BFBox: Searching Face-appropriate Backbone and Feature Pyramid Network for Robust Face Detector

Yang Liu†
North China Electric Power University
Beijing
gxly1314@gmail.com

Xu Tang
Baidu Inc.
Beijing
tangxu02@baidu.com

Abstract

Popular backbones, designed on the task of image classification, have demonstrated their considerable compatibility on the task of general object detection. However, the same phenomenon does not appear on the face detection. This is largely due to the average scale of ground-truth in the Wider Face dataset is far smaller than that of generic objects in the COCO one. To resolve this, the success of Neural Architecture Search (NAS) method inspires us to search face-appropriate backbone and feature pyramid network (FPN) architecture. Firstly, we design the search space for backbone and FPN by comparing performance of feature maps with different backbones and excellent FPN architectures on the face detection. Second, we propose a FPN-attention module to joint search the architecture of backbone and FPN. Finally, we conduct comprehensive experiments on popular benchmarks, including Wider Face, FDDB, AFW and PASCAL Face, display the superiority of our proposed method.

1. Introduction

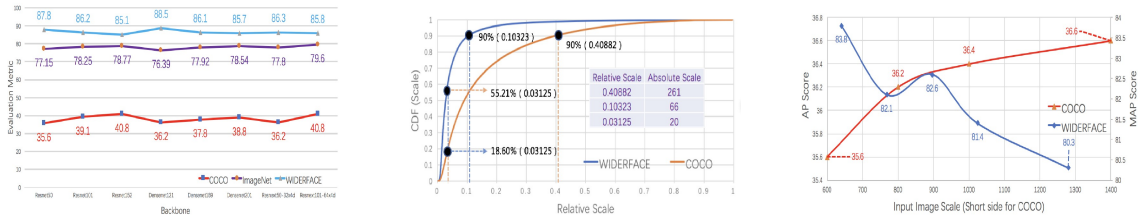
Face detection is a fundamental task in many facial tasks, such as face alignment [3], face recognition [6], face aging [1]. Traditional face detectors, adopting hand-craft features , have been replaced by deep convolutional neural networks with the ability of extracting discriminative face features. Recent state-of-the-art face detectors frequently use the backbone of Resnet50 [10] which achieves excellent performance on the image classification dataset [14]. The remaining network architectures (e.g. head module, predicting branch) are designed by the inspiration of general object detection [18].

Extensive experiments have shown that many backbones [10, 21, 12] have proved their superiority on both popular image classification (ImageNet) and general object detection dataset (COCO) simultaneously. However it is not the case on the face detection. Figure 1(a) shows the performance of various backbones on different datasets (Imagenet [14], COCO [19] and Wider Face [26]). As for two latter detection datasets, we adopt retinanet [18] architecture with diverse backbones. We distinctly discover that the classifier and the detector on general object detection have a consistent performance when adopting the same backbone. This represents that the backbones designed on the classification dataset could be easily apply to the task of general object detection and embrace an excellent MAP (mean average precision) score. However, these backbones perform inconsistently on the face dataset. Note that to extract the robust face features, backbones would be designed by the property of face dataset instead of transferring the backbones from the task of classification to face detection.

To further analyze the characteristics of different detection datasets discussed above, we calculate the distribution of object scales in the COCO and face scales in the Wider Face (shown in figure 1(b)). Object scales in the general object detection and face detection have clear gap. Almost 55% faces in the Wider Face dataset are with small scales (less than 20 pixel) and only 18% in the COCO dataset. Moreover, 90% face scales are smaller than 66 pixels. Most of these faces are matched with anchors in the shallow layers (conv2 and conv3).

In fact, the gap (37%) between object scales in the COCO and Wider Face dataset is much more than this. We unexpected find a mismatching training setting between the two detection tasks. The average scales of original images are 520, 940 pixels on the COCO and Wider Face dataset respectively. Figure 1(c) displays the relation between the input image scale and the detector performance. The mean scale of images in the face is approximately two times larger than that in the ImageNet. Additionally, to achieve better performance, general object detectors frequently enlarge the shorter side of an image to a fixed length (e.g. 800 pixels) and constrain the length (e.g. 1333 pixels) of a longer one in the stage of training. However, this process of data aug-

†Corresponding author.



(a) (b) (c)

Figure 1. Reasons for the inconsistent performance of the same backbone in different tasks. (a) The performance of different backbones on ImageNet, Wider Face, COCO datasets. We evaluate backbones on the two latter datasets by the AP and MAP score separately and adopt top-5 error rates on the ImageNet validation set. For clearly comparing the difference on ImageNet with others, we replace the top-5 error rates with top-5 correction rates (100 - top-5 error rates). (b) Cumulative density curve of face or object scale relative to the fixed scale (640). (c) The relation between the input image scale and the detector performance.

mentation in face detection is completely opposite. Considering that enlarging the scale of the face image plane generally brings the distortion because the faces have more complicated texture information than general objects, face detectors reduce the mean scale from 940 to 640 and consequently get the best results among all listed scales in the figure 1(c).

