Learning Lightweight Face Detector with Knowledge Distillation

Haibo Jin[1], Shifeng Zhang[2], Xiangyu Zhu[2], Yinhang Tang[1], Zhen Lei*[2], and Stan Z. Li[2]

[1]AuthenMetric Inc., Beijing, China
[2]CBSR & NLPR, Institute of Automation, Chinese Academy of Sciences, Beijing, China
haibo.nick.jin@gmail.com, yinhang.tang@authenmetric.com
{shifeng.zhang, xiangyu.zhu, zlei, szli}@nlpr.ia.ac.cn

Abstract

Despite that face detection has progressed significantly in recent years, it is still a challenging task to get a fast face detector with competitive performance, especially on CPU based devices. In this paper, we propose a novel loss function based on knowledge distillation to boost the performance of lightweight face detectors. More specifically, a student detector learns additional soft label from a teacher detector by mimicking its classification map. To make the knowledge transfer more efficient, a threshold function is designed to assign threshold values adaptively for different objectness scores such that only the informative samples are used for mimicking. Experiments on FDDB and WIDER FACE show that the proposed method improves the performance of face detectors consistently. With the help of the proposed training method, we get a CPU real-time face detector that runs at 20 FPS while being state-of-the-art on performance among CPU based detectors.

1. Introduction

Face detection is an essential problem because it is the preceding task of many computer vision applications such as face tracking [9], face alignment [36] and face recognition [37]. Due to the development of convolutional neural networks (CNNs), the performance of face detection has been improved significantly in recent years. However, existing face detectors are usually redundant and computationally inefficient because they utilize large CNNs to maintain superior performance. Although one-stage based detectors such as SSD [16] and YOLO [22] have been designed to accelerate detectors, they are still not fast enough for industrial use, especially for CPU based environments. On the other hand, the performance of detectors drop quickly when the channels of CNNs are reduced to meet speed requirements. Thus, it is a challenging task to get a lightweight face detector with satisfactory performance.

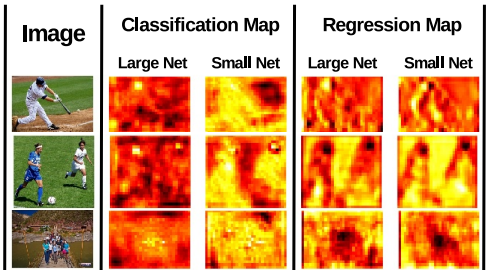


Figure 1: Visualization of classification map samples and regression map samples of a large detector and a small detector trained with regular detection loss. The maps on the same row are from the same anchor.

This work aims to improve the performance of lightweight face detectors with knowledge distillation (KD). Knowledge distillation [1, 6] is a promising method to improve the performance of small networks by mimicking the outputs of large networks. The effectiveness of KD has been validated on classification [1, 6] as well as metric learning [4] tasks. For detection task, it is not straightforward to transfer knowledge from a teacher network to a student network because the outputs of detectors have class imbalance problem. Specifically, the outputs of a detector usually contain a few samples from foreground class and are dominated by the samples from background class. Therefore, simply mimicking all the outputs of the last layer (as in classification and metric learning tasks) will lead to poor model performance. Few works have applied KD to detection problems except [12, 3]. The work [12] proposes to mimic through the feature map right before classification and regression modules. Besides regular detection loss, their student network mimics its region proposal area on the feature map to transfer knowledge from the teacher network during training. Another work [3] proposes a weighted cross-entropy loss to address the class imbalance problem

*Corresponding author.

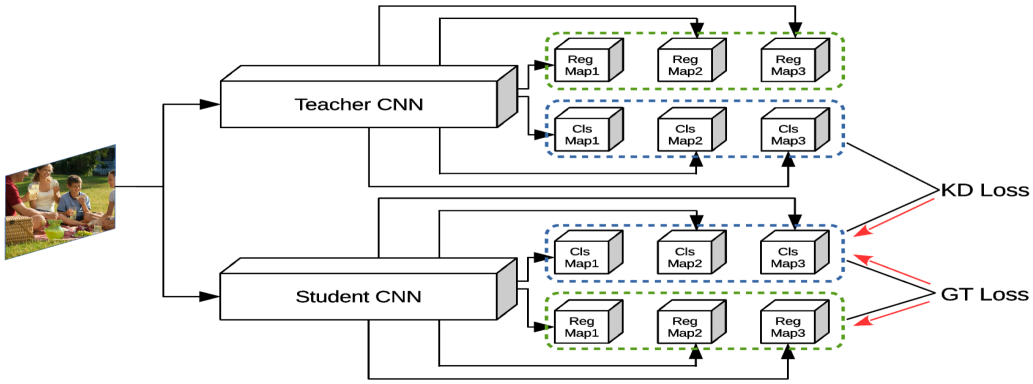


Figure 2: The overall architecture of our face detection model. Red arrows indicate the paths of backpropagation.

