Action Recognition for Each Person with Feature Extraction by Large-scale Object Detector

Akira Mitsuoka[1] and Kunihito Kato[1]

[1] Gifu University, 1-1 Yanagase, Gifu City, Gifu 501-1193, Japan
mitsuoka@cv.info.gifu-u.ac.jp

Abstract. In this study, we focus on human-object interaction (HOI) detection. In previous studies of HOI detection, information about person and objects was given at training phase. However, since there are countless objects, it is difficult to identify and teach objects that are relevant to actions. Given this fact, this research aims to be able to detect HOI based only the positions and actions of people. To achieve this, we attempted to introduce knowledge about objects implicitly by re-training the feature extractor of a large-scale object detector.

Keywords: Action Recognition, Human-Object-Interaction, Object Detection

1 Introduction

The advancements in deep learning have led to the resolution of numerous problems in computer vision. However, identifying individuals and recognizing their actions in video images remains underdeveloped. In particular, the recognition accuracy for actions that capture interactions between people and objects needs to be improved [1]. This study focuses on these actions. The ability to accurately identify these actions can lead to advancements in recognition of complex actions [2]. In human-object interaction detection, not only information related to people but also information related to objects is important. Conventional human-object interaction detection assumes that the positions and classes of both people and objects are taught during the learning process. However, due to the vast number of objects, it is difficult to identify and teach action-related objects in practical applications. With this in mind, we aim to detect human-object interactions only from the positions and actions of people.

To address this issue, we propose utilizing the features of a pre-trained large-scale object detector. A large-scale object detector such as Detic [3], which learns object detection from the ImageNet-21k dataset, enables the detection of objects across a wide range of classes. The feature extractor of Detic implicitly knows features that can identify various object positions and shapes. By re-training this feature extractor for person-by-person action detection, we implicitly introduce the knowledge of object positions and shapes and attempt to detect human-object interactions. The results of experiments using the dataset we created show that the feature extractor trained on 21,000 classes with Detic performs better in capturing human-object interactions than other feature extractors and weights. Furthermore, the experiments revealed that the operation of

Global Average Pooling (GAP) applied to the feature map is essential for recognizing human-object interactions.

The contributions of this paper can be summarized into two points:

1. We propose a model that effectively detects human-object interactions by utilizing knowledge of objects possessed by a large-scale object detector, without using information on object positions and classes during the learning process.
2. In the framework above, GAP applied to wide range of feature map is essential for recognition accuracy.

2 Related Works

2.1 Object Detection

Object detection is a problem predicting the position and class of objects in an image. Early detection models [4,5] using deep learning set up several anchor boxes tied to anchors on feature maps. These anchor boxes are considered correct when the IoU with the Bounding Box from the teacher data exceeds a certain threshold, and are used for training. These anchor boxes are generally in the thousands, so the problem was that it was easy to generate overlapping Bounding Boxes. To solve this problem, detection methods that use object keypoints [6,7,8] instead of anchor points were developed. Literature [6,7] embeds information such as object size at the center position of the object. CornerNet [8] embeds information at the object's top left and bottom right corner and groups them. These keypoint-based methods are simple and fast, but they ignore that the optimal keypoint position changes depending on the shape and occlusion of the object. Therefore, research on dynamic assignment methods [9,10,11] has been active in recent years. ATSS [9] assigns GT based on statistical features. OTA [10] approaches the assignment of predicted bounding boxes as an optimal transport problem. TOOD [11] trains so that the position of the embedded object classification branch and object localization branch get closer. Dynamic assignment methods have achieved improved accuracy on datasets such as COCO but require some additional computational cost. Based on the experimental observation that the localization accuracy is sufficient for the intended dataset, we use CenterNet[6], one of the keypoint detectors.

Large-scale object detection is generally defined as detecting more than 1000 object classes. In this problem setting, the class label distribution has a long-tail problem, and many studies are trying to solve it by addressing this problem. EQL [12] calculates the number of classes per total number of images in the dataset and weights the loss function with this value. Seesaw Loss [13] introduces two coefficients: a relaxation coefficient that reduces loss for classes with fewer numbers and a compensation coefficient that works to increase loss for False Positives, and learns by balancing these coefficients. Federated Loss [14] uses random sampling of class sets as learning data for each iteration and ignores the classes that were not sampled. In contrast, Detic tries to solve it from a different perspective by using class labels of images as additional data. In the field of existing research for learning detection from classification labels, there are weakly supervised object detection and semi-supervised object detection. The distinction between these and Detic is that Detic completely separates the learning of classifiers and object locators by only training the classifier using image data. This is based on the result that the locator itself is sufficiently generalized even for object classes that are not used in training [15]. By separating the training, detection improves accuracy for classes with fewer numbers, which was difficult to detect conventionally.

