The Visual Computer (2021) 37:805–813
https://doi.org/10.1007/s00371-020-01831-7

ORIGINAL ARTICLE

YOLO-face: a real-time face detector

Weijun Chen[1] · Hongbo Huang[1,2] · Shuai Peng[1] · Changsheng Zhou[1,2] · Cuiping Zhang[1,2]

Published online: 12 March 2020

Abstract
Face detection is one of the important tasks of object detection. Typically detection is the first stage of pattern recognition and identity authentication. In recent years, deep learning-based algorithms in object detection have grown rapidly. These algorithms can be generally divided into two categories, i.e., two-stage detector like Faster R-CNN and one-stage detector like YOLO. Although YOLO and its varieties are not so good as two-stage detectors in terms of accuracy, they outperform the counterparts by a large margin in speed. YOLO performs well when facing normal size objects, but is incapable of detecting small objects. The accuracy decreases notably when dealing with objects that have large-scale changing like faces. Aimed to solve the detection problem of varying face scales, we propose a face detector named YOLO-face based on YOLOv3 to improve the performance for face detection. The present approach includes using anchor boxes more appropriate for face detection and a more precise regression loss function. The improved detector significantly increased accuracy while remaining fast detection speed. Experiments on the WIDER FACE and the FDDB datasets show that our improved algorithm outperforms YOLO and its varieties.

Keywords Face detection · YOLO · Deep learning · Anchor box · Loss function

1 Introduction

Face recognition is one of the most widely used applications in computer vision. Face-based authentication and recognition can be employed in many scenarios. Generally, the first stage in face recognition is to detect and locate faces in images or videos. It is quite straightforward that an accurate detection algorithm can benefit the performance of the system remarkably and vice versa. Therefore, face detection is one of the key steps in the application of face recognition systems. Due to the proliferation of mobile devices and smart cameras, collecting images and videos is becoming more and more convenient. However, the computing ability of such devices is limited relatively. The best solution to this problem may depend on finding faster and efficient algorithms.

Since the success utilizes by Alexnet [1], deep learning has been spread into many fields of artificial intelligence including object detection. R-CNN [2] is the first object detection algorithm based on deep learning. It has greatly promoted the performance compared with the traditional algorithms like Adaboost [3], DPM [4], etc. Subsequent algorithms such as SPP-Net [5], Fast R-CNN [6], Faster R-CNN [7] and R-FCN [8] based on R-CNN have further improved in their accuracy or speed. However, these detectors' slow speed has always been a thwart to the widely application in practice. To speed up the detection procedure, YOLO [9] and SSD [10] presented one-stage detectors to solve the speed problem which can be used for real-time detection tasks.

A major problem encountered in face detection is that the detection accuracy for different face scales in one image varies considerably for one same detector. Recently, some face detection approaches have been trying to deal with different scales by using several network architectures to solve this problem. Another method is using different levels of features drawn from the last several layers of the network. In practice, it is very common that we should detect a variety of face scales in normal scenarios. Undoubtedly, the features used in detecting a face scale of 200 × 200 pixels in one image differ a lot from that of detecting a face in 10 × 10 pixels. YOLOv3 [11] uses a network structure similar to FPN [12] to fuse the features of different levels.

✉ Hongbo Huang
hhb@bistu.edu.cn

[1] Computer School, Beijing Information Science and Technology University, Beijing 100101, China

[2] Institute of Computing Intelligence, Beijing Information Science and Technology University, Beijing 100192, China

The main reason is that the algorithm can detect multi-scale objects. YOLOv3 achieved the state-of-the-art results on the COCO dataset [13]. However, when applied to face detection, the performance was not as well as expected. On the one hand, the sizes of anchor boxes in YOLOv3 suitable for the COCO dataset are not necessarily fit for detecting faces; on the other hand, face detection only needs to detect and locate faces and not need to classify eighty kinds of object as in the COCO dataset. To solve this problem, we propose an approach based on YOLOv3 for face detection, mainly focusing on the selection strategy of a series more suitable anchor boxes and using a novel loss function. By training on the WIDER FACE [14] training dataset, the proposed face detector YOLO-face based on YOLOv3 has much better performance compared with YOLOv3. On the three tasks on the WIDER FACE validation dataset, YOLO-face has 21%, 18% and 18% accuracy improvement than YOLOv3, respectively, while maintaining fast detection speed as good as YOLOv3. Our main contributions are summarized as follows:

1. We propose a new backbone network architecture called deeper darknet which outperforms darknet-53, especially in detecting small faces.
2. A new regression loss function that mixes MSE loss and GIoU loss is proposed.
3. Anchor boxes more suitable for face detection are learnt by k-means clustering.

The rest of the paper is organized as follows. In Sect. 2, we briefly reviewed the development of object detection, face detection and the main thoughts of YOLO. Section 3 introduces our motivation and methodology. Section 4 presents our experimental results and analysis. Section 5 concludes our work and gives some advice for future work.

