CLASS-WISE FM-NMS FOR KNOWLEDGE DISTILLATION OF OBJECT DETECTION

Lyuzhuang Liu, Tsubasa Hirakawa, Takayoshi Yamashita, and Hironobu Fujiyoshi*

Chubu University

ABSTRACT

The trade-off between accuracy and speed for an object detection model is important. When we implement an object detection model in embedded devices, a lightweight model can accelerate the detection speed. Meanwhile, the detection accuracy will be decreased. In this paper, we propose a knowledge distillation method for a lightweight object detection model. The proposed method introduces an improved feature map novel non-maximum suppression (FM-NMS) method. The improved FM-NMS uses different focus size with respect to each object class, which can suppress false positives and improve detection accuracy. In our experiments, we use one-stage object detection methods, YOLOv4 as a teacher model and YOLOv4-tiny as a student model, and we apply the proposed method to them. The experimental results demonstrate that the proposed method improves the detection accuracy of the student model while maintaining the lightweight model size.

***Index Terms*—** Object detection, Knowledge distillation, Feature map non-maximum suppression

1. INTRODUCTION

Object detection [1, 2, 3, 4, 5, 6, 7] is widely investigated and used for some application fields, such as autonomous driving [8, 9] and robotics [10, 11]. The existing methods have achieved higher detection accuracy while the network structures of such methods becomes more complex and larger-scale. The large-scale model faced a problem for implementing detection models into an embedding device. Therefore, the trade-off between detection accuracy and processing speed is an important factor.

For building a lightweight detection model, some approaches have been proposed such as quantization of model parameters [12, 13], pruning [14, 15], and knowledge distillation [16]. Among them, knowledge distillation (KD) [16] is an efficient approach for reducing model size and maintaining accuracy. KD uses two networks, one is teacher model and the other is student model, and train the student model with hard target and soft target. By using soft target as an additional loss function, the student model can achieve higher accuracy.

The KD for object detection model have also been proposed [17, 18]. Mehta et al. [18] proposed feature map non-maximum suppression (FM-NMS), which applies non-maximum suppression (NMS) for feature maps. They introduced the FM-NMS for knowledge distillation of object detection. However, object scales in an image are different over object class while the conventional FM-NMS uses the same focus size of NMS across every object classes. This causes the failure detection results of small objects.

To overcome this problem, in this paper, we introduce a novel FM-NMS for KD of object detection model. The proposed FM-NMS uses different focus size of NMS with respect to each object class, which can detect objects considering appropriate object size. Moreover, by using the different focus size, we can remove redundant focus size on NMS. Therefore, we can reduce the computational cost on the NMS process. By introducing the proposed method into a one-stage detection method [6, 19], we can achieve higher detection accuracy

The contribution of this paper are as follows:

- This paper introduces a novel NMS. The proposed NMS uses different focus size for each object class. This achieves more accurate object detection.
- The proposed method provides faster processing speed due to the redundant detection frame is processed by FM-NMS, which saves the time of normal NMS processing. The experimental results show that our method is faster than YOLOv4-tiny.

2. RELATED WORK

Object detection is a widely investigated problem in the computer vision and image processing communities. Over the last decade, due to the development of deep neural network techniques, a lot of approaches have been proposed [1, 2, 3, 4, 6, 7]. The object detection methods can be categorized into a couple of approaches: two-stage and one-stage. Two-stage approach [1, 2, 3] consists of object proposal and classifier. The object proposal predicts candidates of object regions from an input image. Then, each proposal are input to classifier, and object class of each proposal are classified. The other is one-stage, which consists of a single network and predicts objects in an end-to-end manner [4, 6, 19]. You Only

Look Once (YOLO) [4, 6, 19, 7] is one of the one-stage detection method. YOLO predicts confidence score and class probabilities, simultaneously. By using these values, the final detection results are predicted. The one-stage method is faster than two-stage method because the region proposal in the two-stage method becomes the bottle neck of the computational efficiency. In this paper, we use one-stage model in our experiments.

Knowledge distillation (KD) [16] is a method for training a small network while maintaining the accuracy. KD uses two networks: teacher model that is pre-trained larger network and student model that is smaller network than teacher model. KD trains the student model by using hard target that is calculated from correct label of a dataset and soft target that is calculated from the output of the teacher model. The soft target increases the accuracy of the student model.

