Real Time Detection and Tracking Method of Pilot's Head Position Based on MTCNN-DeepSORT

Limin Zhang\textsuperscript{1}, Jie Jiang\textsuperscript{*}, Wei Fang\textsuperscript{1}, Kai Liu\textsuperscript{1}

\textsuperscript{1}Departments of Electronics and Science, University of Naval Aviation, Yantai 264000, China
\textsuperscript{*}hkdx@tsg.edu.com
Email: iamzlm@163.com, 81348541@qq.com, fangw4512@163.com, kailiu5662@163.com

Abstract: In view of the problems of the traditional head detection and tracking technology in pilot cabin simulation training, such as low precision, poor real-time performance and easy to be interfered by the outside world, we propose a head detection and tracking method based on MTCNN-DeepSORT combined with deep learning method. Firstly, the continuous video images in the cockpit are preprocessed by computer vision, and then the processed images are extracted by convolution neural network to detect the face. At the same time, the facial feature points are marked. Through the spatial coordinate calculation of the obtained facial feature points, the pilot's head posture angle is obtained, and then combined with DeepSORT tracking algorithm can track the pilot's head continuously and in real time, and finally complete the detection and tracking of pilot's head position. The experimental results show that the method based on MTCNN-DeepSORT has higher detection accuracy, better real-time performance and stronger robustness than the traditional methods.

1. Introduction
Pilot cockpit simulation training is to restore the real flight environment through flight training simulator, which can carry out takeoff, landing, aerial refueling, formation flight, tactical confrontation and emergency response training in the face of complex situations. With the training mode of low cost, low risk, high effect and high return, it has important strategic significance for future real training and operation. The existing cockpit simulator still remains in the traditional photoelectric method, inertial method, electromagnetic method and so on. These methods complete the detection and tracking by installing sensors in the pilot's head and cockpit. The real-time performance and accuracy of these methods are not high, and they are vulnerable to external interference.

In recent years, with the continuous development of deep learning theory, great breakthroughs have been made in image processing, character recognition and big data analysis. Using deep learning method, through training convolutional neural network, combined with tracking algorithm, the pilot's head position detection and tracking is completed. Under the condition of ensuring the detection accuracy and meeting the real-time requirements, it can still ensure a good detection rate in the face of different illumination, large head posture changes and occlusion conditions, and has strong robustness. It is of great significance for pilots to lock the target first and take advantage of the enemy in close air combat.

2. Construction and Geometrical Dimensions of Specimens
Using the deep learning method, combined with computer vision, the face image is processed by pyramid, and input into the three-layer cascaded convolutional neural network for feature extraction
and non maximum suppression. The face detection frame is output and five key points of face are marked. Then, deep is combined with the algorithm Sort tracking algorithm and EPNP algorithm can calculate the head attitude angle in real time, and finally complete the detection and tracking of pilot's head position. The flow chart is shown in Figure 1:

As two different tasks, target detection and target tracking are related to each other. Firstly, we need to get the bounding box through target detection, and then use it for subsequent tracking. Therefore, the accuracy of bounding box is very important. In this paper, we use MTCNN for face detection, and use three-layer cascade convolution neural network to soft the candidate frames obtained by each layer of convolution neural network NMS processing can effectively remove the overlapping candidate frames, and finally ensure the accuracy of the bounding box.

2.1. Face detection based on MTCNN
For face detection methods, there are early model matching method, HOG-SVM method based on machine learning [1], Viola & Jones method [2], DPM method [3], facet method based on deep learning [4], cascade CNN method [5]. However, these methods are not particularly ideal in industrial applications, and the effect of face alignment based on detection is not ideal due to the general effect of face detection. In view of the above situation, a face detection method based on cascaded convolutional neural network is proposed. Considering that the existing deep learning methods do not take into account the inherent relationship between face detection and alignment, the detection and alignment are solved simultaneously in a network structure. On the other hand, the head posture calculation is added to the network structure, and the pilot's head posture data is obtained by training the convolutional neural network, and the pilot's head position detection and tracking is completed in an end-to-end manner. Firstly, candidate forms are quickly generated from the input face image by cascaded convolutional neural network. The candidate forms are filtered by using non maximum suppression algorithm and boundary box regression to get the accurate face detection frame. At the same time, five facial key feature points are marked. Then, the pilot's head pose angle is finally obtained by using EPNP algorithm through space coordinate transformation. The specific flow Figure2 shows:

2.1.1. Pyramid image processing based on multi scale and multi template
Image pyramid is a representation of different scales of input image. It is mainly used in computer vision and image segmentation. It is a method of image preprocessing. After image pyramid preprocessing, images with different resolutions can be obtained, which can be used to explain the structure of the original input image [6]. Its structure is usually arranged according to the shape of pyramid, and the image resolution decreases gradually from top to bottom, as shown in Figure 3:
Before face detection through convolution network, it is necessary to carry out multi-scale transformation on the image. Usually, the image is scaled to different scale images to form an image pyramid, so that the convolution network can detect faces of different sizes, so as to expand the data set and improve the accuracy of network detection. Compared with the traditional multi-scale transform processing, the multi template processing method is added to enrich the representation of the input image, effectively expand the image data set, so as to ensure the accuracy of face detection in complex scenes. The specific implementation steps are as follows: firstly, the images with different resolutions are obtained by multi-scale processing, and then the images of different scales are sampled by sliding windows of 8x8, 12x12 and 16x16. Finally, the images of different scales and templates are sent to convolution neural network for training. The flow chart is shown in Figure 4:

\[\text{Figure 3. Schematic diagram of image pyramid structure.}\]

\[\text{Figure 4. Multi scale and multi template image pyramid flow diagram.}\]

2.1.2. Network Structure
It is composed of three layers of convolution neural networks, which are Proposal Network, Refine Network and Output Network. Compared with other multi classification object detection and classification tasks, face detection is a binary classification problem. It does not need a larger convolution kernel, but a smaller convolution kernel. Therefore, the convolution kernel is changed from 5 × 5 to 3 × 3, which reduces the size of convolution kernel and increases the depth of the model.

(1) Proposal Network(P-Net)
The first layer of p-net is Full Convolution Network (FCN). The advantage of full convolution network is that it can input images of any size. At the same time, convolution operation is used instead of sliding window operation, which greatly improves the calculation efficiency. The input data are all the images of shape = (12x12x3) to obtain the candidate form and boundary regression vector, and calibrate the candidate form according to the boundary box, Then the overlapped forms are removed by non maximum suppression (NMS). Its structure is shown in Fig. 5:
Figure 5. P-Net Network structure diagram.

The first layer of p-net is full convolution network (FCN). The advantage of full convolution network is that it can input images of any size. At the same time, convolution operation is used instead of sliding window operation, which greatly improves the calculation efficiency. The input data are all the images of shape = (12x12x3) to obtain the candidate form and boundary regression vector, and calibrate the candidate form according to the boundary box, Then the overlapped forms are removed by non maximum suppression (NMS). Its structure is shown in Fig. 6:

Figure 6. R-Net Network structure diagram.

(3) Output Network(O-Net)

The output of R-Net is transferred to O-Net, and the input image shape = (48x48x3). Compared with R-Net, this layer adds a layer of convolution layer, which is mainly applied to the processing of facial details. Its function is the same as R-Net. However, because this layer adds more supervision to the face area, it can mark 5 key points of face (left eye, right eye, nose, left mouth corner) while removing overlapping forms (right corner of the mouth). The network structure is shown in Figure 7:

Figure 7. O-Net Network structure diagram.

From P-Net to R-Net and then to O-Net, with the increasing size and structure of the input image, the extracted features can be more expressive. The first step is to select the candidate window through shallow CNN; the second step is to filter out the window without face with more complex CNN; the third step is to optimize the result with stronger CNN.

