TinaFace: Strong but Simple Baseline for Face Detection

Yanjia Zhu*, Hongxiang Cai*, Shuhan Zhang†, Chenhao Wang†, Yichao Xiong‡
Media Intelligence Technology Co.,Ltd
{yanjia.zhu, hongxiang.cai, shuhan.zhang, chenhao.wang, yichao.xiong}@media-smart.cn

Abstract

Face detection has received intensive attention in recent years. Many works present lots of special methods for face detection from different perspectives like model architecture, data augmentation, label assignment and etc., which make the overall algorithm and system become more and more complex. In this paper, we point out that ***there is no gap between face detection and generic object detection****. Then we provide a strong but simple baseline method to deal with face detection named TinaFace. We use ResNet-50 [11] as backbone, and all modules and techniques in TinaFace are constructed on existing modules, easily implemented and based on generic object detection. On the hard test set of the most popular and challenging face detection benchmark WIDER FACE [48], with single-model and single-scale, our TinaFace achieves 92.1% average precision (AP), which exceeds most of the recent face detectors with larger backbone. And after using test time augmentation (TTA), our TinaFace outperforms the current state-of-the-art method and achieves 92.4% AP. The code is available at* `https://github.com/Media-Smart/vedadet/tree/main/configs/trainval/tinaface`.

1. Introduction

Face detection becomes a very important task in computer vision, since it is the first and fundamental step of most tasks and applications about faces, such as face recognition, verification, tracking, alignment, expression analysis etc.. Therefore, so many methods are presented in this field from different perspectives recently. Some works [6, 7, 49] introduce annotated landmarks information as extra supervision signal, and some of others [51, 57, 37, 17, 26, 25, 58] pay more attention to the design of network. Besides, some new loss designs [51, 57, 16] and data augmentation methods [17, 37] are presented. What's more, a few works [23, 58] begin to redesign the matching strategy and label assignment process. Obviously, face detection seems to be gradually separated out from generic object detection and forms a new field.

Intuitively, face detection is actually an application of generic object detection. To some degree, face is an object. So naturally there are a series of questions to be asked, *"what is the difference between face detection and generic object detection?"*, *"Why not using generic object detection techniques to deal with face detection?"*, and *"is it necessary to additionally design special methods for handling face detection?"*.

First, from the perspective of data, the properties that faces own also exist in objects, like pose, scale, occlusion, illumination, blur and etc.. And the unique properties in faces like expression and makeup can also correspond to distortion and color in objects. Then from the perspective of challenges encountered by face detection like multi-scale, small faces and dense scenes, they all exist in generic object detection. Thus, face detection seems to be just a subproblem of generic object detection. To better and further answer above questions, we provide a simple baseline method based on generic object detection to outperform the current state-of-the-art methods on the hard test set of WIDER FACE [48].

The main contributions of this work can be summarized as:

- Indicating that face detection is actually a one class generic object detection problem and can be handled by techniques in generic object detection.
- Providing a strong but simple baseline method for face detection named TinaFace. All ideas and modules used in TinaFace are based on generic object detection.
- With single-scale and single-model, we achieve 92.1% average precision(AP) in hard settings on the test subset of WIDER FACE, which already exceed most of recent methods with larger backbone and Test Time Augmentation (TTA). Our final model gets 92.4% AP in hard settings on the test subset and outperforms current state-of-the-art methods for face detection.

*Equal contribution.
†Data analysis.
‡Corresponding author.

