Hindawi
Scientific Programming
Volume 2020, Article ID 7845384, 8 pages
https://doi.org/10.1155/2020/7845384

Research Article

CenterFace: Joint Face Detection and Alignment Using Face as Point

Yuanyuan Xu,[1,2] **Wan Yan**,[3] **Genke Yang**,[2] **Jiliang Luo**,[1] **Tao Li**,[4] **and Jianan He**[4]

[1]*College of Information Science and Engineering, Huaqiao University, Xiamen 361021, China*
[2]*Department of Automation, Shanghai Jiaotong University, Shanghai 200240, China*
[3]*Xiamen Star Clouds Network Technology Co., Ltd., Xiamen 361005, China*
[4]*Central Laboratory of Health Quarantine, Shenzhen International Travel Health Care Center and Shenzhen Academy of Inspection and Quarantine, Shenzhen Customs District, Shenzhen 518033, China*

Correspondence should be addressed to Yuanyuan Xu; yyxu@hqu.edu.cn

Received 5 February 2020; Revised 3 June 2020; Accepted 17 June 2020; Published 2 July 2020

Academic Editor: Chenxi Huang

Face detection and alignment in unconstrained environment is always deployed on edge devices which have limited memory storage and low computing power. This paper proposes a one-stage method named CenterFace to simultaneously predict facial box and landmark location with real-time speed and high accuracy. The proposed method also belongs to the anchor-free category. This is achieved by (a) learning face existing possibility by the semantic maps, (b) learning bounding box, offsets, and five landmarks for each position that potentially contains a face. Specifically, the method can run in real time on a single CPU core and 200 FPS using NVIDIA 2080TI for VGA-resolution images and can simultaneously achieve superior accuracy (WIDER FACE Val/Test-Easy: 0.935/0.932, Medium: 0.924/0.921, Hard: 0.875/0.873, and FDDB discontinuous: 0.980 and continuous: 0.732).

1. Introduction

Face detection and alignment is one of the fundamental issues in computer vision and pattern recognition and is often deployed in mobile and embedded devices. These devices typically have limited memory storage and low computing power. Therefore, it is necessary to predict the position of the face box and the landmark at the same time, and it is excellent in speed and precision.

With the great breakthrough of convolutional neural networks (CNN), face detection has achieved remarkable progress in recent years. Previous face detection methods have inherited the paradigm of anchor-based generic object detection frameworks, which can be divided into two categories: two-stage method (Faster-RCNN [1]) and one-stage method (SSD [2]). Compared with the two-stage method, the one-stage method is more efficient and has higher recall rate, but it tends to achieve a higher false positive rate and to compromise the localization accuracy. Then, Hu and Ramanan [3] used a two-stage approach to the Region Proposal Networks (RPN) [1] to detect faces directly, while SSH [4] and S3FD [5] developed a scale-invariant network in a single network to detect faces with mutiscale from different layers.

The previous anchor-based methods have some drawbacks. On the one hand, in order to improve the overlap between anchor boxes and ground truth, a face detector usually requires a large number of dense anchors to achieve a good recall rate. For example, more than 100 k anchor boxes is designed in RetinaFace [6] for a 640×640 input image. On the other hand, the anchor is a hyperparameter design that is statistically calculated from a particular dataset, so it is not always feasible to other applications, which goes against the generality.

In addition, the current state-of-the-art face detectors has achieved considerable accuracy on the benchmark WIDER FACE [7] by using heavy pretrained backbones such as VGG16 [8] and resnet50/152 [9]. First, these detectors are difficult to use in practice because the network consumes too

much time and the model size is also too large. Secondly, it is not convenient for face recognition application without facial landmark prediction. Therefore, joint detection and alignment, as well as better balance of accuracy and latency, are essential for practical applications.

