ASFD: Automatic and Scalable Face Detector

Jian Li*
swordli@tencent.com
Tencent Youtu Lab
Shanghai, China

Bin Zhang*
z-bingo@seu.edu.cn
Southeast University
Nanjing, China

Yabiao Wang
caseywang@tencent.com
Tencent Youtu Lab
Shanghai, China

Ying Tai
yingtai@tencent.com
Tencent Youtu Lab
Shanghai, China

Zhenyu Zhang
joeyzyzhang@tencent.com
Tencent Youtu Lab
Shanghai, China

Chengjie Wang
jasoncjwang@tencent.com
Tencent Youtu Lab
Shanghai, China

Jilin Li
jerolinli@tencent.com
Tencent Youtu Lab
Shanghai, China

Xiaoming Huang
skyhuang@tencent.com
Tencent Youtu Lab
Shanghai, China

Yili Xia†
yili_xia@seu.edu.cn
Southeast University
Nanjing, China

ABSTRACT

Along with current multi-scale based detectors, Feature Aggregation and Enhancement (FAE) modules have shown superior performance gains for cutting-edge object detection. However, these hand-crafted FAE modules show inconsistent improvements on face detection, which is mainly due to the significant distribution difference between its training and applying corpus, *i.e.* COCO vs. WIDER Face. To tackle this problem, we essentially analyse the effect of data distribution, and consequently propose to search an effective FAE architecture, termed AutoFAE by a differentiable architecture search, which outperforms all existing FAE modules in face detection with a considerable margin. Upon the found AutoFAE and existing backbones, a supernet is further built and trained, which automatically obtains a family of detectors under the different complexity constraints. Extensive experiments conducted on popular benchmarks, *i.e.* WIDER Face and FDDB, demonstrate the state-of-the-art performance-efficiency trade-off for the proposed automatic and scalable face detector (ASFD) family. In particular, our strong ASFD-D6 outperforms the best competitor with AP 96.7/96.2/92.1 on WIDER Face test, and the lightweight ASFD-D0 costs about 3.1 ms, *i.e.* more than 320 FPS, on the V100 GPU with VGA-resolution images.

CCS CONCEPTS

• **Computing methodologies** → **Object detection**.

*Both authors contributed equally to this research.
†Correspondence author.

MM '21, October 20–24, 2021, Virtual Event, China

ACM ISBN 978-1-4503-8651-7/21/10…$15.00
https://doi.org/10.1145/3474085.3475372

KEYWORDS

face detection, neural architecture search, multi-task loss, compound scaling

ACM Reference Format:
Jian Li, Bin Zhang, Yabiao Wang, Ying Tai, Zhenyu Zhang, Chengjie Wang, Jilin Li, Xiaoming Huang, and Yili Xia. 2021. ASFD: Automatic and Scalable Face Detector. In *Proceedings of the 29th ACM International Conference on Multimedia (MM '21), October 20–24, 2021, Virtual Event, China.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3474085.3475372

1 INTRODUCTION

Face detection serves as a fundamental step towards various face-related applications, such as face alignment [30], face recognition [13] and face analysis [24]. It aims locate the face region (if any) in a given image, which has been a long standing research topic ranging from [35] to deep learning based methods [5, 43].

Beyond the scope of face, general object detection has been significantly pushed by the development of deep convolution neural networks [12, 22, 26, 27]. Among one of the representative framework, single-stage anchor-based detector with pyramid features has been thoroughly studied recently [17, 22] and is dominant for face detection [6, 15, 33, 42, 43]. In this framework, the regular and dense anchors with different scales and aspect ratios are tiled over all locations of the feature map, and the pyramid features are extracted by the backbone and enhanced by the neck, which is subsequently plugged with both classification and regression branches.

Towards the design of Feature Aggregation and Enhancement (FAE) modules for these methods, Feature Pyramid Network (FPN) and its variants aggregate hierarchical features via the preset pathway, *e.g.* top-down and bottom-up path, to effectively fuse multi-scale features [15, 21, 32, 33, 41]. For another instance, ASPP [2, 25], RFB [20] and RFE [8] modules are proposed to enhance the feature representation by adjusting the effective receptive fields. Recently, Neural Architecture Search (NAS) has been also investigated for object detection, which has achieved remarkable performance gains, such as NAS-FPN [10], AutoFPN [37] and NAS-FCOS [36]. However, such a gain is severely not generalized when applying to face detection.

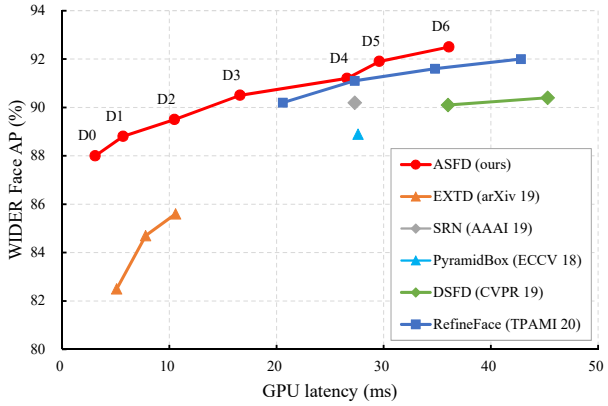


Figure 1: Performance-efficiency trade-off on WIDER Face validation for different face detectors. The proposed ASFD outperforms a range of state-of-the-art methods.

Fig. 2 (a) shows a quantitative investigation of the cutting-edge FAE modules discussed above, in which the significant drops have been shown when they are applied to face domain. Even the automatic learning based method, *a.k.a.* NAS-FCOS [36] that performs 1.6 lower than the baseline. This phenomenon highlights the domain gap between general object and face detection. To explain, we utilize cumulative distribution function to model the corresponding datasets, *e.g.* WIDER Face [39] and COCO [18] in terms of the relative size of boxes and the number of boxes in each image, as presented in Fig. 2 (b) and (c) respectively. As a result, the relative scale of faces is much smaller than objects in generic object detection, and there are more faces in each image than objects in COCO. These characteristics also determine the design principles of modern face detectors. For instance, the shallower feature map is adopted to detect the small faces. And more predicted results are retained before and after the non-maximum-suppression for the high recall rate. Since FAE modules designed for generic object detectors are weak when dealing with small-scale and crowded objects, therefore, false positives inevitably exist when they are applied to face domain, resulting in performance degradation.