when applying KD to two-stage based detectors such as Faster-RCNN [23]. Most lightweight face detectors adopt one-stage approach rather than two-stage approach because the former is more computationally efficient. The last layer of a typical one-stage based detector consists of a classification map and a regression map, used for predicting objectness scores and bounding box coordinates respectively. The class imbalance problem is even more severe in one-stage detectors because they do not have region proposal module to reduce negative samples as in two-stage detectors.

In this paper, we see knowledge distillation (KD) loss as a complement to regular detection loss. In other words, the KD method should make full use of the supervisory signals that differ from ground-truth (GT) labels. We choose the classification map as the connection module for transferring knowledge according to the following reasons. (1) Compared to the feature layers, the classification map gives outputs with a clear physical meaning which is helpful for selecting appropriate samples. (2) Compared to the regression map, the classification map contains much better soft labels. To be more specific, the outputs of classification map are real numbers ranging from zero to one, which are more accurate and smoother than the GT labels marked as either one or zero. On the other hand, the GT labels of regression map are real numbers originally, thus the regression map of teacher network does not give more valuable information. Figure 1 visualizes some samples of regression maps and classification maps of a large net and a small net, both of which are trained with regular detection loss. We can see from the figure that the two models have a much larger gap on classification map than regression map, indicating that the classification map of large net is more worth learning.

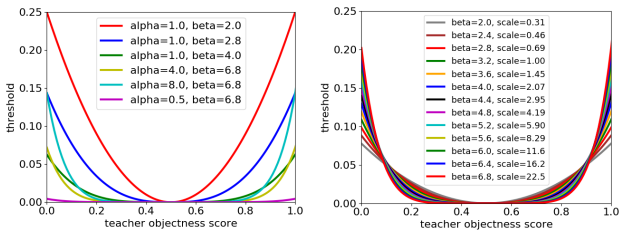
To improve the performance of lightweight face detectors, we propose a new loss function based on KD method. The loss function is a threshold based L2 loss, where the threshold varies for different objectness scores of teacher, see Figure 3(a). Among all the objectness scores in a mini-batch, we only compute L2 loss for the selected ones. A pair of scores (scores from the same index of teacher and student) will be selected if the absolute difference between student objectness score and teacher objectness score exceeds the corresponding threshold. As can be seen from the figure, the function is of lower values for the scores closer to 0.5, giving priority to the scores that differ more from GT labels. As for the scores close to zero or one, they will also be included if the difference is relatively significant. It is worth noting that the proposed loss function naturally handles the class imbalance problem. Simply speaking, imbalance problem is caused by the overwhelming easy negative samples. An easy negative sample will yield a quite small objectness score in both teacher and student, which is not likely to be selected for mimicking by our U-shaped function. Experiments on popular face detection datasets show that our proposed KD loss is able to improve the accuracy of lightweight face detectors considerably.

2. Our Model

2.1. Architecture

Our detection framework is based on SSD [16]. SSD is a one-stage based detection framework with competitive performance to two-stage based detectors. It splits bounding boxes into a set of anchors over different scales of feature maps, which is the key of its success. When equipped with efficient base networks, SSD can be a CPU real-time face detector with satisfactory performance [33].

Figure 2 shows the overall architecture of our model. During training, there are two networks, namely student and teacher. The teacher network is usually larger than the student, thus it has better performance. The parameters of the teacher are frozen while the parameters of the student are updated through two parts of loss functions: GT detection loss and KD loss. The loss function of our model is $L = L_{GT} + \lambda L_{KD}$, where L_{GT} is ground-truth detection loss, L_{KD} is knowledge distillation loss and λ is a scalar to balance the two losses. Following the region proposal network in Faster R-CNN [23], the GT loss is $L_{GT} = L_{cls} + L_{reg}$, where L_{cls} is a two-class Softmax loss



(a) (b)

Figure 3: Visualization of threshold functions. (a) Threshold functions with different α (i.e., scale) and β. (b) Threshold functions with different β and fixed $\gamma = 3.2$.

for classification and L_{reg} is smooth L1 loss for regression. We introduce the KD loss in the next subsection.

