Bounding Box Aware Edge-Cloud Collaborative Method for Multiple Object Detection

Shunsuke Akamatsu
Graduate School of Fundamental Science and Engineering
Waseda University
Tokyo, Japan
s.akamatsu@akane.waseda.jp

Hiroshi Watanabe
Graduate School of Fundamental Science and Engineering
Waseda University
Tokyo, Japan
hiroshi.watanabe@waseda.jp

2024 IEEE 3rd World Conference on Applied Intelligence and Computing (AIC) | 979-8-3503-8459-8/24/$31.00 ©2024 IEEE | DOI: 10.1109/AIC61668.2024.10731035

***Abstract*—The demand for real-time video processing from edge devices including surveillance cameras and smartphones has been increasing. While edge processing power is improving with lighter recognition models and smaller GPUs, achieving high-performance recognition remains a challenge due to limited computational resources. To address the issue, collaborative recognition systems between the edge side and the cloud side are crucial. Previous approaches such as the Edge-Cloud Net (ECNet) have been proposed but they faced challenges in optimizing data transmission because of the large data size of frame images. In this paper, we propose a novel edge-cloud collaborative method for video multiple object detection. This system integrates an original lightweight edge side model that combines YOLOv3 and YOLOv3-tiny and compresses intermediate features before the transmission. Our approach improves the trade-off between transmission amount and detection accuracy, particularly at low bit rates. This approach also focuses on offload controlling based on detected bounding boxes from the edge side model and it enhances the trade-off compared to the previous method.**

***Index Terms*—Edge-Cloud network system, Object detection, Feature compression, Bounding box aware offload controller**

I. Introduction

In recent years, the demand for real-time processing of video captured by edge devices such as surveillance cameras and smartphones has been accelerating. Although the processing power of edge devices is increasing due to lighter recognition models and smaller GPUs, it is still very difficult to achieve high-performance object recognition with limited computational resources. At the same time, when data is sent to a cloud side such as a server center, processing delays can occur in exchange for high accuracy. Therefore, collaborative systems are needed in which lightweight models are placed on the edge side and high-performance models are placed on the cloud side [1]- [5]. Previously, the collaborative object detection system based on Edge-Cloud Net (ECNet) has been proposed [6], [7]. However, in the previous research, the transmission was based on the raw image data output of the edge side model, and therefore, the amount of transmission data size could not be optimally reduced due to the large data size of raw image of each frame. In addition, previous research such as presented in ref. [8] using intermediate features for transmission is limited to detection for image inputs only. Therefore, we propose an edge-cloud collaborative object detection system to address the issue of multiple object detection for video inputs. This is expected to be applied in real-world use cases including large-scale camera analysis and autonomous driving. We created an original lightweight edge model that combines YOLOv3 [9] and its lightweight model YOLOv3-tiny [10]. We also incorporated a mechanism to compress and transmit intermediate features on the edge side by creating common parts with the cloud side model. Fig. 1 shows the comparison between the conventional method of transmitting image data and this proposed method of transmitting intermediate features from the edge side model. For the offload controlling of transmission, we propose an offload method that focuses on the information of detected bounding boxes from the edge side model. This can improve the trade-off between transmission amount and detection accuracy compared to previous methods and it is also effective especially at low bit rates compared to transmitting all data to

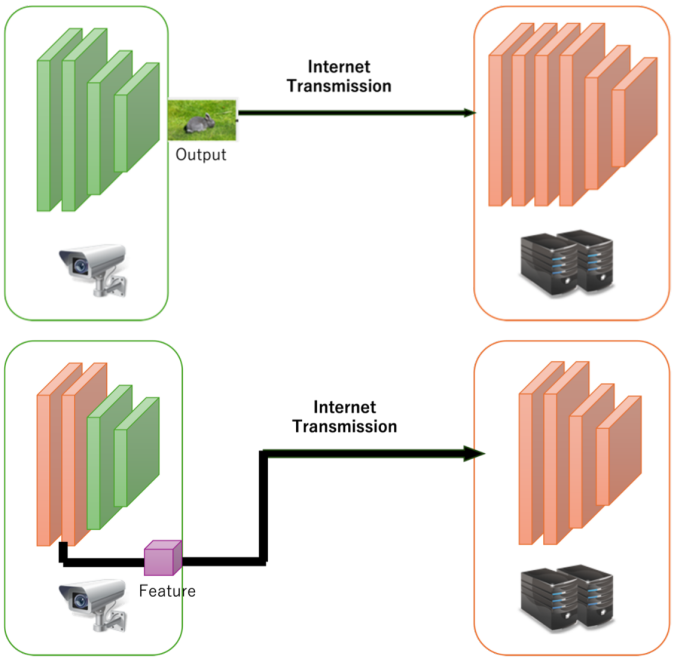

Fig. 1. The difference in transmission of collaborative systems.

