Function-based Object Classification without Explicit Action Labeling

Hyunseo Kim, Yonggeun Shin, Changhee Kim, Sangyoon Kim
Seoul National University
hskim@bi.snu.ac.kr, {ygshin, andykim0531, oiei0806}@snu.ac.kr

Abstract

Function-based object classification is another way of classification that additionally utilizes human action-related information. Our network models, visuo-motor map (VMM) and visuo-motor classifier (VMC) learn the relationship between images, action, and object labels. By implementing VMM and VMC in 3 ways, Conv3D, S3D+attention, and detection network using Faster-RCNN, the utility of action information in object classification and detection are thoroughly investigated. All three models showed greater-than-or-equal-to performances compared to baseline algorithms, and the quality of output is shown in images and attention mappings.

1. Introduction

Object classification has been the core of vision machine learning, and still it is getting attention as a test bed whenever new techniques come out. Being a typical example of supervised learning, object classification learns the mapping between the static image and corresponding object label. Stillness in image data demanded many techniques to improve robustness in classification: data augmentation technique, attention mapping, and carefully organized deep convolution layers. However, object classification still easily suffers from occlusion, viewpoint variation, etc. One might say then video object detection within a sequence of images will ease the problem, but video object detection includes temporal information only in a region proposal stage, not in classification [31].

Therefore, we are suggesting utilizing a sequence of images in object classification, also with additional action labels. Figure 1 shows main difference between our approach and others, video object detection and object classification. There have been some approaches similar to ours, function-based object classification (FOC) and affordance learning. Historically, FOC has been connecting images and additional information gathered from additional physical devices like haptic devices [1] or 3D imagery [12,21]. Acquiring these datasets requires more prerequisites, so it is not easy to use freely. However, affordance learning, which has a similar stance with FOC, has been freer in dataset selection than FOC. They usually used video datasets to extract human interaction with objects [19]. Therefore, this shows the possibility that FOC utilizes video datasets to get action-related information. Affordance learning yet required laborious target tagging, which still narrows down the available data usable.

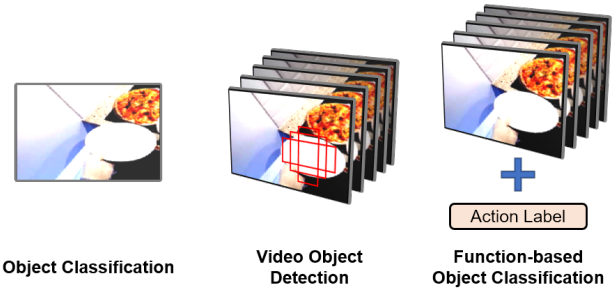


Figure 1. Difference between video object detection, function-based object classification, and object classification

In this paper, FOC without an explicit action label is proposed. This model has two networks, visuo-motor map (VMM) and visuo-motor classifier (VMC). VMM predicts action features from a sequence of image inputs, and VMC predicts object labels from a combination of images and action features. By utilizing both images and actions, this model solves typical classification difficulties like occlusion. In addition to the classification, we tried object detection which is the typical task with video datasets. To do so, we broaden the opportunity the model gets from the wider area, as the detection technique is useful for multi object environment.

There are 3 models proposed in this paper, Conv3D, S3D+attention, and Faster-RCNN with attention. All three models share the core structure (Figure 3), VMM and VMC. Images and attention maps are shown to prove the better performance the models get from utilizing the action information.

2. Related Work

2.1. Affordance Learning

According to Gibson [8], object affordance is potential "action possibility" defined in tools for functional characteristics. The notion "action possibility" drew attention in robotics, producing a lot of previous studies to detect object affordance, especially regarding an ability to grasp objects. Do et al [7] proposed end-to-end deep learning architecture to learn object and affordance detection simultaneously. Also, Ahn et al [20] designed the framework with two separate modules: Convolutional Neural Network (CNN) and Sense Condition Random Fields (CRF), and tested those at real-world scenes. By previous studies, the existence of relationship between object and affordance (or action) can be confirmed, and is likely to help the inference of each other. Our model does not explicitly infer the affordance of an object, but uses affordance-like action information when learning object classification.