2.2 Action Recognition

In the context of action recognition, a task that aims to predict a person's location and action class is known as Spatio-Temporal Action Recognition (SAR). Typical methods [16, 17, 18] for SAR apply a pre-trained detector such as Faster-RCNN [4] to the feature maps obtained by a representative 3D CNN such as I3D [19] or SlowFast [20] and use the resulting regions of interest (RoI) for action recognition. Literature [21] argues that, instead of using feature maps, extracting people from the original video is more effective than using RoI. Furthermore, a method using tubelet [22] attempts to solve SAR by connecting detection results for each frame based on the detection score in the temporal direction. Methods that use graph representation [18, 23] generally define the embedding of information about people and objects as nodes and the edges connecting them as actions. While the introduction of graph representation in datasets such as AVA has achieved a certain level of accuracy improvement, there is the problem that the computational cost increases as the number of objects to be detected increases. Approaches that attempt to model the Spatio-temporal relationship between people and objects without using graph representation, such as [1, 24, 25, 26], can be mentioned. Many of these models require additional detection models in addition to the action recognition model. The main point of differentiation between our model and existing research is that our model attempts to perform action detection in a single model.

Human-Object-Interaction detection aims explicitly to capture spatial interactions between people and objects. Existing research in this field typically assumes that information about both person and objects is provided during training. HOI detection methods can be broadly divided into two-stage methods and one-stage methods. Two-stage methods [27,28,29] first extract features related to people and objects using a backbone network, similar to typical methods for Spatio-temporal action recognition. The extracted features are then used in multiple streams, such as object streams, person streams, and streams for capturing relationships, which are then combined to detect HOI. One-stage methods in literature [30,31,32,33] aim to solve this problem by not using external detectors. Literature [30,31] utilizing the mechanism of CenterNet defines the center point of the object, the person and the interaction, and matches these three points for HOI detection. UnionDet [32] detects Bounding Box surrounding the human and object to obtain HOI. QPIC [33] focuses on the expressiveness of DETR [34] and develops HOI detector based on DETR. These HOI detectors also need to predetermine the object related to the action among countless objects.

3 Method

3.1 Detic feature extractor

In order to utilize knowledge of various object positions and shapes, we employ the feature extractor of the large-scale object detector Detic. The feature extractor of Detic is composed of Swin-base [35] and FPN [36]. Swin is a general backbone network for images developed based on the success of ViT [37]. One of the differences between Swin and ViT is that Swin has a hierarchical downsampling structure. In ViT, the scale of tokens in the spatial direction is the same at all levels, but in images, the scale of objects can change due to various factors. Therefore, it may be problematic to fix the scale of tokens when performing object detection or semantic segmentation. By adopting the conventional CNN-like hierarchical downsampling structure, Swin allows changing the spatial scale of tokens and thus handling various object scales. Another difference is the presence of Shifted Window based Self-Attention. In ViT, the calculation of Self-Attention is proportional to the square of the image size, and the size of this calculation is a problem. With Shifted Window-based Self-Attention, the image is divided into a fixed-size window, and Self-Attention is calculated within it. Thus, the calculation is linear for the image size. Also, the connectivity between windows is provided by shifting the window at each layer. FPN is a top-down structure network with horizontal connections, in contrast to the bottom-up structure of Swin. By adopting FPN, feature extraction becomes more scale-invariant. We use only the topmost output of Detic FPN for subsequent action detection.

3.2 Proposed Model

We have constructed an action detection model as shown in figure 1, using the aforementioned feature extractor. Let $\mathbf{X} \in \mathbb{R}^{H \times W \times 3}$ be the input to the model, the output of the Detic feature extractor is $\mathbf{X}' \in \mathbb{R}^{H/8 \times W/8 \times 256}$, where H and W represent the width and height of the input image. $\mathbf{X}'$ is then upsampled to obtain the feature map $\mathbf{F} \in \mathbb{R}^{H/4 \times W/4 \times 256}$. $\mathbf{F}$ is input to the Heatmap Head, Offset Head, Size Head, and Action Head, respectively. The Heatmap Head, Offset Head, and Size Head are parts for person detection, based on CenterNet. The Heatmap Head, Offset Head, and Size Head respectively learn a heatmap representing the object's central position, an offset of the object center caused by downsampling, and the object's size. The Heatmap Head learns by mapping the feature map passed through the Sigmoid function to the teacher heatmap. The Offset Head and Size Head learn by selecting the vector corresponding to the person's center and mapping it to the teacher data. CenterNet is simple and fast, and can achieve sufficient accuracy in the intended detection problem; that is why we use it.