2 Related work

2.1 Face detection

Face detection is a sub-direction of object detection, and a large range of face detection algorithms are improved from object detection algorithms. Before deep learning introduced in this field, most object detection algorithms utilize handcraft features to complete detection tasks. Due to the insufficient ability of feature representations, researchers have to design diversified detection algorithms to compensate for these defects. Furthermore, sophisticated diagrams are always needed to accelerate the algorithms. The performance of the detectors depends heavily on the computational efficient and expression ability of the features. The representative face detectors based on these handcraft features like Viola–Jones [15], histogram of oriented gradients (HOG) [16] and deformable part model (DPM) are typical algorithms.

The first typical face detector in deep learning is the cascade CNN [17]. It uses three cascade convolutional neural networks to detect faces. The overlapped bounding boxes are removed by non-maximum suppression (NMS) [18]. MTCNN [19] also uses a similar cascading structure, but also predicts five landmarks (eyes, nose and mouth corners) to regress more accurate face positions. DenseBox [20] introduces full convolutional network (FCN [21]) in face detection. Faceness-Net [22] proposes a two-stage approach to detect faces. The first stage applies attribute-aware networks to generate response maps of different facial parts. The second stage refines the generated candidate window by a multitask convolutional neural network (CNN).

Although deep learning-based face detection algorithms have achieved significantly better performance than traditional methods, the accuracy drops remarkably when encountered with small scales and severely occluded faces. To solve these problems in unconstrained scenarios, plenty of researches were carried out and varieties of approaches were proposed. The authors of HR (hybrid resolution) [23] observed that both large context and scale-variant representations are essential, and thus applied massively large receptive fields and multitask training for variant scales to improve the detector performance. Face R-CNN [24] is based on Faster R-CNN. The method conducts online hard example mining (OHEM) and multi-scale training to optimize the model. SSH [25] is improved on SSD. It proposes a method of multi-branches that uses different layers in VGG-net [26] to detect multi-scale faces. FDNet [27] proposes Light-Head Faster R-CNN to improve the face detection performance and simultaneously uses multi-scale training and testing and deformable convolutional neural network. Face R-FCN [28] is based on R-FCN. The method uses smaller size position-sensitive RoI pooling kernel and additional smaller anchors. The method also alternates normal average pooling with position-sensitive average pooling for the last layer in R-FCN. FAN [29] introduces attention mechanism to face detection by using anchor-level attention, which improves the recall of occluded faces while maintaining low false positive rates. On observing the importance of context information, PyramidBox [30] proposed low-level feature pyramid network, PyramidAnchors and context-sensitive predict module to handle the hard face detection problem. Moreover, a data-anchor-sampling method was introduced to augment the training samples across different scales.

2.2 YOLO

Deep learning-based object detection algorithms are first introduced by R-CNN. The detector increases the performance by more than 50% compared with the DPM algorithm,

which is considered to be the best detector before. However, the detection procedure of one image costs approximately 40 seconds. Aimed to accelerate the detection speed and solve the problem of fixed input image size, SPP-Net proposed spatial pyramid pooling (SPP). Due to this structure, the detection speed is significantly faster than R-CNN. Fast R-CNN presents ROI pooling, which is faster than R-CNN in training and detection than R-CNN. Moreover, the method uses softmax instead of SVM [31] as a classifier. Faster R-CNN presented region proposal network (RPN) and handed over the selection of the region proposals to RPN. R-FCN introduces the concepts of position-sensitive score maps, uses deeper shared network layers and then accelerates detection speed remarkably.

YOLO is the first one-stage detector based on CNN.

YOLO uses a single neural network to predict bounding boxes and class probabilities directly from the input images in one evaluation. The approach divides the input image into grid cells and then directly predicts the coordinates and classification for each cell. Although the speed is several times faster than two-stage detectors, the detection accuracy is relatively lower than the counterparts. YOLOv2 [32] made many improvement including using deeper network architectures, automatically learned anchor boxes, improved loss function, multi-scale training, data augmentation, etc. The improved versions of YOLO have shown good performance on PASCAL VOC [33] and remained fast speed, which made the method can meet the need of real-time detection in practice. YOLOv3 applied a new network backbone called darknet-53 and got attractive results on the COCO dataset. We used the architecture of YOLOv3 as our basic network structure and improved in several aspects; we evaluate the method on the WIDER FACE dataset and FDDB dataset and obtain remarkably better performance. The proposed method can be used in real-time tasks of face detection in varying scales.

3 Method

Aimed to solve the detection problem of varying face scales, we propose a method called YOLO-face. We hope that YOLO-face has similar detection speed as YOLO. The architecture of YOLO-face is based on YOLOv3. We improved the model by improved backbone, anchor boxes and loss function to make it more suitable for multi-scale face detection.

3.1 Backbone

We use darknet-53 as our network backbone. The architecture is composed of a feature extraction network and three detection networks. The feature extraction network is based on darknet-53. Darknet-53 is a hybrid of darknet-19 and ResNet [34]. It has successive 3×3 and 1×1 convolutional layers and some shortcut connections. Darknet-53 includes 53 convolutional layers and is significantly larger than darknet-19. This network is much more powerful than darknet-19 and more efficient than ResNet-101 or ResNet-152. It has similar performance to ResNet-152, but is two times faster. To achieve multi-scale integration, the low-level features are merged with high-level features like feature pyramid networks (FPN). This design can make better use of many scales of the image information and thus gives better performance to multi-scale detector.

