Pattern Analysis and Applications (2021) 24:1745–1755
https://doi.org/10.1007/s10044-021-01026-3

THEORETICAL ADVANCES

TRFH: towards real-time face detection and head pose estimation

Shicun Chen[1] · **Yong Zhang**[1] · **Baocai Yin**[1] · **Boyue Wang**[1]

Received: 31 December 2020 / Accepted: 11 August 2021 / Published online: 12 September 2021

Abstract
Nowadays, face detection and head pose estimation have a lot of application such as face recognition, aiding in gaze estimation and modeling attention. For these two tasks, it is usually to design two different models. However, the head pose estimation model often depends on the region of interest (ROI) detected in advance, which means that a serial face detector is needed. Even the lightest face detector will slow down the whole forward inference time and cannot achieve real-time performance when detecting the head pose of multiple people. We can see that both face detection and head pose estimation need face features, so a shared face feature map can be used between them. In this paper, a multi-task learning model is proposed that can solve both problems simultaneously. We directly detect the location of the center point of the bounding box of face; at this location, we calculate the size of the bounding box of face and the head attitude. We evaluate our model's performance on the AFLW. The proposed model has great competitiveness with the multi-stage face attribute analysis model, and our model can achieve real-time performance.

Keywords Multi-task · Face detection · Head pose · Anchor Free

1 Introduction

Face detection and face attribute analysis are important and challenging tasks in computer vision. This paper addresses the face detection and head pose estimation problems which has many applications such as face recognition and human attention modeling. In recent years, due to the high efficiency of CNN, deep learning has achieved good results in computer vision tasks. In face detection and analysis, a large number of efficient models are proposed to solve different tasks, such as face recognition [2, 21], face age estimation [15], face landmarks detection [31, 32], head pose estimation [20, 30] and so on. For these tasks, different models are designed to solve the corresponding tasks and have achieved good accuracy and performance. However, these tasks all require face bounding box to be detected in advance. When we connect face detection models with these head pose estimation models in series, the performance of these models will be less efficient. In particular, the head posture of multiple people is estimated at the same time, such as the modeling of students' attention in class. It is difficult to achieve real-time pose estimation for multiple people at the same time because of the need to calculate each person's head posture separately.

In the task of head pose estimation, both traditional methods and deep learning methods need to detect the ROI of the face, and some even need to detect the landmarks of the face. There are a lot of unnecessary calculations, which increase the computational complexity of the whole model and the overall inference time. Recently, MTCNN, MaskFace and Retinaface [3, 31, 32] realize the multi-task learning model of face detection and face landmarks detection through shared convolution feature map. The inference time of the face landmarks detection model which relies on the pre-detection of face ROI is improved, and the accuracy of face detection is also improved. More recently, Retinaface etc [3] has demonstrated that multi-task learning can increase overall accuracy by adding additional supervision. In addition, it is a shared feature mapping. In the same amount of tasks, the repetitive feature extraction is reduced and the calculation speed is accelerated. In this paper, a multi-task learning model combining face detection and head pose estimation is proposed to reduce the overall time spent in the task of face detection and head pose estimation, which can be better applied in real-time.

✉ Yong Zhang
zhangyong2010@bjut.edu.cn

[1] Beijing University of Technology, Beijing, People's Republic of China

Our network is intended to be applied to face detection and head pose estimation in the classroom where requires high real-time performance. Inspired by Centernet [33], this paper proposes an anchor free single-stage face detection model with head pose estimation. It reduces the overall computational complexity and avoids the detection of the face bounding box before the head pose estimation. The head pose is a three-dimensional vector containing yaw, pitch and roll. For the head pose estimation of a face in an image, most of the faces in the image need to be detected first, and then the head pose estimation of face ROI is carried out. Our model can directly calculate the 3D vector of the head pose during the detection of the face(as shown in Fig. 1). The head pose estimation task is a regression task, but the direct application of regression to solve the head pose does not perform well on large-scale data. Inspired by Hopenet and FSA-Net, we first performed a rough classification of the head pose, and then performed a fine regression.

In general, this paper proposes a multi-task learning model, which can detect faces and estimate head posture simultaneously. Our contribution can be summarized as follows:

(1) An end-to-end multi-task learning model is proposed, which can obtain the head pose estimation while detecting the face. By using the shared feature map, the overall computing time of head pose estimation is reduced, it can achieve real-time head pose estimation for multiple people.
(2) In the head pose estimation process, we did not directly return to the head attitude angle, but made a rough classification of the head attitude angle, and then we made a fine exception to get our head attitude angle. It makes the model more robust.
(3) An anchor-free one-stage face detection model is proposed. For a single task of face detection model, we lose a small amount of accuracy but get a huge improvement in model speed.

2 Related work

Face detection and its attribute analysis have always been a key challenge in computer vision. Many excellent methods have been put forward to solve these tasks. In this section, we will review the previous methods from three aspects: face detection, head posture estimation and multi-task learning.

2.1 Face detection

Face detection is to find the position of the face in the image, is a detailed branch of the object detection task. In the early face detection algorithm, the method of template matching was used. A face template is used to compare with each position of the image to determine whether there is a face here, for example Rowley propose [18, 19]. Viola and Jones proposed to construct a detector using a simple Haar-like feature and a cascade of adaboost classifier [25]. Compared with the previous method, the detection speed is greatly improved and maintains good accuracy. A number of studies have shown that this detector can significantly reduce visual changes in human faces in real-world applications, even with more advanced features and classifiers [29, 32]. Compared with DMP model [12, 27], it shows good performance and has good detection effect on distorted, gender multi-pose and other faces. However, its biggest problem is that it is too slow to be applied in engineering.

