EResFD: Rediscovery of the Effectiveness of Standard Convolution for Lightweight Face Detection

Joonhyun Jeong[1,2] Beomyoung Kim[1,2] Joonsang Yu[1,3] YoungJoon Yoo[1]
[1]NAVER Cloud, ImageVision [2]KAIST [3]NAVER AI Lab
{joonhyun.jeong,beomyoung.kim,joonsang.yu,youngjoon.yoo}@navercorp.com

Abstract

This paper analyzes the design choices of face detection architecture that improve efficiency of computation cost and accuracy. Specifically, we re-examine the effectiveness of the standard convolutional block as a lightweight backbone architecture for face detection. Unlike the current tendency of lightweight architecture design, which heavily utilizes depthwise separable convolution layers, we show that heavily channel-pruned standard convolution layers can achieve better accuracy and inference speed when using a similar parameter size. This observation is supported by the analyses concerning the characteristics of the target data domain, faces. Based on our observation, we propose to employ ResNet with a highly reduced channel, which surprisingly allows high efficiency compared to other mobile-friendly networks (e.g., MobileNetV1, V2, V3). From the extensive experiments, we show that the proposed backbone can replace that of the state-of-the-art face detector with a faster inference speed. Also, we further propose a new feature aggregation method to maximize the detection performance. Our proposed detector EResFD obtained 80.4% mAP on WIDER FACE Hard subset which only takes 37.7 ms for VGA image inference on CPU. Code is available at https://github.com/clovaai/EResFD.

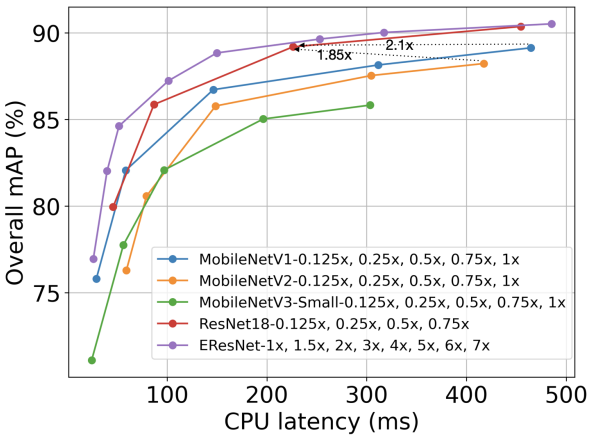


Figure 1. Latency-accuracy Pareto curve. We investigate the latency trend of depthwise separable convolution-based backbones (MobileNets) and standard convolution-based backbones (ResNets) according to adjusting width multiplier on RetinaFace [7] framework. ResNet shows much lower latency compared with MobileNets family even though its mAP is higher than others. One step forward to the observation, we propose a modified ResNet backbone for face detection task, abbreviated as EResNet, which reports far more accurate and faster performance than the others.

1. Introduction

Face detection research has demonstrated significant performance improvement after the advent of recent deep neural network based general object detection approaches such as one-stage detector (*e.g.*, SSD [27], YOLO [34], RetinaNet [25], EfficientDet [41]) and two-stage detector (*e.g.*, Faster R-CNN [35], FPN [24], Mask R-CNN [12], Cascade R-CNN [3]). For applicability in a real-world scenario, real-time face detection has attracted more attention, and recent face detectors commonly adopt the one-stage approach that is simpler and more efficient than the two-stage approach.

Recent studies for real-time face detection methods frequently use the lightweight model consisting of depth-wise separable convolution, which is used in MobileNet [16] and ShuffleNet [59]. Specifically, recent lightweight face detectors, such as RetinaFace [7], SCRFD [9], and CRFace [44], employ MobileNetV1 architecture [17] as the backbone network and reduce the number of channels in depthwise separable convolution layers by adjusting the width multiplier. Following the paradigm of residual block [13], BlazeFace [1] proposed BlazeBlock that consists of depthwise separable convolution layers with skip connection, achieving a stronger performance. In practice, adopting the depthwise separable convolution is a reasonable choice to save the number of floating point operations (FLOPs), which is one of the important measurements for the real-time application. Summing up the following common practice, most

real-time face detectors utilize the depthwise separable convolution layers in their model by default [7, 9, 46].

In this paper, we rethink the common belief for the depthwise separable convolution layer and found out that the standard convolution with reducing the number of channels can achieve a better trade-off between latency and detection performance than depthwise separable convolution. Here, we use ResNet18 [13] as our baseline backbone network for the standard convolution and compare with the depthwise separable convolution-based backbone networks (MobileNetV1 [16], MobileNetV2 [36], and MobileNetV3 [15]). Figure 1 shows the latency and average of mean average precision (mAP) scores on WIDER FACE [47] Easy, Medium, and Hard subsets. ResNet18 demands much higher latency than MobileNet when the width multiplier is not applied ($i.e.$, 1x). However, ResNet18 becomes much faster than MobileNet with higher mAP when reducing the number of channels using the width multiplier. Note that ResNet18-0.5x denotes that width multiplier 0.5 is applied, and it is 2.1 times faster than MobileNetV1-1.0x and 1.85 times faster than MobieNetV2-1.0x even though its mAP is higher than others.

Based on the observation, we propose EResFD, which is a ResNet-based real-time face detector. We firstly propose a slimmed version of ResNet architecture, namely EResNet, by redesigning the new stem layer, and changing the block configuration. Those methods can effectively reduce the inference latency and achieve higher detection accuracy compared with ResNet18. Secondly, we also propose the new feature map enhancement modules; Separated Feature Pyramid Network (SepFPN) and Cascade Context Prediction Module (CCPM). SepFPN aggregates information from high-level and low-level feature maps separately, and CCPM further effectively captures diverse receptive fields by employing a cascade design. Equipped with these architectural designs, our EResFD achieved 3.1% higher mAP on WiderFace Hard subset compared to the state-of-the-arts lightweight face detectors such as FaceBoxes [56].

