Edge-Cloud Collaborative Object Detection Model with Feature Compression

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

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

Recently, dramatic performance improvements in object detection models have increased the demand for real-time video processing tasks in edge devices. However, it is trade-off between real-time processing and high detection accuracy. Hence, a two-phase prediction network model, Edge-Cloud Net (ECNet), has been proposed to coordinate the edge-side AI model and the cloud-side AI model. However, ECNet is originally designed for image classification task. In this study, we propose a method to apply ECNet into object detection tasks utilizing a lightweight edge model connecting YOLOv3 and YOLOv3-tiny. We also implement feature compression to reduce the amount of transmission to be sent to the cloud side. Our approach can reduce the amount of data transmission while concurrently preserving object detection accuracy, particularly in instances when dealing with small bpp.

Keywords: Edge-Cloud Net, YOLOv3, YOLOv3-tiny, Feature Compression.

1. Introduction

These days, the performance of object detection models has improved dramatically with the rise of deep learning-based object detection models. Consequently, there is a need for real-time processing on the edge device side for use cases such as video analysis in surveillance cameras and automated driving. However, the latest high-performance models are so processing-heavy that they cannot be loaded onto the limited computing resources on the edge device side. On the other hand, when processing on the cloud side, such as in a data center, data transmission over the Internet is required, which may cause processing delays or missing data. Therefore, the Edge-Cloud Net (ECNet) [1] is proposed, which combines a lightweight model on the edge side with a high-performance model on the cloud side and achieves a good trade-off between transmission cost and accuracy. In our previous study, we applied the ECNet mechanism to object detection [2]. In that study, we designed a cooperative network with lightweight YOLOv3-tiny [3] on the edge side and high-performance YOLOv3 [3] on the cloud side. However, that approach involves two different networks, even though the edge and the cloud are performing the same task, which increases the computational cost, as well as model training costs for each dataset. To solve the issue, it is necessary to create a shared part of the model between the edge and the cloud. Therefore, we create a model for the edge by joining the first 10 layers of YOLOv3 (YOLOv3 Head) and the second half of YOLOv3-tiny, excluding the first 4 layers (YOLOv3-tiny Tail). In addition, by storing the features obtained by YOLOv3 Head and applying feature compression before transmission, the data volume can be significantly reduced compared to image compression such as JPEG.

2. Edge-Cloud Net for Computer Vision Tasks

Edge-Cloud Net (ECNet) [1] is a technique for deploying models that comprise light processing but low accuracy on the edge side and those with heavy processing but high accuracy on the cloud side for inference. The offload controller determines whether or not to use the cloud side. In [1], ECNet is originally designed for image classification tasks by utilizing Darknet19 and Darknet53, which are YOLOv3 [3] backbones. In [2], lightweight YOLOv3-tiny and high-performance YOLOv3 are utilized for object detection in videos combining masking for each frame image and compressing the image by JPEG. These methods can help to achieve a balance between transmission volume and accuracy.

3. Proposed Method

We create lightweight edge model which combines the first 10 layers of YOLOv3 (YOLOv3 Head) and the second half of YOLOv3-tiny, excluding the first 4 layers (YOLOv3-tiny Tail). We implement a mechanism to store the feature at the connecting point between the YOLOv3 Head and YOLOv3-tiny Tail in the edge model and compress the feature when they are sent to the cloud. Then, the transmitted feature becomes an input to YOLOv3 Tail, which performs inference on the cloud side. The overall network structure is shown in Fig. 1.

For model size, our edge model is very light compared to full model of YOLOv3 for the cloud side operation.

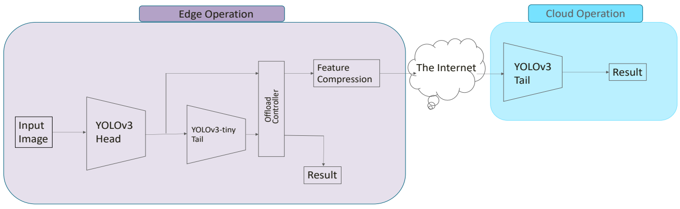

Figure 1: Overall network of proposed method

The number of parameters in our edge model is 18M, while the number of parameters in the cloud model is 62M.

