Task Decoupled Knowledge Distillation For Lightweight Face Detectors

Conference Paper · October 2020

DOI: 10.1145/3394171.3414069

Xiaoqing Liang
Chinese Academy of Sciences
4 PUBLICATIONS **29** CITATIONS
SEE PROFILE

Xu Zhao
Chinese Academy of Sciences
26 PUBLICATIONS **627** CITATIONS
SEE PROFILE

Chaoyang Zhao
48 PUBLICATIONS **948** CITATIONS
SEE PROFILE

Ming Tang
Chinese Academy of Sciences
134 PUBLICATIONS **4,373** CITATIONS
SEE PROFILE

Task Decoupled Knowledge Distillation For Lightweight Face Detectors

Xiaoqing Liang[1,2,*], Xu Zhao[1,3,*], Chaoyang Zhao[1,3], Nanfei Jiang[1,2],
Ming Tang[1,2], and Jinqiao Wang[1,2,4]
[1]National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences,
Beijing, 100190, China.
[2]School of Artificial Intelligence, University of Chinese Academy of Sciences, Beijing, 100049, China.
[3]ObjectEye Inc., Beijing, China.
[4]NEXWISE Co., Ltd., Guangzhou, China.
{xiaoqing.liang,xu.zhao,chaoyang.zhao,nanfei.jiang,tangm,jqwang}@nlpr.ia.ac.cn

ABSTRACT

Face detection is a hot topic in computer vision. The face detection methods usually consist of two subtasks, i.e. the classification subtask and the regression subtask, which are trained with different samples. However, current face detection knowledge distillation methods usually couple the two subtasks, and use the same set of samples in the distillation task. In this paper, we propose a task decoupled knowledge distillation method, which decouples the detection distillation task into two subtasks and uses different samples in distilling the features of different subtasks. We firstly propose a feature decoupling method to decouple the classification features and the regression features, without introducing any extra calculations at inference time. Specifically, we generate the corresponding features by adding task-specific convolutions in the teacher network and adding adaption convolutions on the feature maps of the student network. Then we select different samples for different subtasks to imitate. Moreover, we also propose an effective probability distillation method to joint boost the accuracy of the student network. We apply our distillation method on a lightweight face detector, EagleEye[37]. Experimental results show that the proposed method effectively improves the student detector's accuracy by 5.1%, 5.1%, and 2.8% AP in Easy, Medium, Hard subsets respectively.

CCS CONCEPTS

• **Computing methodologies → Object detection**.

KEYWORDS

face detection, knowledge distillation, lightweight detector, model compression, model acceleration

ACM Reference Format:
Xiaoqing Liang, Xu Zhao, Chaoyang Zhao, Nanfei Jiang, and Ming Tang, and Jinqiao Wang. 2020. Task Decoupled Knowledge Distillation For Lightweight Face Detectors. In *Proceedings of the 28th ACM International Conference on Multimedia (MM'20), October 12–16, 2020, Seattle, WA, USA.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3394171.3414069

1 INTRODUCTION

In recent years, the breakthrough of convolutional neural networks (CNN) in computer vision has led to the upsurge of deep learning. With the powerful ability of CNN to extract features, CNN-based general object detectors have developed rapidly. Face detectors are usually based on one-stage general object detectors[12, 13]. To improve the performance of the face detector, researchers have explored in various aspects. Some methods [3, 26, 33] improve performance by designing better backbone or head structures of the detector. Others [4, 27] improve the accuracy through multi-task joint training. Apart from them, some work [14] focus on designing better anchor mining. Although the performance of face detectors has been greatly improved by these methods, the dramatic increase in the amount of calculations has hindered its further deployment. In practice, face detection is usually deployed in devices with low computing power. Therefore, many lightweight face detectors are proposed, including FaceBoxes [35], MTCNN [34], and EagleEye [37]. To keep lightweight, their complexity is usually suppressed and their accuracy is lower than the heavyweight face detectors (such as PyramidBox [26], SRN [3], and FANet [33]). Therefore, improving the accuracy of lightweight detectors is an important topic.

Knowledge distillation is widely used in model compression and transfer learning. It can transfer the knowledge of the cumbersome model to the lightweight model without introducing any additional calculations, thereby improving the performance of the lightweight model. Since Hinton et al. propose knowledge distillation for learning the similarities between probability in the teacher-student paradigm [8], a vast number of methods [22, 32] for knowledge distillation have emerged. Overall, they are mainly divided into two categories, one is learning from probability [8, 10], and the other [29, 38] is imitating the pixels on the feature map. They enhance the student models by minimizing the distance between the teacher network and the student network under a certain measure.

The above methods mainly focus on the image classification task. Some works also discuss knowledge distillation in the object detection task. Most of these distillation methods make the student

[*]equal contribution.

MM '20, October 12–16, 2020, Seattle, WA, USA

ACM ISBN 978-1-4503-7988-5/20/10…$15.00
https://doi.org/10.1145/3394171.3414069

network imitate the feature map of the teacher network [2, 29, 38]. The difference lies in which feature maps and which pixel points on the feature map are selected. Although these methods have made some improvements in the object detection task, they ignore the difference in the samples required by the classification subtask and regression subtask in the whole detection task. During the training of the detectors, the classification subtask is trained with both positive samples and negative samples. Only in this way can the classifier distinguish whether the area belongs to the background or the foreground. The regression subtask is trained only with positive samples. Since the role of the regressor is to adjust the position of the positive sample so that it fits the corresponding ground truth boxes better. In these distillation methods, the features of regression and classification are both mixed in the same feature map. Directly imitating the pixels on the feature map implies that they select the same sample for distillation for the both regression and classification subtasks, which introduces the sample noises for distillation. Therefore, when performing knowledge distillation, different samples should be considered for the two subtasks. In most of the face detectors, each of the two subtasks use a convolution layer to generate the prediction results. They share the same input feature map. In this paper, we think the feature map should be decoupled for the two tasks to be distilled. After that, we choose different samples to distill according to the different sampling strategies of the two subtasks in the detector training.

