A New Knowledge Distillation Method for Object Detection Based on EMD

Zhihao Liang
Henan University, China
1824120048@vip.henu.edu.cn

Abstract. As a common method of model compression, the knowledge distillation method can distill the knowledge from the complex large model with strong learning ability to student small model with weak learning ability in the training process, to improve the accuracy and performance of the small model. At present, there has been much knowledge distillation methods specially designed for object detection and achieved good results. However, almost all methods failed to solve the problem of performance degradation caused by the high noise in the current detection framework. In this study, we proposed a feature automatic weight learning method based on EMD to solve these two problems. That is, the EMD method is used to process the space vector to reduce the impact of negative transfer and noise as much as possible, and at the same time, the weights are all located adaptive to reduce student model’s learning from the teacher model with poor performance and make students more inclined to learn from good teachers. The loss (EMD Loss) was redesigned, and the HEAD was improved to fit our approach. We have carried out different comprehensive performance tests on multiple datasets, including PASCAL, KITTI, ILSVRC, and MS-COCO, and obtained encouraging results, which can not only be applied to the one-stage and two-stage detectors but also can be used radiatively with some other methods.

Keywords: Object Detection; Knowledge Distillation; EMD.

1. Introduction
In recent years, with the success of deep learning, outstanding breakthroughs have been made in various fields. Researchers have found that deeper architecture can often improve the performance of neural networks, which can lead to slower detection and higher computational costs. Therefore, the current most advanced deep neural networks often consume a large amount of computation and memory, which limits their application in edge devices such as self-driving cars and mobile phones. To solve this problem, several techniques have been proposed, including pruning, digitization, model compression, and knowledge distillation.

Knowledge distillation, also known as the teacher-student model, seeks to transmit a highly parameterized teacher's information to a student with a lighter network. Since students are trained to imitate the logic or characteristics of the teacher, students can inherit knowledge from the teacher and thus tend to achieve greater accuracy. Because of its simplicity and effectiveness, knowledge distillation has become a popular technique for model compression and improving model accuracy. As one of the
most important challenges in computer vision, object detection requires an accurate and effective model. The characteristic of knowledge distillation can solve the conflict between model validity and data quantity. Unfortunately, in the field of computer vision, most existing knowledge distillation methods are designed for image classification and have little effect on improving object detection.

There are generally more background pixels in the picture to be identified than foreground items. Students are always taught to replicate the features of all pixels in previous knowledge distillation so that they have the same importance. As a result, learners’ attention is primarily focused on learning background pixel properties, preventing them from learning foreground object details. Since the foreground pixel is more significant in detection, this mismatch has a considerable impact on knowledge distillation performance. Some distillation frameworks designed for detection tasks have addressed this issue and achieved impressive results, for example, Li [2] by extracting RPN and Wang et al. [3] sample a certain percentage of positive and negative instances to solve this problem, but the percentage requires careful design and can lead to information loss. In addition, Sergy et al. [4] proposed an approach based on attention mechanisms. They set attention to assign weight to the extracted features. For this paper, we put forward a new learning method. The mechanism of our method is also to assign weights but it’s quite different. Compared with the previous L2loss for the whole feature map, we used EMD to automatically learn the spatial matching coefficient of the feature map to avoid the problem that noise would be introduced into the spatial dimension of the feature map and also avoid the manual tedious and possible errors.

Last but not least, we have modified the HEAD so that it can be successfully connected so that students can learn the output of the teacher. We ran systematic tests on some of the most popular detectors and datasets, finally achieved effective results on each detection dataset and detector.

In general, this paper has the following three aspects of contribution:

1. On the basis of the existing knowledge distillation methods, a new knowledge distillation method is proposed for object detection, which is a very challenging task.

2. Loss (EMD Loss) has been redesigned and HEAD has been improved, which is robust and can be generalized to various scenarios.

3. The EMD method in linear programming is adopted to suppress the negative effects of negative transfer and noise more successfully and effectively. Moreover, this method is self-adaptive and improves the learning efficiency and accuracy of students’ network.

4. Our method shows promising improvement on multiple benchmarks and can be applied to both one-stage detectors and two-stage detectors, both anchor-based detectors and anchor-free detectors.