Inspired by the anchor-free universal object detection framework [1, 10–15], this paper proposes a simpler and more effective face detection and alignment method named CenterFace, which is not only lightweight but also powerful. The network structure about the CenterFace is shown in Figure 1, which can be trained end-to-end. We use the center point of the face's bounding box to represent the face, then facial box size and landmark are regressed directly to image features at the center location. So, face detection and alignment are transformed to the standard key point estimation problem [16–18]. The peak in the heat map corresponds to the center of the face. The image features at each peak predict the size of the face and the face key points. This approach was fully evaluated and the latest detection performance were shown on a number of benchmark datasets for face detection, including FDDB [19] and WIDER FACE.

In summary, the main contributions of this work can be summarized as four-fold:

(i) By introducing the anchor-free design, face detection is transformed into a standard key point estimation problem, using only a larger output resolution (output stride is 4) compared to previous detectors

(ii) Based on the multitask learning strategy, the face as point design is proposed to predict the faceBoxes and five key points at the same time

(iii) This paper proposes a feature pyramid network using common layer for accurate and fast face detection

(iv) Comprehensive experimental results based on popular benchmarks FDDB and WIDER FACE, as well as CPU and GPU hardware platforms, have demonstrated the superiority of the proposed method in terms of speed and accuracy

2. Related Works

2.1. Cascaded CNN Methods. The method of cascade convolutional neural network (CNN) [20–22] uses cascaded CNN framework to learn features in order to improve the performance and maintain efficiency. However, there are some problems about cascaded CNN-based detector. (1) The runtime of these detector is negatively correlated with the number of faces on the input image. The speed will dramatically degrade when the number of faces increases. (2) Because these methods optimize each module separately, the training process becomes extremely complicated.

2.2. Anchor Methods. Inspired by generic object detection methods [2, 14, 15, 23–27], which embraced all the recent advancement in deep learning, face detection has recently achieved remarkable progress [3–5, 28]. Different from generic object detection, the ratio of the face scale is usually from 1 : 1 to 1 : 1.5. The latest methods [6, 28] focus on single-stage design, which densely samples' face locations and scales on feature pyramids, demonstrating promising performance and yielding faster speed compared to two-stage methods [29, 30]. However, dense samples result in long time consuming.

2.3. Anchor-Free Methods. In our view, Cascaded CNN methods are also a kind of anchor-free methods. However, these method uses sliding window to detect human faces and relies on image pyramids. It has some shortcomings such as slow speed and complex training process. LFFD [31] regards the RFs as natural anchors which can cover continuous face scales, which is just another way to define anchor, but the training time is about 5 days with two NVIDIA GTX1080TI. Our CenterFace simply represents faces by a single point at their bounding box center; then, facial box size and landmark are regressed directly from image features at the center location. Thus, face detection is transformed into a standard key point estimation problem. And the training time of a NVIDIA GTX2080TI is only one day.

2.4. Multitask Learning. Multitask learning uses multiple supervisory labels to improve the accuracy of each task by utilizing the correlation between tasks. Joint face detection and alignment [17, 20] is widely used because alignment task, paralleling with the backbone, provides better features for face classification task with face point information. Similarly, Mask R-CNN [32] significantly improves the detection performance by adding a branch for predicting an object mask.

3. CenterFace

3.1. Mobile Feature Pyramid Network. We adopted Mobilenetv2 [33] as the backbone and Feature Pyramid Network (FPN) [14] as the neck for the subsequent detection. In general, FPN uses a top-down architecture with lateral connections to build a feature pyramid from a single scale input. CenterFace represents the face through the center point of the face box, and face size and facial landmark are then regressed directly from image features of the center location. Therefore, only one layer in the pyramid is used for face detection and alignment. We construct a pyramid with levels {P-L}, $L = 3, 4, 5$, where L indicates pyramid level. Pl has $1/2^{L}$ resolution of the input. All pyramid levels have $C = 24$ channels, and we define classification loss, box regression loss, and landmark regression loss only on P2.