2.2. Knowledge Distillation for Face Detection

As mentioned in Section 1, the classification map of a single stage detector is a proper place to transfer knowledge. Thus, a straightforward way is to mimic the whole classification map:

$$L_{KD} = \frac{1}{|S|} \sum_{i \in S} \|p_i - q_i\|_2^2, \quad (1)$$

where p_i is the ith objectness score in the classification map of teacher, q_i is the ith score in the classification map of student, S contains all the indices of classification map in a mini-batch and $|S|$ represents the number of indices in S. However, classification map is usually overwhelmed by the samples from background class, and simply mimicking the whole map leads to high false negative predictions. Therefore, the core of applying knowledge distillation is to decide which indices should be included in set S.

A threshold based loss function is proposed in this paper, which acts as a filter of the objectness scores. The new loss is able to select more informative samples so that the small network can be better optimized because it gets better supervisory signals. More concretely, the threshold θ is a function of teacher objectness score:

$$\theta_i = f(p_i; \alpha, \beta) = \alpha(|p_i - 0.5|)^\beta, \quad p_i \in [0, 1] \quad (2)$$

where p_i is a objectness score of teacher network, α is a hyper-parameter that controls the scale of the function and β is a hyper-parameter that controls the shape of the function. Then we select indices according to the above function:

$$S = \{i \mid |p_i - q_i| > \theta_i\}. \quad (3)$$

Figure 3(a) visualizes the threshold functions for several values of α and β. From the figure, we can see a property of the function: when a teacher objectness score is closer to 0.5, its corresponding threshold is relatively lower such that the score is more likely to be added to set S. The reason behind such design is that, the scores closer to 0.5 are more informative samples because they have larger difference from GT labels[1]. Therefore, the proposed threshold function is able to only select informative samples for mimicking to make the knowledge transfer more efficient. Moreover, the class imbalance problem is implicitly solved because the overwhelming easy negative samples will not be considered as informative samples.

Model	FB-1-GT	FBI-1-GT	FBI-1/2-GT
TPR(%)	95.4	96.2	94.1
FPS	20	20	40

Table 1: True positive rate (TPR) of original FB model and our FBI models on FDDB at 1000 false positives.

The hyper-parameters α and β smoothly adjust the function to suit plentiful situations. In practice, we choose β from a discrete set of values (e.g., 2.0, 3.6 and 6.8) while the range of α can be $[0, \infty)$. It is not difficult to see that certain combinations of α and β result in unreasonable threshold functions (e.g., a large β with a small α, as the purple function in Figure 3(a)). In order to avoid such functions that have too large or too small area, we decide to introduce a standard function area $A_{1,\gamma}$ as the normalization term of scale. Then, Equation 2 can be simplified as follows:

$$\theta_i = g(p_i; \beta, \gamma) = \frac{A_{1,\gamma}}{A_{1,\beta}}(|p_i - 0.5|)^\beta, \quad p_i \in [0, 1] \quad (4)$$

where $A_{1,\beta}$ is the area of the function $f(p_i; 1, \beta)$ between zero and one, and $A_{1,\gamma}$ is the area of the function $f(p_i; 1, \gamma)$ between zero and one. Now the term $\frac{A_{1,\gamma}}{A_{1,\beta}}$ replaces α, and it chooses the scale adaptively according to β and γ so that the area of the function between zero and one is always equal to $A_{1,\gamma}$. The candidate values of γ can be just the same as the discrete set of β. As long as γ is properly selected, the new scale will always be reasonable for specific β. For example, if γ is set to be 2.0 and $\beta = 4.0$, then the scale will be $\frac{A_{1,2.0}}{A_{1,4.0}} = 6.67$. Intuitively, it just means that the scale should increase as β increases (compared to γ), or vice versa.

2.3. Implementation Details

We adopt an improved version of FaceBoxes (FB) [33] as the primary base network in this paper. FaceBoxes achieves superior performance on face detection while being efficient on CPU based devices. Compared to the standard SSD, FaceBoxes further designs rapidly digested convolutional layers, multiple scale convolutional layers and anchor densification strategy to boost the performance of a lightweight network. Building upon the architecture, our version replaces the first two C.ReLU convolutional layers with five thinner standard convolutional layers to make it deeper. We

[1]The statement applies to most cases. Although there are exceptional samples that are not close to 0.5 but have large difference from GT labels (i.e., the prediction of teacher is incorrect), the number of such samples is relatively rare and their effect can be ignored.