2.2. Object Classification and Detection

Deep CNN has become a dominant method for object detection from single image [10, 11, 13, 17, 22, 24]. Among them, [22] presented a method for effectively inferring regions where objects exist using a region proposal network. [13] secures better performance than the existing residual network structure by applying residual transformation. Our proposed method is built upon ResNet-50 [13] which is a popular deep CNN algorithm. However, object detection in videos is difficult no matter what architecture is used, because the dataset has complexity and variation such as out-of-focus, occlusion, motion blur, and rare appearance of the object, which cause performance decay. Therefore, the main task of video object detection has become to find how to enhance the detection performance of every frame.

Many studies attempted to increase the detection performance by aggregating information dwelling in between the low-quality image frames. Tubelets with convolutional neural networks(T-CNN) [14, 15] assimilate temporal and contextual information which are locally propagated via nearby frames. Flow Guided Feature Aggregation(FGFA) method [32] and Deep Feature Flow [33] utilize local aggregation of feature maps across adjacent frames. Our model has similar properties in terms of aggregating compressed image sequences and action information to select suitable object labels.

2.3. Action Recognition

Action recognition has been an important topic in video understanding, since ground-truth action labels in video are hard to obtain. Our model also has to learn action recognition for the further object classification tasks. Action recognition tends to utilize long-range time information like follows. Sudhakaran et al [23] presented an action recognition framework called EgoACO that utilizes class activation pooling and Long Short-term Attention. Temporal Segment Network(TSN) provides action recognition based on video framework using long-range temporal architecture [25, 26]. Zhao et al [30] proposed temporal action detection framework, called Structured Segment Network(SSN) which constructs temporal pyramid using activity and completeness classifiers. However, there was a similar approach to ours, Yang et al [28] proposed a weakly supervised object detection network considering not only the object class label but also the action label of the data. They used a keypoint detection network, which is hard to be incorporated into the egocentric video dataset.

(a) Object classification results

(b) Action classification results

(c) Attention map (Grad-CAM)

Figure 2. Classification results. The predicted label is shown with the confidence score and the ground-truth label is shown inside the parenthesis. Red label indicates the wrong prediction and the green indicates the right. The attention map shows the focused area for action prediction.

Algorithm 1 VMMC Detection module for one image

Require: $I^k, [o_{x1}, o_{y1}, o_{x2}, o_{y2}]_N^k, [o]_N^k$

Ensure: I^k ▷ Get image feature from VGG16, assume N: total number of bbox, M: hand/obj bbox number

$I^k = ROI(RPN(I^k))$

$\hat{I}^k, \hat{A}^k, = VMM_{Boxhead}(I^k)$

Function$\{VMC_{Boxpred}\}$

$[\hat{h}_{x1}, \hat{h}_{y1}, \hat{h}_{x2}, \hat{h}_{y2}]_M^k, [\hat{a}]_M^k = HandAction(\hat{A}^k)$

$\hat{O}^k = ObjPred_{attn1}(\hat{A}^k, \hat{I}^k)$

$[\hat{o}_{x1}, \hat{o}_{y1}, \hat{o}_{x2}, \hat{o}_{y2}]_M^k = ObjPred_{attn2}(\hat{O}^k, [\hat{h}_{x1}, \hat{h}_{y1}, \hat{h}_{x2}, \hat{h}_{y2}]_M^k)$

EndFunction

$[\hat{o}_{x1}, \hat{o}_{y1}, \hat{o}_{x2}, \hat{o}_{y2}]_N^k = Concat([\hat{h}_{x1}, \hat{h}_{y1}, \hat{h}_{x2}, \hat{h}_{y2}]_M^k, [\hat{o}_{x1}, \hat{o}_{y1}, \hat{o}_{x2}, \hat{o}_{y2}]_M^k)$

$[\hat{o}]_N^k = Concat([\hat{o}]_M^k, [\hat{a}]_M^k)$

3. Method

3.1. Preliminary

A sequence of images $\{I_t\}_{t=1}^T$ in a video has its pairs, sequences of action labels $\{a_t\}_{t=1}^T$ and object labels $\{o_t\}_{t=1}^T$. As one kind of action continues across a few frames, an image sequence can be grouped into several subsets under the action criteria. Assume K distinct actions present in a video, then sequences of images, actions, and objects can be divided as follows.

$$\{I_t\}_{t=1}^T = [\{I_t^1\}_{1s}^{1e}, ..., \{I_t^k\}_{ks}^{ke}, ..., \{I_t^K\}_{Ks}^{Ke}] \quad (1)$$

$$\{a_t\}_{t=1}^T = [a^1, ..., a^k, ..., a^K] \quad (2)$$

$$\{o_t\}_{t=1}^T = [o^1, ..., o^k, ..., o^K] \quad (3)$$

a^k and o^k is the action label and object label paired with the sequence of images $\{I_t^k\}_{ks}^{ke}$, and A^k is the action feature inferred from $\{I_t^k\}_{ks}^{ke}$, with the start time ks and the end time ke.