2. Related Works

2.1. Object Detection

The Object detection is a long-standing fundamental topic in computer vision that has been the subject of ongoing study for decades. Object detection seeks to identify whether a given picture contains target instances of a particular category. If a target instance exists, this method returns its spatial location and coverage. Object detection is the cornerstone for performing more complicated and upper level visual tasks including object tracking, scene comprehension, segmentation, picture description, activity recognition, and event detection, as it is the foundation of pattern recognition and computer vision. Object detection is widely used in many fields such as artificial intelligence and information technology, including but not limited to machine vision, consumer electronics, cybersecurity, human-computer interaction, automated vehicles, content-based picture retrieval, augmented reality, and intelligent surveillance cameras. Deep learning has recently been used to completely eliminate handcrafted characteristics and learn to extract features in an end-to-end way. Object detection has improved dramatically as a result of these strategies. Those methods normally utilize dense prediction formula, which treats target detection as a semantic segmentation job and assigns a label to each pixel in the image to indicate whether it belongs to the target or not. As an example, dual-view CNN is proposed to process object detection. In this method, the images of the front view and top view are taken as the
output. RCN [5] divided object detection into two stages. Specifically, to differentiate between the target pixel and the background pixel, they first conduct binary segmentation. After that, the pixels are split into target instances.

Several solutions have been proposed to complement network-based monitoring and capture a richer context for scenarios, such as information transfer and multitasking learning. For instance, DAC [6] established a framework that can accomplish both target boundary segmentation and target region segmentation at the same time. The framework also includes a common target boundary and a common target region to form geometric constraints on the target to further improve the final performance. PAD [7] used common target tags as additional inputs and integrated Generative Adversarial Networks (GAN) into the original framework to make the segmenting graph more tag-like. Sequential messaging is performed between top-layer outputs to make better use of the structural information. While the above approach does confer additional performance accruals, multitasking learning requires extra annotations, and messaging is ineffective because it delivers information sequentially. In contrast, our method proposed in the paper does not necessitate additional explanations and does not lengthen the time it takes to reason.

2.2. Knowledge Distillation
The Knowledge distillation was originally proposed by Hinton [13] in 2015 to transfer knowledge ranging from huge to tiny networks. In knowledge distillation, a tiny network of students simulates the intermediate outputs and labels of a large network of teachers, and now it has been widely used in the world as an effective means of model compression. In [8], the student and teacher networks share the same capacity and imitate between pairs of layers that have the same dimension. The researchers also looked at the extraction of knowledge from heterogeneous networks. Recent research extends knowledge extraction to attention extraction. For instance, two types of attentional distillation are introduced, namely the activation-based attentional distillation and gradient-based attentional distillation. In both of these extraction methods, the student network is trained by learning the attention map derived from the teacher network. What’s different about our approach is that it doesn’t require a network of teachers. Distillation occurs in a layered, top-down fashion, in which attention knowledge is disseminated layer by layer. This is new in all the literature. It is worth noting that our focus is on investigating the possibility of differentiating hierarchical attention for self-learning. This is different from the existing research using the weight features of visual attention.

What is more, in Fine-grained Feature Imitation [9], based on the feature that the detection model focuses on the local feature, it is assumed that the difference in feature response of the teacher model at different locations reflects the ability of the teacher model, so the student learns this difference to improve the generalization ability of the model. The student has to learn all locations of the teacher as the imitation region, the imitation region here is chosen to be the full feature, which corresponds to the method in hint learning [10], which is found to reduce the performance of the student. Tao et al. [9] analyzed and experimentally verified the reason: the feature of the detection model contains a large amount of background area, which has a large amount of noise information and will overwhelm the teacher’s knowledge, so the paper proposes to estimate the imitation region based on anchor and GT to avoid learning irrelevant information. But our method is adaptive, which better solves the above problems and avoids the introduction of noise in the spatial dimension, as well as the manual tediousness and possible errors.
3. Methods

3.1. Basic framework and method of Object detection

Fasters-RCNN [11] was utilized as the object detection framework in this section. The FAST-RCNN system is made up of three modules: 1) Convolutional layers are used to extract common features, 2) a region proposal network (RPN) creates object suggestions, and 3) a classification and regression network (RCN) returns the detection scores and spatial adjustment vectors for each item recommendation. RCN and RPN both employ the output of 1) as the feature, and RCN also accepts the RPN result as an input.

It is critical to learn robust models of all three components to get high accuracy target detection outcomes. We build a powerful and effective student object detector using the information gained by the three-part high-capacity instructor detection network. Figure 1 depicts our entire learning structure. To begin, we used Hint Learning[12] to improve the inference accuracy of the trunk and simplified multi-classification target detection network (Faster-RCNN was used as an example), and we encouraged the student network’s feature representation to be similar to that of the teacher network. Second, we employ the knowledge distillation architecture in RPN and RCN [13] to train more powerful classification modules. In order to tackle the major problem of foreground and background unbalance in object detection, weighted cross entropy loss (MSE) is applied to the distillation framework. Lastly, we transform the teacher’s regression output into an upper limit, which means the additional loss is discarded if the student’s regression output is better than the teacher’s regression output. The dark knowledge extraction of the teacher network is separated into three points: hints of middle-layer Feature Maps, Dark knowledge of classification layer in RPN/RCN; and the dark knowledge of regression layer in RPN/RCN.