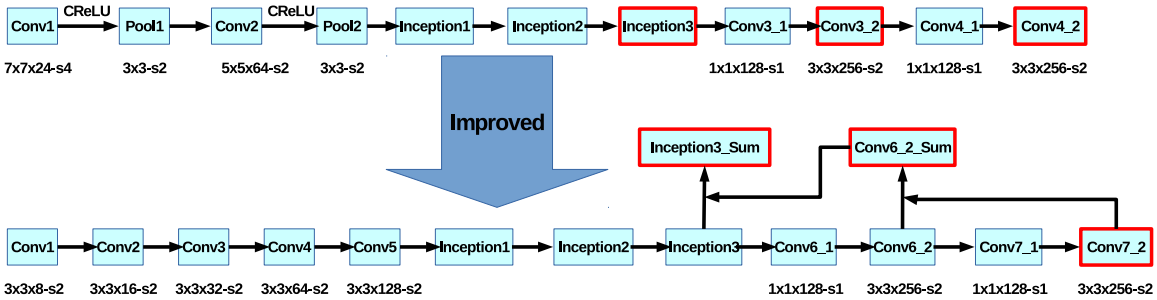


Figure 4: Differences between original FB model and our FBI model. Red blocks are the feature maps associated with prediction module.

FBI-1/2-KD														FBI-1/2-GT
Scale	0.31	0.46	0.69	1.00	1.45	2.07	2.95	4.19	5.90	8.29	11.6	16.2	22.5	N/A
β	2.0	2.4	2.8	3.2	3.6	4.0	4.4	4.8	5.2	5.6	6.0	6.4	6.8	
TPR(%)	95.0	95.0	94.8	94.8	94.9	94.7	95.0	95.0	94.7	95.0	95.0	**95.1**	94.7	94.1

Table 2: Results of FBI-1/2-KD with different β and fixed $\gamma = 3.2$. The corresponding scale value of each β is also given when $\gamma = 3.2$.

also utlize a lightweight version of Feature Pyramid Network (FPN) [15] module to enrich the semantic information of lower feature layers. We name the modified version FaceBoxesImproved (FBI). Figure 4 shows the difference between the two networks. Table 1 gives the performance of the two versions on FDDB. As can be seen, our improved version promotes the true positive rate (TPR) by 0.8% while being as fast as the original model.

The models are trained with SGD, in which the momentum is set to 0.9. The parameters are randomly initialized with "Xavier" method. We use 0.0005 weight decay and 32 batch size. The number of total training iterations is 160k, where the learning rate is 10^{-3} for the first 120k, 10^{-4} for 120k to 140k, and 10^{-5} for the last 20k. We fix the balance parameter λ to 50 so that the value of KD loss is comparable to that of GT loss. The code is implemented in Caffe.

3. Experiments

In this section, we first comprehensively analyze the proposed KD method from several perspectives, then we report the performance of our model on two popular face detection datasets FDDB and WIDER FACE, respectively.

3.1. Model Analysis

The experiments of model analysis are carried out on FDDB dataset because it is a representative face detection benchmark dataset. We denote the models with format Network-N-method. For example, FBI-4-GT refers to a FBI network that has 4 times of channels on each layer of a standard FBI trained with GT loss; FBI-1-KD refers to standard FBI network trained with the proposed KD method.

3.1.1 The Hyper-parameters

Table 1 gives the performance of two FBI models trained with only GT loss. When the number of channels of standard FBI is reduced to a half, its performance drops 2.1%, which is a large gap on FDDB. Training with the proposed KD method can narrow the gap from cutting channels. After simplification, β and γ are two hyper-parameters of the proposed method. Empirically, we find that $\gamma = 3.2$ works the best, although some other choices (e.g., 2.4, 2.8, and 3.6) also work almost as well. Figure 3(b) visualizes the thereshold functions of different β when $\gamma = 3.2$, and Table 2 shows the corresponding results. From the table, we see that the performance of the proposed KD model has stable improvements across a wide range of β when compared to the GT version. Since $\beta = 6.4$ works the best in Table 2, *we set* $\beta = 6.4$ *and* $\gamma = 3.2$ *for the rest experiments, unless otherwise stated.*

Moreover, by setting $\beta = 0$ and changing the values of α, we get a series of functions giving the same thresholds for all the scores, which means that no priority is given to middle scores. Table 3 shows the results of such functions. As can be seen, none of them outperforms the results in Table 2, indicating that more attention shoud be paid to the scores in the middle. When $\alpha = 0$ and $\beta = 0$, it actually mimics all the scores, and its result tells us that such straightforward method does not work.