In the detection task, each image has its own hand-object bounding boxes. Assume an image I_t has N object bounding boxes, then a target of the image is a dictionary with 2 keys, boxes $\{[o_{x1}, o_{y1}, o_{x2}, o_{y2}]_N\}_t$ and labels $\{[o]_N\}_t$. Also, an image I_t has M hand bounding boxes, then a target of the image is a dictionary with 2 keys, boxes $\{[h_{x1}, h_{y1}, h_{x2}, h_{y2}]_M\}_t$ and labels $\{[a]_M\}_t$. There are hand state notations (left and right) in the dataset, but action labels are used instead as the state information is not needed. M is less than or equal to 2, so in practice, hand bounding boxes and object bounding boxes are concatenated before moving into the Faster-RCNN module. After concatenation, the notation is unified.

In this study, the pre-defined number m of frames are randomly chosen from $\{I_t^k\}_{ks}^{ke}$ to be an input to the model, and the notation of that would be in short, $\{I^k\}_m$ afterwards.

3.2. Visuo-Motor Map and Visuo-Motor Classifier

Visuo-Motor Map(VMM) and Visuo-Motor Classifier(VMC) are reciprocally developed to successfully classify or detect objects in videos, with the aid of action inferenced. They have 3 forms in total, gradually complexified to deal with the problems arose in the previous form. First, starting from the basics, VMM and VMC are mainly defined with Conv2D and Conv3D, dealing with the sequence of images using Conv3D. After that, to focus more on object classification, S3D combined with the attention is proposed. The third model is developed for the detection task, using the core structure of Faster-RCNN.

3.2.1 Object Classification

VMM generally, is trained to inference the relationship between the sequence of images and the action label.

$$\{\hat{I}^k\}_m, \hat{A}^k, \hat{a}^k = VMM(\{I^k\}_m) \quad (4)$$

After that, VMC is trained to inference the object label from the sequence of images and the action feature from VMM, as shown in the equation 5. However, the second model, S3D+attention model is different from the first model, in the sense that it uses attention. Also, instead of image sequences, it uses one ResNet image feature as key and value in the attention, which can improve the performance since object classification is usually done on a static image. The equation 6 conceptually shows how attention works as VMC.

$$\hat{o}^k = VMC(\{\hat{I}^k\}_m, \hat{A}^k) \quad (5)$$

As shown in Figure 4, the flow between Conv3D model and the S3D+attention model is similar, yet different in terms of the layers numbers used in each inference. As the pre-trained S3D feature can represent an enough action information, S3D+attention model more focused on improving

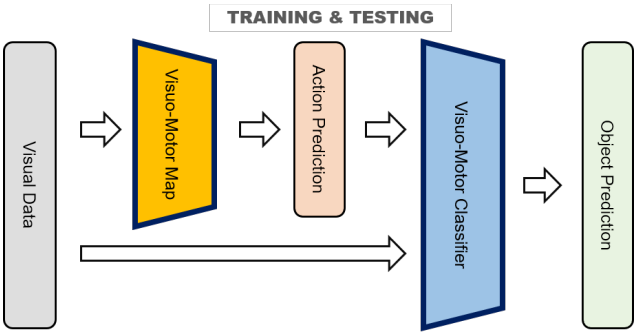


Figure 3. An overview of our entire model, VMM and VMC.

object classification performance.

$$\hat{o}^k = VMC_{Attn}(\hat{A}^k, \{I^k\}_m, \{I^k\}_m) \quad (6)$$

VMM and VMC in the classification task are trained together, but they can be separated and analysed individually. To confirm the effectiveness of action conjecture, the visualization of attention, Grad-CAM [9] is used.

3.2.2 Object Detection

Different from the object classification, the object detection uses more human labor required information, thus provides more action related information. So, the object detection gets more benefit when utilizing action information.

Our model is based on the Faster-RCNN model provided in the torchvision. The box head is transformed to work as VMM and the modified box predictor with 2 attention network works as VMC. Refer to Figure 4c, the box head outputs pre-object features and action features. Both features are classified and predicted as boxes in VMC, the box predictor. The box predictor uses two attention module, one for object feature and one for object bounding box predictor, refer to the algorithm 1 for more details.