The performance of YOLOv3 decreases when detecting small-scale objects. We believe that the network structure plays a critical role in feature extraction and object locating. Since the features of small-scale targets become very small even single points on the feature map after several dimensional reductions, the subsequent layers cannot obtain enough information, which affects the efficiency of feature extraction and the accuracy of detection. Therefore, extracting enough features earlier before the features of small-scale objects on the feature map become too small to obtain enough information facilitates more accurate small object detection. Taking these factors into consideration, we improved the network structure of the original darknet-53 by increasing the number of network layers of the first two residual blocks to obtain more adequate small-scale facial features. Experiments illustrate that the improved backbone has remarkably promoted the performance for face detection. The backbone is shown in Fig. 1.

3.2 Anchor boxes appropriate for face detection

The scales and ratios of anchor boxes are very important hyper-parameters in object detection. Obviously, the shape of anchor boxes should be highly related to the detected targets. For general object detection, the shape of anchor boxes should contain all kinds of possibilities as far as possible. Intuitively, for most faces appeared in an image, the height of a face is always larger than the width. Therefore, the shape of anchor boxes for face detection should not be same as that for general object detection.

Aimed to select anchor boxes appropriate for face detection, we assemble two kinds of anchor boxes. One kind of anchors is drawn from the original YOLOv3, but is converted from flat boxes to slim boxes. Here, flat we mean that the heights of boxes are less than widths, and slim boxes are narrow and tall boxes. Following YOLOv2 and YOLOv3, the other kind of anchors is drawn from running k-means clustering on the WIDER FACE training dataset to get the dimensions of bounding boxes. The process is as follows: The first step is to empirically set the number of seeds, k, for the clustered anchor boxes and then randomly select k anchor boxes as initial clustering centers, subsequently calculating the IoU of the k anchor boxes and all other anchor boxes.