According to the training setting with the highest performance in figure 1(c), we consistently calculate the average scales of input face and general objects during training. The proportion of small face scales in the Wider Face dataset is far less than that of small general objects in the coco dataset (68.3% vs 2.3%). The gap has been enlarged from 37% to 66%. Therefore, the problem of why the same backbone performs differently on the face and coco dataset has been found. That is backbones designed by applying on ImageNet dataset with relatively large ground-truth are not appropriate to Wider Face dataset which has too many faces with small scales. Besides, in section 3.1, we demonstrate the performance of backbones on each pyramid layers to further explain their unsuitability.

As discussed above, the motivation of some tricks is also clear on the face detector. One is the utilization of conv2 layer to enhance the ability of recalling small faces. The other is that different from standard anchor designation in retinanet, most face detectors adopt an equal-proportion interval principle proposed by [28] to reduce the redundant anchors for shallow layers. These tricks are all designed for the improvement of detecting small faces. In this paper, considering the different property (distribution of ground-truth scales) between faces and general objects, we focus on designing the robust backbone and FPN for face detectors.

Recently, some works [31, 9] about neural architecture search illuminate us to design the proper backbone for face detectors. Most of them design network architecture based on CIFAR-10 or ImageNet dataset from enhancement of search efficiency, designing the search space and reducing inference time. As far as we know, no work based on NAS method focuses on improving shallow network layers, which directly helps detecting small faces.

In this paper, we first analyze the reason of why state-of-the-art backbones could not continue their superiority on the Wider Face dataset. Following this, section 3.1 shows the performance of different layers on backbones, which gives us a correct illumination on how to build a face-appropriate search space of backbone that is important for the one-shot NAS method. Second, by concatenating the shallow and deep feature map, FPN both enhances the performance of face and general object detectors significantly, especially for small scale objects and faces. However, as pointed out from section 3.2, there exist various connection modes among all candidate feature maps and FPN proposed by [17] is not the best choice. Therefore, our method formulates a FPN-attention module and further jointly search the face-appropriate backbone and FPN architecture.

In summary, we have made following main contributions:

- We find a meaningful phenomenon that the same backbone performs inconsistently between the task of object detection and face detection.
- Based on this observation, we design the face-appropriate search space of backbone and FPN by multiple exploration in section 3 and 4. We also demonstrate its superiority in section 5.1.
- We further propose a jointly searching backbone and FPN method by introducing a FPN-attention module.
- We achieve state-of-the-art results on AFW, PASCAL Face, FDDB, and Wider Face datasets.

2. Related Work

2.1. Face Detection

In recent years, face detection achieves a great progress both on the robust feature extractors and the connection of

feature maps. [23] first introduces the hand-crafted features and designs a rigid template for detecting faces. Inspired by this well-performing features, DPM [7] utilizes different models to extract all directions of human body. However, the performance of these feature extractors has been surpassed since the emergence of deep convolution networks. A large number of face detectors [5, 16, 29] utilize these backbones and achieve the state-of-the-art performance. However, few works tend to explore how much positive effectiveness these backbones can bring in. Considering there exist many small scale faces, ISRN [27] enlarges the convolution kernel size from 3 to 7 in the shallow resnet architecture, which generates more discriminative features in the shallow layers. In this paper, our proposed method is based on the NAS approach and focuses on searching the face-appropriate backbone and FPN architecture.

Besides, Retianet [18] combines FPN with base detectors and demonstrates their superiority on the general object detection. Many computer vision tasks (eg. image segmentation, Face detection) follow it and further demonstrate the robustness of FPN. Similarly, most of recently state-of-the-art face detectors [16, 24] adopt FPN architecture from p2 to p6. PyramidBox [22] formulates that deep convolution layer merely play no role in the process of feature merging and further addresses this with low feature pyramid network. DSFD [15] designs a feature enhance module which adds the dilation convolution after the current feature map cell interactives with neighbors. All these architectures are designed by artificial experience. Therefore, these methods would not be the best choices. By extensively analyzing the role of different FPN architectures in section 3.2, we design a reasonable search space for FPN.

2.2. Neural Architecture Search

[31] first designs the search space with a stacked cell level architecture and adopts reinforcement learning to optimize the model. However, this directly leads to the extravagant searching time. To reduce this, following methods introduce the weight sharing and gradient descending methods, which could extremely save the computational cost and achieve competitive performance. However, these classifiers are all designed on the ImageNet or CIFAR-10 dataset, leading the less persuasiveness on the object detection, especially on the face detection. DetNAS [4] proposes the backbone search for object detection. The pipeline based on the weight sharing method can be devided into two steps: 1) Supernet firstly pre-trained on the ImangeNet and fine-tuned on the COCO. 2) Then adopts the evolution algorithm to search the architecture of backbone. Because of a large number of backbones (Resnet50, Densenet121, ShuffleNet and so on) have shown consistent performance between ImageNet and COCO dataset, DetNAS can easily utilize the search space of previous works. However, this is not suitable for the face detection as discussed above. Nas-fpn [8] uses reinforcement learning to search the best proper FPN for general object detection, which gives us a progressive inspiration that we simultaneously search the backbone and FPN on the task of face detection by exploring the property of Wider Face dataset, qualitative analysis on diverse backbone and FPN architectures, proposing a FPN-attention module for searching bakcbone and FPN architectures jointly.