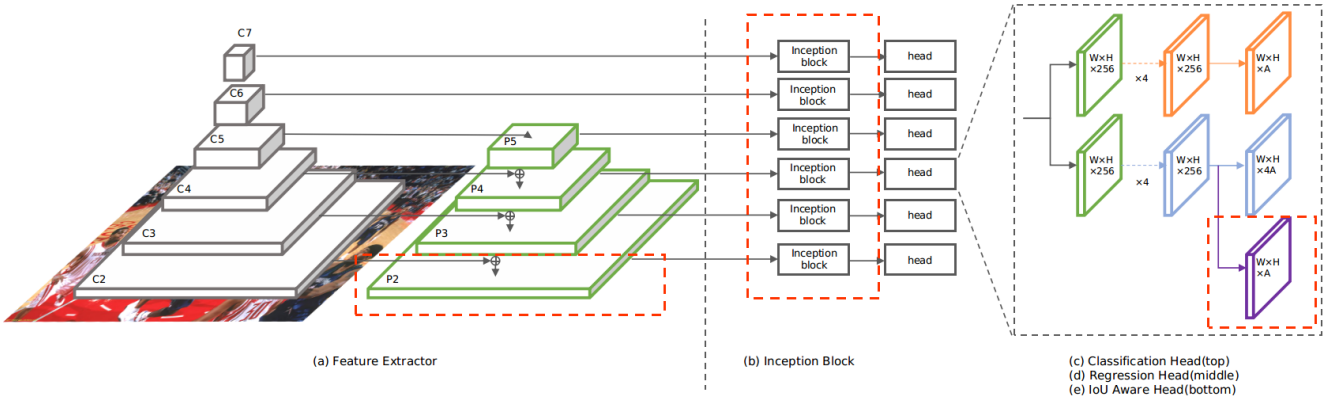


Figure 1: The model architecture of TinaFace. (a) Feature Extractor: ResNet-50 [11] and 6 level Feature Pyramid Network [18] to extract the multi-scale features of input image. (b) Inception block to enhance receptive field. (c) Classification Head: 5 layers FCN for classification of anchors. (d) Regression Head: 5 layers FCN for regression of anchors to ground-truth objects boxes. (e) IoU Aware Head: a single convolutional layer for IoU prediction.

2. Related Work

Generic Object Detection. Generic object detection aims at locating and classifying the existing objects in the given picture. Before the booming of deep learning, generic object detection is mainly based on the hand-crafted feature descriptors like SIFT [24] and HOG [5]. And the most successful methods like DPM [8] combine multi-scale hand-crafted features, sliding window, deformable part and SVM classifier to form a generic object detector.

With AlexNet [15] winning the championship of Large Scale Visual Recognition Challenge 2012 (ILSVRC2012) by a large gap, the era of deep learning is coming, and generic object detection has been quickly dominated by deep learning methods. Two-stage methods start from R-CNN [10] and Fast R-CNN [9]. And soon Faster R-CNN [31] proposes RPN network to replace the selective search to generate proposals by pre-define anchors, which becomes the most classical anchor-based generic object detection method. Based on Faster R-CNN [31], there are so many new methods presented like FPN [18], Mask R-CNN [12], Cascade R-CNN [1] and etc.. In order to overcome the high latency of two-stage methods, many one-stage methods are presented like series of YOLO [30, 28, 29], SSD [22] and RetinaNet [19]. To handling the multiple scale or small objects problem, YOLOs [30, 28, 29] present novel anchor matching strategy including consideration of feedback of proposals and one ground-truth vs. one anchor, and also reweight the regression of width and height of objects. Then SSD [22] uses a hierarchy of backbone features, while FPN [18] presents feature pyramids. Besides, the series of SNIP [34] and SNIPER [35], multi-scale training and multi-scale testing can also deal with the multiple scale problem.

In addition to the new method proposed in generic object detection, developments in other fields, like normalization methods and deep convolutional networks, also promote generic object detection. Batch normalization (BN) [14] normalizes features within a batch along channel dimension, which can help models converge and enable models to train. In order to handle the dependency with batch size of BN, group normalization (GN) [44] divides the channels into groups and computes within each group the mean and variance for normalization. Then for deep convolutional networks, after AlexNet [15], VGG [33] increases depth using an architecture with very small 3×3 convolution filters, GoogLeNet [36] introduces Inception modules to use different numbers of small filters in parallel to form features of different receptive fields and help model to capture objects as well as context at multiple scales, and ResNet [11] demonstrates the importance of the original information flow and presents skip connection to handle the degradation with deeper networks.

Face Detection. As an application of generic object detection, the history of face detection is almost the same. Before the era of deep learning, face detectors are also based on hand-crafted features like Haar [39]. After the most popular and challenging face detection benchmark WIDER FACE dataset [48] presented, face detection develops rapidly focusing on the extreme and real variation problem including scale, pose, occlusion, expression, makeup, illumination, blur and etc.. Almost all the recent face detection methods evolve from the existing generic object detection methods. Based on SSD [22], S^3FD [58] extends anchor-associated layers to C3 stage and proposes a scale compensation anchor matching strategy in order to cover the small faces, PyramidBox [37] proposes PyramidAnchors (PA), Low-level

Feature Pyramid Networks (LFPN), Context-sensitive Predict Module (CPM) to emphasize the importance of context and data-anchor-sampling augmentation to increase smaller faces, and DSFD [16] introduce a dual-shot detector using Improved Anchor Matching (IAM) and Progressive Anchor Loss (PAL). Then Based on RetinaNet [19], RetinaFace [6] manually annotates five facial landmarks on faces to serve as extra supervision signal, RefineFace [57] introduces five extra modules Selective Two-step Regression (STR), Selective Two-step Classification (STC), Scale-aware Margin Loss (SML), Feature Supervision Module (FSM) and Receptive Field Enhancement (RFE), and HAMBox [23] emphasize the strong regression ability of some unmatched anchors and present an Online High-quality Anchor Mining Strategy (HAMBox). Besides, ASFD [51] uses neural architecture search technique to automatically search the architecture for efficient multi-scale feature fusion and context enhancement.