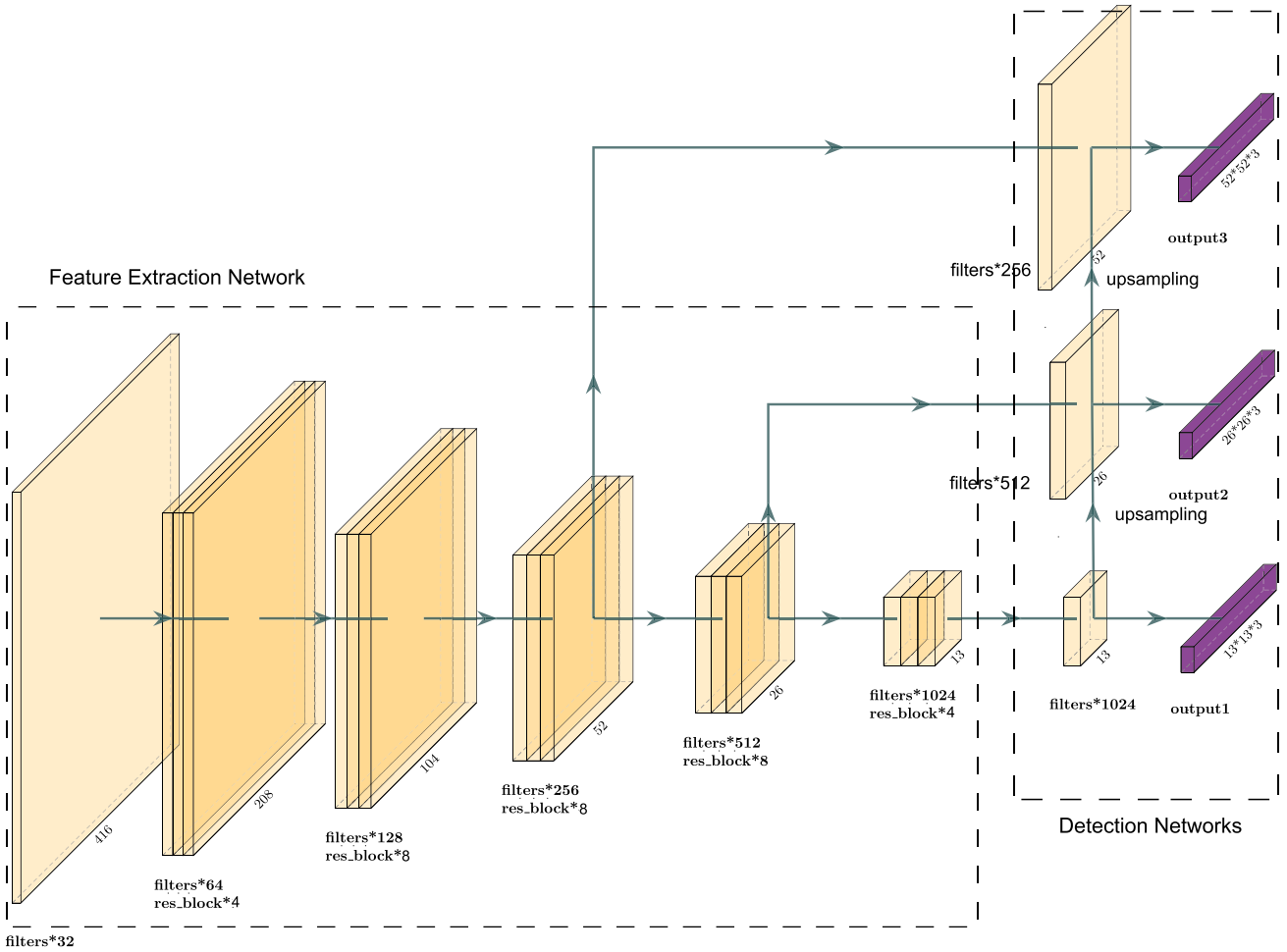


Fig. 1 Proposed architecture. The feature extraction network has 71 convolutional layers, and it reduces the scale of feature maps by the convolutional layer with stride 2. The detection network has a similar structure to FPN to extract features from different map scales

Using the IoU as the distance of anchor boxes, all the face labels are divided into k classes. Then, take the mean values of the k class anchor box sizes as the new clusters centers. Repeat the process until convergence. In our experiments, we set the initial cluster centers k to 9. The horizontal anchor boxes were transposed vertically to fit face detection. The final 9 anchor shapes are: (3, 3); (4, 5); (6, 8); (30, 61); (45, 62); (59, 119); (90, 116); (156, 198); and (326, 373).

3.3 Loss function

During training the model, YOLO optimizes a multi-part loss function that is composed of four parts, namely regression loss, confidence loss, classification loss and the loss responsible for not having any object. The proportion of the four parts sub-loss is 1:1:1:1. This weights distribution is designed for multi-class object detection. However, face detection is a problem of binary classification. To make the total loss function more suitable for face detection, we empirically revise the weights to 2:1:0.5:0.5. The final loss function is as follows.

$$L = 2 \cdot \sum L_{\text{reg}} + \sum L_{\text{objconf}} + 0.5 \cdot \sum L_{\text{noobjconf}} + 0.5 \cdot \sum L_{\text{cls}} \quad (1)$$

Here, L_{reg} is the loss of coordinate regression, L_{objconf} is the loss of confidence in anchors with objects and $L_{\text{noobjconf}}$ is the loss of confidence in anchors with no objects. L_{cls} is the loss of classification.

Traditionally, IoU of predicted location and the corresponding ground truth is commonly used as the evaluation of the optimizer, and the MSE function is used as regression loss. However, as [35] revealed in their work, there is a gap between optimizing MSE and maximizing the IoU value. In particular, it is infeasible to optimize in the case of non-overlapping bounding boxes. To address this weakness, they proposed a generalization to IoU as a new metric, namely GIoU. The new metric has a strong correlation between opti-

Fig. 2 Some face detection result examples. Left: results detected by YOLOv2. Middle: results detected by YOLOv3. Right: results detected by YOLO-face

mizing the MSE function and the metric itself. Inspired by [35], we improved the regression loss by combining the original l_n-norm error with the weighted GIoU loss. The new regression loss can be calculated as follows.

$$\mathrm{GIoU} = \mathrm{IoU} - \frac{A_\mathrm{c} - U}{A_\mathrm{c}} \tag{2}$$

$$L_\mathrm{GIoU} = 1 - \mathrm{GIoU} \tag{3}$$

$$\begin{aligned} L_\mathrm{reg} &= \sum_{c=x,y,w,h} \sum (|\Delta c_\mathrm{pred} - \Delta c_\mathrm{truth}| + \alpha \cdot L_\mathrm{GIoU})^2 \\ &= \sum (|\Delta x_\mathrm{pred} - \Delta x_\mathrm{truth}| + \alpha \cdot L_\mathrm{GIoU})^2 \\ &\quad + \sum (|\Delta y_\mathrm{pred} - \Delta y_\mathrm{truth}| + \alpha \cdot L_\mathrm{GIoU})^2 \\ &\quad + \sum (|\Delta w_\mathrm{pred} - \Delta w_\mathrm{truth}| + \alpha \cdot L_\mathrm{GIoU})^2 \\ &\quad + \sum (|\Delta h_\mathrm{pred} - \Delta h_\mathrm{truth}| + \alpha \cdot L_\mathrm{GIoU})^2 \end{aligned} \tag{4}$$

Here, A_c is the smallest enclosing convex set of the predicted location and the ground truth, α is a real-valued factor, and x, y, w and h are the locations and sizes of bounding boxes, respectively. In our model, we set the factor α to 0.1.

