A Video Object Detection Method of ECNet Based on Frame Difference and Grid Cell Confidence

Shunsuke Akamatsu
School of FSE
Waseda University
Tokyo, Japan
s.akamatsu@akane.waseda.jp

Kei Iino
Graduate School of FSE
Waseda University
Tokyo, Japan
iinokei@akane.waseda.jp

Hiroshi Watanabe
Graduate School of FSE
Waseda University
Tokyo, Japan
hiroshi.watanabe@waseda.jp

Shohei Enomoto
NTT Software Innovation Center
Tokyo, Japan
shohei.enomoto@ntt.com

Xu Shi
NTT Software Innovation Center
Tokyo, Japan
kyoku.shi@ntt.com

Akira Sakamoto
NTT Software Innovation Center
Tokyo, Japan
akira.sakamoto@ntt.com

Takeharu Eda
NTT Software Innovation Center
Tokyo, Japan
takeharu.eda@ntt.com

***Abstract*—In recent years, real-time video processing has advanced thanks to dramatic improvements in object detection algorithms. This has increased the demand for video object detection by various edge devices. An example is the use case of analyzing video captured by CCTV cameras. Therefore, there is a need to perform object detection on the edge within a limited time, but complex object detection models cannot be deployed on the edge with limited resources. For the task of image classification, the Edge-Cloud Net (ECNet) is proposed to achieve both transmission cost and accuracy by sharing processing between the edge and cloud sides using an offload controller. In this study, we have applied ECNet to video object detection. We propose a method to implement a frame differencing mechanism before edge inference and use the confidence value of each grid cell as an offload criterion for cloud transmission. Furthermore, depending on the confidence value, the image is masked to reduce the amount of information sent to the cloud. Our method can reduce the amount of transmission while maintaining the accuracy of video object detection, especially when the data size is small.**

Keywords— Edge-Cloud Net, YOLOv3, YOLOv3-tiny Frame Differences, Confidence value.

I. Introduction

In recent years, object detection algorithms have been improved to achieve very high speed and high accuracy. There is a need for object detection on the edge to improve response time, but object detection models are complex and cannot be implemented on the edge which has limited resources. Therefore, a system that combines processing on the edge and cloud sides is required [1, 2]. The system that cooperates with processing on the edge side and the cloud side is as shown in Fig. 1. It is proposed an Edge-Cloud Network system [3] in which a lightweight model is placed on the edge side and a highly accurate model is placed on the cloud side to achieve a balance between accuracy and transmission volume. They use the YOLOv3 [4] backbone, Darknet19 and Darknet53, to build an Edge-Cloud Net in their study, employing class entropy as an offloading criterion to balance the volume and accuracy of edge and cloud side transmissions. However, it is only effective for image classification and has not demonstrated its usefulness for object detection in videos, which is assumed to be an actual use case such as implementing CCTV cameras. Hence, we propose an object detection approach for videos using ECNet structure.

In this approach, a lightweight YOLOv3-tiny [5] is deployed on the edge side and YOLOv3 is implemented on the cloud side to create a cooperative structure between the edge and the cloud. Furthermore, depending on the value of the frame difference, the result of the previous frame can be applied to the current frame to reduce the amount of inference. Offload criteria are determined by the grid cell's confidence value in the edge inference result, and the image is masked according to this value before transmission. This allows for maintaining accuracy while reducing the amount of transmission.

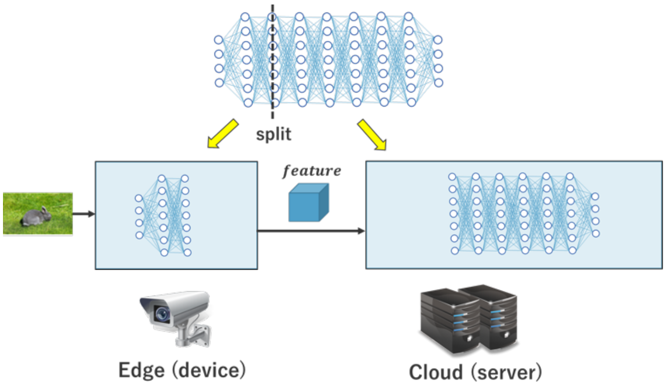


Fig. 1. The concept of collaborative intelligence network proposed in our previous study [2].

