

# EXTD: Extremely Tiny Face Detector via Iterative Filter Reuse

YoungJoon Yoo\*

youngjoon.yoo@navercorp.com

Dongyoon Han\*

dongyoon.han@navercorp.com

Sangdoon Yun\*

sangdoon.yun@navercorp.com

## Abstract

In this paper, we propose a new multi-scale face detector having an extremely tiny number of parameters (EXTD), less than 0.1 million, as well as achieving comparable performance to deep heavy detectors. While existing multi-scale face detectors extract feature maps with different scales from a single backbone network, our method generates the feature maps by iteratively reusing a shared lightweight and shallow backbone network. This iterative sharing of the backbone network significantly reduces the number of parameters, and also provides the abstract image semantics captured from the higher stage of the network layers to the lower-level feature map. The proposed idea is employed by various model architectures and evaluated by extensive experiments. From the experiments from WIDER FACE dataset, we show that the proposed face detector can handle faces with various scale and conditions, and achieved comparable performance to the more massive face detectors that few hundreds and tens times heavier in model size and floating point operations.

## 1. Introduction

Detecting faces in an image is considered to be one of the most practical tasks in computer vision applications, and many studies [46, 30] are proposed from the beginning of the computer vision research. After the advent of deep neural networks, many face detection algorithms [53, 60, 45, 50] applying the deep network have reported significant performance improvement to the conventional face detectors.

The state-of-the-art (SOTA) face detectors [60, 45, 50] for in-the-wild images employ the framework of the recent object detectors [7, 38, 36, 37, 28, 4, 26]. These methods can even handle a various scale of faces with difficult conditions such as distortion, rotation, and occlusion. Among them, the face detectors [60, 32, 54, 44, 3, 58] using multiple feature-maps from different layer locations, which mainly stem from [28, 26, 27], are dominantly used since

\*Clova AI Research, NAVER Corp. We follow the alphabetical order except the first author.

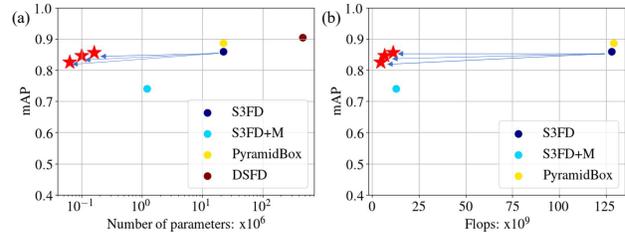


Figure 1. Illustration of the mean average precision (mAP) regarding the parameter size (a) and Flops (b) evaluated on WIDER FACE dataset. Our method (star) shows comparable mAP to S3FD [60] with a significantly smaller model. Red stars denote the proposed models with various sizes. ‘S3FD+M’ denotes the S3FD variation using MobileFaceNet [2] as a backbone network instead of VGG-16 [42]. Best viewed in wide and colored vision.

these methods can handle the faces with various scale in a single forward path.

While these methods achieved impressive detection performance, they commonly share two problems. One is their large number of parameters. Since they use a large classification network such as VGG-16 [42], ResNet [11]-50 or 101, and DenseNet-169 [14], the number of total parameters exceed 20 million, over 80Mb supposing 32-bit floating point for each parameter. Furthermore, the amount of floating point operations (FLOPs) also exceeds 100G, and these make it nearly impossible to use the face detectors in CPU or mobile environment, where the most face applications run in. The second problem, from the architecture perspective, is the limited capacity of the low-level feature map in capturing object semantics. The most single-shot detector (SSD) [28] variant object and face detectors struggle the problem because the low-level feature map passes shallow convolutional layers. To alleviate the problem, the variants of Feature pyramid network (FPN) architecture such as [28, 26, 41, 43] are used but requires additional parameters and memories for re-expanding the feature map.

In this paper, we propose a new multi-scale face detector with extremely tiny size (EXTD) resolving the two mentioned problems. The main discovery is that we can share the network in generating each feature-map, as shown in Figure 2. As in the figure, we design a backbone network such that reduces the size of the feature map by half, and

we can get the other feature maps with recurrently passing the network. The sharing can significantly reduce the number of parameters, and this enables our model to use more layers to generate the low-level feature maps used for detecting small faces. Also, the proposed iterative architecture makes the network to observe the features from various scale of faces and from various layer locations, and hence offer abundant semantic information to the network, without adding additional parameters.

Our baseline framework follows FPN-like structures, but can also be applied to SSD-like architecture. For SSD based architecture, we adopt the setting from [60]. For the FPN architectures, we refer an up-sampling strategy from [23]. The backbone network is designed to have less than 0.1 million parameters with employing inverted residual blocks proposed in MobileNet-V2 [40]. We note that our model does not require any extra layer commonly defined as in [28, 25], and is trained from scratch. We evaluated the proposed detector and its variants on WIDER FACE [53] dataset, the most widely used and similar to the in-the-wild situation.

The main contributions of this work can be summarized as follows: (1) We propose an iterative network sharing model for multi-stage face detection which can significantly reduce the parameter size, as well as provide abundant object semantic information to the lower stage feature maps. (2) We design a lightweight backbone network for the proposed iterative feature map generation with 0.1M number of parameters, which less than 400Kb, and achieved comparable mAP to the heavy face detection methods. (3) We employ the iterative network sharing idea to the widely used detection architectures, FPN and SSD, and show the effectiveness of the proposed scheme.

## 2. Related Works

**Face detectors:** Face detection has been an important research topic since an initial stage of computer vision researches. Viola *et al.* [46] proposed a face detection method using Haar features and Adaboost with decent performance, and several different approaches [22, 31, 51, 30] followed. After deep learning has become dominant, many face detection methods applying the techniques have been published. In the early stages, various attempts were tried to employ the deep architecture to face detection, such as cascade architecture [53, 57], and occlusion handling [52].

Recent face detectors has been designed based on the architecture of generic object detectors including Faster-RCNN [38], R-FCN [4], SSD [28], FPN [26], and RetinaNet [27]. Face RCNN and its variants [47, 15, 56] apply Faster-RCNN, and [50, 62] use R-FCN for detecting faces with meaningful performance improvements.