4 Experiments

4.1 Dataset

We use the WIDER FACE dataset as our training and evaluating dataset. WIDER FACE is a very large dataset for face detection. The data were collected from the Internet and manually cleaned. The dataset consists of 393,703 face bounding box annotations in 32,203 images. Face detection in this dataset is very challenging because of the rich variations in pose, occlusion, scale, facial expression and lighting condition. Specifically, the dataset contains many challenging face patterns such as small scales, severe occlusions and extreme poses. WIDER FACE divided the data into three categories according to the difficulties of detection, i.e., 'Easy,' 'Medium' and 'Hard,' to further evaluate the performance of the detector. The whole database is split into three subsets, namely training (40%), validation (10%) and testing (50%). WIDER FACE is arguably the most popular and widely used dataset for face detection.

FDDB (Face Detection Dataset and Benchmark) is another popular benchmark for evaluating face detection algorithms. It contains 2845 images and totally 5171 faces. We also tested our face detector on the FDDB dataset and found remarkably improvements.

4.2 Training

We trained our proposed YOLO-face on a NVIDIA GeForce GTX 1080Ti GPU using darknet. The input image size was set to 416×416, and the batch size we used was 64. We used the optimizer of SGD with momentum. The learning rate was initialized to 0.001 and exponentially decayed every 4000 steps. To make the model more versatile and generalizable, we used three kinds of data augmentation, i.e., changing the saturation, brightness and hue. We totally trained the model for 20,000 steps. The trained model was evaluated on the WIDER FACE validation dataset and the FDDB dataset. It is worth noting that we have found a dozen of wrong annotations in the WIDER FACE dataset and removed them all before training.

4.3 Results

To evaluate the effect of our proposed YOLO-face, we conducted comprehensive experiments on the WIDER FACE validation set and the FDDB dataset. Figure 2 shows some example face detection results on the dataset. We can easily see that our improved YOLO-face detector outperforms YOLOv2 and YOLOv3. The YOLOv2 and YOLOv3 detector are less capable of detecting small-scale faces and incompetent when tackling with severe occluded faces. YOLO-face shows quite better performance when facing these issues. This means that the learned anchor ratios and scales for face detection practically make sense, especially for detecting severe occluded faces and small faces. In addition, YOLO-face can detect far more faces and less possible of wrong boxes than the original detectors (see Fig. 2). We attribute this to the adaption of GIoU and the improved loss function. Notably, our improved detector increases little computational cost and thus maintains the advantage of fast detection speed.

Furthermore, to investigate YOLO-face performance thoroughly, we conducted several ablation studies. On the one hand, we evaluated the model on three sub-datasets, namely 'Easy,' 'Medium' and 'Hard.' As mentioned in Sect. 4.1, these sub-datasets are divided according to the difficulties of detection. Separately evaluating on these sub-datasets can figure out the adaptability of the detector to different scenarios. On the other hand, to disentangle the effect of the improved anchor box ratios, the GIoU metric and the improved backbone, we separately evaluated the methods by only using one improvement and their combination. Experimental results are shown in Fig. 3. We also compare our proposed YOLO-face and the original YOLOv3 on the FDDB dataset, and the results are shown in Fig. 4. In the experiments, a proposal box is considered to be positive if the IoU between it and the ground truth bounding box is larger than 0.5; otherwise, it is labeled as negative one. As shown in Fig. 2, YOLOv2 is less effective on the WIDER FACE dataset, and YOLOv3 performs much better. On the subset of 'Easy' and 'Medium,' our proposed YOLO-face outperforms both YOLOv2 and YOLOv3. On the subset of 'Hard,' only using one of our improvements, namely anchor box ratios or GIoU metric, shows slightly weak results compared to YOLOv3, but the combined leveraged a lot in performance and outperforms YOLOv3 significantly. We argue the reason for this lies in that for our introduced small-scale anchors the loss functions have difficulties for indicating the anchor boxes regressions to more precise locations, and for only using GIoU metric, the loss function accordingly increased; therefore, the training are relatively less capable of regression small-scale faces more accurately. Apparently, the two improvements of learned anchor box ratios and GIoU metric are compatible and have positive influences on each other. The proposed backbone also contributes to improving the performance. We further conducted some experiments by some other popular face detection methods, and the results are also shown in Fig. 3. To further clarify the speed of our proposed method, we compared YOLO-face with some

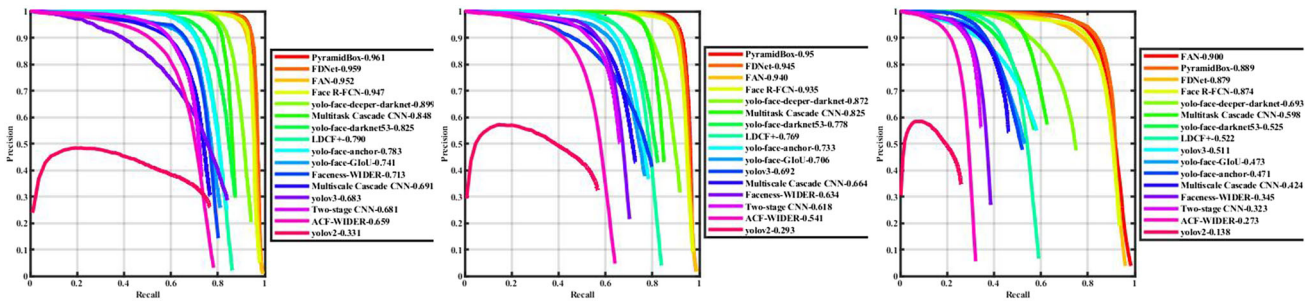


Fig. 3 Detection results on the WIDER FACE val dataset. Left: results on the 'Easy' dataset. Middle: results on the 'Medium' dataset. Right: results on the 'Hard' dataset