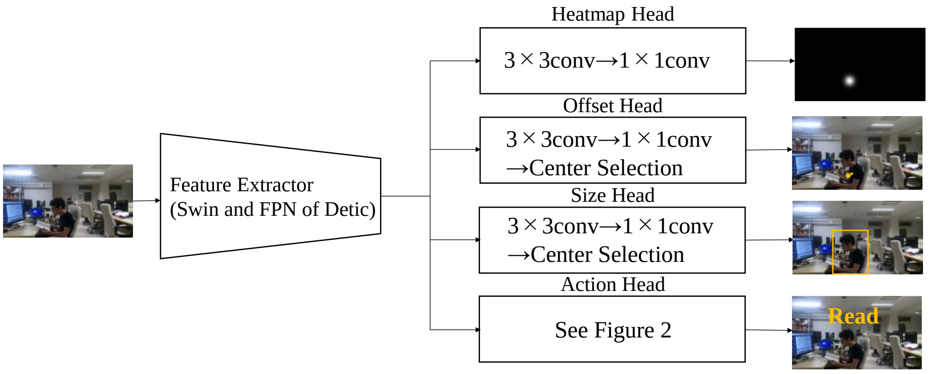


Fig. 1. Overview of the proposed model

The details of the Action Head are shown in Figure 2. In Figure 2, the dotted square represents the human region, while the solid square symbolizes the WRoI region to be described subsequently. Due to empirical observation, the action Head does not use center embeddings. In the Action Head, first, a single 3x3 convolution is applied to the feature map $\mathbf{F}$ to obtain the feature map $\mathbf{F}'$. We extract detection candidate $\hat{P}_i = \{(\hat{x}_1, \hat{y}_1, \hat{x}_2, \hat{y}_2)\}_{i=1}^{N}$ using Detection Head for the feature map $\mathbf{F}'$. $\hat{x}_1$, $\hat{y}_1$, $\hat{x}_2$, and $\hat{y}_2$ are the x-coordinate of the upper left corner, y-coordinate of the upper left corner, x-coordinate of the lower right corner, and y-coordinate of the lower right corner of the bounding box, respectively. N is the number of persons detected. This RoI range often does not include objects related to the HOI, which is a problem during recognition. Therefore, by correcting $\hat{x}_1$ and $\hat{x}_2$ using equation (1), we obtain WRoI with a size of $\hat{h} \times 2\hat{w}$ for each person by cropping the feature map. Through the WRoI, the model can consider objects around the person. We obtain the feature map $\mathrm{F_A} \in \mathbb{R}^{N \times \mathrm{C}}$ by applying GAP and linear layers to the WRoI, where C is the number of action classes, and $\mathrm{F_A}$ is mapped to the teacher data to train the Action Head. Comparative experiments for Action Head are discussed in more detail in Section 4.3.

$$\hat{w} = \hat{x}_2 - \hat{x}_1, \hat{x}_1 = \max(0, \hat{x}_1 - \frac{\hat{w}}{2}), \hat{x}_2 = \min(\frac{\mathrm{W}}{4}, \hat{x}_2 + \frac{\hat{w}}{2}) \qquad (1)$$

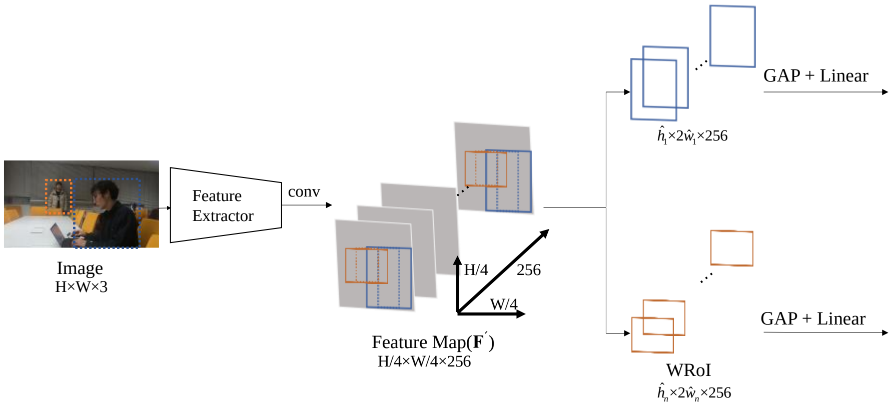


Image
H×W×3
Feature Extractor
conv
H/4
256
W/4
Feature Map($\mathbf{F}'$)
H/4×W/4×256
$\hat{h}_1 \times 2\hat{w}_1 \times 256$
GAP + Linear
WRoI
$\hat{h}_n \times 2\hat{w}_n \times 256$
GAP + Linear

Fig.2. The details of the Action Head

3.3 Loss Function

The loss function is based on Uncertainty Weight Loss [38]. Multi-task learning strongly depends on the relative weighting of the loss for each task, and it is not easy to adjust this weight manually. By using Uncertainty Weight Loss, it is possible to adjust the relative weight according to the task by making the learning parameter. When expressing the loss function of the Heatmap Head, Offset Head, Size Head, and Action Head respectively as L_{H}, L_{O}, L_{S}, and L_{A}, the total loss function L is represented by the following equation (2). Here, p_{D} and p_{A} are learning parameters. We use Focal loss [39] for L_{H}, L1 Loss for L_{O} and L_{S}, and Binary Cross Entropy Loss for L_{A}

$$L_{\mathrm{D}} = L_{\mathrm{H}} + 0.1L_{\mathrm{s}} + L_{\mathrm{O}}$$

$$L = \frac{1}{2}\left(\frac{1}{e^{p_{\mathrm{D}}}}L_{\mathrm{D}} + \frac{1}{e^{p_{\mathrm{A}}}}L_{\mathrm{A}} + p_{\mathrm{D}} + p_{\mathrm{A}}\right) \qquad (2)$$

4 Experiments

To verify the effectiveness of the proposed method in HOI detection, we conducted multiple experiments on the dataset we created. In all experiments, we used Adam as the optimizer and a batch size of 4. The learning rate was explored in the range of [1×10^{-5}, 1×10^{-4}] and the optimal learning rate was applied. The input images were resized to H=896, W=896 and. Also, we adopted random horizontal flipping and HSV color transformation for data augmentation. We trained on the dataset described later for 20 epochs with this setting. As a note, we conducted experiments in which the learning rate was changed for the feature extractor and other parts and experiments in which the feature extractor part was fixed. However, results that improved accuracy were not obtained.