Also, to cope with the various scale of faces with single forward path, object detectors such as SSD, RetinaNet, and FPN are dominantly adopted since they use features from

multiple layer locations for detecting objects with various scale in a single forward path. S3FD [60] achieved promising performance by applying SSD with introducing multiple strategies to handle the small size of faces. FAN [48] uses RetinaNet by applying anchor level attention to detect the occluded faces. After S3FD, many improved versions [44, 54, 61, 21, 58] are introduced and achieved performance gain from the previous methods. FPN based face detection methods [3, 59, 45] achieved SOTA performance by enhancing the expression capacity of the lower-level feature map used for detecting small faces.

The mentioned SOTA methods commonly use classification network such as VGG-16 [42], ResNet-50 or 101 [11], and DenseNet-169 [14] as a backbone of the model. These classification networks have a large number of parameters exceeding 20 million, and the model size is over 80Mb supposing 32-bit floating point for each parameter. Some cascade methods such as [55] report decent mAP with the smaller mount of model size, about 3.8Mb. However, the size is still burdensome to the devices like mobile, because users generally want their applications not to exceed few ten's of Mb. Also, the face detector should mostly be much smaller than the total size of the application because a face detector is usually an end-level function of the application.

Here, we propose a new scheme of iteratively sharing the backbone network, which can be applicable to both SSD and FPN based architectures. The method achieves comparable accuracy to the original models, and the overall model size is extremely smaller as well.

**Lightweight generic object detectors:** Recently, for detecting general objects in condition with a limited resource such as mobile devices, various single-stage, and two-stage lightweight detectors were proposed. For the single-stage detectors, MobileNet-SSD [13], MobileNetV2-SSDLite [40], Pelee [49] and Tiny-DSOD [23] were proposed. For two-stage detectors, Light-Head R-CNN [24] and ThunderNet [35] were proposed. The mentioned methods achieved meaningful accuracy and size trade-off, but we aim to develop a detector which has a much smaller number of parameters with introducing a new paradigm, iterative use of the backbone network.

**Recurrent convolutional network:** The idea of recurrently using convolutional layers has been applied to various computer vision applications. Sharesnet [1] and Iamnn [19] applied recurrent residual network into classification task. Guo *et al.* [9] reduce the parameters by sharing depth-wise convolutional filters in learning multiple visual domain data. The iterative sharing is also applied to dynamic routing [16], fast inference of video [33], feature transfer [29], super-resolution [18], and recently in segmentation [20]. In this paper, we introduce a method applying the concept of iterative convolutional layer sharing in the face detection task, which is the first to the best of our knowledge.

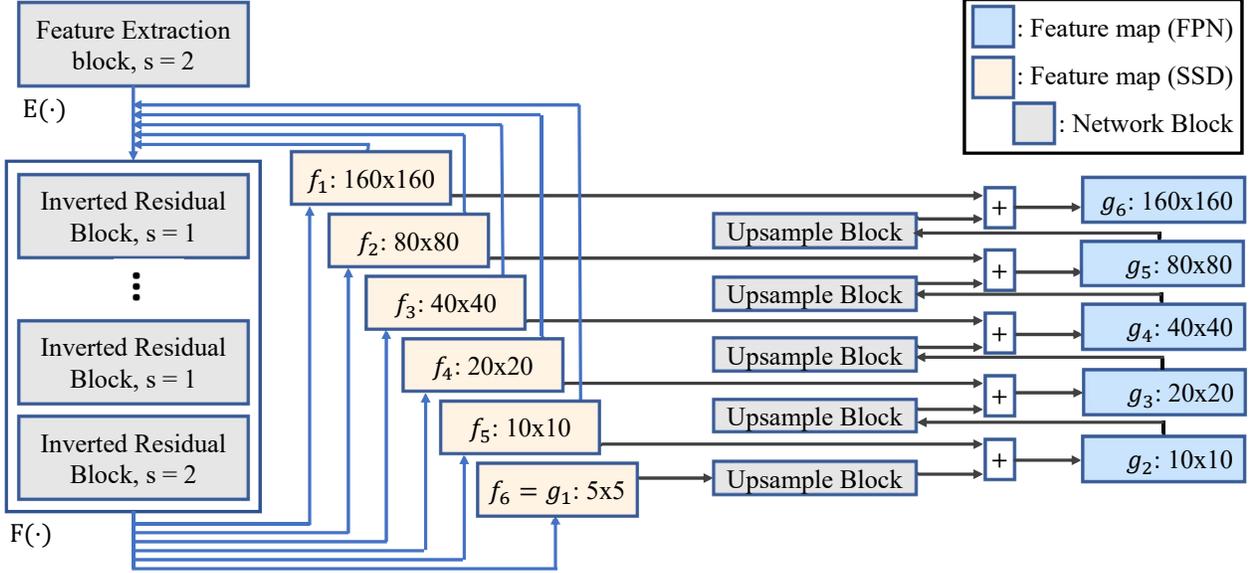


Figure 2. The overall framework of the proposed method. The structure recurrently generates the feature maps  $f_i$  (SSD version), and we upsample the feature maps with skip connection to generate the feature maps  $g_i$  (FPN version). The classification and regression heads can be attached to either  $f_i$  and  $g_i$ .

### 3. EXTENDED

In this section, we introduce the main components of the proposed work including iterative feature map generation, the architectures of the proposed face detection models, backbone networks, and classification and regression head design. Also, implementation details for designing and training the models will be introduced.

#### 3.1. Iterative Feature Map Generation

Figure 2 shows the overall framework of the proposed method with two variations, SSD-like, and FPN-like frameworks. In the proposed method, we get multiple feature maps with different resolutions by recurrently passing the backbone network. Let assume that  $F(\cdot)$  and  $E(\cdot)$  each denotes the backbone network and the first Conv layer with stride two. Then, the iterative process is defined as follows:

$$\begin{aligned} f_i &= F(f_{i-1}), i = 1, \dots, N, \\ f_0 &= E(x). \end{aligned} \quad (1)$$

Here, the set  $\{f_1, \dots, f_N\}$  denote the set of feature maps, and  $x$  is the image. In FPN version, we upsample each feature map and connect the previous feature maps via skip-connection [11, 39]. The upsampling step  $U_i(\cdot)$  is conducted with bilinear upsampling followed by an upsampling block composed of separable convolution and point-wise convolution, inspired by [23]. The resultant set of the feature map  $G = \{g_1, \dots, g_N\}$  is obtained as,

$$\begin{aligned} g_{i+1} &= U_i(g_i) + f_{N-i}, i = 1, \dots, N - 1, \\ g_1 &= f_N. \end{aligned} \quad (2)$$

For the SSD-like architecture, which is the first variant, we extract feature maps  $f_i$  and connect the classification and regression head to the feature maps. In FPN-like architecture, the feature maps  $g_i$  from equation (2) are used. The classification and regression heads are designed by a 3x3 convolutional network and hence, both models are designed as a fully convolutional network. This enables the models to deal with various size of images. The detailed implementation of the heads is introduced in below sections.