Later, with the success of the convolutional neural network in the classification problem [6, 8, 23], it was quickly applied to face detection problem, which greatly exceeded the previous framework in accuracy. Most of the current face detection models are evolved from object detection models,

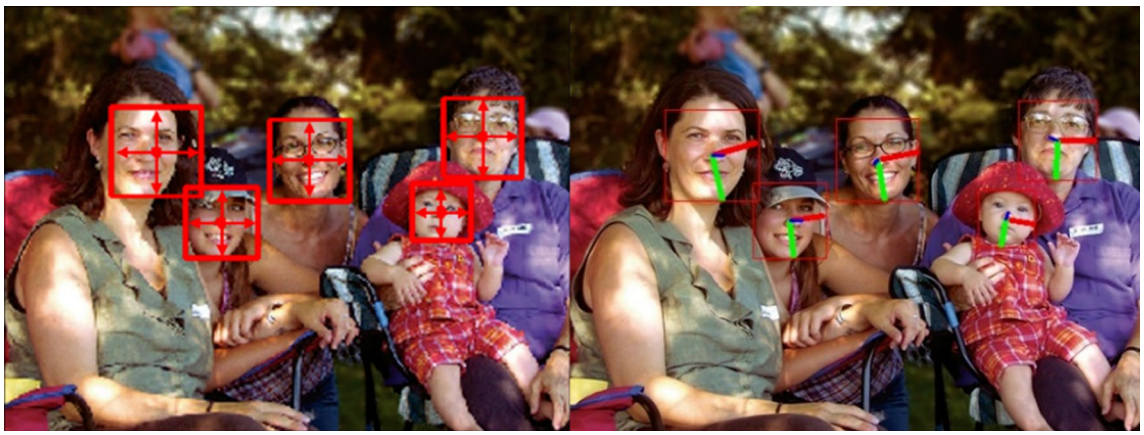

Fig. 1 We detect the face as a point and calculate the size of the face bounding box and the head pose of the face directly at this point

which can be divided into one-stage methods and two-stage methods. Two-stage methods [26] adopting "proposal and refinement", which have high accuracy but slow speed. One-stage [13] adopts intensive sampling of face position and scale, which will lead to the imbalance of positive and negative samples in the training process. To solve this problem, sampling and re-setting is widely used. Compared with two-stage method, one-stage shows excellent performance, but its relative accuracy is slightly lower than two-stage method.

Anchor was widely used in the one-stage and two-stage target detection network, and it was proposed in Faster R-CNN [17]. In recent years, anchor-based target detection has made great progress and proved its effectiveness. However, anchor needs a large number of samples, which aggravates the imbalance of positive and negative samples in the original face detection task. In recent years, with the development of the anchor-free object detection network [33], its performance is getting closer and higher than that of the anchor-based network.

2.2 Head pose estimation

Head pose estimation has been a widely studied problem in computer vision, and there are many differences in the methods. In some of the literature [14, 22], they used pose templates to match real faces to get head poses. Detector arrays [19] were also a popular way to train multiple detectors to detect different head positions. All of these methods consume huge computing resources.

With the success of face landmarks detection [24, 31], face landmarks have become popular to be used to evaluate head pose. Given a set of 2D face landmarks to calculate the 3D head attitude angle such as POSIT [1]. However, the head pose estimation method based on landmarks needs to detect the landmarks of the face, and the landmarks of the face are dense. In some low-resolution images or for small faces, some experts are often unable to demarcate the key points of the face.

Others consider using depth information to assess head pose. Fanelli et al. [4] exploited discriminative random regression forests for head pose estimation with depth images. But, this requires additional device overhead. With the development of deep learning, some end to end deep learning models are gradually studied. Hopenet et al. [20, 30] adopted the deep learning method to transform the regression task of the head pose into the classification task, so as to directly obtain the head pose and make the model more robust. Whether the head pose estimation method based on landmarks or the direct estimation method from a single image, they all need to connect other models to provide additional help. When it is necessary to estimate the head posture of multiple people at the same time, the overall computational complexity will increase exponentially.

2.3 Multi-task methods

Multi-tasking learning is the combination of multiple single tasks into a single model. In recent years, some work has demonstrated that multi-tasking learning can achieve better performance [3, 31, 32] than single-task learning model. They used CNNS to simultaneously detect faces, landmarks, etc. In Hyperface [16], the authors detects faces, landmarks, headpose and gender in the images at the same time. But, it is inefficient and difficult to use in industry. MTCNN uses image pyramid and cascading CNN to predict the position of face bounding box and face landmarks points. Some recent methods use the feature pyramid approach to detect faces of different scales. SSD [10] and so on add additional regression heads for landmarks detection. Retinaface, SSH [3, 13] added semantic models to increase the visual field of perception of the model. Meanwhile, Retinaface proved that this kind of multi-task learning provides additional self-supervision to improve the ability of the model. Then, Maskface proposed RoiAilgn [5] for landmarks detection to optimize the accuracy of landmarks detection and improve the accuracy of the face detection model. Multi-task learning has high efficiency. Self-supervised training can be carried out through inter-task correlation to improve the model. However, there are few researches on multi-task learning model for face detection and head pose. Although hyperface solves multi-person face-related tasks, including head pose, its efficiency is very low.

3 Towards real-time face detection and head pose estimation

3.1 Architecture

In practical applications, such as classroom students' attention modeling, most of the faces are small target faces, and the head pose estimation of multiple people is carried out. So, we need a model to detect small target faces and estimate multi-person head pose in real time. Our model is an one-stage anchor-free multi-task learning model. The position and head posture of the face frame can be obtained directly from RGB images. We refer to Centernet, which is an anchor free target detection model with good accuracy and performance in target detection tasks. Centernet detects the center of the object directly and regresses the size of the box at this point. Centernet is friendly to small targets because the training uses a Gaussian distribution on the sample and targets are detected as points, that can solve the problem of detect small face. We not only need to detect small target face, but also need certain ability to detect large target face. Many of the most advanced works based on anchors have built different structures to detect faces of different sizes.

High-level features are used to detect large faces, while low-level features are used to detect small faces. We also build a feature pyramid to detect faces of different scales. For different scale feature pyramids, we assign different scales of face for supervised training. Traditional FPNs include bottom-up, top-down and lateral connections, which is an effective structure for spatial integration. But its connection is linear and simple without good fusion of semantic information between layers. So we use DLA34 as our backbone network because it contains a similar structure to FPNs. Different from FPNs, the feature fusion design of shallow layer and deep layer is more complex. More semantic information is fused between layers. By designing different face sizes for different layers, the ability of the model is effectively improved. A rough outline of the model as shown in Fig. 2.