4.1 Dataset and Evaluation Metrics

In order to create a dataset for experiments, we filmed inside a university laboratory at 1920 × 1080 and 10fps. We extracted frames from the video data as images, and for each person in each frame, we assigned a Bounding Box and a label of the action class to the person. The action classes selected as typical examples of HOI in the laboratory include "Reading" (Read), "Writing" (Write), "Drinking" (Drink), "Calling" (Call), "Operating a keyboard" (Keyboard), "Operating a mouse" (Mouse), "None of the above classes" (Nothing). Multiple of these action classes may correspond to a person. When collecting data, we made variations in the shooting location and tools used within the dataset to include different variations of the same action class. The details of the action class are shown in Table 1.

In existing action recognition datasets, it is often possible to determine the action class based solely on the pose of the person, the background, or the presence or absence of objects related to the HOI. Figure 3 illustrates an example from the V-COCO dataset [43]. Both Figure 3(a) and Figure3(b) have labels of the action class "work on computer" for each person in the image. By observing the V-COCO dataset, the label "work on computer" is given if computers exist in the image. Conversely, if the label "work on computer" is not given, it can be confirmed that computers do not exist.

Thus, machine learning models can determine the action class just by identifying the presence or absence of computers in the image, regardless of whether the person is working on the computer. Furthermore, existing datasets often have the person partially cut off, which makes learning for action detection unnecessarily challenging. Figure 3(b) is a prime example, with only the hands of the person in the image. In contrast, our dataset was carefully crafted to ensure that if the person's position, pose, object position, and object class cannot all be accurately determined, the action class will not be appropriately assigned. Figures 4(a) and 4(b) are examples of our dataset. Both images contain a keyboard, but only the individual on the left operating the keyboard is labeled as "Keyboard". Additionally, our dataset was constructed such that person's upper torso is visible throughout our dataset. This avoids complicating the localization aspect unnecessarily and focuses model's attention on HOI.

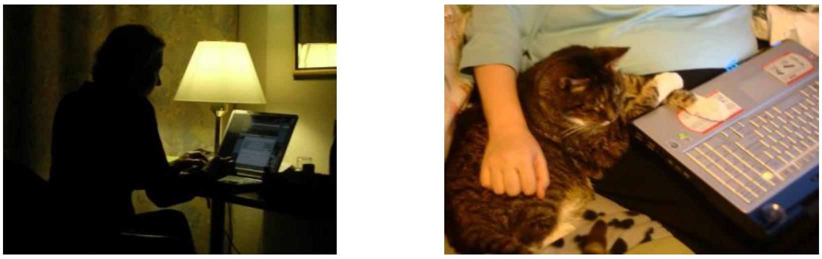


Fig3(a) Fig3(b)

Fig.3. Example of V-COCO dataset [43]

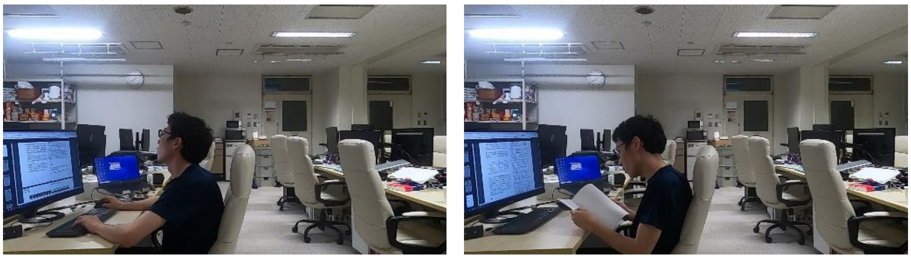


Fig4(a)

Fig4(b)

Fig.4. Example of our dataset

Table 1. The details of our dataset

	Read	Write	Drink	Call	Keyboard	Mouse	Nothing
Train	4,816	1,210	204	815	1,016	909	2,519
Val	1,518	260	94	123	616	611	1,109
Test	2,220	279	121	131	920	300	448

As evaluation metrics, we employ Micro-F1-Accuracy and mAP. We use Micro-F1-Accuracy to evaluate only the accuracy of action recognition in our validation data. On the other hand, mAP is a commonly used evaluation metric for object detection and action detection. We use mAP to verify both localization and recognition accuracy on the test data. We use a threshold of 0.7 for F1-Accuracy and 0.5 for IoU in mAP to determine detection as correct.