For all the cases, we set the image  $x$  to have 640x640 resolution in training phase and use  $N = 6$  number of feature maps. Hence, we get 160x160, 80x80, 40x40, 20x20, 10x10 and 5x5 resolution feature maps. In each location of the feature map, prior anchor candidates for the face is defined, following the same setting as S3FD [60].

One notable property of this architecture is that this method provides more abundant semantic information in lower-level feature maps compared to the face detectors adopting SSD architecture. While the existing methods commonly report the problem that the lower-level feature maps only contain limited semantic information due to their limited length of depth, our iterative architecture repeatedly shows intermediate level features and the various scale of faces to the network. We conjecture that the different features have similar semantics because the target objects in our case are faces, and the faces share homogeneous shapes

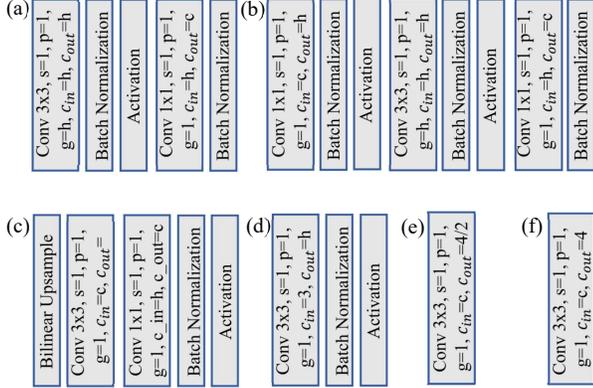


Figure 3. Detailed configuration of the components. The terms  $s$ ,  $p$ ,  $g$ ,  $c_{in}$ , and  $c_{out}$  denote the stride, padding, group, input channel width, and output channel width. Figures (a) and (b) each shows the initial and remaining inverted residual blocks. In (c) and (d), upsampling block and the Feature extraction block are presented. Figures (e) and (f) each denotes the classification and regression head. For the activation function, PReLU or Leaky-ReLU are used for (a) and (b), and ReLU is used for the others.

regardless of their scale dissimilar to general objects. In Section 4, we show that the proposed method clearly enhances the detection accuracy for small size faces, and this can be more improved by taking the FPN architecture.

### 3.2. Model Component Description

In the proposed model, a lightweight backbone network reducing the feature map resolution by half is used. The network is composed of inverted residual blocks followed by one 3x3 convolutional (Conv) filter with stride 2, based on [40, 2]. The inverted residual block is composed of a set of point-wise Conv, separable Conv, and point-wise Conv. In each block, the channel width is expanded in the first point-wise Conv and then, squeezed by the last point-wise Conv filter. The default setting of the network depth is set to 6 or 8, and the output channel width is set to 32,48 or 64, which do not largely exceed overall 0.1 million parameters. Different from MobileNet-V2 [40], PReLU [10] (or leaky-ReLU) is applied and shown to be more successful than ReLU in training the proposed recurrent architecture. This phenomenon will be further discussed in Section 4.

Other than the inverted residual block, the proposed architecture also includes feature extraction block, upsampling blocks, and classification and regression heads. The detailed description of the components is introduced in Figure 3. The figures in (a) and (b) each shows the inverted residual block architecture. Residual skip-connection is applied when the input and output channel width are equivalent, and at the same time, the stride is set to one. The upsampling block in (c) consists of bilinear upsample layer followed by depth-wise and point-wise Conv blocks. Fea-

ture extraction block (d) is defined by a 3x3 Conv network followed by batch normalization and the activation function. The classification (e) and regression (f) heads are also defined by a 3x3 Conv network. The implementation of the head is described in Section 3.3.

### 3.3. Classification and Regression Head Design

For detecting the faces using the generated feature maps, we use a classification head and a regression head for each feature map to classify whether each prior box contains a face, and to regress the prior box to the exact location. The classification and regression heads are both defined as single 3x3 Conv filters as shown in Figure 3. The classification head  $C_i$  has two-dimensional output channel  $c_i$  except  $C_1$  that having four-dimensional channels. For  $C_1$ , we apply Maxout [8] approach to select two of the four channels for alleviating the false positive rate of the small faces, as introduced in S3FD. The regression head  $R_i$  is defined to have output feature  $r_i$  to have four-dimensional channel, and each denotes width, height ratio, and center locations, adopting the dominantly used setting in RPN [38].

### 3.4. Training

The proposed backbone network and the classification and regression head are jointly trained by a multitask loss function from RPN composed of a classification loss  $l_c$  and a regression loss  $l_r$  as,

$$l(\{c_j, r_j\}) = \frac{\lambda}{N_{cls}} \sum_j l_c(c_j, c_j^*) + \frac{1}{N_{reg}} \sum_j c_j^* l_r(r_j, r_j^*) \quad (3)$$

Here,  $j$  is the index of the anchor boxes, and the label  $c_j^* \in \{0, 1\}$  and  $r_j^*$  is the ground truth of the anchor box. The label  $c_j^*$  is set to 1 when Jaccard overlap [6] between the anchor box and ground truth box is higher than a threshold  $t$ . The denominator  $N_{cls}$  denotes the total number of positive and negative samples. The regression loss is computed only for the positive sample and hence, the number  $N_{reg}$  is defined by  $N_{reg} = \sum_j c_j^*$ . The parameter  $\lambda$  is defined to balance the two losses because  $N_{cls}$  and  $N_{reg}$  are different from each other. The vector  $r_j^*$  denotes the ground truth box location and size for the face. The classification loss  $l_c$  and the regression loss  $l_r$  are defined as cross-entropy loss and smooth- $\ell_1$  loss, respectively.