II. Related work

A. Edge-Cloud Net

Edge-Cloud Net [3] is a method to put a light but low accurate model on the edge side and a heavy but high accurate model on the cloud side for inference. The offload controller determines whether the output from the edge side is sent to the cloud side or not. By combining this mechanism with quantization technology during transmission, accuracy can be maintained while reducing the amount of transmission.

B. YOLOv3

YOLOv3 [4] is an object detection algorithm that increases the number of layers in the network and improves detection accuracy compared to the previous version of YOLO algorithm [6]. Compared to the previous version YOLOv2 [7], YOLOv3 not only improves accuracy by changing the network layers from 19 to 53 but also adopts a structure similar to a feature pyramid network [8] to enable detection at multiple scales. In YOLOv3, three grid cell sizes (13×13, 26×26, and 52×52) are provided to enable detection of various-sized objects. In YOLOv3-tiny, the number of convolution layers is reduced to lessen its weight, and two grid cell sizes (13×13 and 26×26) are used to detect objects.

979-8-3503-4017-4/23/$31.00 ©2023 IEEE

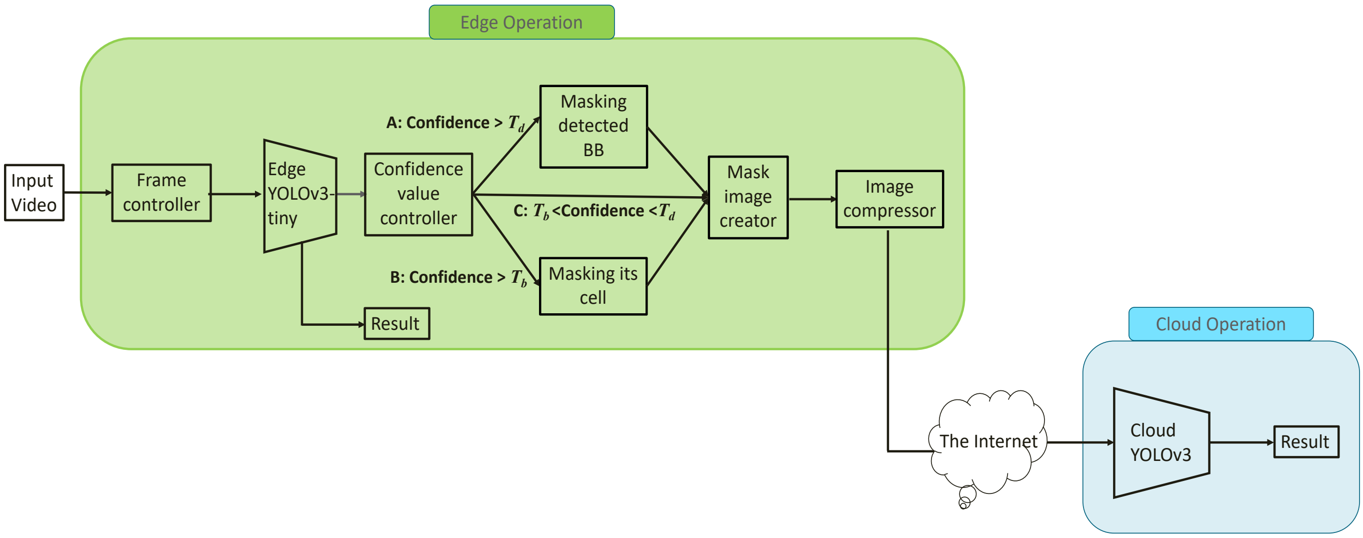


Fig. 2. Entire system of our proposed method.

III. Proposed Method

We use YOLOv3-tiny and YOLOv3 for detection within the ECNet structure. We have implemented an architecture to measure frame difference and grid cell confidence to generate a masked image, and an image compression mechanism. It is then transmitted to the cloud via the Internet and inferred by YOLOv3. The entire structure is shown in Fig. 2.