4.2 Comparison of Feature Extractors

We compared feature extractors to demonstrate the effectiveness of Detic feature extractor in recognizing HOI. The results are shown in Table 2. We used AlexNet [40] and DLA34 [41] for comparison with traditional CNN-based feature extractors, MViT [42] and Swin for comparison with transformer-based feature extractors, and Swin+FPN for comparison with other weights. We used AlexNet and DLA34 with ImageNet-1k pre-trained models, Swin and MViT with ImageNet-21k pre-trained models, Swin+FPN with LVIS pre-trained model, and Detic with LVIS-COCO and ImageNet-21K pre-trained models.

Table 2 shows that the Swin-based models are superior to the other models in terms of accuracy. This result indicates that Swin is an effective feature extractor in images. Next, among the Swin-based models, Detic shows the best accuracy. This result is because Detic has been trained to detect a wide range of object classes and thus has knowledge about object position and shape compared to other feature extractors.

Table 2. Comparison of feature extractors in our validation data

Feature Extractor	F1-Accuracy
AlexNet[40]	0.381
DLA34[41]	0.491
MViT[42]	0.510
Swin[35]	0.558
Swin+FPN	0.575
Detic[3]	**0.718**

4.3 Comparison of Action Head

In order to investigate the optimal structure in Action Head, we conducted experiments comparing this part by fixing the feature extractor to Detic. The results are shown in Table 3. In the table, Center refers to cases where information is embedded in the center, like CenterNet. GAP (RoI) [16] refers to the result of applying RoI Pooling [4] to the feature map $\mathbf{F}^{'}$ and then applying GAP. ACRN [1] attempts to improve the action classification accuracy without object information. ACRN first obtains the feature map $\mathbf{A}$ by replicating the feature map $\mathbf{F}^{'}$ per individual, and the feature map $\mathbf{I}$ by expanding it to the spatial dimensions of $\mathbf{A}$ following the application of GAP to the RoI. ACRN calculates the concatenation of $\mathbf{A}$ and $\mathbf{I}$, then applies several convolutions to this feature map and finally applies GAP. NL[26] applies Non-Local Block to feature map $\mathbf{F}^{'}$ and then applies GAP to the resulting feature map.

From Table 3, firstly, it can be seen that Center is unsuitable for HOI recognition. It also shows that GAP (WRoI), which provides a broader area, and GAP (Feature Map), which provides the entire feature map, are more suitable than GAP (RoI), which provides only a part of the feature map. These results suggest that HOI recognition requires looking at a wide area of the image. Also, ACRN and NL, which perform operations on RoI or feature map $\mathbf{F}^{'}$, did not improve accuracy compared to GAP(RoI) or GAP(Feature Map). This result suggests that the operation of GAP applied to the feature map is essential. Also, it is believed that the reason for not improving the accuracy further is that the Detic feature extractor has already established correlations between individuals and objects.

Table 3. Comparison of Action Head structure in our validation data

Action Head	F1-Accuracy
Center	0.466
ACRN[1]	0.591
NL[26]	0.598
GAP(RoI)[16]	0.611
GAP(WRoI)	**0.718**
GAP(Feature Map)	0.712

4.4 Detection Result

Figure 5 demonstrates instances in which the proposed model successfully detects HOI. As shown in Figures 5(a), 5(b), and 5(c), the keyboard and mouse, which are objects relevant to HOI, are present, but only in Figure 5(a) where the person is operating the keyboard and mouse does the model output 'Keyboard' and 'Mouse.' This confirms that the proposed model recognizes HOI by considering not only the position of the person but also their pose, the position of the object, and the class of the object. Furthermore, Figures 5(d), 5(e), and 5(f) depict scenes captured from different locations but still demonstrate the ability of the model to recognize HOI accurately. This indicates that the proposed model can generalize to HOI, regardless of the capture location. Additionally, in Figure 5(d), the model can output 'Nothing' when there is no interaction with an object, further demonstrating its capabilities.