For offload controller, we use the objectness value and confidence value of the edge model output. The first step here is to extract the parts of the output image where the objectness of each cell is higher than the threshold value x_1. The value of confidence is obtained for the extracted area whose objectness value is larger than x_1, that is, the area where the object is likely to be. The maximum value is obtained for those confidence values, and if the value is less than the threshold value x_2, the feature is sent to the cloud side. This is because a small value of confidence indicates that the inference at the edge is clearly not complementing the object. When the maximum value of confidence exceeds x_2, it utilizes the detection results from the edge model and refrain from transmitting features to the cloud side. By changing these parameters, we control the amount of data sent to the cloud side.

For feature compression, we implement an encoder consisting of convolution1, GDN, and convolution2, and a corresponding decoder. Referring to the paper in [4], each layer's parameters are defined.

4. Experiment

We used COCO2017 test dataset [5] to serve as input for our evaluation. We compared our proposed method with the method proposed in [2], with only JPEG compression at the edge and simply sent to YOLOv3 at the cloud side, and with only feature compression at the edge and simply sent to YOLOv3 at the cloud side.

Fig. 2 shows the relationship between the amount of transmission to the cloud side (bpp) and detection accuracy (mAP) for each of the three methods which are proposed method, previous method [2], and a method simply compress by JPEG and detected by YOLOv3. Our method greatly improves the trade-off between transmission amount and accuracy over comparison method. Fig. 3 shows a comparison between the proposed method and the case where all data is sent to the cloud by using feature compression. The proposed method is especially effective when the data to be transmitted is low bpp.

The proposed method is superior at the low bpp because it reduces the amount of transmission by using only lightweight edge models for images where the models can be easily detected. On the other hand, since the proposed method has a small computational cost, it is considered inferior to simple processing on the cloud side when sending a large amount of data, i.e., at high bpp.

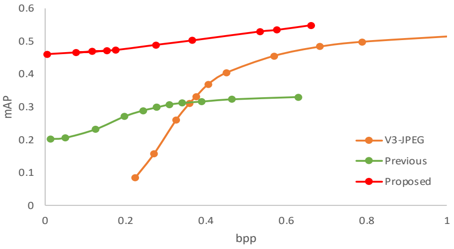

Figure 2: Trade-off between transmission amount (bpp) and accuracy (mAP) of proposed method, previous method [2], and a method simply compress by JPEG and detected by YOLOv3.

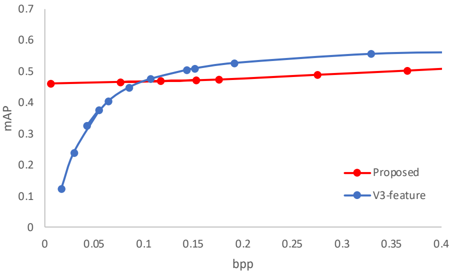

Figure 3: Trade-off between transmission amount (bpp) and accuracy (mAP) of proposed method and the case all data is sent to the cloud by using feature compression.

5. Conclusion

In this paper, we proposed an object detection system in the ECNet framework that utilizes the edge model connecting YOLOv3 and YOLOv3-tiny and introduced feature compression before transmission. We showed that the system is more effective than the conventional image compression methods as well as simple transmission to the cloud side, especially at low bpp.

References

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

[2] S. Akamatsu et al., "A Video Object Detection Method of ECNet Based on Frame Difference and Grid Cell Confidence," IEEE Global Conference on Consumer Electronics, pp. 364-367, (2023).

[3] J. Redmon et al., "YOLOv3: An incremental improvement," arXiv preprint: 1804.02767, (2018).

[4] M. Yamazaki et al., "Deep Feature Compression using Rate-Distortion Optimization Guided Autoencoder," IEEE International Conference on Image Processing, pp. 1216-1220, (2022).

[5] T. Lin et al., "Microsoft COCO: Common Objects in Context," arXiv preprint:1405.0312, (2014).