We summarize the main contributions as follows:

- We propose a ResNet-based extremely lightweight backbone architecture, which is much faster than the baselines on the CPU devices, achieving state-of-the-art detection performance.
- We analyze the behavior of both standard convolution and depthwise separable convolution, and we found that the standard convolution is much faster than the depthwise separable convolution under extremely lightweight parameter constraints.
- We propose a channel dimension preserving strategy to reduce the latency, fitting the number of layers in each layer group to recover the performance degeneration.
- We propose a latency-aware feature enhance module, SepFPN, and CCPM. These enhance modules improve the detection performance on all (large, medium, small) face scales, with much faster speed compared to previous enhance modules.

2. Related Works

Face Detectors Recent face detectors [1, 4, 7, 9, 14, 20, 22, 28–33, 37, 43, 48, 50, 51, 58, 60, 61] achieved impressive performance enhancement. These face detectors inherit the architectural improvement of the general object detectors such as SSD [27] and RetinaNet [25] or two-stage detectors such as Faster R-CNN [35]. The improvement in face detection enabled to detect faces with various densities and scales. To detect dense and small-scale faces, current state-of-the-art group of detectors [7, 9, 20–22, 29, 43, 61] mostly employ large-scaled classification networks with custom-designed upsampling blocks. Besides the ResNet families [13], prototypical choice, various attempts including those from architecture search [50] have been applied. PyramidBox series [22, 43] and DSFD [21] suggested own upsampling blocks to improve the expressiveness of the features for dealing with finer faces. RetinaFace [7], currently the dominant one, infers five-point keypoint landmarks of the faces: eyes, nose, mouth, in addition to the detection box, similar to MTCNN [52]. However, the large memory size requirements of these face detectors critically hinder their applicability on edge devices. Here, we target on reducing the weight parameters of the backbone network to increase the usability of the face detectors.

Lightweight Face Detectors To run the above-mentioned face detectors on mobile or CPU devices, some of the detectors provide their lighter version, mostly substituting their backbones to lighter classification networks utilizing depthwise convolution [17]. After advent of the pioneering works from MobileNetV1 [17] and V2 [36] utilizing depthwise separable convolution and inverted bottleneck block, more refinements [11, 15, 39, 40, 45] on the architectures have brought the performance enhancements. These architectures show the competitive ImageNet [6] classification accuracy to larger classification models and also for the transferred tasks like object detection [27, 35] and segmentation [5]. Following the improvement of the lightweight backbone networks, RetinaFace [7], SCRFD [9], and YuNet [46] use channel-width pruned version of MobileNet [17]. BlazeFace [1] and MCUNetV2 [23] proposed new variants of MobileNet targeting on mobile GPU and CPU environment, and EXTD [48] recursively uses the inverted bottleneck block of the MobileNet for further slimming the network size. Besides the overall tendency of using depthwise separable convolution-based backbones, KPNet [37] proposes its own

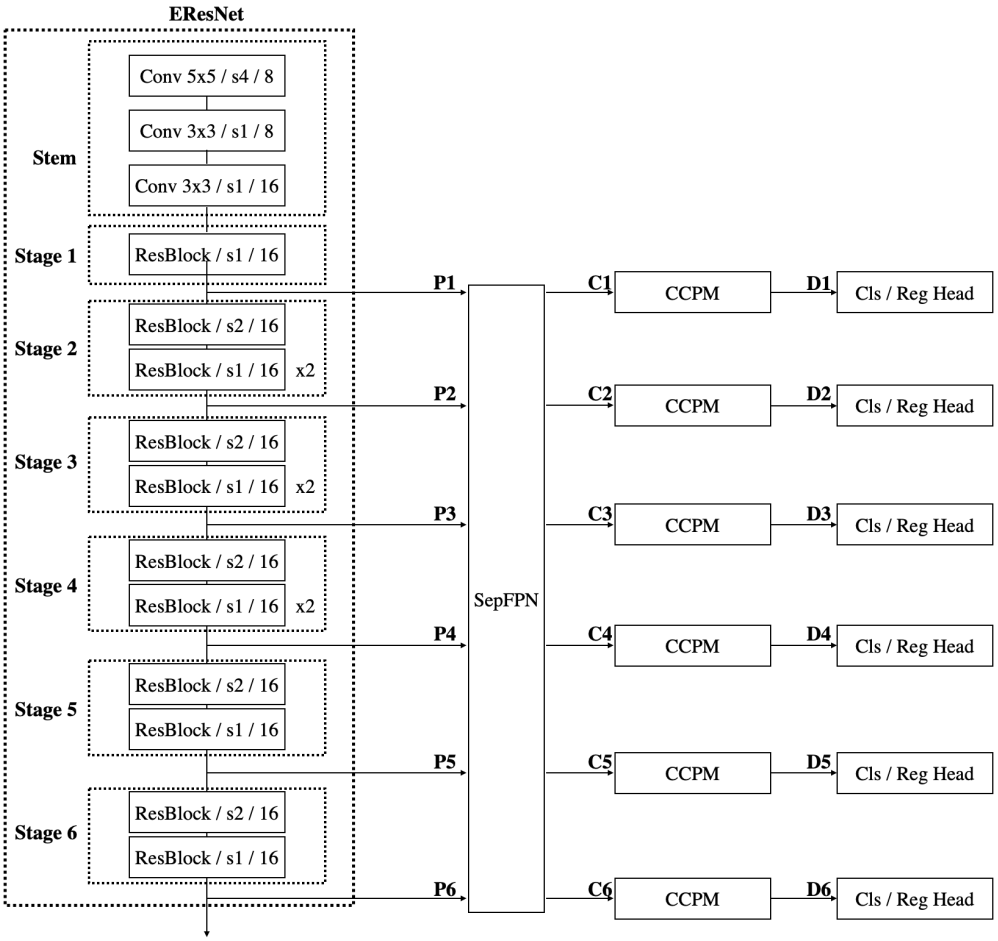


Figure 2. Entire architecture of EResFD. The proposed architecture consists of EResNet with 31 weighted backbone layers, Separated Feature Pyramid Network (SepFPN), and Cascade Context Prediction Module (CCPM). ResBlock denotes the basic residual block, which was proposed in [13]. The first ResBlock of each stage has stride of 2, and every ResBlock has the same number of output channels as 16 in the case of EResFD-1x. For the classification and regression head, a single 1x1 convolution layer is used.

backbone network consisting of standard convolutional network, but its size is still large for edge devices, about 1 million parameters, and focuses on sparse and large scaled faces. In this paper, we rediscover the efficiency of standard convolution layers, which can cover faces with various scales and densities, under extremely lightweight model size and minimal inference time.

