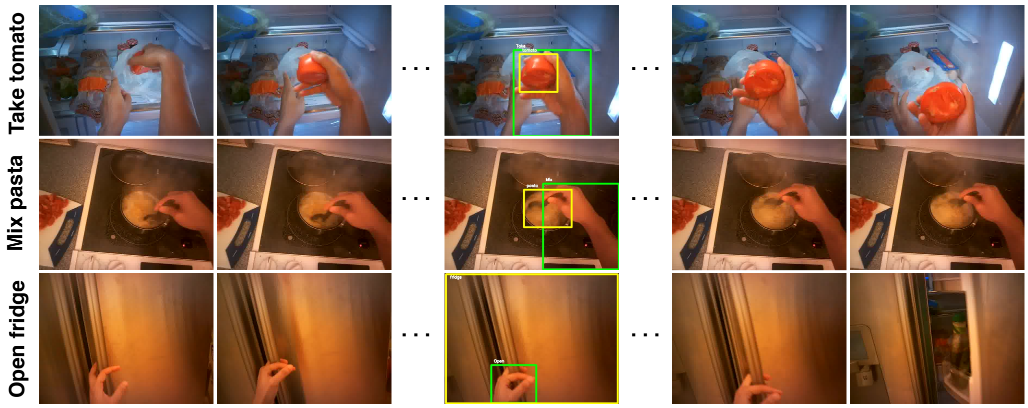

Figure 4: Our sample annotations on the EGTEA Gaze+ dataset. We annotate video frames, hands and the objects having consistent motion with the action in training videos.

a video with C clips, 3D ConvNet architecture C3D takes clips as inputs and returns category scores as outputs. Video is represented as a matrix of clip features, $M^{\mathcal{A}}$. Then, an RNN model, action-RNN, is trained over $M^{\mathcal{A}}$ to predict action categories.

4. Experimental Evaluation

4.1. EGTEA Gaze+ Dataset with Frame Level Annotations

We perform our experiments on the EGTEA Gaze+ dataset [13, 44] which includes first-person meal preparation activity videos. The dataset is extended from the GTEA Gaze+ and it consists of 86 cooking videos performed by 32 different subjects. It includes 106 fine-grained action categories with 19 verb and 53 object categories. While some action categories (e.g., *cut tomato*) refer one object category, others (e.g., *pour water-faucet-pot*) refer multiple objects. The dataset contains 3 train-test splits, each of which has 8229 training and 2022 test video samples. Since the study [13] published the EGTEA Gaze+ dataset reports results on split1, we perform the experimental evaluation of models on the same split.

Current version of the EGTEA Gaze+ dataset does not include frame level annotations. In our study, video frames, the hands and the objects having consistent motion with the action category are annotated in training videos (see Figure 4). The annotated frames with hands correspond to the middle frames of clips used in training of 3D ConvNet verb models (see Section 3.1). *hand* is not included among object categories, therefore *wash hand* action category in *wash* verb and *hand* object is just recognized using additional *wash-hand* verb category (no object). Moreover, some video samples cannot be assigned to a verb category due to the absence of hands, therefore *inspect-read recipe* action is identified using only *recipe* object. As a result, annotations for 20 verb categories are provided for training of verb models (19 verbs + *hold*).

Moreover, objects interacting with hands are used in our object model. Therefore, the frames with hand annotations are labelled with object locations and same object categories are populated with additional annotations from other frames of the videos. Annotations of *water*, and *seasoning* are skipped since they are quite ambiguous to label. Moreover, the *hand* is not included among object categories, since the hand annotations are used to train auxiliary hand model in the hand-scale verb model (see Section 3.1.2). As a result, annotations for 50 out of 53 object categories are provided for the object models.

4.2. Experimental Setting

The details of the best performed neural network action recognition architectures with their verb and object submodels are summarized in Table 4. For each component of the decomposition model, we have a base model and an RNN model. Base models for verb component are build based on C3D and I3D networks, and base model of object component is based on YOLO. While we keep the original architectures of C3D and YOLO, we fine-tune the I3D architecture with a simple shallow network over our dataset. Later, our Verb and Object-RNNs are trained and their outputs are used as inputs of action models. We train the networks for minimizing the cross-entropy loss and we use ADAM as the optimization algorithm.

Late fusion strategy is proposed as our main action recognition architecture with higher recognition performance. The best late fusion model using C3D-d^o (*Verb-Object Late Fusion-fc7(with d^o)-BiLSTM Bw+Fw* in Table 7) with 50.29% mAcc consists of a shallow network of 3 fully connected layers with 4096, 1024, 512 neurons and a 106-dimensional softmax output (see row 4 in Table 4). It is trained using batch size 10 and learning rate 1e-5. The Verb-RNN model of C3D-d^o (*C3D full+hand (concatenation)-fc7-BiLSTM*, see Table 5) is constructed using 1 layer BiLSTM with 768 cell size and trained using batch size 10, learning rate 1e-5. On the other hand Object-RNN model of C3D-d^o (*YOLO-s^o + d^o-BiLSTM*, see Table 6) is constructed with 1 layer BiLSTM with 1024 cell size, and it is trained using batch size 10 and learning rate 1e-4.

Later, our model is improved with I3D feature (*Our I3D-full+hand-BiLSTM Bw+Fw* in Table 8). The I3D action model (see row 5 in Table 4) consists of 2 fully connected layers with 1024 neurons. It is trained with batch size 20 and learning rate 1e-4. The Verb-RNN of I3D model is basically constructed using 1 layer BiLSTM with 512 cell size and trained using batch size 10 and learning rate 1e-6. The Object-RNN of I3D is same with C3D-d^o.

Finally, we extend our late fusion strategy with an attention module, I3D-Att as our best performing model (*Our I3D-full+hand-BiLSTM+Attention*). The verb and object-RNNs of the model are constructed using 1 layer with 512 and 256 cell sizes, respectively (see row 6 in Table 4). Training is performed with batch size 10 and learning rate 1e-4 for verb-RNN and 1e-5 for object-RNN, respectively. The action model using Recurrent Verb-Object Attention Late Fusion setting contains two 2048-dimensional fully connected layers with dropout rate of 0.5. It is trained using batch size 20, learning rate of 1e-5.

One main problem we face is the imbalanced samples of the dataset. Using all ground truth verb data, we observe that the model tends to classify categories with more samples, but misclassify the categories with fewer samples (e.g., *take* verb category contains 1886 samples, but *squeeze* verb category has 29 samples). Thus, as the accuracy gets increased, the mean class accuracy drops significantly. To solve this problem, we train verb models using balanced subsets. If a category has more than 250 samples, we randomly select 250 samples. If a category has less than 200 samples, we apply some aggregation strategies and we generate new clips by sampling around the ground truth clips. For all other categories, we keep their original number of samples. Thus, we train all reported verb recognition experiments with balanced data having maximum 250 and minimum 200 samples for all categories.

4.3. Experiments on Count-Based Baseline Models

In this section, recognition performances of verb, object and action models are evaluated, respectively, when a simple count-based strategy is followed to compute video category score over clip categories for the related task. For individual verb and object recognition tasks, verb and object streams

Verb Base Model	**mAcc (%)**
C3D full	39.99
C3D hand20	37.13
C3D hand10	35.91

Table 1: Experiments to show the effect of hand-volume ROI scales on video verb recognition results with mean class accuracies (mAcc).

are the stacked softmax prediction scores over clips and the final video verb category or object category is identified using a simple distribution. For action recognition task, we first obtain the distribution of verb and object categories over clips and we apply a fusion model called Count-Based Verb-Object Multiplication strategy (see Figure 2 (a)).

4.3.1. Count-Based Verb Models in Multiple Scale

Region of interests (ROIs) in different scales encode different amount of information from the hand and the background. In order to examine the effect of scale in hand-centric verb model performance, we evaluate verb recognition in various scales. Verb recognition is analyzed by computing the main verb category appearing in the video. The category with the maximum value on the V-dimensional verb category score vector $\mathbf{v}$ (histogram) returns the predicted verb class of the video (see Count-Based Verb-Object Multiplication model in Section 3.3).

The performance of each scale is shown in Table 1. The first verb model is trained in full-scale mode (see Section 3.1.1). Other models, hand-scale verb models, are trained using different enlargement scales around hand bounding boxes (hand10 and hand20 verb models mean that 10% and 20% of enlargements with respect to the width and the height of the detected hand regions are applied, respectively). We observe that the full-scale verb model outperforms the hand-scale verb models, since the full region encodes the scene information including hand motion, object and background. The context provided by each element enhances the ability to recognize action in videos [16].