In order to increase the visual field of perception of the model, we design the semantic model after DLA-34 different step size output. Before the semantic model, we add a 1*1 convolutional layer to unify the feature map into 256 channels. The semantic model is designed with reference to Retinaface, as shown in Fig. 3. We set the input channel of the semantic model to 256 and then feed it into two branches. Three feature maps of 128, 64 and 64 channels are obtained, and finally the three feature maps are spliced into 256 channels as the output of the semantic model. After the semantic model, we get our shared feature map. Then, we design the 1*1 convolutional layer of different channels to match our different tasks, such as face classification as channel 1.

3.2 Multi-task loss

For the supervision training of face detection at different scales, we minimize the following multi-task loss

$$L = L_{det} + L_{offset} + L_{size} + L_{head} \tag{1}$$

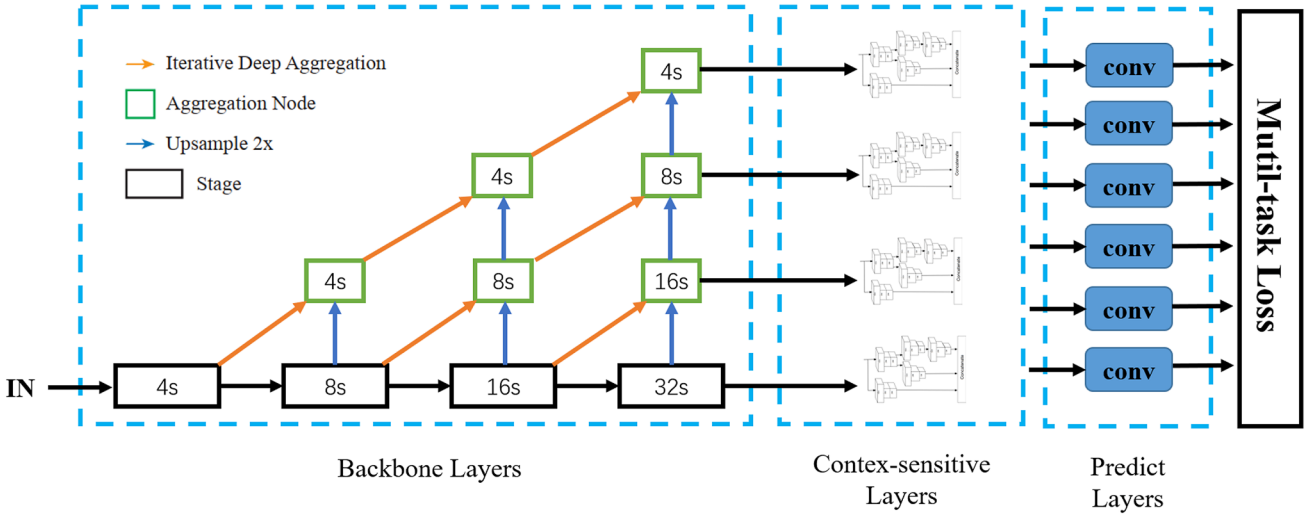

Fig. 2 Rough outline of the model.We have chosen DLA-34 as our backbone. Then, we connect context-sensitive model after output of different scales to increase the visual field of perception of the model. We set up different convolution heads for different tasks. We use multilosses to constrain our model

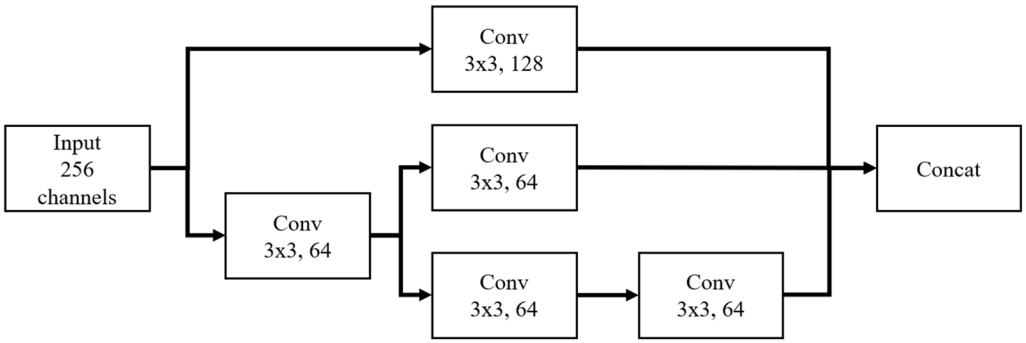

Fig. 3 Context module. The information in the convolutional layer includes the size of the convolution kernel and the output channels

, where L_{det} is the loss of face bin class, L_{offset} is bounding box location regression loss, L_{size} is the loss of face bounding box size, L_{head} is the loss of head pose.

For each face bounding box, we calculate the coordinates of its center point as the point we want to detect. (x_1, y_1, x_2, y_2) is coordinates of the upper left and lower right corner of the face bounding box. And, the keypoint can be $\left(\frac{x_1+x_2}{2}, \frac{y_1+y_2}{2}\right)$. We do pixel level point classification. L_{det} is focal loss (Formula 2), for each ground truth keypoint p, we compute a low-resolution equivalent $\tilde{p} = \left\lfloor \frac{p}{R} \right\rfloor$, where R is the stride of the output. We then splat all ground truth keypoints onto a heatmap $Y \in [0,1]^{\frac{w}{R} \times \frac{H}{R}}$using a Gaussian kernel $Y_{xyc} = \exp\left(-\frac{(x-\tilde{p}_x)^2+(y-\tilde{p}_y)^2}{2\sigma p^2}\right)$, where is an object size-adaptive standard deviation. The heatmap under asynchronous length is shown in Fig. 4.

$$L_{det} = \frac{1}{N} \sum_{xy} \begin{cases} \left(1-\hat{Y}_{xy}\right)^{\alpha} \log\left(\hat{Y}_{xy}\right) & \text{if} \quad \hat{Y}_{xy} > 0.9 \\ \left(1-Y_{xy}\right)^{\beta} \left(\hat{Y}_{xy}\right)^{\alpha} \log\left(1-\hat{Y}_{xy}\right) & \text{if} \quad \hat{Y}_{xy} < 0.8 \end{cases} \tag{2}$$

where α and β are hyper-parameters of the focal loss, and N is the number of keypoints in image I. The normalization by N is chosen as to normalize all positive focal loss instances to 1. We use $\alpha = 2$ and $\beta = 4$ in all our experiments.