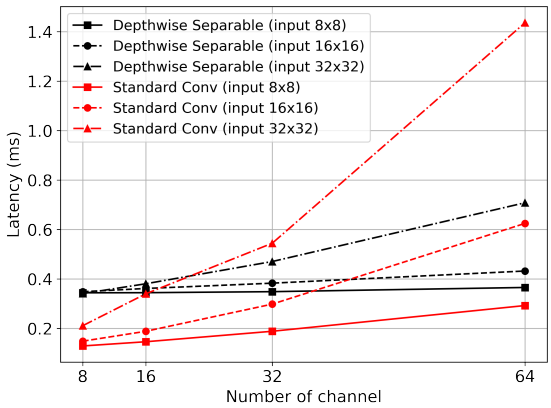
3. EResFD

As seen in Figure 1, ResNet with standard convolution achieves both faster inference time and higher detection performance compared to the widely used backbone, MobileNets [15, 17, 36] which heavily uses depthwise separable convolution layers. From this observation, we revisit the ResNet architecture. Figure 2 illustrates the proposed face detection architecture, named as efficient-ResNet (EResNet) based Face Detector, EResFD. It consists of two main parts; modified ResNet backbone architecture and newly proposed feature enhancement modules. We modify several parts of ResNet to reduce the latency while preserving the detection performance based on empirical analysis on the network, and we also propose both the new feature pyramid module and context prediction module, which are called Separated Feature Pyramid Network (SepFPN) and Cascade Context Prediction Module (CCPM), respectively. Both modules improve the detection performance, and also show comparable or even faster latency compared to previous state-of-the-art CPU detectors [18, 57].

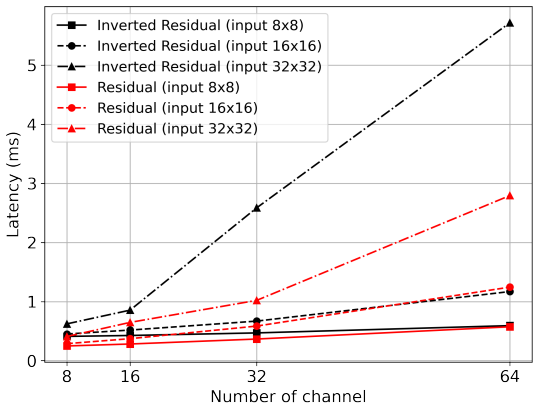
3.1. Rethinking ResNet Architecture

3.1.1 Convolutional Layer Analysis

Depthwise separable convolution is introduced to reduce the multiplication and accumulation cost of the convolution, which occupy most of the computation time during the inference. Table 1 shows the comparison of computational cost between the standard and depthwise separable



(a) standard convolution vs depthwise separable convolution



(b) residual block (ResNet) vs inverted bottleneck block (MobileNetV2)

Figure 3. Illustration of latency comparison with varying channel size: (a) standard and depthwise separable convolution, (b) residual block (ResNet) and inverted bottleneck block (MobileNetV2).

Table 1. Comparison of computational cost between standard and depthwise separable convolution. We calculate the FLOPs count for three kinds of setting, and each value indicates the multiply-add count for single layer.

Type	Standard	Depthwise Separable	
Operation	3x3 Conv	Depthwise Conv	Pointwise Conv
FLOPs (H,W=16, C=16)	1.18M	0.07M	0.13M
FLOPs (H,W=16, C=32)	4.71M	0.15M	0.52M
FLOPs (H,W=16, C=64)	18.87M	0.29M	2.10M

convolution for each stage. Depthwise separable convolution has much smaller FLOPs, and hence it can significantly reduce the computational cost. However, previous work [2] claimed that FLOPs is not always matched with actual latency. Latency can be bounded by memory access and hardware accelerator, $i.e.$, CPU or GPU, so the target hardware characteristic should be considered for the network design.

To check the relationship between FLOPs and latency, we investigate the behavior of both standard convolution and depthwise separable convolution on CPU. We measured the latency of both convolutional layers on CPU, and Figure 3a shows the comparison result. Considering faster inference with small-sized input image ($e.g.$, less than 320x), we tested with input sizes 8×8, 16×16, and 32×32. As input size increases, the latency of standard convolution is steeply increasing, but depthwise separable convolution shows a small amount of latency growth. However, standard convolution achieves smaller latency than depthwise separable convolution on the extremely lightweight condition. For all the input sizes, standard convolution is faster than depthwise separable convolution when its channel dimension is equal to or smaller than 16 as shown in Figure 3a. Since ResNet and MobileNets each consist of standard convolution and depthwise separable convolution, respectively, we can conclude that ResNet has a chance to become faster than MobilNets when we extremely reduce the channel size.

Furthermore, we also analyze the block-level behavior of each convolutional layer. We use residual block and inverted residual block, which consist of standard and depthwise separable convolution, respectively. The residual block consists of two standard 3×3 convolution. MobileNetV2 has an inverted residual block, which includes one depthwise convolution and two pointwise convolution. The inverted residual block commonly expands the number of channels for the depthwise convolution, which is called the expansion ratio. In MobileNetV2, the expansion ratio is set to 6 for most inverted residual blocks [36], which is reported to preserve the classification ability of the block compared to standard convolution counterpart [10]. Here, we use the equivalent expansion ratio for the latency comparison. Figure 3b shows the block-level latency, and we found that residual block is much faster than the inverted residual block in most cases. The residual block has 9.43M FLOPs and the inverted residual block has 7.18M FLOPs when the input size is 16×16 and the number of channels is 32. Even though residual block has more multiply-add operations, its latency is faster than inverted residual block.

The latency trend of each layer and block shows that standard convolution has a chance to surpass the depthwise separable convolution in terms of the latency. This trend is also the same on network-level analysis as we mentioned in Section 1. Therefore, we propose an efficient backbone originating from the ResNet.

Table 2. Latency breakdown of ResNet18-0.25x model. Stem denotes 7x7 convolution layer followed by maxpool layer, which reduces spatial size by 4 times. For Stage $1 \sim 4$, strides of output feature map are set to $4 \sim 32$.