To sum up, methods presented in face detection almost cover every part of deep learning training from data processing to loss designs. It is obvious that all of these methods focus on the challenge of small faces. However, actually there are so many methods in generic object detection, which we mention above, solving this problem. Therefore, based on some of these methods, we present TinaFace, a strong but simple baseline method for face detection.

3. TinaFace

Basically, we start from the one-stage detector RetinaNet [19] as some previous works do. The architecture of TinaFace is shown in Figure 1 where the red dashed boxes demonstrate the different parts from RetinaNet [19].

3.1. Deformable Convolution Networks

There is an inherent limitation in convolution operation, that is, we feed it with a strong prior about the sampling position which is fixed and rigid. Therefore, it is hard for networks to learn or encode complex geometric transformations, and the capability of models is limited. In order to further improve the capability of our model, we employ DCN [4] into the stage four and five of the backbone.

3.2. Inception Module

Multi-scale is always a challenge in generic object detection. The most common ways to deal with it are multi-scale training, FPN architecture and multi-scale testing. Besides, we employ inception module [36] in our model to further enhance this ability. The inception module uses different numbers of 3×3 convolutional layers in parallel to form features of different receptive fields and then combine them, which help model to capture objects as well as context at multiple scales.

3.3. IoU-aware Branch

IoU-aware [43] is an extremely simple and elegant method to relieve the mismatch problem between classification score and localization accuracy of a single-stage object detector, which can help resort the classification score and suppress the false positive detected boxes (high score but low IoU). The architecture of IoU-aware is shown in Figure 1, and the only difference is the purple part, a parallel head with a regression head to predict the IoU between the detected box and the corresponding ground-truth object. And this head only consists of a single 3×3 convolution layer, followed by a sigmoid activation layer. At the inference phase, the final detection confidence is computed by following equation,

$$score = p_i^{\alpha} IoU_i^{(1-\alpha)} \tag{1}$$

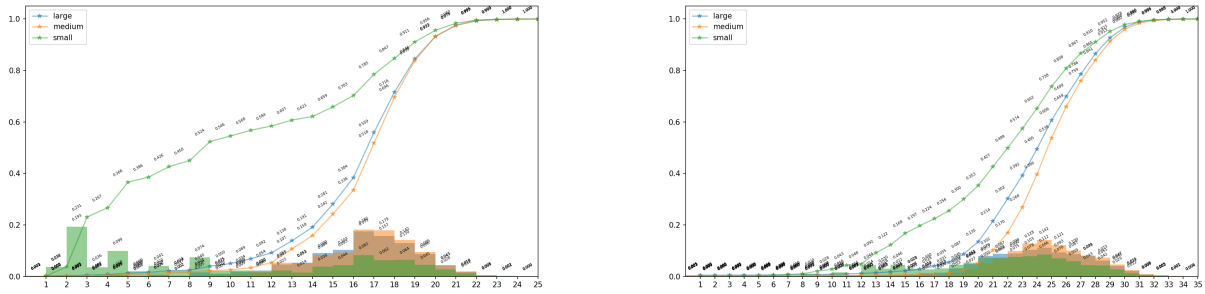
where p_i and IoU_i are the original classification score and predicted IoU of ith detected box, and $\alpha \in [0,1]$ is the hyperparameter to control the contribution of the classification score and predicted IoU to the final detection confidence.

3.4. Distance-IoU Loss

The most common loss used in bbox regression is Smooth L1 Loss [9] , which regresses the parameterizations of the four coordinates (box's center and its width and height). However, these optimization targets are not consistent with the regression evaluation metric IoU, that is, lower loss is not equivalent with higher IoU. Therefore, we turn to different IoU losses presented in past few years, directly regressing the IoU metric, such as GIoU [32], DIoU and CIoU [61]. The reason we choose DIoU [61] as our regression loss is that small faces is the main challenge of face detection since there are about two thirds data in WIDER FACE [48] belong to small object and DIoU [61] is more friendly to small objects. Practically, DIoU gets better performance on APsmall of the validation set of MS COCO 2017 [20]. And theoretically, DIoU is defined as:

$$L_{DIoU} = 1 - IoU + \frac{\rho^2(\mathbf{b}, \mathbf{b}^{gt})}{c^2} \tag{2}$$

where $\mathbf{b}$ and $\mathbf{b}^{gt}$ denote the central points of predicted box and ground-truth box, $\rho(\cdot)$ is the Euclidean distance, and c is the diagonal length of the smallest enclosing box covering the two boxes. The extra penalty term $\frac{\rho^2(\mathbf{b}, \mathbf{b}^{gt})}{c^2}$ proposes to minimize the normalized distance between central points of predicted box and ground-truth box. Compared to large objects, the same distance of central points in small objects will be penalized more, which help detectors learn more about small objects in regression.