3. Effectiveness of Shallow Layers and FPN Architectures

In this section, we firstly qualitative analyze the reason of why resnet50 has better performance than resnet101 and resnet152. Then, we compare the performance of multiple FPN architectures with different backbones and discover some meaningful conclusions to help us build a face-appropriate search space for FPN.

3.1. Backbone

Extensive experiments above have shown that different from the same backbone performs almost consistently accuracy between ImageNet and COCO dataset, it has explicit difference between ImageNet and Wider Face dataset. This novel phenomenon needs further qualitative analysis except for the exploration of face scale distribution in the section 1. Therefore, we compare the performance of different layers on the backbones with same block architecture, demonstrating the reasonability of our proposed view (design a face-appropriate backbone).

Table 1. Performance of the resnet block with different depths and FPN architectures on Wider Face validation hard subset.

Architecture	p2	p3	p4	p5	p6	p7	All Layers
Resnet50 + FPN	87.8	87.6	86.8	88.0	88.2	87.9	**87.8**
Resnet101 + FPN	85.2	85.2	87.3	88.6	88.2	88.3	86.2
Resnet152 + FPN	84.7	84.4	86.8	88.4	88.0	88.3	85.1
Resnet50 + lFPN	**88.1**	**88.4**	87.8	87.8	87.6	87.7	**88.2**
Resnet50 + FEM	87.7	87.8	**88.4**	**88.9**	88.4	**88.5**	**88.0**
Resnet101 + lFPN	86.0	85.8	87.2	88.1	87.8	88.3	**85.8**
Resnet101 + FEM	85.2	85.1	87.2	88.4	**88.6**	88.2	**85.4**

In figure 1(a), we can see that there exists an unexplainable problem on the face detection that why the backbones consist of the same block architecture perform a downward trend with the increasing of stacked block. To resolve this, we show AP scores of each pyramid layers of resnet50, resnet101 and resnet152 respectively in table 1. For shallow pyramid layers (p2 – p3), face detector with resnet50 has the highest AP score. For deeper convolution layers (p4 – p7), we unexpected discover that instead of resnet152, resnet101 has the highest score on the face detector. Considering few faces are detected on deep layers, this enhancement can be ignored comparing with shallow pyramid layers. Moreover,

experiments also display that enhancing the depth of backbones has little influence on extracting deep discriminative face features and even becomes an obstacle for generating shallow robust face features. Therefore, different from that backbones in the general object detection highly emphasize the depth of deep convolution layers, high quality face-appropriate backbones should focus on the architecture of shallow layers.

3.2. Feature Pyramid Network

Besides, understanding roles that FPN architectures play on the face detector is critical to design face-appropriate FPN search space. SRN, PyramidBox, DSFD are three state-of-the-art models on face detection and introduce FPN, lFPN, FEM architectures respectively. For fairly comparing these three architectures, we incorporate them with ResNet50, ResNet101 and table 1 shows their performance on each pyramid layers. We can see that in p2 and p3, lfpn has the best AP score and in other layers FEM is the best. Additionally, we find Resnet50 with lFPN and FEM increases the 0.4%, 0.2% AP than baseline and Resnet101 unexpected decreases 0.4%, 0.8% AP. This interesting experiment result illustrates the performance of feature maps connection is highly related to the architecture of backbones.

To design the efficient search space for FPN, we conduct a meaningful experiment to explore which kinds of layers (current layer or others) are important during the process of feature concatenation. Table 2 shows the performance between current layer and other layers with the same resolution that is generated by up-sampling or down-sampling strategy. The AP score in the current pyramid layer is higher than others with considerable gap. All these conclusions enlighten us to design a face-appropriate search space for FPN architecture in section 4.2.2.

Table 2. Performance of the resnet block with different depths and FPN architectures on Face validation hard subset.

Pyramid Layer	p2	p3	p4	p5	p6
p2	**87.8**	78.8	74.2	68.3	64.2
p3	84.4	**87.6**	72.8	64.3	65.4
p4	83.2	81.7	**86.8**	76.5	72.4
p5	77.4	79.3	82.4	**88.0**	81.2
p6	68.5	74.2	77.8	85.4	**88.2**

4. Method

In this section, we first review the SPNAS [9] method that gives us deep inspiration. Second, we formulate the face-appropriate search spaces for backbone and FPN architecture. Finally, we propose a FPN-attention module to help search the backbone and FPN on the face detection simultaneously.