Component	Latency (ms)	Ratio (%)
Stem	24.1	44
Stage 1	10.5	19
Stage 2	7.5	13
Stage 3	6.6	12
Stage 4	6.5	12
Total	55.2	100

Table 3. Latency of stem layers on ResNet18-0.25x model. Ratio denotes the portion of stem latency compared to the overall network latency.

Stem	ResNet	EResNet
Stem FLOPs	180.6 M	11.5 M
Stem Latency (Ratio)	24.1ms (44%)	4.5ms (13%)

3.1.2 Stem Layer Modification

We further thoroughly analyzed the ResNet architecture and observed that the stem layer occupies a large amount of entire latency. Table 2 shows the latency breakdown of ResNet18 with width multiplier 0.25, and it shows that almost half of the total latency is originated from stem layers. The backbone network is highly lightened by applying the small width multiplier, so the proportion of the stem layer becomes larger. Moreover, ResNet stem layer consists of 7×7 convolution with stride of 2, so it requires a large amount of computation compared to others. The number of computations (FLOPs) is proportional to the square of kernel size (K^2) and reciprocal-square of stride ($1/S^2$). If kernel size is reduced to 5, its FLOPs becomes about 50% of 7×7 convolution, and FLOPs further decrease to about 13% when its stride of 4 is applied simultaneously.

To reduce the latency of stem layers, we first change stride to 4 for the convolutional layer, which is already adopted in the previous work [57].We also reduce the kernel size from 7 to 5, but it can hurt the detection performance because it is directly related to the receptive field. To alleviate this problem, we introduce two additional convolutional layers right after 5×5 convolution. Owing to those convolutional layers, the receptive field size becomes larger than the original stem layer, but its computation complexity is still much lower than the original. Table 3 shows comparison results on the stem layer. By adopting a smaller kernel size and bigger stride, EResNet stem layer has much smaller FLOPs, and also achieves much shorter latency compared to the original ResNet stem layer.

Table 4. Latency breakdown of ResNet18 models where channels are doubled or preserved for stage 2,3,4. Width multiplier is set to be 0.25 for ResNet-preserved model to keep number of output channels as 16 for all the stages.

Model		ResNet	ResNet-Preserved
Stage 2	Latency (ms)	7.5	4.2
	FLOPs (M)	157.3	45.5
Stage 3	Latency (ms)	6.6	1.6
	FLOPs (M)	157.3	11.4
Stage 4	Latency (ms)	6.5	1.1
	FLOPs (M)	157.3	2.8

3.1.3 Architecture Reconfiguration for Face

In modern backbone architecture, The number of channels is continuously increasing from the bottom to the top layer [13, 38, 39]. In ResNet, for example, the channel dimension is doubled when its spatial dimension is decreased (stride of 2). This designing trend is based on that high-level features are highly related to the specific classes [49]. When the number of object classes increases, high-level feature dimension has to be enlarged accordingly. However, there is only one object class in the face detection task, so we suppose that the channel dimension may be reduced compared with the object detection task.

To accelerate face detection speed, EResNet backbone is designed by reducing the number of channels. From the assumption, we propose the channel dimension preserving strategy, which means that we do not double the channel dimension for every stage. Table 4 shows the number of FLOPs and latency when our channel-preserving strategy is applied, and it shows that our method significantly reduced the latency and FLOPs of each stage. We note that the number of FLOPs is also proportional to input and output channel dimension, and hence the amount of the reduction is huge. The latency reduction amount is not as large as FLOPs reduction, but it is still remarkable. Based on the intuition (Figure 3a) that standard convolution is faster than depthwise separable convolution under the channel dimension less than 16, we set the channel dimension of our EResNet-1x architecture to 16.

The network capacity also decreases due to the channel dimension reduction, so we adjust the stage configuration to compensate for the performance degeneration. Figure 2 shows the detailed stage configurations. We insert one more block for stages $2\sim4$ to improve small face detection performance and add two extra stages (stages 5 and 6) for the large face. The number of residual blocks increases from 8 to 14, so additional residual blocks can increase the inference time. However, the channel-preserving strategy significantly reduces FLOPs, and the computational cost of each block is also much smaller than the original residual block. For this reason, EResNet architecture is still much faster than the vanilla ResNet architecture as shown in Figure 1.

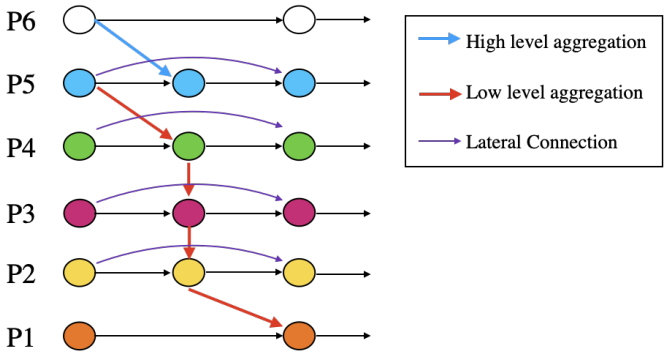


Figure 4. Architecture of SepFPN. P1∼P6 denotes the intermediate features from low to high-level layers. The high-level features and low-level features are aggregated separately.

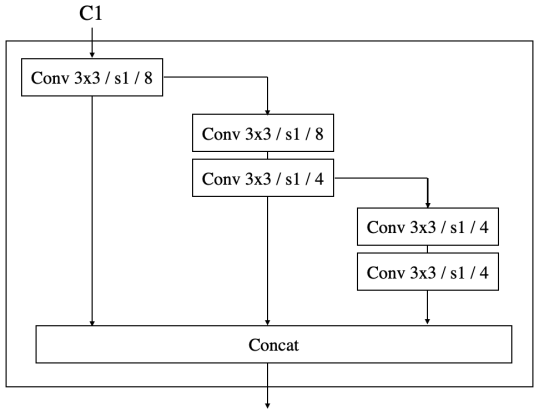


Figure 5. Architecture of CCPM in case of EResFD-1x. We only visualize CCPM for the feature map C1 in Figure 2 for simplicity.