In this paper, we propose a task decoupled distillation framework that can distill face detectors with different samples for different subtasks. First, we propose a novel feature decoupling method to decouple the mixed features into classification features and the regression features without introducing any extra calculations at inference. To be specific, in the teacher network, we generate the corresponding task-specific features by adding task-specific convolutions. While in the student network, we convert the mixed feature to corresponding task-specific features by adaption convolutions. Then we propose an innovative sampling strategy that guides us to select more appropriate sample for each subtask. Specifically, for the regression subtask, we only select pixels of positive sample on the feature map for imitation. For the regression subtask, in addition to selecting pixels of positive samples on the feature map, we also select pixels of negative samples for imitation. To prevent that a large number of negative samples dominate the loss in the classification subtask, we only collect the top-ranked samples in descending order of loss value, i.e. hard negative samples, to ensure the stability of the gradient during training.

In order to transfer the knowledge from the teacher network to the student network as much as possible, we use the probability distillation simultaneously with the above feature distillation method. When performing probability distillation, the easy anchor samples, which have output probability near to 1 or 0, are less informative. This is because they are similar to the ground truth probability. However, the detectors usually contain a large amount of easy anchor samples, they would offer too much useless information that restricts the performance of the distillation. Therefore, we only select anchors that contain more useful information, which have corresponding predicted probability close to 0.5. By this simple but effective strategy, the probability knowledge of the teacher network can be transferred to the student network. In the end, through joint feature imitation and probabilistic distillation training, the accuracy of the student network can be further improved.

Significant improvements have been achieved by using our methods in face detection without introducing any extra computation.

In short, our contributions are summarized as follows:

1. We propose a novel feature decoupling method that decouples coupled features into separated features for different subtasks.

2. We propose a more reasonable sampling strategy to select different samples for the different subtasks of the knowledge distillation process.

3. We propose an effective way to transfer the knowledge from the probability. And the accuracy of the student network can be further improved through joint feature distillation and probability distillation.

4. We apply the proposed distillation method on a lightweight face detection method and improves the accuracy by 5.1%, 5.1%, and 2.8% AP in Easy, Medium, Hard subsets respectively.

2 RELATED WORK

In this section, we briefly introduce the work related to our method, namely face detection and knowledge distillation.

Face Detection. With the rapid development of deep learning in recent years, CNNs have demonstrated a powerful ability to extract image features [7, 24, 25, 30]. At the same time, it also has an effect on the task of general object detection. Object detection is a basic research direction in the field of computer vision, providing basic information for other advanced computer vision analysis tasks. According to the pipeline of object detection, the current object detector can be divided into two categories: one-stage object detector and two-stage object detector. The one-stage detectors mainly include SSD [13], YOLO [1, 18–20], etc. And the two-stage detector mainly includes Fast-RCNN [5] and its variants [6, 11, 21]. Face detection is a subtask of general object detection. Since the one-stage detector is more efficient, face detectors are usually based on one-stage object detectors (SSD [13], etc.).

To improve the performance of the face detector, researchers have explored in various aspects. Some methods improve performance by designing specific structures in the detector. PyramidBox [26] utilizes the context information to improve the face detection results. FANet [33] designs a novel hierarchical feature pyramid to better merge the feature maps of different stages. SRN [3] is inspired by the RefineDet [36]. It appends a refinement branch to refine the classification results for small objects and regression results for large objects. Other methods improve the accuracy through multi-task joint training. DFS [27] propose a semantic segmentation branch to best utilize detection supervision information meanwhile applying attention mechanism in a self-supervised manner. RetinaFace [4] manually annotate five facial landmarks on the WIDER FACE [31] dataset and observe significant improvement in hard face detection with the assistance of this extra supervision signal. Beyond these, there is the strategy for anchor mining. HAMBox [14] proposes an online high-quality anchor mining strategy, which explicitly helps outer faces compensate with high-quality anchors. In pursuit of speed, lightweight face detectors have also been proposed. MTCNN [34] designs a cascade CNNs to filter background patches in a coarse to fine way. Faceboxes [35] is based on the SSD

[13] framework and propose several useful modules for building efficient networks. EagleEye [37] designs five strategies for building efficient face detectors which shows a good trade-off between high accuracy and fast speed on the popular embedded device with low computation power. In order to improve the accuracy as much as possible, some face detectors are often over-parameterized. Therefore, it is difficult to apply in practice because they are computationally expensive. On the contrary, there are also some face detectors that oversimplify parameters in pursuit of speed. It often causes a cliff-like drop in performance. Therefore, how to trade off the speed and performance of the face detector has become a question worth discussing.

Knowledge Distillation. Knowledge distillation is a common method of model compression. The main purpose of knowledge distillation is to transfer the knowledge of the teacher network to the student network. It allows the student network to have the same generalization performance as the teacher network while maintaining the original parameter amount. Since Hinton et al. proposed knowledge distillation for training the student through the output of the softmax layer of the teacher network [8], a vast number of methods for knowledge distillation have emerged.

According to the form of knowledge for distillation, it can be divided into two categories: knowledge from probability and knowledge from intermediate layers. For knowledge from probability, there are many recent works to study this. RKD [17] proposes distance-wise and angle-wise distillation losses that penalize structural differences in relations. Ensemble Distribution Distillation [15] proposes a way to both preserve the distributional information of an ensemble and improve the performance of a Prior Network. In addition to the above, distilling the knowledge from intermediate layers is also followed by many researchers. FitNets [22] trains the student with a deeper network but fewer channels. Attention transfer [32] trains the student by imitating the attention map of the teacher's. Besides, similarity-preserving KD [28] proposes a new form of knowledge distillation loss that is inspired by the observation that semantically similar inputs tend to elicit similar activation patterns in a trained network.

In addition to image classification, knowledge distillation is also surveyed by some prior works in object detection tasks. Since the detection task which consists of classification subtask and regression subtask is more complicated than the image classification. Wang et al. propose distilling object detectors by imitating the fine-grained feature [29]. Chen et al. propose a hierarchical distillation technique for pedestrian detection [2]. Zhu et al. propose the mask guided structure including not only the entire feature map but also region features covered by the object [38].