3.2. Face as Point. Let $[x_1, y_1, x_2, y_2]$ be the bounding box of face. Facial center point lies at $c = [(x_1 + x_2)/2$ and $(y_1 + y_2)/2]$. Let $I \in R^{W \times H \times 3}$ be an input image of width W and height H. Our aim is to produce the heat map $Y \in [0, 1]^{W/R \times H/R}$, where R is the output stride, and we use the default output stride of $R = 4$. During training, the prediction $\widehat{Y}_{x,\,y} = 1$ corresponds to a face center, while $\widehat{Y}_{x,\,y} = 0$ is background. For each ground

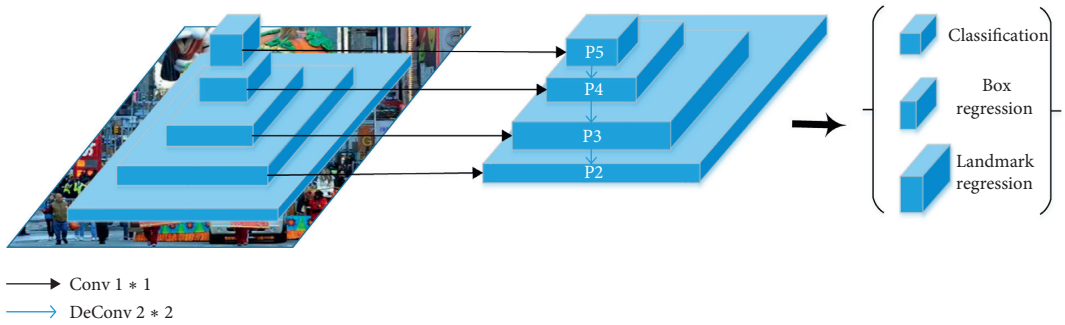

Figure 1: Architecture of the CenterFace.

truth $Y_{x,y}$, we calculate the equivalent heat map by using y an unnormalized 2D Gaussian to represent the ground truth. The training loss is a variant of focal loss [15]:

$$L_c = \begin{cases} -\left(1-\hat{Y}_{xy}\right)^{\alpha}\log\left(\hat{Y}_{xy}\right) & \text{if } Y_{xy}=1, \\ -\left(1-Y_{xy}\right)^{\beta}\left(\hat{Y}_{xy}\right)^{\alpha}\log\left(1-\hat{Y}_{xy}\right) & \text{otherwise,} \end{cases} \tag{1}$$

where α and β are hyperparameters of the focal loss, which are designated as $\alpha=2$ and $\beta=4$ in all our experiments following Law and Deng [34].

To gather global information and to reduce memory usage, downsampling is applied to an image convolutionally, and the size of the output is usually smaller than the image. Hence, a location (x, y) in the image is mapped to the location $(x/n, y/n)$ in the heatmaps, where n is the downsampling factor. When we remap the locations from the heatmaps to the input image, some pixel may be not alignment, which can greatly affect the accuracy of facial boxes. To address this issue, we predict position offsets to adjust the center position slightly before remapping the center position to the input resolution:

$$o_k = \left(\frac{x_k}{n} - \left\lfloor\frac{x_k}{n}\right\rfloor, \frac{y_k}{n} - \left\lfloor\frac{y_k}{n}\right\rfloor\right), \tag{2}$$

where o_k is the offset and x_k and y_k are the x and y coordinate for face center k. We apply the L1 Loss at ground-truth center position.

3.3. Box and Landmark Prediction. To reduce the computational burden, we use a single size prediction $S \in R^{W/4\times H/4}$ for facial box and landmarks. Each ground-truth bounding box is specified as $G=(x1, y1, x2, y2)$. During training, our goal is to learn a transformation that maps the networks position outputs $(\hat{h}, \hat{w})$ to center position in the feature maps:

$$\begin{aligned} \hat{h} &= \log\left(\frac{x_2}{R} - \frac{x_1}{R}\right), \\ \hat{w} &= \log\left(\frac{y_2}{R} - \frac{y_1}{R}\right), \end{aligned} \tag{3}$$

where R is the stride of networks, which are designated as $R=4$.