3.2. Feature Enhance Modules

3.2.1 SepFPN

To improve the detection performance for small objects, feature pyramid network (FPN) [24] is widely adopted [7, 43, 57]. FPN propagates the context of high-level features from deep layers into low-level features from shallow layers (top-down), enriching the low-level features to better detect the small objects. However, previous work [43] claimed that aggregating high-level features onto low-level features can hurt detection performance for small faces. This is because the large receptive field of the high-level features might convey irrelevant global contexts to the low-level features, obscuring their local contexts and thereby impeding their detection ability of small local faces.

To resolve this, we propose a new FPN module, separated feature pyramid network (SepFPN, Figure 4). From the above-mentioned observation [43], we assume that a significant disparity of receptive field among the aggregation features can lead to performance degradation. To address this, we separately organize the aggregation features in a hierarchical manner. Specifically, we ensure that high-level features are aggregated solely with other high-level features, while low-level features are exclusively combined with other low-level ones. Each of these two separated top-bottom paths shares similar contexts with similar sizes of receptive field within its aggregation group, consistently enhancing the detection ability across all the scales of faces.

For the aggregation details, we follow BiFPN [42] to aggregate features in a learnable manner with a simple element-wise weighted summation, where the latency overhead is negligible. We also introduce a lateral connection to avoid dilution of each original feature, as in BiFPN. Meanwhile, although BiFPN and several heavyweight object detectors [26, 42] proposed to append an additional bottom-up aggregation path ($i.e.$, from low to high-level features), we do not employ this scheme due to its large latency overhead.

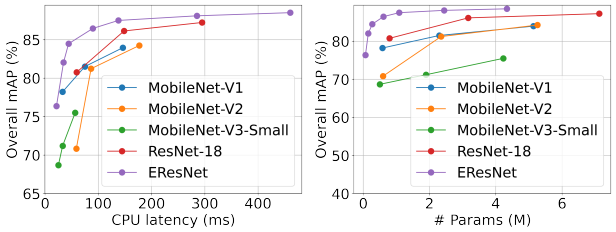
3.2.2 CCPM

To further supplement the feature information, we also propose cascade context prediction module (CCPM, Figure 5) with a latency-aware module design. While the context prediction module [22, 30, 55] was originally proposed to enlarge the receptive field, our cascade design of CCPM aims for the same objective but promotes faster latency. Specifically, our cascade structure can effectively enrich the large size of receptive field by reusing the previously convolved features, while ensuring faster speed than the previous heavyweight enhance modules using a large number of convolution layers [30], densely-connected convolution layers [22] and convolution layers with large asymmetric kernel [55]. Owing to these advantages, CCPM helps to construct a highly efficient face detector that achieves high detection performance with low latency.

4. Experiment

In this section, we evaluate our proposed EResFD by analyzing the effectiveness of each component of EResFD and by comparing with the state-of-the-art (SOTA) face detectors. For quantitatively measuring the accuracy of detection, we used WIDER FACE [47] dataset. For training on WIDER FACE, color distortion, zoom-in and out augmentation, max-out background label, and multi-task loss are used, following S3FD [58]. For evaluation, we employed flip and multi-scale testing [58], where all these predictions are merged by Box voting [8] with intersection-over-union (IoU) threshold at 0.3. In the case of using RetinaFace framework [7][1] , we used single-scale testing where the original image size is maintained. For measuring latency, we used Intel Xeon CPU (E5-2660v3@2.60 GHz) with VGA input resolution (480×640).

[1]We obtained source code from https://github.com/biubug6/Pytorch_Retinaface



(a) latency vs accuracy (b) parameter vs accuracy

Figure 6. Performance comparison of various backbone networks in terms of CPU latency and number of parameters. We applied width multipliers 1x, 1.5x, 2x, 3x, 4x, 6x, 8x for EResNet. For the other backbones, we applied width multipliers 0.25x, 0.5x, 0.75x.

Table 5. Performance comparison of various FPN modules on WIDER FACE.

Model	Latency (ms)	mAP (%)			
		Easy	Medium	Hard	Overall
EResNet	20.9	85.09	82.78	61.20	76.36
+ FPN [24]	24.0	85.30	84.25	75.45	81.67
+ LFPN [43]	23.8	85.24	84.21	76.58	82.01
+ PANet [26]	35.5	87.96	86.82	77.95	84.24
+ BiFPN [42]	35.6	87.16	85.95	77.56	83.56
+ SepFPN (Ours)	**27.0**	87.68	86.30	77.68	83.89

4.1. Component Study

Backbone Network. Figure 6 shows the comparison result with widely used lightweight backbone; ResNet18, MobileNetV1, V2, and V3. The experimental results show that EResNet achieves superior inference latency given the similar mAP condition and also has higher mAP given the similar latency condition, as shown in Figure 6a. Our proposed stem layer and channel dimension preserving strategy are shown to be very helpful for latency reduction, while maintaining the powerful face detection performance. In addition, EResNet also outperforms other comparison methods in terms of the number of parameters. In Figure 6b, EResNet shows the highest mAP with a much smaller number of parameters. To further prove the general effectiveness of EResNet backbone, we additionally compared it with various backbone architectures on RetinaFace framework in Figure 1. For all the backbones, we only employed 3 detection heads from P2 $\sim$ P4 in Figure 2, following [7]. The results further corroborate that our EResNet architecture has the best latency-accuracy trade-off among the various backbones. From those experiments, we found that the proposed methods effectively reduce both latency and parameters without causing mAP degeneration.

SepFPN. We measured the detection performance and latency of several other FPN modules on the EResNet backbone, in Table 5. Compared to the lightweight FPN modules such as FPN and LFPN, our SepFPN achieves much more detection accuracy gain with a small increase of latency. Meanwhile, compared to the heavyweight FPN modules ($i.e.$, PANet, BiFPN) with bottom-up aggregation path, our SepFPN achieves comparable or even higher detection performance, while exhibiting 24% shortened latency. This experiment shows that the bottom-up path would not be an essential block for efficient face detection. We further empirically studied on the separation position of SepFPN, and Table 6 shows the result. The latency is not significantly affected by the separation position, but the accuracy is very sensitive according to the separation position. We observed that P5 achieves the best mAP for all different kinds of face sizes, and hence we applied P5 for all other experiments.