4.1. Single Path One-Shot NAS Method

Early NAS methods constantly sample the architecture from pre-defined search space directed by the reinforce learning. However, each sampling process are trained from scratch, leading the expensive computation cost. ENAS [20] resolves this by proposing a weight sharing method. Actually, there exists a problem in these approaches and pointed out by SPNAS [9]: a deep coupling between the supernet weights and architecture parameters. To alleviate this, SPNAS further proposes a single path supernet and uniform sampling. The target of SPNAS is that the optimization of supernet weights Wa should be in a way that all architectures are optimized simultaneously. This is expressed as:

$$W_{\mathcal{A}} = \underset{W}{\operatorname{argmin}} \mathbb{E}_{a\sim\Gamma(\mathcal{A})} \left[\mathcal{L}_{\text{train}}(\mathcal{N}(a, W(a)))\right]. \quad (1)$$

As represented in all NAS works, $\mathcal{A}$ represents architecture search space by a directed acyclic graph(DAG). A sampled architecture (a) is a subgraph and $\Gamma(\mathcal{A})$ is a uniform distribution. $W(a)$ denote weights in the a and updated in each step alone.

4.2. Search Space

Although SPNAS and FPN-NAS methods both demonstrate their excellence in the image classification and generic object detection respectively, directly applying them on the face detectors to search face-appropriate backbone and FPN inevitably bring following two problems.

1) Search space of backbone and FPN architecture should different with that in general object detection (discussed in section 3 and 1).

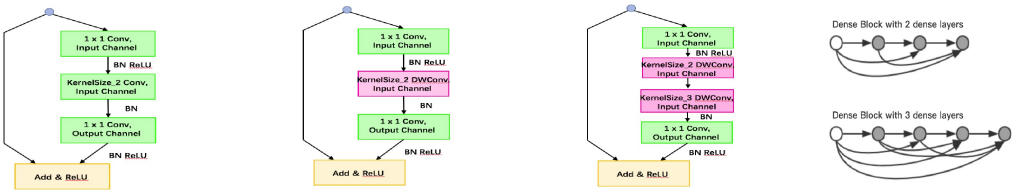
2) FPN-NAS focuses on designing the optimal FPN with a fixed backbone. However, table 1 demonstrates the strong correlation between the architecture of backbone and FPN. In other words, to search the excellent face detector, these two architectures should be searched simultaneously, instead of finding the backbone on the condition that the backbone search is completed.

In following sections, we resolve this two problems by designing the face-appropriate backbone and FPN search space, proposing a jointly searching backbone and FPN architecture method problems.

4.2.1 Backbone

Extensively ablative experiments have shown in section 3.1 and clearly bring two convincing conclusions:

1) For backbones consisting of the same blocks (e.g. bottleneck block, densenet block), as the network deepens, the performance decreases gradually on the face detectors. This



(a) (b) (c) (d)

Figure 2. The search space for face-appropriate backbone. All these four blocks are deigned for shallow layers. And only first three blocks are for deep layers. Input channel from conv2 to conv7 are 64, 256, 256, 512, 512, 512 and output channel are 256, 256, 512, 512, 512, 256. a) KernelSize_2 contains [3x3, 5x5, 7x7, 9x9] in shallow layers and [3x3, 5x5, 7x7] in deep layers. b) KernelSize_2 is the same as that in a). c). [KernelSize_2, KernelSize_3] contains [3x3, 5x5], [5x5, 7x7] in all layers. d). We design two dense blocks: 2 dense layers with 32 growth rate and 3 dense layers with 16 growth rate.

is an inconsistent phenomenon when applying on the general object detection and image classification. Therefore, previous search space on the image classification can not be appropriate to face detectors.

2) Following this view, another meaningful conclusion has been mined by further efforts on analyzing the performance of each layers on the backbone. That is, the performance in shallow layers has significant difference while adopting different architectures of backbones. However, this difference merely disappeared in deeper layers. Note that face-appropriate backbones should pay more attention on the designation of shallow layers.

Furthermore, considering that the accuracy of SPNAS method is highly related to block components in the search space, we begin to search suitably candidate blocks for shallow layers. We first select many state-of-the-art blocks (such as bottleneck, densenet block, shufflenet block, inceptionv4 block and some heuristic blocks) to validate their effectiveness on the shallow layers. Table 3 and 4 show their performance and some candidate architectures with lower performance are ignored to show. Inspired by this, figure 2 further displays the final search space. Each block consists of a maximum of 4 different convolution layer. Why adding this constraint is that incorporating redundant blocks in the search space decreases the model performance (please refer to section 5.1).

Table 3. Performance of state-of-the-art blocks on p2/p3 on Wider Face validation hard subset.

Conv	Dwconv	3x3	5x5	7x7	9x9	11x11	p2	p3
✓		✓					87.8	87.6
✓			✓				87.7	87.8
✓				✓			88.1	87.4
✓					✓		87.4	87.3
✓						✓	87.8	85.4
	✓	✓					87.6	87.2
	✓		✓				88.0	87.4
	✓			✓			**88.2**	87.8
	✓				✓		87.7	87.6
	✓	✓	✓				**88.2**	**88.0**
	✓	✓		✓			87.8	87.6
	✓		✓	✓			88.1	87.7

Table 4. Performance of dense block on p2/p3 on Wider Face validation hard subset.