α	0.0	0.1	0.2	0.3	0.4	0.5	0.6
β	0.0	0.0	0.0	0.0	0.0	0.0	0.0
TPR(%)	88.4	94.5	94.4	94.4	94.2	94.2	94.1

Table 3: Results of FBI-1/2-KD with different α and $\beta = 0$.

3.1.2 Teacher Network

Intuitively, a better teacher network should learn better soft labels and give larger improvement to a student with KD method. In this part, we use a set of teacher networks with various model capacities, namely FBI models with 1/2/4 times the number of channels. Table 4 shows the performance of the teacher networks. Compared to FBI-1-GT,

Teacher	TPR(%) of Teacher	TPR(%) of Student
FBI-1-GT	96.2	95.1
FBI-2-GT	97.3	95.4
FBI-4-GT	97.3	95.4

Table 4: Results of teacher networks and the student FBI-1/2-KD trained with different teachers.

FBI-2-GT achieves a better result as its number of channels is doubled. However, the performance does not increase any more when the number of channels is further doubled. We then use these teachers to do KD training for a 1/2 FBI network, and the results can be seen in Table 4. As we can see, a better teacher does give a larger improvement. *Since FBI-2-GT already gives the best result, we use FBI-2-GT as the teacher network in the rest experiments.*

3.1.3 Visualization

To better understand how KD method helps student learn from teacher, we visualize the classification map of the teacher FBI-2-GT, the student FBI-1/2-KD as well as FBI-1/2-GT in Figure 5. The first column displays the original image, and the second to the fourth column display a foreground class classification map sample of the teacher, the student without KD and the student with KD, respectively. It is easy to see that the student without KD cannot learn the classification map well by itself, and its maps have significant difference from that of teacher. Also, we find that smaller networks tend to have more false positives. After applying KD method, the visual quality of the maps of the student has been greatly improved, and the maps are quite similar to the ones of the teacher. The sample images in Figure 5 are from the validation set of WIDER FACE, which are not included in the training data. Therefore, it indicates that the distilled knowledge also generalizes to unseen data.

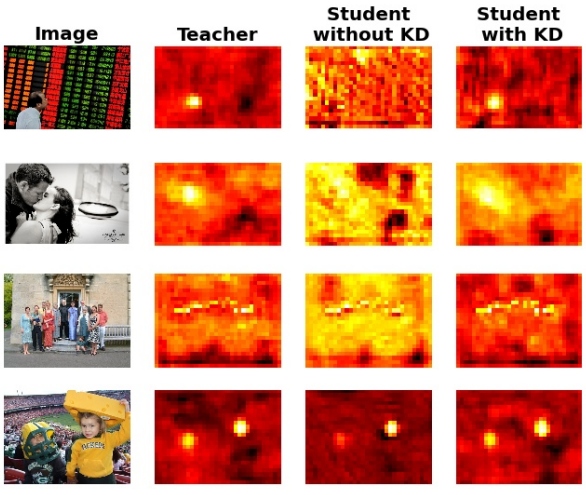


Figure 5: Visualization of classification map samples of teacher network, student without KD and student with KD. The maps on the same row are from the same anchor.

Method	TPR(%)	FPS
ACF [26]	85.2	20
CasCNN [11]	85.7	14
FaceCraft [21]	90.8	10
STN [2]	91.5	10
MTCNN [31]	94.4	16
FaceBoxes [33]	95.4	20
ICC-CNN [32]	96.5	12
FBI-1-KD (ours)	**96.8**	20
FBI-1/2-KD (ours)	95.4	**40**

Table 5: Accuracy and efficiency of our models and state-of-the-art CPU based face detectors. The TPR is measured on FDDB at 1000 false positives. Note that the original FB [33] reports 96.0% based on multi-scale testing while we use single scale testing (3× of the input) for FDDB.

