Feature map masking based single-stage face detection

Xi Zhang [1,2,3], Junliang Chen [1,2,3], Weicheng Xie [1,2,3] and Linlin Shen [1,2,3] *
[1] Computer Vision Institute, College of Computer Science and Software Engineering
[2] Shenzhen Institute of Artificial Intelligence and Robotics for Society
[3] Guangdong Key Laboratory of Intelligent Information Processing
Shenzhen University, PR China
zhangxi1987see@gmail.com, chenjunliang2016@email.szu.edu.cn, {wcxie, llshen}@szu.edu.cn

Abstract

Although great progress has been made in face detection, a trade-off between speed and accuracy is still a great challenge. We propose in this paper a feature map masking based approach for single-stage face detection. As feature maps extracted from feature pyramid network might contain face unrelated features, we propose a mask generation branch to predict those significant units for face detection. The masked feature maps, where only important features are left, are then passed through the following detection process. Ground truth masks, directly generated from the training images, based on the face bounding boxes, are used to train the feature mask generation module. A mask constrained dropout module has also been proposed to drop out significant units of the shared feature maps, such that the detection performance can be further improved. The proposed approach is extensively tested using the WIDER FACE dataset. The results suggest that our detector with ResNet-152 backbone, achieves the best precision-recall performance among competing methods. As high as 95.4%, 94.0% and 86.9% accuracies have been achieved on the easy, medium and hard subsets, respectively.

1. Introduction

Face detection is a fundamental and essential task in various face applications. The pioneering work by Viola-Jones [16] applies AdaBoost algorithm with Haar-Like features to train cascaded classifiers. Most of the subsequent works rely on hand-crafted features and carefully designed classifiers [24, 10]. In recent years, convolutional neural network (CNN) [13, 4] achieves great progress. The deeply learned features gradually replace these hand-crafted features. These universal object detectors, such as R-CNN [12], SSD [8] and YOLO [11], apply CNN as the feature extractor and achieve much better performance. They provide the new baselines for face detection. Although there are a large number of studies based on CNN in face detection, detecting tiny faces remains a great challenge.

New anchors and networks [8, 21, 22] and merging contextual features [23, 17, 15, 6, 7] are common approaches to detect faces with different scales. However, little attention has been paid to significant units of shared feature map, which play a significant role in face detection. In the heatmap, the highlighted units of shared feature maps correspond to extracted face features of original image. Here, the highlighted units are defined as the significant units. Actually, as shown in Fig. 1, if the occupied area of face ground truth (GT) boxes is small in the whole image, the area of the significant units of shared feature maps are also small in the corresponding detection head. These significant units can be used to construct sparse shared feature maps, i.e. only these significant units of shared feature maps are used for classification and localization tasks. There are two issues related to the significant units. The first one is how to label these significant units. The second one is how to employ significant units in face detectors. We address these issues by introducing a facial feature masking module (FFMM) and a mask constrained dropout module (MCDM) to make full use of these significant units in the following steps.

Firstly, in our network, not only face class labels and ground truth boxes in original images, but also the class labels of significant units in shared feature maps are necessary. To this end, we use face ground truth boxes to indicate significant units. In this work, we use a weakly-supervised solution to label significant units for generating the ground truth mask.

Secondly, the significant units should be online labeled in inference and applied to classification and localization tasks. In this work, we propose a facial feature masking module to retain the significant units of shared feature maps.