A. Measure of frame difference

For the input video, the absolute value of the difference in pixel values is obtained for each frame. For those values that are within the threshold (T_f), it does not perform inference with edge. This is because the result from the previous frame is used as the result for the current frame since when the difference is small, there is a high probability that the object in the current frame is the same as the object in the previous frame. This enables fewer inferences to be made and thus reduces the amount of transmission.

B. Measure of grid cell confidence and masked image

Based on the inference results at the edge, confidence value for each grid cell is used as the offloading criterion. The confidence value ranges from 0 to 1 and is indicated by the following equation [9].

$$p(class_i) = p(object) \times p(class_i \mid object) \tag{1}$$

$$confidence = max(p(class_i)) \tag{2}$$

In this formula, i represents the class type of the object in the cell, $p(object)$ is the probability that the cell has an object, and $p(class_i)$ is the probability that the object is of class i for that object. We introduce T_d and T_b as the threshold parameters. T_d is used as a threshold to determine if an object is detected in that grid cell or not, and T_b is used as the threshold that defines whether the cell is background or not.

When the confidence value is high, i.e., when confidence > T_d (A), the model on the edge side has a high probability of capturing an object, so that cell uses the results of the model on the edge side and masks the detected bounding box.

When the confidence value is low, i.e., when confidence < T_b (B), the edge-side model determines that the cell is likely to be a background part in the image, thus, the cell uses the results of the edge model and masks the detected background part.

When neither of these is the case, that is, when the value of confidence satisfies T_b < confidence < T_d (C), the cell is sent to the cloud as it is. This creates an image in which the bounding box detected by the edge-side model and the background portion are masked.

The algorithm for processing each grid cell by the value of confidence is shown in Fig. 3. An example of the process according to the confidence level of each grid cell is shown in Fig. 4. (A), (B), and (C) in this figure correspond to the processing of the confidence value controller.

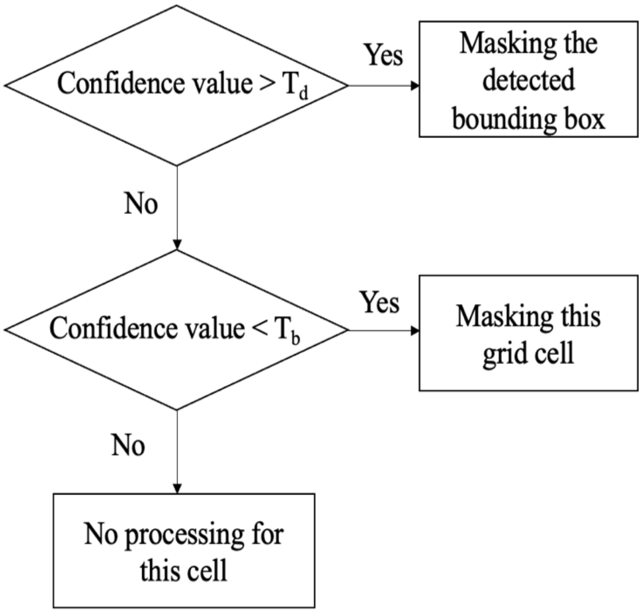


Fig. 3. Algorithm for processing to each grid cell.

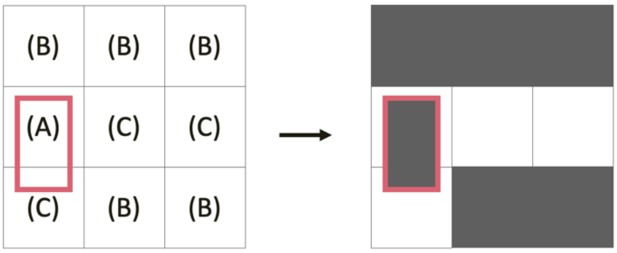


Fig. 4. An example of of the process according to the confidence level of each grid cell. In this case, an object is detected in the cell in column 1, row 2.