The primary obstacle for the classification in the face detection task is a class imbalance problem between the face and the background, especially regarding the small faces. To alleviate the problem, we also adopt the strategies including online hard negative mining and scale compensation anchor matching introduced in S3FD. Using the hard negative mining technique, we balance the ratio of positive and negative samples  $N_{neg}/N_{pos}$  to 3 and the balancing parameter  $\lambda$  is set to 4. Also, from the scale compensation anchor matching strategy, we first pick the positive samples where the

Jaccard overlap is over 0.35, and then further pick the remaining samples in sorted order from the samples that their Jaccard overlap is larger than 0.1 if the number of positive samples is insufficient.

For Data augmentation, we follow the conventional augmentation setting from S3FD. The augmentation includes color distortions [12], random crop, horizontal flip, and vertical flip. The proposed method is implemented with PyTorch [34] and NAVER Smart Machine Learning (NSML) [17] system. Please refer Appendix A to see the detailed training and optimization settings for training the proposed network. Code will be available at <https://github.com/clovaai>.

## 4. Experiments

In this section, we quantitatively and qualitatively analyze the proposed method with various ablations. For the quantitative analysis, we compare the detection performance of the proposed method and the SOTA face detection algorithms. Qualitatively, we show that our method can successfully detect faces in various conditions.

### 4.1. Experimental Setting

**Datasets:** we tested the proposed method and ablations of the method with WIDER FACE [53] dataset, which is most recent and is similar to in-the-wild face detection situation. The images in the dataset are divided into Easy, Medium, and Hard cases which are roughly categorized by different scales: large, medium, and small, of faces. The Hard case includes all the images of the dataset, and the Easy and Medium cases both are the subsets of the Hard case. The dataset has total 32,203 images with 393,703 labeled faces and is split into training (40%), validation (20%) and testing (40%) set. We trained the detectors with the training set and evaluated them with validation and test sets.

**Comparison:** Since our method followed the training and implementation details such as anchor design, data augmentation, and feature-map resolution design equivalent to S3FD [60], which has become one of the baseline methods in face detection field, we mostly evaluated the performance by comparing the S3FD model and its SOTA variations [44, 21]. The other techniques based on the S3FD model such as Pyramid anchor [44], Feature enhancement module, Improved anchor matching, and Progressive anchor loss [21] would be able to be adapted to the proposed model without revising the proposed structure. Also, we used the MobileFaceNet [2], the face variant of the MobileNet-V2 [40], to the S3FD model instead of VGG-16 to see the effectiveness of the proposed method compared to the case of using the lightweight backbone network.

**Variations:** We applied the proposed recurrent scheme mainly into the FPN-based structure. For the model, we designed three variations which have a different number of

parameters, lighter one having 0.063M parameters with 32 channels for each feature maps, intermediate one having 0.1M parameters with 48 channels, and the heavier one with 64 channels and 0.16M parameters when designed as FPN. See Appendix B for the detailed configuration of the backbone networks for each case.

Also, we tested different activation functions: ReLU, PReLU, and Leaky-ReLU for each model. The negative slope of the Leaky-ReLU is set to 0.25, which is identical to the initial negative slope of the PReLU. In the following section, we will term each variation by a combination of abbreviations; *EXTD-model-channel-activation*. For example, the term *EXTD-FPN-32-PReLU* denotes the proposed model combined with FPN, with feature channel width 32 and with activation function PReLU.

As an ablation, we also applied the proposed recurrent backbone into SSD-like structure as well. The ablation was trained and tested with the same conditions to the FPN-based version and abbreviated as *SSD*. Same as FPN case, for example, the term *EXTD-SSD-32-PReLU* denotes the proposed model combined with SSD, with feature channel width 32 and with activation function PReLU.

### 4.2. Performance Analysis

In Table 1, we list the quantitative evaluation results of face detection in WIDER FACE dataset and the comparison to the SOTA face detectors. The table shows the mAP of the models on Easy, Medium, Hard cases for both validation and test sets of the dataset. Also, the table includes model information such as their backbone networks, number of parameters, and total number of adder arithmetics (Madds). In Figure 4, the precision recall curve for the proposed and the other methods are presented. Figure 5 shows the examples of the face detection results from images with various conditions. In Figure 6, we evaluate the latency of the models in terms of the resolution of images, which measured via a machine with CPU i7 core and NVIDIA TITAN-X. For a fair comparison, all the inference processes of the models are implemented by PyTorch 1.0.

**Comparison to the Existing Methods:** The results in Table 1 shows that some variations of the proposed method achieved comparable performance to the baseline model S3FD. Among lighter models and intermediate models, *EXTD-FPN-32-PReLU* and *EXTD-FPN-48-PReLU* each got a mAP score 3.4% and 1.2% lower than S3FD in WIDER Face hard validation set. When compared to S3FD trained scratch, *EXTD-FPN-64-PReLU* achieved even performances. For the heavier version, we found that our FPN variant achieved nearly the same accuracy, only 0.3% in WIDER FACE hard validation set and 0.8% in test set to S3FD in spite of the huge model size and memory usage gaps. It is meaningful in that the proposed detectors: lighter, intermediate, and heavier versions, are about 343, 220, and

Model	Backbone	# Params	# Madds (G)	WIDER FACE		
				Easy (mAP)	Medium (mAP)	Hard (mAP)
PyramidBox [44]*	VGG-16	57 M	129	0.961 / 0.956	0.950 / 0.946	0.887 / 0.887
DSFD [21]-ResNet101*	ResNet101	399 M	-	0.963	0.954	0.901
DSFD-ResNet152*	ResNet152	459 M	-	0.966 / 0.960	0.957 / 0.953	0.904 / 0.900
S3FD [60]*	VGG-16	22 M	128	<b>0.942 / 0.937</b>	<b>0.930 / 0.925</b>	<b>0.859 / 0.858</b>
S3FD - Scratch	VGG-16	22 M	128	0.931	0.920	0.846
S3FD + MobileFaceNet [2]	MobileFaceNet	1.2 M	12.7	0.881	0.859	0.741
EXTD-FPN-32-PReLU	-	0.063 M	4.52	0.896	0.885	0.825
EXTD-FPN-48-PReLU	-	0.10 M	6.67	0.913	0.904	0.847
<b>EXTD-FPN-64-PReLU</b>	-	0.16 M	11.2	<b>0.921 / 0.912</b>	<b>0.911 / 0.903</b>	<b>0.856 / 0.850</b>

Table 1. Quantitative comparison to recent state-of-the-art face detectors on WIDER FACE dataset. ‘\*’ denotes results reported in the original papers. For the proposed model with highest validation mAP, we list the mAPs from validation set and that from test set at the left-side and right-side of the slash in fifth to seventh columns. The other cases, mAPs from the validation set are listed.