Different from box regression, the regression of the five facial landmarks adopts the target normalization method based on the center position:

$$\begin{aligned} \widehat{lm_x} &= \frac{lm_x}{\text{box}_w} - \frac{c_x}{\text{box}_w}, \\ \widehat{lm_y} &= \frac{lm_y}{\text{box}_h} - \frac{c_y}{\text{box}_h}, \end{aligned} \tag{4}$$

where lm_x and lm_y are the x and y coordinates for face landmark, c_k and c_k are the x and y coordinates for face center, and box_w and box_h are width and height of the face. We also use smooth L1 loss to facial box and landmark prediction at the center location.

For any training face center, we minimise the following multitask loss:

$$L = L_c + \lambda_{\text{off}}L_{\text{off}} + \lambda_{\text{box}}L_{\text{box}} + \lambda_{lm}L_{lm}, \tag{5}$$

where λ_{off}, λ_{box}, and λ_{lm} is used to scale the loss, and we use 1, 0.1, and 0.1, respectively, in all our experiments.

3.4. Training Details

3.4.1. Dataset. The proposed method is trained on the training set of WIDER FACE benchmark, including 12,880 images with more than 150,000 valid faces in scale, pose, expression, occlusion, and illumination. RetinaFace [6] introduces five levels of face image quality and annotates five landmarks on faces.

3.4.2. Data Augmentation. Data augmentation is important to improve the generalization. We use random flip, random scaling [35], color jittering, and randomly crop square patches from the original images and resize these patches into 800×800 to generate larger training faces. Faces that are less than 8 pixels are discarded directly.

3.4.3. Training Parameters. We train the CenterFace using Adam optimiser with a batch-size 8 and learning rate $5e-4$ for 140 epochs, with the learning rate dropped $10x$ at 90 and 120 epochs, respectively. The downsampling layers of MobilenetV2 are initialized with ImageNet pretrain and the

up-sampling layers are randomly initialized. The training time is about one day with one NVIDIA GTX2080TI.

4. Experiments

In this section, we firstly introduce the runtime efficiency of CenterFace and then evaluate it on the common face detection benchmarks.

4.1. Running Efficiency. The existing CNN face detectors can be accelerated by GPUs, but they are not fast enough in most practical applications, especially CPU-based applications. As described below, our CenterFace is efficient enough to meet practical requirements and its model size is only 7.2 MB. In Table 1, comparing with other detectors, our method can exceed the real-time running speed (>100 FPS) at different resolutions by using a single NVIDIA GTX2080TI. Owing to the DSFD, PyramidBox, S3FD, and SSH are too slow when running on CPU platforms, and we only evaluate the proposed CenterFace, FaceBoxes, MTCNN, and CasCNN at VGA-resolution images on CPU and the mAP means the true positive rate at 1000 false positives on FDDB. As listed in Table 2, our CenterFace can run at 30 FPS on the CPU with state-of-the-art accuracy.

Table 1: Running efficiency on GTX2080TI.

Approach	640×480 (ms)	1280×720 (ms)	1920×1080 (ms)
DSFD [29]	78.08	187.78	393.82
PyramidBox [28]	50.51	142.34	331.93
S3FD [5]	21.75	55.73	119.53
LFFD [31]	7.60	16.37	31.41
CenterFace	5.51	6.47	8.79

Table 2: Running efficiency on CPU.

Approach	CPU-model	mAP (%)	FPS
CasCNN [21]	E5-2620@2.00	85.7	14
MTCNN [20]	N/A@2.60	94.4	16
FaceBoxes [36]	E5-2660v3@2.60	96.0	20
CenterFace	I7-6700@2.6	98.0	30

Table 3: Evaluation results on FDDB.