(a) (b)

Figure 2: The cumulative distribution and density function of the number of positive samples assigned to each ground-truth. Different colors represent different scales of ground-truth based on the evaluation across scales on COCO dataset. (a) distribution of Retinaface's [6] settings. (b) distribution of this work's settings.

Table 1: AP performance on WIDER FACE validation subset

Baseline	DIoU	Inception	IoU-aware	DCN	TTA	Easy	Medium	Hard
√	-	-	-	-	-	0.959	0.952	0.924
√	√	-	-	-	-	0.959	0.952	0.927
√	√	√	-	-	-	0.958	0.952	0.928
√	√	√	√	-	-	0.963	0.955	0.929
√	√	√	√	√	-	0.963	0.957	0.930
√	√	√	√	√	√	**0.970**	**0.963**	**0.934**

4. Experiments

4.1. Dataset

WIDER FACE dataset [48] is the largest face detection dataset, which contains 32,203 images and 393,703 faces. Since its variety of scale, pose, occlusion, expression, illumination and event, it is difficult and close to reality. The whole dataset is divided into train/val/test sets by ratio 50%/10%/40% within each event class. Furthermore, based on the detection rate of EdgeBox [64], each subset is defined into three levels of difficulty: 'Easy', 'Medium', 'Hard'. From the name of these three levels, we know that 'Hard' is more challenging. And from further analysis, we find that data in 'Hard' covers 'Medium' and 'Easy', which demonstrate that performance on 'Hard' can better reflect the effectiveness of different methods.

4.2. Implementation Details

Feature Extractor. We use ResNet-50 [11] as backbone and Feature Pyramid Network (FPN) [18] as neck to construct the feature extractor. This combination is widely used in almost all detectors, so we think it can serve as a fair playground for replication and comparison. In order to cover the tiny faces, FPN [18] we employed extends to level P_2 like some previous works do. In total, there are 6 levels in FPN [18] from level P_2 to P_7.

Losses. The losses of classification, regression and IoU prediction are focal loss, DIoU loss and cross-entropy loss, respectively.

Normalization Method Batch Normalization (BN) [14] is an extremely important technique for deep learning. It can help models converge and enable various networks to train. However, the performance of the model will degrade with the batch size decreasing especially when batch size is smaller than 4, caused by inaccurate batch statistics estimation. Considering that large volume GPUs are not widely used, which may cause problems for replication, with GeForce GTX 1080 Ti, we replace all the BN layer in network with Group Normalization [44] which is a simple alternative to BN and independent of batch sizes, and the performance of which is stable.

Anchor and Assigner Settings Basically, we set 6 anchors from the set $2^{4/3} \times \{4, 8, 16, 32, 64, 128\}$ since there are 6 levels in our FPN [18]. We adjust the base scale to $2^{4/3}$ in order to better cover the tiny faces, use the mean

Table 2: AP performance of different methods on WIDER FACE validation subset and test subset

Method	Backbone	Val			Test		
		Easy	Medium	Hard	Easy	Medium	Hard
AInnoFace [53] †	ResNet-152	0.970	0.961	0.918	0.965	0.957	0.912
RetinaFace [6] †	ResNet-152	0.969	0.961	0.918	0.963	0.956	0.914
RefineFace [57] †	ResNet-152	0.972	0.962	0.920	0.966	0.958	0.914
ASFD-D6 [51] †	ResNet-152	**0.972**	**0.965**	0.925	**0.967**	**0.962**	0.921
HAMBox [23] †	ResNet-50	0.970	0.964	0.933	0.959	0.955	0.923
TinaFace **(ours)**	ResNet-50	0.963	0.957	0.930	0.952	0.947	0.921
TinaFace **(ours)** †	ResNet-50	0.970	0.963	**0.934**	0.958	0.953	**0.924**

† Note that different methods may use different TTA methods. (It is difficult to verify since most of these methods do not provide codes).

value of aspect ratio of ground-truths as anchor ratio, and set three scales at step $2^{1/3}$ in each level. For assigner, the IoU threshold for matching strategy is 0.35, and ignore-zone is not applied.