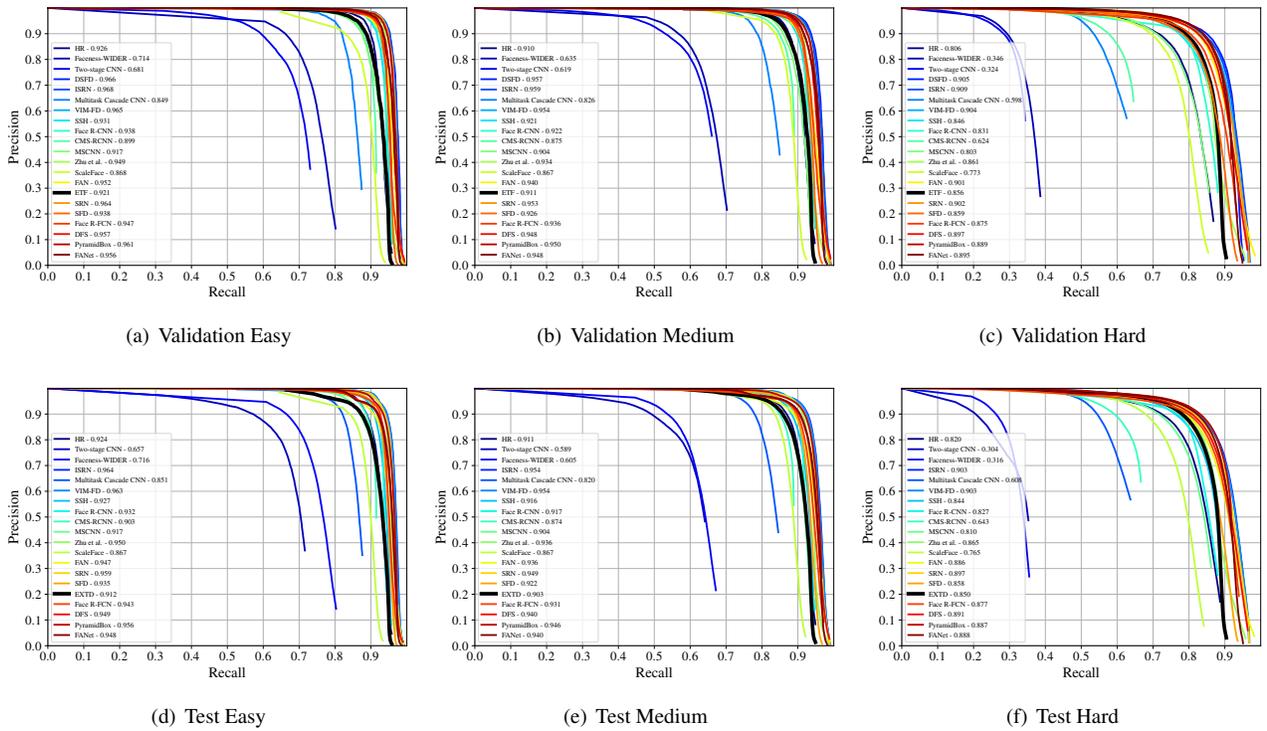


Figure 4. ROC curves on WIDER FACE dataset. Best viewed in wide vision. The curves from our method are illustrated by ‘black’.

138 times lighter in model size and are 28.3, 19.2, and 11 times lighter in Madds.

When compared to SOTA face detectors such as PyramidBox [44] and DSFD [21], our best model *EXTD-FPN-64-PReLU* achieved lower results. The margin between PyramidBox and the proposed model on WIDER FACE hard case was 3.4%. Considering that PyramidBox inherits from S3FD and our model follows the equivalent training and detection setting to S3FD, our model would have a possibility to further increase the detection performance by adding the schemes proposed in PyramidBox. The mAP gap to DSFD, which is tremendously heavier, is about 5.0%, but it would be safe to suggest that the proposed method offers

more decent trade-off in that DSFD uses about 2860 times more parameters than the proposed method. This is also meaningful result in that our method did not use any kind of pre-training of the backbone network using the other dataset such as ImageNet [5]. Figure 4 shows the ROC curves of the proposed *EXTD-FPN-64-PReLU* and the other methods. From the graphs, we can see that our method is included in the SOTA group of the detectors using heavy-weight pre-trained backbone networks.

When it comes to our SSD-based variations, they got lower mAP results than FPN-based variants. However, when compared with the S3FD version trained with MobileFaceNet backbone network, the proposed SSD variants