Method	Disc ROC curves score	Cont ROC curves score
DFSD [29]	0.984	0.754
PyramidBox [28]	0.982	0.757
S3FD [5]	0.981	0.754
MTCNN [20]	0.944	0.708
Faceboxes3.2 [36]	0.960	0.729
LFFD [31]	0.973	0.724
CenterFace	0.980	0.732

4.2. Evaluation on Benchmarks

4.2.1. FDDB Dataset. FDDB contains 2845 images with 5171 unconstrained faces collected from the Yahoo news website. We evaluate our face detector on FDDB against the other state-of-the-art methods, and the results are shown in Table 3 and Figure 2, respectively. We also add DFSD, PyramidBox, and S3FD detectors, whereas these detectors are much slower due to the larger backbone and denser anchors. Our CenterFace can also achieve good performance on both discontinuous and continuous ROC curves, i.e., 98.0% and 72.9% when the number of false positives equals to 1,000 and it outperforms LFFD, FaceBoxes, and MTCNN evidently.

4.2.2. WIDER FACE Dataset. Until now, WIDER FACE is the most widely used benchmark for face detection. The WIDER FACE dataset is split into training (40%), validation (10%), and testing (50%) subsets by randomly sampling from 61 scene categories. All the compared methods are trained on the training set. For testing on WIDER FACE, we follow the standard practices of [6] and employ flip as well as multiscale strategies. Box voting [36] is applied on the union set of predicted faceBoxes using an IoU threshold at 0.4. We report the results on the testing sets in Table 4, respectively. The proposed method CenterFace achieves 0.932 (Easy), 0.921 (Medium), and 0.873 (Hard) for testing set. Although it has gaps with state-of-the-art methods, but consistently outperforms SSH (using VGG16 as the backbone), LFFD, FaceBoxes, and MTCNN. Additionally, CenterFace is better than S3FD that uses VGG16 as the backbone and dense anchors on hard parts.

Furthermore, we also test on WIDER FACE not only with the original image but also with a single inference, and our CenterFace also produces the good average precision (AP) in all the subsets of both validation sets, i.e., 92.2% (Easy), 91.1% (Medium), and 78.2% (Hard) for the validation set. Figure 3 shows some qualitative results on the WIDER FACE dataset.

4.2.3. AFLW Dataset. To evaluate the accuracy of face alignment, we compare CenterFace with MTCNN on the AFLW dataset. The mean error is measured by the distances between the estimated landmarks and the ground truths and normalized with respect to the interocular distance. As shown in Figure 4, we give the mean error of each facial landmark on the AFLW dataset [37]. CenterFace significantly decreases the normalized mean errors (NME) from 6.2% to 6.9% when compared to MTCNN.

4.3. Parameter, FLOPs, and Model Size. In this section, the comparison method is studied from the perspective of parameters, computation, and model size. Edge devices always have limited storage. We use FLOPs to measure the computation at resolution 640×480. The number of parameters is closely related to the size of the model. However, the model size may vary slightly with different libraries, and less parameters do not mean less computation. All the information is presented in Table 5.

