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

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 EAR models to better understand actions by capturing hand-object interaction 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] have 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.
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 $\theta_1$ and $\theta_2$ 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.

$$L_{\theta_1} = L_{\text{ACTN}_1} + L_{\text{HOCL}_1} + L_{\text{OSL}_1}$$
$$L_{\theta_2} = L_{\text{ACTN}_2} + L_{\text{HOCL}_2} + L_{\text{OSL}_2}$$

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, $X = \{x_i|1 \leq i \leq N\}$ be the $N$ videos in the dataset,
and $Y_A = \{y_{A,i} | y_{A,i} \in \{1, 2, ..., M_A\}, 1 \leq i \leq N\}$ be the ground truth action labels of $X_1$. VRM1 minimizes the cross-entropy loss between predicted probabilities and ground truth labels:

$$\mathcal{L}_{ACTN1} = - \sum_{i=1}^{N} \sum_{m_A=1}^{M_A} I_A(m_A, y_{A,i}) \log(p_1^{m_A}(x_i))$$

(3)

The probability $p_1^{m_A}(x_i)$ for action label $m_A$ of video $x_i$ is given by

$$p_1^{m_A}(x_i) = \frac{\exp(z_{1}^{m_A})}{\sum_{m=1}^{M_A} \exp(z_{1}^{m})}$$

(4)

where $z_{1}^{m_A}$ is the logit from $x_i$ to action label $m_A$. $I_A(m_A, y_{A,i})$ is an indicator function representing the ground truth action label:

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

(5)

The loss function $\mathcal{L}_{ACTN2}$ for VRM2 can be defined similarly by replacing $X$ with the masked videos $\hat{X} = \{\hat{x}_i|1 \leq i \leq N\}$ 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 the hand 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:

$$\mathcal{L}_{HOCL1} = - \sum_{i=1}^{N} \sum_{m_A=1}^{M_A} I_A(m_A, y_{A,i}) \log\left(\frac{p_2^{m_A}(\hat{x}_i)}{p_1^{m_A}(x_i)}\right)$$

(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 (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 $Y_S = \{y_{S,i}|y_{S,i} = (y_{1,i}, y_{L,i}) | y_{1,i}, y_{L,i} \in \{1, 2, ..., M_S\}, 1 \leq i \leq N\}$ be pairs of the initial state $y_1$ 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:

$$\mathcal{L}_{OSL1} = \frac{\gamma}{|F_i|} \sum_{i=1}^{N} \sum_{f=-1}^{F_i} \sum_{m_S=1}^{M_S} I_S(m_S, y_{S,i}, f, L_i) \log\left(\frac{I_S(m_S, y_{S,i}, f, L_i)}{p_1^{m_S}(x_i)}\right)$$

(7)

where $L_i$ is the frame length of video $x_i$, $F_i = \{f_i|f_i \in \{1, ..., L_i\}, 1 \leq i \leq N\}$ is the set of frames for which object state prediction is performed, $\gamma$ is a hyperparameter that determines the effect of object state prediction in the learning process. The value $p_1^{m_S}(x_i)$ for the adjective label $m_S$ in the $f$-th frame of the video data $x_i$ is calculated using

$$p_1^{m_S}(x_i) = \frac{\exp(z_{1}^{m_S})}{\sum_{m=1}^{M_S} \exp(z_{1}^{m})}$$

(8)
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_rot) 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 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, α = 4, β = 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-
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 Swin-B, we used AdamW [12] with learning rate 3e-5 to optimize the parameters. For both the SlowFast and Swin-B, training with HOCL and OSL improved the overall accuracy of SlowFast and Swin-B by 2.84% and 1.89%, respectively. Table 2 presents the results for each model on the MECCANO, which contains actions in an industrial-like domain. As with EGTEA, performance gains were observed for both SlowFast and 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 6546.
<table>
<thead>
<tr>
<th>Model</th>
<th>HOCL</th>
<th>OSL</th>
<th>Overall Verb</th>
<th>Overall Noun</th>
<th>Overall Action</th>
<th>Unseen participants Verb</th>
<th>Unseen participants Noun</th>
<th>Unseen participants Action</th>
<th>Tail classes Verb</th>
<th>Tail classes Noun</th>
<th>Tail classes Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>SlowFast</td>
<td>✗</td>
<td>✗</td>
<td>57.28</td>
<td>44.01</td>
<td>32.14</td>
<td>45.26</td>
<td>34.65</td>
<td>23.00</td>
<td>37.26</td>
<td>25.31</td>
<td>12.40</td>
</tr>
<tr>
<td></td>
<td>✓</td>
<td>✗</td>
<td>59.70</td>
<td>45.16</td>
<td>33.63</td>
<td>50.89</td>
<td>36.15</td>
<td>26.29</td>
<td>38.13</td>
<td>25.28</td>
<td>12.21</td>
</tr>
<tr>
<td></td>
<td>✗</td>
<td>✓</td>
<td>56.35</td>
<td>38.50</td>
<td>27.96</td>
<td>48.36</td>
<td>32.86</td>
<td>22.72</td>
<td>34.43</td>
<td>19.23</td>
<td>8.18</td>
</tr>
<tr>
<td></td>
<td>✓</td>
<td>✓</td>
<td>56.81</td>
<td>42.16</td>
<td>31.53</td>
<td>46.67</td>
<td>33.80</td>
<td>24.51</td>
<td>35.97</td>
<td>21.22</td>
<td>10.21</td>
</tr>
<tr>
<td>Swin-B</td>
<td>✗</td>
<td>✗</td>
<td>54.81</td>
<td>52.58</td>
<td>34.39</td>
<td>46.85</td>
<td>44.69</td>
<td>26.95</td>
<td>39.61</td>
<td>33.01</td>
<td>15.97</td>
</tr>
<tr>
<td></td>
<td>✓</td>
<td>✗</td>
<td>52.13</td>
<td>52.63</td>
<td>32.72</td>
<td>44.32</td>
<td>43.76</td>
<td>24.88</td>
<td>37.36</td>
<td>33.72</td>
<td>15.43</td>
</tr>
<tr>
<td></td>
<td>✗</td>
<td>✓</td>
<td>54.33</td>
<td>51.15</td>
<td>33.41</td>
<td>47.51</td>
<td>43.47</td>
<td>27.32</td>
<td>40.48</td>
<td>32.40</td>
<td>15.91</td>
</tr>
<tr>
<td></td>
<td>✓</td>
<td>✓</td>
<td>53.32</td>
<td>52.17</td>
<td>33.52</td>
<td>46.85</td>
<td>44.60</td>
<td>26.85</td>
<td>39.94</td>
<td>33.70</td>
<td>17.10</td>
</tr>
</tbody>
</table>

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
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


[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


