Egocentric Action Recognition by Capturing Hand-Object Contact and Object State

Tsukasa Shiota Motohiro Takagi Kaori Kumagai Hitoshi Seshimo Yushi Aono
NTT Human Informatics Laboratories, NTT Corporation
{tsukasa.shiota, motohiro.takagi, kaori.kumagai, hitoshi.seshimo, yushi.aono}@ntt.com

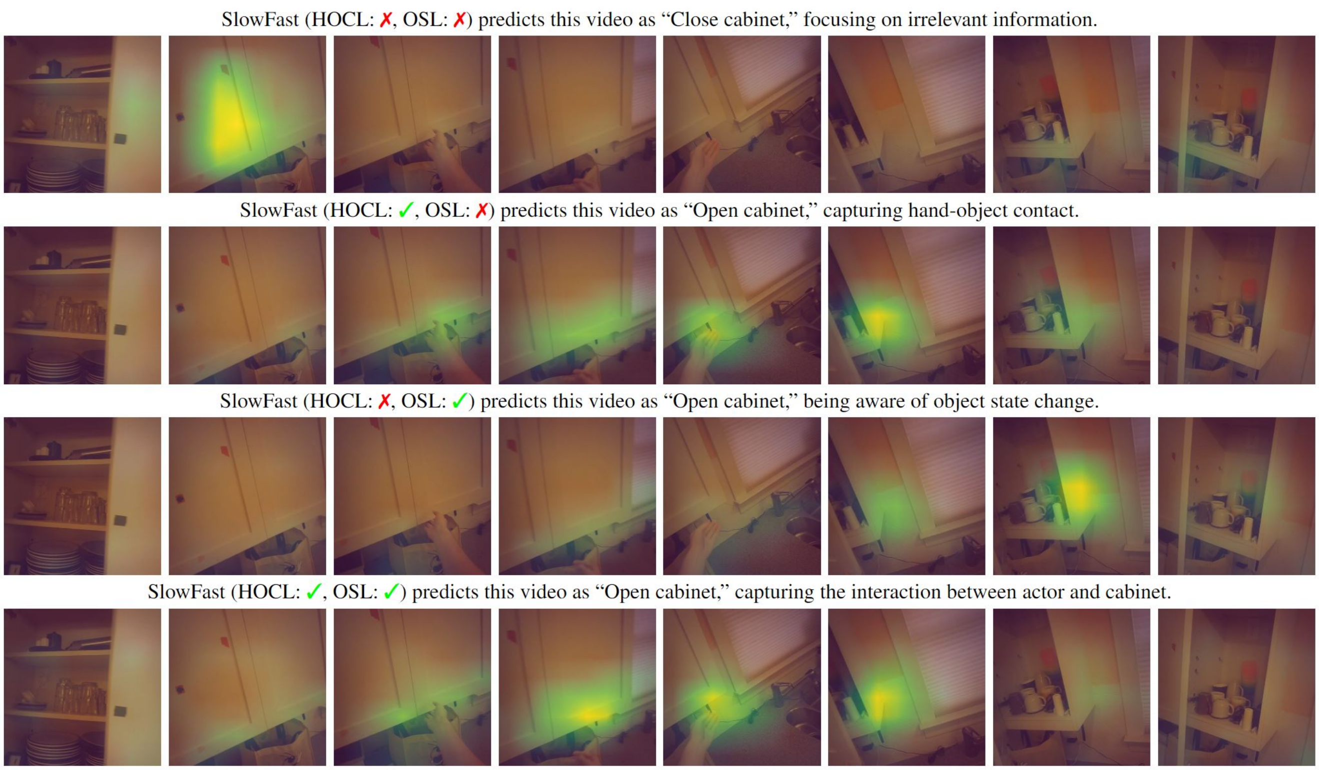

Figure 1. Visualization of the difference in ROI between a model trained with pairs of a video and action label and ones trained with our proposed methods. We used GradCAM [46] for visualization. The video is in EGTEA dataset and shows "Open cabinet" action, in which a person opens a cabinet with his/her left hand. The model trained with only pairs of a video and action label does not focus on the interaction between actor and cabinet (1st row), whereas one trained with our proposed methods does (4th row).

Abstract

Improving the performance of egocentric action recognition (EAR) requires accurately capturing interactions between actors and objects. In this paper, we propose two learning methods that enable recognition models to capture hand-object contact and object state change. We introduce Hand-Object Contact Learning (HOCL), which enables the model to focus on hand-object contact during actions, and Object State Learning (OSL), which enables the model to focus on object state changes caused by hand actions. Evaluation using a CNN-based model and a transformer-based model on the EGTEA, MECCANO, and EPIC-KITCHENS 100 datasets demonstrated the effectiveness of applying HOCL and OSL. Their application improved overall accuracy by up to 2.24% on EGTEA, 3.97% on MECCANO, and 1.49% on EPIC-KITCHENS 100. In addition, HOCL and OSL improved the performance on data with small training samples and one from unfamiliar scenes. Qualitative analysis revealed that their application enabled the models to precisely capture the interaction between actor and object.

1. Introduction

Egocentric action recognition (EAR) becomes a primary task to understand human behavior since some potential applications using first-person-view videos, *e.g*., worker safety [7] and healthcare [38], have recently been developed. Breakthroughs in deep neural networks led to end-to-end action recognition models that utilize convolutional neural networks (CNNs) [29] and transformers [55]. These advances are thanks to the efforts of many researchers who have developed various large-scale datasets that contain third-person-view videos [5,6,24,28,30,44,47,50] and first-person-view videos [8, 9, 22, 23, 31, 43, 49].

Those models, such as SlowFast [20] and Video Swin Transformer [37], generally learn actions using pairs of a video and action label in the end-to-end manner. However, training with only the pair data can often cause them to learn spatio-temporal information irrelevant to actions. This causes EAR models to not perform as well as they should. The 1st row in Figure 1 shows an example using SlowFast. The one trained with pairs of a video and action label does not capture the action, resulting in the wrong prediction.

A principal element to learn spatio-temporal information relevant to actions is to capture (1) the contact between the actor's hands and the objects relevant to the action and (2) how the object's state changes due to the action. As proposed by Gibson's affordance [21], when we interact with surrounding objects, we perceive the object's state, affect the object with our body (mainly hands), and often change the object's state. For example, "Open fridge" in Figure 1 can be viewed as a hand pulling the door of a closed fridge and the state of the fridge changing to open. If EAR models capture those, it is expected not only to improve EAR performance, but also to achieve robustness unaffected by the number of samples per class or the diversity of the shooting scene. This motivates us to design a method to train end-to-end EAR models that appropriately focus on the interaction between hand and object.

In this paper, we introduce two learning methods that enable end-to-end recognition models to better understand actions by capturing hand-object contact and the resulting object state. Our proposed method consists of two types of learning: Hand-Object Contact Learning (HOCL) and Object State Learning (OSL). HOCL is realized by using two models which has the same structure; one learns actions from raw videos, and the other learns them from videos containing only information related to hands and objects. The two models learn actions collaboratively; as a result, the model predicting with the raw videos acquires knowledge related to hand-object contact during actions. OSL is realized by defining a frame-by-frame object state prediction task. The models simultaneously learn the interacting object's state in each frame of the video as well as actions shown in the video. This helps the models to sufficiently learn object state changes from the hand actions.

