Paper:

Object–Action Interaction Region Detection in Egocentric Videos

Shinobu Takahashi, Kazuhiko Kawamoto

Chiba University., 1-33, Yayoi-cho, Inage-ku, Chiba-shi, Chiba, 263-8522 Japan
E-mail: shinobu-graffiti@chiba-u.jp
kawa@faculty.chiba-u.jp

Abstract. We propose a deep model for detecting object–action interaction regions in egocentric videos. This task includes both object detection and action recognition simultaneously, and we need to detect the only object involved in the action. We design a two–stream deep architecture that enables learning by annotating a single frame in a video clip so that we can avoid the time-consuming annotation that assigning bounding boxes and object classes to every frame of the video clip. In this paper, we report the results of a comparative evaluation of four possible structures of the output layer of the two-stream architecture for multitask learning. Experimental results on the EPIC-KITCHENS dataset shows that the structure of fusing object detection and action recognition provides better performance than the other structures.

Keywords: Object–Action Interaction, Egocentric Activity Recognition, Deep Learning

1. Introduction

As part of egocentric activity recognition, the ability to explicitly detect the region of an object involved in a recognized action would be useful for interaction analysis such as grasp analysis [1], important object detection [2], attention estimation [3], and fine-grained video logging. **Figure 1** illustrates the difference between object detection and object–action interaction region detection. The former detects all objects in an image and the latter detects only the region showing an interaction relationship between the object and the action. In this study, we propose a multitask learning method that adds region detection to egocentric activity recognition. Several studies have detected an object involved in an activity from an image[1, 2] and from a video[3]. Our method not only detects the object from a video, similar to [3], but also recognizes the activity.

Accomplishing this task requires attaching a bounding box to every frame, which is very costly. To solve this problem, we propose a model that can detect interaction regions in all frames of a video clip while only requiring a bounding box in one frame. The proposed model combines a two-stream architecture used in third-person activity detection with deep multitask learning in which activity recognition is learned separately for object recognition and action recognition, as shown in **Fig.2**. However, a method to determine from which output layer of the architecture, the three outputs of object recognition, action recognition and region detection should come, has not yet been established. In this study, we propose four models to determine the structure of the output layer for multitask learning. We then conduct validation experiments using the EPIC-KITCHENS dataset[4].

(a) Object detection (b) Interaction region detection

Fig. 1. Difference between object detection and object–action interaction region detection.

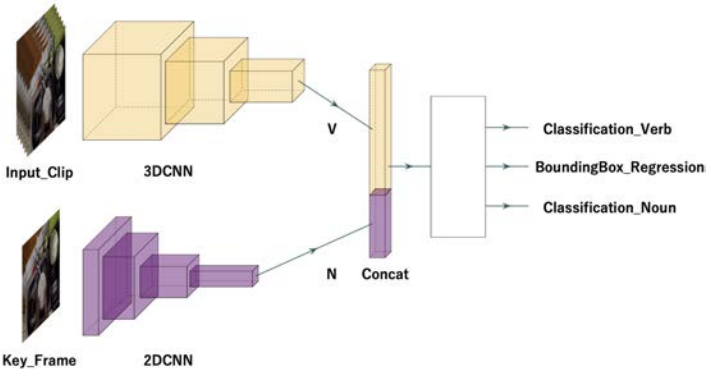
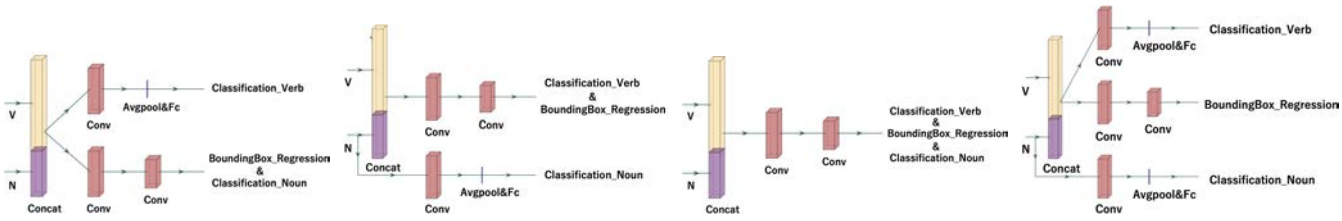