In the above method, they are similar in imitating the pixel on the feature map. Their difference lies in the selection of feature maps or pixels on the feature maps. Although their work has promoted the development of distillation in detection tasks, they still have some limitations. Since the feature map of the detection task mixes the features of classification and the features of regression, imitating pixels directly means taking the same sample for the regression subtask and the classification subtask. However, different subtasks have different requirements on samples. Specifically, training classifiers require both positive samples and negative samples, while training regressors only need positive samples. Taking the same sample for different subtasks often leads to suboptimal results. Our method uses decoupling to separate classification features and regression features, so as to select the samples needed for different subtasks. In addition to using the imitation feature to distill the detector, there are also some methods to extract knowledge in the probability maps. Jin et al. propose a novel loss function based on knowledge distillation to boost the performance of lightweight face detectors [10]. Inspired by this, in addition to imitating the feature map, we also combined the similarities between the probability to further improve the performance.

3 METHOD

Figure 1 demonstrates the pipeline of proposed task decoupled knowledge distillation method. It uses a heavyweight teacher face detector to improve the lightweight face detector by mimicking their feature maps. Based on the fact that detection task has two subtasks, i.e. the classification subtask and the regression subtask, we make two improvements in the knowledge distillation. Firstly, we propose the feature decoupling method to decouple the feature maps of both the teacher and student. Secondly, we propose the new sampling strategy to choose different samples in distilling the two subtasks. Besides feature imitation, we also perform the probability distillation to boost the accuracy of the student network.

3.1 Baseline face detector

To validate the proposed distillation method, we choose EagleEye [37] as our baseline. EagleEye [37] is a lightweight face detector based on the single-stage detection method SSD [13]. The backbone it takes is the $1/8$ MobileNet [9] which is the pruned MobileNet [9] with the channel numbers of each layer equals to the $1/8$ of the original MobileNet [9]. To further increase the speed of the network, it adopts successive downsampling convolutions at the beginning of the network. Meanwhile, it uses an efficient context module, information preserving activation function and focal loss to significantly improve the accuracy of the lightweight detector without adding too much computation costs. EagleEye [37] shows a good trade-off between high accuracy and fast speed on the device with low computation power.

Since the code of EagleEye has not been released, in this paper, we implement it by ourselves with one slight modification. Specifically, we design a new context module with the same structure of SSH [16] context module, but we use depthwise convolutions instead of vanilla convolutions.

3.2 Feature decoupling and sampling strategy

As discussed in the introduction, we decouple the task of distilling face detection into two distillation subtasks, which are the classification and the regression. In this subsection, we firstly introduce how we decouple the detector. After decoupling, we introduce the novel sampling strategy that guides us to select pixels on the feature map for distillation. Finally, we design the loss function based on sampling strategy.

Feature Decoupling. Since the single-shot face detectors usually use multiple branches to generate the detection results. Usually, the last feature maps of each branch are used for generating the last classification and regression results. In this paper, we call these

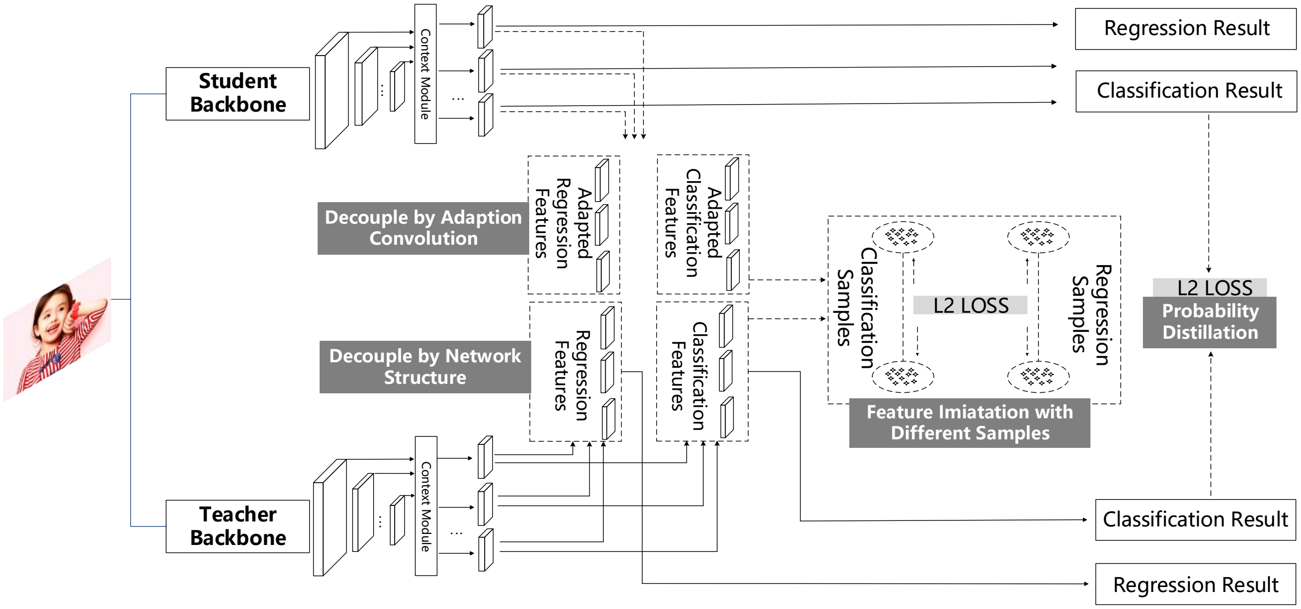


Figure 1: Overview of the task decoupled knowledge distillation method.

feature maps as *predicting feature maps*. The two subtasks share the same predicting feature map in almost all detectors, before using our sampling strategy, it is necessary to decouple the coupled predicting feature maps into separate features before distilling. We design a novel feature decoupling method, which introduces no extra calculations at the inference time of the student detector. As shown in Figure 1, we decouple the mixed features of teacher network by adding a convolution layer before the predicting convolution of each subtask respectively, while the prediction feature map of the student network remains coupled. When distilling, we convert coupled features from the student into separate features by adaption convolution. Note that the adaption convolution only exists during distillation, and is removed during inference. Therefore, the structure of the student is not affected by the feature decoupling. So the runtime efficiency of the lightweight student is kept.