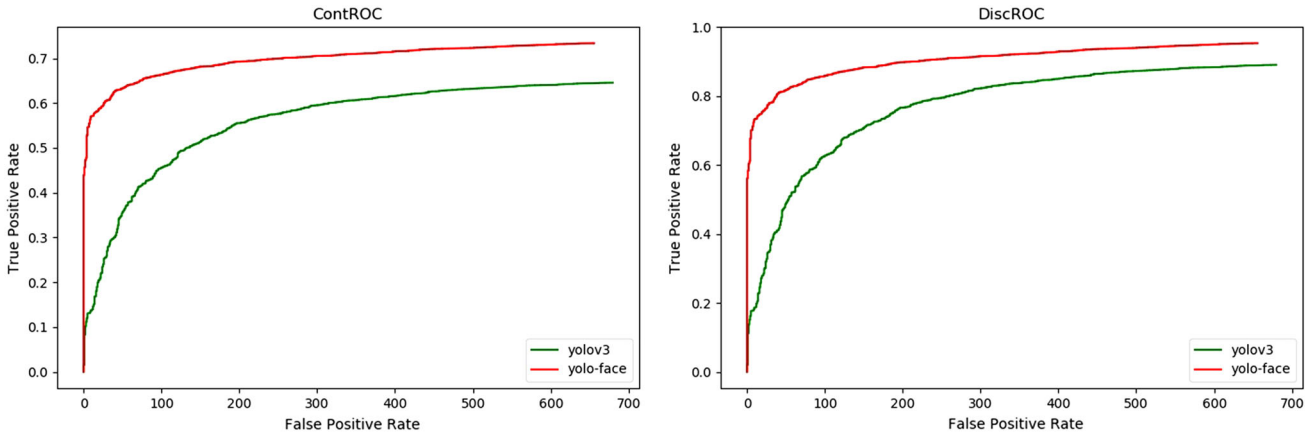


Fig. 4 Detection results on the FDDB dataset. Left: ContROC. Right: DiscROC

Table 1 Detection recall of YOLOv2, YOLOv3, our proposed YOLO-face and other face detectors on the WIDER FACE validation dataset

	Easy	Medium	Hard
LDCF+	0.790	0.769	0.522
Faceness-WIDER	0.713	0.634	0.456
Multi-scale cascade CNN	0.691	0.664	0.424
Multitask cascade CNN	0.848	0.825	0.598
Two-stage CNN	0.681	0.618	0.323
ACF-WIDER	0.659	0.541	0.273
PyramidBox	0.961	0.95	0.889
FDNet	0.959	0.945	0.879
FAN	0.952	0.940	0.900
Face R-FCN	0.947	0.935	0.874
YOLOv2	0.331	0.293	0.138
YOLOv3	0.683	0.692	0.511
YOLO-face–anchor	**0.783**	**0.733**	**0.471**
YOLO-face–GIoU	**0.741**	**0.706**	**0.473**
YOLO-face–darknet-53	**0.825**	**0.778**	**0.525**
YOLO-face–deeper darknet	**0.899**	**0.872**	**0.693**

Table 2 Detection speed of YOLO-face, FAN, Face R-FCN and PyramidBox

	Speed (fps)
FAN-1200	11 (Titan xp GPU)
FAN-400	42 (Titan xp GPU)
Face R-FCN	3 (K80 GPU)
PyramidBox	3 (Titan RTX GPU)
YOLO-face (darknet-53)	**45 (1080Ti GPU)**
YOLO-face (deeper darknet)	**38 (1080Ti GPU)**

Table 3 Detection recall of FAN-400 and YOLO-face on WIDER FACE validation dataset

	Recall
FAN-400	0.546
YOLO-face (darknet-53)	**0.693**

recently proposed detectors, as shown in Tables 1 and 2. According to [29], FAN is 4× slower than YOLO-face. We have conduct experiments with PyramidBox on same dataset, and the result shows that PyramidBox is about 10× slower. Similar results were obtained with Face R-FCN. The speed of FDNet was not reported in their paper [27], and because we cannot get the source code, we have some difficulties in comparing the speeds directly. But FDNet is a two-stage detector and it uses a multi-scale test and a large number of proposals (6000) in RPN; it is unlikely to be faster than YOLO-face. It is worth mentioning that when using FAN-400, i.e., the input size is set to 400, FAN has similar detection speed with YOLO-face, but our proposed method has better performance in the hard task of the WIDER FACE validation dataset. See Table 3. In Tables 1, 2 and 3 the bolds are experimental datas of our proposed YOLO-Face. Additionally, YOLO-face is a real-time face detector that maintains fast speed of the original YOLO method, and it is among the best face detectors that have equilibrium between performance and speed.