Evaluation with the SlowFast and Video Swin Transformer on three datasets (EGTEA [31], MECCANO [43], and EPIC-KITCHENS 100 [9]) demonstrated that our methods improve overall accuracy by up to 2.24% on EGTEA, 3.97% on MECCANO, and 1.49% on EPIC-KITCHENS 100. Qualitative analysis demonstrated that models trained using our methods can recognize actions by capturing hand-object contact and object state.

Our contributions are summarized as follows: First, we have designed two learning methods, HOCL and OSL, that enable models to understand actions more precisely by taking into account hand-object interaction. Second, we demonstrated their effectiveness in improving the performance of EAR in different domains and with a range of dataset scales. Third, we showed that they are capable of robust EAR without being affected by the number of samples per class or unfamiliar scenes.

2. Related Work

Action Recognition Models. Many methods for performing action recognition using CNNs [29] have been studied. One approach is to utilize 2D-CNNs. Methods based on this approach recognize actions by extracting frame-level spatial features and aggregating them in the temporal direction using average pooling [51,56] or RNNs [13,33,61]. TSM [34] and RubiksNet [17] were devised to efficiently capture temporal features by training models while shifting the spatio-temporal features of frames. Another prominent approach is to use 3D-CNNs, which extend 2D-CNNs to the temporal dimension [6,19,53]. Since 3D-CNNs have a larger number of parameters than 2D-CNNs and thus higher training costs, advanced methods, such as P3D [41] and R(2+1)D [54], have been devised to reduce model complexity. Feichtenhofer *et al*. proposed SlowFast [20], which has a slow pathway for capturing spatial features at a low frame rate and a fast pathway for capturing temporal features at a high frame rate. It demonstrated high performance among CNN-based methods and thus has been used in many studies.

On the other hand, transformers [55] has attracted much attention in recent years. Unlike CNNs, which repeatedly perform local convolution operations, transformers take into account the global relationships of the data. Various transformer-based models, such as BERT [11], ViT [14], and Swin Transformer [36], have been proposed and extended to video recognition [1, 3, 16, 37, 57]. Liu *et al*. recently reported the Video Swin Transformer [37], which extends the Swin Transformer to video tasks. It captures local spatio-temporal features by defining a set of spatio-temporal directions of patches as a 3D window and computing self-attention between patches in the 3D window. It also captures global spatio-temporal features between 3D windows by shifting the 3D window for each layer.

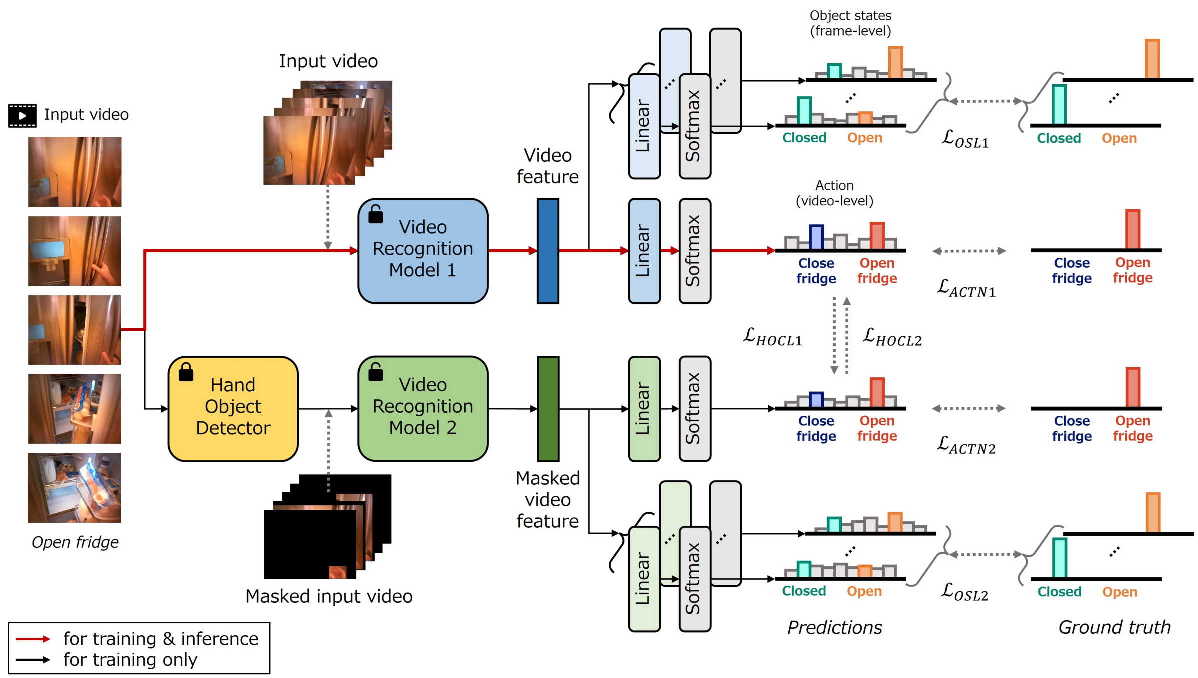

Figure 2. Overview of our proposed method. In the training phase, HOCL and OSL methods are integrated into the action learning. In the inference phase, we only use video recognition model 1, which takes a raw video as input, to classify actions (indicated by a red arrow).

Most models are expected to learn the nature of actions in the end-to-end manner. As our visualization and a previous study has suggested [25], training with pairs of a video and action label often lead to learning actions without focusing on actions, resulting in inadequate performance.

Egocentric Action Recognition. The use of context information, such as the motions of human body parts and the information of active objects, is a promising approach to EAR. Methods have been devised that utilize hand information given that the actor's hands provide important context information [27, 52]. Some studies have approached EAR by utilizing information about where the actor looks during actions, *i.e.*, eye gaze [26, 39]. Since many actions in first-person videos involve interactions between an actor and objects, methods have been devised that utilize information about the active object [18, 35, 58]. Similar to our approach, several reported methods combine more than one type of contextual information for EAR [10, 32, 60]. Recent studies on EAR primarily focused on improving accuracy by exploiting expensive additional resources, such as detailed hand/object detection results. While feature fusion approaches is a promising way to improve recognition performance, increasing the computational cost (*e.g.*, increase in model size), especially for inference, have been ignored. Our methods differ in that they improve performance without requiring additional cascaded processes for inference.