To better understand the advantage of our settings, we utilize the detection analysis tool [1] and conduct two experiments to get the distribution of positive samples assigned to each ground-truth shown in Figure 2. As illustrated in Figure 2a, although RetinaFace [6] can recall most of the faces, it does not pay attention to the imbalance problem across scales, that is, small ground-truths get less positive anchors to train, while large one can get more, which leads the degraded performance on small ground-truths. Turning to Figure 2b, we notice that the imbalanced problem is largely relieved. The distribution of the number of positive assigned samples is highly similar across scale.

Data Augmentation. First, crop the square patch from the original picture with a random size from the set $[0.3, 0.45, 0.6, 0.8, 1.0]$ of the short edge of the original image and keep the overlapped part of the face box if its centre is within the crop patch. Then do photo distortion and random horizontal flip with the probability of 0.5. Finally, resize the patch into 640×640 and normalize.

Training Settings. We train the model by using SGD optimizer (momentum 0.9, weight decay 5e-4) with batch size 3×4 on three GeForce GTX 1080 Ti. The schedule of learning rate is annealing down from 3.75e-3 to 3.75e-5 every 30 epochs out of 630 epochs using the cosine decay rule. And in the first 500 iterations, learning rate linearly warms up from 3.75e-4 to 3.75e-3.

Testing Settings. Single Scale testing only contains a keep-ratio resize, which guarantees that the short and long edge of image do not surpass 1100 and 1650. Test Time Augmentation(TTA) is composed of multi-scale (the short edge of image is $[500, 800, 1100, 1400, 1700]$), shift (the direction is $[(0,0),(0,1),(1,0),(1,1)]$), horizontal flip and box voting.

4.3. Evaluation on WIDER FACE

As shown in Table 1, we present the AP performance of models described in Section 3 on WIDER FACE validation subset. Our baseline model using single scale testing gets 95.9%, 95.2%, 92.4% in the three settings on the validation subset. Then we introduce DIoU [61], Inception [36], IoU-aware [43], DCN [4] modules and TTA to further improve the performance of detector by 1.1%, 1.1%, 1.0% on three settings, respectively.

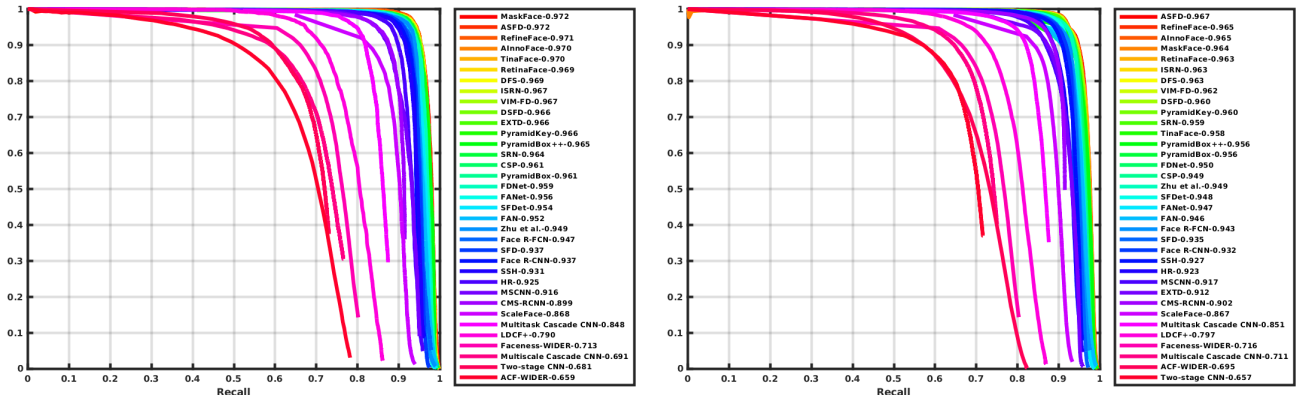
4.4. Comparsion with other methods on WIDER FACE

As shown in Figure 3, we compare TinaFace with recent face detection methods [51, 57, 53, 6, 7, 49, 38, 17, 56, 60, 21, 16, 50, 3, 41, 54, 37, 59, 52, 42, 63, 58, 26, 40, 13, 2, 46, 62, 55, 27, 48, 47, 45] on both validation and testing subsets. For better comparsion, we pick up top-5 methods to form the Table 2 (HAMBox [23] isn't listed in Figure 3 since its results are not updated on the official website of WIDER FACE [2]).