2.1.3. Loss Function

For feature description, there are three parts: face non face classifier, boundary box regression and face feature points. The loss function of these three parts is introduced below.
Loss function of face non face classifier

The cross entropy loss function of face classification is as follows:

\[
L^{\text{det}} = -(y \log(p) + (1-y) \log(1-p))
\]  

Where, \(p\) is the probability of face appearance and \(y\) is the real label of the region.

Boundary box regression loss function

Using Euclidean distance as the regression loss function of distance measurement, the equation is as follows:

\[
L^{\text{box}} = \|y' - y\|^2
\]  

Among them, \(y'\) is predicted value border coordinate, \(y\) is true value box coordinate, which is represented by a quaternion array \((X_{\text{left}}, Y_{\text{left}}, \text{Width}, \text{Height})\).

Loss function of facial feature points

Facial feature point location is also a regression problem, and its loss function is to calculate the distance deviation between the predicted key point position and the actual position, and the Euclidean distance is used as the distance measurement. The equation is as follows:

\[
L^{\text{landmark}} = \|y' - y\|^2
\]  

Among them, \(y'\) is the prediction result and \(y\) is the actual position of the key points. As a total of five key points need to be predicted, each point has two coordinate values, so \(y\) is a decimal.

In different CNN layers, the proportion of allocation is different for different loss functions. In P-Net and R-Net, the main task is to distinguish whether it is a face or not, so \(L^{\text{det}}\) accounts for a large proportion; in O-Net, the main goal is to find out the key points of the face, so \(L^{\text{landmark}}\) accounts for a large proportion.

2.1.4. Soft NMS algorithm

Non maximum suppression (NMS) is often used in target detection, edge detection, face detection and other fields. It is used for post-processing of generated candidate frames to remove redundant candidate frames and obtain the best detection frame, so as to speed up the detection efficiency [7].

The idea is to suppress non maxima by searching local maxima. The implementation effect is shown in Figure 8:

\[
\text{IOU} = \frac{\text{Area of overlap}}{\text{Area of union}}
\]

As shown in the Figure 9:

\[
\begin{align*}
\text{IOU} & = 0.423 & \text{Poor} \\
\text{IOU} & = 0.735 & \text{Good} \\
\text{IOU} & = 0.967 & \text{Excellent}
\end{align*}
\]
In NMS algorithm, the score of the window whose IOU is greater than the threshold is set to 0, and the equation is as follows:

\[
s_i = \begin{cases} 
  s_i, & \text{if } \text{iou}(M, h_i) < N_i \\
  0, & \text{if } \text{iou}(M, h_i) \geq N_i
\end{cases} \quad (4)
\]

It can be seen from the formula that all boxes with IOU greater than the threshold are directly deleted, and the scores of adjacent detection frames are forced to return to zero. In this case, if a real object appears in the overlapping area, the object detection will fail and the average precision of the algorithm will be reduced. On the other hand, the threshold setting of NMS is also prone to problems. Once the threshold is set too small, it will be deleted by mistake. If it is set too high, it will lead to false detection [9]. In view of the above situation, the traditional NMS algorithm is improved, that is, in the process of algorithm implementation, it is no longer simply to delete the detection box whose IOU is greater than the threshold, but to reduce the confidence score by establishing the confidence function. The confidence reset function is improved in two forms [10], one is linear weighting:

\[
s_i = \begin{cases} 
  s_i, & \text{if } \text{iou}(M, h_i) < N_i \\
  s_i(1-\text{iou}(M, bi), \text{iou}(M, h_i)) \geq N_i
\end{cases} \quad (5)
\]

One is Gaussian weighting:

\[
s_i = s_i e^{-\frac{\text{iou}(M, bi)}{\sigma}}, \forall bi \notin D \quad (6)
\]

By updating the weight variable continuously, the updated value is determined as 1-ov by linear weighting. On the other hand, sigma parameter is introduced in Gaussian weighting, so that the value of IOU in the original NMS algorithm is no longer directly 0 or 1. Soft NMS is more accurate for the determination of face detection frame, which effectively improves the network training effect.