Thirdly, the face detector should make full use of the sig-

*Corresponding author
978-1-7281-9186-7/20/$31.00 ©2020 IEEE

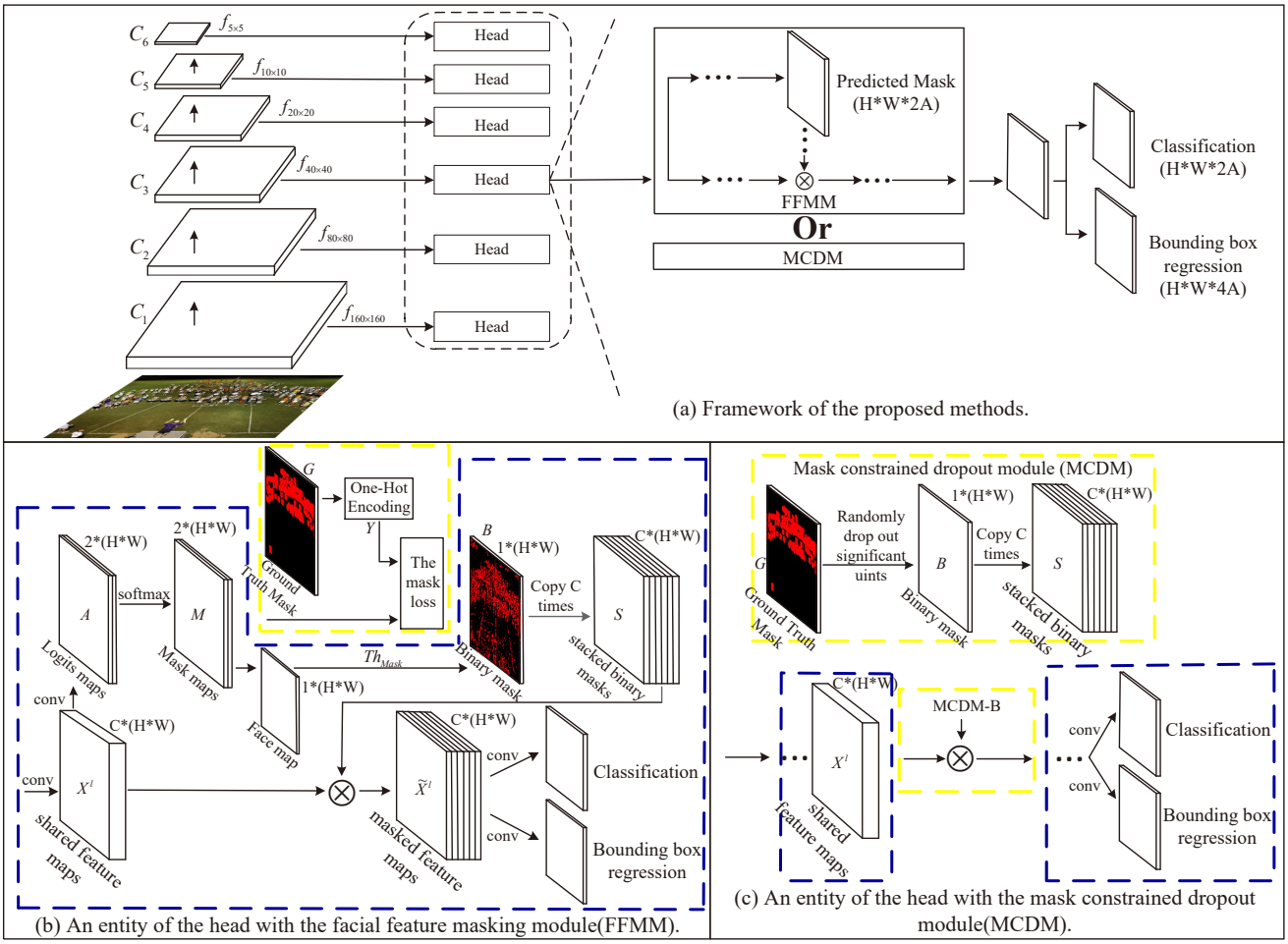

Figure 1. Overall framework (a) and an entity with FFMM (b) and MCDM (c). Here, W is width, H is height and C is the channel number of feature maps. $\otimes$ represents the element-wise product of two matrixes. C_n is multi-scale feature maps. MCDM-B represents that the MCDM is located behind the shared feature maps. In (b) and (c), the content within yellow dotted box only exists in training and the content of mazarine dotted box is shared both in training and testing.

nificant units. Inspired by Dropout [14], we introduce the mask constrained dropout module (MCDM) to make these units more robust and create useful features in shared feature maps.

In addition, we use a simple feature enhancement module (FEM) to demonstrate that MCDM is effective for promoting accuracy.

For clarity, there are three main contributions of our work.

1. We use a weakly-supervised solution to generate ground truth masks using face ground truth boxes. 2. To utilize the significant units of feature maps, we propose FFMM for generating the masks. Meanwhile, the proposed module can also make a screening in shared feature maps and ignore most background features to improve face detection accuracy. As FFMM can highlight the area of feature maps where faces exist, the masked feature maps are sparse. Structurally, if the FFMM is robust, it only pays attention to prediction head where faces exist. 3. We propose MCDM to randomly drop significant units of shared feature maps in training to make these units more robust and create useful features.