Sampling Strategy. The purpose of our sampling strategy is to select different samples on different subtasks for imitation. When training the single-stage detector, the classification subtask is trained with both positive samples and negative samples, and the regression subtask is trained only with positive samples. To ensure consistency, we take the same sampling strategy when performing knowledge distillation. In the classification subtask, we select positive samples and negative samples on the feature map for distillation, while in the regression subtask, we only select positive samples. To prevent that a large number of negative samples might dominate the loss in the classification subtask, we collect the top-ranked samples in descending order of loss value, i.e. hard negative samples. Borrowed by the practice in OHEM [23], the number of the negative samples we collect is three times as many as positive samples.

Loss Function. The loss function of distillation is divided into two parts, one is the loss of the classification subtask, the other is the loss of the regression subtask. The loss function of the classification subtask for distillation is defined as following:

$$L_{D1} = \frac{1}{2N_1} \sum_{i=1}^{W} \sum_{j=1}^{H} \sum_{c=1}^{C} P_{ij} (f(s)_{ijc} - t_{ijc})^2 \tag{1}$$

$$\text{where} \qquad N_1 = \sum_{i=1}^{W} \sum_{j=1}^{H} P_{ij} \tag{2}$$

Here, H, W, C means the shape of the feature map and the subscript ijc is the coordinates to a pixel in the feature map. f means classification adaption convolution which transforms mixed features to classification features, and the parameter s represents mixed features of the student network. t represents classification features in the decoupled teacher network. P indicates the binary mask which consists of integers 0 and 1. The positions of all positive samples and difficult negative samples (top-ranked samples in descending order of loss) are 1, the rest are 0. N_1 means the number of positive samples in the classification subtask.

Similarly, the loss function of the regression subtask is defined as following:

$$L_{D2} = \frac{1}{2N_2} \sum_{i=1}^{W} \sum_{j=1}^{H} \sum_{c=1}^{C} Q_{ij} (g(s)_{ijc} - v_{ijc})^2 \tag{3}$$

$$\text{where} \qquad N_2 = \sum_{i=1}^{W}\sum_{j=1}^{H} Q_{ij} \tag{4}$$

Different from the symbol in the classification subtask, g means regression adaption convolution while v represents regression features in the decoupled teacher network. Q indicates the binary mask. The position of all positive samples is 1, and the rest is 0. N_2 means the number of positive samples in the regression subtask.

Finally the loss of feature imitation is defined as the sum of L_{D1}, L_{D2}:

$$L_{imitation} = L_{D1} + L_{D2} \tag{5}$$

3.3 Probability distillation

Similar to the probability distillation of image classification, our method uses the probability of the teacher's output as a soft label to supervise the probability of the student's. Since the one-stage detector will tile dense anchors on the image, the probability of its output represents the probability of the corresponding category for each anchor. At the same time, only a small part of the anchors are assigned to ground truth faces objects, most of the anchors belong to the background. When we select the probability of all anchors output by the detector for distillation, since most anchors only carry meaningless background information, they will occupy a dominant position in the loss, thus overwhelming meaningful samples.

Therefore, we only select samples with a large amount of information for distillation. Therefore, we have designed a mask to filter out samples with a small amount of information. As shown in Eqn.6, symbol X represents the mask, the subscript i represents the i-th probability, p means the probability of the teacher's output, q means probability of the student's, M means the number of all samples, N means the number of selected samples.

$$L_{prob} = \frac{1}{N}\sum_{i=1}^{M} X_i(p_i - q_i)^2 \tag{6}$$

$$\text{where} \qquad N = \sum_{i=1}^{M} X_i \tag{7}$$

In order to choose the probability with effective information, we think that the uncertain samples predicted by the teacher network are regarded as effective probability. That is, the output probability of teacher network is close to 0.5. The greater the uncertainty of the teacher network, the more difficult it is to distinguish the foreground and background. At the same time, it can also reflect the similarity of its foreground and background to a certain extent. The specific definition of X_i is as follows:

$$X_i = \begin{cases} 1 & T < p_i < 1 - T \\ 0 & otherwise \end{cases} \tag{8}$$

$$\text{where} \qquad 0 \leq T < 0.5 \tag{9}$$

The symbol p represents the probability of teacher output. And the subscript i represents the i-th probability. T is the hyperparameter which ranges from 0 to 0.5. It represents the threshold of teacher's prediction uncertainty. When the threshold T is closer to 0.5, it represents an increase in the uncertainty of these samples. The average amount of information per sample increases, but the total amount of samples decreases. Conversely, when the threshold T is closer to 0, it represents a decrease in the uncertainty of these samples. The average amount of information per sample decreases, but the total amount of samples increases.

The loss function of the original detection task, denoted as L_{det}, consists of regression loss and classification loss. $L_{imitation}$ means the loss of feature imitation, L_{prob} means the loss of probability distillation. The final overall loss is as follows:

$$L_{training} = L_{imitation} + L_{prob} + L_{det} \tag{10}$$

3.4 Implement details

Before sampling, we need to define positive samples and negative samples. Similar to [29], we calculate the intersection over union between the ground truth and all anchors. And we find the maximum one, denoted as MaxIoU. Then we get a threshold value θ which is calculated as half of the MaxIoU. Next, the anchors whose IoU with the ground truth is larger than θ are defined as positive anchors. The pixels on the feature map corresponding to positive anchors are defined as positive samples. Meanwhile, the remaining pixels are defined as negative samples.

For training details, we train the detector on 2 NVIDIA Titan X GPUs, with a batch size of 32 (=16 × 2) for 80 epochs with the learning rate starting from 0.02 multiplying 0.1 at the 55th and the 68th epoch, respectively. And for distilling details, we employ the EagleEye [37] 6× as the teacher network when distilling EagleEye. That is, we increase the number of channels in the backbone of EagleEye [37] by 6 times while other parameter settings remain unchanged. Due to the large graphics card memory occupied by RetinaFace [4] and FaceBoxes [35], limited by hardware, we use 3 times corresponding to their own size as the teacher network. Before distilling, we use a regular training method (just use ground truth to supervise the network) to get the student network and teacher network. Then we use the model trained in an ordinary way as our pre-trained model, and use the trained teacher network for distillation. Finally, we combine feature imitation and probability distillation to distill at the same time.