In this paper, a novel NAS based face detector framework termed Automatic and Scalable Face Detector (ASFD) is introduced, which is designed upon the basis of quantitative observations as above. The proposed ASFD is equipped with an effective FAE module, namely AutoFAE, which is discovered in a face-suitable search space, and then automatically scaled up/down to meet different requirements. In particular, we first analyze why the domain gap between the generic object and face detection would cause such an impact as Fig. 2 (a). The performance degradation in the face domain is caused by the large semantic differences and unreasonable receptive fields for aggregated features. Then, we propose a face-suitable search space that aggregates a feature with similar-scale ones and enriches the feature presentation with different operations for different pyramid levels. And the AutoFAE module is searched by a gradient-NAS method [19, 38], and can achieve consistent gains on both face detection and generic object detection, as presented in Fig. 2 (a). Finally, we build a supernet consisting of the found

Pyramid Level	P2	P3	P4	P5	P6	P7
P2	82.9	84.3	85.0	84.7	84.5	82.7
P3	83.0	82.9	85.1	84.8	84.5	83.2
P4	83.0	83.2	82.9	84.5	83.8	83.3
P5	82.7	83.0	83.0	82.9	83.7	83.2
P6	82.8	83.0	82.9	82.9	82.9	83.1
P7	82.7	82.8	83.3	83.0	83.0	82.9

Table 1: Performance of FPN on Hard subset of WIDER Face validation while a pyramid level (indicated by the row) is aggregated by a specific level (indicated by the column).

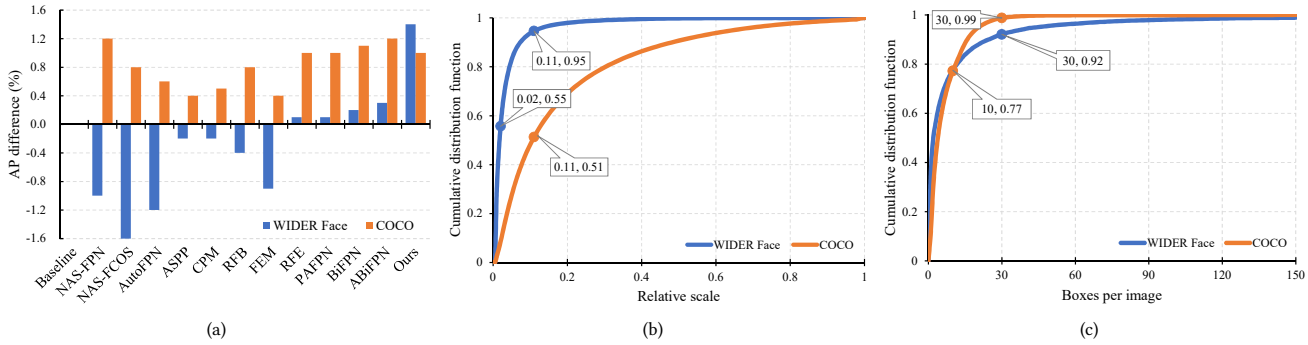
AutoFAE and a series of backbones, *e.g.* ResNet [12], and automatically obtain the proposed ASFD family to meet different complexity constraints via a one-shot NAS [7, 11]. It is worth noting that the ASFD family achieves the state-of-the-art performance-efficiency trade-off, as presented in Fig. 1 [5, 15, 33, 40, 42]. Especially, the light-weight ASFD-D0 can run more than 320 FPS with VGA-resolution images on a V100 GPU, and the strong ASFD-D6 obtains the highest AP scores on popular benchmarks, *i.e.* WIDER Face and FDDB. To sum up, this work makes following contributions:

- We observe an interesting phenomenon that some previous FAE modules perform well in generic object detection but fail in face detection, and conduct extensive experiments to illustrate why this phenomenon occurs.
- Based on the observations, we design a face-suitable search space for feature aggregation and enhancement modules, and discover an effective and generalized AutoFAE module via a joint searching method.
- Extensive experiments conducted on the popular benchmarks demonstrate the better performance-efficiency trade-off of the proposed ASFD.

2 RELATED WORK

2.1 Feature Aggregation and Enhancement.

In recent years, generic object detection and face detection have been dominated by deep learning based methods. SSD [22] is the first to predict objects using the multi-scale pyramid features, FPN [16] proposes to enrich the feature presentation of multi-scale features by a top-down pathway. Recently, many works are devoted to how to aggregate and enhance multi-scale features effectively. [21] and [32] enhance the entire feature hierarchy by the bottom-up path augmentation. [25] proposes a novel recursive FPN that incorporates extra feedback connections from FPN into the bottom-up backbone layers. Nowadays, NAS-based methods have demonstrated much success in exploring a better architecture for feature fusion and refinement [10, 36, 37]. Besides, feature enhancement modules are also be widely studied. Inception [28, 29] aims to capture different size of receptive fields via a multi-branch structure. [42] introduces rectangle receptive fields by a novel enhancement module. [15, 20, 25] adopt dilated convolution with different rates to enhance the feature discriminability and robustness. However, as illustrated in Fig. 2, some of them seem to be ineffective in face detection.



(a) (b) (c)

Figure 2: (a) Comparison of our AutoFAE against other FAE modules on WIDER Face and COCO validation. The performance gaps with the baseline are indicated by blue and orange bars respectively, and RetinaNet is adopted as the baseline. (b) Cumulative distribution function (CDF) of the relative scale of bounding boxes. 51% of objects in COCO have a relative scale below 0.11. For the same scale, the proportion in WIDER Face is 95%, while for a similar proportion, 55% of faces in WIDER Face are less than 0.02. (c) CDF of the number of boxes in each image. The distribution of images containing more than 10 boxes for WIDER Face is long-tailed, *e.g.* 99% of images in COCO have less than 30 objects, while there are many images in WIDER Face that contain more than 150 faces.