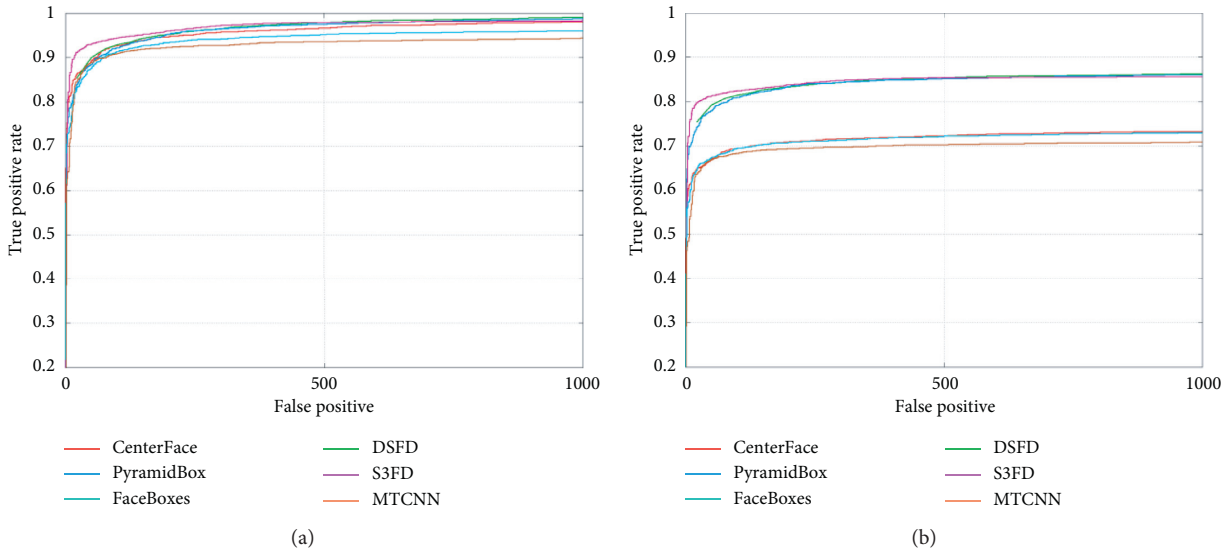

Figure 2: Evaluation on the FDDB dataset. (a) Discontinuous ROC curves. (b) Continuous ROC curves.

Table 4: Performance results on the testing set of WIDER FACE.

Method	Easy	Medium	Hard
RetinaFace [6]	0.963	0.956	0.914
DSFD [29]	0.960	0.953	0.900
PramidBox [28]	0.956	0.946	0.887
S3FD [5]	0.928	0.913	0.840
SSH [4]	0.927	0.915	0.844
MTCNN [20]	0.851	0.820	0.607
FaceBoxes [36]	0.839	0.763	0.396
LFFD [31]	0.896	0.865	0.770
CenterFace	0.932	0.921	0.873

(a)

Figure 3: Continued.

(b)

Figure 3: Face detection results on WIDER face.

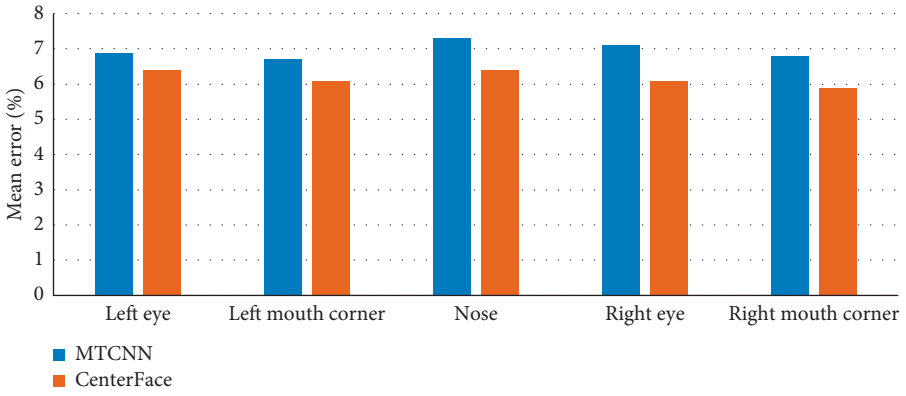

Figure 4: Evaluation on AFLW for face alignment.

Table 5: Number of parameters, FLOPs, and model sizes.