3. Proposed Method

3.1. Overview

Our HOCL, which helps a model to learn actions more accurately on the basis of hand-object contact, and OSL, which helps a model to efficiently learn actions capturing object state changes due to actions, are integrated into the action learning in the training phase, as shown in Figure 2. The loss functions for optimizing learnable parameters $\boldsymbol{\theta_1}$ and $\boldsymbol{\theta_2}$ for the video recognition model 1 and 2 (VRM1 and VRM2), considering hand-object contact and object state, are defined as in Equation (1) and (2), respectively.

$$\mathcal{L}_{\boldsymbol{\theta_1}} = \mathcal{L}_{ACTN1} + \mathcal{L}_{HOCL1} + \mathcal{L}_{OSL1} \tag{1}$$

$$\mathcal{L}_{\boldsymbol{\theta_2}} = \mathcal{L}_{ACTN2} + \mathcal{L}_{HOCL2} + \mathcal{L}_{OSL2} \tag{2}$$

In the inference phase, we only use VRM1, which has learnt the relationships among action, hand-object contact, and object state change, to classify actions. In the following sections, we describe each loss function and the learning/inference procedure in detail.

3.2. Action Learning

First, for our main purpose, we define a loss function that minimizes the difference between the ground truth action label and the predicted action probabilities by VRM1. Let M_A be the number of actions defined in a dataset, $X = \{\boldsymbol{x}_i | 1 \leqq i \leqq N\}$ be the N videos in the dataset, and $Y_A = \{y_{A,i} | y_{A,i} \in \{1, 2, ..., M_A\}, 1 \leqq i \leqq N\}$ be the ground truth action labels of X_1. VRM1 minimizes the cross-entropy loss between predicted probabilities and ground truth labels:

$$\mathcal{L}_{ACTN1} = -\sum_{i=1}^{N} \sum_{m_A=1}^{M_A} I_A(m_A, y_{A,i}) \log\left(p_1^{m_A}(\boldsymbol{x}_i)\right) \tag{3}$$

The probability $p_1^{m_A}(\boldsymbol{x}_i)$ for action label m_A of video $\boldsymbol{x}_i$ is given by

$$p_1^{m_A}(\boldsymbol{x}_i) = \frac{exp\left(z_1^{m_A}\right)}{\sum_{m=1}^{M_A} exp\left(z_1^{m}\right)} \tag{4}$$

where z^{m_A} is the logit from $\boldsymbol{x}_i$ to action label m_A. $I_A(m_A, y_{A,i})$ is an indicator function representing the ground truth action label:

$$I_A(m_A, y_{A,i}) = \begin{cases} 1, & m_A = y_{A,i} \\ 0, & m_A \neq y_{A,i} \end{cases} \tag{5}$$

The loss function $\mathcal{L}_{ACTN2}$ for VRM2 can be defined similarly by replacing X with the masked videos $\hat{X} = \{\hat{\boldsymbol{x}}_i | 1 \leqq i \leqq N\}$ generated from X using a hand object detector.

3.3. Hand-Object Contact Learning

To capture hand-object contact, we prepare VRM1, which learns actions from the raw video, and VRM2, which does from the masked video, and they collaboratively learn actions in the manner of deep mutual learning [59]. The masked video contains only the hand-object region information extracted by a hand-object detector; thus, VRM1 captures the contact between hand the object associated with actions by approximating its predicted action probabilities to those of VRM2.

To realize HOCL, we propose a loss function that minimizes the difference between the predicted action probabilities by VRM1 and the ones by VRM2. Specifically, VRM1 minimizes the Kullback-Leibler (KL) divergence between the predicted probabilities output by VRM1 and VRM2:

$$\mathcal{L}_{HOCL1} = \sum_{i=1}^{N} \sum_{m_A=1}^{M_A} p_2^{m_A}(\hat{\boldsymbol{x}}_i) \log\left(\frac{p_2^{m_A}(\hat{\boldsymbol{x}}_i)}{p_1^{m_A}(\boldsymbol{x}_i)}\right) \tag{6}$$

VRM2 likewise minimizes the KL divergence between the predicted action probabilities by VRM2 and VRM1.

3.4. Object State Learning

To capture object state changes during actions, we define a frame-level object state prediction task. The state of an object (*e.g.*, a door) is generally described by adjectives

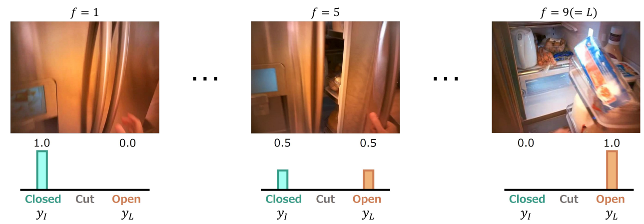

Figure 3. An example of pseudo-object state distributions.

(*e.g.*, broken, closed). Therefore, we manually annotated the states of an object before and after each action, *i.e.*, the initial and final states of an object, with adjectives. For example, for the "Open fridge" action, the initial state is "closed" and the final state is "open" since a closed fridge is opened by the action. If an action does not change the object's state (*e.g.*, hold spoon), the same adjective (*e.g.*, grasped) is assigned to both the initial and final states. Actions that do not affect the state of an object, such as "read recipe" and "wait", are labeled as none. We listed all initial and final states in the supplementary material. After that, we automatically generate pseudo-object state distributions corresponding to each video frame (Figure 3). VRM1 learns the frame-by-frame distributions as well as the video-level actions so that it captures the object state changes associated with actions.

To realize OSL, we propose a loss function that minimizes the difference between the ground truth object state values and the predicted object state values by VRM1. Let M_S be the number of adjectival labels defined in the dataset, and let $Y_S = \{\boldsymbol{y_{S,i}} = (y_{I,i}, y_{L,i}) | y_{I,i}, y_{L,i} \in \{1, 2, ..., M_S\}, 1 \leqq i \leqq N\}$ be pairs of the initial state y_I and last state y_L of an object corresponding to the ground truth action labels Y_A. VRM1 minimizes the KL divergence between the predicted distributions and pseudo-object state distributions:

$$\begin{aligned}\mathcal{L}_{OSL1} =& \frac{\gamma}{|F_i|} \sum_{i=1}^{N} \sum_{f}^{F_i} \sum_{m_S=1}^{M_S} \\ & I_S(m_S, \boldsymbol{y_{S,i}}, f, L_i) \log\left(\frac{I_S(m_S, \boldsymbol{y_{S,i}}, f, L_i)}{p_{1,f}^{m_S}(\boldsymbol{x}_i)}\right)\end{aligned} \tag{7}$$

where L_i is the frame length of video $\boldsymbol{x}_i$, $F_i = \{f_i | f_i \in \{1, ..., L_i\}, 1 \leqq i \leqq N\}$ is the set of frames for which object state prediction is performed, γ is a hyperparameter that determines the effect of object state prediction in the learning process. The value $p_{1,f}^{m_S}(\boldsymbol{x}_i)$ for the adjective label m_S in the f-th frame of the video data $\boldsymbol{x}_i$ is calculated using

$$p_{1,f}^{m_S}(\boldsymbol{x}_i) = \frac{exp\left(z_{1,f}^{m_S}\right)}{\sum_{m=1}^{M_S} exp\left(z_{1,f}^{m}\right)} \tag{8}$$