Fig. 2. Backbone model

2. Related work

2.1. Egocentric activity recognition

In the study of egocentric activity recognition, deep multitask learning is proposed to enable diverse activity recognition by dividing activities into objects and actions[5, 6]. The proposed method uses deep multitask learning to apply the model used in third-person activity



(a) Proposed model 1 (b) Proposed model 2 (c) Proposed model 3 (d) Proposed model 4

Fig. 3. Proposed models

detection to egocentric activity recognition. Some studies have improved accuracy in datasets that have training data such as viewpoint and hand location in every frame by explicitly learning their location information[7, 8]. These studies differ from our study in that they use the location information of all the frames for learning and output the viewpoints and hand location, although they are similar to our study in that they learn the location information explicitly. Other studies have used object detectors to obtain and use position information for all frames to improve accuracy[9]. Although these studies use location information with an object detector, they are different from ours because they do not include a location output task.

2.2. Third-person activity detection

Third-person activity detection is the task of detecting humans and recognizing their activities in each frame of a video. Some studies on such detection obtain human location information using object detectors[10].

By contrast, YOWO[11] explicitly learns human location information from videos itself. YOWO is a two-stream architecture that uses a 3D-CNN to extract spatio-temporal features and a 2D-CNN to extract spatial features. The architecture of YOWO uses a video clip as input to the 3D-CNN; a frame of the video clip is used as input to the 2D-CNN as a keyframe. The output of YOWO is the same as that of YOLOv2[12]. YOWO detects the regions containing humans and recognizes their activities in key frames, allowing the network to learn even if only one frame of a video clip has human location information. By using YOWO as a basis for our proposed method, the computation cost of attaching a bounding box to every frame is avoided.

3. Proposed method

Object–action interaction region detection in egocentric videos consists of two tasks: egocentric activity recognition and region detection from videos. We combine deep multitask learning and third-person activity detection to address these two tasks. These two tasks requires object recognition, whereas third–person activity detection, such as YOWO [8], does not do it because the target object is only human. Hence, our deep architecture has three outputs for object recognition, action recognition, and region detection, as shown in **Fig.2**. The aim of this paper is to determine the late fusion structure of the architecture for these three tasks. To do it, we consider four possible structures, as shown in **Fig.3**.

Proposed model 1 is designed to compute object recognition and region detection from the same output, and action recognition from another output as in **Fig.3(a)**, based on the idea of adding action recognition to object detection. The output structure of the region detection including object recognition is represented as $S \times S \times [B \times (4 + 1 + k_{noun})]$, where (S, B, k_{noun}) is the number of grids, the number of anchor boxes, and the number of object classes, respectively. The "4" indicates that the four information of the center position, width, and height of the bounding box are stored, and "1" indicates that the information of the confidence score is stored. The loss function of the region detection is the sum of the position loss L_{pos} and the confidence loss L_{conf}, and each of them is defined as follows:

$$L_{pos} = \sum_{i=1}^{S^2} \sum_{j=1}^{B} \mathbf{1}_{ij} \{ (\hat{x}_{ij} - x_{ij})^2 + (\hat{y}_{ij} - y_{ij})^2 + (\hat{w}_{ij} - w_{ij})^2 + (\hat{h}_{ij} - h_{ij})^2 \} \quad (1)$$

$$L_{conf} = \sum_{i=1}^{S^2} \sum_{j=1}^{B} \{ \mathbf{1}_{ij} (\hat{c}_{ij} - 1)^2 + (1 - \mathbf{1}_{ij}) (\hat{c}_{ij} - 0)^2 \} \quad (2)$$

where $(\hat{x}, \hat{y}, \hat{w}, \hat{h})$ is the predicted value of the center position, width, and height of the bounding box, respectively, and those with no $\hat{}$ denote the true value. c is the confidence score. The value of $\mathbf{1}_{ij}$ is 1 for a correct grid, otherwise 0. The loss function of the region detection is the same in all the proposed models. The loss functions L_{noun} and L_{verb} of object recognition and action recognition are defined as follows:

$$L_{noun} = \sum_{i=1}^{S^2} \sum_{j=1}^{B} \sum_{k=1}^{k_{noun}} \{ \mathbf{1}_{ij} (p_{ijk} \log \hat{p}_{ijk}) \} \quad (3)$$

$$L_{verb} = \sum_{k=1}^{k_{verb}} (p_k \log \hat{p}_k) \quad (4)$$

where k_{verb} is the number of action classes and $p \in [0, 1]$ is class probability. The loss function of the entire architecture is the sum of L_{pos}, L_{conf}, L_{noun}, and L_{verb} after