For keypoint, we do not simply multiply the step size by the coordinates of the heat map to get the coordinates of the original image directly, which is obviously not accurate enough. In the process of transforming, the image coordinates into heatmap coordinates, there must be some loss. We calculate the real point coordinates and the offset map to heatmaps by the loss as follows:

$$L_{offset} = \frac{1}{N} \sum |P - (p - \tilde{p})| \tag{3}$$

where P is the predictive value.

The length and width of the face frame are directly obtained by regression. L_{size} as defined by the following formula:

$$L_{\text{size}} = \frac{1}{N} \sum \left|\hat{S}_p - s\right| \tag{4}$$

where s is the truth size of the bounding box.

For head pose estimation, we minimize the following mutil loss:

$$L_{\text{head}} = H(y, \hat{y}) + \alpha \text{MSE}(y, \hat{y}) \tag{5}$$

where H is the cross-entropy loss, MSE is the squared error loss functions. y is the true label, $\hat{y}$ is the predicted value. Section 3.3 describes the details.

3.3 Headpose estimation

Generally speaking, the head pose estimation belongs to the regression task. The three vectors of the head pose can be obtained through direct regression. But this approach does not work very well for large scale data. Inspired by Hopenet

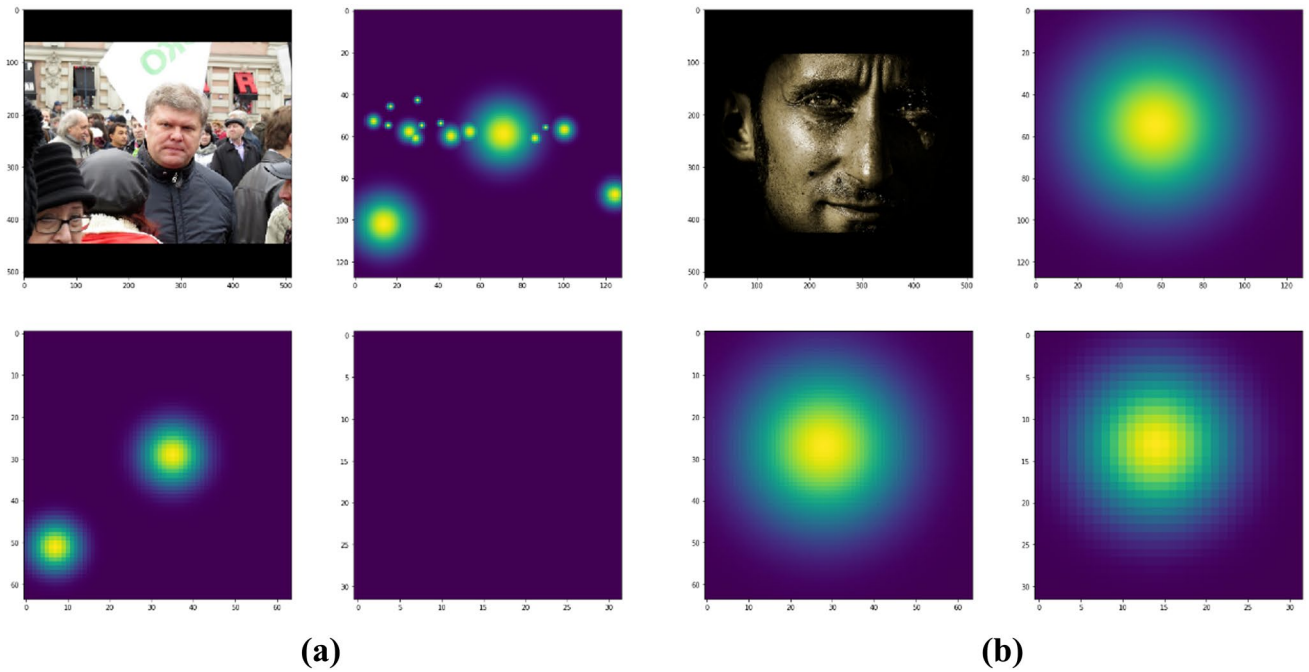

Fig. 4 Different stride of the output (4, 8, 16) face bounding boxes of keypoint in heatMap Gaussian distribution

and FSA-Net, we set up three classifiers corresponding to three different Euler angles of head attitude respectively to make a rough positioning of the angle. We only detect the three head attitude angles with an angle of −99° to 99°. In general, most of the angles of head posture are concentrated in this range. We divide it into a category every 3°, a total of 66 categories for each head attitude angle. In the loss function, we use following loss to calculate the classification loss:

$$H(y,\hat{y}) = \frac{1}{N}\sum_{i} -\left[y_i \log\left(\hat{p}_i\right) + \left(1-y_i\right)\log\left(1-\hat{p}_i\right)\right] \quad (6)$$

We calculate the angle based on the expected value of the classification. The final angle is obtained by multiplying the confidence level of classification with the corresponding category. We use MSE to calculate the probable losses:

$$MSE = \frac{1}{n}\sum_{i}^{n}\left(\hat{y}_i - y_i\right)^2 \quad (7)$$

Then, we add the classification loss to the regression loss to get the loss of our head pose estimation, as shown in Formula 5. Where α is a hyperparameter, we set it to 0.1 in our model.

4 Experiments and results

We use the open dataset AFLW(Annotated Facial Landmarks in the Wild) [11] in our training. In the face detection experiment, we test the accuracy of the model not only on AFLW but also on AFW [34], FDDB [7], and Pascal face [28] datasets. The other experiments are carried out on the AFLW dataset. In addition to evaluating the performance of our model on public datasets, we also evaluate the actual effect of our model in our practical application process of classroom student's attention modeling.

4.1 Training

4.1.1 The data processing

During training, images are resized with a randomly chosen scale factor between 0.6 and 1.3. Then, we randomly flip the image with a 50% probability and distort the color. Then, we cropped the random area of the image into a 512*512 resolution image. If the cropped image does not contain any bounding box of face, we perform normal cropping on the image to include at least one bounding box of face. This enables us to include more positive samples in training batches. In ALFW dataset, samples greater than 99° and less than −99° from yaw, pitch and roll angles are excluded.