4.3.2. Combination of Count-Based Verb Models

When the verb recognition accuracies are investigated in the category level, it has been observed that the hand-scale verb model outperforms the full-scale one in some verb categories such as *open, put, crack* verbs. Therefore, we also analyze the performance of verb models when multiple scales

Combination	Verb Base Model	mAcc (%)
weighted average	C3D full+hand10	46.99
	C3D full+hand20	45.63
	C3D full+hand10+hand20	46.38
max-pooling	C3D full+hand10	46.91
	C3D full+hand20	44.91
	C3D full+hand10+hand20	43.19

Table 2: Experiments to show the effect of combined verb models on video verb recognition results with mean class accuracies (mAcc). The set of weight parameters for C3D full+hand10 combination is $\{\beta_{full} = 0.5,\ \beta_{hand10} = 0.5\}$, for C3D full+hand20 combination is $\{\beta_{full} = 0.5,\ \beta_{hand20} = 0.5\}$, and for C3D full+hand10+hand20 combination is $\{\beta_{full} = 0.4,\ \beta_{hand10} = 0.3,\ \beta_{hand20} = 0.3\}$.

are combined. We use two methods to combine, weighted average and max-pooling. In the first method, softmax values of the clip sequences from full-scale and hand-scale models are weighted averaged at the clip level. The weight parameters $\{\beta_{full},\ \beta_{hand10},\ \beta_{hand20}\}$ are empirically searched in the range of [0-1]. In the second method, the max-pooling is applied over the softmax values of the clip sequences.

Combining verb video representations, we compute the verb category score vector $\mathbf{v}$. The category with the maximum value on vector $\mathbf{v}$ returns the predicted verb class of the video (see Count-Based Verb-Object Multiplication model in Section 3.3). The combination performances are reported in Table 2. The result shows that the combination of softmax values enhances the mean class accuracy (mAcc) of the verb model up to 46.99% (since the full-scale verb model achieves the best accuracy, we keep the full-scale verb model and combine it with the hand-scale models). We observe that the combination of verb models enables the model to capture low-level appearance features both in coarse scale and fine scale. Particularly, hand-scale and full-scale models vote for the clip score together where hand-scale model returns scores over local hand ROI and this helps to recognize harder video instances that are misclassified without local details. The weights of the best combination in weighted averaging method are 0.5 and this shows the best score is achieved with equal contribution of the scales. Although the accuracy of the weighted average method is slightly higher than the max-pooling method, we select the max-pooling method for the count-based action recognition experiments due to its simplicity.

Action Model with Simple Counting	mAcc (%)
Count-Based Verb-Object Multiplication	33.87
Count-Based Action Baseline	23.89

Table 3: Action recognition results with mean class accuracies (mAcc) using Count-Based Verb-Object Multiplication model and Count-Based Action Baseline model on split1.

4.3.3. Count-Based Object Model

We also perform the evaluation of object stream at the video level. For the object recognition, the category with the maximum value on the O-dimensional object category score vector **o** (histogram) returns the predicted object class of the video (see Count-Based Verb-Object Multiplication model in Section 3.3). Particularly, this is to compute how often the main object category appears in the video and the main object is the assigned category for a video frame with the maximum YOLO confidence. 63.41% mean class accuracy (mAcc) is achieved for video object classification on split1 (see row 2 of Table 6). For *trash*, *mixture*, *condiment* object categories, low accuracies are achieved since these objects are hard to detect. Moreover, we observe that for some categories labelling is ambiguous and ground truth labeling causes low accuracies for these object categories. For instance, *tomato container* instances are visually similar to *grocery bag* instances and they are getting mixes up with each other. In another example, $fridge$, $fridge$ $drawer$ and *drawer* instances are getting confused and used interchangeably in labelling.

We also perform a simpler evaluation of object recognition over video samples as a baseline, where we do not use the histograms to identify the predicted object category but instead we find the category with the maximum score over all YOLO detections of all frames. This results in 50.70% mAcc value as reported in the first row of Table 6. It shows that histogram based model better evaluates the object recognition over videos.

4.3.4. Count-Based Action Model

Our first fusion strategy for action recognition is based on a simple multiplication of the verb category score vector **v** and the object category score vector **o** (see Count-Based Verb-Object Multiplication model in Section 3.3). The verb scores are computed using C3D network. The results are reported in Table 3 with a mean class accuracy (mAcc) of 33.87% on split1.

For comparison, we also construct a Count-Based Action Baseline model having the same implementation with the verb model, but architecture learns action categories rather than verb categories. The Action Baseline model is trained over video clips in a supervised setting using annotated action frames (clips). For each clip of the test video, a softmax output over action labels is retrieved from the C3D model, then the frequently observed action label on the clip sequence of the video is evaluated using a histogram proposed in Section 3.3 (instead of applying count based model on verb and object streams separately, we apply the same model only on action scores). We observe that the Count-Based Verb-Object Multiplication model outperforms the Count-Based Action Baseline model with almost 10% accuracy. Although multiplication of stream scores is a simple technique for action recognition, its performance is higher than the baseline model without any learning (see Table 3). Here, we have the same training instances for all models, but verb-object multiplication trains over a smaller number of categories with more samples for verb and object streams compared to action baseline model. This might help neural network models to better train. Moreover, our model contains fine-grained representations of the video instances compared to action baseline model using detection models used in the background for objects and hands.

4.4. Experiments on Recurrent-Based Models

In this section, recognition performances of verb, object and action models are evaluated, respectively, when recurrent structures are used for the related task. Particularly, we evaluate the recurrent-based fusion strategies we propose for action recognition (see Figure 2 (b-e)). Recurrent models get the verb and the object representations of video as a set of clip features, and then they use RNN models (LSTM or BiLSTM) to encode the temporal dynamics within streams.

4.4.1. Recurrent-Based Verb Models

We examine the individual performances of recurrent-based verb models. From the verb experiments of count-based verb model (see Section 4.3.2), we know that the combination of features in multiple scales improves the performance of verb recognition. Therefore, we combine verb models in different scales: the full-scale and hand-scale verb models. We use hand20 verb model for split1 due to its performance reported in Table 1. Here, verb matrices of multiple verb models are combined by concatenation or max-pooling, and then the combined feature matrix is fed into either BiLSTM or LSTM recurrent model to analyze the verb recognition performance. Two different

Fusion Model		Verb Base	Verb-RNN	Object Base	Object-RNN	Action Model
Multiplication C3D softmax/fc7		C3D Full&Hand	BiLSTM	YOLO	BiLSTM	
	soft	original	100 cell, 1 layer	original	100 cell, 1 layer	
	fc7	✓	768 cell, 1 layer	✓	✓	
	input	Clip($112 \times 112 \times 3 \times 16$)	full *softmax/fc7* hand *softmax/fc7*	Img($416 \times 416 \times 3$)	Obj s^o	
	output	Verb-full *softmax* Verb-hand *softmax*	Verb-RNN *softmax*	Obj *softmax*	*Obj-RNN softmax*	
Early C3D(concat) softmax		C3D Full&Hand		YOLO		BiLSTM
	soft	original		original		100 cell, 1 layer
	input	Clip($112 \times 112 \times 3 \times 16$)		Img($416 \times 416 \times 3$)		Verb-concat *softmax* Obj *softmax*
	output	Verb-full *softmax* Verb-hand *softmax*		Obj *softmax*		*softmax* (106)
Late C3D softmax/fc7		C3D Full&Hand	BiLSTM	YOLO	BiLSTM	Shallow Network
	soft	original	100 cell, 1 layer	original	100 cell, 1 layer	fc(512)-drop(0.5)- fc(256)-drop(0.5)
	fc7	✓	768 cell, 1 layer	✓	✓	✓
	input	Clip($112 \times 112 \times 3 \times 16$)	full *softmax/fc7* hand *softmax/fc7*	Img($416 \times 416 \times 3$)	Obj s^o	Verb-RNN *softmax* Obj-RNN *softmax*
	output	Verb-full *softmax* Verb-hand *softmax*	Verb-RNN *softmax*	Obj *softmax*	Obj-RNN *softmax*	*softmax* (106)
Late C3D bw+fw softmax/fc7/ fc7 with d^o		C3D Full&Hand	BiLSTM	YOLO	BiLSTM	Shallow Network
	soft	original	100 cell, 1 layer	original	100 cell, 1 layer	fc(1024)–drop(0.5)– fc(1024)–drop(0.5)- fc(512)–drop(0.5)– fc(256)–drop(0.2)
	fc7	✓	768 cell, 1 layer	✓	✓	fc(1024)-drop(0.5)- fc(512)-drop(0.5)
	w/ d^o	✓	✓	✓	1024 cell, 1 layer	fc(4096)-drop(0.5)- fc(1024)-drop(0.5) fc(512)-drop(0.5)
	input	Clip($112 \times 112 \times 3 \times 16$)	full *softmax/fc7* hand *softmax/fc7*	Img($416 \times 416 \times 3$)	Obj $s^o/s^o + d^0$	Verb-RNN *Bw+Fw* Obj-RNN *Bw+Fw*
	output	Verb-full *softmax* Verb-hand *softmax*	Verb-RNN *softmax*	Obj *softmax*	Obj-RNN *softmax*	*softmax* (106)
Late I3D bw+fw with d^o		Our I3D(fine-tuned)	BiLSTM	YOLO	BiLSTM	Shallow Network
	model	fc1(1024)-drop(0.5)- fc2(1024)-drop(0.5)	512 cell, 1 layer	original	1024 cell, 1 layer	fc(1024)-drop(0.5)- fc(1024)-drop(0.5)
	input	*Mixed-5c* *MaxPool3d-5a-2x2*	Our I3D *fc2*	Img($416 \times 416 \times 3$)	Obj $s^o + d^0$	Verb-RNN *Bw+Fw* Obj-RNN *Bw+Fw*
	output	Verb *softmax*	Verb-RNN *softmax*	Obj *softmax*	Obj-RNN *softmax*	*softmax* (106)
Late I3D-Att bw+fw with d^o		Our I3D(fine-tuned)	BiLSTM-Att	YOLO	BiLSTM-Att	Shallow Network
	model	fc1(1024)-drop(0.5)- fc2(1024)-drop(0.5)	512 cell, 1 layer Att Module	original	256 cell, 1 layer Att Module	fc(2048)-drop(0.5)- fc(2048)-drop(0.5)
	input	*Mixed-5c* *MaxPool3d-5a-2x2*	Our I3D *fc2*	Img($416 \times 416 \times 3$)	Obj $s^o + d^0$	Verb-RNN *Bw+Fw* Obj-RNN *Bw+Fw*
	output	Verb *softmax*	Verb-RNN *softmax*	Obj *softmax*	Obj-RNN *softmax*	*softmax* (106)