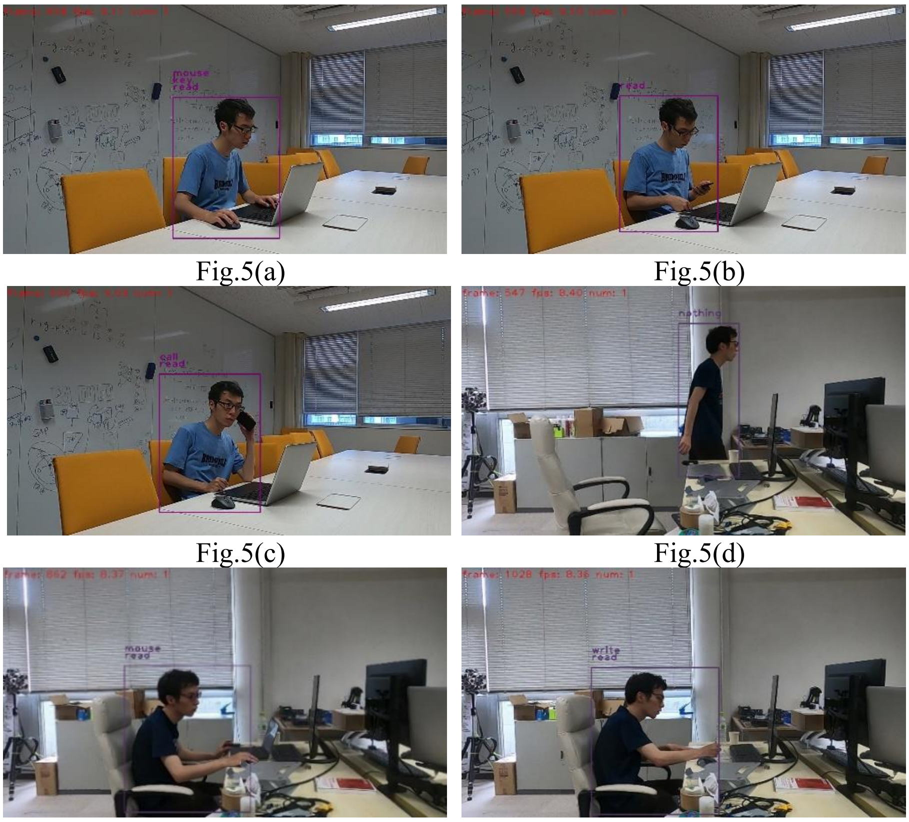


Fig.5(a) Fig.5(b)

Fig.5(c) Fig.5(d)

Fig.5(e) Fig.5(g)

Fig. 5. Example of successful HOI detection by our proposed model

Figure 6 shows examples of the proposed model's failure in detecting HOI. Figure 6(a) is an example of over-detection, which is caused by insufficient learning of the detection task for the HOI recognition task. This problem can be resolved by sufficiently training the model, as the positions of the detections are not incorrect in many cases. Figure 6(b) is an example of incorrect detection of HOI, where the model identified incorrectly due to the right hand being close to the mouse and having a pose close to holding the mouse. Figures 6(c) and 6(d) are examples of HOI not being detected. Figure 6(c) fails to detect "Read" and "Keyboard", while Figure 6(d) fails to detect "Drink". This is due to a shortage of training data for scenes of one-handed keyboard operation and scenes with the "Drink" label, which can be resolved by increasing the relevant training data.

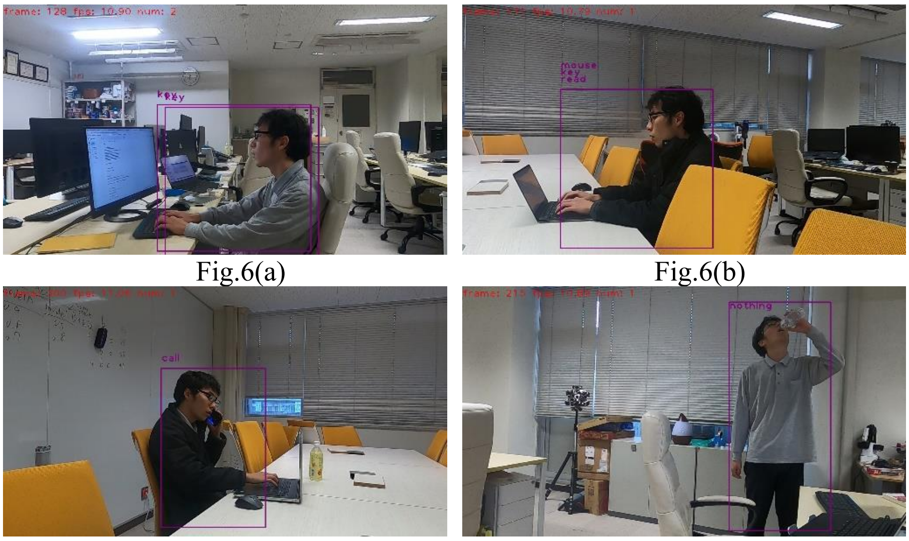


Fig.6(a)

Fig.6(b)

Fig.6(c)

Fig.6(d)

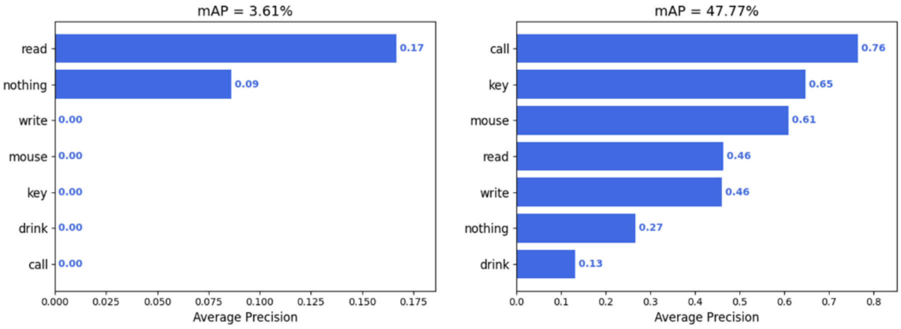
Fig. 6. Example of unsuccessful HOI detection by our proposed model

Table 4 shows the comparison of mAP on the test data. In terms of feature extractors, Detic achieved better results than other models and weights. Furthermore, in the Action Head, it was effective for all feature extractors to apply GAP to WRoI instead of Center. This confirms the effectiveness of the proposed method.