2.2 Neural Architecture Search.

NAS first uses reinforcement learning to search for hyper-parameters in the structure or used in the training process [31, 44, 45]. Recent researches focus on the automatic search of network architecture. Based on the idea of weight sharing, some works try to build the final structure by stacking a searched cell several times [19, 38], and other methods [1, 4, 7, 11, 23] decouple the training and searching process and directly train a supernet by randomly sampling a single-path network at each time. As for their applications on object detection to fuse the multi-scale features, NAS-FPN [10] searches the irregular connections among pyramid layers with an RNN controller for aggregating the multi-scale features. AutoFPN [37] and NAS-FCOS [36] discover the aggregation modules within a fully-connected search space densely connecting any two layers, in which features from some layers that damage the aggregated feature may be introduced causing accuracy degradation. BFBox [23] is the first attempt of NAS on face detection and proposes a face-suitable search space. Although the novel backbone and neck networks are discovered among the search space, its performance is still worse than the state-of-the-art face detectors.

In this work, the sparse cross-scale connections of FA module are searched based on a face-suitable search space rather than in a violent fully-connected manner. And various FE modules with different operations and topologies are discovered for different pyramid levels.

3 PROBLEM ANALYSIS

Fig. 2 (a) is sufficient to illustrate the inconsistency between general object detection and face detection. In order to further analyze the reason why this phenomenon occurs, the effects of feature aggregation and enhancement modules are discussed respectively in this section.

Module	P2	P3	P4	P5	P6	P7
ASPP	86.5	86.8	86.9	87.2	**87.5**	87.1
CPM	86.8	86.9	87.0	87.1	**87.4**	87.2
RFB	86.8	86.9	87.0	87.3	**87.4**	87.2
RFE	87.4	**87.6**	87.5	87.4	87.2	87.1

Table 2: Performance of different feature enhancement modules when operated on different pyramid levels.

3.1 Feature Aggregation (FA).

Firstly, extensive experiments are conducted to explore the relationship between performance and cross-connection of FA modules. These experiments are to add a FA module (for simplicity, FPN [16]) in turn between any two pyramid features and aggregate one feature with another one after resizing to the same shape. Table 1 shows the results on the diagonal indicating RetinaNet without FPN, and upper triangle and lower triangle representing top-down and bottom-up paths respectively, especially, red and blue fonts indicate top-down and bottom-up paths in FPN and PAFPN. it is clear to conclude that aggregating multi-scale features through top-down paths is superior to the bottom-up ones, especially these two layers used are close to each other. As the distance increasing, some connections even cause performance degradation, *e.g.* $AP_{.50}$ drops 0.2 when P2 is aggregated by P7. Therefore, small faces only occur in the shallow features cannot be enhanced by semantic-rich features with large scale difference. It reveals that NAS-FPN, AutoFPN and NAS-FCOS are sub-optimal to fuse features through a fully-connected or irregular connection in the face domain.

3.2 Feature Enhancement (FE).

Similar experiments are conducted to demonstrate the effects of different feature enhancements modules. As shown in Table 2, there

are significant performance differences when a FE module is applied to the different pyramid layers. In general, ASPP [2, 25], CPM [15, 33] and RFB [20] employ dilated convolution with different rates to enlarge the receptive fields, they can obtain the consistent performance when applied to different pyramid layers. Particularly, they would damage the shallow features especially the first two layers, and cause severe performance degradation. RFE [42] aims to enrich the features by introducing the rectangle receptive fields and performs well on all pyramid layers especially the shallow layers. A meaningful conclusion can be drawn that the shallower features seem to prefer a more diverse receptive field while the deeper layers favor a larger one. This is mainly because detecting faces with occlusion or extreme-pose that appear in the shallow layers expect more robust features, and the large faces require features with large receptive fields to locate accurately.

In summary, reasons for the aforementioned problem are: (1) *The unreasonable connection in FA modules would cause performance degradation*, (2) *Features from different layers should be enhanced by different operations.*

4 METHODOLOGY

The framework of our ASFD is based on the simple and effective RetinaNet [17], which contains three main components: the backbone for extracting pyramid features, the neck for fusing and enhancing the features, and the head for regression and classification. Our goal is to discover a *better neck architecture* for RetinaNet and scale the ASFD to satisfy different complexity requirements automatically.

4.1 Search Space of AutoFA and AutoFE

4.1.1 AutoFA. In order to address the limitation of previous NAS-based FPN [10, 36, 37] when applied on the face domain, the above analysis motivates us to design a module that aggregates a feature by the similar-scale features instead of directly using those with large differences in scale.

To this end, we propose a fundamental building cell of AutoFA for aggregating the pyramid features sequentially, shown in Fig. 3. Initially, the cell contains a pyramid feature pool with a specific one activated, and a candidate feature pool with aggregated features of previous steps. During the searching phase, the specific pyramid feature is selected sequentially, candidate features are chosen with the corresponding probability $\boldsymbol{\alpha}$. Firstly, these candidate features are aggregated together after resizing to the same shape and weighting by $\boldsymbol{\alpha}$; then, it is fused with the pyramid feature and a convolution layer is performed to obtain the corresponding aggregated feature. At last, it is appended to the candidate feature pool for the later feature fusion. Assume that a pyramid feature and the corresponding aggregated feature are $\mathbf{F}_i$ and $\mathbf{C}_i$, the basic cell can be formulated as,

$$\mathbf{C}_i = f_{post}\left(\beta_0 \mathbf{F}_i + \beta_1 f_{pre}\left(\sum\nolimits_{j<i} \alpha_j f_{re}(\mathbf{C}_j)\right)\right), \tag{1}$$

in which $\sum_{j<i} \alpha_j = 1$, $f_{post}(\cdot)$ and $f_{pre}(\cdot)$ are two convolution operations for feature aggregation, and $f_{re}(\cdot)$ is for resizing the feature to the same size, *i.e.* bilinear interpolation for upsampling and maxpooling with stride 2 for downsampling. Once the searching process is done, the final discrete structure can be obtained

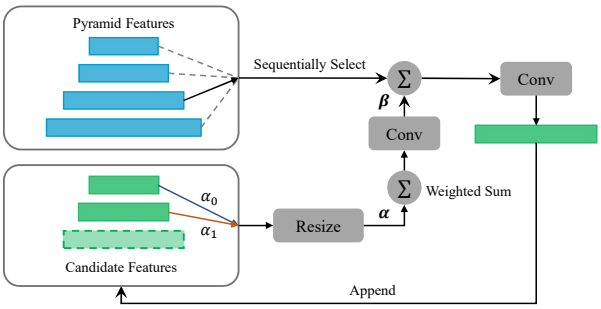


Figure 3: Illustration of the basic cell of AutoFA. $\sum$ means the sum weighted by a factor. $\boldsymbol{\alpha}$ indicates the probability to choose a candidate feature, and $\boldsymbol{\beta}$ is the score for weighting the importance of different features.