Table 4: Architectures details of action models from Table 7 and Table 8 with their verb and object base models and RNN structures. ✓ marks indicate the repetition of the above model structure.

Verb Base Model	Feature	Verb-RNN	mAcc (%)
C3D full	softmax	BiLSTM	46.49
C3D full+hand (concatenation)	softmax	BiLSTM	53.95
C3D full+hand (max-pooling)	softmax	BiLSTM	50.43
C3D full+hand (max-pooling)	softmax	LSTM	48.13
C3D full	fc7	BiLSTM	53.23
C3D full+hand (concatenation)	fc7	BiLSTM	**59.50**

Table 5: Video verb recognition results with mean class accuracies (mAcc) on split1 with recurrent-based methodologies.

intermediate features of C3D verb models are experimented on to construct verb feature matrices, softmax prediction scores and fc7 layer features.

We first test using softmax prediction scores. The verb matrices are the stacked softmax outputs per clip. According to the experimental results given in Table 5, it is observed that the combination of full-scale and hand-scale verb models helps in verb recognition with 7.46% improvement over full-scale verb model on split1. The feature concatenation method outperforms the max-pooling in verb models. Moreover, the verb model with BiLSTM structure (50.43%) gives higher accuracy than LSTM structure (48.13%), and we therefore continue with BiLSTM for the rest of the experiments. When compared to Table 2, it is clearly seen that recurrent verb models outperform the simple count-based models in recognition. This shows that recurrent models are better to model verb streams. The best BiLSTM verb model is constructed using 1 layer with 100 cell size (see Table 4).

In order to improve verb recognition performance in recurrent-based models, we also test our experiments with fc7 layers instead of softmax scores. Extracted fc7 layers from the full-scale and the hand-scale verb models (please note that hand features are combined into a single feature vector using max-pooling, if there are multiple hands) are concatenated per video clip. Then, the video representation as stacked clip features is fed into the BiLSTM model. The experiments are conducted over split1, and we obtain 53.23% accuracy using full-scale verb model and 59.50% accuracy using concatenated features. Results show that fc7 that is an earlier feature layer improves recognition rates significantly. The best BiLSTM verb model with fc7 features is constructed using 1 layer with 768 cell size (see Table 4).

Object Base Model	**Feature**	**Object Recognition**	**mAcc (%)**
YOLO	s^o	Max-pooling	50.70
YOLO	s^o	Count-based	63.41
Object Base Model	**Feature**	**Object-RNN**	**mAcc (%)**
YOLO	s^o	BiLSTM	70.59
YOLO	s^o+d^o	BiLSTM	**70.83**
YOLO	s^o	LSTM	68.73

Table 6: Video object recognition results with mean class accuracies (mAcc) using count-based and recurrent-based strategies. For the recurrent models, d^o indicates that distance-based spatial layout features are also integrated while modelling object features. Otherwise, s^o features are used. (Please see Section 3.2.1)

4.4.2. Recurrent-Based Object Models

We evaluate individual performances of recurrent-based object models using BiLSTM and LSTM structures (see Section 3.2.1). Object features, $[\boldsymbol{s^o}, \boldsymbol{d^o}]$, are extracted over video frames by the object model in a matrix form, and then they are fed into RNN object model. BiLSTM object model using stand-alone $\boldsymbol{s^o}$ achieves 70.59% accuracy on split1. The BiLSTM model has 1 layer with 100 cell size (see Table 4). Extending model with spatial layout feature using $[\boldsymbol{s^o}, \boldsymbol{d^o}]$, BiLSTM object model achieves 70.83% accuracy over split1. The model has 1 layer with 1024 cell size (see Table 4).

According to Table 6, it is observed that the recurrent-based object models outperform the count-based object model with more than 7% improvement. This means that modeling temporal dynamics for video object recognition significantly improves performance. It has been seen that BiLSTM object model also improves the accuracy compared to LSTM. However, the effect of $\boldsymbol{d^o}$ is very low, but we will later show its effect on action results.

4.4.3. Recurrent-Based Action Models

We make use of the outputs of the recurrent-based verb and object models and we train action models based on different fusion strategies (see Table 7).

Comparing various fusion strategies, we show simple multiplication results in comparable performance with more complicated fusion strategies without any training. Moreover, we show late fusion strategies can achieve better recognition compared to early fusion with many training advantages. Moreover, the results of the late fusion strategies outperform other strategies with addition of bw+fw features and spatial layout feature d^o. Finally, comparing softmax and fc7 results, we observe that RNN achieved comparable performance on simple softmax features.

Fusion Model	Verb Base Model	Feature	Verb/Object/Action-RNN	mAcc (%)
Verb-Object Multiplication	C3D full+hand	softmax	BiLSTM	45.29
Verb-Object Early Fusion	C3D full+hand	softmax	LSTM	44.36
Verb-Object Early Fusion	C3D full+hand	softmax (max-pooling)	LSTM	44.10
Verb-Object Early Fusion	C3D full+hand	softmax	BiLSTM	45.18
Verb-Object Early Fusion	C3D full+hand	softmax (max-pooling)	BiLSTM	46.47
Verb-Object Late Fusion	C3D full+hand	softmax	BiLSTM	45.45
Verb-Object Late Fusion	C3D full+hand	softmax	BiLSTM Bw+Fw	49.01
Verb-Object Action Baseline	C3D full	softmax	BiLSTM	25.36
Verb-Object Multiplication	C3D full+hand	fc7	BiLSTM	45.54
Verb-Object Late Fusion	C3D full+hand	fc7	BiLSTM	46.62
Verb-Object Late Fusion	C3D full+hand	fc7	BiLSTM Bw+Fw	48.46
Verb-Object Late Fusion	C3D full+hand	fc7 (with d^o)	BiLSTM Bw+Fw	**50.29**

Table 7: Action recognition results with mean class accuracies (mAcc) on videos using recurrent-based fusion strategies. Bw means Backward function output and Fw means Forward function output. Concatenation is applied to merge full and hand scale verb features unless specified as ax-pooling. d^o indicates that distance-based features are also integrated while modelling object features. Otherwise, s^o features are used.

Recurrent verb-object multiplication. In this fusion setting, action category of a video is simply identified with multiplication of verb-RNN and object-RNN score vectors (see Figure 2 (b)). The verb category score vector $\mathbf{v}$ is extracted using the verb model *C3D full+hand(concatenation)-softmax-BiLSTM* (see row 2 in Table 5 and row 1 in Table 4) and object vector $\mathbf{o}$ is extracted using the object model *YOLO-s^o-BiLSTM* (see Table 6 and row 1 in Table 4), respectively. Given a test video, we simply multiply the verb and the object vectors. The verb-object pair with the maximum value of the matrix obtained by multiplication is selected as the predicted action category of the video. This experiment is applied over split1 with a 45.29% mAcc. We observe that temporal action model significantly outperforms simple Count-Based Verb-Object Multiplication model with 11.42% gain in accuracy (see Section 4.3.4 and Table 7). This means that modelling the temporal dynamics helps in action recognition as well.