Dense Block	Number of Dense Layers	Growth Rate	number of Blocks	p2	p3
Dense Block	2	16	4	87.7	86.8
Dense Block	2	32	4	87.9	86.6
Dense Block	3	16	4	87.8	86.4
Dense Block	3	32	4	**88.1**	**87.1**

4.2.2 Feature Pyramid Network

Section 3.2 illustrates that different FPN architectures have their particular advantages on the feature concatenation and shows the role of each feature map in the process of feature concatenation. Therefore, in our method, the search space make two improvements on the basis of Nas-fpn's which contains connections between any two layers. One is adding the receptive field module after feature merging. The other is that when selecting any layer resolution as output, the same resolution feature must be selected as a candidate. This could remarkably enhance the feature representation both in shallow and deep layers with following proposed method.

4.3. Joint Searching Backbone and FPN Architecture

Inspired by the great compatibility of FPN architecutre in many conputer vision tasks, an intuitive method to search excellent backbone and FPN architecture for face detectors is that searching the backbone first, then the FPN (intuitive search method). However, this is sub-optimal because incorporating different backbones with the same FPN architecture achieve inconsistent performance (please refer to section 5.1) unexpectedly.

To explore how to solve this, we directly apply SPNAS to train a supernet for face detectors. The pipeline is easy: in each iteration, we random sample an architecture from the search space of backbone and FPN simultaneously and then optimized by the equation 1 (two-step search method). Although this sounds reasonable, there are two issues:

1)There are no parameters in FPN, which obey the

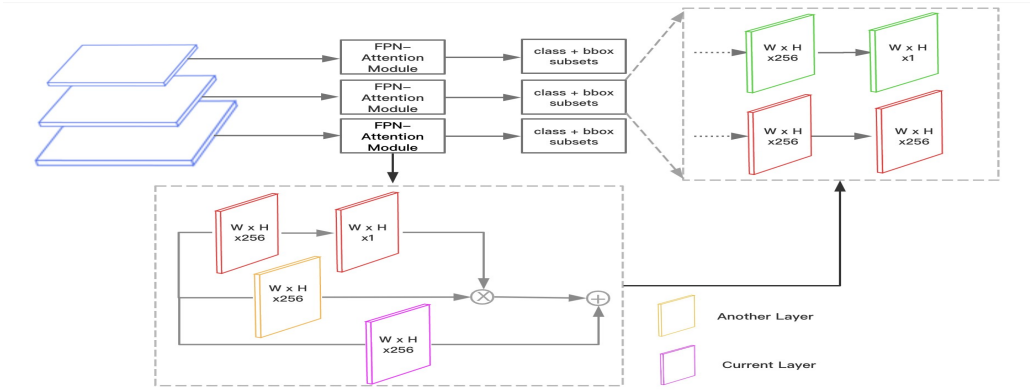


Figure 3. The pipeline of training a supernet. In each step of optimization, an architecture of backbone is randomly sampled. Then we randomly sample an architecture of FPN by following steps with 5 times (p2 - p6). Step 1. Randomly select a feature layer fi from candidates and the output feature resolution is the same as $fi's$. Step 2. Randomly select another feature layer fj from candidates. Combine fi and up-sampling or down-sampling fj by element-wise operation. Step 3. Add the Combination feature map to candidates.

weight sharing method. It is not convincing that parameters in the backbone implicitly represent the role of FPN and this method achieves lower accuracy (please refer to section 5.1).

2)The correlation between the backbone and FPN architecure is reduced remarkably. The reason is that for each sampling process, one backbone only can integrate with single FPN module while others are neglected.

Our method resolves these two problems by introducing an FPN-attention module and shown in figure 3. This module helps layer merge more confidential information. For the first problem, this new proposed tensor satisfies the weight sharing method and the performance of different FPN modules is highly related to this tensor. For the second one, this module could entirely resolve the inhomogeneous sampling in searching FPN architecture for it explicitly represents the compatibility between a backbone and any candidate FPN architectures.

4.4. Search Algorithm

Following [2], our search algorithm adopts random search method. Evolutionary algorithm can easily satisfy some hard constraints (e.g., FLOPS or inference speed) and demonstrate its excellence in SPNAS, so why not use this? The reason lies in the process of mutation. This procedure is not suitable for ours. Because the backbone and FPN architecture of parents are connected closely, by the mutation to generate children, it is difficult to generate high quality children.

5. Experiments

In this section, we first elaborate the implementation details of the baseline and our method. Then we conduct ablative experiments to demonstrate the superiority of our methods. Finally, our method achieves state-of-the-art on popular face benchmarks.