979-8-3503-8459-8/24/$31.00 ©2024 IEEE
DOI: 10.1109/AIC.2024.189

the cloud side.

The remainder of this paper is organized as follows: Section 2 describes related work of our study. Section 3 illustrates our proposed method regarding network structure, offloading control, and feature compression. Section 4 discusses the experiments and their results. The last section presents conclusion from the results.

II. Related Works

A. Edge-cloud Network for Computer Vision Tasks

To tackle the trade-off between the transmission amount and recognition accuracy, there are several studies have been conducted using edge-cloud collaborative networks for computer vision tasks. Edge-Cloud Net (ECNet) [6] performs the image classification task by placing models with light processing load while low accuracy as the edge side classification and models with heavy processing load while high accuracy as the cloud side classification. They use Darknet19 and Darknet53, the backbone of YOLO9000 [11] and YOLOv3, respectively. For the object detection task, the method that combines high-speed YOLOv3-tiny on the edge with high-accuracy YOLOv3 on the cloud has been proposed [7]. They mask each frame image depending on the confidence score and apply image compression using JPEG [12] format to reduce transmission amount. Nevertheless, the considerable data size of raw frame images has constituted a significant challenge in previous studies, impeding the attainment of optimal trade-off efficiency.

B. Feature Compression

An important aspect of building an edge-cloud collaborative object recognition system is to reduce the size of the data transmitted from the edge to the cloud. Therefore, it is essential to compress the transmitted data on the edge side before transmission over the Internet. While conventional image compression includes traditional compression methods such as JPEG and learning-based compression methods, the amount of transmitted data size is still large. In contrast, feature compression, which uses intermediate features of edge side models, can greatly compress the amount of data to be transmitted compared to raw image data. Moreover, it is also highly compatible with edge-cloud collaborative systems, where the focus is on machine recognition rather than reconstruction for human vision. Therefore, feature compression methods have been proposed in edge-cloud collaborative recognition systems in recent years [13] - [16].

C. Real-time Object Detection

In recent years, Convolutional Neural Network (CNN) based algorithms have been proposed for real-time object detection to achieve both highly accurate detection and quick response time. Typical examples include Regional-based Convolutional Neural Network (R-CNN) [17] and You Only Look Once (YOLO) [18]. R-CNN combines region search and image recognition algorithms for object detection, resulting in high accuracy but slow processing speed. Fast R-CNN [19] and Faster R-CNN [20] have been proposed to achieve faster inference, but their processing speed is insufficient to achieve real-time detection. In contrast to other conventional object detection methods, YOLO, however, achieves faster processing speed by simultaneously searching for possible regions and identifying classes. YOLOv3, which is used in this proposed network, has improved detection accuracy over previous YOLO models by increasing the number of layers and allowing detection at multiple scales. In addition, YOLOv3-tiny is a lightweight model with fewer layers than YOLOv3, which enables faster inference [21]. In terms of the model size and detection performance, YOLOv3 can achieve 57.9 mAP while 20 FPS and YOLOv3-tiny can achieve 33.1 mAP while 220 FPS on COCO dataset [22]. In summary, YOLOv3 is a highly accurate model and YOLOv3-tiny is a lightweight model for detection.

III. Proposed Method

A. Network Structure

We use the model structure of YOLOv3-tiny and YOLOv3 for our entire network structure. On the edge side, the first 13 layers of YOLOv3 (Edge Head / Cloud Head), and the second half of YOLOv3-tiny, excluding the first 8 layers (Edge Tail) via the Connect layer which adopts the size of input are implemented. On the cloud side, we just deploy the latter part of YOLOv3 (Cloud Tail). We also implement a mechanism that stores features at the Edge Head output on the edge side and compresses the features when sending them to the cloud side. The transmitted features then become input to the Cloud Tail, which performs inference on the cloud side. The overall collaborative network structure is shown in Fig. 2.