Our general learning objectives can be expressed in the following form:

\[
L_{RCN} = \frac{1}{N} \sum_{i} L_{RCN}^{cls} + \lambda \frac{1}{N} \sum_{i} L_{RCN}^{reg}
\]

\[
L_{RPN} = \frac{1}{M} \sum_{i} L_{RPN}^{cls} + \lambda \frac{1}{M} \sum_{i} L_{RPN}^{reg}
\]
\[ L = L_{RCN} + L_{RPN} + \gamma L_{Hint} \] (3)

RCN’s batch size is \( N \), while RPN’s batch size is \( M \). The classifier loss function \( L_{cls} \) combines the hard soft max loss using ground truth labels and the soft knowledge distillation loss [13] of the classifier (2). In addition, \( L_{reg} \) is the bounding box regression loss, which combines smoothed L1 loss [14] with our newly suggested teacher bounded L2 regression loss of (4). Finally, \( L_{hint} \) stands for the hint-based loss function that motivates the student to imitate the teacher’s feature response, which is written as (6). \( \lambda \) and \( \gamma \) are hyper-parameters that determine the balance between distinct losses in the example above. Throughout the tests, we keep them at 1 and 0.5, respectively.

3.2. EMD for Feature Maps

About Earth Mover’s Distance. Earth Mover’s Distance first appeared in 2000 in an IJCV journal article [15] as a histogram similarity measure for efficiency focused on transportation problems. As a distance measure between two sets of weighted items or distributions, the basic distance between individual objects is built up. It can be formalized to the well-studied optimal transportation problem in discrete form. In particular, assume a collection of sources or suppliers \( S = \{ s_i \mid i = 1, 2, \ldots, m \} \) are required to convey to a set of destinations or demanders \( D = \{ d_j \mid j = 1, 2, \ldots, n \} \), where \( s_i \) signifies the supply units of supplier \( i \) and \( d_j \) denotes the demand of \( j \)-th demander. \( c_{ij} \) denotes the cost per unit transferred from supplier \( i \) to demander \( j \), whereas \( \pi_{ij} \) denotes the transport strategy. Finding the best method \( \pi^* = \{ \pi_{ij} \mid i = 1, \ldots, m, j = 1, \ldots, n \} \) with the lowest transit cost from suppliers to demanders is the answer to the transportation problem. It essentially turns the problem into a linear program that can be solved in polynomial time.

EMD for feature maps. For the application of feature maps, this paper draws lessons from the ideas of C Zhang [16] et al., and adapts them to our method. To compare two feature maps \( X \in \mathbb{R}^{H \times W \times C} \) we flatten them into a set of local representations \( \{ x_1, x_2, \ldots, x_{nm} \} \) with each vector acting as a supplier or demander in the set. Then, the optimal transportation cost between the two sets of vectors can be expressed by the similarity of the two feature maps. Concretely speaking, the cost among two feature nodes \( x_i \) and \( y_j \) is defined as follows:

\[ c_{ij} = 1 - \frac{x_i^T y_j}{\|x_i\| \|y_j\|} \] (4)

By solving the above linear programming issue, we can get the optimal transportation scheme. However, where nodes with comparable representations produce fewer matching costs among themselves. We can determine the similarity score \( s \) among two image feature representations after finding the best transport method \( \pi^* \) with:

\[ s(X, Y) = \sum_{i=1}^{H} \sum_{j=1}^{W} (1 - c_{ij}) \pi_{ij}^* \] (5)

Furthermore, for feature maps, the corresponding loss is defined as:

\[ L_{EMD}, \|X - Y\|_{EMD} = 2 - 2 \cdot s(X, Y), \] (6)
Figure 2. Self-EMD. crops in the form of a spatial pyramid. To produce the spatial pyramid crops, we run many average pooling layers in parallel with various kernel sizes and strides. For better visualization, we project feature map crops back onto the picture.