5.1. Ablation Study

For the sake of fairly comparing different experiments, we evaluate them on the authoritative and robust Wider Face dataset [26]. This famous dataset, contains 32203 images together with 393703 faces. Additionally, images consisting of 61 event classes have a high degree of variability in scale, pose and occlusion. For images in each classes, they are divided into training (40%), validation (10%) and testing (50%) set. Besides, by the difficulty of detection results, all faces are classified into Easy, Medium, Hard subsets. The performance of detectors on the validation set are taken into account in ablative experiments. Average Precision(AP) score are regarded as the evaluation metric. For our method, we search the architecture on the training set and evaluate it on the validation set.

Data Augmentation. Data augmentation in our method is expressed as:

- Color distort: Applying some photometric distortions similar to [11].
- Data anchor sampling: Resizing all training images by reshaping a random face in this image to a smaller or slightly larger scale.
- Horizontal flip: Resizing the cropped image patch generated in data augmentation to 640 X 640. Then each image adopts horizontal flip with a probability of 0.5.

Training Details. In the phase of training, we adopt retinanet [18] as baseline. The remaining architecture can be seen in the figure 3. The process of training model contains the supernet training and the best architecture training. These two training procedures adopt the same settings (e.g.

data augmentation, optimizer, etc.). Supernet is trained for 150000 iterations and the other is 80000 iterations.

Training Settings. Training Settings: For anchor settings, we set 6 anchor scales from the set {16, 32, 64, 128, 256, 512} and anchor ratio as 1. The threshold of IOU for positive anchors is changed to 0.35, and ignore zone only contains the faces whose scale is smaller than 8. For optimization details, each training iteration contains 7 images per GPU on 4 NVIDIA Tesla P40s. Models are optimized by synchronized SGD. The momentum and weight decay are set to 0.9 and 5$\times$10-5, respectively. For learning rate schedule, the initial learning rate is set to 1e-3 and decreases to 1e-4 in 60000 iterations.

Final Model Selection. After training the supernet completely , we randomly sample around 10000 architectures or architectures under FLOPs constraint from the search space, and then evaluated on the validation dataset. To quickly evaluate the candidate architecture, we shrink the Face validation dataset into 200 images. Besides, before evaluating the model, we need recalculate the Batch Normalization operations on a random subset of training data (500 images). Finally, we select the architecture with the highest AP score among all 10000 candidate ones and then train this final architecture on the Face training dataset from scratch. Based on the one-shot model, each architecture only takes almost 60 seconds on a P40 GPU. For all 10000 architectures, the total time we need is 160 GPU-hours.

The Effect of Our Designed Face-appropriate Search Space. To demonstrate the effectiveness of backbone search space, we fix the FPN architecture and only search the backbone of detectors. The following ablative experiments are shown in table 5. FDB(k) represents our designed search space for backbone, k denoted as the number of blocks in the search space of shallow layers. ICB(k) is a search space in image classification applied in SPNAS. From the table 5, we can see FDB(4) achieves the best performance with 90.8% AP which exceeds FDB(3), FDB(6), FDB(8) with 0.6%, 2.7%, 3.4% AP and ICB(4) with 2.6% AP on the validation hard set. Note that the capacity of search space should be proper instead of the huge one has better performance.

Similarly, we adopt resnet50 backbone for exploring the efficiency of our proposed FPN search space. Table 6 show the performance of different FPN modules and ours outperforms the others 0.8% AP at least on the validation hard set.

Table 5. Performance of our designed face-appropriate search space for backbone on Wider Face validation set.

	Easy	Medium	Hard
FDB(4) + FPN	**95.8**	**95.2**	**90.8**
FDB(3) + FPN	95.2	94.5	90.2
FDB(6) + FPN	94.2	92.4	88.1
FDB(8) + FPN	93.8	93.2	87.4
ICB(4) + FPN	95.2	94.4	88.2

Table 6. Performance of our designed face-appropriate search space for FPN on Wider Face Validation set.

	Easy	Medium	Hard
Resnet50 + FPN	95.1	94.4	87.8
Resnet50 + lFPN	94.9	94.3	88.2
Resnet50 + FEM	95.4	**94.8**	88.0
Resnet50 + Nas-fpn	95.3	94.2	88.6
Resnet50 + Ours	**95.7**	94.6	**89.4**

Table 7. Performance of jointly searching backbone and FPN on Wider Face validation set.

	Easy	Medium	Hard
Intuitive Search Method	95.6	94.8	89.2
Two-step Search Method	95.8	95.2	89.7
Joint Searching Backbone and FPN	**96.5**	**95.7**	**91.7**

The Effect of jointly searching backbone and FPN. By qualitative analysis in section 3, our method formulates jointly searching method with our proposed FPN-attention module. Table 7 shows the performance of our jointly searching method and two-step searching method (as discussed in section 4.3). Besides, we also compare our method with intuitive search method (discussed in section 4.3). The results further prove the dominance of our method instead of depending on qualitative analysis solely.