Model	# Params	# Madds (G)	WIDER FACE		
			Easy (mAP)	Medium (mAP)	Hard (mAP)
EXTD-SSD-32-ReLU	0.056 M	4.35	0.791 (-0.105)	0.770 (-0.115)	0.629 (-0.196)
EXTD-SSD-32-LReLU	0.056 M	4.35	0.851 (-0.045)	0.836 (-0.049)	0.736 (-0.089)
EXTD-SSD-32-PReLU	0.056 M	4.35	0.870 (-0.026)	0.855 (-0.030)	0.757 (-0.068)
EXTD-FPN-32-ReLU	0.063 M	4.52	0.741 (-0.155)	0.735 (-0.150)	0.642 (-0.182)
EXTD-FPN-32-LReLU	0.063 M	4.52	0.892 (-0.004)	0.884 (-0.001)	0.824 (-0.001)
<b>EXTD-FPN-32-PReLU</b>	0.063 M	4.52	<b>0.896</b>	<b>0.885</b>	<b>0.825</b>
EXTD-SSD-48-ReLU	0.086 M	6.63	0.868 (-0.045)	0.852 (-0.052)	0.742 (-0.105)
EXTD-SSD-48-LReLU	0.086 M	6.63	0.879 (-0.034)	0.860 (-0.044)	0.744 (-0.103)
EXTD-SSD-48-PReLU	0.086 M	6.63	0.897 (-0.016)	0.879 (-0.025)	0.774 (-0.073)
EXTD-FPN-48-ReLU	0.10 M	6.67	0.894 (-0.019)	0.885 (-0.019)	0.825 (-0.022)
EXTD-FPN-48-LReLU	0.10 M	6.67	0.911 (-0.002)	0.901 (-0.003)	0.846 (-0.001)
<b>EXTD-FPN-48-PReLU</b>	0.10 M	6.67	<b>0.913</b>	<b>0.904</b>	<b>0.847</b>
EXTD-SSD-64-ReLU	0.14 M	10.6	0.887 (-0.034)	0.867 (-0.044)	0.752 (-0.104)
EXTD-SSD-64-LReLU	0.14 M	10.6	0.896 (-0.025)	0.878 (-0.033)	0.769 (-0.087)
EXTD-SSD-64-PReLU	0.14 M	10.6	0.905 (-0.016)	0.888 (-0.023)	0.784 (-0.072)
EXTD-FPN-64-ReLU	0.16 M	11.2	0.910 (-0.011)	0.900 (-0.011)	0.844 (-0.012)
EXTD-FPN-64-LReLU	0.16 M	11.2	0.914 (-0.007)	0.906 (-0.005)	0.850 (-0.006)
<b>EXTD-FPN-64-PReLU</b>	0.16 M	11.2	<b>0.921</b>	<b>0.911</b>	<b>0.856</b>

Table 2. Variation study on WIDER FACE validation dataset. The models with boldface denotes the representative models for each block. The value in the parentheses shows the margin between the best model in the block (written in boldface).

**Effect of the Model Architecture:** From the table, we can find two common observations among the proposed variations. First, for all the different channel width, FPN based architecture achieved better detection performance compared to SSD based architecture, especially for detecting small faces. The idea of expanding the number of layers for reaching the largest sized feature-map, for detecting the smallest size of objects, is a common strategy for SSD variant methods. This approach assumes that typical SSD structure passes too small number of layers and hence, the resultant feature-map could not import much information useful for the detection task. In the face detection task, this assumption seems to be correct in that the FPN based models notably achieved superior detection performance on small faces compared to SSD based models for all the cases.

Second, for both SSD based and FPN based model, channel width was another key factor for performance enhancement. As the channel width increased by 32 to 64, we can see that the detection accuracy significantly enhanced for all the cases; Easy, Medium, and Hard. Considering that we used a smaller number of layers for 48 and 64 channel cases than the case with 32 channel, this shows that having enough size of channel width is critical for embedding sufficient information to the feature map for detecting faces.

**Effect of the Activation functions:** From the evaluation, we found that the choice of the activation function is another factor governing the detection performance of the proposed method. In all the cases including FPN based and SSD based structures, PReLU was the most effective choice when it comes to mAP, but the gap between Leaky-ReLU

was not that significant for the FPN variants. When tested with SSD based architecture, PReLU outperformed Leaky-ReLU with larger margin than those using FPN structure.

It is worth noting that ReLU occurred notable performance decreases especially when the channel width was small for both SSD and FPN cases. When the channel width was set to 32, mAP for all the three cases were lower than 10% to 20% compared to those using other activation functions. The decreases were alleviated as the channel width increased. When the channel width was 48, the gap was about 2.2%, and in the channel width 64 case, the margin was about 1.2%. From the results, we conjecture that the nature of ReLU that set all the negative values to zero occurs information loss in the proposed iterative process since it makes the feature map too sparse, and this information loss would be much critical when the channel width is small.

## 5. Conclusion

In this paper, we proposed a new face detector which significantly reduces the model sizes as well as maintaining the detection accuracy. By re-using backbone network layers recurrently, we reduced the vast amount of the network parameters and also obtained comparable performance to recent deep face detection methods using heavy backbone networks. We showed that our methods achieved very close mAP to the baseline S3FD only with hundreds time smaller parameters and with using tens time smaller Madd without using pre-training. We expect that our method can be further improved by applying recent techniques of the SOTA detectors which integrated to S3FD.

## Acknowledgement

We are grateful to Clova AI members with valuable discussions, and to Jung-Woo Ha for proofreading the manuscript.