3.2. Results on FDDB

FDDB [8] dataset consists of 5,171 faces from 2,845 images, which are collected from Yahoo news. In this part, we report the performance of our model on FDDB. Since the focus of this work is lightweight face detectors, we first compare our models with the state-of-the-art CPU based face detectors on both accuracy and efficiency in Table 5. Our measurements are based on VGA-resolution image. We use Intel Xeon E5-2660v3@2.60GHz as CPU during testing. Following FaceBoxes [33], we filter the predicted bounding boxes with confidence threshold 0.05, and keep top 400 of them. After applying NMS with jaccard overlap 0.3, we keep the top 200 boxes. We also use the standard FBI as student besides the half-channel FBI. From the table, we can see that our FBI-1-KD model achieves the best performance among the existing CPU based face detectors and it still runs at 20 FPS on CPU. Our faster model FBI-1/2-KD runs at 40 FPS while its performance beats most other slower detectors. It is worth noting that the performance of FBI-1/2-KD even catches up with the original FB model

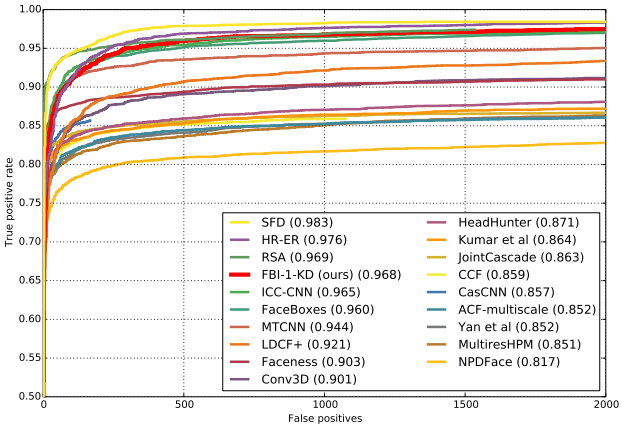
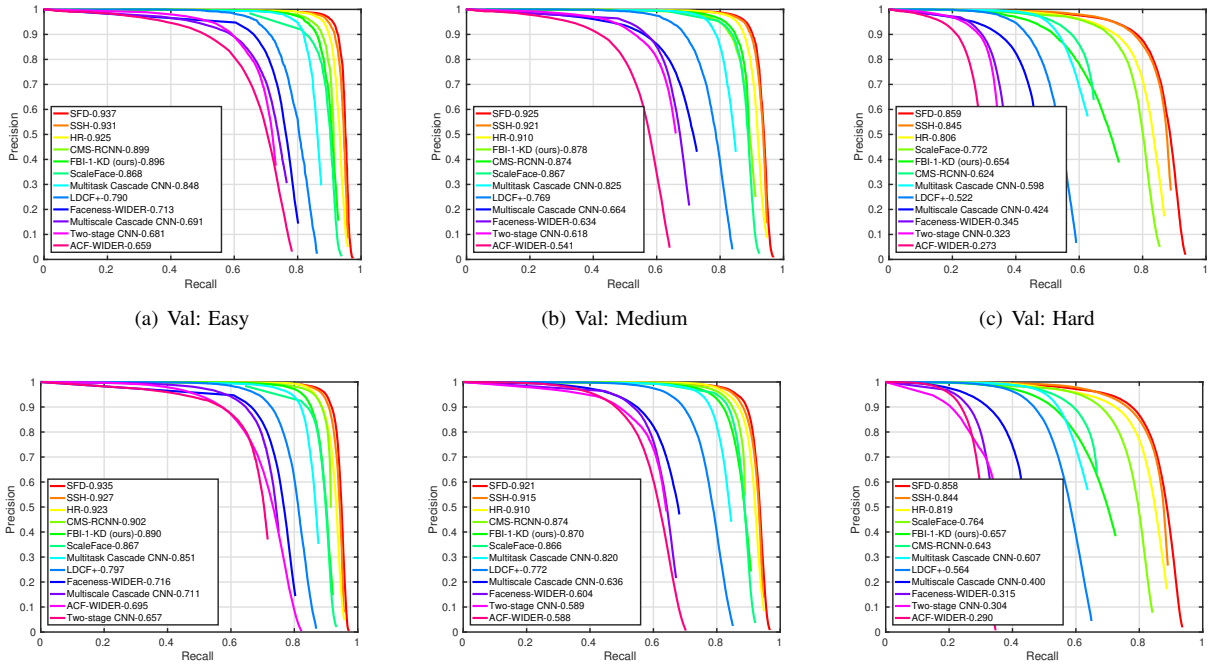


Figure 6: Discrete ROC curves of our FBI-1-KD and the state-of-the-art models on FDDB.