Method	Parameters (M)	FLOPs (G)	Model sizes (MB)
DSFD [29]	141.38	140.19	458
PyramidBox [28]	57.18	236.58	218
S3FD [5]	22.46	96.6	86
SSH [4]	19.75	99.8	79
LFFD [31]	2.15	9.25	9
CenterFace	1.83	2.06	7.2

For the most advanced methods DSFD and PyramidBox, they have a large number of parameters, FLOPs, and model sizes. Evidently, the proposed method has much more efficient computation and light network, which demonstrates the superiority of the concise network design.

5. Conclusion

This paper introduces the CenterFace that has the superiority of the proposed method, performs well on both speed and accuracy, and simultaneously predicts facial box and landmark location. Our proposed method overcomes the drawbacks of the previous anchor-based method by translating face detection and alignment into a standard key point estimation problem. CenterFace represents the face through the center point of the face box, and face size and facial landmark are then regressed directly from image features of the center location. Comprehensive and extensive

experiments are made to fully analyze the proposed method. The final results demonstrate that our method can achieve real-time speed and high accuracy with a smaller model size, making it an ideal alternative for most face detection and alignment applications.

Data Availability

The data used to support the findings of this study have been deposited in the http://mmlab.ie.cuhk.edu.hk/projects/WIDERFace/WiderFace_Results.html repository.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported in part by the National Key R&D Program of China (2018YFC0809200) and Natural Science Foundation of Shanghai (16ZR1416500).

References

[1] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: towards real-time object detection with region proposal networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 6, pp. 1137–1149, 2017.

[2] W. Liu, D. Anguelov, D. Erhan et al., "SSD: Single shot multibox detector," in *Computer Vision–ECCV 2016*, Springer, Berlin, Germany, 2016.

[3] P. Hu and D. Ramanan, "Finding tiny faces," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Honolulu, HI, USA, July 2017.

[4] M. Najibi, P. Samangouei, R. Chellappa, and L. S. Davis, "SSH: Single stage headless face detector,", October 2017.

[5] S. Zhang, X. Zhu, Z. Lei, H. Shi, X. Wang, and S. Z. Li, "S3FD: Single shot scale-invariant face detector," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, October 2017.

[6] J. Deng, J. Guo, Y. Zhou, J. Yu, I. Kotsia, and S. Zafeiriou, "RetinaFace: single-stage dense face localisation in the wild," 2019, https://arxiv.org/abs/1905.00641.

[7] S. Yang, P. Luo, C.-C. Loy, and X. Tang, "Wider face: a face detection benchmark," in *Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[8] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014, https://arxiv.org/abs/1409.1556.

[9] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, Las Vegas, NV, USA, June 2016.

[10] H. Law and J. Deng, "Cornernet: Detecting objects as paired keypoints," in *Computer Vision—ECCV, 2018*, Springer, Berlin, Germany, 2018.

[11] C. Zhu, Y. He, and M. Savvides, "Feature selective anchor-free module for single-shot object detection," 2019, https://arxiv.org/abs/1903.00621.

[12] J. Redmon and A. Farhadi, "Yolov3: an incremental improvement," 2018, https://arxiv.org/abs/1804.02767.

[13] R. Girshick, "Fast R-CNN," in *Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV)*, IEEE, Santiago, Chile, December 2015.

[14] T.-Y. Lin, P. Dollar, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Honolulu, HI, USA, July 2017.

[15] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollar, "Focal loss for dense object detection," in *Proceedings of the 2017 IEEE International Conference on Computer Vision (ICCV)*, Venice, Italy, October 2017.

[16] Z. Cao, G. Hidalgo, T. Simon, S.-E. Wei, and Y. Sheikh, "OpenPose: realtime multi-person 2D pose estimation using part affifinity fields," 2018, https://arxiv.org/abs/1611.08050.

[17] A. Newell, Z. Huang, and J. Deng, "Associative embedding: end-to-end learning for joint detection and grouping," 2017, https://arxiv.org/abs/1611.05424.