Algorithm 1 Training Procedure

Input: Train set X, label set Y_A, Y_S, learning rate η
Initialize: Initialize $\boldsymbol{\theta_1}$ and $\boldsymbol{\theta_2}$ to the same conditions

repeat
  Randomly sample $\boldsymbol{x}$ from X and generate $\hat{\boldsymbol{x}}$.
  Compute predictions of VRMs by (4) and (8).
  Compute stochastic gradient and update $\boldsymbol{\theta_1}$:

$$\boldsymbol{\theta_1} \leftarrow \boldsymbol{\theta_1} + \eta \frac{\partial \mathcal{L}_{\boldsymbol{\theta_1}}}{\partial \boldsymbol{\theta_1}}.$$

  Update predictions of VRM1 by (4) and (8)
  Compute stochastic gradient and update $\boldsymbol{\theta_2}$:

$$\boldsymbol{\theta_2} \leftarrow \boldsymbol{\theta_2} + \eta \frac{\partial \mathcal{L}_{\boldsymbol{\theta_2}}}{\partial \boldsymbol{\theta_2}}.$$

  Update predictions of VRM2 by (4) and (8)
until convergence

where $z_{1,f}^{m_S}$ is the logit from the f-th frame in $\boldsymbol{x}_i$ to adjective label m_S. $I_S(m_S, \boldsymbol{y_{S,i}}, f, L_i)$ is an indicator function that represents the pseudo-object state distribution, as shown in Equation (9).

$$I_S(m_S, \boldsymbol{y_{S,i}}, f, L_i) = \begin{cases} 1, & m_S = y_{I,i} \; and \; y_{I,i} = y_{L,i} \\ 1 - \left(\frac{f-1}{L_i-1}\right), & m_S = y_{I,i} \; and \; y_{I,i} \neq y_{L,i} \\ \frac{f-1}{L_i-1}, & m_S = y_{L,i} \; and \; y_{I,i} \neq y_{L,i} \\ 0, & otherwise \end{cases} \tag{9}$$

Figure 3 shows an example of pseudo-object state distributions for "Open fridge." Using equation (9), we can determine the value for "closed" (the initial state y_I) to be 1 and the one for "open" (the last state y_L) to be 0 for the first frame ($f = 1$) of the video, whose length (L) is 9. The values for the other object states (*e.g.*, cut) are permanently 0. As the video continues (the frame number f increases), the values for "closed" and "open" are linearly switched. VRM2 likewise minimizes the KL divergence between the predicted distributions and pseudo-object state distributions.

3.5. Training and Inference Procedure

In the training phase, VRM1 and VRM2 are optimized collaboratively in the manner of mini-batch learning. We prepare a mini-batch of videos for VRM1 and then generate masked videos by using a hand-object detector for VRM2. VRM1 and VRM2 then compute their predictions; the $\boldsymbol{\theta_1}$ and $\boldsymbol{\theta_2}$ are updated in turn at each iteration. This procedure is iteratively performed until convergence. Algorithm 1 summarizes the training procedure.

In the inference phase, we only calculate Equation (4) using VRM1, which has learnt the relationships among action, hand-object contact, and object state. In other words, models trained with HOCL and OSL can predict action labels from only raw video, just as models trained with pairs of a video and action label in the end-to-end manner.

4. Evaluation

4.1. Datasets and Settings

Datasets and Evaluation Metrics. We evaluated the performance of our proposed methods on two domain datasets: EGTEA [31] and MECCANO [43]. We also evaluated it on a large-scale dataset, EPIC-KITCHENS 100 (EPIC-100) [9]. **EGTEA** contains 10,321 segments of 106 actions in the kitchen environment (*e.g.*, open fridge). It defines three 8:2 data splits for the train and test sets. In our experiments, we randomly selected 1,000 videos from the action segments in the original train set and defined three 7:1:2 data splits for the train, validation, and test sets. We manually annotated the initial and final states for the 106 actions. The total number of adjectives was 20. We calculated the overall accuracy and mean class accuracy averaged across all three splits. **MECCANO** contains 8,839 action segments of 61 types of industrial-like domain actions (*e.g.*, plug_rod). We used the train, validation, and test sets as defined. We manually annotated the initial and final states for the 61 actions. The total number of adjectives was 13. We calculated the top-{1,5} accuracy and macro-averaged precision, recall, and f1 score. In a resulting table, they are denoted as Acc@1, Acc@5, P, R, and F1, respectively. **EPIC-100** contains action segments labeled with a combination of 97 verbs (*e.g.*, open) and 300 nouns (*e.g.*, fridge). In our experiments, we used segments P01 to P27 in the original train set as a train set (55,191 segments) and the remaining segments as a validation set (12,026 segments). We used the segments in the original validation set as a test set (9,668 segments). We manually annotated the initial and final states for all verb-noun pairs in the train and validation sets. The total number of adjectives was 94. Since each action is a combination of a verb and noun, we predicted both labels using two heads per video recognition model and set the top-scoring verb and noun pair as the action label. We calculated the top-1 verb, noun, and action accuracy for "overall," "unseen participants," and "tail classes" settings.

Video Recognition Models. We conducted our experiments with two video recognition models pre-trained on the Kinetics dataset [6]: SlowFast (SlowFast 8×8 ResNet-50, $\alpha = 4, \beta = 1/8$) [20], a CNN-based state-of-the-art model, and Video Swin Transformer (Swin-B) [37], a state-of-the-art transformer-based model.

Hand Object Detector. For hand-object detection, we used the Faster-RCNN [45] trained on 100DOH + Egocen-

Table 1. Results for each model w/ and w/o HOCL and OSL on EGTEA dataset [31]. **Bolded** scores indicate best ones in each model.

Model	HOCL	OSL	Overall acc. Split1	Overall acc. Split2	Overall acc. Split3	Overall acc. Average	Mean class acc. Split1	Mean class acc. Split2	Mean class acc. Split3	Mean class acc. Average
SlowFast	✗	✗	64.29	63.20	64.57	64.02	56.48	55.07	55.89	55.81
	✓	✗	**69.78**	65.68	63.98	66.48	**61.43**	56.35	55.20	57.66
	✗	✓	67.66	**65.73**	64.03	65.81	59.43	55.85	55.65	56.97
	✓	✓	69.09	65.68	**65.81**	**66.86**	59.36	**57.39**	**57.66**	**58.14**
Swin-B	✗	✗	65.83	64.44	62.00	64.09	59.62	55.58	56.21	57.14
	✓	✗	67.16	64.99	62.15	64.77	60.96	55.55	55.77	57.43
	✗	✓	**68.25**	64.94	**64.23**	65.81	**62.19**	56.75	**57.88**	58.94
	✓	✓	67.80	**66.47**	63.68	**65.98**	61.77	**58.64**	57.41	**59.27**

Table 2. Results for each model w/ and w/o HOCL and OSL on MECCANO dataset [43]. **Bolded** scores indicate best ones in each model.