3.3. Baseline Algorithm for Performance Evaluation

3.3.1 Object Classification

Based on ResNet [13], a modified network using grouped convolution named ResNeXt is introduced in [27]. The bottleneck layer of ResNet consists of three sequential convolution networks with a residual channel. ResNeXt has similar architecture, but the convolution layers are modified. Unlike ResNet, input channels of the the second convolution layer are divided, convoluted, and then concatenated. Utilizing this structure enhanced classification task performance without increase in model complexity. For object classification task, this network was chosen as a baseline. The network outputs single classification label class for a given input image.

3.3.2 Object Detection

An open source algorithm named MMDetection [3] was adopted as a baseline network to compare the object detection performance of the proposed framework and the benchmark. The two stage detector structure consists of Backbone, Neck, Dense-Head and ROI-Head networks. Through configuration settings, the detector can be customized based on various pre-trained backbone networks including Fast R-CNN [10], Faster R-CNN [22], and R-FCN [5]. The network outputs bounding box, confidence score, and classification label for the detected objects in each RGB channel input image. For our baseline configuration, Faster R-CNN backbone was adopted.

4. Experiments

4.1. Dataset and Evaluation Setup

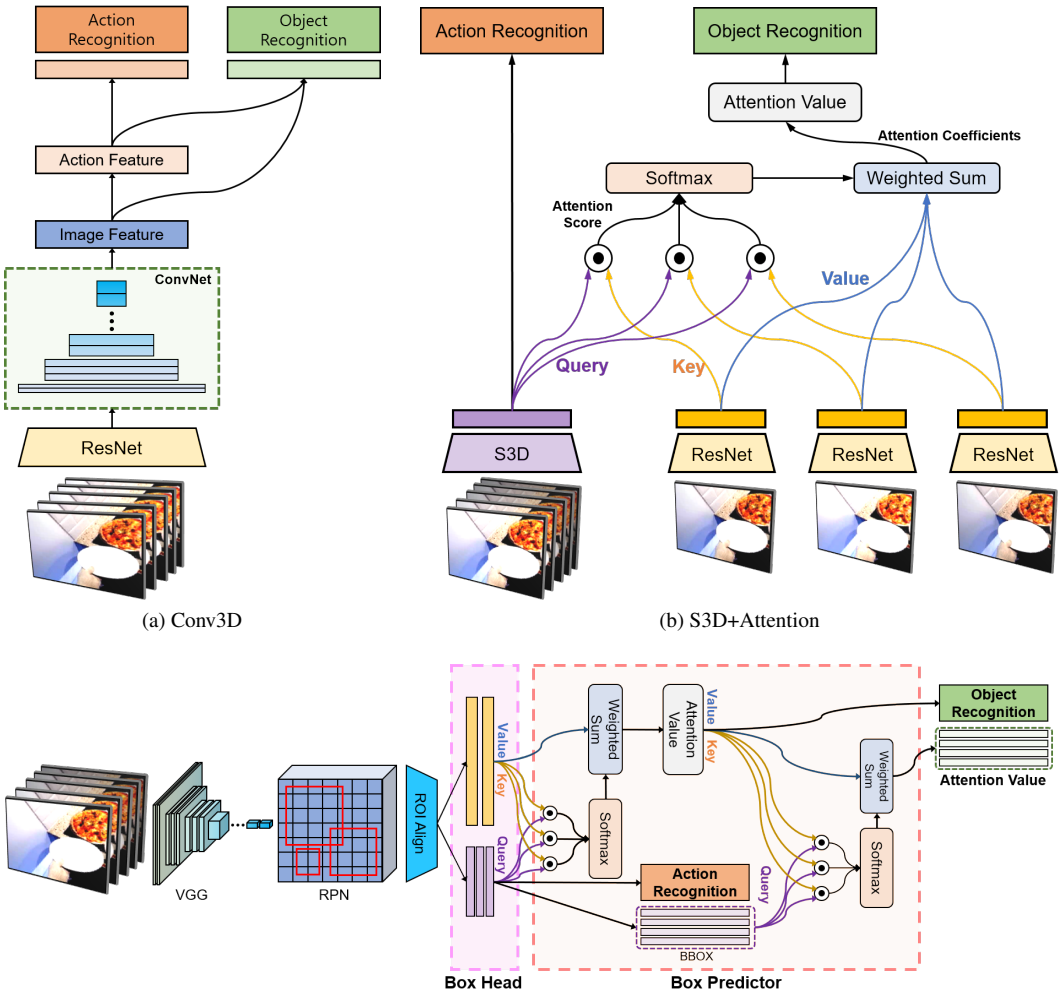
In order to train and validate the VMM and VMC models, dataset containing various objects and human actions was needed. In this study, the Epic-Kitchens [6] dataset which satisfies these requirements was utilized. Each subset video of the dataset consists of RGB image frames recorded from egocentric, first-person view. The description of the scene, verb and noun labels for each action and object pair, and start / end frame information for each action are provided from the performer of the video. The bounding boxes of the narrated objects are suggested by Faster-RCNN and annotated by Amazon Mechanical Turk (AMT) workers.