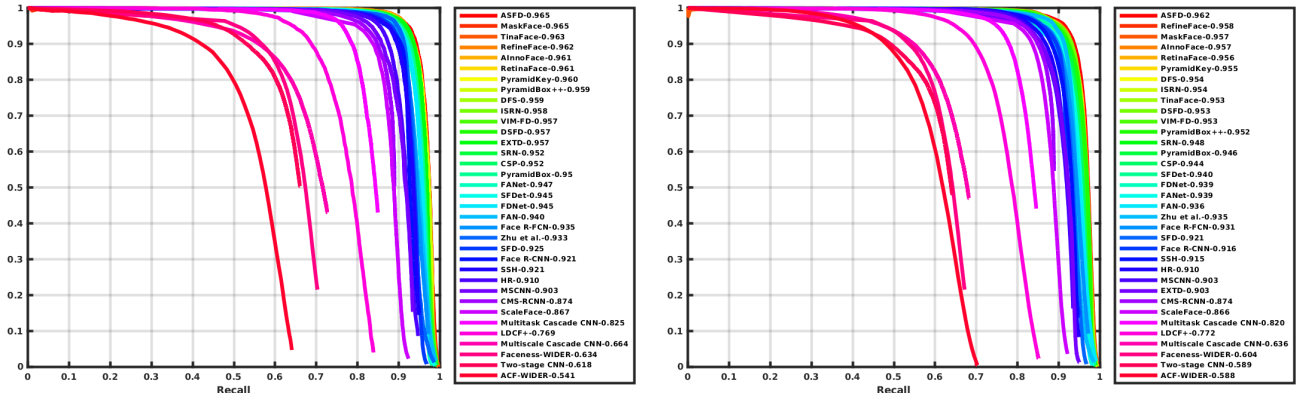
Surprisingly, with single-scale and single-model, our model already gets very promising and almost state-of-the-art performance especially in the hard setting, which respectively outperforms ASFD-D6 [51] in validation subset and test subset. Moreover, our model uses ResNet-50 as backbone, which is much smaller than what ASFD-D6 [51] uses. In the case of using the same backbone, our final model with TTA outperforms the current state-of-the-art method HAMBox [23].

[1] https://github.com/Media-Smart/volkscv

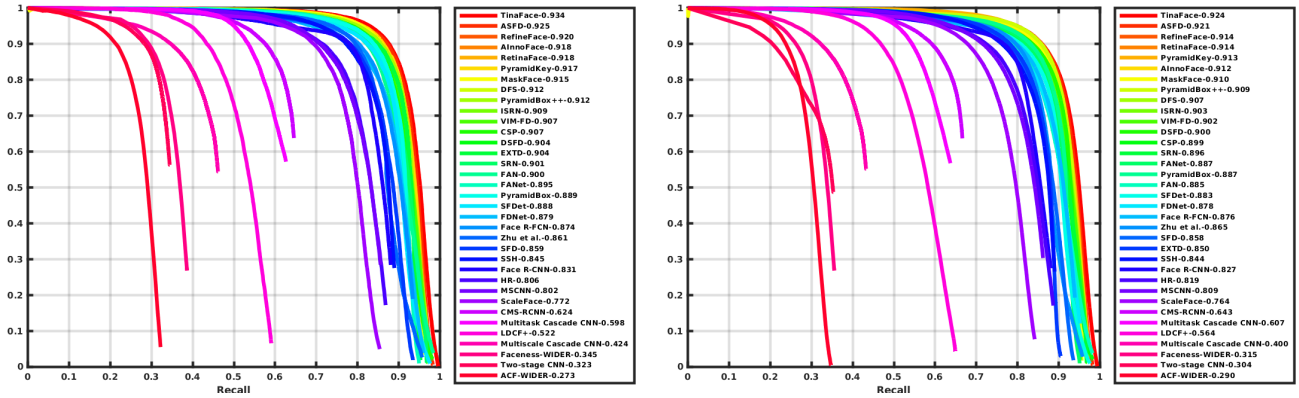
[2] http://shuoyang1213.me/WIDERFACE/WiderFace_Results.html



(a) Val: Easy (b) Test: Easy



(c) Val: Medium (d) Test: Medium



(e) Val: Hard (f) Test: Hard

Figure 3: Precision-recall curves on the WIDER FACE validation and test subsets.

5. Conclusion

In this paper, we point out that face detection is actually a one class generic object detection problem. It indicates that methods presented in generic object detection can be used for handling this problem. Then we present a strong but simple baseline method based on generic object detection for dealing with face detection named TinaFace to further illustrate

this point. The whole network is simple and straightforward, and all the recent tricks equipped are easily implemented and built on existing modules. On the hard setting of the test subset of WIDER FACE, Our model without TTA already exceeds most recent face detection methods like ASFD-D6, which will be extremely efficient and effective. Besides, our final model achieves the state-of-the-art face detection performance.