Model	HOCL	OSL	Acc@1	Acc@5	P	R	F1
SlowFast	✗	✗	38.08	70.63	0.178	0.151	0.141
	✓	✗	**42.05**	70.99	0.169	0.156	0.150
	✗	✓	41.37	**74.25**	0.146	0.139	0.133
	✓	✓	40.70	73.33	**0.198**	**0.160**	**0.151**
Swin-B	✗	✗	44.03	76.76	0.206	0.163	0.166
	✓	✗	42.01	75.77	**0.239**	0.154	0.161
	✗	✓	44.00	75.59	0.231	**0.184**	**0.185**
	✓	✓	**44.81**	**77.01**	0.201	0.163	0.167

tric data [48] [8] [31] [49]. For EGTEA and MECCANO, we extracted hand-object bounding box (bbox) coordinates from the videos. For EPIC-100, we used pre-extracted bbox coordinates published by the authors of EPIC-100.

4.2. Implementation Details

We used the PyTorch [40] and PyTorchVideo [15] for implementation and the default settings for all parameters except those explicitly mentioned. To train the SlowFast model, we used stochastic gradient descent [4] with momentum 0.5, learning rate 5e-3, and weight decay 1e-4 to optimize the parameters for 60 epochs on EGTEA and 40 epochs on MECCANO. The mini-batch size was set to 16. For SlowFast on EPIC-100, we used the settings proposed by the EPIC-100 authors [9] except for the mini-batch size, which we set to 16. We used a temporal stride of 2, or its horizontal flip, with the length of the shorter side randomly sampled from 256 to 320 pixels, to randomly clip 224×224 pixels from 64 successive frames in each video. For inference, we scaled the shorter spatial side to 256 pixels and took 256 × 256 pixels from a 32-frame clip uniformly sampled from the entire video. To train the Swin-B model, we used the AdamW [12] with learning rate 3e-5 to optimize the parameters for 50 epochs on EGTEA and 40 epochs on MECCANO. For Swin-B on EPIC-100, we used the settings for the Kinetics dataset proposed by the Swin-B authors [37]. The mini-batch size was set to 64. We randomly cropped 224×224 pixels from 64 successive frames in each video using a temporal stride of 2 or its horizontal flip. For inference, we scaled the shorter spatial side to 224 pixels and took 224 × 224 pixels from a 32-frame clip uniformly sampled from the entire video. For all model variants, the dimensions of the video feature and masked video feature were set to 1024. Parameter γ, which adjusts the effect of OSL, was set to 0.5. Object state prediction was conducted using the slow pathway frames for SlowFast and using eight uniformly sampled frames for Swin-B, meaning that $|F_i|$ equaled 8 in the loss function.

4.3. Main Results

First, we compare the overall performance on the three datasets. The results on the EGTEA are shown in Table 1. For both the SlowFast and Swin-B, training with HOCL and OSL improved the overall accuracy across all data splits and the average scores. In particular, training with both HOCL and OSL improved the overall accuracy of SlowFast and Swin-B by 2.84% and 1.89%, respectively. Table 2 presents the results on the MECCANO, which contains actions in an industrial-like domain. As with EGTEA, performance gains on top-$\{1,5\}$ accuracy were observed for both SlowFast and Swin-B. Table 3 presents the results for each model on the EPIC-100. The overall accuracy of SlowFast improved by 1.49% and the one of Swin-B is comparable to the baseline. These results demonstrate that the use of HOCL and OSL is mostly effective in different domains and with a range of dataset scales.

We also verify the performance in terms of a metric that evaluates data with small training samples and unseen data. Since mean class accuracy on EGTEA and F1-score on MECCANO treat performance for each class equally, they are influenced by performance for classes with fewer samples compared to overall accuracy. In the “tail classes” setting on EPIC-100, each model is evaluated only on the minor classes, which comprised 20% of the training instances. Note that the majority of verb and noun labels in the EPIC-100, specifically 86 out of 97 verb labels and 228 out of

Table 3. Results for each model w/ and w/o HOCL and OSL on EPIC-100 dataset [9]. **Bolded** scores indicate best ones in each model.

Model	HOCL	OSL	Overall			Unseen participants			Tail classes		
			Verb	Noun	Action	Verb	Noun	Action	Verb	Noun	Action
SlowFast	✗	✗	57.28	44.01	32.14	45.26	34.65	23.00	37.26	**25.31**	**12.40**
	✓	✗	**59.70**	**45.16**	**33.63**	**50.89**	**36.15**	**26.29**	**38.13**	25.28	12.21
	✗	✓	56.35	38.50	27.96	48.36	32.86	22.72	34.43	19.23	8.18
	✓	✓	56.81	42.16	31.53	46.67	33.80	24.51	35.97	21.22	10.21
Swin-B	✗	✗	**54.81**	52.58	**34.39**	46.85	**44.69**	26.95	39.61	33.01	15.97
	✓	✗	52.13	**52.63**	32.72	44.32	43.76	24.88	37.36	**33.72**	15.43
	✗	✓	54.33	51.15	33.41	**47.51**	43.47	**27.32**	**40.48**	32.40	15.91
	✓	✓	53.32	52.17	33.52	46.85	44.60	26.85	39.94	33.70	**17.10**

Table 4. Results for each model w/ and w/o HOCL and OSL on EGTEA dataset [31] using models pretrained on EPIC-100 dataset [9]. **Bolded** scores indicate best ones in each model.

Model	HOCL	OSL	Overall acc.				Mean class acc.			
			Split1	Split2	Split3	Average	Split1	Split2	Split3	Average
SlowFast	✗	✗	70.28	**67.56**	68.73	68.86	61.90	57.87	59.36	59.71
	✓	✗	**71.36**	67.46	**69.12**	**69.31**	**63.70**	**58.15**	**61.09**	**60.98**
	✗	✓	69.24	64.23	65.36	66.28	61.73	55.80	57.57	58.37
	✓	✓	69.83	67.36	67.74	68.31	61.33	57.32	57.14	58.60
Swin-B	✗	✗	**71.76**	68.99	66.01	68.92	**66.05**	61.12	60.05	62.41
	✓	✗	69.04	69.14	65.66	67.95	62.87	61.13	58.12	60.71
	✗	✓	71.32	**70.67**	**66.95**	**69.65**	65.89	**63.96**	59.69	**63.18**
	✓	✓	71.46	69.34	**66.95**	69.25	65.70	61.53	**60.83**	62.69

300 noun labels, belong to the tail classes. In the "unseen participants" setting on the EPIC-100, each model is evaluated on participant data not presented in the train set; in other words, it predicts actions in unseen scenes. The tables show that training with our proposed methods improved the performance for minor classes and for unfamiliar scenes. In particular, training with both HOCL and OSL improved the mean class accuracy of the SlowFast and Swin-B by 2.33% and 2.13% on the EGTEA dataset, respectively. On the MECCANO, the use of HOCL and/or OSL increased F1-scores to 1.0pt and 1.9pt for SlowFast and Swin-B. In the "tail classes" and "unseen participants" settings on EPIC-100, SlowFast and Swin-B with our proposed method are equal or better performance. These results indicate that HOCL and OSL make EAR models robust for minor classes and unfamiliar scenes.