5 Conclusion

We use YOLOv3 as the backbone of our proposed face detector and improved in several aspects, including learning specific anchor box scales and ratios for human faces, introducing GIoU into the new loss function and using the new network structure. The improved method was trained on the WIDER FACE dataset and evaluated on the FDDB dataset and the WIDER FACE dataset. Comprehensive experiments were conducted to compare the proposed method with

some popular face detectors. Results demonstrate that the improved method can achieve equilibrium between performance and speed. The proposed method is also adaptive and flexible. It may achieve more accurate results by adaptively adjusting to specific scenarios. Some improvements may be used such as larger input image size, anchor box scales appropriate for specific scenarios and more training data. Face detection is one of the specific directions in object detection with special properties in specific scenarios. Further research can be conducted on proposing more robust and accurate face detection algorithms in various uncontrolled challenging scenarios.

Acknowledgements We wish to acknowledge Qinglin Ran, Kuo Zhang and Canwei Zhang for their advices and discussions for this work.

Funding This work is supported by the Beijing municipal education committee scientific and technological planning Project (KM201811232024, KM201611232022) and Beijing excellent talents youth backbone Project (9111524401).

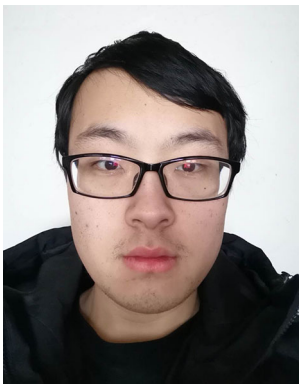
Compliance with ethical standards

Conflict of interest The authors declare that they have no conflict of interest.

References

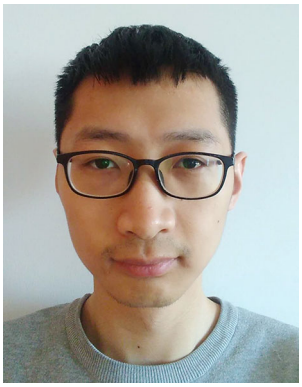
1. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: Pereira, F., Burges, C.J.C., Bottou, L., Weinberger, K.Q. (eds.) Advances in Neural Information Processing Systems, vol. 25, pp. 1097–1105. Curran Associates, Inc., New York (2012)
2. Girshick, R., Donahue, J., Darrell, T., Malik, J.: Rich feature hierarchies for accurate object detection and semantic segmentation. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2014)
3. Viola, P., Jones, M.: Rapid object detection using a boosted cascade of simple features. In: CVPR (1), vol. 1, pp. 511–518 (2001)
4. Felzenszwalb, P.F., Girshick, R.B., McAllester, D., Ramanan, D.: Object detection with discriminatively trained part-based models. IEEE Trans. Pattern Anal. Mach. Intell. **32**(9), 1627–1645 (2010). https://doi.org/10.1109/tpami.2009.167
5. He, K., Zhang, X., Ren, S., Sun, J.: Spatial pyramid pooling in deep convolutional networks for visual recognition. IEEE Trans. Pattern Anal. Mach. Intell. **37**(9), 1904–1916 (2015)
6. Girshick, R.: Fast r-CNN. arXiv preprint arXiv:1504.08083 (2015)
7. Ren, S., He, K., Girshick, R., Sun, J.: Faster R-CNN: towards real-time object detection with region proposal networks. In: Cortes, C., Lawrence, N.D., Lee, D.D., Sugiyama, M., Garnett, R. (eds.) Advances in Neural Information Processing, pp. 91–99 (2015)
8. Dai, J., Li, Y., He, K., Sun, J.: R-fcn: object detection via region-based fully convolutional networks. In: Lee, D.D., Sugiyama, M., Luxburg, U.V., Guyon, I., Garnett, R. (eds.) Advances in Neural Information Processing Systems, vol. 29, pp. 379–387. Curran Associates, Inc., New York (2016)
9. Redmon, J., Divvala, S., Girshick, R., Farhadi, A.: You only look once: Unified, real-time object detection. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2016)
10. Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.-Y., Berg, A.C.: SSD: Single Shot MultiBox Detector. In: Proceedings of the European Conference on Computer Vision (ECCV), pp. 21–37 (2016)
11. Redmon, J., Farhadi, A.: YOLOv3: an incremental improvement (2018). arXiv preprint arXiv:1804.02762
12. Lin, T.Y., Dollar, P., Girshick, R., He, K., Hariharan, B., Belongie, S.: Feature pyramid networks for object detection. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2017)
13. Chen, X., Fang, H., Lin, T., Vedantam, R., Gupta, S., Dollar, P., Zitnick, L.C.: Microsoft COCO captions: data collection and evaluation server (2015). arXiv preprint arXiv:1504.00325
14. Yang, S., Luo, P., Loy, C.C., Tang, X.: Wider face: a face detection benchmark. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2016)
15. Viola, P., Jones, M.J.: Robust real-time face detection. International Journal of Computer Vision **57**(2), 137–154 (2004)
16. Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. In: Schmid, C., Soatto, S., Tomasi, C. (eds.) International Conference on Computer Vision and Pattern Recognition (CVPR '05), vol. 1, pp. 886–893. IEEE Computer Society, San Diego (2005)
17. Cai, Z., Vasconcelos, N.: Cascade r-CNN: delving into high quality object detection. In: The IEEE Conference on Computer Vision and Pattern (2018)
18. Neubeck, A., Van Gool, L.: Efficient non-maximum suppression. In: 18th International Conference on Pattern Recognition (ICPR'06), vol. 3, pp 850–855 (2006). https://doi.org/10.1109/ICPR.2006.479
19. Zhang, K., Zhang, Z., Li, Z., Qiao, Y.: Joint face detection and alignment using multitask cascaded convolutional networks. IEEE Signal Process. Lett. **23**(10), 1499–1503 (2016)
20. Huang, L., Yang, Y., Deng, Y., Yu, Y.: DenseBox: unifying landmark localization with end to end object detection (2015). arXiv preprint arXiv:1509.04874
21. Long, J., Shelhamer, E., Darrell, T.: Fully convolutional networks for semantic segmentation. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2015)
22. Yang, S., Luo, P., Loy, C.C., Tang, X.: Faceness-net: face detection through deep facial part responses. IEEE Trans. Pattern Anal. Mach. Intell. **40**(8), 1845–1859 (2018). https://doi.org/10.1109/TPAMI.2017.2738644
23. Hu, P., Ramanan, D.: Finding tiny faces. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)(2017)
24. Wang, H., Li, Z., Ji, X., Wang, Y.: Face R-CNN (2017). arXiv preprint arXiv:1706.01061
25. Najibi, M., Samangouei, P., Chellappa, R., Davis, L.S.: SSH: single stage headless face detector. In: The IEEE International Conference on Computer Vision (ICCV) (2017)
26. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition (2014). arXiv preprint arXiv:1409.1556
27. Zhang, C., Xu, X., Tu, D.: Face detection using improved faster RCNN (2018). arXiv preprint arXiv:1802.02142
28. Wang, Y., Ji, X., Zhou, Z., Wang, H., Li, Z.: Detecting faces using region-based fully convolutional networks (2017). arXiv preprint arXiv:1709.05256
29. Wang, J., Yuan, Y., Yu, G.: Face attention network: an effective face detector for the occluded faces (2017). arXiv preprint arXiv:1711.07246
30. Tang, X., Du, D.K., He, Z., Liu, J.: Pyramidbox: a context-assisted single shot face detector. In: Proceedings of the European Conference on Computer Vision (ECCV), pp. 797–813 (2018)
31. Burges, C.J.C.: A tutorial on support vector machines for pattern recognition. Data Min. Knowl. Discov. **2**(2), 121–167 (1998)