The KD is also applied for object detection. Chen et al. [17] proposed a method of simultaneous knowledge distillation for the feature-extraction layer, classification loss, and regression loss using Faster R-CNN [3] as a detection model. However, a two-stage object detection model is more time-consuming than the one-stage object-detection model, making it difficult to implement in embedded terminals. Mehta et al. [18] proposed the feature map non-maximum suppression (FM-NMS). The FM-NMS applies the NMS process to feature maps. The FM-NMS searches for the maximum points in 3×3 neighboring grid cells of the feature map output from the teacher model. They conducted KD using the output of FM-NMS process as a soft target to improve the accuracy of a one-stage object detector, YOLOv2 [5], and achieved better detection accuracy. However, the FM-NMS does not take into account the suitable focus size for each class. In this paper, we propose a novel FM-NMS that consider different focus size depending on each object class.

3. PROPOSED METHOD

In this paper, we propose a KD method with a novel FM-NMS process. The conventional FM-NMS is carried out with the same focus size for all classes. However, because the appropriate focus size is different over each object class. The proposed FM-NMS uses the different focuses, which results in improving detection accuracy and reducing computational cost.

Figure 2 shows the overview of the proposed method. In the proposed method, we first input an image to the pre-trained teacher model and student model. Then, we apply the proposed FM-NMS process for the teacher model output, and use the output value as a soft target. With the soft target and correct label, we calculate soft target and hard target losses, respectively. Finally, we update the parameter of the student model by using the soft and hard target loss.

Hereafter, we describe the details of the proposed method.

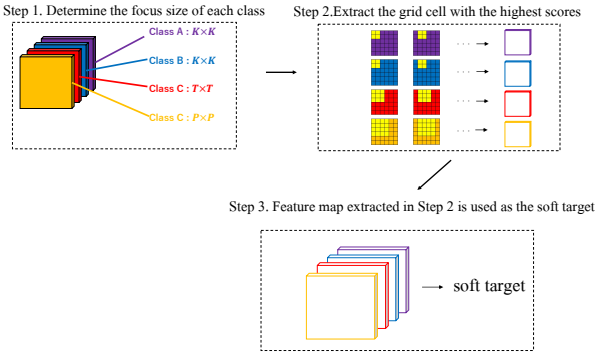


Fig. 1. The process flow of the proposed FM-NMS for each class.

3.1. FM-NMS Process for Each Class

The proposed FM-NMS uses different focus size depending on object classes. We show the process flow of the proposed FM-NMS in Fig. 1. In this method, larger object class uses larger focus size while smaller object class uses smaller focus size. The process is consists of the following three steps. In Step 1, we determine the focus size of each class in accordance with the actual data set to be used. The focus size of a class with a large object size is set to be large, while the focus size of a class with a small object size is set to be small. In Step 2, among the focus sizes determined in Step 1, the grid cells corresponding to the points with the highest scores in the class corresponding to this focus size are extracted. In Step 3, the feature map extracted in Step 2 is used as the soft target, and the loss of the confidence output, classification output, and regression output of the student model is calculated, respectively.

3.2. Distillation Loss

In this paper, we assume that we use YOLOv4 [6] as a baseline model. Figure 3 shows the network architecture of YOLOv4. The smallest size of the feature map output by YOLOv4 is 19×19. In this feature map size, if each grid cell contains five bounding box, we get $19 \times 19 \times 5 = 1805$ detection results. Because not only the target object but also the background information is contained in the grid cells, the loss value should take background fields into account. We therefore use the confidence output of the teacher model as KD. The confidence output of the background is smaller than that of the target object. We can suppress the effect of the background from the distilled knowledge by using it as the coefficient of the losses for the classification and regression outputs of the teacher and the student models, respectively.

The loss function used for the proposed method L is defined as follows:

$$L = L_h(s, T) + \alpha L_s(s, t), \tag{1}$$

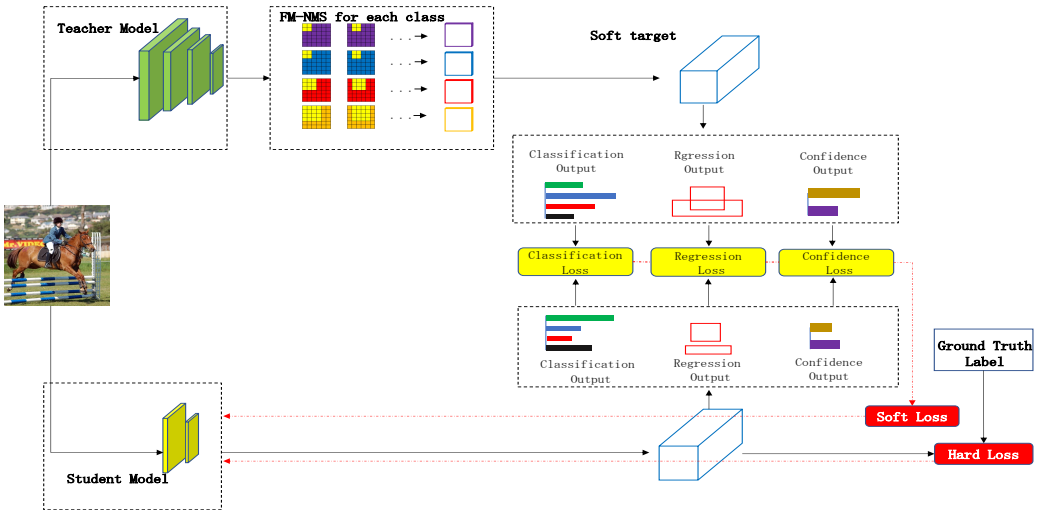


Fig. 2. Learning flow of the proposed method.

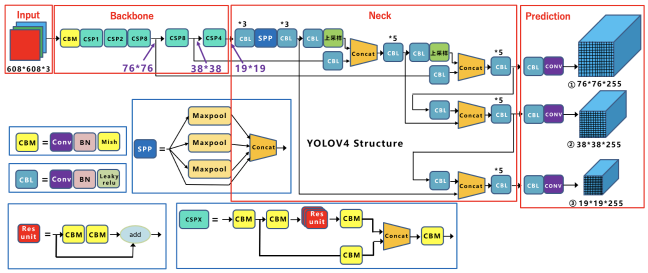


Fig. 3. Network structure of YOLOv4.

where s is the student model output, t is teacher model output, and T is correct label. α is a coefficient for the soft target loss $L_s(s,t)$. The hard target loss L_h is calculated from the output of the student model and correct label, which is defined by

$$L_h = L_{conf}(s,T) + L_{cls}(s,T) + L_{reg}(s,T). \quad (2)$$

Here, $L_{conf}(s,T)$ is the confidence loss. $L_{cls}(s,T)$ is the classification loss. $L_{reg}(s,T)$ is the bounding box regression loss. Every losses are calcluated by mean squared error. The soft target loss L_s is defined as follows:

$$\begin{aligned} L_s =& L_{conf_{kd}}(s,t) \\ &+ c_t L_{cls_{kd}}(s,t) \\ &+ c_t L_{reg_{kd}}(s,t), \end{aligned} \quad (3)$$

where $L_{conf_{kd}}(s,t)$ is the confidence loss of the student and teacher models. $L_{cls_{kd}}(s,t)$ is the classification loss of the student and teacher models. And, $L_{reg_{kd}}(s,t)$ is the bounding box regression loss of the student and the teacher models. Note that, these losses are calcluated by using mean squared error. c_t is the confidence output of the teacher model. Here we use c_t as the coefficient for $L_{cls_{kd}}(s,t)$ and $L_{reg_{kd}}(s,t)$. By using this coefficient, we can suppress the effect of the background from the distilled knowledge.

4. EXPERIMENTS

In this section, we evaluate the proposed method in terms of detection accuracy and computational efficiency of the proposed method.

4.1. Experimental Settings

We use PASCAL VOC 2007 and 2012 datasets [20]. The PASCAL VOC dataset contains 20 object classes and 16,551 samples for training, 4,952 for testing.

As a baseline model, we use YOLOv4 [6] as teacher model and YOLOv4-tiny [19] as student model. The comparative methods are as follows:

YOLOv4 and YOLOv4-tiny These models are trained with only hard target. Especially, YOLOv4 is used as a baseline for the larger detection model and as the teacher model for soft target.

Mehta et al. [18] We use YOLOv4 and YOLOv4-tiny as teacher and student models, respectively. We train the student model by their method.

Ours The proposed method. We set α in Eq. (1) as 1.0.

As an evaluation metric of the object detection, we use mean average precision (mAP).