## References

- [1] A. Boulch. Sharesnet: reducing residual network parameter number by sharing weights. *arXiv preprint arXiv:1702.08782*, 2017. [2](#)
- [2] S. Chen, Y. Liu, X. Gao, and Z. Han. Mobilefacenet: Efficient cnns for accurate real-time face verification on mobile devices. In *Chinese Conference on Biometric Recognition*, pages 428–438. Springer, 2018. [1](#), [4](#), [5](#), [6](#)
- [3] C. Chi, S. Zhang, J. Xing, Z. Lei, S. Z. Li, and X. Zou. Selective refinement network for high performance face detection. *arXiv preprint arXiv:1809.02693*, 2018. [1](#), [2](#)
- [4] J. Dai, Y. Li, K. He, and J. Sun. R-fcn: Object detection via region-based fully convolutional networks. In *Advances in neural information processing systems*, pages 379–387, 2016. [1](#), [2](#)
- [5] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR09*, 2009. [6](#)
- [6] D. Erhan, C. Szegedy, A. Toshev, and D. Anguelov. Scalable object detection using deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2147–2154, 2014. [4](#)
- [7] R. Girshick. Fast r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 1440–1448, 2015. [1](#)
- [8] I. J. Goodfellow, D. Warde-Farley, M. Mirza, A. Courville, and Y. Bengio. Maxout networks. *arXiv preprint arXiv:1302.4389*, 2013. [4](#)
- [9] Y. Guo, Y. Li, R. Feris, L. Wang, and T. Rosing. Depthwise convolution is all you need for learning multiple visual domains. *arXiv preprint arXiv:1902.00927*, 2019. [2](#)
- [10] K. He, X. Zhang, S. Ren, and J. Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision*, pages 1026–1034, 2015. [4](#), [11](#)
- [11] 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. [1](#), [2](#), [3](#)
- [12] A. G. Howard. Some improvements on deep convolutional neural network based image classification. *arXiv preprint arXiv:1312.5402*, 2013. [5](#)
- [13] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*, 2017. [2](#)
- [14] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4700–4708, 2017. [1](#), [2](#)
- [15] H. Jiang and E. Learned-Miller. Face detection with the faster r-cnn. In *2017 12th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2017)*, pages 650–657. IEEE, 2017. [2](#)
- [16] I. Kemaev, D. Polykovskiy, and D. Vetrov. Reset: Learning recurrent dynamic routing in resnet-like neural networks. *arXiv preprint arXiv:1811.04380*, 2018. [2](#)
- [17] H. Kim, M. Kim, D. Seo, J. Kim, H. Park, S. Park, H. Jo, K. Kim, Y. Yang, Y. Kim, et al. Nsm1: Meet the mlaas platform with a real-world case study. *arXiv preprint arXiv:1810.09957*, 2018. [5](#)
- [18] J. Kim, J. Kwon Lee, and K. Mu Lee. Deeply-recursive convolutional network for image super-resolution. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1637–1645, 2016. [2](#)
- [19] S. Leroux, P. Molchanov, P. Simoons, B. Dhoedt, T. Breuel, and J. Kautz. Iamnn: Iterative and adaptive mobile neural network for efficient image classification. *arXiv preprint arXiv:1804.10123*, 2018. [2](#)
- [20] H. Li, P. Xiong, H. Fan, and J. Sun. Dfanet: Deep feature aggregation for real-time semantic segmentation. *arXiv preprint arXiv:1904.02216*, 2019. [2](#)
- [21] J. Li, Y. Wang, C. Wang, Y. Tai, J. Qian, J. Yang, C. Wang, J. Li, and F. Huang. Dsfd: dual shot face detector. *arXiv preprint arXiv:1810.10220*, 2018. [2](#), [5](#), [6](#)
- [22] S. Z. Li, L. Zhu, Z. Zhang, A. Blake, H. Zhang, and H. Shum. Statistical learning of multi-view face detection. In *European Conference on Computer Vision*, pages 67–81. Springer, 2002. [2](#)
- [23] Y. Li, J. Li, W. Lin, and J. Li. Tiny-dsod: Lightweight object detection for resource-restricted usages. *arXiv preprint arXiv:1807.11013*, 2018. [2](#), [3](#), [11](#)
- [24] Z. Li, C. Peng, G. Yu, X. Zhang, Y. Deng, and J. Sun. Lightweight r-cnn: In defense of two-stage object detector. *arXiv preprint arXiv:1711.07264*, 2017. [2](#)
- [25] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie. Feature pyramid networks for object detection. *arXiv preprint arXiv:1612.03144*, 2016. [2](#)
- [26] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie. Feature pyramid networks for object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2117–2125, 2017. [1](#), [2](#)
- [27] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 2980–2988, 2017. [1](#), [2](#)
- [28] 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. [1](#), [2](#)
- [29] Y. Liu. *Efficient Recurrent Residual Networks Improved by Feature Transfer*. PhD thesis, Delft University of Technology, 2017. [2](#)
- [30] M. Mathias, R. Benenson, M. Pedersoli, and L. Van Gool. Face detection without bells and whistles. In *European conference on computer vision*, pages 720–735. Springer, 2014. [1](#), [2](#)

- [31] T. Mita, T. Kaneko, and O. Hori. Joint haar-like features for face detection. In *Tenth IEEE International Conference on Computer Vision (ICCV'05) Volume 1*, volume 2, pages 1619–1626. IEEE, 2005. 2
- [32] M. Najibi, P. Samangouei, R. Chellappa, and L. S. Davis. Ssh: Single stage headless face detector. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4875–4884, 2017. 1
- [33] B. Pan, W. Lin, X. Fang, C. Huang, B. Zhou, and C. Lu. Recurrent residual module for fast inference in videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1536–1545, 2018. 2
- [34] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer. Automatic differentiation in pytorch. 2017. 5
- [35] Z. Qin, Z. Li, Z. Zhang, Y. Bao, G. Yu, Y. Peng, and J. Sun. Thundernet: Towards real-time generic object detection. *arXiv preprint arXiv:1903.11752*, 2019. 2
- [36] J. Redmon and A. Farhadi. Yolo9000: better, faster, stronger. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7263–7271, 2017. 1
- [37] J. Redmon and A. Farhadi. Yolov3: An incremental improvement. *arXiv preprint arXiv:1804.02767*, 2018. 1
- [38] 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. 1, 2, 4
- [39] O. Ronneberger, P. Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015. 3
- [40] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4510–4520, 2018. 2, 4, 5, 11
- [41] Z. Shen, Z. Liu, J. Li, Y.-G. Jiang, Y. Chen, and X. Xue. Dsod: Learning deeply supervised object detectors from scratch. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1919–1927, 2017. 1
- [42] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 1, 2
- [43] S. Sun, J. Pang, J. Shi, S. Yi, and W. Ouyang. Fishnet: A versatile backbone for image, region, and pixel level prediction. In *Advances in Neural Information Processing Systems*, pages 754–764, 2018. 1
- [44] 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. 1, 2, 5, 6
- [45] W. Tian, Z. Wang, H. Shen, W. Deng, B. Chen, and X. Zhang. Learning better features for face detection with feature fusion and segmentation supervision. *arXiv preprint arXiv:1811.08557*, 2018. 1, 2
- [46] P. Viola, M. Jones, et al. Rapid object detection using a boosted cascade of simple features. 1, 2
- [47] H. Wang, Z. Li, X. Ji, and Y. Wang. Face r-cnn. *arXiv preprint arXiv:1706.01061*, 2017. 2
- [48] J. Wang, Y. Yuan, and G. Yu. Face attention network: an effective face detector for the occluded faces. *arXiv preprint arXiv:1711.07246*, 2017. 2
- [49] R. J. Wang, X. Li, and C. X. Ling. Pelee: A real-time object detection system on mobile devices. In *Advances in Neural Information Processing Systems*, pages 1963–1972, 2018. 2
- [50] Y. Wang, X. Ji, Z. Zhou, H. Wang, and Z. Li. Detecting faces using region-based fully convolutional networks. *arXiv preprint arXiv:1709.05256*, 2017. 1, 2
- [51] B. Yang, J. Yan, Z. Lei, and S. Z. Li. Aggregate channel features for multi-view face detection. In *IEEE international joint conference on biometrics*, pages 1–8. IEEE, 2014. 2
- [52] 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. 2
- [53] 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. 1, 2, 5
- [54] S. Yang, Y. Xiong, C. C. Loy, and X. Tang. Face detection through scale-friendly deep convolutional networks. *arXiv preprint arXiv:1706.02863*, 2017. 1, 2
- [55] B. Yu and D. Tao. Anchor cascade for efficient face detection. *IEEE Transactions on Image Processing*, 28(5):2490–2501, 2019. 2
- [56] C. Zhang, X. Xu, and D. Tu. Face detection using improved faster rcnn. *arXiv preprint arXiv:1802.02142*, 2018. 2
- [57] 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. 2
- [58] S. Zhang, L. Wen, X. Bian, Z. Lei, and S. Z. Li. Single-shot refinement neural network for object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4203–4212, 2018. 1, 2
- [59] S. Zhang, R. Zhu, X. Wang, H. Shi, T. Fu, S. Wang, and T. Mei. Improved selective refinement network for face detection. *arXiv preprint arXiv:1901.06651*, 2019. 2
- [60] 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. 1, 2, 3, 5, 6
- [61] C. Zhu, R. Tao, K. Luu, and M. Savvides. Seeing small faces from robust anchor’s perspective. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5127–5136, 2018. 2
- [62] C. Zhu, Y. Zheng, K. Luu, and M. Savvides. Cms-rcnn: contextual multi-scale region-based cnn for unconstrained face detection. *arXiv preprint arXiv:1606.05413*, 2016. 2