Similar experiment is conducted using fc7 features for the verb model *C3D full+hand(concatenation)-fc7-BiLSTM* (see row 6 in Table 5 and row 1 in Table 4) with the same object model. We obtain 45.54% accuracy with a slight improvement over softmax score features.

Recurrent verb-object early fusion. In this fusion setting, action recognition is performed utilizing RNN structures over combined low-level verb-object representations (see Figure 2 (c)). In this experiment, either max-pooling or concatenation is applied to the verb softmax values from full-scale and hand-scale verb models. Combined verb scores are concatenated with object softmax values, and final values are employed by the RNN action model. According to BiLSTM results, 46.47% and 45.18% accuracies are achieved for action recognition over split1 using max-pooling and concatenation, respectively. According to LSTM results, 44.10% and 44.36% accuracies are achieved for action recognition over split1 for max-pooling and concatenation, respectively. For both combination types, experimental results show that BiLSTM structure improves the accuracy compared to LSTM in Table 7. The best BiLSTM early fusion model with 46.47% mAcc is simply constructed using 1 layer with 100 cell size (see row 2 in Table 4). Please note that we do not conduct fc7 experiments for early fusion, since the dimensions of fc7 verb features and object score vector $\boldsymbol{s^o}$ are imbalanced for concatenation.

Recurrent verb-object late fusion. In this fusion setting, action recognition is performed with a shallow neural network that takes the RNN encodings of individual verb and object streams as the concatenated verb category score vector $\mathbf{v}$ and object category score vector $\mathbf{o}$ as inputs (see Figure 2 (d)). Here, the verb vector $\mathbf{v}$ is the output of the verb model *C3D full+hand(concatenation)-softmax-BiLSTM* (see Table 5) and the object vector $\mathbf{o}$ is the output of the object model *YOLO-s^o-BiLSTM* (see Table 6). As given in Table 7, 45.45% accuracy is obtained over split1 with ($[\overrightarrow{\mathbf{v}}, \overrightarrow{\mathbf{o}}]$) (see row 3 in Table 4). This result is slightly lower than the early fusion strategy. Using the concatenated outputs of forward and backward functions, $[\overrightarrow{\mathbf{v}}, \overleftarrow{\mathbf{v}}, \overrightarrow{\mathbf{o}}, \overleftarrow{\mathbf{o}}]$, the model performance later improves up to 49.01% mAcc.

Conducting experiments using fc7 layer as verb features, we obtain 46.62% accuracy using forward feature as input of the shallow network and 48.46% accuracy using forward and backward feature combination as input of the shallow network. While fc7 provides improvement over softmax feature with BiLSTM, it shows slightly worse performance for BiLSTM Bw+Fw. If we further extend the inputs with an object model using $[\boldsymbol{s^o}, \boldsymbol{d^o}]$ representation (the other object models use $\boldsymbol{s^o}$), the performance is 50.29%. Figure 5 shows a comparison chart reporting category based accuracies and results indicate that hand-object interaction through $\boldsymbol{d^o}$ improves action recognition in 48 categories with 1.83% again in accuracy. By analyzing the improved samples, we found that $\boldsymbol{d^o}$ helps to correct the verb and action predictions for many video instances. $\boldsymbol{d^o}$ models the layout of hands and objects with respect to each other within a frame and our RNN structure models the temporal

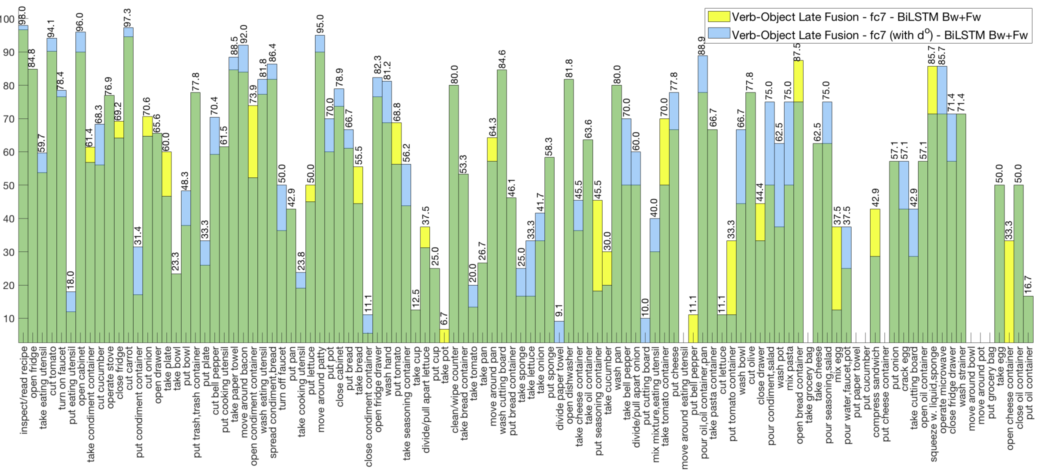

Figure 5: Comparison of Action Models *Verb-Object Late Fusion-fc7-BiLSTM Bw+Fw* with 48.46% accuracy and *Verb-Object Late Fusion-fc7 (with d^o)-BiLSTM Bw+Fw* with 50.29% accuracy (see Table 7). Maximum value per category over two models are shown (best viewed in color).

dynamics of the layout within frame and across frames. This helps to correct verb and action predictions as shown in Figure 6.

These experiments show that (i) fc7 features, (ii) forward and backward extension, and (iii) object model with spatial layout $\boldsymbol{d^o}$ improve the performance significantly. Figure 7 shows same recognition results using the best performed C3D model and how it predicts in challenging video cases. The best shallow network with 50.29% mAcc consists of 3 fully connected layers with 4096, 1024, 512 neurons and a 106-dimensional softmax output (see row 4 in Table 4).

Recurrent action baseline. Recurrent action baseline is also experimented over C3D action model (see Figure 2 (f)). In this experiment, accuracy reaches to 25.36% in Table 7. Result shows that the Recurrent Action Baseline model is better than the Count-Based Action Baseline model due to modelling of temporal dynamics in action videos (see Table 3), and the two-stream recurrent fusion strategies that rely on either early or late fusion are better than the baseline models.