The rest of the paper is organized as follows. Section 2 provides an overview of the related works. Section 3 introduces our method. Section 4 presents the experiments and Section 5 concludes the paper.

2. Related Works

Multi-stage v.s. Single-stage. Both multi-stage and single-stage object detection methods have been used in face detection. Generally speaking, the accuracy of the multi-stage object detector is higher while its speed is slower. The more stages it has, the higher computation cost it requires. Faster R-CNN [12] firstly applies the Region Proposal Network (RPN) for generating region proposals.

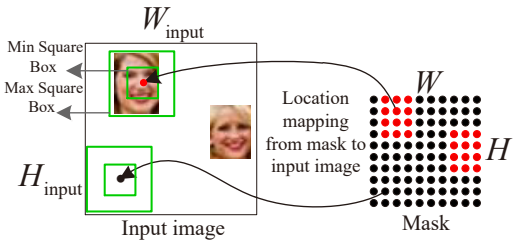

Figure 2. The red and black points are 1 and 0 in mask, respectively.

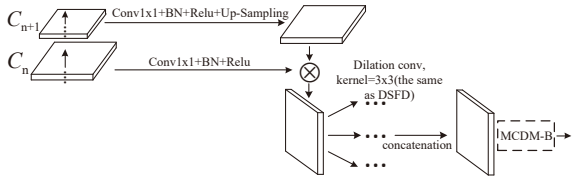

Figure 3. A feature enhanced MCDM.

Then it refines the region proposals by Fast R-CNN detector [3]. In the cascaded methods, the bounding box proposals and the subsequent pixel or feature resampling stage are computationally intensive. LDCF [9] applied CNN as the feature extractor in the traditional face detection framework. It is faster but does not perform well for tiny face detection. Faceness [18] trains a series of CNNs for facial attribute recognition to detect partially occluded faces. In the first stage, the generated response maps of different facial parts are employed to produce face proposals, which also do not work well for tiny face. Multiscale Cascade CNN [19] is a multi-scale two-stage cascade framework and employs a divide and conquer strategy to address the high variability of scales. Its multiscale detection networks are time-consuming. Two-stage CNN [19] is similar to Multi-scale Cascade CNN [19]. Though a single network, instead of multiple networks, is used for different scales, it is still time-consuming. MTCNN [20] proposes a cascaded structure with three stages of carefully designed deep convolutional networks to detect face in a coarse-to-fine manner. While the computational costs of multi-stage detectors increase with the number of faces in an image, the speed of single-stage face detectors is constant. SSD [8] removes proposal generation and subsequent pixel or feature resampling stages to improve detection speed. YOLO [11] computes a global feature map and uses a fully-connected layer to predict detections in a fixed set of regions. Both SSD and YOLO do not work well for small objects detection. S^3FD [22] proposes a max-out background label for the detection layer at the lowest level to reduce the false positives of small faces and a scale compensation anchor matching strategy with two stages to improve the recall. The latest single-stage detector is as accurate as the multi-stage approaches and its speed is fast, so we use it for face detection in this work.

Context-associated Detectors. Recently, some works show the importance of contextual information for tiny face detection. DSSD [17] augments SSD with deconvolution layers to introduce additional large-scale context in object detection. The deeper the backbone, the more inference time it costs. Pyramidbox [15] improves the utilization of contextual information to provide extra supervision for small faces. Its Low-level Feature Pyramid Network is relatively bloated and can be optimized. Pyramidbox++ [7] proposes the Dense Context Module with dense connection to enlarge receptive field and pass information more efficiently. Its dense context module is time-consuming. RetinaFace [2] uses independent context modules to increase the receptive field. But it adds extra annotations of five facial landmarks to improve performance. To use the context relationship between anchors, DSFD [6] proposes a feature enhancement module that incorporates multi-level dilated convolutional layers to enhance the semantic of the features. Inspired by the above methods, we propose the simple feature enhancement module.