C. *Image compression*

The image masked by the inference result on the edge side is compressed in JPEG [10] format according to the compression ratio value. Compression reduces the amount of data sent to the cloud side. The compression ratio is calculated by the following formula [2].

$$Compression\ ratio = I_{after} / I_{before} \tag{3}$$

In this formula, I represents the data size of the image, where I_{before} is the data size before compression, and I_{after} is the data size after compression. The compression ratio is calculated by the ratio of before and after processing, and the smaller the value, the more the amount of data is reduced.

IV. EXPERIMENT

A. *Evaluation method*

We selected the fixed viewpoint data from the MOT17 dataset [11] to serve as input for our evaluation. The reason for limiting the video from a fixed camera is to ensure that frame difference can be measured correctly. The case where all detection is performed by YOLOv3-tiny on the edge side is used as the reference value, and the case where only image compression is performed on the edge side and detection by YOLOv3 on the cloud side is used for comparison.

B. *Results*

The position of the mask on the image changes depending on the confidence value. Fig. 5 shows the data of a frame in MOT17 with bounding box mask (A), background mask (B), and both masks applied. In addition, the results of applying masks to various types of image frames are shown in Fig. 6. Furthermore, when the transmitted image is compressed using JPEG, the detection accuracy of YOLOv3 on the cloud side varies with the compression ratio, as shown in Fig. 7. In this figure, average precision (AP) is used to evaluate accuracy, and the AP loss rate is shown by the following equation [2].

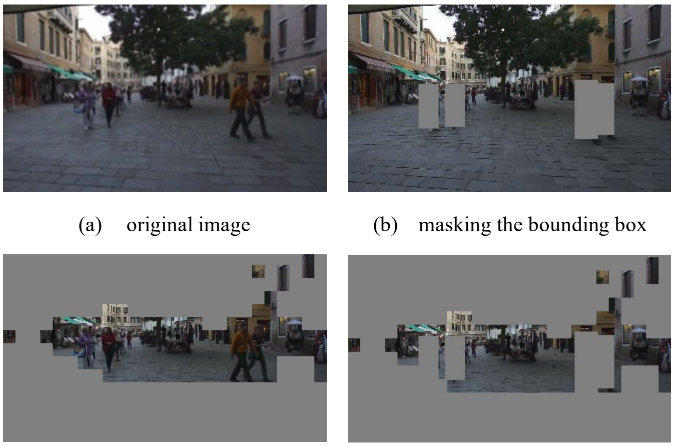
$$AP\ loss\ rate = 10\log_{10}\{(AP_{before} — AP_{after}) / AP_{before}\} \tag{4}$$

In this formula, AP_{before} and AP_{after} represent the accuracy of object detection before and after compression, respectively.

For each parameter, T_f adopts a smaller value because the use of results for all frames has a significant impact on accuracy. T_d and T_b are set empirically and through experimentation shown in Table 1. Taking these into account, we set the thresholds as T_f = 0.003, T_d = 0.65, T_b = 0.001 (Ours1) and the thresholds as T_f = 0.0007, T_d = 0.5, T_b = 0.001 (Ours2).

TABLE I. VALUE OF EACH THRESHOLD SET

	Threshold value		
	T_f	T_d	T_b
Ours1	0.003	0.65	0.001
Ours2	0.0007	0.5	0.001



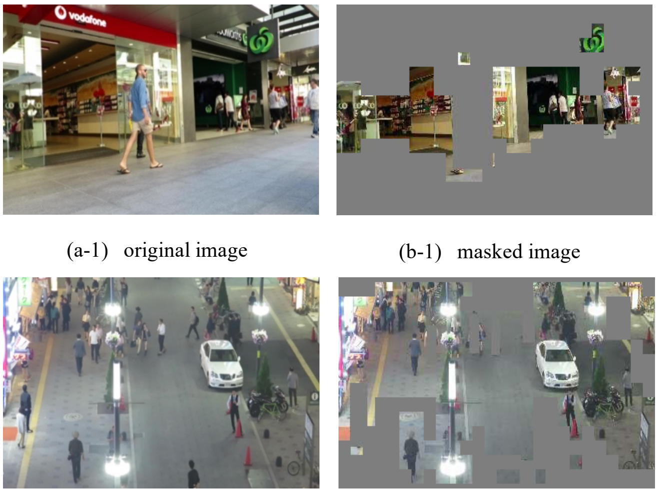
(a) original image

(b) masking the bounding box

(c) masking the background

(d) masking both parts

Fig. 5. Result from proposed method of masking the data. (upper left: original image, upper right: masking the bounding box(A), lower left: masking the background(B), lower right: masking both)



(a-1) original image

(b-1) masked image

(a-2) original image

(b-2) masked image

Fig. 6. Results of comparison between the original image and the masked image by confidence value in several scenes.