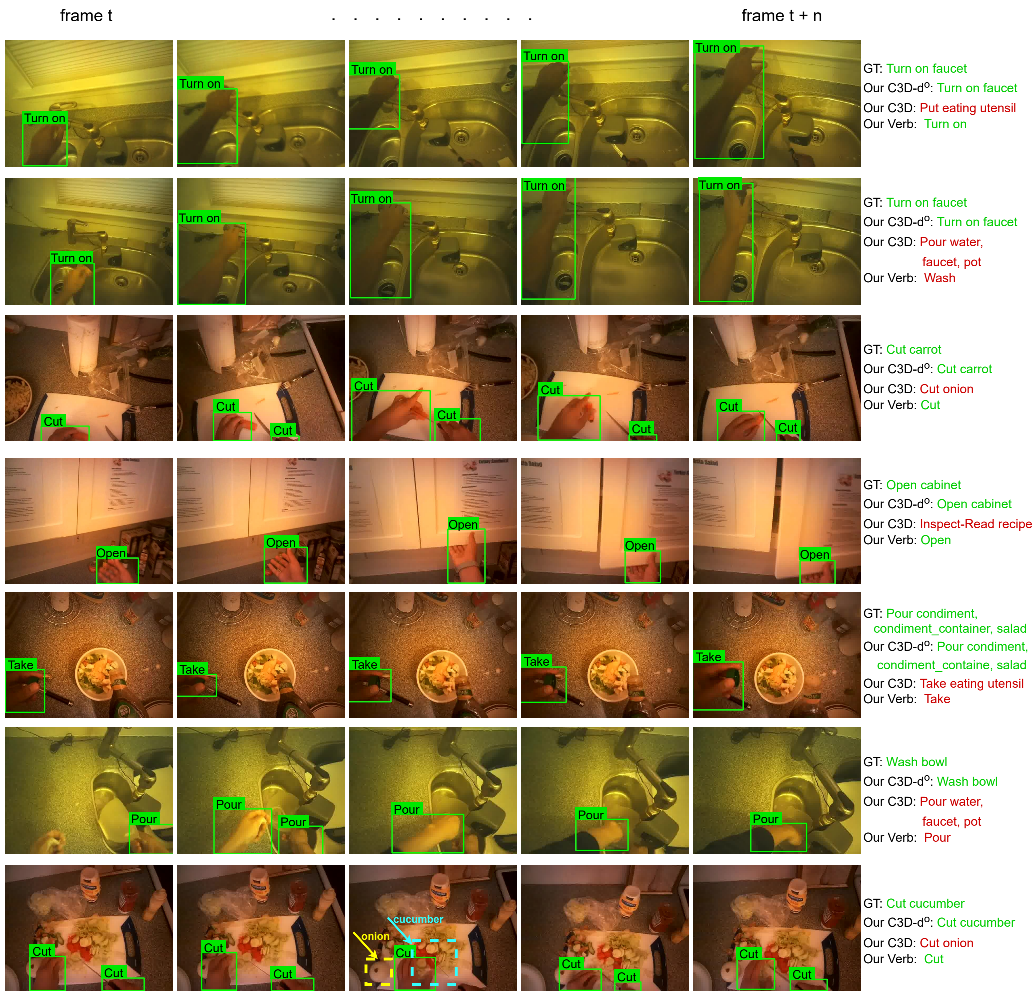

Figure 6: Visual comparison of action models for test samples from EGTEA Gaze+ dataset to evaluate the effect of spatial layout $\boldsymbol{d^o}$ (best viewed in color). For fair comparison, the late fusion strategies of *Our C3D-d^o* (*Verb-Object Late Fusion-fc7(with d^o)-BiLSTM Bw+Fw*) and *Our C3D* (*Verb-Object Late Fusion-fc7-BiLSTM Bw+Fw*) (see Table 7) action models are considered. Both action models are constructed using same verb model. It can be easily appeared from the visual samples that action model with object model with distance scores d^o improves the action performances and corrects the verb and object predictions. For instance, for the video clip sample in the last row, although the verb action is predicted as *cut* and there is *onion* object in the background, *Our C3D* fails by predicting the video as *cut onion* action category. *Our C3D-d^o* action model predicts action category of *cut cucumber* correctly since it takes into consideration the hand and *cucumber* object locations and interaction.

Figure 7: Visualization of same video samples to show the improvements of C3D action model for the failure cases of the verb model. The predictions of the best action model using *Our C3D-d^o (Verb-Object Late Fusion-fc7(with d^o)-BiLSTM Bw+Fw)*(see Table 7) and the predictions of its verb-RNN stream are illustrated. Even if the verb model predicts incorrectly, action model corrects the prediction of the verb stream. The verb predictions are confused due to the similar background of the video and the similarity of the hand movements. For instance, in the first row, the hand action is categorized as *cut* since the background and interacted objects are proper for that action although the ground truth verb action is *take*. Our C3D action model handles this failure cases of verb model and predicts the video correctly as *cut eating utensil* with correction of verb category.

Verb Base Model	Scale	Feature (pooled)	Verb-RNN	mAcc (%)
Pre-trained I3D	full –	Mixed-5c (1x1x1x1024) –	BiLSTM	66.03
Pre-trained I3D	full hand	Mixed-5c (1x1x1x1024) MaxPool3d-5a-2x2 (1x1x1x832)	BiLSTM	71.62
Pre-trained I3D	full hand	Mixed-5c (3x1x1x1024) MaxPool3d-5a-2x2 (1x1x1x832)	BiLSTM	73.15
Our I3D	full hand	Mixed-5c (3x1x1x1024) MaxPool3d-5a-2x2 (1x1x1x832)	BiLSTM	**73.43**
Our I3D	full hand	Mixed-5c (3x1x1x1024) MaxPool3d-5a-2x2 (1x1x1x832)	BiLSTM+Attention	**74.81**

Base Model	Scale	Feature (pooled)	Verb/Object-RNN	mAcc (%)
Pre-trained I3D	full hand	Mixed-5c (1x1x1x1024) MaxPool3d-5a-2x2 (1x1x1x832)	BiLSTM Bw+Fw	51.15
Pre-trained I3D	full hand	Mixed-5c (3x1x1x1024) MaxPool3d-5a-2x2 (1x1x1x832)	BiLSTM Bw+Fw	53.82
Our I3D	full hand	Mixed-5c (3x1x1x1024) MaxPool3d-5a-2x2 (1x1x1x832)	BiLSTM Bw+Fw	**54.56**
Our I3D	full hand	Mixed-5c (3x1x1x1024) MaxPool3d-5a-2x2 (1x1x1x832)	BiLSTM (Attention) Bw+Fw	**56.07**

Table 8: I3D verb and action recognition results on split1 videos with recurrent-based models. The action models are based on recurrent verb-object late fusion strategy, and trained using backward and forward function outputs. The object stream used in action models is based on model *YOLO-*s^o + d^o*-BiLSTM*. The reported results as the last verb and action models are trained with attention module. Attention module is applied both on verb and object models, but we obtain no improvement for object recognition.

4.4.4. Other 3D ConvNet Architectures: I3D

The base encoding models used for the feature extraction may improve the recognition rates significantly. Following this, we extend our experiments by using other 3D ConvNet architecture called I3D RGB [40] to categorize clips (following the original setting we use 25-frame clips) as the base model for the verb stream and we conduct two experiments. First, we use the pre-trained I3D model that is trained on the Kinetics dataset with 400 action categories [2]. Later, we train a shallow neural network model that fine-tunes over I3D intermediate features using our verb and action annotations. In both cases, the base model encodes videos as verb matrices and we train our verb and action models using recurrent models on top of these matrices as before. We report the verb and the action recognition accuracies, respectively, in Table 8. Please note that we conduct I3D experiments with the setting that is best performed for C3D, therefore we investigate the Recurrent Verb-Object Late Fusion strategy that performs best for C3D (given in Table 7) and the object

[2]https://github.com/deepmind/kinetics-i3d

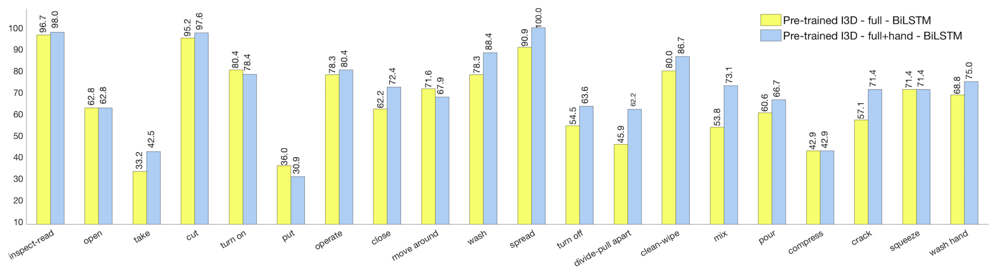

Figure 8: Comparison of multiple scale I3D verb models over 20 verb categories of the EGTEA Gaze+ dataset. Yellow bars show the recognition results of verb model *Pre-trained I3D-full-BiLSTM* and blue shows the results of verb model *Pre-trained I3D-full+hand-BiLSTM* (see in Table 8). Multiple scales using full and hand scales improve recognition with 5.59% gain.

stream is object model *YOLO-s^o+d^o-BiLSTM*. Figure 9 shows predictions over a set of test samples using I3D and C3D models. When we analyze the samples, we observe I3D resolves some challenging cases and improves recognition performance over C3D. I3D is a dense network compared to C3D and intermediate layers consists of inception modules. Using a more advance base model for verb stream, we improve recognition performance by 4.27%.

For I3D experiments, two intermediate layers from the I3D model are selected for encoding the full-scale and the hand-scale verb information, respectively. For the full-scale encoding used to fetch coarser details in clips, we pick the outputs of $3\times7\times7\times1024$-dimensional Mixed-5c layer. On the other hand, for the hand-scale encoding corresponding to finer details on hands, we pick the outputs of an earlier layer, $3\times14\times14\times832$-dimensional MaxPool3d-5a-2x2 layer for hand-volumes. We apply 20% enlargement on detected hand regions before extracting features of hand-volumes. Finally, each clip is represented as the concatenation of these features. If there are multiple hands, we apply max-pooling on features of hand-volumes.