3. Our Method

We firstly introduce the overall face detection framework, which consists of a backbone pretrained on ImageNet [1] for multi-level feature extraction, and the corresponding heads to process the extracted feature maps. Then, we introduce FFMM and MCDM. Finally, we give the details of the overall loss function.

3.1. Overall framework

Fig. 1 (a) shows the framework of our face detector. We take the six layers $f_{160\times160}$, $f_{80\times80}$, $f_{40\times40}$, $f_{20\times20}$, $f_{10\times10}$ and $f_{5\times5}$ with different sizes from the backbone as inputs. As the feature maps may contain both useful facial features and irrelevant background information, we propose Facial Feature Masking Module (FFMM) to mask out the useless features by setting them to zero. We predict a facial feature mask with the same size of the shared feature map to indicate the units which are significant for face detection. The maps, called stacked mask maps, are then multiplied with the shared feature map to highlight the useful features for the following bounding box regression and face/non-face classification. The masked feature maps contain only useful features for face detection, are thus much more sparse than the original ones. The following convolutions can focus on these features and achieve more efficient face detection. Based on FFMM, a specific dropout, called Mask Constrained Dropout Module (MCDM), has also been designed to drop out the neurons that generate the highlighted features.

3.2. Facial feature masking module (FFMM)

In this subsection, we introduce mask prediction, ground truth mask generation and loss function.

Mask prediction. As mentioned in the framework, given a feature map $\mathbf{X}^l \in \mathbb{R}^{C\times H\times W}$ passed to the l-th detection head, stacked binary masks $\mathbf{S} \in \mathbb{R}^{C\times H\times W}$ need to be generated to enhance the important facial features. As shown in Fig. 1 (b), the main steps are shown as follows:

1. We apply 3×3 convolutions on shared feature maps to generate logits maps $\mathbf{A} \in \mathbb{R}^{2\times H\times W}$, which represent the unnormalized probability of face/non-face for each pixel.

2. The softmax operation is applied on $\mathbf{A}$ to generate probability mask $\mathbf{M} \in \mathbb{R}^{2\times H\times W}$:

$$\begin{aligned} M_i(u,v) &= softmax(A_i(u,v)) \\ &= \frac{e^{A_i(u,v)}}{\sum_{j=1}^{2} e^{A_j(u,v)}},\ \sum_{i=1}^{2} M_i(u,v) = 1 \end{aligned} \tag{1}$$

where (u, v) is a coordinate position. $A_i(u, v)$ is the logit at (u, v) for i-th channel (face/background) of $\mathbf{A}$. $M_i(u, v)$ is the probability of the predicted class i at (u, v). M_1 and M_2 are the probabilities of background and face, respectively.

3. We take out the second channel of the probability mask $\mathbf{M}$, i.e. $M_2 \in \mathbb{R}^{H\times W}$, where the probability of face is stored, to generate a binary mask $\mathbf{B} \in \mathbb{R}^{H\times W}$:

$$B(u,v) = \begin{cases} 1, & M_2(u,v) > Th_{Mask} \\ 0, & otherwise. \end{cases} \tag{2}$$

where $B(u, v)$ is the value at (u, v) of $\mathbf{B}$. Th_{Mask} is a threshold.

4. The mask $\mathbf{B}$ is repeated for C times and then stacked to align with the shared feature maps $\mathbf{X}^l$. The stacked binary masks $\mathbf{S}$ have the same shape as that of the shared feature map $\mathbf{X}^l$.

5. We use Hadamard product (denoted as $\otimes$) to multiply shared feature maps $\mathbf{X}^l$ with the stacked binary masks $\mathbf{S}$ to generate masked feature maps $\widetilde{\mathbf{X}}^l \in \mathbb{R}^{C\times H\times W}$.

6. The masked feature maps $\widetilde{\mathbf{X}}^l$ are passed to the detection head for classification and localization.

The binary map generation branch only contributes to the information propagation in forward computation and is not included in backward computation during training.