In the proposed FM-NMS, we decide the focus size of each class from the average area of each class. FM-NMS was performed with a size of 2×2 for the classes with small areas, i.e., bottle, pottedplant, chair, TV monitor, sheep, and boat. Also, we use 4×4 focus size for the larger area object classes,

Table 1. mAPs in Pascal VOC dataset

Method	mAP	AP																			
		Aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	person	plant	sheep	sofa	train	TV
YOLOv4	86.4	94.2	91.8	85.6	79.0	80.5	91.7	95.8	91.2	74.1	91.3	78.7	89.9	92.5	92.0	89.8	62.6	86.5	82.1	91.9	87.4
YOLOv4-tiny	46.3	56.1	61.8	45.3	39.7	59.41	41.8	68.7	21.3	45.9	65.5	17.0	29.6	47.9	52.0	65.5	33.7	64.3	23.6	27.2	59.0
Mehta et al. [18]	57.8	68.5	75.2	53.1	44.8	58.9	57.4	80.6	49.5	45.8	67.8	32.0	51.3	65.7	68.0	74.5	40.2	65.1	40.0	51.3	66.5
Ours	58.9	66.1	71.4	53.5	46.5	59.2	62.7	81.6	47.4	50.5	66.2	36.4	51.8	69.8	69.9	76.0	40.3	64.9	43.3	56.4	63.9

i.e., train, aeroplane, bus, motorbike, diningtable, and horse. For the other classes, we set the focus size as 3×3.

4.2. Quantitative Results

We show the mAP for each method in Tab. 1. The proposed method achieved the highest mAP excluding YOLOv4. Comparing YOLOv4-tiny trained with only hard target, the use of the soft target loss increases the detection accuracy. Moreover, our method outperforms the method of Mehta et al. [18]. This result indicates the proposed FM-NMS is efficient for improving accuracy.

Focusing on each class, the proposed method improves the APs for potteplant, chair, boat, car, person, bird, horse, bus, diningtable, motorbike, train, dog, and sofa. Meanwhile, the APs for the other classes are lower than that of the method proposed by Mehta et al. [18]. The proposed FM-NMS uses the different focus sizes over different classes. Therefore, the appropriate focus size is important to improve the detection accuracy, which includes one of our future works.

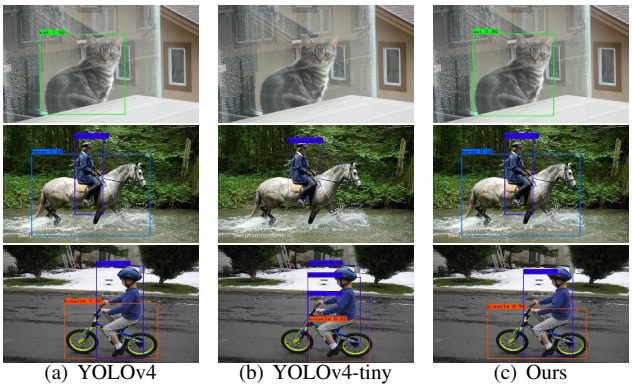
4.3. Qualitative Results

Figure 4 shows examples of the detection results of each method. The YOLOv4-tiny failed to detect some objects. Although our method uses the same network architecture, we can detect objects successfully. Meanwhile, our method can detect objects as with the results of YOLOv4. Therefore, the soft target affects on the detection accuracy and suppress the effect of background regions.

4.4. Processing Speed

We evaluate the processing speed of the proposed method. We compare the processing speed with YOLOv4-tiny. All of this experiment is conducted with Intel Xeon Gold 5122 CPU and Quadro RTX 8000 GPU. Each method is implemented using PyTorch framework.

Table 2 shows the processing speed of each method. Note that we exclude the time for image pre-processing and visualization of detection results from the measurement time. Although these network structures are the same and the inference time are not change, the proposed method is faster than YOLOv4-tiny. This reason is that the proposed FM-NMS selects focus size that is efficient for each class. By removing redundant focus size, we can reduce the time of FM-NMS while maintaining detection accuracy. From these results, the proposed FM-NMS is efficient in terms of both accuracy and computational efficiency.



(a) YOLOv4 (b) YOLOv4-tiny (c) Ours

Fig. 4. Examples of detection results.