4.1.2 Training details

We train our model by using the SGD optimizer with the momentum of 0.9 and the weight decay of 0.0001. In the AFLW dataset, the batch size is 16. Our backbones is pretrained on the ImageNet dataset. Our initial learning rate is set at 0.001. At the 10th epoch, we set the learning rate as 0.01, and after 30 epochs, we adopt the step attenuation strategy. When our validation loss is not decreasing, we multiply the learning rate by 0.1. We set the minimum learning rate to 0.00001.

4.2 Results of face detection

We evaluate the accuracy of face detection on AFW, AFLW, FDDB and Pascal face datasets. AFLW is a massive Facial database with multiple poses and perspectives. The image is from Flickr crawl. There are 21,997 pictures and 25,993 faces. Most of them are RGB images, but a few are grayscale. Among them, 59% were female, and 41% were male. We use 60% of the ALFW dataset for training and the remaining 40% for testing. The AFW dataset was collected from Flickr, and the images in this dataset contain large variations in appearance and viewpoint. In total, there are 205 images with 468 faces in this dataset. The FDDB dataset consists of 2,845 images containing 5,171 faces collected from news articles on the Yahoo website. This dataset is the most widely used benchmark for unconstrained face detection. The PASCAL faces dataset was collected from the test set of the PASCAL person layout dataset, which is a subset of PASCAL VOC. This dataset contains 1335 faces from 851 images with large appearance variations.

In the anchor-based method, to get more positive samples, samples with ROI greater than 0.5 are generally selected as positive samples and those with ROI less than 0.3 as negative samples. In order to obtain the same effect as the anchor-based method, increase the number of positive samples in the training process and balance the proportion of positive and negative samples, we take the Gaussian distribution value greater than 0.9 as the positive sample and the value less than 0.8 as the negative sample. Samples with Gaussian values between 0.8 and 0.9 are ignored. The Centerpoint equal to 1 in Centernet is compared as a positive sample, and the rest were all negative samples in our comparison experiment. Through Fig. 5, it is found that our method can effectively improve the accuracy of the model.

Our model will be used for real-time human attention modeling. We choose some models with both accuracy and model inference speed to compare, such as MTCNN, SSH and Retinaface et.al [3, 9, 13, 25, 32, 34]. In the input process, we do not stack image pyramid because in the actual application process, the real-time performance of the model would be greatly reduced. We believe that this operation

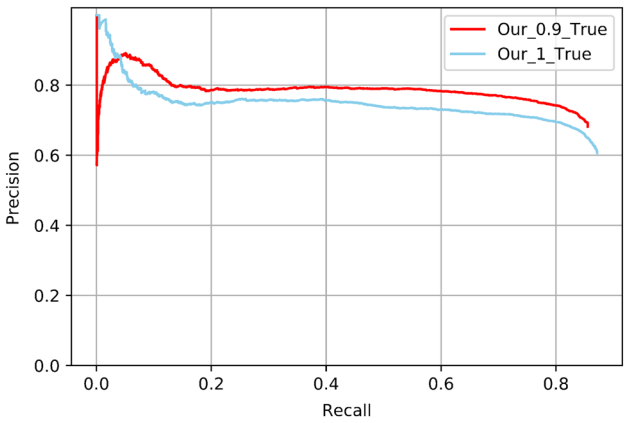

Fig. 5 Experimental results of model with different samples

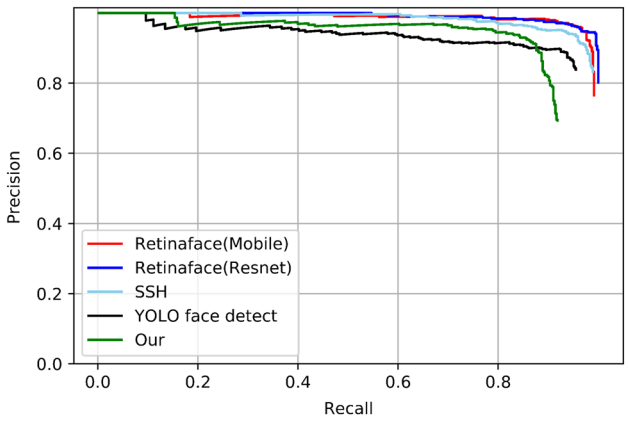

Fig. 7 Precision-recall curves on the AFW dataset

only has a certain effect on the refresh accuracy of the datasets. Figures 6, 7 and 8 show the precision-recall curves of different detectors corresponding to AFLW, AFW and Pascal face datasets, respectively. Figure 9 shows the Receiver Operating Characteristic (ROC) of the models on the FDDB dataset.

From the experimental results on several datasets, we can see that our model's performance has achieved the state-of-the-art. On the FDDB dataset, some methods use FDDB as training data in a 10-fold cross-validation fashion. And our method does not use the FDDB dataset for training. Because our model selection is friendly to small targets, while AFLW datasets and other datasets are mostly large target faces. Although our model is trained on the AFLW dataset, it can be found that our model still pays more attention to small target faces than most models, so the effect on the AFLW dataset is not very good. In some subsequent subjective evaluations, it is found that our models performed better than these models in the classroom, where most small target faces are detected.

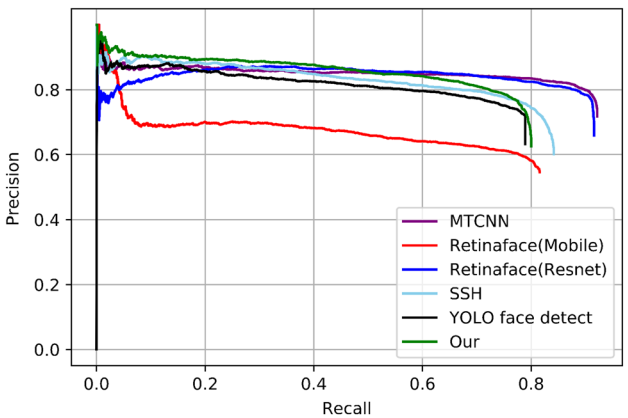

Fig. 6 Precision-recall curves on the AFLW dataset

4.3 Results of head pose estimation

We evaluate the attitude errors of the three head attitude angles as a contrast to some head pose estimation models. Since all the methods we compare require human faces as input, we select the face detected by our model to send into the comparison method, so as to ensure the same data we use in the comparison process and the fairness of the comparison. We calculate the mean absolute error of each three head pose angles, and it turns out that smaller is better. The experimental results are shown in Table 1 and Fig. 10. The blue line indicates the direction the subject is facing; the green line for the downward direction while the redone for the side.