Ground truth mask. To train FFMM that generates the proposed facial feature mask, the ground truth mask $\mathbf{G} \in \mathbb{R}^{H\times W}$ of every detection head needs to be created from the input image. Figure 2 shows an example mask generated for the input image with height H_{input} and width W_{input}, where two faces are available. Each pixel in the mask (x_{mask}, y_{mask}) is firstly mapped to a pixel (x_{input}, y_{input}) in the input image as below:

$$\begin{aligned} x_{input} &= (x_{mask} + 0.5)\frac{W_{input}}{W}, \\ y_{input} &= (y_{mask} + 0.5)\frac{H_{input}}{H} \end{aligned} \tag{3}$$

where x_{input} and y_{input} are used as the precise center for the following bounding box location. Based on the receptive field size associated with the head, a set of square boxes centering at (x_{input}, y_{input}), with size ranging from Min to Max, can be generated. We compute IoU between face box and the generated square box, both of which are in floating representation to avoid rounding error. The pixel value of (x_{mask}, y_{mask}) is set as 1 if and only if any of its square boxes has a Jaccard overlap ($> Th_{GT}$) with the ground truth box.

Loss function. To compute the mask loss of FFMM, we should generate the mask label $\mathbf{Y} \in \mathbb{R}^{2\times H\times W}$, which is the one-hot encoding of the ground truth mask $\mathbf{G}$. As the face masks in detection head at shallow level account for small regions, the predictions are tending to background. As a result, the predicted probability for foreground may be small. Therefore, we propose to filter out the nonsignificant units with a threshold Th_{Mask}. As shown in Fig. 1 (b), the cross entropy loss is used to measure the difference between the predicted probability mask $\mathbf{M}$ and ground truth mask $\mathbf{G}$:

$$L_{mask} = -\sum_{u=1}^{H}\sum_{v=1}^{W}\sum_{i=1}^{2} M_i(u,v)\log(Y_i(u,v)) \tag{4}$$

$M(u, v)$ and $Y(u, v)$ are the class prediction and one-hot label on location (u, v), respectively.

3.3. Mask Constrained Dropout Module (MCDM)

As dropout has been widely proved to enhance the generalization performance of network, we also applied the strategy to our ground truth mask $\mathbf{G}$ during the training of face detector. The pixels with value 1 in $\mathbf{G}$ generated in section 3.2 are randomly dropped, i.e. set to 0, with a dropout probability η. Our MCDM can thus be seen as a special case of general dropout strategy. As each pixel in high-level features is very important, we apply our MCDM in low-level features. In addition, Pyramidbox [16] has proven that shared features at adjacent levels have strong relationship. Therefore, we adopt FEM to enhance the features. The enhanced features are multiplied with the stacked binary masks generated by MCDM.

As shown in Fig. 3, in our FEM, a 1×1 convolution, followed by batch normalization and ReLU operations, is firstly used to normalize the two adjacent shared feature maps. The feature map with lower resolution is upsampled to the same size with the higher resolution one, before the element-wise product. Three branches with different dilated convolutions are then applied to the fused feature maps and the outputs are further combined to generate the final enhanced feature map for MCDM.

3.4. Overall Loss Function

As our detector consists of three branches, i.e. FFMM, classification and bounding box regression, the overall loss

Figure 4. Examples of the detection results for UF-SSD+FFMM-×1

function is a combination of different losses to supervise the training of the three branches: $L = \lambda_1 L_{mask} + \lambda_2 L_{C+L}$. λ_1 and λ_2 are the balanced loss weights. While L_{mask} supervises the prediction of facial feature mask, L_{C+L} is the same as RPN [12] and defined as below:

$$L_{C+L}(p_i, t_i) = \frac{\lambda_3}{N_{conf}} \sum_i L_{conf}(p_i, p_i^*) + \frac{1}{N_{loc}} \sum_i p_i^* L_{loc}(t_i, t_i^*), \quad (5)$$

where N_{conf} denotes the total number of positive and negative anchors, and N_{loc} is the number of positive anchors. p_i is the predicted probability of the i-th anchor. The i-th anchor will be positive only when p_i^* is 1, otherwise negative. The classification loss L_{conf} is the softmax loss over two classes including face and non-face. $L_{loc}(t_i, t_i^*)$ is the smooth L_1 loss between the regression prediction t_i and the ground truth offsets t_i^* of the i-th anchor, where $t_i = \{t_x, t_y, t_w, t_h\}_i$ and $t_i^* = \{t_x^*, t_y^*, t_w^*, t_h^*\}_i$. We only compute the localization loss of positive anchors. λ_3 is the balanced loss weight.