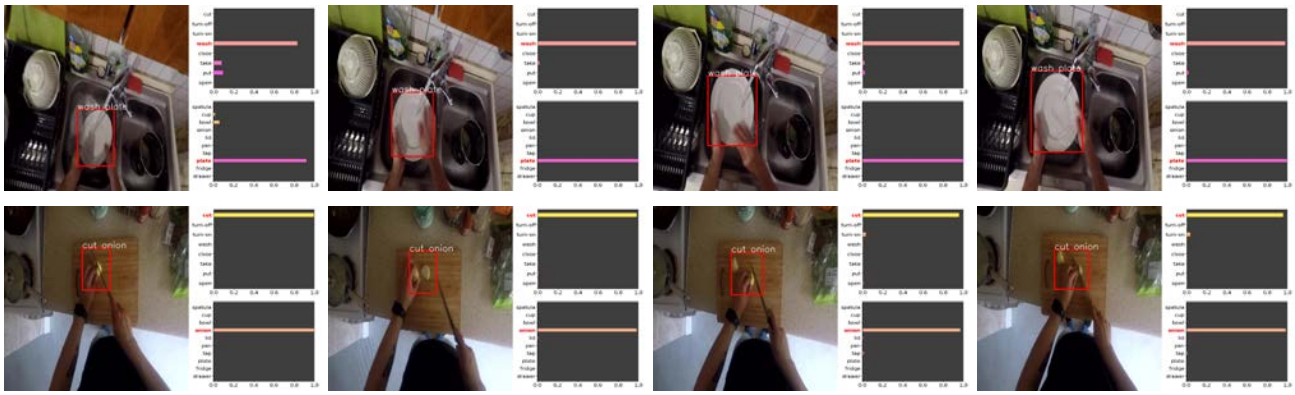


Fig. 4. Successful detection results: correct action label is *wash_plate* in the top row and *cut_onion* in the bottom row.

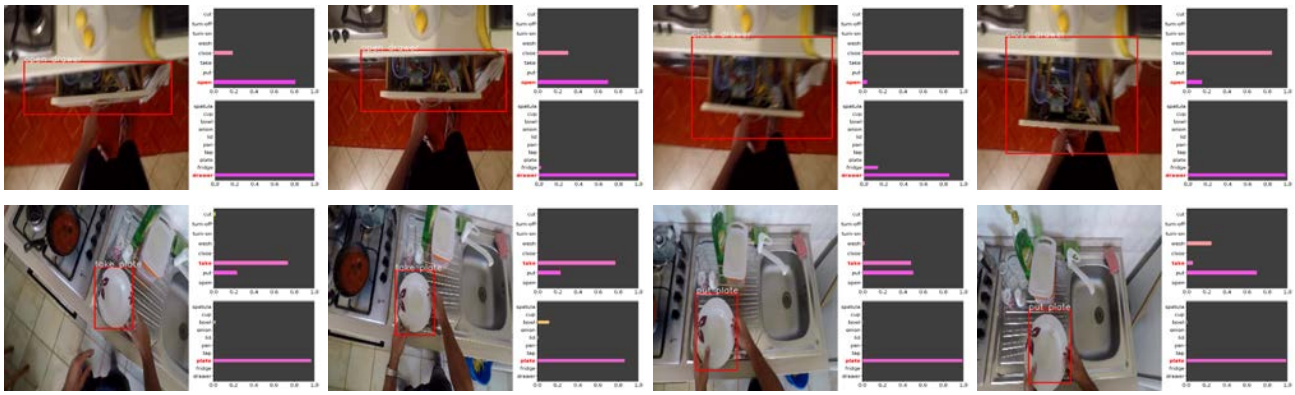


Fig. 5. Unsuccessful detection results: correct action label is *open_drawer* in the top row and *take_plate* in the bottom row.

multiplying each of them by each coefficient. Note that the coefficients are the same in all the proposed models.