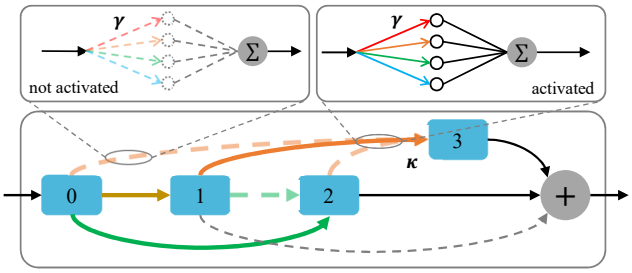


Figure 4: Illustration of the basic structure of AutoFE with 4 nodes. The bold colored arrows have two states: not activated and activated, in which not activated arrows mean disconnecting, $\boldsymbol{\kappa}$ indicates the probability. Thin colored arrows indicate different operations with probability $\boldsymbol{\gamma}$.

according to the probability score $\boldsymbol{\alpha}$ and importance score $\boldsymbol{\beta}$. For a given pyramid feature, it is aggregated by candidate features with probability $\alpha_i \geq 0.5$, and $\boldsymbol{\beta}$ is retained as the initial value for weighting the pyramid feature and candidate feature. In this approach, aggregated features of the discrete cell can be denoted as,

$$\begin{aligned} \mathbf{T}_i &= f_{pre}\left(\sum\nolimits_{j<i} [\alpha_j \geq 0.5] \cdot f_{re}(\mathbf{C}_j)\right), \\ \mathbf{C}_i &= f_{post}\left(\beta_0 \cdot \mathbf{F}_i + \beta_1 \cdot \mathbf{T}_i\right), \end{aligned} \tag{2}$$

where $[\,\cdot\,]$ equals 1 if the inner expression is true.

Similar to PAFPN [21] and BiFPN [32], our AutoFA aggregates the pyramid features along a top-down path and a bottom-up path, each of them is comprised of several basic building cells. For the top-down path, pyramid features are selected in the order of decreasing resolution for aggregation, *i.e.* from P7 with stride 128 to P2 with stride 4, same as Fig. 3. As the counterpart, the aggregation along bottom-up path is in the reversed order.

4.1.2 AutoFE.. The incompatibility of those FE modules for some pyramid levels has been revealed and an important conclusion has been drawn in the aforementioned analysis. To discover the suitable enhancement module for each pyramid layer, we propose a basic cell for our AutoFE that includes several intermediate features transformed by the candidate operations, which include {1 × 1 conv,

1×3 conv, 3×1 conv, 3×3 conv, 1×5 conv, 5×1 conv, 5×5 conv }. As presented in Fig. 4, the basic cell is conducted as a directed acyclic graph with several nodes, where node 0 is input and others are intermediate features. Each node i is connected to the previous node $j < i$ with two status indicated by κ_{ji}, *i.e.* activated if and only if κ_{ji} is maximum among κ_{*i}, otherwise not activated. In this way, the previous feature is transformed by the different operations; otherwise, it is not activated. Assume that the feature of ith node is $\mathbf{F}_i$, it can be formulated as follow,

$$\mathbf{F}_i = \sum_{j<i} [j = \arg\max_j \kappa_{ji}] \cdot f_{op}(\mathbf{F}_j, \gamma_{ji}), \tag{3}$$

where $f_{op}(\cdot)$ is the sum weighted by γ_{ji} when processed by the activated operations. Different from [19, 38], the output of the cell is the sum of features of all leaf nodes, *i.e.* the intermediate features who are not input to the other nodes, given by

$$\mathbf{F}_{out} = \sum_i \left[\sum_{k>i} [i = \arg\max_i \kappa_{ik}] = 0 \right] \cdot \mathbf{F}_i. \tag{4}$$

In particular, the commonly used convolutions with different kernel shapes and dilation rates are adopted for $f_{op}(\cdot)$.

However, during the search, Eq. 3 cannot be optimized because it is equivalent to discrete sampling, which is not differentiable. To allow back-propagation, we use the Gumbel-Max method [9] to re-formulate Eq. 3 in an efficient way that samples a discrete probability as follow,

$$\begin{aligned} \mathbf{F}_i &= \sum_{j<i} h_{ji} \cdot f_{op}(\mathbf{F}_j, \gamma_{ji}), \\ \text{s.t.} \quad h_{ji} &= \text{onehot}(\arg\max_j(\kappa_{ji} + o_{ji})), \end{aligned} \tag{5}$$

where o_{ji} is the *i.i.d.* sample drawn from Gumbel$(0, 1)$ [9]. Then, softmax function is used to relax the argmax function so as to make Eq. 5 being differentiable, in which $\tilde{h}_{ji}$ is for approximating h_{ji}, denoted by,

$$\tilde{h}_{ji} = \frac{\exp(\kappa_{ji} + o_{ji}/\tau)}{\sum_{j'<i} \exp(\kappa_{j'i} + o_{j'i}/\tau)}, \tag{6}$$

where τ is the softmax temperature. In this way, argmax is used in the forward pass to achieve discrete sampling of connections between two nodes, but softmax in Eq. 6 is adopted during backward pass to allow gradient back-propagation.

Finally, the discrete architecture of AutoFE is obtained by retaining the connections and operations among intermediate features according to the maximum of $\boldsymbol{\kappa}$ and $\boldsymbol{\gamma}$.

4.2 Search Strategy of AutoFA and AutoFE

We have transformed the discrete network structure into several architecture parameters through the design of face-suitable search space for AutoFA and AutoFE. In detail, $\boldsymbol{\alpha}$ is adopted to make the decision on choosing candidate features for a pyramid feature, $\boldsymbol{\beta}$ is used to balance the importance of pyramid and candidate features. For AutoFE, $\boldsymbol{\kappa}$ is employed for selecting the connection of intermediate nodes, and $\boldsymbol{\gamma}$ indicates the probability of different operations. Similar to [3, 19, 38], we utilize the bi-level optimization method to alternately optimize the network parameters, *e.g.* parameters of convolution layers, and architecture parameters in an end-to-end manner.