4. Experiments

4.1. Dataset and Evaluation Metrics

We carry out all the experiments on the WIDER FACE dataset, which consists of 32203 images and 393703 labeled faces. We use the data in training set (about 13k images) for training, and present ablation study results in terms of average precision (AP) on validation set, which consists of around 3k images. Our main results are reported on both validation set and testing set (about 16k images), in terms of precision recall (PR) curves.

4.2. Training Details

For all the experiments, we take the Universal Face Single Shot MultiBox Detector (UF-SSD) as the detector.

Data Augmentation. As there are many small faces in the original images, we generate new data with larger faces to solve the scale imbalance problem. Following [8, 22], we randomly crop a square patch from the original image with a ratio [0.3, 1] of its shortest side. We only keep the overlapping between the patch and face box when the center of the face box is within the patch. The cropped patch is resized to 640 × 640 and horizontally flipped with a probability of 0.5, followed by photo-metric distortions [5].

Table 1. Ablation studies of FFMM based on VGG-16 [13] backbone on WIDER FACE validation set. Th_{GT} and Th_{Mask} are set to 10^{-3} and 10^{-2}, respectively. "×1" denotes applying FFMM only on $f_{160\times160}$ layer, while "×6" denotes applying FFMM on all shared layers.

Architecture (%)	Easy	Medium	Hard
UF-SSD(VGG-16)	92.10	90.00	77.50
UF-SSD+FFMM-×1	92.40	90.80	**83.50**
UF-SSD+FFMM-×6	**93.40**	**91.60**	80.90

Table 2. Ablation studies of MCDM based on VGG-16 [13] backbone on WIDER FACE validation set. Th_{GT} is set to 10^{-3}. "MCDM-B" denotes applying MCDM after the enhanced feature map $ef_{160\times160}$.

Architecture (%)	Easy	Medium	Hard
UF-SSD(VGG-16)+FEM	93.40	92.10	85.00
UF-SSD+FEM+MCDM-$B_{\eta=0.10}$	**94.00**	**92.70**	**86.00**
UF-SSD+FEM+MCDM-$B_{\eta=0.20}$	93.30	92.20	85.50
UF-SSD+FEM+MCDM-$B_{\eta=0.30}$	93.70	92.40	85.30

Table 3. Efficiency analysis of various methods on WIDERFACE validation set.

Method	Speed
UF-SSD(VGG-16)	55.50(ms)
UF-SSD+FFMM-×1	63.19(ms)
Pyramidbox(VGG-16) [15]	98.17(ms)

Optimization details. We train the detectors for 200K iterations using the stochastic gradient descent (SGD) optimizer. The initial learning rate is 10^{-3} or 2×10^{-4} and divided by 10 after 100K, 150K and 180K iterations. Weight decay and momentum are set to 5×10^{-4} and 0.1, respectively. Th_{Mask} is set to 10^{-2}. The balanced loss weight λ_1, λ_2 and λ_3 are set to 1.0, 1.0 and 4.0, respectively.

4.3. Ablation Studies

For ablation studies, we use VGG-16 [13] as the backbone and take the outputs conv3_3, conv4_3, conv5_3, conv_fc7, conv6_2 and conv7_2 from the backbone to carry out detections.

FFMM. As shown in the second row in Tab. 1, UF-SSD+FFMM-×1 obtains improvements of 0.3%, 0.8% and 6.0% on easy, medium and hard subsets, respectively. While those of UF-SSD+FFMM-×6 are 1.3%, 1.6% and 3.4%, respectively. The results show that FFMM can boost detection performance, especially on the hard subset. Moreover, applying FFMM on more heads is helpful to detect easy and medium faces, but the improvement on hard subset drops a little.

MCDM. As shown in the first row in Tab. 2, we take UF-SSD(VGG-16)+FEM as the baseline for comparisons.