4 EXPERIMENTS

4.1 Dataset

We train our face detector on WIDER FACE [31] training set and report face detection performance on the validation set. Wider Face dataset [31] is a large face detection dataset. It has 32,203 images with 393, 703 annotated faces, varying largely in scales, poses, occlusions, and illuminations. The images are divided into three splits, including 40% for training, 10% for validation, and 50% for testing. The faces are classified into three subsets according to their levels of detection difficult: Easy, Medium, and Hard. Generally, the Hard subset contains a great number of tiny faces. The official evaluation metric is the Average Precision (AP) for each subset. If not specified, we use single-scale testing at the resolution of the original image by default. For the evaluation metric of running speed, we take

Table 1: Compare the impact of different thresholds T on performance.

T	Easy	Medium	Hard
0	88.1	83.2	48.6
0.1	**89.2**	**84.2**	**49.0**
0.2	88.8	83.9	49.0
0.3	88.6	84.0	49.0
0.4	88.3	83.4	48.9

Table 2: Compare with other distillation methods. TDKD: Task decoupled knowledge distillation method.

Method	Easy	Medium	Hard
Teacher	93.3	90.7	58.8
Student	86.3	81.0	46.2
Vanilla Distillation	86.8	80.7	45.6
Fine-grained Feature Imitation [29]	86.8	81.1	47.0
Novel Loss KD [10]	87.3	82.3	46.8
TDKD (ours)	**89.2**	**84.2**	**49.0**

Table 3: Compare the effect of TDKD on other lightweight face detectors. TDKD: Our proposed task decoupled knowledge distillation method.

Method	Easy	Medium	Hard
FaceBoxes-teacher	87.7	81.9	44.5
FaceBoxes	82.6	75.5	38.8
FaceBoxes+TDKD	84.6	77.8	40.3
RetinaFace-teacher	90.7	89.3	84.3
RetinaFace	87.6	85.8	79.6
RetinaFace+TDKD	89.6	87.9	81.4

advantage of NCNN [1] to accelerate the model inference and timing is performed on the Raspberry Pi 3b+ which uses a quad-core ARM Cortex-A53 processor. In this section, we report the FPS (Frames Per Second) of the models at 640 × 480 resolution inputs.

4.2 Compare the impact of different thresholds T on performance

As shown in Table 1, we explored the effect of different thresholds T on performance. We found an interesting phenomenon: when T is equal to 0, the accuracy of the network is the worst. When T increased from 0 to 0.1, the accuracy suddenly increased and reached the highest at the same time. Continuing to increase T, the result tends to be stable. According to Eqn.8, it can be seen that when T is equal to 0, the distribution of the selected probability interval is from 0 to 1.0. That is, we select all the probability as samples for distillation. In order to ensure the accuracy of the detection, the one-stage detector will tile dense anchors on the image. And they are mainly dominated by easy positive and negative samples, as shown in Figure 2. Therefore, when we remove these samples and let the network focus on those samples that are difficult to distinguish in the teacher network, it helps students learn the similarities between the classes, thereby improving the accuracy of the student network. As shown in the first two rows of the Table 1, when the value of T is set to 0.1 to filter easy samples, the accuracy is significantly improved compared to 0, which further proves our conjecture. As the value of T increases, the difficulty of the samples increases. However, the number of samples decreases. Therefore, when the value of T increases from 0.2 to 0.4, the accuracy change is less obvious in the bottom three columns of the table. Finally, we set the threshold T to 0.1 as our final result.

4.3 Effectiveness of each strategy

In Table 4, we demonstrate the effectiveness of each strategy. We apply the feature decoupling imitation and probability distillation to the lightweight face detector, EagleEye. As shown in the fifth and sixth columns, the accuracy is improved slightly, i.e. 0.1% and 0.4% AP improvements in the Easy and Medium subsets. Note that our feature decoupling method does not change student network structure as well as introduce any extra calculations. We think that it is easier to transfer knowledge through separate features than mixed features. Therefore, a slight improvement has been achieved while the student network keeps unchanged. As shown in the sixth columns and the seventh columns of Table 4, along with the sampling strategy, the accuracy has been further improved by 0.9%, 1.3%, 1.8% AP in the Easy, Medium, and Hard subsets. It proves the effectiveness of our sampling strategy. The samples required for different subtasks are indeed different. Besides, selecting hard negative samples to the classification subtask can improve the discriminating ability of classifier. After that, we combine the probability distillation with feature decoupling imitation. As shown in the last two columns of Table 4, the performance is further improved by 1.4%, 1.4%, 0.2% AP in the Easy, Medium and Hard subsets respectively. Comparing vanilla distillation methods, our method surpasses it by 2.4%, 3.5%, 3.4% AP in the Easy, Medium, and Hard subsets. Decoupling the network and selecting different samples for different subtasks can further improve the effect of feature distillation in the detector. In the end, compared to the network without distillation, our method surpass the baseline by 2.9%, 3.2%, 2.8% AP in the Easy, Medium, and Hard subsets. It can be seen AP in the Hard subset gets the most benefits. Limited by the amount of parameters, lightweight face detectors tend to perform poorly on small faces. By knowledge distillation, the knowledge learned by the teacher network can be transferred to the student network, especially the knowledge about small faces, which brings a large increase in AP on Hard subset.

[1] https://github.com/Tencent/ncnn

Table 4: Effectiveness of each strategy.