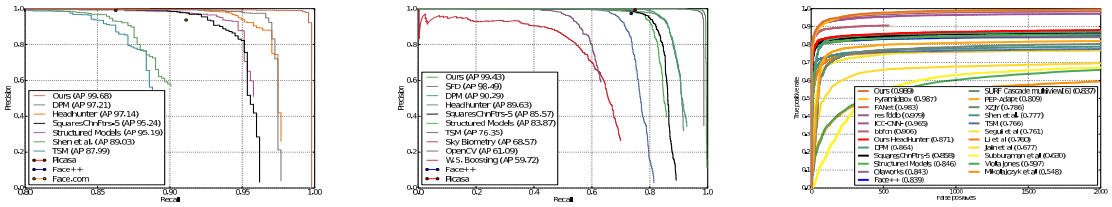
The Effect of random search method. Although SPNAS has demonstrated the superiority of evolutionary algorithm in searching backbone, directly applying it on searching backbone and FPN simultaneously has a lower performance (shown in table 8). The hyper parameters in evolutionary algorithm are set similarly with that in the SPNAS, population size P=50, max iterations T=20, k=10, and with probability 0.1 to produce a new candidate.

Table 8. Performance of search algorithm on Wider Face validation set.

	Easy	Medium	Hard
Evolutionary Algorithm	96.4	95.7	90.8
Random Search Method	**96.5**	**95.7**	**91.7**

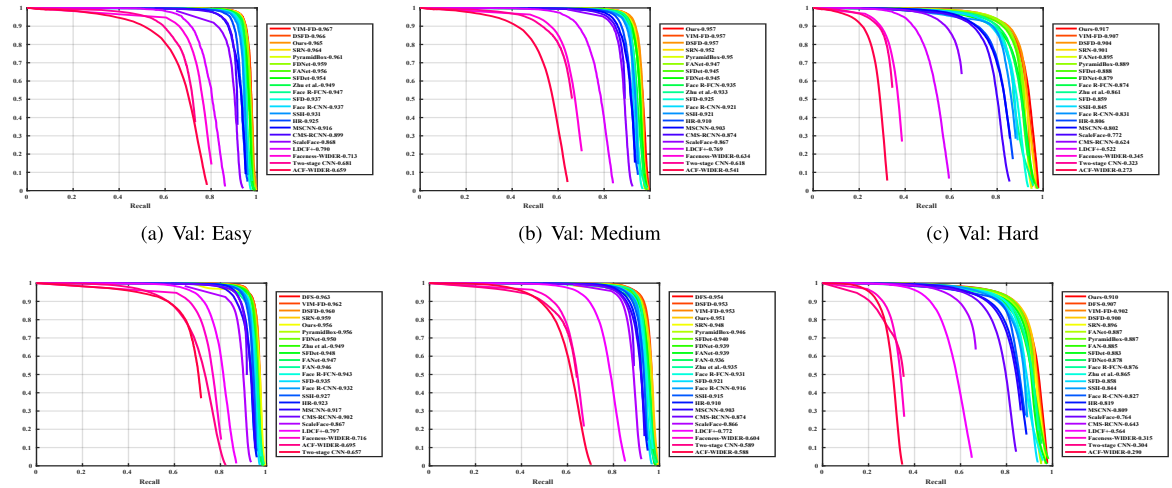
Our proposed backbone vs Others. To fairly compare our method with others equipped with different backbones (resnet50, densenet121, inceptionv4) and FPN architecture, we add the FLOPs constraint in the process of final model selection. From the table 9, our backbone under FLOPs constraint outperforms resnet50, resnet101, densenet121, 2.4%, 4.0%, 1.7%AP respectively on the validation hard subset.

At the same time, with the constraint decreasing, our method has a remarkable enhancement. However, this is an opposite phenomenon comparing to resnet, densenet architecture. Note that our searched backbone is more suitable for face detectors.



(a) AFW (b) PASCAL Face (c) FDDB

Figure 4. Evaluation on the common face detection datasets.



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

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

Figure 5. Precision-Recall (PR) curves on Wider Face validation and testing subsets.

Table 9. Performance of our proposed backbone and others on Wider Face validation set.

	FLOPs	Easy	Medium	Hard
Resnet50 + FPN	3.8G	95.1	94.4	87.8
Resnet101 + FPN	7.6G	94.9	93.5	86.2
Densenet121 + FPN	3.0G	95.3	93.4	88.5
Ours + FPN	**2.9G**	**95.7**	**95.1**	**90.2**
Ours (final model)	4.1G	**96.5**	**95.7**	**91.7**

5.2. Evaluation on Common Benchmarks

To evaluate the robustness of our proposed method, We test it on common face detection benchmarks, including Wider Face [26], Annotated Faces in the Wild (AFW) [30], PASCAL Faces [25], FDDB [13]. Our face detector is trained by using Wider Face training set only and is tested on all these benchmarks with state-of-the-art performance.

AFW Dataset. This dataset [30] contains 205 images with 473 annotated faces. Figure 4(a) shows that our detector outperforms others significantly.

PASCAL Face Dataset. This popular benchmark consists of 851 images with 1,335 annotated faces. Figure 4(b) demonstrates the superiority of our proposed method.