(a) Val: Easy (b) Val: Medium (c) Val: Hard

(d) Test: Easy (e) Test: Medium (f) Test: Hard

Figure 7: Precision-recall curves on WIDER FACE validation and testing sets.

while being two times faster.

We compare our FBI-1-KD model to state-of-the-art methods [34, 7, 17, 32, 33, 31, 20, 28, 13, 18, 10, 2, 27, 11, 26, 25, 5, 14, 24] on FDDB and the discrete ROC curves can be seen in Figure 6. The performance of our model is still competitive among the existing best models, where most of them are too heavy to run on CPU.

3.3. Results on WIDER FACE

WIDER FACE [29] dataset consists of 393,703 faces from 32,203 images, with large variations on pose, occlusion, scale and illumination. According to the difficulty of detection, the images are divided into three levels: easy, medium and hard. The dataset is randomly split into three parts, namely training set (40%), validation set (10%) and testing set (50%). Among the three sets, the bounding box ground truth of the testing set is not released, and the users are required to submit the final predictions to the authors to get evaluation result. In this paper, we train the models only on WIDER FACE training set, so we evaluate our models on both WIDER FACE validation and testing set. We use single scale testing ($2\times$ of the input) for WIDER FACE.

To show the effectiveness of the proposed KD method, we first evaluate our KD models against the corresponding GT models on WIDER FACE validation set. From Table 6, it can be seen that the models with KD outperform the corresponding GT versions consistently on all the three levels.

We compare our FBI-1-KD model with state-of-the-art models [34, 19, 7, 35, 30, 31, 20, 28, 29, 26] on WIDER FACE validation and testing sets in Figure 7. Our CPU based model is still competitive among the best models, and it even outperforms several GPU based methods such as ScaleFace and Faceness.

Method	Easy	Medium	Hard
FBI-1/2-GT	84.1	81.4	58.0
FBI-1/2-KD	86.2 (+2.1)	83.2 (+1.8)	58.6 (+0.6)
FBI-1-GT	88.5	86.8	65.1
FBI-1-KD	89.6 (+1.1)	87.8 (+1.0)	65.4 (+0.3)

Table 6: Comparison of the GT method and the proposed KD method on WIDER FACE validation set.

4. Conclusion

In this work, we propose a novel training method based on KD to improve the performance of lightweight face detectors. The proposed method is carefully designed to transfer knowledge from teacher to student through the classification map of single stage detectors. The experiments on FDDB and WIDER FACE show that our method improves the performance of face detectors consistently compared to several baselines. Notably, a face detector trained with our method achieves 96.8% TPR (at 1000 false positives) on FDDB while it still runs at 20 FPS on CPU.

Acknowledgements

This work was supported by the National Key Research and Development Plan (Grant No.2016YFC0801002), the Chinese National Natural Science Foundation Projects #61876178, #61806196, #61872367, #61572501 and AuthenMetric R&D Funds.

References

[1] L. J. Ba and R. Caruana. Do deep nets really need to be deep? In *NIPS*, 2014.

[2] D. Chen, S. Ren, Y. Wei, X. Cao, and J. Sun. Joint cascade face detection and alignment. In *ECCV*, 2014.

[3] G. Chen, W. Choi, X. Yu, T. Han, and M. Chandraker. Learning efficient object detection models with knowledge distillation. In *NIPS*, 2017.

[4] Y. Chen, N. Wang, and Z. Zhang. Darkrank: Accelerating deep metric learning via cross sample similarities transfer. *arXiv: 1707.01220*, 2017.

[5] G. Ghiasi and C. C. Fowlkes. Occlusion coherence: Detecting and localizing occluded faces. *arXiv: 1506.08347*, 2015.

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

[7] P. Hu and D. Ramanan. Finding tiny faces. In *CVPR*, 2017.

[8] V. Jain and E. Learned-Miller. Fddb: A benchmark for face detection in unconstrained settings. Technical report, UMass Amherst, 2010.

[9] M. Kim, S. Kumar, V. Pavlovic, and H. Rowley. Face tracking and recognition with visual constraints in real-world videos. In *CVPR*, 2008.