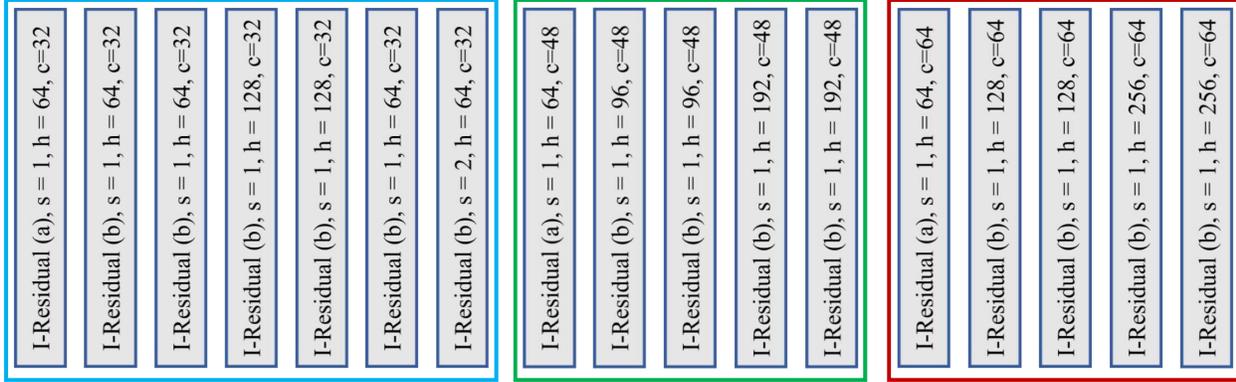


Figure 7. Backbone architectures for the recursive feature generation.

Description	1	2	3	4	5	6	7	8	9	10	11	12	13	14
I-Residual type	(a)	(b)												
Output channel width	64	64	64	64	64	64	128	128	128	128	128	128	128	128
Hidden channel width	64	128	128	128	128	128	256	256	256	256	256	256	256	512
Stride	2	1	1	1	1	2	2	1	1	1	1	1	1	2

Table 3. Structure of MobileFaceNet backbone attached in S3FD. Three extra layers are attached to further reduce the feature map size.

## Appendix A. Implementation detail

For training the proposed architecture, a stochastic gradient descent optimizer (SGD) with learning rate  $1e^{-3}$ , with 0.9 momentum, 0.0005 weight decay, and batch size 16 is used. The training is conducted from scratch, and the network weights were initialized with He-method [10]. The maximum iteration number is basically set to 240K, and we drop the learning rate to  $1e^{-4}$  and  $1e^{-5}$  at 120K and 180K iterations. Also, we test the architecture with twice larger iterations 480K as well. In this case, the learning rate is dropped at 240K and 360K iterations. Similar to the other networks using depth-wise separable networks [40, 23], further performance improvements were observed when training the network with larger iteration.

## Appendix B. Detailed Architecture Information

Figure 7 shows the detailed structures of the backbone network for the variation having channel sizes 32, 48, and 64. The layers in ‘blue’, ‘green’, and ‘red’ boxes in the figure each denotes the version of the proposed detectors having channel width to 32, 48, and 64. Each model has parameter size 0.063M, 0.10M, and 0.16M respectively, when designed as FPN structure. The term ‘I-Residual’ denotes the inverted residual block (a) and (b), where the configuration of the block is introduced in Figure 3 of the paper. The heavier versions which have 0.10M, and 0.16M model parameters are designed to have less number of parameters to reduce the parameter when compared to the lightest version. The results in the paper show that the width of the channels for each layer is more critical than the depth of the layers for the detection performance in the proposed model.

## Appendix C. Implementation of S3FD with MobileFaceNet Backbone

In the paper, we implemented the S3FD variation where the backbone network was set to MobileFaceNet instead of VGG-16. The backbone network consists of 14 inverted residual blocks followed by 3x3 convolutional filter which has output channel width 64 and stride two. The lowest-level inverted residual block is defined as in I-Residual (a), and the others are defined as I-Residual (b). The detailed setting of each blocks are described in Table 3. We added a classification and regression head at the bottom of layers 6, 7, and 14. After layer 14, three extra layers defined by 3x3 convolutional filter with output channel width 128 are attached. This extra layer setting is equivalent to original S3FD, and the resolutions of the feature maps are [64, 128, 128, 128, 128, 128] with total parameter number 1.2 million. The MobileFaceNet backbone itself is a reduced version of MobileNet-V2, and we only used the part of the MobileFaceNet layers. However, we can still see that the backbone network requires a large number of parameters which makes challenging to be embedded in smaller devices.