4.2. Implementation Details

Our main networks are VMM and VMC. We aim to expand the architecture proposed by C. Castellini et al [2]. VMM and VMC were proposed to be implemented separately at first, but as our dataset is different from the original paper, we could combine two networks. However, we gave separate notation for two networks for the direct comparison.

Conv2D and 3D. The backbone network of VMM is the modified ResNet-50 [13] proposed by Tushar et al [19]. By using modified ResNet-50, the sequence of images can be pre-processed together, outputs an integrated image feature. After the backbone, the images passes through Conv3d layers to produce image and action features, then Conv2d layers to produce classification logits from the features. The fully-connected layers were not used for better visualization of Grad-CAM attention. The dimension reduction of input feature inside the Conv2d layer is required due to absence of the fully connected layer, so it is done with the custom squeeze functional layer. Batch normalization and the ReLU/Leaky ReLU activation layers are applied by default.

S3D Separable 3D CNN (S3D) [29] has multiple layers of 2D+1D Inception blocks, which easily learn spatial



(a) Conv3D

(b) S3D+Attention

(c) Faster R-CNN

Figure 4. Network structure

and temporal information. Therefore, S3D can work as useful action feature extractor. In this paper, S3D pretrained on Kinetics-400 dataset is used [18], except for the last layer which is used for the action classification. For action classification, the additional processing with Conv3D layer is done. For object classification, each image in the sequence is separately processed with pre-trained ResNet50. The images act as a key-value pair in the attention, and the action features from S3D act as a query. A linear layer is used to extract object classification score from the resultant attention value.

Scaled Dot product Attention Scaled dot product attention is the simplest form of attention. Query, key, and value should have the same batch size and the same length either in the width or height. The attention score is calculated within the dot product between query and key, then the attention coefficient is calculated from the weighted sum of the value, weighted by the attention score. More details can be found in Figure 4. Two models among 3 models suggested in this paper use this attention, S3D+attention model in classification task and the detection model.

Faster-RCNN. Faster-RCNN [22] is a two-stage object detector, which has a region proposal network and the box predictor network. We used VGG16 pre-trained network as a backbone, then the default region proposal network the torchvision provides. After multi-scale ROI align, the box head gets convolution features. The default box head usually just flatten the input and reduce the dimension by fully connected layers. However, to combine information throughout the sequence of images, the convolution operation preceded before fully connected layer, outputting action feature.

The box predictor utilized two attention modules. First, it predicted action class and the hand bounding boxes with

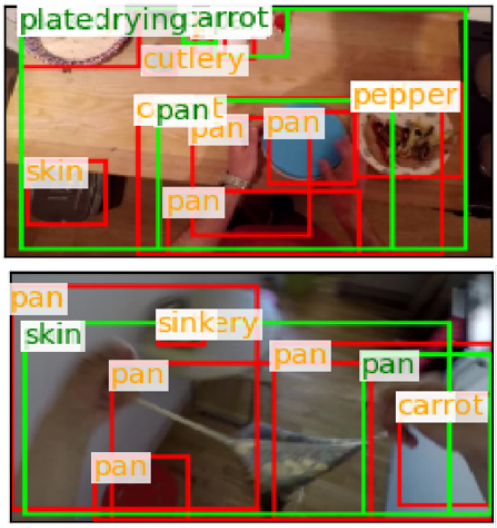


Figure 5. Detection results of proposed framework. Bounding boxes and names denoted in green are the ground truth data and the prediction results are displayed in red color.

fully connected layers. Then, it used action feature and image features from box head, as a query and the key-values respectively. For the object classification, the resultant object feature from the first attention passed through the fully connected layer. Lastly, the hand bounding boxes contextualized the object features to make the object bounding boxes.