Table 6. Ablation study of SepFPN with various separation position. In case of separation position is 5 (Figure 4), high level features are only aggregated from P6 to P5 and the rest low-level features are aggregated from P5 to P1.

Separation Position	Latency (ms)	mAP (%)			
		Easy	Medium	Hard	Overall
P3	26.9	85.95	84.39	75.31	81.88
P4	26.7	87.16	85.70	77.12	83.33
P5	27.0	87.68	86.30	77.68	83.89

Table 7. Performance comparison of various feature enhance modules before detection head on WIDER FACE. For fair comparisons, we fix baseline backbone network as EResNet-1x equipped with LFPN [43].

Model	Latency (ms)	mAP (%)			
		Easy	Medium	Hard	Overall
Baseline	23.8	85.24	84.21	76.58	82.01
+ SSH [30]	35.5	87.49	86.34	79.28	84.37
+ CPM [43]	42.4	87.47	86.74	80.00	84.74
+ FEM [20]	41.5	86.90	86.15	79.22	84.09
+ DCM [22]	48.1	87.48	86.51	79.85	84.61
+ CCPM (ours)	**33.8**	87.25	86.38	79.90	84.51

CCPM. Table 7 shows a performance comparison result of various feature enhance modules. The feature enhance module makes large performance gain, but several previous works [20, 22, 43] show similar detection performance. Our CCPM module mainly focuses on latency reduction, and experimental result shows that CCPM achieves the fastest latency, satisfying the purpose. In addition, CCPM also achieves higher overall mAP than SSH, which is the fastest among all the previous methods mentioned in the Table.

4.2. Comparison with SOTA Detectors

We compare our proposed method with the SOTA real-time CPU detectors on WIDER FACE validation dataset. Table 8 (upper part) shows the comparison result. **EResNet-**

Table 8. Comparison with previous works on WIDER FACE validation set. All models are evaluated with multi-scale testing, following [7, 58]. For measuring FLOPs and Latency, VGA resolution (480×640) is used. For MTCNN, we used input sizes designated by [53].

Method	Backbone	Feature Enhance Module	# Params	# FLOPs	Latency	mAP (%)			
						Easy	Medium	Hard	Overall
MTCNN [53]	P-,R-,O-Net [53]	-	0.12M	14M	4.0ms	85.10	82.00	60.70	75.93
FaceBoxes [57]	FaceBoxes [57]	FPN + DCH	0.66M	156M	35.7ms	88.50	86.20	77.30	84.00
RetinaFace [7]	MobileNetV1-0.25x [17]	FPN + SSH	0.42M	754M	58.5ms	88.67	87.09	**80.99**	85.58
EResFD	**EResNet-1x**	-	0.07M	228M	20.9ms	85.09	82.78	61.20	76.36
EResFD	**EResNet-1x**	**SepFPN**	0.08M	250M	27.0ms	87.68	86.30	77.68	83.89
EResFD	**EResNet-1x**	**SepFPN + CCPM**	0.09M	298M	37.7ms	**89.02**	**87.96**	80.41	**85.80**

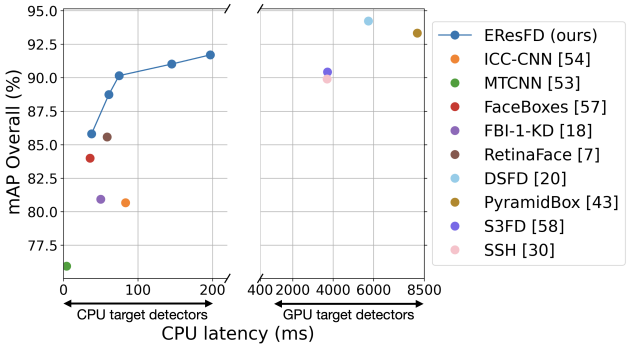
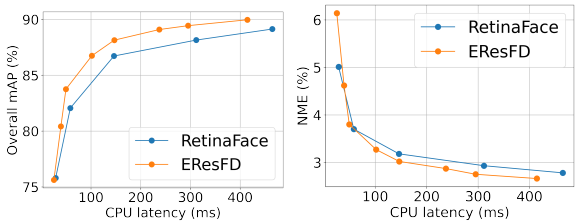


Figure 7. Performance comparison of EResFD with other SOTA CPU target detectors [7, 18, 53, 54, 57] and GPU target detectors [20,30,43,58]. For RetinaFace [7], MobileNetV1-0.25x backbone was used.



(a) Detection (b) Landmark detection

Figure 8. Performance of face detection on WIDER FACE and landmark detection on AFLW [19] dataset. Based on RetinaFace framework where face bounding boxes with facial landmarks can be jointly detected, we only replaced the backbone network from MobileNetV1 to EResNet, while FPN and SSH are replaced by our proposed SepFPN and CCPM, respectively for EResFD.

1x indicates EResNet backbone architecture shown in Figure 2, with width multiplier 1. In the case of RetinaFace [7] backbone, width multiplier 0.25 is applied. MTCNN shows the smallest FLOPs and Latency, but it has large mAP degradation for medium and hard case. EResFD has the smallest number of parameters and also achieves the highest overall mAP. The latency of EResFD is similar to that of FaceBoxes [56], but its detection performance is much higher. Moreover, the proposed method achieves similar or slightly higher mAP compared with RetinaFace [7], but its latency is about 64% of that of RetinaFace.

Table 8 (bottom part) also shows the ablation study for the proposed modules; SepFPN and CCPM. SepFPN improves the overall mAP by about 7.5%, but its latency only increases by 6 *ms*. Moreover, we also achieve 1.9% overall mAP improvement when CCPM is further applied. We observed that proposed modules can make a large performance improvement even when jointly applied.

In addition, we also compare the latency and detection performance with other CPU and GPU target face detectors in Figure 7. As we already mentioned above, our method achieves the highest mAP among all the CPU target face detectors. In addition, we found that it also shows comparable detection performance compared with GPU target detectors. EResFD shows similar detection accuracy with S3FD and SSH, but it is about 19x faster.

Furthermore, we compare the proposed method with RetinaFace, one standard lightweight face detector in this field. Since RetinaFace detects face and facial landmarks at the same time, we covered landmark detection as well. We measured the face detection performance on WIDER FACE and landmark detection performance on AFLW. Figure 8 shows comparison on face and landmark detection. Our EResFD achieves higher mAP for face detection and lower NME for landmark detection, while reducing latency.