The results for all datasets shows that HOCL tends to contribute more for SlowFast whereas OSL contributes more for Swin-B. We attribute this to the architectural differences between SlowFast and Swin-B. SlowFast repeats convolutional operations internally and is able to learn local spatio-temporal features. It thus tends to have higher affinity with HOCL, which is a constraint that focuses on contact between hands and objects during actions, *i.e.*, local spatio-temporal information. On the other hand, Swin-B captures the relationship between patches and between windows and thus can learn global spatio-temporal features. Therefore, it tends to have higher affinity with OSL, which is a constraint that focuses on object state changes due to actions, *i.e.*, global spatio-temporal information. Further analysis of these affinities in line with previous studies is required [2, 42].

4.4. Results on First-Person Video Pretraining

We also evaluated the effectiveness of our methods for models pretrained on a large dataset from the same viewpoint and domain. We used models pretrained on the Kinetics dataset for all the experiments discussed above. However, the type of dataset used in the pretraining phase greatly affects the performance of recognition models. EGTEA and EPIC-100 contain various actions in the kitchen environment; therefore, we pretrained the SlowFast and Swin-B models on all the train and validation data in EPIC-100 and then compared the performance of each model on EGTEA. To pretrain SlowFast and Swin-B, we followed the settings proposed by the EPIC-100 authors and the settings for Kinetics proposed by the Swin-B authors. The results with pretraining on EPIC-100 are shown in Table 4. They show

Table 5. Performance comparison with other methods on EGTEA dataset. We report the scores on split1 in line with them.

Method	Modality	Overall	Mean class
Yifei *et al*. [26]	RGB+flow+gaze	-	62.6
Min and Corso [39]	RGB+flow+gaze	69.5	**62.8**
SlowFast (ours, best)	RGB	**69.7**	61.4
Swin-B (ours, best)	RGB	68.2	62.1

Table 6. Performance comparison with an other method on EPIC-100 dataset. We report the overall scores in line with it.

Method	Modality	Action	Verb	Noun
Wang *et al*. [58]	RGB+flow+obj	28.8	**60.4**	37.4
SlowFast (ours, best)	RGB	**33.6**	59.7	45.1
Swin-B (ours, best)	RGB	33.5	53.3	**52.1**

that our proposed methods are effective even when the models are pretrained on a large dataset in the same domain. This indicates that HOCL and OSL can help EAR models to focus on the appropriate spatio-temporal information and thereby achieve more accurate recognition even when abundant training resources are available.

4.5. Comparison with other methods utilizing human body motions and active objects

Models trained with our proposed methods predict actions with only RGB of a video for inference. Therefore, it is appropriate to consider the latest end-to-end EAR models, such as SlowFast and Swin-B, as the baseline models for comparison. On the other hand, we also compare with existing methods utilizing not only RGB but also human body motions and active objects. Table 5 and 6 show performance comparison with them. The results show that SlowFast and Swin-B with our proposed methods are comparable or better performance even though (1) the existing methods utilize the information of human body motions and active objects for inference and (2) they are trained on more training samples. Note that the experimental setting for compared methods is slightly different. Specifically, we extracted a validation set from the train set as mentioned in section 4.1; thus, our models trained fewer training samples.

4.6. Qualitative Analysis

We qualitatively evaluated the two proposed methods to better understand their behaviors by visualizing where is focused upon to recognize actions by models trained on each method. Example visualizations of the ROI for the slow pathway of SlowFast are shown in Figure 1. We used the GradCAM [46] to visualize the ROI with the SlowFast model for each combination of HOCL and OSL and for neither one. SlowFast with only action learning incorrectly recognized the action as "Close cabinet," whereas SlowFast trained on the two proposed methods correctly recognized "Open cabinet." The model trained with only action learning (1st row) reacted strongly to the closed cabinet in the second frame and thus recognized incorrectly. This shows that the SlowFast model cannot adequately capture information relevant to the action in the training phase. On the other hand, when SlowFast was trained on both HOCL and OSL, it recognized the action by strongly responding to the region where the left hand contacted the cabinet and where the cabinet state changed from open to closed during the action. This analysis suggests that our proposed methods enable EAR models to learn actions considering hand-object interactions. We have confirmed this trend in multiple cases across domains; however, due to page limits, other example visualizations are presented in the supplementary material.

5. Limitations

Our proposed methods require annotation of an additional adjective label for the datasets. Our proposed methods require annotation of an additional adjective label for the datasets. Annotation cost for OSL is proportional to the number of action labels defined in a dataset, not the number of videos or frames; thus, it is unlikely to be a barrier to incorporating this idea into other tasks/datasets. However, additional manual annotation is unavoidable. It is necessary to design a method that works in an unsupervised or self-supervised manner.

Neither an untrained model nor one trained on the proposed methods can prioritize the actions to be recognized on the current datasets. We often perform multiple actions simultaneously. For example, we might lift a loaf of bread with our left hand and simultaneously grasp a knife with our right hand to slice it. In this situation, the video shows both "Take bread" and "Take eating_utensil"; therefore, there are two ground truth actions. However, recognition models cannot determine which action is salient. We discuss this point in detail with a visualization example in the supplementary material.

6. Conclusion

Our proposed methods, HOCL and OSL, help EAR models to classify actions more accurately by focusing on hand-object contact and object state change. Experiments demonstrated that the two proposed methods improved recognition performance in different domains on datasets of various scales. This improved recognition performance, especially for classes with a few instances in the train set and for unseen data. We also showed that our proposed methods incorporated the relationships among action, hand-object contact, and object state change into EAR models through visualization.

References

[1] Anurag Arnab, Mostafa Dehghani, Georg Heigold, Chen Sun, Mario Lučić, and Cordelia Schmid. ViViT: A Video Vision Transformer. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 6836–6846, 2021. 2

[2] Nicholas Baker, Hongjing Lu, Gennady Erlikhman, and Philip J Kellman. Deep Convolutional Networks Do not Classify Based on Global Object Shape. *PLoS computational biology*, 14(12):e1006613, 2018. 7

[3] Gedas Bertasius, Heng Wang, and Lorenzo Torresani. Is Space-Time Attention All You Need for Video Understanding? In *Proceedings of the 38th International Conference on Machine Learning (ICML)*, pages 813–824, 2021. 2

[4] Léon Bottou. Stochastic Gradient Learning in Neural Networks. In *Proceedings of Neuro-Nîmes 91*, 1991. 6

[5] Fabian Caba Heilbron, Victor Escorcia, Bernard Ghanem, and Juan Carlos Niebles. ActivityNet: A Large-Scale Video Benchmark for Human Activity Understanding. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 961–970, 2015. 2

[6] João Carreira and Andrew Zisserman. Quo Vadis, Action Recognition? A New Model and the Kinetics Dataset. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4724–4733, 2017. 2, 5

[7] Sara Colombo, Yihyun Lim, and Federico Casalegno. Deep Vision Shield: Assessing the Use of HMD and Wearable Sensors in a Smart Safety Device. In *Proceedings of the 12th ACM International Conference on PErvasive Technologies Related to Assistive Environments*, page 402–410, 2019. 2