Fig. 8 shows the relationship between the data size transmitted to the cloud side (bpp) and the average precision (AP) (IoU=0.5). Our proposed method is especially effective when the amount of data sent to the cloud side is small while accuracy is required.

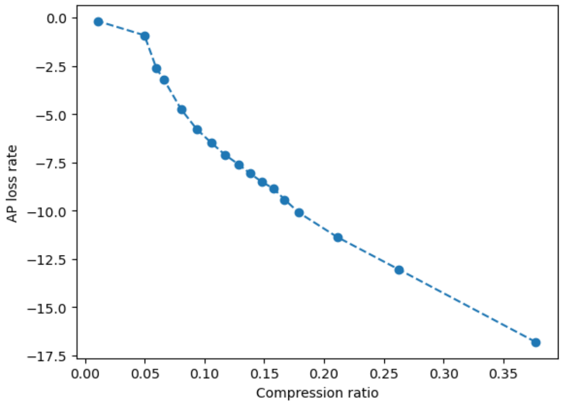


Fig. 7. Effect on AP of object detection by YOLOv3 on the cloud side when compression ratio is changed.

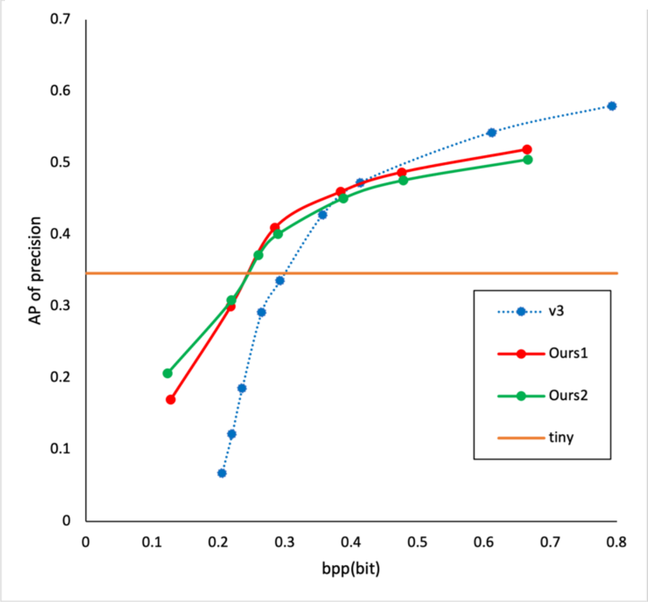


Fig. 8. Object detection AP for our methods, YOLOv3 with varying compression ratio (only cloud side system) and YOLOv3-tiny (only edge side system)

C. *Discussion*

The result shows that the proposed method is effective when the amount of data to be transmitted is small. This is because by processing the maximum amount of what can be processed by the model on the edge side, the overall transmission amount can be reduced even when the image compression ratio is small. As a result, it is considered that better quality images can be sent even with the same small transmission amount. Furthermore, the results of the detection by YOLOv3-tiny on the edge side and the re-inference of the masked image by YOLOv3 on the cloud side are shown in Fig. 9. It can be observed that YOLOv3 on the cloud side successfully compensates for the parts that were not detected completely on the edge side.

V. Conclusion

In this paper, we proposed an object detection system for video using the ECNet framework. The video object detection process is divided into two parts, with the lightweight YOLOv3-tiny on the edge side and the high-accuracy YOLOv3 on the cloud side. Furthermore, we reduced the number of inferences by using the frame difference and introduced a method of masking each frame image by the confidence value of each grid cell. This showed that the proposed method is more effective at low bit rates than simply transmitting to the cloud for processing. As a future task, we plan to devise a system that can maintain accuracy even when the transmission amount is large.