Strategy	Teacher	Our Impl.					
Vanilla Distillation			✓				
Fine-grained Feature Imitation [29]				✓	✓	✓	✓
Feature Decoupling					✓	✓	✓
Sampling Strategy						✓	✓
Probability Distillation							✓
Easy	93.3	86.3	86.8	86.8	86.9	87.8	89.2
Medium	90.7	81.0	80.7	81.1	81.5	82.8	84.2
Hard	58.8	46.2	45.6	47.0	47.0	48.8	49.0

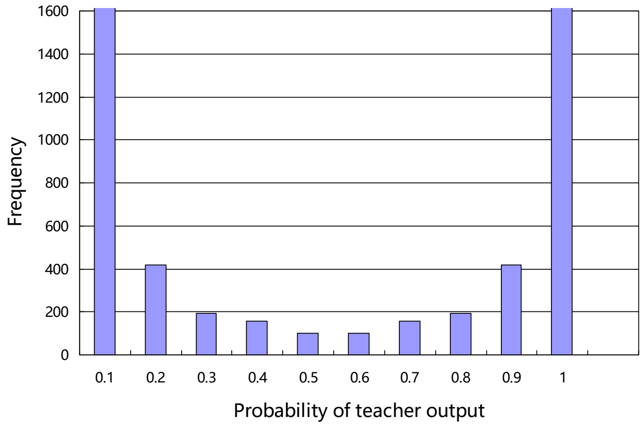


Figure 2: Probability distribution histogram of teacher output

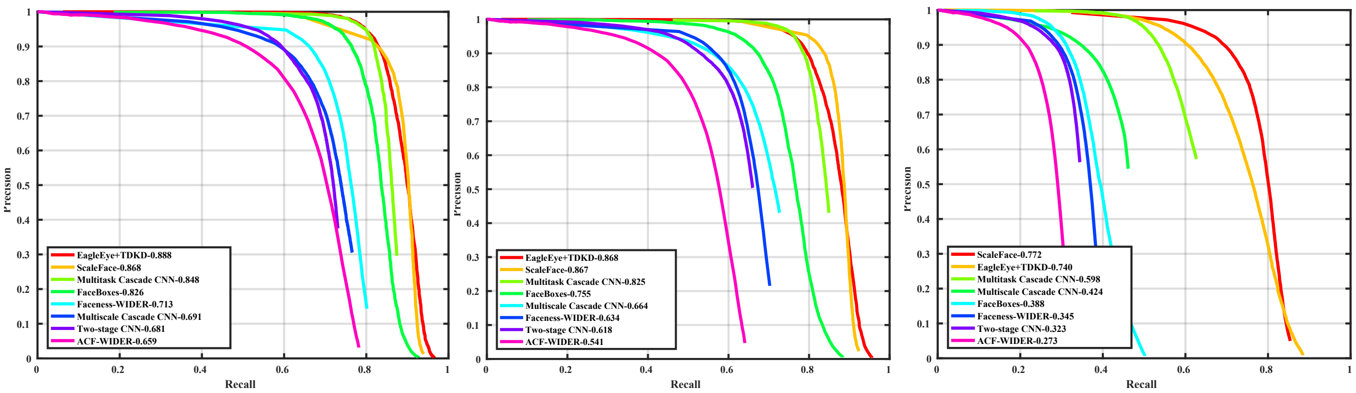
4.4 Distillation on different face detectors

In Table 3, we apply our distillation method to other lightweight face detectors, i.e RetinaFace [4] and FaceBoxes [35]. We use mobilenet-0.25 RetinaFace [4] as the student network. The corresponding teacher network is three times the size of the student network, that is, we multiply the number of channels in each layer of the student network by 3 to get the teacher network. Since the RetinaFace [4] only reports the AP in the Hard by testing with the short side of the image scaling to 1600 in the paper, we also report the results under the same condition for fairness. We use PyTorch to implement RetinaFace [4]. Under the same testing condition, our AP in the Hard reached 79.6%, which is 1.4% higher than the 78.2% reported in the paper. After distillation, our method has further improved 2.0%, 2.1%, 1.8% AP in the Easy, Medium, and Hard subsets. It is worth noting that our results are 3.2% AP higher than the results reported by the paper in the Hard subset. As for FaceBoxes [35], after equipped with TDKD, AP increased by 2.0%, 2.3%, and 1.5% in the Easy, Medium, and Hard subsets, respectively. We apply the parameters set on the EagleEye directly to these methods without further adjustment, and the accuracy has steadily improved without introducing any extra calculation. It can be seen that our method is generally applicable to lightweight face detectors.

4.5 Compare with other distillation methods

In Table 2, we used EagleEye as the baseline and compared other distillation methods. For the vanilla distillation method, we imitate all the pixels on the feature map. Compared with the baseline, although it has improved by 0.5% AP in the Easy subset, it has dropped by 0.3% and 0.6% AP in the Medium, and Hard subsets, respectively. We think it is because the ratio of the positive sample to the negative sample is larger if the scale of anchors become larger. In this case, it will not cause the imbalance of the positive and negative samples. As the anchor scale becomes smaller, the number of negative samples increases sharply, resulting in an imbalance between positive and negative samples. For the Easy indicator, the accuracy is slightly improved due to the large proportion of big faces. For the Medium and Hard indicators, because they have a large proportion of small faces, the accuracy decreases. Besides, there are more small faces in the Hard subset, compared with the Medium subset, it declines more severely. This phenomenon further confirms our conjecture.

And for fine-grained feature imitation, its main idea is to select some pixels on the feature map which are close to ground truth boxes. It avoids the problem of sample imbalance caused by small anchors. Compared with the vanilla distillation, it mainly improves the AP in the Medium and Hard subsets, which are improved by 0.4 % and 1.4 %, respectively. But compared to the baseline, it is increased by 0.5 %, 0.1 %, 0.8 % AP in the Easy, Medium, Hard subsets respectively. Different from the feature map imitation, the novel loss KD directly imitates the probability map instead. Compared with baseline, it is improved by 1 %, 1.3 %, 0.6 % AP in the Easy, Medium and Hard subsets respectively. It mainly improved AP in the Easy and Medium subsets, but the AP improved in the Hard subset is relatively weak. Compared with the above methods, the method we proposed has been significantly improved AP in the Easy, Medium, and Hard subsets. We lead by a large margin with vanilla distillation, i.e. 2.4 %, 3.5 %, 3.4 % AP in the Easy, Medium, and Hard subsets. At the same time, the method of comparing fine-grained feature imitation and novel loss KD has also improved significantly, with 2.4 %, 3.1%, 2.0% and 1.9%, 1.9%, 2.2% AP improved in the Easy, Medium and Hard subsets compared with fine-grained feature imitation and novel loss KD. In the end, our method is 2.9%, 3.2%, 2.8% AP higher than the baseline without introducing any extra calculations in the Easy, Medium, and Hard subsets respectively.



(a) Easy

(b) Medium

(c) Hard

Figure 3: Precision-recall curves on WIDER FACE validation sets.