B. Bounding Box Aware Offload Controller

We implement a two-step process for offload control using the information of detected bounding boxes from the edge side model. The first step is to use the number of bounding boxes detected by the edge side model and we introduce x as the threshold parameter. If the difference in this number from the number of detected bounding boxes in the previous frame image is greater than a threshold of x percent, the frame image is transmitted to the cloud side. This is because when the next frame has similar feature to the previous frame, the number of detected bounding boxes does not change significantly. Hence, the inference results from the previous frame can be reliable, and the inference on the edge side is considered sufficient. For the second step, we use the minimum value of the detected bounding box area. Generally, less accurate models are not good at detecting small objects [25], and hence we use this value as a parameter to control the amount of transmission.

C. Feature Compression

From the output of the Edge Head model, intermediate features are stored, and the features are compressed and sent to the cloud side. We implement an encoder consisting of two convolution layers and a GDN [23] layer followed by a corresponding decoder for feature compression. Fig. 3 presents the architecture of feature compression. We also refer to the

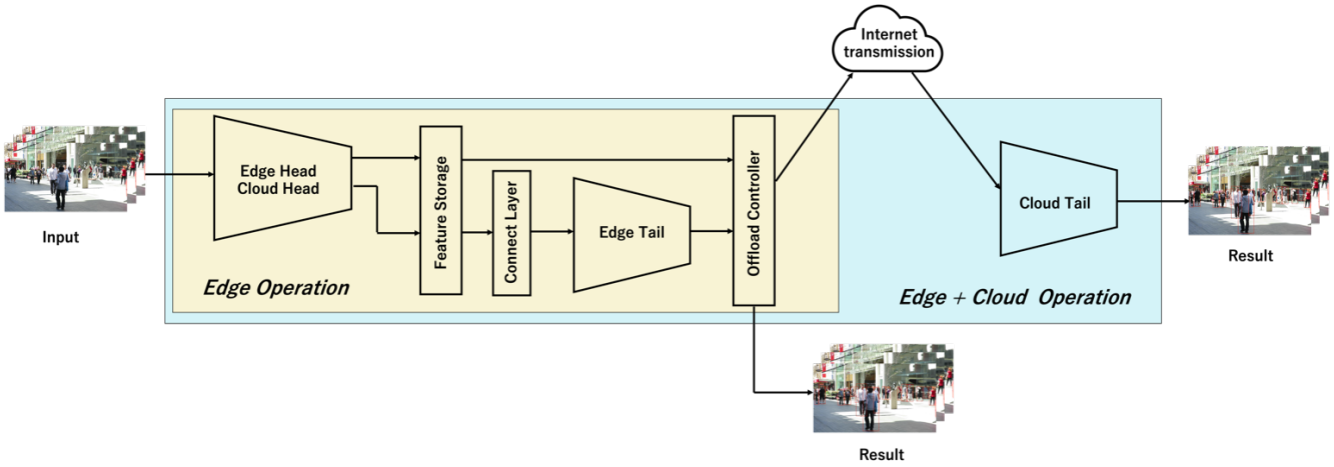

Fig. 2. The overall network structure of proposed method.

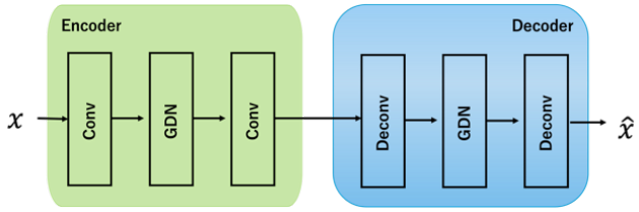

Fig. 3. The architecture of feature compression.

TABLE I
PARAMETERS FOR FEATURE COMPRESSION

Layer	Number of filters	Stride
Convolution 1	256	1
GDN	192	-
Convolution 2	128	2

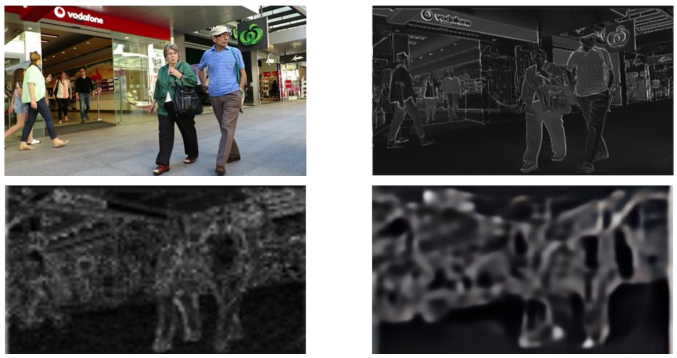

Fig. 4. MOT17 data samples and the features at specific layers. (upper left: raw frame image, upper right: feature at the second layer, lower left: feature at the ninth layer, lower right: feature at the thirteenth layer (the last layer of the shared part))

paper in ref. [24] and define the parameters of each layer as shown in Table 1.