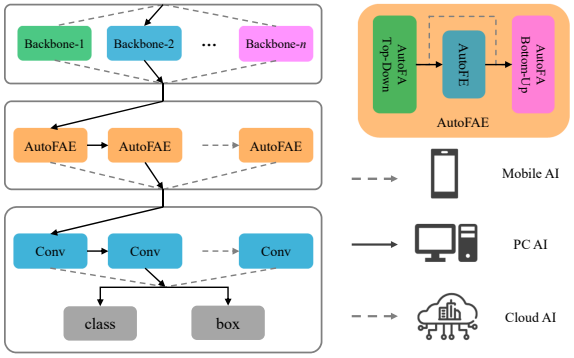


Figure 5: The architecture of the supernet to automatically obtain a detector for different AI systems.

4.3 Auto Model Scaling

We automatically obtain the ASFD family with different complexities on the basis of a supernet, as shown in Fig. 5, which is comprised of the backbones in parallel, the stacked AutoFAE modules, and the stacked convolutions for prediction head. Our method aims to search for a better composition to meet different complexity requirements, *i.e.* which backbone to pick, how many AutoFAE modules and convolutions in head to stack, whether to skip AutoFE for each AutoFAE, and what the number of feature channels.

4.3.1 Training. Based on the idea of weight sharing, the supernet is trained by alternately training a single-path network through uniformly sampling [7, 11]. For instance, as presented in Fig. 5, the single-path is composed of backbone-2, and AutoFAE and prediction convolutions, which are both stacked two layers. Furthermore, a scalable method is proposed for training the supernet compatible with different feature channels. The supernet is optimized with the maximal feature channels during a long warm-up period until it tends to converge. Then, the candidate feature channels are gradually added to be sampled in the descending order, in which the corresponding tensors are sliced out along each dimension to fit the calculations.

4.3.2 Searching. The searching phase is based on the genetic algorithm [4, 7, 11] and directly takes inference latency into fitness. At first, populations are randomly initialized with genes encoded by the 5 degrees of freedom of the supernet, which would be removed if against the constraints. After the initialization, they are evaluated on a mini validation set to obtain the fitness. At each iteration, only the top-k populations with better finesses are retained to generate the next generation by mutation and crossover. By repeating this procedure several times, we can discover a single-path network with the best fitness.

5 EXPERIMENTS

5.1 Experimental Setup

5.1.1 Baseline. If not specified, RetinaNet [17] with FPN is utilized as the baseline of the face detector. Compared to the original generic object detection application, it has the following differences: (1) 6 levels of pyramid features are used for predicting with anchor scales

Module	$AP_{.50}$			AP		
	Easy	Medium	Hard	Easy	Medium	Hard
Baseline	95.1	94.0	87.2	61.9	59.2	46.5
NAS-FPN	95.1	93.9	86.2	61.8	59.0	45.7
NAS-FCOS	94.7	93.1	85.6	61.5	58.1	45.1
AutoFPN	94.6	93.4	86.0	61.4	58.5	45.6
PAFPN	95.3	94.1	87.3	62.1	59.4	46.8
BiFPN	95.5	94.4	87.4	62.3	59.6	46.9
ABiFPN	95.3	94.5	87.5	62.2	59.7	47.0
FEM-FPN	95.2	94.0	86.7	62.1	59.4	46.5
DARTS	95.1	93.5	86.5	61.8	58.6	45.6
PC-DARTS	95.0	93.7	86.6	61.8	58.9	46.0
AutoFA	95.4	94.4	**87.8**	**62.8**	**60.2**	**47.4**

Table 3: Comparison with state-of-the-art feature aggregation modules on WIDER Face validation.

Module	$AP_{.50}$			AP		
	Easy	Medium	Hard	Easy	Medium	Hard
Baseline	94.7	92.8	82.9	61.7	58.4	45.0
ASPP	94.8	93.0	83.4	62.1	58.8	45.3
RFB	94.5	92.7	83.0	61.5	58.5	45.2
CPM	94.6	92.8	83.0	61.5	58.4	45.2
FEM-CPM	94.5	92.9	83.3	61.9	58.7	45.5
RFE	94.5	92.8	83.2	61.8	58.7	45.4
DARTS	94.7	92.9	83.0	61.6	58.6	45.1
PC-DARTS	94.6	93.0	83.0	61.8	58.6	45.2
AutoFE	**95.2**	**93.2**	**83.5**	**62.1**	**59.0**	**45.8**

Table 4: Comparison with state-of-the-art feature enhancement modules on WIDER Face validation.

$\{4, 8, 16, 32, 64, 128\}$ and aspect ratio 1:1.5. (2) The IoU threshold for anchor matching is changed to 0.4 and the ignore-zone is not implemented. (3) Top-2000 predictions with confidence higher than 0.05 are processed by non-maximum suppression with a threshold 0.4 to produce at most 750 final detections. The results are reported using $AP_{.50}$ measured with a constant IoU threshold 0.5, as well as AP averaged under IoU thresholds from 0.5 to 0.95 with step 0.05 to demonstrate the performance at high IoU.

5.1.2 Train Details. We use the ImageNet-pretrained models to initialize the backbone parameters, and 'kaiming' method for others. SGD algorithm is employed to optimize the network parameters with momentum 0.9, weight decay 5×10^{-4} and initial learning rate 0.01 per 32 images. For ablative studies, the learning rate is multiplied by factor 0.1 at 30, 40 epochs and ended at 50 epochs. For the main results, it is divided by 10 at 60, 100 epochs and ended at 120 epochs.

5.1.3 Search Details. The training set of WIDER Face is divided into two mini training and a validating subsets, with a ratio of $9:9:2$, they are used for updating network and architecture parameters, and evaluating the searched modules respectively. Adam algorithm with learning rate 0.01 is adopted for optimizing the architecture parameters, which are frozen at the first 50 epochs and updated during $50\sim100$ epochs, and other settings are same as training details. To determine the final AutoFAE module, we run the searching algorithm 3 times with different random seeds and pick the best one based on its performance on the mini validation. All training and searching experiments are conducted on 8 V100 GPUs. The AutoFA and AutoFE can be searched within 3 to 4 hours. The commonly used supernet takes about 12 hours for training, and the ASFD families could be sampled within 1.5 to 4 hours.