4.4 Inference efficiency

We calculate the model inference time of the face detection model and the head pose estimation model on the test dataset, respectively. The inconsistent image size in the dataset will lead to inconsistent reasoning speed of the model. Therefore, we calculate the overall running time of the model on the test dataset, and calculate the average time consumed by each image, as shown in Table 2. We calculate the reasoning speed of the model on Tesla P100 GPU, and make statistics on some advanced face detection models with fast reasoning speed. We compare the reasoning speed of the single face detection model and face detection plus head pose evaluation. From the table, we can see that the speed of our model is basically the same as that of the current one-stage face detector, but with the head pose estimation model, we can see that our model is better than the multi-step head pose model.

We also compare the model inference time with different numbers of people in an image. We use images of the same size but with different numbers of people to prevent the impact of image size on the speed of the model,

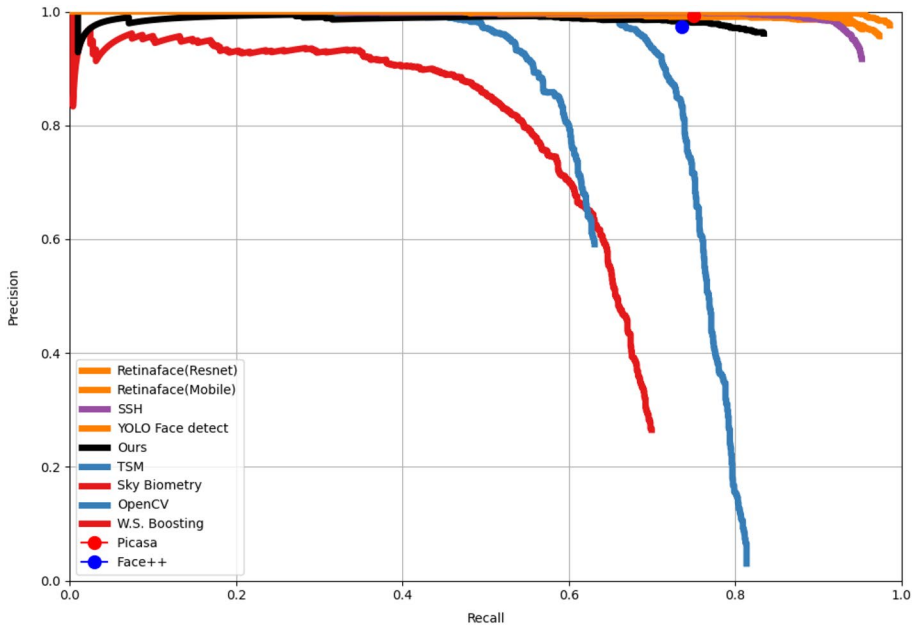

Fig. 8 Precision-recall curves on the Pascal Face dataset

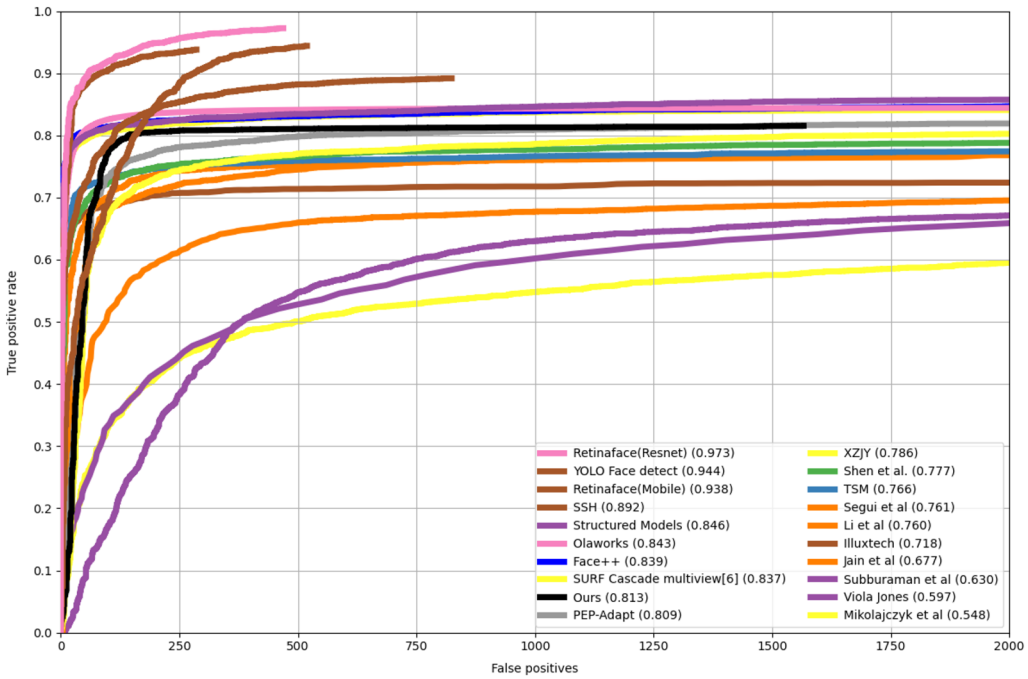

Fig. 9 ROC curves of on the FDDB dataset

because we only compare the model reasoning speed with different numbers of people. We divide the experiment into four groups, with 17 and 34 more closely related to the number of students in the classroom. In addition, we compared the reasoning time of the model with 1 person and 5 people. Each group of experimental pictures for 10, each group repeated the experiment three times to take the average value, to ensure the fairness of the results, as shown in Table 3. It can be seen that the reasoning time of the multi-step head pose estimation model increases with the increase of the number of people, while our model basically does not change. It can be seen that our model has a huge advantage when dealing with multiple people.