Proposed model 2 is symmetrical to proposed model 1, and is designed to compute action recognition and region detection from the same output, and object recognition from another output as in **Fig.3(b)**. The output structure of the region detection including action recognition is represented as $S \times S \times [B \times (4+1+k_{verb})]$. The loss functions L_{noun} and L_{verb} are defined as follows:

$$L_{noun} = \sum_{k=1}^{k_{noun}} (p_k \log \hat{p}_k) \tag{5}$$

$$L_{verb} = \sum_{i=1}^{S^2} \sum_{j=1}^{B} \sum_{k=1}^{k_{verb}} \{\mathbf{1}_{ij}(p_{ijk} \log \hat{p}_{ijk})\} \tag{6}$$

Proposed model 3 is designed to compute object recognition, action recognition, and region detection from one and the same output as **Fig.3(c)**, based on the idea of dividing output of YOWO into object recognition and action recognition. The output structure in this model is represented as $S \times S \times [B \times (4+1+k_{noun}+k_{verb})]$. The loss functions L_{noun} and L_{verb} are defined Eq.3 and Eq.6 respectively.

Proposed model 4 is designed to compute object recognition, action recognition and region detection from separate outputs as **Fig.3(d)**. The output structure of the region detection in this model is represented as $S \times S \times [B \times (4+1)]$. The loss functions L_{noun} and L_{verb} are defined Eq.5 and Eq.4 respectively.

V and N in **Fig.3(a)** and **Fig.3(d)** are features obtained from the 3D-CNN and 2D-CNN of the common backbone, respectively. "Classification Noun", "Classification Verb", and "Bounding Box Regression" represent the output of object recognition, action recognition, and region detection, respectively. We treat the region with the highest class-specific confidence score among the output of region detection as the detected region.

4. Experiments in egocentric videos

In this experiment, we assess the four proposed models using activity scene in kitchens. We use a model without region detection for comparison with the proposed models. For the model without region detection, we use proposed model 4 but with the output layer of region detection removed. Note that the backbone is the same as the proposed models.

Table 1. Experimental results

	Object	Action	Activity	Region	Activity∧Region
Proposed model 1	0.677	0.620	0.477	0.520	0.333
Proposed model 2	0.712	0.619	0.488	0.548	0.367
Proposed model 3	0.683	0.609	0.499	0.535	0.361
Proposed model 4	0.717	0.629	0.508	0.565	0.345
Without region detection	0.725	0.645	0.528		

4.1. Dataset

We use EPIC-KITCHENS for our experiments. EPIC-KITCHENS is a collection of 432 videos by 32 participants performing kitchen-related activities in their homes; it consists of 125 action classes and 331 object classes. In this study, we extracted 10 object classes, 8 action classes, and 24 activity classes (combinations of objects and actions) from this dataset to construct a new dataset.

4.2. Evaluation method

As an evaluation index, we define *Acc* as the accuracy of objects, actions, activities, and regions. An activity is correct when both the object and the action are correct. The region is considered correct when the intersection over union (IoU) between the correct label and the predicted label obtained from the input/output is greater than 0.5. In our study, we denote the accuracy when the region and the activity are both correct as "Activity∧Region" *Acc*. "Activity∧Region" *Acc* is the true accuracy of object–action region interaction detection.

Fig. 6. Confusion matrix of the action recognition results in proposed model 2

4.3. Implementation details

We use ResNet50 [13] fine-tuned with the Kinetics dataset[14] as a 3D-CNN. An architecture based on YOLOv2 pre-trained in ImageNet[15] is used as a 2D-CNN. To reduce overfitting issues, we use data augmentation techniques that exploit random spatial cropping and random horizontal flipping approaches. We use the mini-batch stochastic gradient decent algorithm with momentum and a weight decay strategy to optimize the loss function. We set the number of training epochs to 150, and batch size to 8. The learning rate is initialized as 0.0001 and reduced with a factor of 0.5 after 50 epochs.