5.2 Ablation Study

5.2.1 Effect of Search Space for AutoFA and AutoFE. To demonstrate the effectiveness of our proposed face-suitable search space for feature aggregation and enhancement modules, the AutoFA and AutoFE modules are discovered and compared to the state-of-the-art modules respectively.

At the first stage, the AutoFA module is searched through a RetinaNet that replaces the FPN with several basic aggregation modules. As shown in Table 3, simulations are conducted by comparing to the commonly used FA modules, in which DARTS [19] and PC-DARTS [38] illustrate the results based on a fully-connected search space [36]. Our AutoFA manages to address the limitations of previous NAS-based methods and outperforms them with a large margin, which is more than 1.0 points on all three subsets indicated by AP. Besides, it is also significantly better than the hand-crafted ones composed of top-down and bottom-up paths, *i.e.* PAFPN [21], BiFPN [32], and ABiFPN [41], demonstrating the superiority of connections between the multi-scale features of AutoFA. Such the large improvement is mainly from predictions with the high IoU, which shows that the features of different scales are fully aggregated and it is helpful for more distinguishable classification and more accurate location.

Then, RetinaNet without FPN is adopted as the baseline to better highlight the effectiveness of FE modules. Different FE modules are placed between the backbone and detection head to refine the multi-scale features, as shown in Table 4. In particular, DARTS and PC-DARTS discover FE modules by following their original settings in image classification. However, they only improve the baseline by minor advantages. With the specified face-suitable search space, the found AutoFE improves the baseline by 0.5/0.4/0.6 points of $AP_{.50}$ and 0.4/0.6/0.8 points of AP, far exceeding the other state-of-the-art modules and demonstrating the superiority of our face-suitable search space.

5.2.2 Effect of Joint Searching AutoFAE. The AutoFAE module is composed of AutoFA and AutoFE two modules, which can be obtained by cascading the discovered AutoFA and AutoFE modules or jointly searching in an end-to-end manner. As presented in Table 5, only a minor improvement is achieved by cascading the discovered AutoFA and AutoFE directly. And AutoFAE found by the joint searching way can further improve $AP_{.50}$ and AP by clear margins, demonstrating the state-of-the-art performance of proposed AutoFAE.

5.2.3 Effect of Different Positions of AutoFE. Review that our AutoFAE is built upon the top-down and bottom-up paths. Therefore, we have three ways to build the final AutoFAE module. In detail, the

Method	$AP_{.50}$			AP		
	Easy	Medium	Hard	Easy	Medium	Hard
Baseline	95.1	94.0	87.2	61.9	59.2	46.5
AutoFA+AutoFE	95.4	94.5	87.9	62.8	60.2	47.5
Joint Search	**95.7**	**95.0**	**88.6**	**62.9**	**60.5**	**47.8**

Table 5: The effect of searching method for the AutoFAE.

Position	$AP_{.50}$			AP		
	Easy	Medium	Hard	Easy	Medium	Hard
Baseline	95.1	94.0	87.2	61.9	59.2	46.5
Before	95.2	94.3	87.5	62.4	60.1	46.9
Middle	**95.7**	**95.0**	**88.6**	**62.9**	**60.5**	**47.8**
After	95.3	94.4	88.0	62.4	60.2	47.3

Table 6: The effect of the position of AutoFE and AutoFA.

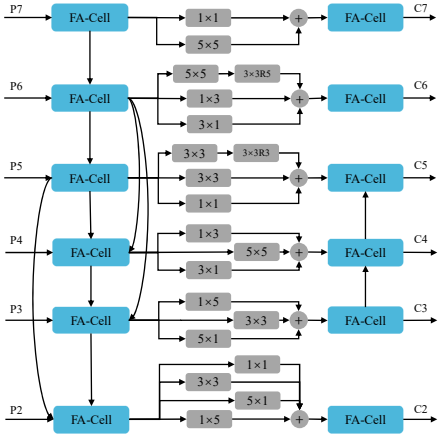


Figure 6: The architecture of the discovered AutoFAE, in which FA-Cell is the basic cell indicated by Fig. 3, $m \times n$ denotes the convolution kernel size, and Rx is the dilated rate.

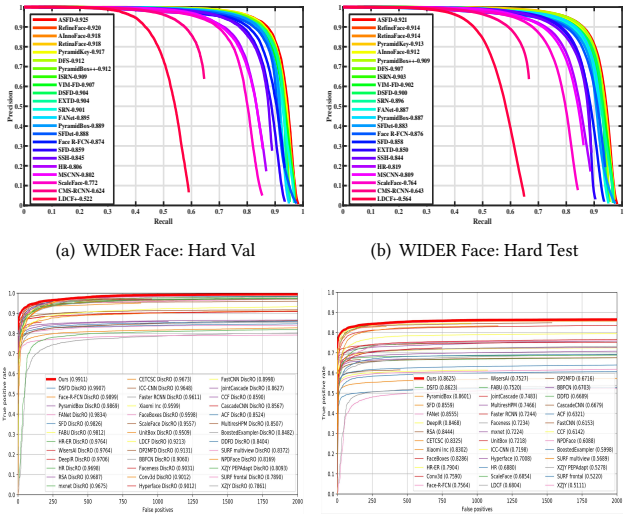
AutoFE module can be plugged before and after the AutoFA, as well as between the top-down and bottom-up paths. In this way, three modules are obtained by utilizing the joint searching method. As shown in Table 6, we observe that the best performance is achieved when AutoFE is in the middle position. This is mainly because similar presentation is generated after the top-down aggregation. Placing AutoFE before the bottom-up path can further enhance these features to carry different context information.

5.3 Analysis on AutoFAE

We visualize the architecture of AutoFAE in Fig. 6, which can match the previous conclusions drawn in the problem analysis section perfectly. In general, the AutoFA module aggregates pyramid features along with the sparse cross-scale and similar-scale connections instead of a fully connected manner like [36, 37], which avoids the performance degradation caused by large scale differences. Most of these cross-scale connections appear on the top-down path of AutoFA, in which the shallow features that lack semantic information

Model	Single Path	$AP_{.50}$			AP			Lat.
		Easy	Medium	Hard	Easy	Medium	Hard	
D0	R18-FA-H×1-64	95.7	94.8	88.0	63.7	61.1	48.3	3.1
D1	R18-FA-H×3-128	96.1	95.2	88.8	64.1	61.5	48.9	5.7
D2	R34-FA-H×3-192	96.4	95.6	89.5	64.6	62.3	49.6	10.5
D3	R50-FAE-H×3-192	96.6	95.9	90.5	65.1	62.8	50.4	16.6
D4	R50-FAE-H×4-256	97.0	96.3	91.2	65.8	63.3	50.9	26.2
D5	R101-FAE-H×4-256	97.0	96.5	91.9	65.9	63.3	51.7	29.6
D6	R101-FAE-FA-FA-H×4-256	97.2	96.5	92.5	66.2	63.5	52.3	36.1

Table 7: The family of ASFD, where latency (ms) is measured with VGA-resolution images and on Nvidia V100 GPU.