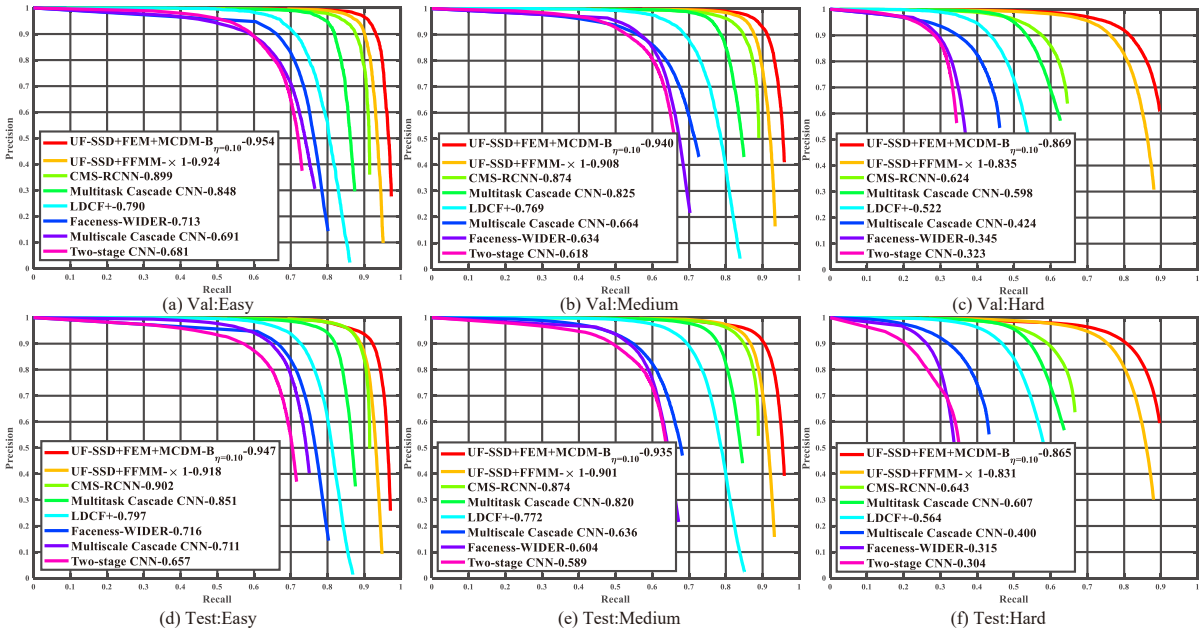

Figure 5. Precision-recall (PR) curves on WIDER FACE validation and testing sets.

The second row in the table indicates that when η is 0.10, the baseline integrated with MCDM-B gets the best performances of 94.00%, 92.70% and 86.00%, with improvements of 0.6%, 0.6% and 1.0% on easy, medium and hard subsets, respectively. It indicates that MCDM can effectively boost the detection performance on all subsets, which justifies the effectiveness of our approach.

In summary, both FFMM and MCDM can notably boost the detection performance.

4.4. Comparisons with State-of-the-art Methods

After ablation studies, we compare our approaches with state-of-the-art methods in Fig. 4. The UF-SSD+FEM+MCDM-$B_{\eta=0.10}$ uses ResNet-152 (RES-152) as the backbone [4]. The PR curves of our UF-SSD+FEM+MCDM-$B_{\eta=0.10}$ and UF-SSD+FFMM-$\times$1 are at the top of that of other state-of-the-art methods on all subsets. In particular, UF-SSD+FEM+MCDM-$B_{\eta=0.10}$ achieves the best performances of 95.4%, 94.0%, 86.9% on easy, medium and hard subsets, respectively, which is a new state-of-the-art. UF-SSD+FFMM-$\times$1 also surpasses all other state-of-the-art methods, such as [19, 23, 18, 9, 20], especially on hard subset that contains massive tiny faces, except for our UF-SSD+FEM+MCDM-$B_{\eta=0.10}$. We can see that the two curves on validation and testing sets are very similar. Specifically, the accuracy of both methods drops 0.4% from validation sets to testing ones on hard subset, which justifies the generalization ability of two models. As shown in Fig. 4, our UF-SSD+FFMM-$\times$1 can detect many tiny faces. In particular, UF-SSD+FFMM-$\times$1 detects 734 faces of the World Largest Selfie image shown at Fig. 4, which confirms the effectiveness of our proposed FFMM on detection of tiny faces.