Using pre-trained I3D model, we first examine how the multiple scales help in performance and compare full-scale (66.03% mAcc) vs. combination of full-scale and hand-scale models (71.62% mAcc). We also show the details for all verb categories in Figure 8. There is a significant improvement over 18 verb categories with the addition of fine-grained details through hand-scale model. This is what we expect from the combination of fine and coarse scale features. Then, we experiment on two different feature setting. We obtain 71.62% mAcc and 73.15% mAcc for verb recognition (the BiLSTM

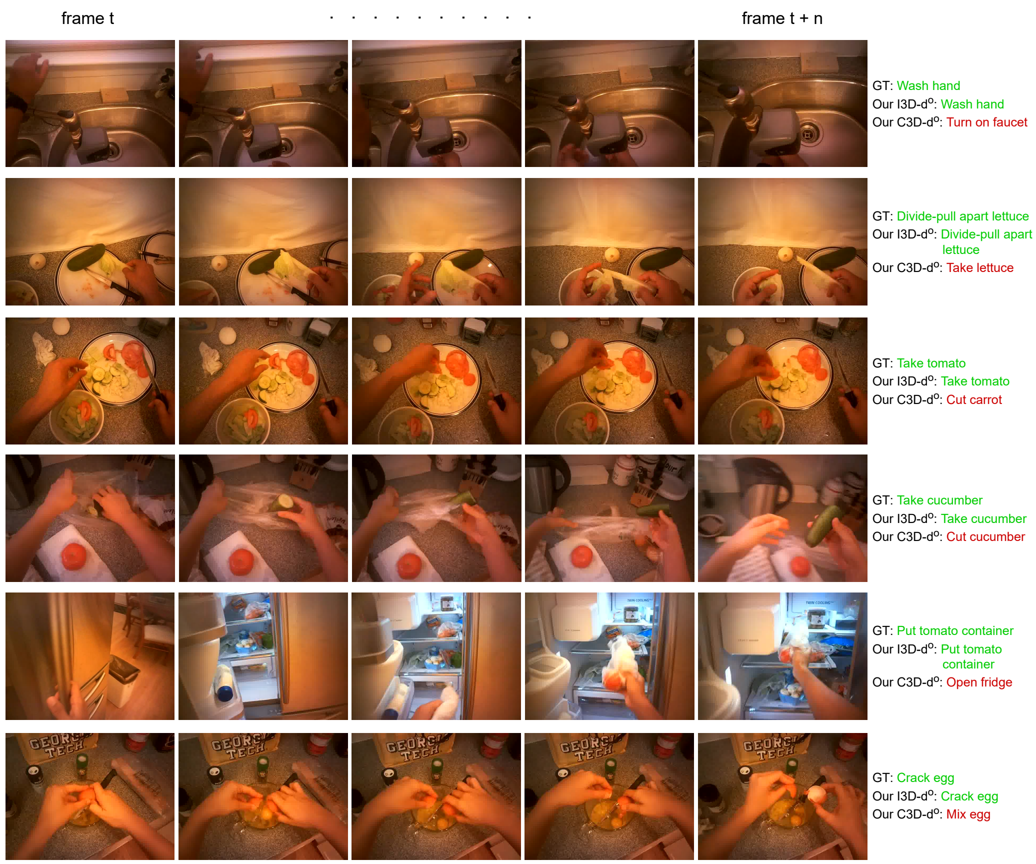

Figure 9: Visual comparison of predictions of I3D and C3D based action models on test samples from EGTEA Gaze+ dataset (best viewed in color). For fair comparison, the same late fusion settings of C3D-d^o (*Verb-Object Late Fusion-fc7(with d^o)-BiLSTM Bw+Fw*) from Table 7 and I3D-d^o (*Our I3D-full+hand-BiLSTM Bw+Fw*) from Table 8 action models are considered. Both models are constructed using the same object model. These challenging videos are predicted true by I3D action model while predicted false by C3D action model. The illustration of successive frames of videos shows the challenging videos are truly predicted by I3D while missed by C3D. For the video sample in the first row, while the *turn-on faucet* action is performed at the beginning of the video, I3D based action model predicts the main video action correctly. The same challenge is also valid for the second and fifth row video samples. In the third row video sample, both object and verb categories are wrong in C3D. Although the background is proper for *cut* verb action, I3D overcomes this challenging situation.

verb models contain 1 layer with 728 cells). Moreover, we obtain 51.15% mAcc and 53.82% mAcc in action recognition (the action models contain 2 1024-dimensional fully connected layers with dropout rate of 0.5). In the first feature setting, the full-scale features and the hand-scale features are pooled both in spatial and temporal dimensions, and then concatenated. In the second one, the full-scale features are pooled just in the spatial dimension and concatenated with the pooled hand-volume features. Results show that higher dimensional representations encode the data better. Please note that further experiments can be conducted on other network layers with various pooling settings to improve the recognition performance. Keeping the features as in the original dimension is good to encode spatial and temporal information, but here we prefer to apply pooling over spatial domain to decrease feature dimensions.

Our fine-tuned model is trained over extracted I3D features. Unlike two separate models introduced on C3D for full-scale verb and hand-scale verb features respectively, here we train a single model on concatenated I3D full-scale and hand-scale features. Representing each clip with a concatenated feature vector, we train a verb model over ground truth action clips. This model contains 2 1024-dimensional fully connected layers with dropout rate of 0.5. Using this verb model, we represent each verb category score vector $\mathbf{v}$ using the 1024-dimensional second fully connected layer output. Later, $\mathbf{v}$ and $\mathbf{o}$ are concatenated and fed into the shallow network. This shallow network similarly contains 2 1024-dimensional fully connected layers with dropout rate of 0.5 (see row 5 in Table 4) . The verb and the action models result in recognition accuracies of 73.43% and 54.56%, respectively. Our supervised setting slightly performs better than the pre-trained models, since it provides fine-tuning over EGTEA Gaze+ dataset.

4.4.5. Recurrent Models with Attention

As proposed in Section 3.3, we improve fusion strategies with a self-attention module and we propose a model called Recurrent Verb-Object Attention Late Fusion strategy. The attention module can be easily applied to early fusion strategy that reduces verb and object streams into a single pathway in earlier stages, but we work on late fusion since it is superior in recognition performance (see Table 7). Please note that we conduct experiments for the best performing setting, therefore attention module is applied for late fusion strategy and we use our I3D models. The verb and object BiLSTM models with attention module are constructed using 1 layer with 512

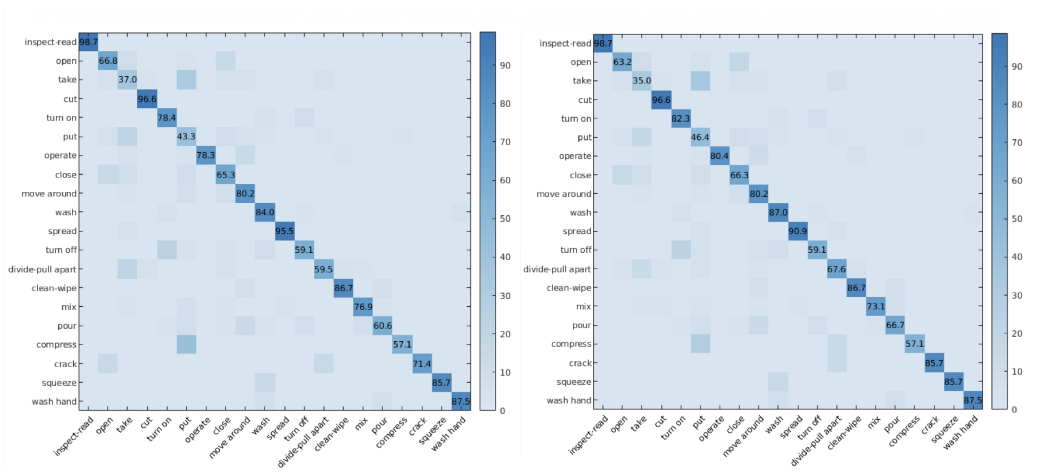

Figure 10: Confusion matrices reporting the performances of verb models *Our I3D-full+hand-BiLSTM* (left) and *Our I3D-full+hand-BiLSTM+Attention* (right) with 73.43% and 74.81% mAcc, respectively, over 20 verb categories of the EGTEA Gaze+ dataset (see Table 8).

and 256 cell sizes, respectively. Finally, the action model using Verb-Object Attention Late Fusion setting contains two 2048-dimensional fully connected layers with dropout rate of 0.5 (see row 6 in Table 4).

In Verb-Object Attention Late Fusion setting, the verb, object and action model results are 74.81%, 70.83% and 56.07%, respectively. According to the results, the recurrent verb and action models with attention block outperform the non-attention models as seen in Table 8. The verb model results in 1.38% gain and action model results in 1.51% gain in recognition accuracies. However, we observe no improvement on object model. For comparison, more details can be found in Figure 10 and in Figure 11. We also show some failure cases in Figure 12. The failures are caused by the similarity of the object categories and hand movements.

5. Comparison and Discussion

This section presents the comparison of the proposed model on the EGTEA Gaze+ dataset for action recognition, as well as a detailed discussion.