4.4. Results and Discussions

The results are summarized in **Table 1**. In Activity∧Region *Acc*, proposed model 2 was the highest with 36.7%, and proposed model 3 was the second highest with 36.1%, which are both about 3% and 2% higher than proposed model 1 and proposed model 4, respectively. Among the proposed models, proposed model 4, which has three separate outputs, was the highest value in all *Acc* except Activity∧Region *Acc*. The model without region detection outperformed the proposed models in Object, Action, and Activity *Acc*. From the above, we consider that the interrelationships among object recognition, action recognition, and region detection is learned more when the same output is used for both action recognition and region detection, or when the same output is used for object recognition, action recognition, and region detection. For the task in our study, it is effective to use the same output for action recognition and region detection at least for improving the accuracy.

Figure 4 and **Figure 5** show examples of successful and unsuccessful detection results of proposed model 3, respectively. The top-right and bottom-right bar graphs in each figure show the probability of action and object classification, respectively, and the red square on the left represents the detected region. In **Fig.5**, "*open*" is misrecognized as "*close*", and "*take*" is misrecognized as "*put*". **Figure 6** shows the confusion matrix of the action recognition results of proposed model 2. The vertical axis represents the correct label and the horizontal axis represents the predicted label. In **Fig.6**, it can be seen that the diagonal components are large and the recognition is mostly correct, but there are many misrecognitions in contrastive actions such as "*open*"-"*close*", "*put*"-"*take*", and "*turn-on*"-"*turn-off*". These results suggest that greater consideration of time series is needed.

5. Conclusion

In this study, we proposed a model for detecting the object–action interaction region in egocentric videos by combining a two-stream architecture used in third-person activity detection with deep multitask learning. The proposed models can reduce the cost of annotation because the interaction region can be detected in all frames if there is a bounding box in one frame of a video clip. We performed experiments on four different models with different structures of the output layers and found that the accuracy is improved when the same output is used for at least action recognition and region detection.

Acknowledgements

This work was supported by JSPS KAKENHI Grant Number JP19K12039.

References:

[1] S. Dandan et al, Understanding Human Hands in Contact at Internet Scale. CVPR, pp.9869-9878, 2020.

[2] B. Geda et al, Unsupervised learning of important objects from first-person videos. ICCV, pp.1956-1964, 2017.

[3] Z. Zehu et al, A Self Validation Network for Object-Level Human Attention Estimation. NeurIPS, pp.14729-14740, 2019.

[4] D. Damen et al., Scaling egocentric vision: The epic-kitchens dataset. ECCV，pp.720-736, 2018.

[5] M. Ma et al., Going deeper into first-person activity recognition. CVPR, pp.1894-1903, 2016.

[6] S. Kobayashi and K. Kawamoto, First-Person Activity Recognition by Deep Multi-task Network with Hand Segmentation. In Proc. of ISCIIA&ITCA, 5 pages, 2018.

[7] G. Kapidis et al., Multitask Learning to Improve Egocentric Action Recognition. ICCV, pp.4396-4405, 2019.

[8] M. Lu et al., Learning Spatiotemporal Attention for Egocentric Action Recognition. ICCV, pp.4425-4434, 2019.

[9] X. Wang et al., Symbiotic Attention with Privileged Information for Egocentric Action Recognition. arXiv preprint arXiv:2002.03137, 2020.

[10] Y. Zhang et al., A structured model for action detection. CVPR, pp.9975-9984, 2019.

[11] O. Köpüklü et al., You Only Watch Once: A Unified CNN Architecture for Real-Time Spatiotemporal Action Localization. arXiv preprint arXiv:1911.06644, 2019.

[12] J. Redmon et al., YOLO9000: better, faster, stronger. CVPR, pp.7263-7271, 2017.

[13] C. Szegedy et al., Inception-v4, inception-resnet and the impact of residual connections on learning. AAAI, 2017.

[14] W. Kay et al., The kinetics human action video dataset. arXiv preprint arXiv:1705.06950, 2017.

[15] J. Deng et al., Imagenet: A large-scale hierarchical image database. CVPR, pp.248-255, 2009.