[8] Dima Damen, Hazel Doughty, Giovanni Maria Farinella, Sanja Fidler, Antonino Furnari, Evangelos Kazakos, Davide Moltisanti, Jonathan Munro, Toby Perrett, Will Price, and Michael Wray. Scaling Egocentric Vision: The EPIC-KITCHENS Dataset. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 720–736, 2018. 2, 6

[9] Dima Damen, Hazel Doughty, Giovanni Maria Farinella, Antonino Furnari, Jian Ma, Evangelos Kazakos, Davide Moltisanti, Jonathan Munro, Toby Perrett, Will Price, and Michael Wray. Rescaling Egocentric Vision: Collection, Pipeline and Challenges for EPIC-KITCHENS-100. *International Journal of Computer Vision (IJCV)*, 130:33–55, 2022. 2, 5, 6, 7

[10] Eadom Dessalene, Chinmaya Devaraj, Michael Maynord, Cornelia Fermüller, and Yiannis Aloimonos. Forecasting action through contact representations from first person video. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 45(6):6703–6714, 2023. 3

[11] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, pages 4171–4186, 2019. 2

[12] Jimmy Ba Diederik P. Kingma. Adam: A Method for Stochastic Optimization. *CoRR*, abs/1412.6980, 2014. 6

[13] Jeffrey Donahue, Lisa Anne Hendricks, Sergio Guadarrama, Marcus Rohrbach, Subhashini Venugopalan, Kate Saenko, and Trevor Darrell. Long-Term Recurrent Convolutional Networks for Visual Recognition and Description. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2625–2634, June 2015. 2

[14] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. In *International Conference on Learning Representations (ICLR)*, 2021. 2

[15] Haoqi Fan, Tullie Murrell, Heng Wang, Kalyan Vasudev Alwala, Yanghao Li, Yilei Li, Bo Xiong, Nikhila Ravi, Meng Li, Haichuan Yang, Jitendra Malik, Ross Girshick, Matt Feiszli, Aaron Adcock, Wan-Yen Lo, and Christoph Feichtenhofer. PyTorchVideo: A deep learning library for video understanding. In *Proceedings of the 29th ACM International Conference on Multimedia*, 2021. 6

[16] Haoqi Fan, Bo Xiong, Karttikeya Mangalam, Yanghao Li, Zhicheng Yan, Jitendra Malik, and Christoph Feichtenhofer. Multiscale Vision Transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 6824–6835, 2021. 2

[17] Linxi Fan, Shyamal Buch, Guanzhi Wang, Ryan Cao, Yuke Zhu, Juan Carlos Niebles, and Li Fei-Fei. RubiksNet: Learnable 3D-Shift for Efficient Video Action Recognition. In *Proceedings of the European Conference on Computer Vision (ECCV)*, page 505–521, 2020. 2

[18] Alireza Fathi and James M. Rehg. Modeling Actions through State Changes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2579–2586, 2013. 3

[19] Christoph Feichtenhofer. X3d: Expanding Architectures for Efficient Video Recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 203–213, 2020. 2

[20] Christoph Feichtenhofer, Haoqi Fan, Jitendra Malik, and Kaiming He. SlowFast Networks for Video Recognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 6202–6211, 2019. 2, 5

[21] James J. Gibson. *The Ecological Approach to Visual Perception*. Houghton Mifflin, 1979. 2

[22] Raghav Goyal, Samira Ebrahimi Kahou, Vincent Michalski, Joanna Materzynska, Susanne Westphal, Heuna Kim, Valentin Haenel, Ingo Fruend, Peter Yianilos, Moritz Mueller-Freitag, Florian Hoppe, Christian Thurau, Ingo Bax, and Roland Memisevic. The "something something" video database for learning and evaluating visual common sense. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 5842–5850, 2017. 2

[23] Kristen Grauman, Andrew Westbury, Eugene Byrne, Zachary Chavis, Antonino Furnari, Rohit Girdhar, Jackson Hamburger, Hao Jiang, Miao Liu, Xingyu Liu, Miguel Martin, Tushar Nagarajan, Ilija Radosavovic, Santhosh Kumar

Ramakrishnan, Fiona Ryan, Jayant Sharma, Michael Wray, Mengmeng Xu, Eric Zhongcong Xu, Chen Zhao, Siddhant Bansal, Dhruv Batra, Vincent Cartillier, Sean Crane, Tien Do, Morrie Doulaty, Akshay Erapalli, Christoph Feichtenhofer, Adriano Fragomeni, Qichen Fu, Abrham Gebreselasie, Cristina González, James Hillis, Xuhua Huang, Yifei Huang, Wenqi Jia, Weslie Khoo, Jáchym Kolář, Satwik Kottur, Anurag Kumar, Federico Landini, Chao Li, Yanghao Li, Zhenqiang Li, Karttikeya Mangalam, Raghava Modhugu, Jonathan Munro, Tullie Murrell, Takumi Nishiyasu, Will Price, Paola Ruiz, Merey Ramazanova, Leda Sari, Kiran Somasundaram, Audrey Southerland, Yusuke Sugano, Ruijie Tao, Minh Vo, Yuchen Wang, Xindi Wu, Takuma Yagi, Ziwei Zhao, Yunyi Zhu, Pablo Arbeláez, David Crandall, Dima Damen, Giovanni Maria Farinella, Christian Fuegen, Bernard Ghanem, Vamsi Krishna Ithapu, C. V. Jawahar, Hanbyul Joo, Kris Kitani, Haizhou Li, Richard Newcombe, Aude Oliva, Hyun Soo Park, James M. Rehg, Yoichi Sato, Jianbo Shi, Mike Zheng Shou, Antonio Torralba, Lorenzo Torresani, Mingfei Yan, and Jitendra Malik. Ego4D: Around the World in 3,000 Hours of Egocentric Video. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 18995–19012, 2022. 2

[24] Chunhui Gu, Chen Sun, David A. Ross, Carl Vondrick, Caroline Pantofaru, Yeqing Li, Sudheendra Vijayanarasimhan, George Toderici, Susanna Ricco, Rahul Sukthankar, Cordelia Schmid, and Jitendra Malik. AVA: A Video Dataset of Spatio-Temporally Localized Atomic Visual Actions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 2

[25] Yun He, Soma Shirakabe, Yutaka Satoh, and Hirokatsu Kataoka. Human Action Recognition Without Human. In *Proceedings of the European Conference on Computer Vision Workshop (ECCVW)*, pages 11–17, 2016. 3

[26] Yifei Huang, Minjie Cai, Zhenqiang Li, Feng Lu, and Yoichi Sato. Mutual Context Network for Jointly Estimating Egocentric Gaze and Action. *IEEE Transactions on Image Processing (TIP)*, 29:7795–7806, 2020. 3, 8

[27] Georgios Kapidis, Ronald Poppe, Elsbeth van Dam, Lucas Noldus, and Remco Veltkamp. Multitask Learning to Improve Egocentric Action Recognition. In *2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW)*, pages 4396–4405, 2019. 3