[18] X. Zhou, A. Karpur, L. Luo, and Q. Huang, "Starmap for category-agnostic keypoint and viewpoint estimation," in *Computer Vision–ECCV 2018*, Springer, Berlin, Germany, 2018.

[19] V. Jain, "FDDB: A Benchmark for Face Detection in Unconstrained Settings," UMASS Amherst Technical Report, University of Massachusetts, Amherst, MA, USA, 2010.

[20] K. Zhang, Z. Zhang, Z. Li, and Y. Qiao, "Joint face detection and alignment using multitask cascaded convolutional networks," *IEEE Signal Processing Letters*, vol. 23, no. 10, pp. 1499–1503, 2016.

[21] D. Chen, S. Ren, Y. Wei, X. Cao, and J. Sun, "Joint cascade face detection and alignment," in *Computer Vision–ECCV 2014*, Springer, Berlin, Germany, 2014.

[22] H. Li, Z. Lin, X. Shen, J. Brandt, and G. Hua, "A convolutional neural network cascade for face detection," in *Proceedings of the 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Boston, MA, USA, June 2015.

[23] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: unified, real-time object detection," in *Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, IEEE, Las Vegas, NV, USA, June 2016.

[24] J. Huang, V. Rathod, C. Sun et al., "Speed/accuracy trade-offs for modern convolutional object detectors," in *Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Honolulu, HI, USA, July 2017.

[25] L. Liu, W. Ouyang, X. Wang et al., "Deep learning for generic object detection: a survey," 2018, https://arxiv.org/abs/1809.02165.

[26] B. Jiang, R. Luo, J. Mao, T. Xiao, and Y. Jiang, "Acquisition of localization confidence for accurate object detection," in *Computer Vision–ECCV 2018*, Springer, Berlin, Germany, 2018.

[27] T. Kong, F. Sun, H. Liu, Y. Jiang, and J. Shi, "Consistent optimization for single-shot object detection," 2019, https://arxiv.org/abs/1901.06563.

[28] X. Tang, D. K. Du, Z. He, and J. Liu, "Pyramidbox: a context-assisted single shot face detector," in *Computer Vision–ECCV 2018*, Springer, Berlin, Germany, 2018.

[29] C. Chi, S. Zhang, J. Xing, Z. Lei, S. Z. Li, and X. Zou, "Selective refinement network for high performance face detection," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, pp. 8231–8238, 2019.

[30] C. Zhang, X. Xu, and D. Tu, "Face detection using improved faster RCNN," 2018, https://arxiv.org/ftp/arxiv/papers/1802/1802.02142.pdf.

[31] Y. He, D. Xu, L. Wu, M. Jian, S. Xiang, and C. Pan, "LFFD: a light and fast face detector for edge devices," 2019, https://arxiv.org/abs/1904.10633.
[32] K. He, G. Gkioxari, P. Dollar, and R. Girshick, "Mask R-CNN," 2017, https://arxiv.org/abs/1703.06870.
[33] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L. Chen, "MobileNetV2: inverted residuals and linear bottlenecks," in *Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Salt Lake City, UT, USA, June 2018.
[34] X. Zhou, D. Wang, and P. Krahenbuhl, "Objects as points," https://arxiv.org/abs/1904.07850.
[35] B. Singh, M. Najibi, and L. S. Davis, "SNIPER: efficient multi-scale training," 2018, https://arxiv.org/abs/1805.09300.
[36] S. Zhang, X. Zhu, Z. Lei, H. Shi, X. Wang, and S. Z. Li, "FaceBoxes: a cpu real-time face detector with high accuracy," in *Proceedings of IEEE International Joint Conference on Biometrics*, pp. 1–9, IEEE, Denver, CO, USA, October 2017.
[37] M. Köstinger, P. Wohlhart, P. M. Roth, and H. Bischof, "Annotated facial landmarks in the wild: a large-scale, real-world database for facial landmark localization," in *Proceedings of the In IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pp. 2144–2151, Barcelona, Spain, November 2011.