Dataset loader. For a dataset loader, we randomly select one image frame between the start and stop frame of a specific action capture for the Epic-Kitchen dataset. Then, the shared action and object labels of the selected image frames are used. Also, input images are resized to 256 pixels and transformed using random crop and random horizontal flip. The images had to be cropped into 224 size in S3D+attention model due to memory deficiency. In the detection task, no other transforms than the normalization is used, to conserve the ratio and the size for the bounding box coordinates.

The bounding boxes provided in the dataset should be read with protobuf library which is provided. However, the hand bounding boxes and the object bounding boxes could not be opened at the same time, due to the library import error. In consequence, they had to be pre-processed in advance to be used in the detection model.

Our model is trained along with an Adam optimizer with 0.8 beta1 value with 0.0001 learning rate. In Conv3D model, the batch size was 30, and 4 frames were concatenated to be an image sequence input. In S3D model, the batch size was 2, and 5 concatenated image frames were used. In the detection task, only 2 images could be processed at the same time due to the limitation in GPU memory.

4.3. Main Results

Classification. The final test accuracy is 26% for the action classification task and 4.7% for the object classification. Two models (Conv3D and S3D+attention) showed similar results. The case of action classification seems to be relatively accurate and learned, but there was a data imbalance problem with too many 'take' and 'put' labels (Figure 2b). In object classification, ground truth objects were often not visible in the sample image due to frame escape or illusion, occlusion, etc (Figure 2a). This can occur from the random selection of one image between start and stop frame. To improve object classification performance, we refer to the object-action correspondence frame list provided in the dataset. However, the list was just a list of evenly sampled frames, sampled every 30 frames. The substitution of sampling strategy was not very effective. In order to investigate the reason of poor classification, the last layer of the model is visualized with Grad-CAM (Figure 2c) to verify whether the model focuses on the right place for the task. The attention map highlights the area where the action is assumed to be occurred, but the attention map is very broad and not focused. The high gap between object and action classification accuracy is due to the nature of the dataset, but also due to the appearance of multiple objects in an image. The object classifier gets distracted by the multiple objects.

Detection. The final detection results are shown in Figure 5. Ground truth multiple object bounding boxes are given in the dataset, but the quality of ground truth is very disappointing. Hands in the dataset are prone to be classified as carrot, because the ground truth labelled human arms as carrots multiple times. Also, the bounding boxes tend to be bigger than the actual object. The inference result has low correspondence with the ground truth, but the proposed region seems to be accurate enough.

4.4. Baseline Classification and Detection Results

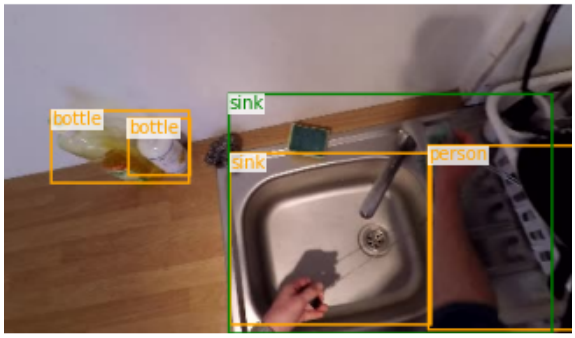
ResNeXt Initialized ResNeXt network model was trained and tested for object classification task using Epic-Kitchens [6] dataset. The cardinality of the bottleneck layer was set to 32, and classification loss for each mini batch was defined as the cross entropy loss. Training loss decreased during the training process, but the validation loss had no improvement. The model was not able to learn the ground truth object label from the given dataset, with a test accuracy of under 3%.

The reason for the low performance index is due to the unique characteristics of the Epic-Kitchens dataset. In the image classification datasets including STL-10 [4] and CIFAR-100 [16], there is one main object in the given single image. In Epic-Kitchens, however, various objects as well

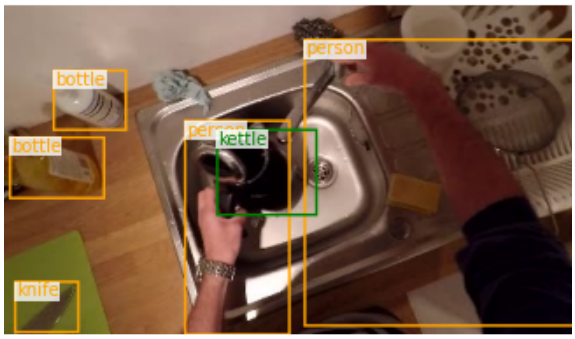
as object related to the participant's action exist together in the image taken at a relatively wide angle. Although the object of interest exists in the image and is correctly labeled as the ground truth, it is not easy for the vanilla ResNeXt network to find the main noun in the given image and to classify the entire image based on the label of the noun. Therefore, it can be concluded that in order to find the main noun related to the motion and perform the image classification task according to this, it is necessary to spatially attend to the image frame and / or infer to the participant's action as demonstrated in the proposed object action recognition framework.