IV. EXPERIMENT

A. Evaluation Method

We use the MOT17 [26] dataset for performance evaluation. MOT17 dataset is a more accurate ground truth-filled version of the previously published MOT16 dataset. There are 14 video sequences captured with both static and moving cameras, including 7 training sequences and 7 test sequences. The MOT17 data sample and the features at specific layers which are the second, ninth, and thirteenth (the last layer of the shared part) layers on the edge side model are shown in Fig. 4. For performance comparison, we use the previous method [7] and the case where the compression of intermediate feature is solely performed on the edge side and all data processed for inference on the cloud side.

B. Result

First of all, we compare the size for each model, and a comparison of the total number of parameters used to process for each model is presented in Table 2. Our edge model which combines YOLOv3 and YOLOv3-tiny is much lighter in processing compared to the full YOLOv3 model for the cloud side inference. Table 3 also shows a comparison of those detection results for the bounding box area as well as the number of detected boxes from the edge side model and the cloud side model. It shows that the edge side model has difficulty detecting small objects compared to the more accurate cloud side model. Examples of the actual differences between inference results from the edge model and from the cloud model are shown in Fig. 5.

Secondly, we evaluate the performance of our proposed method. Fig. 6 shows the trade-off between the data transmission amount to the cloud side (bpp) and overall detection accuracy (mAP) of the proposed method by changing the threshold x and the previous method. Our proposed method significantly improves the trade-off between the transmission amount and detection accuracy in contrast to the previous approach, which placed the full YOLOv3-tiny model on the edge and the full YOLOv3 model on the cloud, with compression applied using JPEG. Fig. 7 also shows a comparison between the proposed

TABLE II
THE COMPARISON OF THE NUMBER OF PARAMETERS

Model	Edge		Cloud
	Edge Detection	*Compression*	*Cloud Detection*
Parameters [M]	10.0	3.77	61.9

TABLE III
THE COMPARISON OF DETECTED BOUNDING BOX AREA

	Min	Average	Max	Number
Edge side model	383.99	2619.14	49403.12	114.78
Cloud side model	23.51	1263.65	30349.77	91.40

method and the approach where feature compression is applied exclusively and all data is sent to the cloud. From the results, our proposed method excels particularly when the volume of transmitted data is low.

Last not the least, we focus on the comparison of computational complexity by the edge side model and the cloud side model. Hence, we estimate the computational parameters for processing each frame image. The computation amount per frame image is calculated as follows:

$$P = E_d + (transmitted\ data/all\ data) * (E_c + C_d), \quad (1)$$

where P represents the amount of calculation parameters for one frame image of processing. E_d and E_c indicate respectively the number of parameters in the edge side detection and compression model, and C_d stands for the number of parameters in the cloud side detection model. The number of parameters for each detection and compression model is detailed in Table 2. The relationship between the number of calculation parameters required for processing per frame image and detection accuracy is shown in Fig. 8.

Fig. 5. Detection differences between edge and cloud models for the MOT17 dataset. (left: edge side model, right: cloud side model)

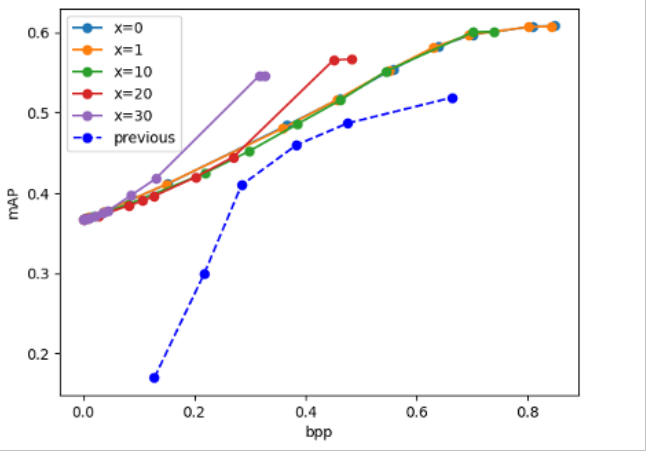

Fig. 6. The trade-off between bitrate (bpp) and mAP of proposed method (changing in threshold x) and previous method. [7]