2.2. Face tracking based on Deep SORT

Deep SORT is a multi-target tracking algorithm. Its basic idea is tracking by detection. It uses motion model and appearance information to correlate data. The running speed is mainly determined by detection algorithm. The algorithm detects the target in each frame, and then matches the previous motion track with the current detection object through the weighted Hungarian matching algorithm to form the motion track of the object[11]. The weights are obtained by the weighted sum of Mahalanobis distances of points and trajectories and the similarity of image blocks[12]. Kalman filter is used to predict the covariance matrix of motion distribution when calculating Mahalanobis distance. The innovations are as follows:

1. Reid model is introduced to calculate cosine distance by appearance information.
2. Deep refers to the introduction of ReID model, which is a depth model to extract appearance information, and finally outputs 128dD vector.
3. In Hungary matching algorithm, cascade matching is used. It refers to using Hungarian algorithm to match different disappearing time trajectories.

2.2.1. Tracking processing and state estimation

1. Trajectory processing

   Track generation: if the matching algorithm does not match the target, it may be a new track generated. Create a new tracker for these targets, but mark it as "persistent" (because these targets may be noise from the detector output). When the consecutive frames are matched, the new track is confirmed to be valid and marked as "confirm"; otherwise, it is considered as noise track and marked as "deleted".

   Track disappearance: each track has a variable a to record the time from the last successful match to the current time. If \( a > \text{Max\_Age} \) threshold, the trajectory is considered to be terminated, and the trajectory is cancelled in the subsequent tracking.

   When the target and the track are matched, the target is added to the matching track.
State estimation

An 8-dimensional space is used to describe the state \((u, v, r, h, \dot{x}, \dot{y}, \dot{r}, \dot{h})\) of the trajectory at a certain time, which represents the position, aspect ratio, height of the bounding box center, and the corresponding velocity information in the image coordinates. Then a Kalman filter is used to predict the update trajectory. The Kalman filter uses a uniform model and a linear observation model. The observation variable is \((u, v, r, h)\).

2.2.2. Weight measurement in Hungarian algorithm

The innovation of Deep SORT is to weigh the weight of Hungarian algorithm by using motion matching degree and appearance matching degree. The weight of each line is used, and \(b_{ij} = \prod_{m=1}^{n} b_{ij}^{(m)}\) is used to determine the initial matching line.

(1) Motion matching

The Mahalanobis distance between the positions predicted by Kalman filter is used to describe the degree of motion matching \(d^{(1)}\):

\[
    d^{(1)}(i, j) = (d_j - y_i) T S_i^{-1} (d_j - y_i) \quad (7)
\]

Represents the motion matching degree between the first detection and the first track, where is the covariance matrix of the trajectory predicted by the Kalman filter in the observation space at the current time, is the prediction quantity bbox of the trajectory at the current time, and is the bbox of the first detection.

Mahalanobis distance considers the uncertainty of state estimation by measuring the standard deviation of deviation from the average orbit position. In addition, this measure can be used to exclude unlikely associations by thresholding the Mahalanobis distance at the 95% confidence interval calculated from the inverse distribution. This indicator is measured by \(b_{ij}^{(1)}\):

\[
    b_{ij}^{(1)}(i, j) = I[d^{(1)}(i, j) \leq t^{(1)}] \quad (8)
\]

(2) Appearance matching degree

There is a serious problem of idswitch in sort which only uses motion information. The Reid model can alleviate this problem and make the target reconnect to the previous track.

For each detection, including the detections in the trajectory, the feature vector \(r\) of unit norm is extracted by depth network. Then, the minimum cosine distance between detection and the eigenvectors of detections contained in the trajectory is used as the apparent matching degree between detection and tracker.