5. Conclusion

This paper rediscovers the efficiency of standard convolution-based architecture for lightweight face detection. The extensive experimental results showed that the standard convolutional block achieves superior performance compared to depthwise separable convolution, contrary to the common trend in this field. Based on the observation, we propose an efficient architecture EResNet, which includes a modified stem layer and channel dimension preserving strategy. Also, we propose SepFPN and CCPM for the feature enhancement, which boosts the detection performance without sacrificing latency and parameter size. Summing up the observations and architecture suggestions for face detection, we establish a new state-of-the-art real-time CPU face detector, EResFD, achieving the SOTA face detection performance among the lightweight detectors.

References

[1] Valentin Bazarevsky, Yury Kartynnik, Andrey Vakunov, Karthik Raveendran, and Matthias Grundmann. Blazeface: Sub-millisecond neural face detection on mobile gpus. *arXiv preprint arXiv:1907.05047*, 2019. 1, 2

[2] Irwan Bello, William Fedus, Xianzhi Du, Ekin Dogus Cubuk, Aravind Srinivas, Tsung-Yi Lin, Jonathon Shlens, and Barret Zoph. Revisiting resnets: Improved training and scaling strategies. *Advances in Neural Information Processing Systems*, 34, 2021. 4

[3] Zhaowei Cai and Nuno Vasconcelos. Cascade r-cnn: Delving into high quality object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6154–6162, 2018. 1

[4] Leilei Cao, Yao Xiao, and Lin Xu. Emface: Detecting hard faces by exploring receptive field pyraminds. *arXiv preprint arXiv:2105.10104*, 2021. 2

[5] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *Proceedings of the European conference on computer vision (ECCV)*, pages 801–818, 2018. 2

[6] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li 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

[7] Jiankang Deng, Jia Guo, Yuxiang Zhou, Jinke Yu, Irene Kotsia, and Stefanos Zafeiriou. Retinaface: Single-stage dense face localisation in the wild. *arXiv preprint arXiv:1905.00641*, 2019. 1, 2, 6, 7, 8

[8] Spyros Gidaris and Nikos Komodakis. Object detection via a multi-region and semantic segmentation-aware cnn model. In *Proceedings of the IEEE international conference on computer vision*, pages 1134–1142, 2015. 6

[9] Jia Guo, Jiankang Deng, Alexandros Lattas, and Stefanos Zafeiriou. Sample and computation redistribution for efficient face detection. In *International Conference on Learning Representations*, 2022. 1, 2

[10] Dongyoon Han, YoungJoon Yoo, Beomyoung Kim, and Byeongho Heo. Learning features with parameter-free layers. *arXiv preprint arXiv:2202.02777*, 2022. 4

[11] Kai Han, Yunhe Wang, Qi Tian, Jianyuan Guo, Chunjing Xu, and Chang Xu. Ghostnet: More features from cheap operations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1580–1589, 2020. 2

[12] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017. 1

[13] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 1, 2, 3, 5

[14] Toan Minh Hoang, Gi Pyo Nam, Junghyun Cho, and Ig-Jae Kim. Deface: Deep efficient face network for small scale variations. *IEEE Access*, 8:142423–142433, 2020. 2

[15] Andrew Howard, Mark Sandler, Grace Chu, Liang-Chieh Chen, Bo Chen, Mingxing Tan, Weijun Wang, Yukun Zhu, Ruoming Pang, Vijay Vasudevan, et al. Searching for mobilenetv3. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1314–1324, 2019. 2, 3

[16] Andrew G Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*, 2017. 1, 2

[17] Andrew G Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*, 2017. 1, 2, 3, 8

[18] Haibo Jin, Shifeng Zhang, Xiangyu Zhu, Yinhang Tang, Zhen Lei, and Stan Z Li. Learning lightweight face detector with knowledge distillation. In *2019 International Conference on Biometrics (ICB)*, pages 1–7. IEEE, 2019. 3, 8

[19] Martin Koestinger, Paul Wohlhart, Peter M Roth, and Horst Bischof. Annotated facial landmarks in the wild: A large-scale, real-world database for facial landmark localization. In *2011 IEEE international conference on computer vision workshops (ICCV workshops)*, pages 2144–2151. IEEE, 2011. 8

[20] Jian Li, Yabiao Wang, Changan Wang, Ying Tai, Jianjun Qian, Jian Yang, Chengjie Wang, Jilin Li, and Feiyue Huang. Dsfd: dual shot face detector. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5060–5069, 2019. 2, 7, 8

[21] Jian Li, Yabiao Wang, Changan Wang, Ying Tai, Jianjun Qian, Jian Yang, Chengjie Wang, Jilin Li, and Feiyue Huang. Dsfd: Dual shot face detector. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019. 2

[22] Zhihang Li, Xu Tang, Junyu Han, Jingtuo Liu, and Ran He. Pyramidbox++: high performance detector for finding tiny face. *arXiv preprint arXiv:1904.00386*, 2019. 2, 6, 7

[23] Ji Lin, Wei-Ming Chen, Han Cai, Chuang Gan, and Song Han. Mcunetv2: Memory-efficient patch-based inference for tiny deep learning. *arXiv preprint arXiv:2110.15352*, 2021. 2

[24] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2117–2125, 2017. 1, 6, 7

[25] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 2980–2988, 2017. 1, 2

[26] Shu Liu, Lu Qi, Haifang Qin, Jianping Shi, and Jiaya Jia. Path aggregation network for instance segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8759–8768, 2018. 6, 7

[27] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C

Berg. Ssd: Single shot multibox detector. In *European conference on computer vision*, pages 21–37. Springer, 2016. 1, 2

[28] Yang Liu and Xu Tang. Bfbox: Searching face-appropriate backbone and feature pyramid network for face detector. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13568–13577, 2020. 2

[29] Yang Liu, Xu Tang, Junyu Han, Jingtuo Liu, Dinger Rui, and Xiang Wu. Hambox: Delving into mining high-quality anchors on face detection. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 13043–13051. IEEE, 2020. 2