References

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

[2] Zhaowei Cai et al. "A unified multi-scale deep convolutional neural network for fast object detection". In: *European conference on computer vision*. Springer. 2016, pp. 354–370.

[3] Cheng Chi et al. "Selective refinement network for high performance face detection". In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 33. 2019, pp. 8231–8238.

[4] Jifeng Dai et al. "Deformable convolutional networks". In: *Proceedings of the IEEE international conference on computer vision*. 2017, pp. 764–773.

[5] Navneet Dalal and Bill Triggs. "Histograms of oriented gradients for human detection". In: *2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05)*. Vol. 1. IEEE. 2005, pp. 886–893.

[6] Jiankang Deng et al. "Retinaface: Single-stage dense face localisation in the wild". In: *arXiv preprint arXiv:1905.00641* (2019).

[7] Samuel WF Earp et al. "Face Detection with Feature Pyramids and Landmarks". In: *arXiv preprint arXiv:1912.00596* (2019).

[8] Pedro Felzenszwalb, David McAllester, and Deva Ramanan. "A discriminatively trained, multiscale, deformable part model". In: *2008 IEEE conference on computer vision and pattern recognition*. IEEE. 2008, pp. 1–8.

[9] Ross Girshick. "Fast r-cnn". In: *Proceedings of the IEEE international conference on computer vision*. 2015, pp. 1440–1448.

[10] Ross Girshick et al. "Rich feature hierarchies for accurate object detection and semantic segmentation". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2014, pp. 580–587.

[11] Kaiming He et al. "Deep residual learning for image recognition". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 770–778.

[12] Kaiming He et al. "Mask r-cnn". In: *Proceedings of the IEEE international conference on computer vision*. 2017, pp. 2961–2969.

[13] Peiyun Hu and Deva Ramanan. "Finding Tiny Faces". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. July 2017.

[14] Sergey Ioffe and Christian Szegedy. "Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift". In: *International Conference on Machine Learning*. 2015, pp. 448–456.

[15] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. "Imagenet classification with deep convolutional neural networks". In: *Communications of the ACM* 60.6 (2017), pp. 84–90.

[16] Jian Li et al. "DSFD: Dual Shot Face Detector". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2019.

[17] Zhihang Li et al. "Pyramidbox++: High performance detector for finding tiny face". In: *arXiv preprint arXiv:1904.00386* (2019).

[18] Tsung-Yi Lin et al. "Feature pyramid networks for object detection". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017, pp. 2117–2125.

[19] Tsung-Yi Lin et al. "Focal loss for dense object detection". In: *Proceedings of the IEEE international conference on computer vision*. 2017, pp. 2980–2988.

[20] Tsung-Yi Lin et al. "Microsoft coco: Common objects in context". In: *European conference on computer vision*. Springer. 2014, pp. 740–755.

[21] Wei Liu et al. "High-Level Semantic Feature Detection: A New Perspective for Pedestrian Detection". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2019.

[22] Wei Liu et al. "Ssd: Single shot multibox detector". In: *European conference on computer vision*. Springer. 2016, pp. 21–37.

[23] Yang Liu et al. "HAMBox: Delving Into Mining High-Quality Anchors on Face Detection". In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2020.

[24] David G Lowe. "Distinctive image features from scale-invariant keypoints". In: *International journal of computer vision* 60.2 (2004), pp. 91–110.

[25] Mahyar Najibi, Bharat Singh, and Larry S. Davis. "FA-RPN: Floating Region Proposals for Face Detection". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2019.

[26] Mahyar Najibi et al. "SSH: Single Stage Headless Face Detector". In: *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*. Oct. 2017.

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

[28] Joseph Redmon and Ali Farhadi. "YOLO9000: better, faster, stronger". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017, pp. 7263–7271.

[29] Joseph Redmon and Ali Farhadi. "Yolov3: An incremental improvement". In: *arXiv preprint arXiv:1804.02767* (2018).

[30] Joseph Redmon et al. "You only look once: Unified, real-time object detection". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 779–788.

[31] Shaoqing Ren et al. "Faster r-cnn: Towards real-time object detection with region proposal networks". In: *Advances in neural information processing systems*. 2015, pp. 91–99.

[32] Hamid Rezatofighi et al. "Generalized intersection over union: A metric and a loss for bounding box regression". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2019, pp. 658–666.

[33] Karen Simonyan and Andrew Zisserman. "Very deep convolutional networks for large-scale image recognition". In: *arXiv preprint arXiv:1409.1556* (2014).

[34] Bharat Singh and Larry S Davis. "An analysis of scale invariance in object detection snip". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018, pp. 3578–3587.