Table 9 reports results of our proposed method with the state-of-the-art models and it shows that our performance is comparable with the state-of-the-art. Baseline models I3D RGB, I3D Joint, I3D+Gaze returns 47.26%, 49.79%, and 51.21% accuracies, respectively. Results show that I3D is a

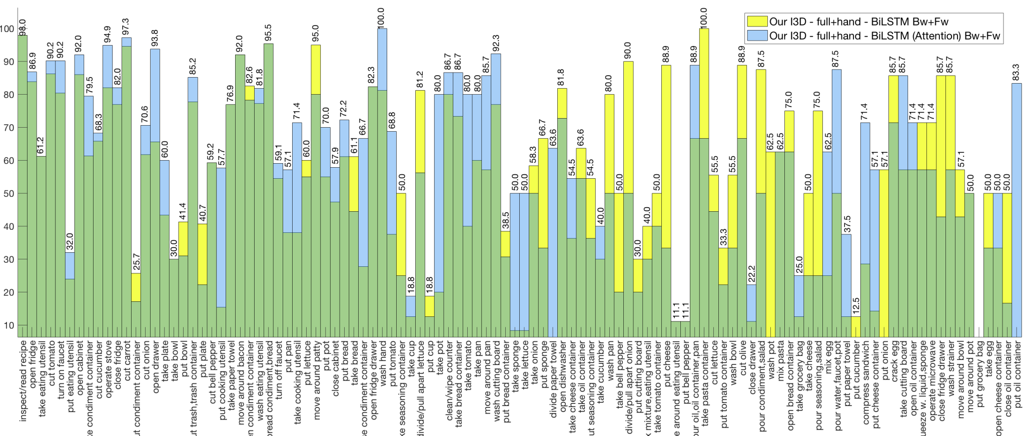

Figure 11: Comparison of action models *Our I3D-full+hand-BiLSTM Bw+Fw* and *Our I3D-full+hand-BiLSTM(Attention) Bw+Fw* with 54.56% and 56.07% mAcc over 106 action categories of the EGTEA Gaze+ dataset (best viewed in color).

Action Models	**mAcc (%)**
EgoIDT + Gaze [30]	46.50
I3D RGB [13]	47.26
I3D Joint [13]	49.79
I3D+Gaze [13]	51.21
Li et al. [13]	53.30
MCN [14]	55.63
Ours	**56.07**
LSTA-RGB [45]	57.94
RU [46]	60.20
LSTA [45]	61.86

Table 9: The comparison with state-of-the-art action recognition models on the first split of EGTEA Gaze+ dataset with accuracy in mAcc.

powerful feature, and I3D Joint with joint modelling of RGB and Flow features improves the recognition significantly. Moreover, I3D+Gaze has the highest accuracy among three, since gaze is an important clue for egocentric videos. Our models rely only on RGB modality, and they are better than the I3D based models and Li et al. [13]. This means the performance gap can be increased with integration of other modalities into our pipeline. Integration is simple, where each stream of our verb-object decomposition model can be further extended with two-stream approaches to add Flow or Gaze

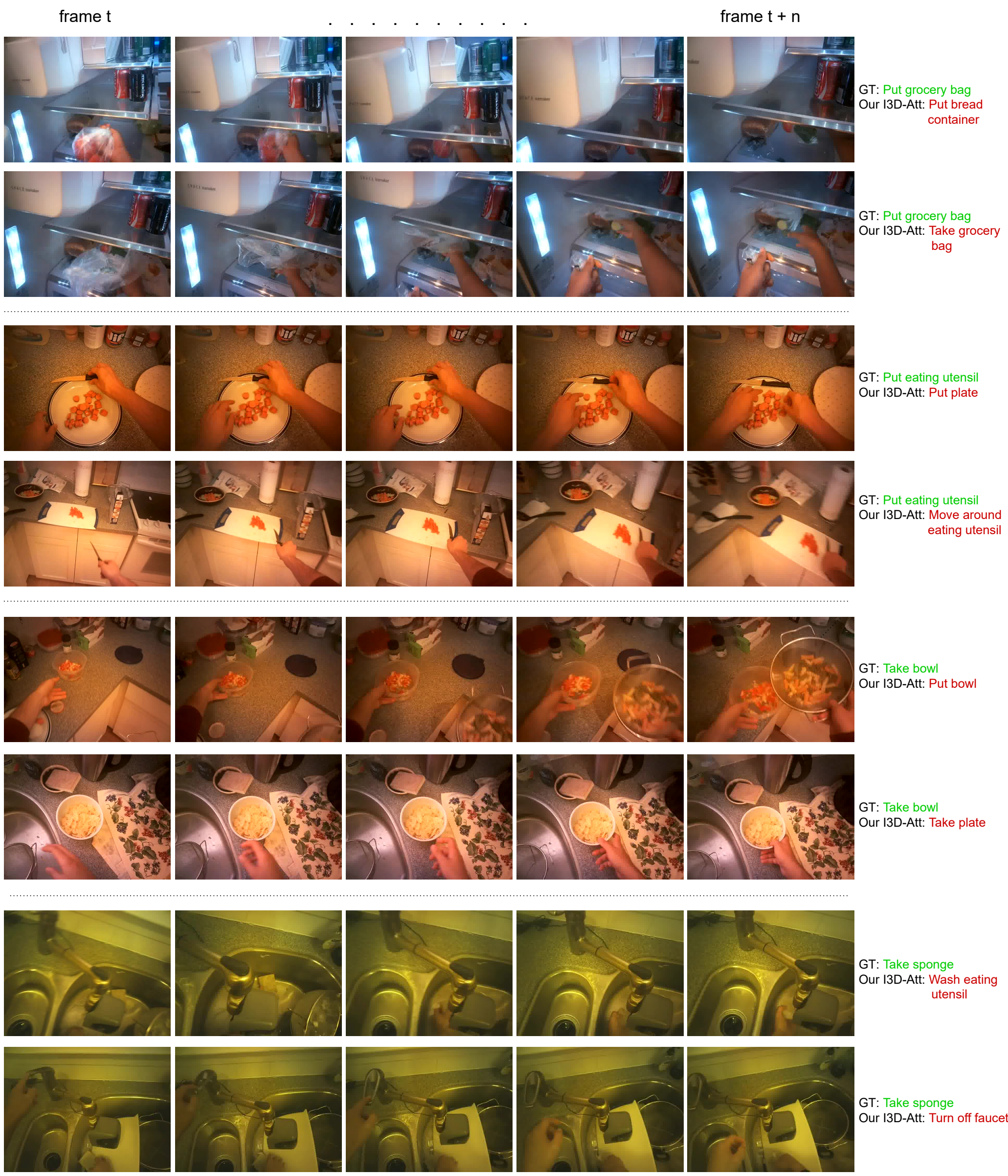

Figure 12: Failure cases of *I3D-Att* (*Our I3D-full+hand-BiLSTM(Attention) Bw+Fw*, see Table 8) action model on a set of test samples from split1.

modalities.

We report lower results than the recently published studies the RU [46] and the LSTA [45]. In [46], the Rolling-Unrolling LSTM (RU) processes appearance from RGB frames, motion from optical flow, as well as object features. These modalities are fused using an attention mechanism. Similarly, LSTA [45] is a two-stream model; one stream for encoding appearance information from RGB frames and the second stream for encoding motion information from optical flow. Both models use flow as a low-level feature. Following a similar discussion, we state that our model focuses on a single modality with RGB features. The standalone performance of LSTA RGB stream, LSTA-RGB, [45] has 57.94% mAcc and this is slightly higher than our best model with 56.07%. Please note that we use the term two-stream to indicate the verb-object decomposition of action model, but other models use the term of two-stream architecture following the work [32] where each stream models a different low-level modality, namely RGB and Flow. This means that the performance of object and verb streams can be further improved with other modalities.

Investigating fine-grained recognition in first-person view, our aim is on how the recognition rates can be improved within the model just using RGB modality. We focus on hands, their actions in multiple scales and their interactions with other objects. Verb stream is modelled using RGB with a single modality and 3D convolutional neural network models are investigated for modelling multiple scales of hand regions. Similarly, object-stream is modelled using RGB frames and spatial layout features.

On the other hand, our proposed model is based on action decomposition with two semantically meaningful components, verb and object. In this work, we show that we achieve comparable results with the state-of-the-art models. Even if we have slightly lower performance than some of the approaches, our model has many architectural advantages over conventional action recognition models. First, decomposition is good for zero-shot learning as proposed in [7]. Second, in large scale datasets, the number of video instance can vary for each category, and while some categories have a large amount of training samples, some other categories have very few training samples. Through decomposition based models, we have less number of categories with more samples to train each component of neural network architectures, and this helps to solve the problem related to dataset imbalance. Moreover, models based on action decomposition are architecturally more flexible for extending the model later for more categories. For example, fixing the object component with trained model, an addition of a new verb category will cost to fine-tune the verb component and the simple fusion model (observing late fusion outperforms early fusion).

6. Conclusion

We have developed compositional model including two complementary steps, verb and object, to perform action recognition in first-person videos. Late and early fusion strategies, based on recurrent neural network structures pools verb models and object model to recognize the video at action level. Experimental results show that decomposing actions model into verb and object using recurrent neural networks significantly improves the performance compared to the baseline action model for large number of action classes. Hand information is an important clue to determine the action in the first-person vision. We have shown that spatial-temporal modelling of hand regions improves both verb and action recognition performances.

References

[1] R. Megret, D. Szolgay, J. Benois-Pineau, P. Joly, J. Pinquier, J.-F. Dartigues, C. Helmer, Wearable video monitoring of people with age dementia: Video indexing at the service of helthcare, in: International Workshop on Content-based Multimedia Indexing, 2008, pp. 101–108.