Table 2. Processing speed

Method	Processing speed (fps)
YOLOv4-tiny	23.4
Ours	49.0

5. CONCLUSION

In this paper, we propose a novel FM-NMS and used for KD of object detection model. The proposed FM-NMS uses different focus size depending on object classes, which enables to detect object considering appropriate object size. Moreover, by removing redundant focus size, we can reduce the computational costs on the FM-NMS process. The experimental results show that the proposed method achieved higher mAP on Pascal VOC dataset. Also, we show that the proposed FM-NMS reduces the processing time comparing YOLOv4-tiny. Our future work includes applying KD for the intermediate layer features, deciding optimal focus size of the proposed FM-NMS, and extensive experiment to show the effectiveness of the proposed method.

6. REFERENCES

[1] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Computer Vision and Pattern Recognition (CVPR)*, 2014, pp. 580–587.

[2] Ross Girshick, "Fast R-CNN," in *International Conference on Computer Vision (ICCV)*, 2015, pp. 1440–1448.

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

[4] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi, "You only look once: Unified, real-time object detection," in *Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 779–788.

[5] Joseph Redmon and Ali Farhadi, "Yolo9000: better, faster, stronger," in *Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 7263–7271.

[6] Alexey Bochkovskiy, Chien-Yao Wang, and Hong-Yuan Mark Liao, "Yolov4: Optimal speed and accuracy of object detection," *arXiv preprint arXiv:2004.10934*, 2020.

[7] Chien-Yao Wang, Hong-Yuan Mark Liao, Yueh-Hua Wu, Ping-Yang Chen, Jun-Wei Hsieh, and I-Hau Yeh, "Cspnet: A new backbone that can enhance learning capability of cnn," in *Computer Vision and Pattern Recognition (CVPR) Workshops*, 2020, pp. 390–391.

[8] Di Feng, Ali Harakeh, Steven L. Waslander, and Klaus Dietmayer, "A review and comparative study on probabilistic object detection in autonomous driving," *IEEE Transactions on Intelligent Transportation Systems*, pp. 1–20, 2021.

[9] Tolga Turay and Tanya Vladimirova, "Toward performing image classification and object detection with convolutional neural networks in autonomous driving systems: A survey," *IEEE Access*, vol. 10, pp. 14076–14119, 2022.

[10] Zhen Zeng, Yunwen Zhou, Odest Chadwicke Jenkins, and Karthik Desingh, "Semantic mapping with simultaneous object detection and localization," in *International Conference on Intelligent Robots and Systems (IROS)*, 2018, pp. 911–918.

[11] Hao Sun, Zehui Meng, Pey Yuen Tao, and Marcelo H. Ang, "Scene recognition and object detection in a unified convolutional neural network on a mobile manipulator," in *International Conference on Robotics and Automation (ICRA)*, 2018, pp. 5875–5881.

[12] Yi Wei, Xinyu Pan, Hongwei Qin, Wanli Ouyang, and Junjie Yan, "Quantization mimic: Towards very tiny cnn for object detection," in *European Conference on Computer Vision (ECCV)*, September 2018, pp. 267–283.

[13] Rundong Li, Yan Wang, Feng Liang, Hongwei Qin, Junjie Yan, and Rui Fan, "Fully quantized network for object detection," in *Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 2810–2819.

[14] Zihao Xie, Li Zhu, Lin Zhao, Bo Tao, Liman Liu, and Wenbing Tao, "Localization-aware channel pruning for object detection," *Neurocomputing*, vol. 403, pp. 400–408, 2020.

[15] Sanjukta Ghosh, Shashi K K Srinivasa, Peter Amon, Andreas Hutter, and André Kaup, "Deep network pruning for object detection," in *International Conference on Image Processing (ICIP)*, 2019, pp. 3915–3919.

[16] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean, "Distilling the knowledge in a neural network," *arXiv preprint arXiv:1503.02531*, 2015.

[17] Guobin Chen, Wongun Choi, Xiang Yu, Tony Han, and Manmohan Chandraker, "Learning efficient object detection models with knowledge distillation," in *Advances in neural information processing systems*, 2017, vol. 30.

[18] Rakesh Mehta and Cemalettin Ozturk, "Object detection at 200 frames per second," in *European Conference on Computer Vision (ECCV) Workshops*, Septemner 2018.

[19] Chien-Yao Wang, Alexey Bochkovskiy, and Hong-Yuan Mark Liao, "Scaled-yolov4: Scaling cross stage partial network," in *Computer Vision and Pattern Recognition (CVPR)*, 2021, pp. 13029–13038.

[20] M. Everingham, S. M. A. Eslami, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes challenge: A retrospective," *International Journal of Computer Vision*, vol. 111, no. 1, pp. 98–136, January 2015.