Figure 7 shows the results of mAP for each class. When using the DLA34 feature extractor and the Center action head, the model made random predictions for "Read" and "Nothing" while not predicting any other class. On the other hand, the proposed model could make predictions to some extent for each class. Also, it can be confirmed from the figure that the accuracy of "Drink" is poor throughout the dataset due to the lack of "Drink" samples in the training data. This problem can be addressed by improving the training dataset.

Table 4. Comparison of detection accuracy on our test data

Feature Extractor	Action Head	mAP
DLA34	Center	3.61
DLA34	WRoI	8.94
Swin＋FPN	Center	6.92
Swin＋FPN	WRoI	35.0
Detic	Center	11.4
Detic	WRoI	**47.7**



mAP in case of DLA34+Center

mAP in case of Detic+WRoI (Proposed model)

Fig. 7. The results of mAP for each class

5 Conclusion

In this study, we proposed to use the knowledge of a large-scale object detector's feature extractor to detect human-object interactions solely based on the position of people and the actions related to them. Through experiments on the created dataset, we confirmed that our proposed method can recognize HOI more accurately than using other feature extractors or weights. We also showed that the operation of GAP applied to the wide range of feature map is essential when recognizing HOI. Furthermore, the proposed model demonstrated the capability of performing individualized action recognition end-to-end.

Actions involve not only the spatial property of how a person interacts with an object but also the temporal property of how the person has acted. In this study, we did not consider the temporal aspect when recognizing actions. As a prospect, we aim to improve the detection of a broader range of action classes by associating HOI over time through tracking and by building a model based on this information.

References

1. Sun, C., Shrivastava, A., Vondrick, C., Murphy, K., Sukthankar, R. and Schmid, C.:Actor-centric relation network. Proceedings of the European Conference on Computer Vision, pp.318-334, (2018)
2. Sugimura, Y., Uchida, D. Suzuki, G. and Endou, T.:Proceedings of the Annual Conference of JSAI JSAI2020 (0),pp.4Rin157-4Rin157, (2020)
3. Zhou, X., Girdhar, R., Joulin, A., Krähenbühl, P. and Misra, I.:Detecting twenty-thousand classes using image-level supervision. Proceedings of European Conference on Computer Vision, pp.350-368, (2022)
4. Ren, S., He, K., Girshick, R. and Sun, J.:Faster r-cnn: Towards real-time object detection with region proposal networks. Advances in neural information processing systems (28), (2015)
5. Redmon, J. and Farhadi, A.: Yolov3: An incremental improvement. arXiv preprint arXiv:1804.02767, (2018)
6. Zhou, X., Wang, D. and Krähenbühl, P.: “Objects as points”, arXiv preprint arXiv:1904.07850.（2019)
7. Tian, Z., Shen, C., Chen, H. and He, T.: Fcos: Fully convolutional one-stage object detection. Proceedings of the IEEE/CVF international conference on computer vision, pp.9627-9636, (2019)
8. Law, H. and Deng, J.: Cornernet: Detecting objects as paired keypoints. Proceedings of the European conference on computer vision, pp.734-750. (2018)
9. Zhang, S., Chi, C., Yao, Y., Lei, Z. and Li, S. Z.: Bridging the gap between anchor-based and anchor-free detection via adaptive training sample selection. Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp.9759-9768. (2020)
10. Ge, Z., Liu, S., Li, Z., Yoshie, O. and Sun, J.: Ota: Optimal transport assignment for object detection. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 303-312. (2021)