(a) WIDER Face: Hard Val (b) WIDER Face: Hard Test

(c) FDDB: Discontinuous (d) FDDB: Continuous

Figure 7: Evaluation on the popular benchmarks of ASFD.

are aggregated with not only the adjacent layer but also the others with rich context. Besides, AutoFE modules with different operations and topological structures are found for different pyramid layers. Particularly, dilated convolutions only appear in the later levels for enlarging the receptive fields, the others are almost rectangle convolutions for more diverse features. Thus, the large faces are located, and small faces in occlusion and with extreme-poses are well distinguished.

5.4 Model Scaling

Next, the supernet is trained on the basis of the final AutoFAE and backbone networks of ResNet series [12]. Then, the genetic algorithm is adopted to search the single-path networks with 50 populations and 50 iterations. We discover 7 single-path networks under the different GPU inference latencies, *e.g.* 5ms, 10ms and so on. These networks are trained for 150 epochs with the commonly used pyramid anchors [33], and multi-scale test is employed with factors 0.5, 1.0, 1.5, 2.0. The detailed results are presented in Table 7, in which the network architecture is indicated by single path. For instance, "R101-FAE-FA-FA-H×4-256" means ResNet101 is adopted as the backbone, AutoFAE modules are stacked 3 times and AutoFE module is skipped within the last two modules, convolution layers are repeated 4 times in prediction head, and the feature channel is 256. Obviously, our ASFD family makes a better trade-off between

Figure 8: Illustration of ASFD to various large variations. Red bounding boxes indicate the detection confidence is above 0.8.

performance and efficiency by scaling the components and channels, especially the ASFD-D0 costs about 3.1 ms, *i.e.* more than 320 FPS.

5.5 Evaluation on Benchmarks

We evaluate our ASFD-D6 on the popular benchmarks, *i.e.* WIDER Face [39] and FDDB [14], which is trained only on the training set of WIDER Face and test on these benchmarks without any fine-tuning. Our ASFD-D6 obtains the highest $AP_{.50}$ scores with 97.2/96.5/92.5 on WIDER Face validation, 96.7/96.2/92.1 on WIDER Face test, and 99.11 and 86.25 on FDDB discontinuous and continuous curves, outperforming the prior competitors by a considerable margin and setting a new state-of-the-art face detector, shown as Fig. 7 (Easy and Medium results of WIDER Face are ignored due to the space limitation). More examples of our ASFD on handling face with various variations are shown in Fig. 8 to demonstrate its effectiveness.

5.6 Generalization on Generic Object Detection

To demonstrate the generalization ability of our AutoFAE module, we evaluate the final AutoFAE module with three typical detectors, RetinaNet [17], FCOS [34] and Faster RCNN [26] on COCO. In particular, the original FPN module is replaced with our AutoFAE by connecting the corresponding pyramid layers, as presented in Table 8, our AutoFAE module can consistently adapt to the general object domain and different detectors, with AP improvements from 0.5 to 1.0 points.

Model	FPN			AutoFAE		
	AP	$AP_{.50}$	$AP_{.75}$	AP	$AP_{.50}$	$AP_{.75}$
RetinaNet	36.5	55.1	39.0	**37.5**	**56.6**	**39.9**
FCOS	38.6	57.2	41.7	**39.2**	**57.6**	**42.2**
Faster RCNN	37.4	58.1	40.4	**37.9**	**58.3**	**41.2**

Table 8: The generalization of AutoFAE on generic object detection dataset *i.e.* COCO.

6 CONCLUSION

Neural architecture search has demonstrated its successes in generic object detection about feature aggregation and enhancement. However, they cannot adapt to the domain difference between face and generic object detection and cause severe performance drops when applied to the face domain. In this paper, we analyze the reason for this phenomenon occurs and propose a face-suitable search space for feature aggregation and enhancement modules. And a better FAE module termed as AutoFAE is discovered using bi-level optimization, which outperforms the current state-of-the-art FAE modules in face detection and can be generalized to general object tasks. Finally, we automatically obtain a family of detectors with different complexities based on a supernet that achieves a better performance-efficiency trade-off.

REFERENCES

[1] Han Cai, Chuang Gan, and Song Han. 2020. Once for all: Train one network and specialize it for efficient deployment. In *ICLR*.

[2] Liang-Chieh Chen, George Papandreou, Florian Schroff, and Hartwig Adam. 2017. Rethinking atrous convolution for semantic image segmentation. *arXiv* (2017).

[3] Xin Chen, Lingxi Xie, Jun Wu, and Qi Tian. 2019. Progressive differentiable architecture search: Bridging the depth gap between search and evaluation. In *ICCV*. 1294–1303.

[4] Yukang Chen, Tong Yang, Xiangyu Zhang, Gaofeng Meng, Xinyu Xiao, and Jian Sun. 2019. DetNAS: Backbone search for object detection. In *NIPS*. 6642–6652.

[5] Cheng Chi, Shifeng Zhang, Junliang Xing, Zhen Lei, Stan Z Li, and Xudong Zou. 2019. Selective refinement network for high performance face detection. In *AAAI*, Vol. 33. 8231–8238.

[6] Cheng Chi, Shifeng Zhang, Junliang Xing, Zhen Lei, Stan Z Li, and Xudong Zou. 2019. Selective refinement network for high performance face detection. In *AAAI*, Vol. 33. 8231–8238.

[7] Xiangxiang Chu, Bo Zhang, Ruijun Xu, and Jixiang Li. 2019. Fairnas: Rethinking evaluation fairness of weight sharing neural architecture search. *arXiv* (2019).

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

[9] Xuanyi Dong and Yi Yang. 2019. Searching for a robust neural architecture in four gpu hours. In *CVPR*. 1761–1770.