About Marginal weights. The process of solving $\pi_{i,j}$ is differentiable, so it can be solved by means of gradient descent. The marginal weight of each node (e.g., $s_i$), which regulates the overall transporting units $\pi_{i,j}$ from it, plays an essential part in the solving process. Intuitively, the node with the higher weight in a comparison of two sets is more significant, and vice versa. The weight of each node is set as the dot product between a node feature and the average node feature in the other set in the pioneering works [16] that use EMD for few-shot classification, which is indicated as:

$$s_i = \max\{x_i^T \cdot \frac{\sum_{j=1}^{HW} y_j}{HW}, 0\}.$$  \hspace{1cm} (7)

Because features in supervised classification frequently contain high-level semantic interpretations, the co-occurring areas should be given a lot of weight, while the high-variance background should be given less.

4. Experiments
In various jobs, we carry out experiments. First, we compare our technique with various methods of classification distillation. We explore with various settings and datasets of varied architectures. We also use our technique to recognize objects and to segment instances. Our method also improves the baseline model by large margins consistently.
Table 1. Object detection results. To assess results, we employ AP in a variety of scenarios. MV2 represents MobileNetV2 as backbone, and R101 stands for ResNet101.

| Method                  | mAP  | AP50  | AP75  | API   | APm   | APs   |
|-------------------------|------|-------|-------|-------|-------|-------|
| Teacher Faster R-CNN w/ R101-FPN w/ KD | 39.44 (+1.51) | 60.27 | 43.04 | 51.97 | 42.51 | 22.89 |
| w/ FitNets              | 38.76 (+0.83) | 59.62 | 41.80 | 50.70 | 42.20 | 22.32 |
| w/ FGFI                | 39.44 (+1.51) | 60.27 | 43.04 | 51.97 | 42.51 | 22.89 |
| w/ Our Method          | 40.36 (+2.43) | 60.97 | 44.08 | 52.87 | 43.81 | 23.60 |

| Teacher Faster R-CNN w/ R101-FPN w/ KD | 39.44 (+1.51) | 60.27 | 43.04 | 51.97 | 42.51 | 22.89 |
| w/ FitNets              | 38.76 (+0.83) | 59.62 | 41.80 | 50.70 | 42.20 | 22.32 |
| w/ FGFI                | 39.44 (+1.51) | 60.27 | 43.04 | 51.97 | 42.51 | 22.89 |
| w/ Our Method          | 40.36 (+2.43) | 60.97 | 44.08 | 52.87 | 43.81 | 23.60 |

| Teacher Faster R-CNN w/ R50-FPN w/ KD | 39.44 (+1.51) | 60.27 | 43.04 | 51.97 | 42.51 | 22.89 |
| w/ FitNets              | 38.76 (+0.83) | 59.62 | 41.80 | 50.70 | 42.20 | 22.32 |
| w/ FGFI                | 39.44 (+1.51) | 60.27 | 43.04 | 51.97 | 42.51 | 22.89 |
| w/ Our Method          | 40.36 (+2.43) | 60.97 | 44.08 | 52.87 | 43.81 | 23.60 |

| Teacher RetinaNet101 w/ KD | 36.76 (+0.61) | 56.60 | 39.40 | 48.17 | 40.56 | 21.87 |
| w/ FitNets              | 36.30 (+0.15) | 55.95 | 38.95 | 47.14 | 40.32 | 20.10 |
| w/ FGFI                | 37.29 (+1.14) | 57.13 | 40.04 | 49.71 | 41.47 | 21.01 |
| w/ Our Method          | 38.48 (+2.33) | 58.22 | 41.46 | 51.15 | 42.72 | 22.67 |

4.1. Object Detection

Our approach is also used for other visual duties of your computer. When we identify objects, such as the categorization method, we distill the backbone characteristics between the student and the teacher. The supplemental file contains more details. The most popular, open-source, Detectron2 is our strong basis for evaluating our technique and using the typical COCO2017 dataset. As a teacher, we utilize Detectron2’s best pretrained model. Student models are trained according to tradition using the usual training procedure. On COCO2017 validation set, every performance is assessed. We do two-side and two-way experiments.

Since only a few methods are claimed to be detectable, we reproduce the popular methods and the latest techniques. The comparison is provided in the table. We notice the improvement in detection performance in knowledge distillation techniques, such as KD [13] and FitNets [17]. However, the profit is minimal. FGFI is intended for detection and works better in this job than other techniques. Nevertheless, our technique is far superior to it.

We also differ in the experimental environment for generality checks. We alter backbone topologies in the two-stage technique FasterRCNN. The distillation of information between similar style architectures increases mAP ResNet18 and ResNet50 correspondingly by 3.49 and 2.43. They are large numbers. The baseline is promoted between 29.47 and 33.71 by the distillation between ResNet50 and MobileNetV2 [18]. The distance between the student and the teacher are little on the one-phase detector RetinaNet [19]. The success in difficult object detection tests illustrates our method’s universality and efficiency.