Table 5: Comparison with state-of-the-art lightweight face detectors. EagleEye* is the self implemented EagleEye. TDKD: task decoupled knowledge distillation. Symbol M represent Multi-scale testing.

Approach	Easy	Medium	Hard	FPS
1/8 MobileNet-SSD	74.7	65.5	34.7	10
Faceboxes [35]	82.6	75.5	38.8	3.4
EagleEye [37]	84.1	79.1	46.2	20
EagleEye*	86.3	81.0	46.2	19.8
EagleEye*+TDKD	**89.2**	**84.2**	**49.0**	**19.8**
MTCNN(M) [34]	85.1	82.0	60.7	5.4
EagleEye*(M)+TDKD	**88.8**	**86.8**	**74.0**	-

4.6 Comparison with state-of-the-art lightweight face detectors

In Table 5, we further compare our method with other state-of-the-art lightweight face detectors. Since MTCNN [34] uses the image pyramid as input for better dealing with multi-scale faces, we also report the results equipped with image pyramid for a fair comparison. It is marked as symbol M after the corresponding method in the bottom half of the table. In single-scale test results (i.e. in the top half of the table), our method surpasses all the methods in accuracy while maintaining the fastest speed. Specially, compared with EagleEye, the accuracy of our method is 5.1%, 5.1%, and 2.8% AP higher in the Easy, Medium, and Hard subsets, respectively. In multi-scale results, our method still surpasses MTCNN [34] by a large margin in accuracy. It shows that significant improvements have been achieved by our methods in face detection without introducing any extra computation. As for speed, our method is only 0.2 FPS slower than EagleEye. Since we use a stronger baseline, EagleEye*. With adding a few calculations, we get 2.2% and 1.9% AP improvements in Easy and Medium subsets, respectively. Still, our method is faster than any other method. In particular, it's nearly twice as fast as 1/8 MobileNet-SSD.

5 CONCLUSION

In this paper, we propose a task decoupled knowledge distillation method for lightweight face detectors. Through the feature decoupling and sampling strategy we proposed, we perform a more accurate sampling of two tasks in the detection task, thereby reducing the sample noise in feature distillation. Moreover, we propose a simple but effective method to conduct probability distillation for the detectors. The feature ablation experiments show that our strategies are effective. Since knowledge distillation is not related to the network itself, we can integrate it with any lightweight face detector without introducing any extra calculations. In the end, by simply applying our method, we successfully build a new lightweight face detector. It achieves the new-state-of-the art on both the runtime efficiency and accuracy.

6 ACKNOWLEDGMENTS

This work was supported by the Research and Development Projects in the Key Areas of Guangdong Province (No.2019B010153001), and National Natural Science Foundation of China under Grants 61772527, 61806200, 61976210, 61876086, and 61702510.

REFERENCES

[1] Alexey Bochkovskiy, Chien-Yao Wang, and Hong-Yuan Mark Liao. 2020. YOLOv4: Optimal Speed and Accuracy of Object Detection. *CoRR* abs/2004.10934 (2020). arXiv:2004.10934 https://arxiv.org/abs/2004.10934

[2] Rui Chen, Haizhou Ai, Chong Shang, Long Chen, and Zijie Zhuang. 2019. Learning Lightweight Pedestrian Detector with Hierarchical Knowledge Distillation. In *2019 IEEE International Conference on Image Processing, ICIP 2019, Taipei, Taiwan, September 22-25, 2019.*

[3] Cheng Chi, Shifeng Zhang, Junliang Xing, Zhen Lei, Stan Z. Li, and Xudong Zou. 2019. Selective Refinement Network for High Performance Face Detection. *international joint conference on artificial intelligence* (2019).

[4] Jiankang Deng, Jia Guo, Yuxiang Zhou, Jinke Yu, Irene Kotsia, and Stefanos Zafeiriou. 2019. RetinaFace: Single-stage Dense Face Localisation in the Wild. *CoRR* abs/1905.00641 (2019). arXiv:1905.00641 http://arxiv.org/abs/1905.00641

[5] Ross B. Girshick. 2015. Fast R-CNN. *CoRR* abs/1504.08083 (2015). arXiv:1504.08083 http://arxiv.org/abs/1504.08083

[6] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross B. Girshick. 2020. Mask R-CNN. *IEEE Trans. Pattern Anal. Mach. Intell.* 42, 2 (2020), 386–397. https://doi.org/10.1109/TPAMI.2018.2844175

[7] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep Residual Learning for Image Recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016.* IEEE Computer Society, 770–778. https://doi.org/10.1109/CVPR.2016.90

[8] Geoffrey E. Hinton, Oriol Vinyals, and Jeffrey Dean. 2015. Distilling the Knowledge in a Neural Network. *CoRR* abs/1503.02531 (2015).

[9] Andrew G. Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. 2017. MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications. *CoRR* (2017).

[10] Haibo Jin, Shifeng Zhang, Xiangyu Zhu, Yinhang Tang, Zhen Lei, and Stan Z. Li. 2019. Learning Lightweight Face Detector with Knowledge Distillation. In *2019 International Conference on Biometrics, ICB 2019, Crete, Greece, June 4-7, 2019.* IEEE, 1–7. https://doi.org/10.1109/ICB45273.2019.8987309

[11] Tsung-Yi Lin, Piotr Dollár, Ross B. Girshick, Kaiming He, Bharath Hariharan, and Serge J. Belongie. 2017. Feature Pyramid Networks for Object Detection. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017.* IEEE Computer Society, 936–944. https://doi.org/10.1109/CVPR.2017.106

[12] Tsung-Yi Lin, Priya Goyal, Ross B. Girshick, Kaiming He, and Piotr Dollár. 2020. Focal Loss for Dense Object Detection. *IEEE Trans. Pattern Anal. Mach. Intell.* 42, 2 (2020), 318–327. https://doi.org/10.1109/TPAMI.2018.2858826

[13] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott E. Reed, Cheng-Yang Fu, and Alexander C. Berg. 2016. SSD: Single Shot MultiBox Detector. In *Computer Vision - ECCV 2016 - 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part I.*