FDDB Dataset. Comparing with two above datasets, FDDB dataset has lower image resolutions and complicated scenes, such as occlusions, huge poses. Figure 4(c) shows our method also achieves the states-of-the-art performance on this challenging dataset.

Wider Face Dataset. We test our models on both testing and validation sets. Figure 5 shows the precision-recall curves both on the validation and testing sets. Our approach achieves 96.5% (Easy), 95.7% (Medium), 91.7% (Hard) AP on the validation dataset and 95.6% (Easy), 95.1% (Medium), 91.0% (Hard) AP on test dataset.

6. Conclusions

We first find an inconsistent phenomenon that backbones with high performance on COCO and ImageNet dataset always get lower AP score on Wider face dataset. This instructive appearance inspires us to design face-appropriate architectures. Then by analyzing the performance of feature maps on various backbones, we design the scientific search space for backbone and FPN architecture. Finally, to enhance the correlation between FPN and backbone during training, we propose a FPN-attention module and to search backbone and FPN architecture jointly.

References

[1] Grigory Antipov, Moez Baccouche, and Jean-Luc Dugelay. Face aging with conditional generative adversarial networks. In *2017 IEEE International Conference on Image Processing (ICIP)*, pages 2089–2093. IEEE, 2017.

[2] Gabriel Bender, Pieter-Jan Kindermans, Barret Zoph, Vijay Vasudevan, and Quoc Le. Understanding and simplifying one-shot architecture search. In *International Conference on Machine Learning*, pages 549–558, 2018.

[3] Xudong Cao, Yichen Wei, Fang Wen, and Jian Sun. Face alignment by explicit shape regression. *International Journal of Computer Vision*, 107(2):177–190, 2014.

[4] Yukang Chen, Tong Yang, Xiangyu Zhang, Gaofeng Meng, Chunhong Pan, and Jian Sun. Detnas: Neural architecture search on object detection. *arXiv preprint arXiv:1903.10979*, 2019.

[5] Cheng Chi, Shifeng Zhang, Junliang Xing, Zhen Lei, Stan Z Li, and Xudong Zou. Selective refinement network for high performance face detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 8231–8238, 2019.

[6] J Deng, J Guo, and S Zafeiriou. Arcface: additive angular margin loss for deep face recognition. corr abs/1801.07698 (2018), 1801.

[7] Pedro F Felzenszwalb, Ross B Girshick, David McAllester, and Deva Ramanan. Object detection with discriminatively trained part-based models. *IEEE transactions on pattern analysis and machine intelligence*, 32(9):1627–1645, 2009.

[8] Golnaz Ghiasi, Tsung-Yi Lin, and Quoc V Le. Nas-fpn: Learning scalable feature pyramid architecture for object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7036–7045, 2019.

[9] Zichao Guo, Xiangyu Zhang, Haoyuan Mu, Wen Heng, Zechun Liu, Yichen Wei, and Jian Sun. Single path one-shot neural architecture search with uniform sampling. *arXiv preprint arXiv:1904.00420*, 2019.

[10] 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.

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

[12] Forrest Iandola, Matt Moskewicz, Sergey Karayev, Ross Girshick, Trevor Darrell, and Kurt Keutzer. Densenet: Implementing efficient convnet descriptor pyramids. *arXiv preprint arXiv:1404.1869*, 2014.

[13] Vidit Jain and Erik Learned-Miller. Fddb: A benchmark for face detection in unconstrained settings. 2010.

[14] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[15] 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.

[16] 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.

[17] 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.

[18] 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.

[19] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014.

[20] Hieu Pham, Melody Y Guan, Barret Zoph, Quoc V Le, and Jeff Dean. Efficient neural architecture search via parameter sharing. *arXiv preprint arXiv:1802.03268*, 2018.

[21] 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.

[22] 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.

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

[24] Jianfeng Wang, Ye Yuan, and Gang Yu. Face attention network: An effective face detector for the occluded faces. *arXiv preprint arXiv:1711.07246*, 2017.

[25] Junjie Yan, Xuzong Zhang, Zhen Lei, and Stan Z Li. Face detection by structural models. *Image and Vision Computing*, 32(10):790–799, 2014.

[26] 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.

[27] Shifeng Zhang, Rui Zhu, Xiaobo Wang, Hailin Shi, Tianyu Fu, Shuo Wang, and Tao Mei. Improved selective refinement network for face detection. *arXiv preprint arXiv:1901.06651*, 2019.

[28] 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.

[29] Chenchen Zhu, Ran Tao, Khoa Luu, and Marios 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.

[30] Xiangxin Zhu and Deva Ramanan. Face detection, pose estimation, and landmark localization in the wild. In *2012*

IEEE conference on computer vision and pattern recognition, pages 2879–2886. IEEE, 2012.

[31] Barret Zoph, Vijay Vasudevan, Jonathon Shlens, and Quoc V Le. Learning transferable architectures for scalable image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8697–8710, 2018.