[28] Hildegard Kuehne, Hueihan Jhuang, Estibaliz Garrote, Tomaso Poggio, and Thomas Serre. HMDB: A Large Video Database for Human Motion Recognition. In *Proceedings of International Conference on Computer Vision (ICCV)*, pages 2556–2563, 2011. 2

[29] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998. 2

[30] Yingwei Li, Yi Li, and Nuno Vasconcelos. RESOUND: Towards Action Recognition without Representation Bias. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 513–528, 2018. 2

[31] Yin Li, Miao Liu, and James 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 (ECCV)*, 2018. 2, 5, 6, 7

[32] Yin Li, Zhefan Ye, and James M. Rehg. Delving Into Egocentric Actions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 287–295, 2015. 3

[33] Zhenyang Li, Kirill Gavrilyuk, Efstratios Gavves, Mihir Jain, and Cees G.M. Snoek. VideoLSTM convolves, attends and flows for action recognition. *Computer Vision and Image Understanding*, 166:41–50, 2018. 2

[34] Ji Lin, Chuang Gan, and Song Han. TSM: Temporal Shift Module for Efficient Video Understanding. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 7083–7093, 2019. 2

[35] Yang Liu, Ping Wei, and Song-Chun Zhu. Jointly Recognizing Object Fluents and Tasks in Egocentric Videos. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 2943–2951, 2017. 3

[36] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin Transformer: Hierarchical Vision Transformer Using Shifted Windows. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 10012–10022, 2021. 2

[37] Ze Liu, Jia Ning, Yue Cao, Yixuan Wei, Zheng Zhang, Stephen Lin, and Han Hu. Video Swin Transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3202–3211, 2022. 2, 5, 6

[38] Ralph Maddison, Susie Cartledge, Michelle Rogerson, Nicole Sylvia Goedhart, Tarveen Ragbir Singh, Christopher Neil, Dinh Phung, and Kylie Ball. Usefulness of Wearable Cameras as a Tool to Enhance Chronic Disease Self-Management: Scoping Review. *JMIR mHealth and uHealth*, 7(1):e10371, 2019. 2

[39] Kyle Min and Jason J. Corso. Integrating Human Gaze Into Attention for Egocentric Activity Recognition. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 1069–1078, 2021. 3, 8

[40] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems (NeurIPS)*, pages 8024–8035, 2019. 6

[41] Zhaofan Qiu, Ting Yao, and Tao Mei. Learning Spatio-Temporal Representation with Pseudo-3D Residual Networks. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 5533–5541, 2017. 2

[42] Maithra Raghu, Thomas Unterthiner, Simon Kornblith, Chiyuan Zhang, and Alexey Dosovitskiy. Do Vision Transformers See Like Convolutional Neural Networks? In *Proceedings of the Advances in Neural Information Processing Systems (NeurIPS)*, pages 12116–12128, 2021. 7

[43] Francesco Ragusa, Antonino Furnari, Salvatore Livatino, and Giovanni Maria Farinella. The MECCANO Dataset: Understanding Human-Object Interactions From Egocentric Videos in an Industrial-Like Domain. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 1569–1578, 2021. 2, 5, 6

[44] Nishant Rai, Haofeng Chen, Jingwei Ji, Rishi Desai, Kazuki Kozuka, Shun Ishizaka, Ehsan Adeli, and Juan Carlos Niebles. Home Action Genome: Contrastive Compositional Action Understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11184–11193, 2021. 2

[45] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. In *Proceedings of the 28th International Conference on Neural Information Processing Systems (NeurIPS)*, 2015. 5

[46] Ramprasaath R. Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-CAM: Visual Explanations from Deep Networks via Gradient-Based Localization. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 618–626, 2017. 1, 8

[47] Amir Shahroudy, Jun Liu, Tian-Tsong Ng, and Gang Wang. NTU RGB+D: A Large Scale Dataset for 3D Human Activity Analysis. In *Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1010–1019, 2016. 2

[48] Dandan Shan, Jiaqi Geng, Michelle Shu, and David F. Fouhey. Understanding Human Hands in Contact at Internet Scale. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9869–9878, 2020. 6

[49] Gunnar A. Sigurdsson, Abhinav Gupta, Cordelia Schmid, Ali Farhadi, and Karteek Alahari. Actor and Observer: Joint Modeling of First and Third-Person Videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7396–7404, 2018. 2, 6

[50] Gunnar A. Sigurdsson, Gül Varol, Xiaolong Wang, Ali Farhadi, Ivan Laptev, and Abhinav Gupta. Hollywood in Homes: Crowdsourcing Data Collection for Activity Understanding. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 510–526, 2016. 2

[51] Karen Simonyan and Andrew Zisserman. Two-Stream Convolutional Networks for Action Recognition in Videos. In *Proceedings of the 27th International Conference on Neural Information Processing Systems (NeurIPS)*, page 568–576, 2014. 2

[52] Bugra Tekin, Federica Bogo, and Marc Pollefeys. H+O: Unified Egocentric Recognition of 3D Hand-Object Poses and Interactions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4511–4520, 2019. 3

[53] Du Tran, Lubomir Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. Learning Spatiotemporal Features with 3d Convolutional Networks. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 4489–4497, 2015. 2

[54] Du Tran, Heng Wang, Lorenzo Torresani, Jamie Ray, Yann LeCun, and Manohar Paluri. A Closer Look at Spatiotemporal Convolutions for Action Recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6450–6459, 2018. 2

[55] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is All you Need. In *Proceedings of the 30th International Conference on Neural Information Processing Systems (NeurIPS)*, 2017. 2

[56] Limin Wang, Yuanjun Xiong, Zhe Wang, Yu Qiao, Dahua Lin, Xiaoou Tang, and Luc Val Gool. Temporal Segment Networks: Towards Good Practices for Deep Action Recognition. In *Proceedings of the 14th European Conference on Computer Vision (ECCV)*, pages 20–36, 2016. 2

[57] Rui Wang, Dongdong Chen, Zuxuan Wu, Yinpeng Chen, Xiyang Dai, Mengchen Liu, Yu-Gang Jiang, Luowei Zhou, and Lu Yuan. BEVT: BERT Pretraining of Video Transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 14733–14743, 2022. 2

[58] Xiaohan Wang, Linchao Zhu, Yu Wu, and Yi Yang. Symbiotic Attention for Egocentric Action Recognition with Object-centric Alignment. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2020. 3, 8

[59] Ying Zhang, Tao Xiang, Timothy M. Hospedales, and Huchuan Lu. Deep Mutual Learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4320–4328, 2018. 4

[60] Yang Zhou, Bingbing Ni, Richang Hong, Xiaokang Yang, and Qi Tian. Cascaded Interactional Targeting Network for Egocentric Video Analysis. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1904–1913, 2016. 3

[61] Linchao Zhu, Du Tran, Laura Sevilla-Lara, Yi Yang, Matt Feiszli, and Heng Wang. FASTER Recurrent Networks for Efficient Video Classification. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 13098–13105, 2020. 2