[10] Golnaz Ghiasi, Tsung-Yi Lin, and Quoc V Le. 2019. NAS-FPN: Learning scalable feature pyramid architecture for object detection. In *CVPR*. 7036–7045.

[11] Zichao Guo, Xiangyu Zhang, Haoyuan Mu, Wen Heng, Zechun Liu, Yichen Wei, and Jian Sun. 2020. Single path one-shot neural architecture search with uniform sampling. *ECCV*.

[12] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *CVPR*. 770–778.

[13] Yuge Huang, Yuhan Wang, Ying Tai, Xiaoming Liu, Pengcheng Shen, Shaoxin Li, Jilin Li, and Feiyue Huang. 2020. Curricularface: adaptive curriculum learning loss for deep face recognition. In *CVPR*. 5901–5910.

[14] Vidit Jain and Erik Learned-Miller. 2010. *FDDB: A benchmark for face detection in unconstrained settings.* Technical Report. UMass Amherst technical report.

[15] Jian Li, Yabiao Wang, Changan Wang, Ying Tai, Jianjun Qian, Jian Yang, Chengjie Wang, Jilin Li, and Feiyue Huang. 2019. DSFD: Dual shot face detector. In *CVPR*. 5060–5069.

[16] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. 2017. Feature pyramid networks for object detection. In *CVPR*. 2117–2125.

[17] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. 2017. Focal loss for dense object detection. In *ICCV*. 2980–2988.

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

[19] Hanxiao Liu, Karen Simonyan, and Yiming Yang. 2019. Darts: Differentiable architecture search. *ICLR*.

[20] Songtao Liu, Di Huang, et al. 2018. Receptive field block net for accurate and fast object detection. In *ECCV*. 385–400.

[21] Shu Liu, Lu Qi, Haifang Qin, Jianping Shi, and Jiaya Jia. 2018. Path aggregation network for instance segmentation. In *CVPR*. 8759–8768.

[22] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. 2016. SSD: Single shot multibox detector. In *ECCV*. Springer, 21–37.

[23] Yang Liu and Xu Tang. 2020. BFBox: Searching Face-Appropriate Backbone and Feature Pyramid Network for Face Detector. In *CVPR*. 13568–13577.

[24] Hongyu Pan, Hu Han, Shiguang Shan, and Xilin Chen. 2018. Mean-variance loss for deep age estimation from a face. In *CVPR*. 5285–5294.

[25] Siyuan Qiao, Liang-Chieh Chen, and Alan Yuille. 2020. DetectoRS: Detecting Objects with Recursive Feature Pyramid and Switchable Atrous Convolution. *arXiv* (2020).

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

[27] Karen Simonyan and Andrew Zisserman. 2014. Very deep convolutional networks for large-scale image recognition. *arXiv* (2014).

[28] Christian Szegedy, Sergey Ioffe, Vincent Vanhoucke, and Alexander A Alemi. 2017. Inception-v4, inception-resnet and the impact of residual connections on learning. In *AAAI*.

[29] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. 2016. Rethinking the inception architecture for computer vision. In *CVPR*. 2818–2826.

[30] Ying Tai, Yicong Liang, Xiaoming Liu, Lei Duan, Jilin Li, Chengjie Wang, Feiyue Huang, and Yu Chen. 2019. Towards highly accurate and stable face alignment for high-resolution videos. In *AAAI*, Vol. 33. 8893–8900.

[31] Mingxing Tan and Quoc V Le. 2019. Efficientnet: Rethinking model scaling for convolutional neural networks. *arXiv* (2019).

[32] Mingxing Tan, Ruoming Pang, and Quoc V Le. 2020. Efficientdet: Scalable and efficient object detection. In *CVPR*. 10781–10790.

[33] Xu Tang, Daniel K Du, Zeqiang He, and Jingtuo Liu. 2018. Pyramidbox: A context-assisted single shot face detector. In *ECCV*. 797–813.

[34] Zhi Tian, Chunhua Shen, Hao Chen, and Tong He. 2019. Fcos: Fully convolutional one-stage object detection. In *Proceedings of the IEEE international conference on computer vision*. 9627–9636.

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

[36] Ning Wang, Yang Gao, Hao Chen, Peng Wang, Zhi Tian, and Chunhua Shen. 2019. NAS-FCOS: Fast neural architecture search for object detection. *arXiv preprint arXiv:1906.04423* (2019).

[37] Hang Xu, Lewei Yao, Wei Zhang, Xiaodan Liang, and Zhenguo Li. 2019. Auto-FPN: Automatic network architecture adaptation for object detection beyond classification. In *ICCV*. 6649–6658.

[38] Yuhui Xu, Lingxi Xie, Xiaopeng Zhang, Xin Chen, Guo-Jun Qi, Qi Tian, and Hongkai Xiong. 2020. PC-darts: Partial channel connections for memory-efficient differentiable architecture search. *ICLR* (2020).

[39] Shuo Yang, Ping Luo, Chen-Change Loy, and Xiaoou Tang. 2016. Wider face: A face detection benchmark. In *CVPR*. 5525–5533.

[40] YoungJoon Yoo, Dongyoon Han, and Sangdoo Yun. 2019. EXTD: Extremely tiny face detector via iterative filter reuse. *arXiv* (2019).

[41] Bin Zhang, Jian Li, Yabiao Wang, Zhipeng Cui, Yili Xia, Chengjie Wang, Jilin Li, and Feiyue Huang. 2020. ACFD: Asymmetric Cartoon Face Detector. *arXiv* (2020).

[42] Shifeng Zhang, Cheng Chi, Zhen Lei, and Stan Z Li. 2020. RefineFace: Refinement neural network for high performance face detection. *IEEE TPAMI* (2020).

[43] Shifeng Zhang, Xiangyu Zhu, Zhen Lei, Hailin Shi, Xiaobo Wang, and Stan Z Li. 2017. S3FD: Single shot scale-invariant face detector. In *ICCV*. 192–201.

[44] Barret Zoph and Quoc V Le. 2017. Neural architecture search with reinforcement learning. *ICLR*.

[45] Barret Zoph, Vijay Vasudevan, Jonathon Shlens, and Quoc V Le. 2018. Learning transferable architectures for scalable image recognition. *CVPR*, 8697–8710.