MMDetection To test the object recognition performance of the baseline toolbox [3], images from the Epic-Kitchens [6] dataset were applied to the backbone network provided in the source code. Figure 6 shows some frames among the results of performed object detection task for an image set input with confidence score threshold set to 0.5. The classified label of the objects are displayed on top of the bounding boxes. The detection results of MMDetection suggest that the network is able to recognize most of the major objects present in the given image frame. However, there were some limitations regarding hand object detection and labeling. Since the region proposal network of the backbone layer sets equal importance to each pixel, hand held objects were misclassified as 'person', combined with the participant's hands and arms.

Moreover, at most two labeled action noun objects and their bounding boxes are present in each frame of Epic-Kitchens dataset ground truth label data. This means that only one or two objects used in the participant's action or the ones ready for use are labeled. Bounding boxes predicted by MMDetection, however, are generated from the region proposal network without being supervised to concentrate more on specific image regions where hand-object actions take place. Although various objects in the given scene are well detected by the network, ground truth object detection accuracy measures (intersection of union, mean average precision) marked significantly low scores. The inference results of baseline algorithm support our assumption that taking not only the object pixels itself but also the related action into account is necessary for enhancing detection performance especially for objects directly involved in the participant's action.



(a) Correct object class labeling results with bounding box intersection of union accuracy of 0.446.



(b) Noun detection fail case. Hand held object is classified as a person, together with the participant's hands.

Figure 6. Multiple object detection and classification results using MMDetection. Bounding boxes and names denoted in green are the ground truth data and the prediction results are displayed in orange color.

5. Conclusion

This paper introduced an object action recognition framework using visuo-motor map and visuo-motor classifier structure. Convolution layers and S3D layers were adopted for object and action classification tasks, and object detection was performed based on Faster R-CNN network. S3D network was introduced to overcome the limitations of Conv3D, but the results confirmed that only slight improvement was gained regarding the classification performance. The contribution of the research stands out when comparing the object recognition results of the proposed framework with the baseline algorithm results. Unlike other datasets mentioned in section 4, the images in the Epic-Kitchens dataset contain multiple surrounding objects in addition to the ground truth object used directly for hand-object action. Therefore, it is difficult to discriminate the main noun class only with inference to the single image frame without action recognition or attention, and this was demonstrated by deteriorated classification and detection performance based on ResNeXt and MMDetection, respectively.

Future works aim at resolving major implementation issues caused by the lack of resources. Increase in batch size and the number of epochs, applying multi-head self attention layer may enhance the performance of the classifier. Also, training and testing on various datasets are expected to generalize the network capabilities, allowing the proposed framework to perform well on diverse scenes and actions.

References

[1] C. Castellini, T. Tommasi, N. Noceti, F. Odone, and B. Caputo. Using object affordances to improve object recognition. *IEEE Transactions on Autonomous Mental Development*, 3(3):207–215, 2011. 1

[2] Claudio Castellini, Tatiana Tommasi, Nicoletta Noceti, Francesca Odone, and Barbara Caputo. Using object affordances to improve object recognition. *IEEE transactions on autonomous mental development*, 3(3):207–215, 2011. 4

[3] Kai Chen, Jiaqi Wang, Jiangmiao Pang, Yuhang Cao, Yu Xiong, Xiaoxiao Li, Shuyang Sun, Wansen Feng, Ziwei Liu, Jiarui Xu, et al. Mmdetection: Open mmlab detection toolbox and benchmark. *arXiv preprint arXiv:1906.07155*, 2019. 4, 7

[4] Adam Coates, Andrew Ng, and Honglak Lee. An analysis of single-layer networks in unsupervised feature learning. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pages 215–223. JMLR Workshop and Conference Proceedings, 2011. 6

[5] Jifeng Dai, Yi Li, Kaiming He, and Jian Sun. R-fcn: Object detection via region-based fully convolutional networks. In *Advances in neural information processing systems*, pages 379–387, 2016. 4