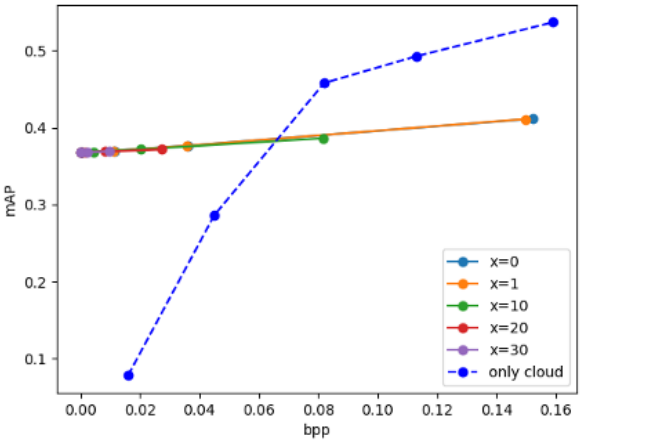

Fig. 7. The trade-off between bitrate (bpp) and mAP of proposed method (changing in threshold x) and the case all data is sent to the cloud side by using feature compression.

C. Discussion

From the results, the proposed method improves the trade-off between bitrate of transmitted data and object detection accuracy significantly compared to existing previous methods. This shows the advantage of compression with intermediate features over methods that use the output of edge models. Furthermore, by focusing on the detected bounding box from the edge side output, the proposed method discriminates between easy and difficult cases in detecting multiple objects. This leads to the reduction of the transmission amount to the cloud side. In addition, compared to sending all the data to the cloud side, the edge side model is very light processing, and the cost of computation can be controlled by changing the threshold value according to the required use cases.

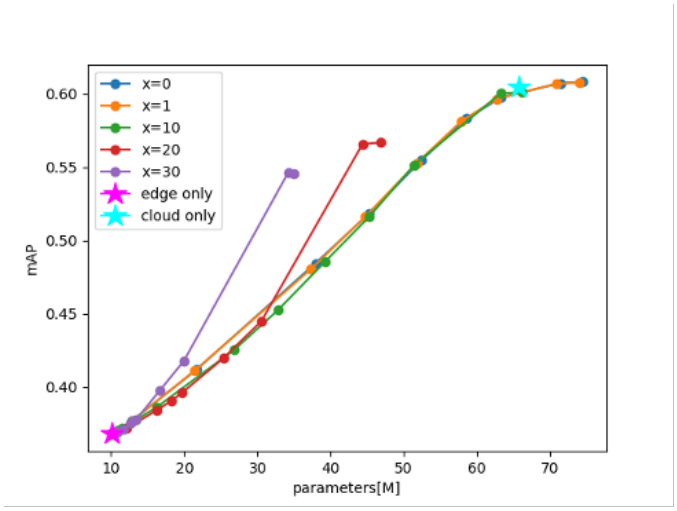

Fig. 8. The relationship between the number of calculation parameters required for each image and detection accuracy.

V. Conclusion

We propose the edge-cloud collaborative method for multiple object detection in videos. We combine offload controller focusing on bounding boxes and feature compression to reduce and control the amount of transmission to the cloud side. Our proposed method improves the trade-off between the transmission amount and detection accuracy in comparison with the previous method. It is also effective compared to sending all data to the cloud side especially at low bit rates. Hence, the proposed method is expected to be applied to object detection in cases where network bandwidth is limited. Given the limitations of the current study in terms of the efficiency of the trade-off at high bitrates, future research will focus on networks and offload control mechanisms that can achieve high efficiency at any bitrate bandwidth.

References

[1] I. V. Bajić, W. Lin and Y. Tian, "Collaborative Intelligence: Challenges and Opportunities," IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 8493-8497, Jun. 2021.

[2] J. C. Lee, Y. Kim, S. Moon and J. H. Ko, "A Reconfigurable Neural Architecture for Edge–Cloud Collaborative Real-Time Object Detection," IEEE Internet of Things Journal, Vol. 9, No. 23, pp. 23390-23404, Dec. 2022.