32. Redmon, J., Farhadi, A.: Yolo9000: better, faster, stronger. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2017)
33. Everingham, M., Eslami, S.M.A., Van Gool, L., Williams, C.K.I., Winn, J., Zisserman, A.: The Pascal visual object classes challenge: a retrospective. Int. J. Comput. Vis. **111**(1), 98–136 (2014)
34. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2016)
35. Rezatofighi, H., Tsoi, N., Gwak, J., Sadeghian, A., Reid, I., Savarese, S.: Generalized intersection over union: a metric and a loss for bounding box regression. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2019)



Weijun Chen received the Bachelor's degree from Beijing Information and Science Technology University. He is a Master Candidate at the Computer School in Beijing Information and Science Technology University. His research interests include object detection and generative adversarial networks.

Hongbo Huang is an associate professor at the Computer School of Beijing Information Science and Technology University, China. And he is also the director of Institute of Computing Intelligence of Beijing Information Science and Technology University. He received his Ph.D. degree in Control Science and Engineering in 2015 from University of Science and Technology Beijing. His research interests include computer vision, machine learning and video semantic analysis.



Shuai Peng received the Bachelor's degree from Beijing Information and Science Technology University. He is a Master Candidate at the Computer School in Beijing Information and Science Technology University. His research interests include face recognition and human pose estimation.

Changsheng Zhou is an associate professor at the Computer School of Beijing Information Science and Technology University, China. He received his B.S. and M.S. degree from Harbin Engineering University, Harbin, China, and Ph.D. degree from Beihang University, Beijing, China. His research interests include data mining and image processing.

Cuiping Zhang received the M.S. and Ph.D. degree from Beijing Jiaotong University, Beijing, China. Since 2017, she has been an assistant professor at the Computer School of Beijing Information Science and Technology University, China. She also holds a postdoctoral position currently at Beijing Jiaotong University. Her research interests include the traffic big data analysis, machine learning, algorithms and application in intelligent transportation systems.