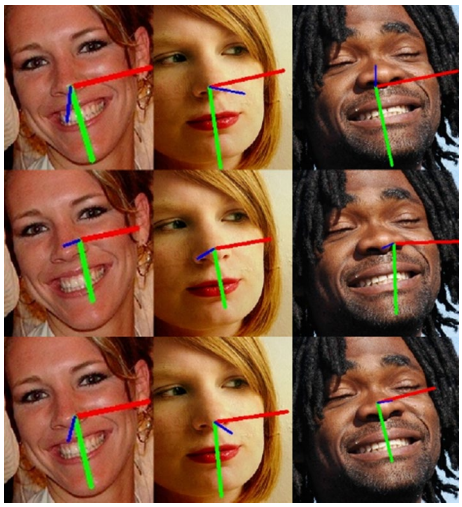

Fig. 10 Pose estimation on the AFLW dataset. From top to bottom, they are ground truth, results of Hopenet and our model

Table 1 Mean average error of Euler angles across different methods on the AFLW dataset

Method	Yaw	Pitch	Roll	MAE
Dlib	23.153	10.545	13.633	15.777
Hopenet	8.84	15.41	14.1	12.78
Our	5.49	23.81	17.26	15.52

Table 2 Inference time(ms) of different models

Method	Inference time
Retinaface(mobile)	0.2
yolo_face_dect	0.017
SSH	0.12
Retinaface(mobile)+hopenet	0.32
yolo_face_dect+hopenet	0.137
SSH+hopenet	0.24
Our	0.071

Table 3 Inference time(ms) of different models with different numbers of people

Nums of people	SSH+Hopenet	YOLO_face+Hopenet	Our
1	0.152	0.156	0.015
5	0.183	0.18	0.016
17	0.32	0.252	0.015
34	0.58	0.46	0.017

Table 4 Frames per second of different models in classroom surveillance video

Method	Frames per second
Retinaface(mobile)+Hopenet	9
MTCNN+Hopenet	1.8
SSH+hopenet	3.9
Our	40.69

4.5 Results in classroom student's attention modeling

The classroom containing a large number of small target faces is a suitable application of our model. So, we also do some subjective precision comparisons on our classroom videos, and some model comparisons on the speed of reasoning. Although most of the models use feature pyramid to optimize the detection of faces at different scales, due to the existence of anchor, when the size difference between the target face and anchor is large, it cannot be detected well. And, most models do not make special anchor settings for small target faces. Our model is based on Centernet, small target friendly. And, the students in the classroom are mostly small targets, so subjectively, our model for face detection of students in the classroom is quite effective, as shown in Figs. 11, 12.

Then, we count the frames of face detection and head pose estimation of different models in actual classroom(see Table 4). In order to be fair, we ensure the same operating environment during the test. Our test uses a single Tesla p100 GPU with a CPU of Intel Xeon E5-2620. It can be seen that our model can easily achieve the real-time detection effect in the classroom with a large number of people. The multi-step head pose estimation model needs to re-extract each person's feature map. Therefore, the multi-step head pose estimation model is difficult to achieve real-time effect in a large number of peoples. Our model can reduce the computation process of extracting feature again in head pose estimation by sharing the feature map with face detection task. It can be seen that our model has a great advantage in dealing with this multi-person head posture.

5 Conclusions

In this paper, an effective multi-task learning model is proposed, which combines face detection with head pose estimation to detect and analyze small target faces. Through the shared feature map, we can get the position of the bounding box and the pose angle of the head at the same time. It reduces the steps of detecting the region of interest before

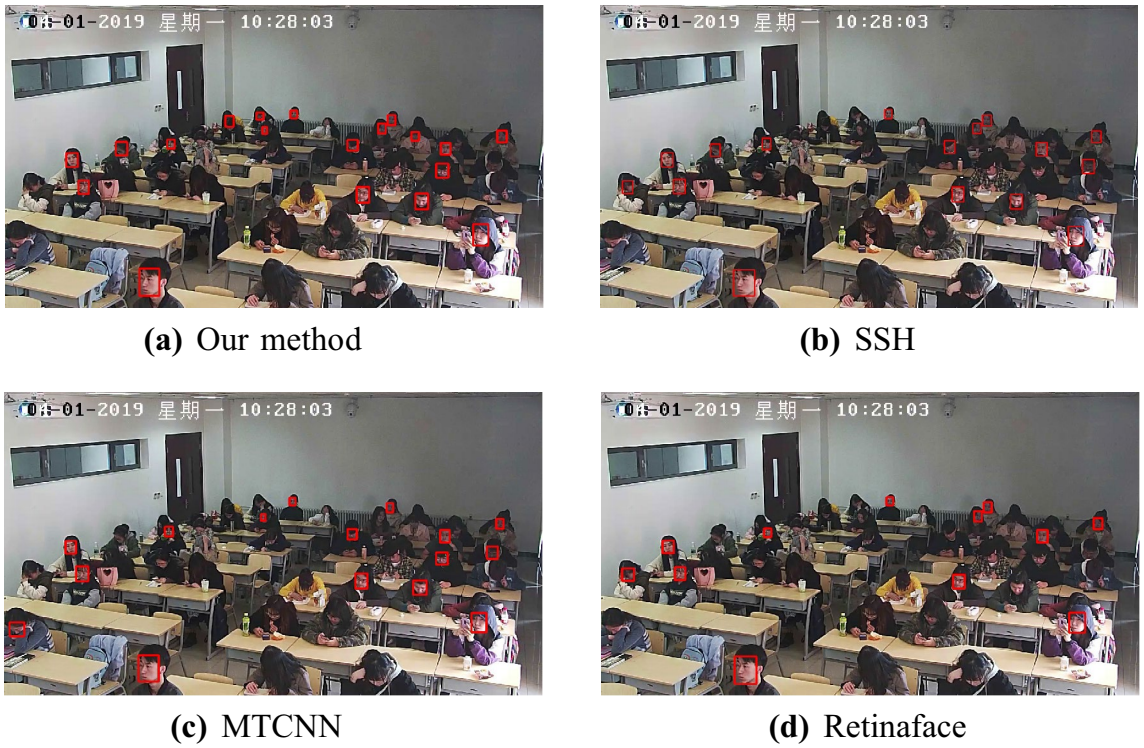

Fig. 11 Different face detection models for student face detection in classroom. It can be seen that our model detects more small target faces than other models

estimating the head pose from the field and eliminates the overall computational complexity. The efficiency of single person head pose estimation is slightly improved, and for multi-person head pose estimation, our model can still work in real time. It is very helpful for the application of multi-person pose estimation, such as the students attention modeling in the classroom. We also estimation student head pose in real classroom, and the results are remarkable. Our model can be better applied in practice. Our preliminary experimental results show that our method is more suitable for front-end real-time analysis systems and can more efficiently estimation the head pose of a large number people. In the future, we still need to make great efforts in precision and speed of model reasoning.