[35] Bharat Singh, Mahyar Najibi, and Larry S Davis. "Sniper: Efficient multi-scale training". In: *Advances in neural information processing systems*. 2018, pp. 9310–9320.

[36] Christian Szegedy et al. "Going deeper with convolutions". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2015, pp. 1–9.

[37] Xu Tang et al. "Pyramidbox: A context-assisted single shot face detector". In: *Proceedings of the European Conference on Computer Vision (ECCV)*. 2018, pp. 797–813.

[38] Wanxin Tian et al. "Learning better features for face detection with feature fusion and segmentation supervision". In: *arXiv preprint arXiv:1811.08557* (2018).

[39] Paul Viola and Michael Jones. "Robust real-time face detection". In: *null*. IEEE. 2001, p. 747.

[40] Hao Wang et al. "Face r-cnn". In: *arXiv preprint arXiv:1706.01061* (2017).

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

[42] Yitong Wang et al. "Detecting faces using region-based fully convolutional networks". In: *arXiv preprint arXiv:1709.05256* (2017).

[43] Shengkai Wu, Xiaoping Li, and Xinggang Wang. "IoU-aware single-stage object detector for accurate localization". In: *Image and Vision Computing* (2020), p. 103911.

[44] Yuxin Wu and Kaiming He. "Group normalization". In: *Proceedings of the European conference on computer vision (ECCV)*. 2018, pp. 3–19.

[45] Bin Yang et al. "Aggregate channel features for multi-view face detection". In: *IEEE international joint conference on biometrics*. IEEE. 2014, pp. 1–8.

[46] Shuo Yang et al. "Face detection through scale-friendly deep convolutional networks". In: *arXiv preprint arXiv:1706.02863* (2017).

[47] Shuo Yang et al. "From facial parts responses to face detection: A deep learning approach". In: *Proceedings of the IEEE international conference on computer vision*. 2015, pp. 3676–3684.

[48] Shuo Yang et al. "Wider face: A face detection benchmark". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 5525–5533.

[49] Dmitry Yashunin, Tamir Baydasov, and Roman Vlasov. "MaskFace: multi-task face and landmark detector". In: *arXiv preprint arXiv:2005.09412* (2020).

[50] YoungJoon Yoo, Dongyoon Han, and Sangdoo Yun. "Extd: Extremely tiny face detector via iterative filter reuse". In: *arXiv preprint arXiv:1906.06579* (2019).

[51] Bin Zhang et al. "ASFD: Automatic and Scalable Face Detector". In: *arXiv preprint arXiv:2003.11228* (2020).

[52] Changzheng Zhang, Xiang Xu, and Dandan Tu. "Face detection using improved faster rcnn". In: *arXiv preprint arXiv:1802.02142* (2018).

[53] Faen Zhang et al. "Accurate face detection for high performance". In: *arXiv preprint arXiv:1905.01585* (2019).

[54] Jialiang Zhang et al. "Feature agglomeration networks for single stage face detection". In: *Neurocomputing* 380 (2020), pp. 180–189.

[55] Kaipeng Zhang et al. "Joint face detection and alignment using multitask cascaded convolutional networks". In: *IEEE Signal Processing Letters* 23.10 (2016), pp. 1499–1503.

[56] Shifeng Zhang et al. "Improved selective refinement network for face detection". In: *arXiv preprint arXiv:1901.06651* (2019).

[57] Shifeng Zhang et al. "Refineface: Refinement neural network for high performance face detection". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2020).

[58] Shifeng Zhang et al. "S3fd: Single shot scale-invariant face detector". In: *Proceedings of the IEEE international conference on computer vision*. 2017, pp. 192–201.

[59] Shifeng Zhang et al. "Single-shot scale-aware network for real-time face detection". In: *International Journal of Computer Vision* 127.6-7 (2019), pp. 537–559.

[60] Yundong Zhang, Xiang Xu, and Xiaotao Liu. "Robust and high performance face detector". In: *arXiv preprint arXiv:1901.02350* (2019).

[61] Zhaohui Zheng et al. "Distance-IoU Loss: Faster and Better Learning for Bounding Box Regression." In: *AAAI*. 2020, pp. 12993–13000.

[62] Chenchen Zhu et al. "Cms-rcnn: contextual multi-scale region-based cnn for unconstrained face detection". In: *Deep learning for biometrics*. Springer, 2017, pp. 57–79.

[63] Chenchen Zhu et al. "Seeing Small Faces From Robust Anchor's Perspective". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2018.

[64] C Lawrence Zitnick and Piotr Dollár. "Edge boxes: Locating object proposals from edges". In: *European conference on computer vision*. Springer. 2014, pp. 391–405.