[30] Mahyar Najibi, Pouya Samangouei, Rama Chellappa, and Larry S Davis. Ssh: Single stage headless face detector. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4875–4884, 2017. 2, 6, 7, 8

[31] Mahyar Najibi, Bharat Singh, and Larry S Davis. Fa-rpn: Floating region proposals for face detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7723–7732, 2019. 2

[32] Shuaihui Qi, Xiaofeng Song, Zhiyuan Li, and Tao Xie. Fast and efficient face detector based on large kernel attention for cpu device. *Journal of Real-Time Image Processing*, 20(4):1–11, 2023. 2

[33] Leonardo Ramos and Bernardo Morales. Swiftface: Real-time face detection. *arXiv preprint arXiv:2009.13743*, 2020. 2

[34] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali 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. 1

[35] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems*, 28:91–99, 2015. 1, 2

[36] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4510–4520, 2018. 2, 3, 4

[37] Guanglu Song, Yu Liu, Yuhang Zang, Xiaogang Wang, Biao Leng, and Qingsheng Yuan. Kpnet: Towards minimal face detector. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 12015–12022, 2020. 2

[38] Christian Szegedy, Sergey Ioffe, Vincent Vanhoucke, and Alexander A Alemi. Inception-v4, inception-resnet and the impact of residual connections on learning. In *Thirty-first AAAI conference on artificial intelligence*, 2017. 5

[39] Mingxing Tan and Quoc Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *International Conference on Machine Learning*, pages 6105–6114. PMLR, 2019. 2, 5

[40] Mingxing Tan and Quoc V Le. Efficientnetv2: Smaller models and faster training. *arXiv preprint arXiv:2104.00298*, 2021. 2

[41] Mingxing Tan, Ruoming Pang, and Quoc V Le. Efficientdet: Scalable and efficient object detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10781–10790, 2020. 1

[42] Mingxing Tan, Ruoming Pang, and Quoc V Le. Efficientdet: Scalable and efficient object detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10781–10790, 2020. 6, 7

[43] Xu Tang, Daniel K Du, Zeqiang He, and Jingtuo Liu. Pyramidbox: A context-assisted single shot face detector. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 797–813, 2018. 2, 6, 7, 8

[44] Noranart Vesdapunt and Baoyuan Wang. Crface: Confidence ranker for model-agnostic face detection refinement. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1674–1684, 2021. 1

[45] Bichen Wu, Alvin Wan, Xiangyu Yue, Peter Jin, Sicheng Zhao, Noah Golmant, Amir Gholaminejad, Joseph Gonzalez, and Kurt Keutzer. Shift: A zero flop, zero parameter alternative to spatial convolutions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 9127–9135, 2018. 2

[46] Wei Wu, Hanyang Peng, and Shiqi Yu. Yunet: A tiny millisecond-level face detector. *Machine Intelligence Research*, pages 1–10, 2023. 2

[47] Shuo Yang, Ping Luo, Chen-Change Loy, and Xiaoou Tang. Wider face: A face detection benchmark. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5525–5533, 2016. 2, 6

[48] YoungJoon Yoo, Dongyoon Han, and Sangdoo Yun. Extd: Extremely tiny face detector via iterative filter reuse. *arXiv preprint arXiv:1906.06579*, 2019. 2

[49] Matthew D Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. In *European conference on computer vision*, pages 818–833. Springer, 2014. 5

[50] Bin Zhang, Jian Li, Yabiao Wang, Ying Tai, Chengjie Wang, Jilin Li, Feiyue Huang, Yili Xia, Wenjiang Pei, and Rongrong Ji. Asfd: Automatic and scalable face detector. *arXiv preprint arXiv:2003.11228*, 2020. 2

[51] Faen Zhang, Xinyu Fan, Guo Ai, Jianfei Song, Yongqiang Qin, and Jiahong Wu. Accurate face detection for high performance. *arXiv preprint arXiv:1905.01585*, 2019. 2

[52] Kaipeng Zhang, Zhanpeng Zhang, Zhifeng Li, and Yu Qiao. Joint face detection and alignment using multitask cascaded convolutional networks. *IEEE Signal Processing Letters*, 23(10):1499–1503, 2016. 2

[53] Kaipeng Zhang, Zhanpeng Zhang, Zhifeng Li, and Yu Qiao. Joint face detection and alignment using multitask cascaded convolutional networks. *IEEE Signal Processing Letters*, 23(10):1499–1503, 2016. 8

[54] Kaipeng Zhang, Zhanpeng Zhang, Hao Wang, Zhifeng Li, Yu Qiao, and Wei Liu. Detecting faces using inside cascaded contextual cnn. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3171–3179, 2017. 8

[55] Shifeng Zhang, Cheng Chi, Zhen Lei, and Stan Z Li. Refineface: Refinement neural network for high performance face detection. *arXiv preprint arXiv:1909.04376*, 2019. 6

[56] Shifeng Zhang, Xiaobo Wang, Zhen Lei, and Stan Z Li. Faceboxes: A cpu real-time and accurate unconstrained face detector. *Neurocomputing*, 364:297–309, 2019. 2, 8

[57] Shifeng Zhang, Xiaobo Wang, Zhen Lei, and Stan Z Li. Faceboxes: A cpu real-time and accurate unconstrained face detector. *Neurocomputing*, 364:297–309, 2019. 3, 5, 6, 8

[58] Shifeng Zhang, Xiangyu Zhu, Zhen Lei, Hailin Shi, Xiaobo Wang, and Stan Z Li. S3fd: Single shot scale-invariant face detector. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 192–201, 2017. 2, 6, 8

[59] Xiangyu Zhang, Xinyu Zhou, Mengxiao Lin, and Jian Sun. Shufflenet: An extremely efficient convolutional neural network for mobile devices. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6848–6856, 2018. 1

[60] Jiashu Zhu, Dong Li, Tiantian Han, Lu Tian, and Yi Shan. Progressface: Scale-aware progressive learning for face detection. In *European Conference on Computer Vision*, pages 344–360. Springer, 2020. 2

[61] Yanjia Zhu, Hongxiang Cai, Shuhan Zhang, Chenhao Wang, and Yichao Xiong. Tinaface: Strong but simple baseline for face detection. *arXiv preprint arXiv:2011.13183*, 2020. 2