5. Conclusion

In this paper, we have studied feature automatic weight learning method based on EMD to solve these two problems. That is, EMD method is used to process the space vector to reduce the impact of negative transfer and noise as much as possible, and at the same time, weights are allocated adaptive, to reduce
STU model’s learning from the teacher model with poor performance and make students more inclined to learn from good teachers. The loss (EMD loss) has been revised to match our methodology and the HEAD has been enhanced. In addition to the one-stage and two-stage detector capabilities, we have conducted several thorough performance tests, including PASCAL, ILSVRC and MS-COCO, as well as encouragement tests, that may also be utilized with certain other techniques lucratively. We will also use the features inside the stage for future work. In our framework, more loss functions will also be examined.

References

[1] Guy Hadash, Einat Kermany, Boaz Carmeli, Ofer Lavi, George Kour, and Alon Jacovi. Estimate and replace: A novel approach to integrating deep neural networks with existing applications. arXiv preprint arXiv: 1804.09028, 2018.

[2] Quanquan Li, Shengying Jin, and Junjie Yan. Mimicking very efficient network for object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), July 2017.

[3] Tao Wang, Li Yuan, Xiaopeng Zhang, and Jiashi Feng. Distilling object detectors with fine-grained feature imitation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), June 2019.

[4] Zagoruyko S, Komodakis N. Paying More Attention to Attention: Improving the Performance of Convolutional Neural Networks via Attention Transfer. 2016.

[5] George Kour and Raid Saabne. Fast classification of handwritten on-line arabic characters. In Soft Computing and Pattern Recognition (SoCPaR), 2014 6th International Conference of, pages 312–318. IEEE, 2014.

[6] Kaiwen Duan, Song Bai, Lingxi Xie, Honggang Qi, Qingming Huang, and Qi Tian. Centernet: Keypoint triplets for object detection. In Proceedings of the IEEE International Conference on Computer Vision, pages 6569–6578, 2019.

[7] Mark Everingham, Luc Van Gool, C. K. I. Williams, J. Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge, 2010.

[8] T. Furlanello, Z. Lipton, M. Tschanen, L. Itti, and A. Anandkumar. Born-again neural networks. In International Conference on Machine Learning, pages 1602–1611, 2018.

[9] Wang T, Yuan L, Zhang X, et al. Distilling Object Detectors With Fine-Grained Feature Imitation[C]// 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). IEEE, 2020.

[10] Adriana Romero, Nicolas Ballas, Samira Ebrahimi Kahou, Antoine Chassang, Carlo Gatta, and Yoshua Bengio. Fitnets: Hints for thin deep nets. arXiv preprint arXiv:1412.6550, 2014.

[11] S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In Advances in neural information processing systems, pages 91–99, 2015.

[12] A. Romero, N. Ballas, S. E. Kahou, A. Chassang, C. Gatta, and Y. Bengio. Fitnets: Hints for thin deep nets. arXiv preprint arXiv:1412.6550, 2014.

[13] G. Hinton, O. Vinyals, and J. Dean. Distilling the knowledge in a neural network. arXiv preprint arXiv:1503.02531, 2015.

[14] R. Girshick. Fast r-cnn. In Proceedings of the IEEE International Conference on Computer Vision, pages 1440–1448, 2015.

[15] The Earth Mover’s Distance as a Metric for Image Retrieval[M]. Kluwer Academic Publishers, 2000.

[16] Zhang C, Cai Y, Lin G, et al. DeepEMD: Few-Shot Image Classification with Differentiable Earth Mover’s Distance and Structured Classifiers[J]. 2020.

[17] Adriana Romero, Nicolas Ballas, Samira Ebrahimi Kahou, Antoine Chassang, Carlo Gatta, and Yoshua Bengio. Fitnets: Hints for thin deep nets, 2014.

[18] Joseph Redmon and Ali Farhadi. Yolo9000: Better, faster, stronger. In Proceedings of the IEEE
Conference on Computer Vision and Pattern Recognition (CVPR), July 2017.

[19] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In Proceedings of the IEEE International conference on computer vision, pages 2961–2969, 2017.

[20] J. Yim, D. Joo, J. Bae, and J. Kim. A gift from knowledge distillation: Fast optimization, network minimization and transfer learning. In IEEE Conference on Computer Vision and Pattern Recognition, pages 4133–4141, 2017.

[21] Zhuguanyu Wu, Zixiao Hu. "Object detection based on self feature distillation", Journal of Physics: Conference Series, 2021.