4.5. Efficiency Analysis

We now test the efficiency of our approach with that of literature works like Pyramidbox and UF-SSD (VGG-16). Table 3 shows the mean speed to process an image in the WIDER FACE validation set, recorded on a GTX 1080Ti GPU. One can observe from the table that the mean processing time of our approach is 63.19ms, which is similar with that of UF-SSD (VGG-16) and much more efficient than Pyramidbox. FFMM filters out the unimportant units which may generate low-quality predictions. The decrease of low-quality predictions reduces the process time of NMS. Therefore, our model is more efficient.

5. Conclusion

In this paper, we propose the facial feature masking module (FFMM) and mask constrained dropout module (MCDM) for face detection. FFMM can be regarded as a screening on feature maps to obtain significant units for detection. As the masked feature map is sparse, the computation cost can be saved. MCDM randomly drops the significant units of feature maps to make them more robust and thus generate more useful features.

Both of our approaches can boost the detection performance. We will improve the proposed modules in our future work, for example, by applying fewer FFMMs without decrease of the performance. We will also improve the empirical threshold in FFMM to make it adaptive.

References

[1] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

[2] J. Deng, J. Guo, Y. Zhou, J. Yu, I. Kotsia, and S. Zafeiriou. Retinaface: Single-stage dense face localisation in the wild. *arXiv preprint arXiv:1905.00641*, 2019.

[3] R. Girshick. Fast r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 1440–1448, 2015.

[4] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[5] A. G. Howard. Some improvements on deep convolutional neural network based image classification. *arXiv preprint arXiv:1312.5402*, 2013.

[6] J. Li, Y. Wang, C. Wang, Y. Tai, J. Qian, J. Yang, C. Wang, J. Li, and F. Huang. Dsfd: dual shot face detector. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5060–5069, 2019.

[7] Z. Li, X. Tang, J. Han, J. Liu, and R. He. Pyramidbox++: High performance detector for finding tiny face. *arXiv preprint arXiv:1904.00386*, 2019.

[8] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg. Ssd: Single shot multibox detector. In *European conference on computer vision*, pages 21–37. Springer, 2016.

[9] E. Ohn-Bar and M. M. Trivedi. To boost or not to boost? on the limits of boosted trees for object detection. In *2016 23rd international conference on pattern recognition (ICPR)*, pages 3350–3355. IEEE, 2016.

[10] M.-T. Pham and T.-J. Cham. Fast training and selection of haar features using statistics in boosting-based face detection. In *2007 IEEE 11th International Conference on Computer Vision*, pages 1–7. IEEE, 2007.

[11] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 779–788, 2016.

[12] 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.

[13] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[14] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1):1929–1958, 2014.

[15] X. Tang, D. K. Du, Z. He, and J. Liu. Pyramidbox: A context-assisted single shot face detector. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 797–813, 2018.

[16] P. Viola and M. J. Jones. Robust real-time face detection. *International journal of computer vision*, 57(2):137–154, 2004.

[17] M. Xu, L. Cui, P. Lv, X. Jiang, J. Niu, B. Zhou, and M. Wang. Mdssd: Multi-scale deconvolutional single shot detector for small objects. *arXiv preprint arXiv:1805.07009*, 2018.

[18] S. Yang, P. Luo, C.-C. Loy, and X. Tang. From facial parts responses to face detection: A deep learning approach. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3676–3684, 2015.

[19] S. Yang, P. Luo, C.-C. Loy, and X. Tang. Wider face: A face detection benchmark. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5525–5533, 2016.

[20] K. Zhang, Z. Zhang, Z. Li, and Y. Qiao. Joint face detection and alignment using multitask cascaded convolutional networks. *IEEE Signal Processing Letters*, 23(10):1499–1503, 2016.

[21] 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 *2017 IEEE International Joint Conference on Biometrics (IJCB)*, pages 1–9. IEEE, 2017.

[22] S. Zhang, X. Zhu, Z. Lei, H. Shi, X. Wang, and S. Z. Li. S3fd: Single shot scale-invariant face detector. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 192–201, 2017.

[23] C. Zhu, Y. Zheng, K. Luu, and M. Savvides. Cms-rcnn: contextual multi-scale region-based cnn for unconstrained face detection. In *Deep learning for biometrics*, pages 57–79. Springer, 2017.

[24] Q. Zhu, M.-C. Yeh, K.-T. Cheng, and S. Avidan. Fast human detection using a cascade of histograms of oriented gradients. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, volume 2, pages 1491–1498. IEEE, 2006.