Fig. 12 Illustration of our approach on a classroom monitoring system

Funding This work was supported in part by the National Natural Science Foundation of China under Grant No.62072015, U19B2039, U1811463, 61906011, 61632006, 61672071, in part by the National Key R&D Program of China under Grant 2020YFB1600700, 2018YFB1600903.

References

1. DeMenthon DF, Davis LS (1995) Model-based object pose in 25 lines of code. Int J Comput Vision 15(1–2):123–141
2. Deng J, Guo J, Xue N, Zafeiriou S, Arcface: additive angular margin loss for deep face recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 4690–4699
3. Deng J, Guo J, Zhou Y, et al (2019) Retinaface: single-stage dense face localisation in the wild. arXiv preprint arXiv:1905.00641
4. Fanelli G, Weise T, Gall J, Van Gool L, Real time head pose estimation from consumer depth cameras. In: Joint pattern recognition symposium, pp. 101–110. Springer
5. He K, Gkioxari G, Dollár P, Girshick R, Mask r-cnn. In: Proceedings of the IEEE international conference on computer vision, pp. 2961–2969
6. He K, Zhang X, Ren S, Sun J, Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 770–778
7. Jain V, Learned-Miller E (2010) Fddb: a benchmark for face detection in unconstrained settings. Tech. rep, UMass Amherst technical report
8. Krizhevsky A, Sutskever I, Hinton GE, Imagenet classification with deep convolutional neural networks. Adva Neural Inform Process Syst 1097–1105
9. Li H, Hua G, Lin Z, Brandt J, Yang J (2013) Probabilistic elastic part model for unsupervised face detector adaptation. In: Proceedings of the IEEE international conference on computer vision, pp. 793–800
10. Liu W, Anguelov D, Erhan D, Szegedy C, Reed S, Fu CY, Berg AC, Ssd: Single shot multibox detector. In: European conference on computer vision, pp. 21–37. Springer
11. Koestinger M, Wohlhart P, Roth PM, Bischof H (2011) Annotated Facial Landmarks in the Wild: a large-scale, real-world database for facial landmark localization. In: Proc. First IEEE International Workshop on Benchmarking Facial Image Analysis Technologies
12. Mathias M, Benenson R, Pedersoli M, Van Gool L, Face detection without bells and whistles. In: European conference on computer vision, pp. 720–735. Springer
13. Najibi M, Samangouei P, Chellappa R, Davis LS, Ssh: Single stage headless face detector. In: Proceedings of the IEEE international conference on computer vision, pp. 4875–4884
14. Ng J, Gong S (2002) Composite support vector machines for detection of faces across views and pose estimation. Image Vision Comput 20(5–6):359–368
15. Pan H, Han H, Shan S, Chen X, Mean-variance loss for deep age estimation from a face. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 5285–5294
16. Ranjan R, Patel VM, Chellappa R (2017) Hyperface: a deep multi-task learning framework for face detection, landmark localization, pose estimation, and gender recognition. IEEE Trans Pattern Anal Mach Intell 41(1):121–135
17. Ren S, He K, Girshick R, Sun J. Faster r-cnn: towards real-time object detection with region proposal networks. Adv Neural Inform Process Syst 91–99
18. Rowley HA, Baluja S, Kanade T (1998) Rotation invariant neural network-based face detection. In: Proceedings IEEE computer society conference on computer vision and pattern recognition (Cat. No. 98CB36231), pp. 38–44. IEEE
19. Rowley HA, Baluja S, Kanade T (1998) Neural network-based face detection. IEEE Trans Pattern Anal Mach Intell 20(1):23–38
20. Ruiz N, Chong E, Rehg JM, Fine-grained head pose estimation without keypoints. In: Proceedings of the IEEE conference on computer vision and pattern recognition workshops, pp. 2074–2083
21. Schroff F, Kalenichenko D, Philbin J, Facenet: a unified embedding for face recognition and clustering. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 815–823
22. Sherrah J, Gong S, Ong EJ (2001) Face distributions in similarity space under varying head pose. Image Vision Comput 19(12):807–819
23. Simonyan K, Zisserman A (2014) Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556
24. Sun K, Zhao Y, Jiang B, et al (2019) High-resolution representations for labeling pixels and regions. arXiv preprint arXiv:1904.04514
25. Viola P, Jones MJ (2004) Robust real-time face detection. Int J Comput Vision 57(2):137–154
26. Wang H, Li Z, Ji X, et al (2017) Face r-cnn[J]. arXiv preprint arXiv:1706.01061
27. Yan J, Lei Z, Wen L, Li SZ, The fastest deformable part model for object detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2497–2504
28. Yan J, Zhang X, Lei Z, Li SZ (2014) Face detection by structural models. Image Vision Comput 32(10):790–799
29. Yang B, Yan J, Lei Z, Li SZ, Aggregate channel features for multi-view face detection. In: IEEE international joint conference on biometrics, pp. 1–8. IEEE
30. Yang TY, Chen YT, Lin YY, Chuang YY, Fsa-net: Learning fine-grained structure aggregation for head pose estimation from a single image. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 1087–1096
31. Yashunin D, Baydasov T, Vlasov R (2020) MaskFace: multi-task face and landmark detector. arXiv preprint arXiv:2005.09412
32. Zhang K, Zhang Z, Li Z, et al (2016) Joint face detection and alignment using multitask cascaded convolutional networks. IEEE Signal Process Lett 23(10):1499–1503
33. Zhou X, Wang D, Krähenbühl P (2019) Objects as points. arXiv preprint arXiv:1904.07850
34. Zhu X, Ramanan D (2012) Face detection, pose estimation, and landmark localization in the wild. In: 2012 IEEE conference on computer vision and pattern recognition, pp. 2879–2886. IEEE