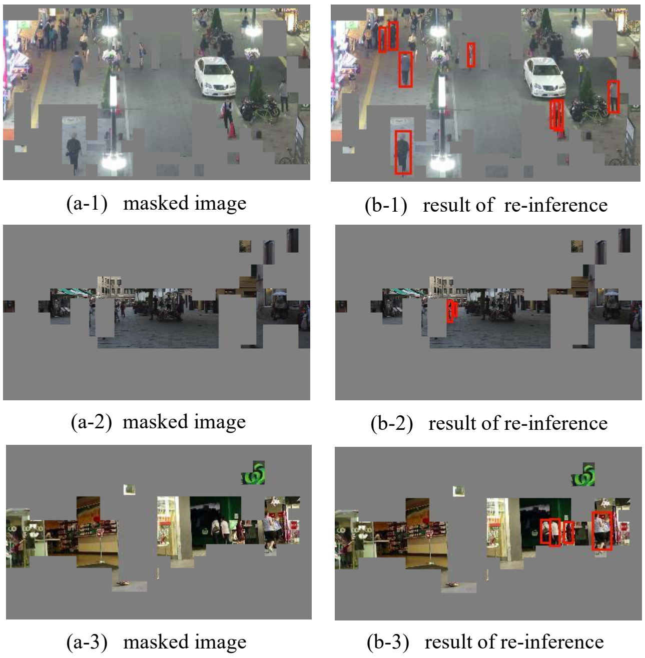

(a-1) masked image
(b-1) result of re-inference
(a-2) masked image
(b-2) result of re-inference
(a-3) masked image
(b-3) result of re-inference

Fig. 9. Results of the masked iamge by YOLOv3-tiny on the edge side and the re-inference of the masked image by YOLOv3 on the cloud side.

References

[1] H. Choi and I. V. Bajic, “Deep Feature Ccmpression for Collaborative Object Detection,” IEEE International Conference on Image Processing (ICIP), pp. 3743-3747, Oct. 2018.

[2] K. Iino et al., "Inter-Feature-Map Differential Coding of Surveillance Video," IEEE Global Conference on Consumer Electronics (GCCE), pp. 286-289, Oct. 2022.

[3] L. Hu et al., “ECNet: A Fast, Accurate, and Lightweight Edge- Cloud Network System based on Cascading Structure,” IEEE Global Conference on Consumer Electronics (GCCE), pp. 259-262, Oct. 2020.

[4] J. Redmon and A. Farhadi, “YOLOv3: An Incremental Improvement,” arXiv:1804.02767, Apr. 2018.

[5] P. Adarsh, P. Rathi and M. Kumar, “YOLO v3-Tiny: Object Detection and Recognition using one stage improved model,” International Conference on Advanced Computing & Communication Systems (ICACCS), pp. 687-694, Mar. 2020.

[6] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, “You only look once: Unified, real-time object detection,” IEEE Computer Vision and Pattern Recognition Conference (CVPR), pp. 779–788, Dec. 2016.

[7] J. Redmon and A. Farhadi, “YOLO9000: Better, faster, stronger,” 3 IEEE Computer Vision and Pattern Recognition Conference (CVPR), pp. 6517–6525, Jan. 2017.

[8] T. Lin et al., “Feature Pyramid Networks for Object Detection,” IEEE Computer Vision and Pattern Recognition Conference (CVPR), pp. 2117-2125, Jan. 2017.

[9] Z. Huang, F. Li, X. Luan and Z. Cai, “A Weakly Supervised Method for Mud Detection in Ores Based on Deep Active Learning,” Mathematical Problems in Engineering, vol. 2020, May. 2020.

[10] G. K. Wallace, "The JPEG still picture compression standard," IEEE Transactions on Consumer Electronics (TCE), vol. 38, no. 1, pp. xviii-xxxiv, Feb. 1992.

[11] A. Milan, L. Leal-Taixe, I. Reid, S. Roth, and K. Schindler, “MOT16: A Benchmark for Multi-Object Tracking,” arXiv:1603.00831, Mar. 2016.