[2] G. Meditskos, P.-M. Plans, T. G. Stavropoulos, J. Benois-Pineau, V. Buso, I. Kompatsiaris, Multi-modal activity recognition from egocentric vision, semantic enrichment and lifelogging applications for the care of dementia, Journal of Visual Communication and Image Representation 51 (2018) 169–190.

[3] N. Das, E. Ohn-Bar, M. M. Trivedi, On performance evaluation of driver hand detection algorithms: Challenges, dataset, and metrics, in: International Conference on Intelligent Transportation Systems, 2015, pp. 2953–2958.

[4] J. Lee, M. S. Ryoo, Learning robot activities from first-person human videos using convolutional future regression, in: IEEE Conference on Computer Vision and Pattern Recognition Workshops, 2017, pp. 1–2.

[5] H. Rhodin, C. Richardt, D. Casas, E. Insafutdinov, M. Shafiei, H.-P. Seidel, B. Schiele, C. Theobalt, Egocap: egocentric marker-less motion capture with two fisheye cameras, ACM Transactions on Graphics (TOG) 35 (2016) 162.

[6] M. Ma, H. Fan, K. M. Kitani, Going deeper into first-person activity recognition, in: IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 1894–1903.

[7] Y. C. Zhang, Y. Li, J. M. Rehg, First-person action decomposition and zero-shot learning, in: IEEE Winter Conference on Applications of Computer Vision, IEEE, 2017, pp. 121–129.

[8] H. Wang, C. Schmid, Action recognition with improved trajectories, in: Proceedings of the IEEE international conference on computer vision, 2013, pp. 3551–3558.

[9] A. Fathi, X. Ren, J. M. Rehg, Learning to recognize objects in egocentric activities, in: IEEE Conference on Computer Vision and Pattern Recognition, 2011, pp. 3281–3288.

[10] M. Cai, F. Lu, Y. Gao, Desktop action recognition from first-person point-of-view, IEEE Transactions on Cybernetics 49 (2018) 1616–1628.

[11] T. Ishihara, K. M. Kitani, W.-C. Ma, H. Takagi, C. Asakawa, Recognizing hand-object interactions in wearable camera videos, in: IEEE International Conference on Image Processing, 2015.

[12] J. Kumar, Q. Li, S. Kyal, E. A. Bernal, R. Bala, On-the-fly hand detection training with application in egocentric action recognition, in: IEEE Conference on Computer Vision and Pattern Recognition Workshops, 2015, pp. 18–27.

[13] Y. Li, M. Liu, J. M. Rehg, In the eye of beholder: Joint learning of gaze and actions in first person video, in: Proceedings of the European Conference on Computer Vision, 2018, pp. 619–635.

[14] Y. Huang, Z. Li, M. Cai, Y. Sato, Mutual context network for jointly estimating egocentric gaze and actions, arXiv preprint arXiv:1901.01874 (2019).

[15] E. H. Spriggs, F. De La Torre, M. Hebert, Temporal segmentation and activity classification from first-person sensing, in: IEEE Conference on Computer Vision and Pattern Recognition Workshops, 2009, pp. 17–24.

[16] A. Fathi, A. Farhadi, J. M. Rehg, Understanding egocentric activities, in: IEEE International Conference on Computer Vision, 2011, pp. 407–414.

[17] H. Pirsiavash, D. Ramanan, Detecting activities of daily living in first-person camera views, in: IEEE Conference on Computer Vision and Pattern Recognition, IEEE, 2012, pp. 2847–2854.

[18] M. S. Ryoo, B. Rothrock, L. Matthies, Pooled motion features for first-person videos, in: IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 896–904.

[19] Y. Poleg, A. Ephrat, S. Peleg, C. Arora, Compact cnn for indexing egocentric videos, in: IEEE Winter Conference on Applications of Computer Vision, IEEE, 2016, pp. 1–9.

[20] Y. Zhou, B. Ni, R. Hong, X. Yang, Q. Tian, Cascaded interactional targeting network for egocentric video analysis, in: IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 1904–1913.

[21] S. Singh, C. Arora, C. Jawahar, Trajectory aligned features for first person action recognition, Pattern Recognition 62 (2017) 45–55.

[22] R. E. Schapire, Explaining adaboost, in: Empirical inference, Springer, 2013, pp. 37–52.

[23] N. Dalal, B. Triggs, Histograms of oriented gradients for human detection, in: IEEE Conference on Computer Vision and Pattern Recognition, 2005.

[24] A. Cartas, P. Radeva, M. Dimiccoli, Contextually driven first-person action recognition from videos, in: presentation at EPIC@ ICCV2017 workshop, 2017, p. 8.

[25] J. Pont-Tuset, P. Arbelaez, J. T. Barron, F. Marques, J. Malik, Multiscale combinatorial grouping for image segmentation and object proposal generation, IEEE Transactions on Pattern Analysis and Machine Intelligence 39 (2017) 128–140.

[26] G. Gkioxari, R. B. Girshick, J. Malik, Contextual action recognition with r*cnn, in: IEEE International Conference on Computer Vision, 2015, pp. 1080–1088.

[27] A. Fathi, Y. Li, J. M. Rehg, Learning to recognize daily actions using gaze, in: European Conference on Computer Vision (ECCV), 2012, pp. 314–327.

[28] Y. Li, A. Fathi, J. M. Rehg, Learning to predict gaze in egocentric video, in: IEEE International Conference on Computer Vision, 2013, pp. 3216–3223.

[29] K. M. Kitani, T. Okabe, Y. Sato, A. Sugimoto, Fast unsupervised ego-action learning for first-person sports videos, in: IEEE Conference on Computer Vision and Pattern Recognition, 2011, pp. 3241–3248.

[30] Y. Li, Z. Ye, J. M. Rehg, Delving into egocentric actions, in: IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 287–295.

[31] H. Wang, A. Kläser, C. Schmid, L. Cheng-Lin, Action recognition by dense trajectories, in: IEEE Conference on Computer Vision and Pattern Recognition, 2011, pp. 3169–3176.

[32] K. Simonyan, A. Zisserman, Two-stream convolutional networks for action recognition in videos, in: Advances in Neural Information Processing Systems, 2014, pp. 568–576.

[33] Y. Tang, Y. Tian, J. Lu, J. Feng, J. Zhou, Action recognition in rgb-d egocentric videos, in: International Conference on Image Processing, 2017, pp. 3410–3414.

[34] M. Hahn, N. Ruiz, J.-B. Alayrac, I. Laptev, J. M. Rehg, Learning to localize and align fine-grained actions to sparse instructions, arXiv preprint arXiv:1809.08381 (2018).

[35] G. Kapidis, R. Poppe, E. van Dam, L. P. Noldus, R. C. Veltkamp, Egocentric hand track and object-based human action recognition, arXiv preprint arXiv:1905.00742 (2019).

[36] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 770–778.

[37] D. Damen, H. Doughty, G. Maria Farinella, S. Fidler, A. Furnari, E. Kazakos, D. Moltisanti, J. Munro, T. Perrett, W. Price, et al., Scaling egocentric vision: The epic-kitchens dataset, in: Proceedings of the European Conference on Computer Vision, 2018, pp. 720–736.

[38] J. Redmon, A. Farhadi, Yolov3: An incremental improvement, arXiv preprint arXiv:1804.02767 (2018).

[39] D. Tran, L. Bourdev, R. Fergus, L. Torresani, M. Paluri, Learning spatiotemporal features with 3d convolutional networks, in: IEEE International Conference on Computer Vision, 2015, pp. 4489–4497.

[40] J. Carreira, A. Zisserman, Quo vadis, action recognition? a new model and the kinetics dataset, in: IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 6299–6308.

[41] J. Redmon, A. Farhadi, Yolo9000: better, faster, stronger, in: IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 7263–7271.

[42] L. Shen, S. Yeung, J. Hoffman, G. Mori, L. Fei-Fei, Scaling human-object interaction recognition through zero-shot learning, in: IEEE Winter Conference on Applications of Computer Vision, 2018, pp. 1568–1576.

[43] Z. Lin, M. Feng, C. N. d. Santos, M. Yu, B. Xiang, B. Zhou, Y. Bengio, A structured self-attentive sentence embedding, arXiv preprint arXiv:1703.03130 (2017).

[44] Georgia tech egocentric activity datasets, `http://www.cbi.gatech.edu/fpv/`, ???? Accessed: 2019-06-14.

[45] S. Sudhakaran, S. Escalera, O. Lanz, Lsta: Long short-term attention for egocentric action recognition, in: IEEE Conference on Computer Vision and Pattern Recognition, 2019, pp. 9954–9963.

[46] A. Furnari, G. M. Farinella, What would you expect? anticipating egocentric actions with rolling-unrolling lstms and modality attention, in: IEEE International Conference on Computer Vision, 2019, pp. 6252–6261.