[14] Yang Liu, Xu Tang, Xiang Wu, Junyu Han, Jingtuo Liu, and Errui Ding. 2019. HAMBox: Delving into Online High-quality Anchors Mining for Detecting Outer Faces. *CoRR* abs/1912.09231 (2019). arXiv:1912.09231 http://arxiv.org/abs/1912.09231

[15] Andrey Malinin, Bruno Mlodozeniec, and Mark J. F. Gales. 2020. Ensemble Distribution Distillation. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020.* OpenReview.net. https://openreview.net/forum?id=BygSP6Vtvr

[16] Mahyar Najibi, Pouya Samangouei, Rama Chellappa, and Larry S. Davis. 2017. SSH: Single Stage Headless Face Detector. In *IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, October 22-29, 2017.*

[17] Wonpyo Park, Dongju Kim, Yan Lu, and Minsu Cho. 2019. Relational Knowledge Distillation. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019.* Computer Vision Foundation / IEEE, 3967–3976. https://doi.org/10.1109/CVPR.2019.00409

[18] Joseph Redmon, Santosh Kumar Divvala, Ross B. Girshick, and Ali Farhadi. 2016. You Only Look Once: Unified, Real-Time Object Detection. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016.* IEEE Computer Society, 779–788. https://doi.org/10.1109/CVPR.2016.91

[19] Joseph Redmon and Ali Farhadi. 2017. YOLO9000: Better, Faster, Stronger. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017.* IEEE Computer Society, 6517–6525. https://doi.org/10.1109/CVPR.2017.690

[20] Joseph Redmon and Ali Farhadi. 2018. YOLOv3: An Incremental Improvement. *CoRR* abs/1804.02767 (2018). arXiv:1804.02767 http://arxiv.org/abs/1804.02767

[21] Shaoqing Ren, Kaiming He, Ross B. Girshick, and Jian Sun. 2017. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *IEEE Trans. Pattern Anal. Mach. Intell.* 39, 6 (2017), 1137–1149. https://doi.org/10.1109/TPAMI.2016.2577031

[22] Adriana Romero, Nicolas Ballas, Samira Ebrahimi Kahou, Antoine Chassang, Carlo Gatta, and Yoshua Bengio. 2015. FitNets: Hints for Thin Deep Nets. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings.*

[23] Abhinav Shrivastava, Abhinav Gupta, and Ross B. Girshick. 2016. Training Region-Based Object Detectors with Online Hard Example Mining. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016.*

[24] Karen Simonyan and Andrew Zisserman. 2015. Very Deep Convolutional Networks for Large-Scale Image Recognition. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, Yoshua Bengio and Yann LeCun (Eds.). http://arxiv.org/abs/1409.1556

[25] Christian Szegedy, Sergey Ioffe, Vincent Vanhoucke, and Alexander A. Alemi. 2017. Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, February 4-9, 2017, San Francisco, California, USA*, Satinder P. Singh and Shaul Markovitch (Eds.). AAAI Press, 4278–4284. http://aaai.org/ocs/index.php/AAAI/AAAI17/paper/view/14806

[26] Xu Tang, Daniel K. Du, Zeqiang He, and Jingtuo Liu. 2018. PyramidBox: A Context-Assisted Single Shot Face Detector. *european conference on computer vision* (2018), 812–828.

[27] Wanxin Tian, Zixuan Wang, Haifeng Shen, Weihong Deng, Binghui Chen, and Xiubao Zhang. 2018. Learning Better Features for Face Detection with Feature Fusion and Segmentation Supervision. *CoRR* abs/1811.08557 (2018). arXiv:1811.08557 http://arxiv.org/abs/1811.08557

[28] Frederick Tung and Greg Mori. 2019. Similarity-Preserving Knowledge Distillation. In *2019 IEEE/CVF International Conference on Computer Vision, ICCV 2019, Seoul, Korea (South), October 27 - November 2, 2019.* IEEE, 1365–1374. https://doi.org/10.1109/ICCV.2019.00145

[29] Tao Wang, Li Yuan, Xiaopeng Zhang, and Jiashi Feng. 2019. Distilling Object Detectors With Fine-Grained Feature Imitation. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019.*

[30] Saining Xie, Ross B. Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. 2017. Aggregated Residual Transformations for Deep Neural Networks. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017.* IEEE Computer Society, 5987–5995. https://doi.org/10.1109/CVPR.2017.634

[31] Shuo Yang, Ping Luo, Chen Change Loy, and Xiaoou Tang. 2016. WIDER FACE: A Face Detection Benchmark. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016.* IEEE Computer Society, 5525–5533. https://doi.org/10.1109/CVPR.2016.596

[32] Sergey Zagoruyko and Nikos Komodakis. 2017. Paying More Attention to Attention: Improving the Performance of Convolutional Neural Networks via Attention Transfer. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings.*

[33] Jialiang Zhang, Xiongwei Wu, Jianke Zhu, and Steven C. H. Hoi. 2017. Feature Agglomeration Networks for Single Stage Face Detection. *arXiv preprint arXiv:1712.00721* (2017).

[34] Kaipeng Zhang, Zhanpeng Zhang, Zhifeng Li, and Yu Qiao. 2016. Joint Face Detection and Alignment using Multi-task Cascaded Convolutional Networks. *CoRR* (2016).

[35] Shifeng Zhang, Xiaobo Wang, Zhen Lei, and Stan Z. Li. 2019. Faceboxes: A CPU real-time and accurate unconstrained face detector. *Neurocomputing* (2019).

[36] Shifeng Zhang, Longyin Wen, Xiao Bian, Zhen Lei, and Stan Z. Li. 2018. Single-Shot Refinement Neural Network for Object Detection. In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018.* 4203–4212.

[37] Xu Zhao, Xiaoqing Liang, Chaoyang Zhao, Ming Tang, and Jinqiao Wang. 2019. Real-Time Multi-Scale Face Detector on Embedded Devices. *Sensors* (2019).

[38] Yousong Zhu, Chaoyang Zhao, Chenxia Han, Jinqiao Wang, and Hanqing Lu. 2019. Mask Guided Knowledge Distillation for Single Shot Detector. In *IEEE International Conference on Multimedia and Expo, ICME 2019, Shanghai, China, July 8-12, 2019.*