[6] 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 *European Conference on Computer Vision (ECCV)*, 2018. 4, 6, 7

[7] Thanh-Toan Do, Anh Nguyen, and Ian Reid. Affordancenet: An end-to-end deep learning approach for object affordance detection. In *2018 IEEE international conference on robotics and automation (ICRA)*, pages 5882–5889. IEEE, 2018. 2

[8] James J Gibson. The theory of affordances. the ecological approach to visual perception, 1979. 2

[9] Jacob Gildenblat and contributors. Pytorch library for cam methods. https://github.com/jacobgil/pytorch-grad-cam, 2021. 4

[10] Ross Girshick. Fast r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 1440–1448, 2015. 2, 4

[11] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 580–587, 2014. 2

[12] Helmut Grabner, Juergen Gall, and Luc Van Gool. What makes a chair a chair? In *CVPR 2011*, pages 1529–1536, 2011. 1

[13] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 2, 4

[14] Kai Kang, Hongsheng Li, Junjie Yan, Xingyu Zeng, Bin Yang, Tong Xiao, Cong Zhang, Zhe Wang, Ruohui Wang, Xiaogang Wang, et al. T-cnn: Tubelets with convolutional neural networks for object detection from videos. *IEEE Transactions on Circuits and Systems for Video Technology*, 28(10):2896–2907, 2017. 2

[15] Kai Kang, Wanli Ouyang, Hongsheng Li, and Xiaogang Wang. Object detection from video tubelets with convolutional neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 817–825, 2016. 2

[16] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009. 6

[17] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25:1097–1105, 2012. 2

[18] Kyle Min and Jason J Corso. Tased-net: Temporally-aggregating spatial encoder-decoder network for video saliency detection. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2394–2403, 2019. 5

[19] Tushar Nagarajan, Christoph Feichtenhofer, and Kristen Grauman. Grounded human-object interaction hotspots from video, 2019. 1, 4

[20] Anh Nguyen, Dimitrios Kanoulas, Darwin G Caldwell, and Nikos G Tsagarakis. Object-based affordances detection with convolutional neural networks and dense conditional random fields. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 5908–5915. IEEE, 2017. 2

[21] Michael Pechuk, Octavian Soldea, and Ehud Rivlin. Learning function based object classification from 3d imagery. *CVIU*, April 2007. 1

[22] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems*, 28:91–99, 2015. 2, 4, 5

[23] Swathikiran Sudhakaran, Sergio Escalera, and Oswald Lanz. Learning to recognize actions on objects in egocentric video with attention dictionaries. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021. 2

[24] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9, 2015. 2

[25] Limin Wang, Yuanjun Xiong, Zhe Wang, Yu Qiao, Dahua Lin, Xiaoou Tang, and Luc Van Gool. Temporal segment networks: Towards good practices for deep action recognition. In *European conference on computer vision*, pages 20–36. Springer, 2016. 2

[26] Limin Wang, Yuanjun Xiong, Zhe Wang, Yu Qiao, Dahua Lin, Xiaoou Tang, and Luc Van Gool. Temporal segment networks for action recognition in videos. *IEEE transactions on pattern analysis and machine intelligence*, 41(11):2740–2755, 2018. 2

[27] Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1492–1500, 2017. 4

[28] Zhenheng Yang, Dhruv Mahajan, Deepti Ghadiyaram, Ram Nevatia, and Vignesh Ramanathan. Activity driven weakly supervised object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2917–2926, 2019. 2

[29] Da Zhang, Xiyang Dai, Xin Wang, and Yuan-Fang Wang. S3d: Single shot multi-span detector via fully 3d convolutional networks, 2018. 4

[30] Yue Zhao, Yuanjun Xiong, Limin Wang, Zhirong Wu, Xiaoou Tang, and Dahua Lin. Temporal action detection with structured segment networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2914–2923, 2017. 2

[31] Haidi Zhu, Haoran Wei, Baoqing Li, Xiaobing Yuan, and Nasser Kehtarnavaz. A review of video object detection: Datasets, metrics and methods. *Applied Sciences*, 10(21), 2020. 1

[32] Xizhou Zhu, Yujie Wang, Jifeng Dai, Lu Yuan, and Yichen Wei. Flow-guided feature aggregation for video object detection. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 408–417, 2017. 2

[33] Xizhou Zhu, Yuwen Xiong, Jifeng Dai, Lu Yuan, and Yichen Wei. Deep feature flow for video recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017. 2