[3] Y. Yuan, S. Gao, Z. Zhang, W. Wang, Z. Xu and Z. Liu, "Edge-Cloud Collaborative UAV Object Detection: Edge-Embedded Lightweight Algorithm Design and Task Offloading Using Fuzzy Neural Network," IEEE Transactions on Cloud Computing, Vol. 12, No. 1, pp. 306-318, Mar. 2024.

[4] Z. Cao, Z. Li, Y. Chen, H. Pan, Y. Hu and J. Liu, "Edge-Cloud Collaborated Object Detection via Difficult-Case Discriminator," 2023 IEEE International Conference on Distributed Computing Systems (ICDCS), pp. 259-270, Jul. 2023.

[5] Y. Chen, Y. Lin, Y. Hu, C. Hsia, Y. Lian and S. Jhong, "Distributed Real-Time Object Detection Based on Edge-Cloud Collaboration for Smart Video Surveillance Applications," IEEE Access, Vol. 10, pp. 93745-93759, Aug.2022.

[6] L. Hu, T. Wang, H. Watanabe, S. Enomoto, X. Shi, A. Sakamaoto, and T. Eda, "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.

[7] S. Akamatsu, K. Iino, H. Watanabe, S. Enomoto, X.Shi, A. Sakamoto and T. Eda "A Video Object Detection Method of ECNet Based on Frame Difference and Grid Cell Confidence", IEEE Global Conference on Consumer Electronics (GCCE), pp364-367, Oct. 2023.

[8] S. Akamatsu, K. Iino, H.Watanabe, S.Enomoto, A.Sakamaoto and T.Eda, "Edge-Cloud Collaborative Object Detection Model with Feature Compression," IIEEJ International Conference on Image Electronics and Visual Computing (IEVC), Mar. 2024, *in press*.

[9] J. Redmon and A. Farhadi, "YOLOv3: An Incremental Improvement," arXiv:1804.02767, Apr. 2018.

[10] 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.

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

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

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

[14] S. Suzuki, M. Takagi, S. Takeda, R. Tanida and H. Kimata, "Deep Feature Compression With Spatio-Temporal Arranging for Collaborative Intelligence," IEEE International Conference on Image Processing (ICIP), pp. 3099-3103, Oct. 2020.

[15] M. Hossain, Z. Duan, Y. Huang and F. Zhu, "Flexible Variable-Rate Image Feature Compression for Edge-Cloud Systems," IEEE International Conference on Multimedia and Expo Workshops (ICMEW), pp. 182-187, Jul.2023.

[16] Z. Duan and F. Zhu, "Efficient Feature Compression for Edge-Cloud Systems," Picture Coding Symposium (PCS), pp187-191, Dec. 2022.

[17] R. Girshick, J. Donahue, T. Darrell and J. Malik, "Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation," IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 580-587, Jun. 2014.

[18] 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.

[19] R. Girshick, "Fast R-CNN," IEEE International Conference on Computer Vision (ICCV), pp. 1440-1448, Dec. 2015.

[20] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 39, No. 6, pp1137-1149, Jun. 2017.

[21] D.Xiao, F.Shan, Z.Li, B.T.Le, X.Liu, and X.Li, "A Target Detection Model Based on Improved Tiny-Yolov3 Under the Environment of Mining Truck," IEEE Access, Vol. 7, pp. 123757–123764, Jul. 2019.

[22] R. Huang, J. Pedoeem, and C. Chen, "YOLO-LITE: A Real-Time Object Detection Algorithm Optimized for Non-GPU Computers," IEEE International Conference on Big Data (Big Data), pp. 2503-2510, Dec. 2018.

[23] J. Ballé, V. Laparra and E. Simoncelli, "Density Modeling of Images using a Generalized Normalization Transformation," International Conference on Learning Representations (ICLR), May. 2016.

[24] M. Yamazaki, Y. Kora, T. Nakao, X. Lei and K. Yokoo, "Deep Feature Compression using Rate-Distortion Optimization Guided Autoencoder," IEEE International Conference on Image Processing (ICIP), pp. 1216-1220, Oct. 2022.

[25] Z.Wang, K.Xie, X. Zhang, H. Chen, C. Wen and J. He, "Small-Object Detection Based on YOLO and Dense Block," IEEE Access, Vol. 9, pp. 56416-56429, Apr. 2021.

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