11. Feng, C., Zhong, Y., Gao, Y., Scott, M. R. and Huang, W.: Tood: Task-aligned one-stage object detection. Proceedings of the IEEE/CVF International Conference on Computer Vision. pp.3490-3499. (2021)
12. Tan, J., Wang, C., Li, B., Li, Q., Ouyang, W., Yin, C. and Yan, J.: Equalization loss for long-tailed object recognition. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp.11662-11671. (2020)
13. Wang, J., Zhang, W., Zang, Y., Cao, Y., Pang, J., Gong, T., ... and Lin, D.: Seesaw loss for long-tailed instance segmentation. Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp. 9695-9704. (2021)
14. Zhou, X., Koltun, V. and Krähenbühl, P.: Probabilistic two-stage detection. arXiv preprint arXiv:2103.07461. (2021)
15. Zareian, A., Rosa, K. D., Hu, D. H. and Chang, S. F.: Open-vocabulary object detection using captions. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition,pp. 14393-14402. (2021)
16. Gu, C., Sun, C., Ross, D. A., Vondrick, C., Pantofaru, C., Li, Y. ... and Malik, J.: Ava: A video dataset of spatio-temporally localized atomic visual actions. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition pp. 6047-6056. 2018
17. Girdhar, R., Carreira, J., Doersch, C. and Zisserman, A.: Video action transformer network. Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp.244-253. (2019)
18. Zhang, Y., Tokmakov, P., Hebert M. and Schmid, C.: A structured model for action detection, Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 9975-9984, (2019)
19. Carreira, J. and Zisserman, A.: Quo vadis, action recognition? a new model and the kinetics dataset. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp.6299-6308. (2017)
20. Feichtenhofer, C., Fan, H., Malik, J. and He, K.: Slowfast networks for video recognition. Proceedings of the IEEE/CVF international conference on computer vision, pp. 6202-6211. 2019
21. Wu, J., Kuang, Z., Wang, L., Zhang, W. and Wu, G.: Context-aware rcnn: A baseline for action detection in videos. Proceedings of European Conference on Computer Vision, pp.440-456. (2020)
22. Singh, G., Saha, S., Sapienza, M., Torr, P. H. and Cuzzolin, F.: Online real-time multiple spatiotemporal action localisation and prediction. Proceedings of the IEEE International Conference on Computer Vision, pp.3637-3646. (2017)
23. Wang, X. and Gupta, A.: Videos as space-time region graphs. Proceedings of the European conference on computer vision, pp.399-417. (2018)
24. Jiajun, T., Jin, X., Xinzhi, M., Bo, P. and Cewu, L.: Asynchronous interaction aggregation for action detection. Proceedings of European Conference on Computer Vision, pp.71−87, (2020)
25. Pan, J., Chen, S., Shou, M. Z., Liu, Y., Shao, J. and Li, H.: Actor-context-actor relation network for spatio-temporal action localization. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition pp.464-474, (2021)
26. Wang, X., Girshick, R., Gupta, A., and He, K.: Non-local neural networks. Proceedings of the IEEE conference on computer vision and pattern recognition, pp.7794-7803. 2018
27. Chao, Y. W., Liu, Y., Liu, X., Zeng, H. and Deng, J.: Learning to detect human-object interactions. Proceedings of IEEE winter conference on applications of computer vision, pp.381-389. (2018).

28. Gao, C., Xu, J., Zou, Y., and Huang, J. B.: Drg: Dual relation graph for human-object interaction detection. Proceedings of European Conference on Computer Vision, pp.696-712. (2020)
29. Gao, C., Zou, Y., and Huang, J. B.: ican: Instance-centric attention network for human-object interaction detection. arXiv preprint arXiv:1808.10437. (2018)
30. Liao, Y., Liu, S., Wang, F., Chen, Y., Qian, C. and Feng, J.: Ppdm: Parallel point detection and matching for real-time human-object interaction detection. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp.482-490. (2020)
31. Wang, T., Yang, T., Danelljan, M., Khan, F. S., Zhang, X. and Sun, J.: Learning human-object interaction detection using interaction points. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp.4116-4125. (2020)
32. Kim, B., Choi, T., Kang, J. and Kim, H. J.: Uniondet: Union-level detector towards real-time human-object interaction detection. Proceedings of European Conference on Computer Vision, pp. 498-514. (2020)
33. Tamura, M., Ohashi, H., and Yoshinaga, T.: Qpic: Query-based pairwise human-object interaction detection with image-wide contextual information. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp.10410-10419. (2021)
34. Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A. and Zagoruyko, S.: End-to-end object detection with transformers. In European conference on computer vision, pp. 213-229. (2020)
35. Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., ... and Guo, B.: Swin transformer: Hierarchical vision transformer using shifted windows. Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 10012-10022. (2021)
36. Lin, T. Y., Dollár, P., Girshick, R., He, K., Hariharan, B. and Belongie, S.: Feature pyramid networks for object detection. Proceedings of the IEEE conference on computer vision and pattern recognition, pp.2117-2125. (2017)
37. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., ... and Houlsby, N.:An image is worth 16x16 words: Transformers for image recognition at scale. arXiv preprint arXiv:2010.11929. (2020)
38. Kendall, A., Gal, Y. and Cipolla, R.: Multi-task learning using uncertainty to weigh losses for scene geometry and semantics. In Proceedings of the IEEE conference on computer vision and pattern recognition, pp.7482-7491. (2018)
39. Lin, T. Y., Goyal, P., Girshick, R., He, K. and Dollár, P. Focal loss for dense object detection. In Proceedings of the IEEE international conference on computer vision pp. 2980-2988. (2017)
40. Krizhevsky, A., Sutskever, I. and Hinton, G. E. Imagenet classification with deep convolutional neural networks. Communications of the ACM, pp.84-90, (2017)
41. Yu, F., Wang, D., Shelhamer, E. and Darrell, T. Deep layer aggregation. Proceedings of the IEEE conference on computer vision and pattern recognition pp. 2403-2412. (2018)
42. Fan, H., Xiong, B., Mangalam, K., Li, Y., Yan, Z., Malik, J. and Feichtenhofer, C.: Multiscale vision transformers. Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 6824-6835. (2021)
43. Saurabh, G and Jitendra, M.: “Visual semantic role labeling”, arXiv preprint arXiv:1505.04474, (2015)