[10] V. Kumar, A. Namboodiri, and C. V. Jawahar. Visual phrases for exemplar face detection. In *ICCV*, 2015.

[11] H. Li, Z. Lin, X. Shen, J. Brandt, and G. Hua. A convolutional neural network cascade for face detection. In *CVPR*, 2015.

[12] Q. Li, S. Jin, and J. Yan. Mimicking very efficient network for object detection. In *CVPR*, 2017.

[13] Y. Li, B. Sun, T. Wu, and Y. Wang. Face detection with end-to-end integration of a convnet and a 3d model. In *ECCV*, 2016.

[14] S. Liao, A. K. Jain, and S. Z. Li. A fast and accurate unconstrained face detector. *IEEE TPAMI*, 2015.

[15] T.-Y. Lin, P. Dollr, R. Girshick, K. He, B. Hariharan, and S. Belongie. Feature pyramid networks for object detection. In *CVPR*, 2017.

[16] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg. Ssd: Single shot multibox detector. In *ECCV*, 2016.

[17] Y. Liu, H. Li, J. Yan, F. Wei, X. Wang, and X. Tang. Recurrent scale approximation for object detection in cnn. In *ICCV*, 2017.

[18] M. Mathias, R. Benenson, M. Pedersoli, and L. V. Gool. Face detection without bells and whistles. In *ECCV*, 2014.

[19] M. Najibi, P. Samangouei, R. Chellappa, and L. Davis. Ssh: Single stage headless face detector. In *ICCV*, 2017.

[20] E. Ohn-Bar and M. M. Trivedi. To boost or not to boost? on the limits of boosted trees for object detection. In *ICPR*, 2016.

[21] H. Qin, J. Yan, X. Li, and X. Hu. Joint training of cascaded cnn for face detection. In *CVPR*, 2016.

[22] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi. You only look once: Unified, real-time object detection. In *CVPR*, 2016.

[23] S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *NIPS*, 2015.

[24] V. A. Sindagi and V. M. Patel. Dafe-fd: Density aware feature enrichment for face detection. In *WACV*, 2019.

[25] J. Yan, Z. Lei, L. Wen, and S. Z. Li. The fastest deformable part model for object detection. In *CVPR*, 2014.

[26] B. Yang, J. Yan, Z. Lei, and S. Z. Li. Aggregate channel features for multi-view face detection. In *IJCB*, 2014.

[27] B. Yang, J. Yan, Z. Lei, and S. Z. Li. Convolutional channel features. In *ICCV*, 2015.

[28] S. Yang, P. Luo, C. C. Loy, and X. Tang. From facial parts responses to face detection: A deep learning approach. In *ICCV*, 2015.

[29] S. Yang, P. Luo, C. C. Loy, and X. Tang. Wider face: A face detection benchmark. In *CVPR*, 2016.

[30] S. Yang, Y. Xiong, C. C. Loy, and X. Tang. Face detection through scale-friendly deep convolutional networks. *arXiv: 1706.02863*, 2017.

[31] K. Zhang, Z. Zhang, Z. Li, and Y. Qiao. Joint face detection and alignment using multitask cascaded convolutional networks. *SPL*, 2016.

[32] K. Zhang, Z. Zhang, H. Wang, Z. Li, Y. Qiao, and W. Liu. Detecting faces using inside cascaded contextual cnn. In *ICCV*, 2017.

[33] S. Zhang, X. Zhu, Z. Lei, H. Shi, X. Wang, and S. Z. Li. Faceboxes: A cpu real-time face detector with high accuracy. In *IJCB*, 2017.

[34] S. Zhang, X. Zhu, Z. Lei, H. Shi, X. Wang, and S. Z. Li. S^3fd: Single shot scale-invariant face detector. In *ICCV*, 2017.

[35] C. Zhu, Y. Zheng, K. Luu, and M. Savvides. Cms-rcnn: Contextual multi-scale region-based cnn for unconstrained face detection. *arXiv: 1606.05413*, 2016.

[36] X. Zhu, Z. Lei, X. Liu, H. Shi, and S. Z. Li. Face alignment across large poses: A 3d solution. In *CVPR*, 2016.

[37] X. Zhu, Z. Lei, J. Yan, D. Yi, and S. Z. Li. High-fidelity pose and expression normalization for face recognition in the wild. In *CVPR*, 2015.