Of course, if the trajectory is too long, the appearance will change. If the minimum distance is used as the measurement, the minimum cosine distance is only calculated for the detections within the latest \(L_a = 100\) of the track:

\[
    d_{ij}^{(2)}(i, j) = \min \{1 - r_j^T r_i | r_i^{(1)} \in R_j\} \quad (9)
\]

Similarly, the measure can also determine a threshold function

\[
    b_{ij}^{(2)}(i, j) = I[d^{(2)}(i, j) \leq t^{(2)}] \quad (10)
\]

This threshold needs to be adjusted according to the actual situation. For face detection, it is usually set to 0.6.

(3) Comprehensive matching degree

The weighted sum of motion matching degree and appearance matching degree is as follows:

\[
    c_{ij} = \lambda d^{(1)}(i, j) + (1 - \lambda) d^{(2)}(i, j) \quad (11)
\]

With the threshold function, only when \(b_{ij}\) is 1, can it be considered as a preliminary match:

\[
    b_{ij} = \prod_{m=1}^{n} b_{ij}^{(m)} \quad (12)
\]
2.2.3. Cascade matching

In order to make the current detection target be connected with the track which is closer to the current
time, the track with smaller vanishing time should be preferentially matched when matching.
Moreover, when two trajectories compete for the same detection, the one with long vanishing time
often gets smaller Mahalanobis distance, because Kalman filter has been predicting and has not been
updated, resulting in the dispersion of covariance matrix. At this time, cascade matching is introduced
to make more frequently seen objects assign higher priority.

2.3. Head pose calculation based on efficient perspective-n-point (EPNP)

EPNP algorithm is a non iterative and closed PNP algorithm. By representing the 3D coordinates in
the world coordinate system as the weighted sum of a group of virtual control points [13], it can be
used to solve the pose. For face pose calculation, EPNP algorithm needs to know at least the three-
dimensional coordinates of four non coplanar face feature points. By matching the corresponding
points in the two-dimensional camera coordinate system, the rotation vector is calculated, and the
rotation vector is converted into quaternion, and finally the Euler angle containing the posture
information is calculated. For the 3D face model, the universal head model of the University of
Coimbra is used to obtain the 3D coordinate information of five key points of the face, including
eyeball, nose tip and mouth corner. The internal parameters of the camera are determined by Zhang
Zhengyou camera calibration method [14]. According to the five facial key points output by
convolution neural network, the pilot's head posture data can be obtained by EPNP algorithm. The
specific calculation process is as follows:

Firstly, the mathematical model of camera coordinate system is established:

\[
\begin{align*}
\begin{bmatrix} x_i \\ y_i \\ z_i \\ 1 \end{bmatrix} &= \mathbf{K} \begin{bmatrix} R & t \end{bmatrix} \begin{bmatrix} X_i \\ Y_i \\ Z_i \\ 1 \end{bmatrix} \\
&= \begin{bmatrix} 0 & f_{x_0} & u_{x_0} \\ 0 & f_{y_0} & v_{y_0} \\ 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} R & t \end{bmatrix} \begin{bmatrix} X_i \\ Y_i \\ Z_i \\ 1 \end{bmatrix}
\end{align*}
\]

(13)

Where \( \mathbf{K} = \begin{bmatrix} f_{x_0} & s & u_{x_0} \\ 0 & f_{y_0} & v_{y_0} \\ 0 & 0 & 1 \end{bmatrix} \) is the camera's internal parameters, \( \mathbf{R} = \begin{bmatrix} R^T_1 \\ R^T_2 \\ R^T_3 \end{bmatrix} \) is the rotation matrix,
\( t = (t_x, t_y, t_z)^T \) is the translation vector.

The relationship between the image coordinate system \((x, y)\) and the camera coordinate system
\((X, Y, Z)\) is as follows:

\[
\begin{align*}
\begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} &= \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \\
&= \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} X_i \\ Y_i \\ Z_i \\ 1 \end{bmatrix}
\end{align*}
\]

(14)

Since the head posture is three degrees of freedom, it is represented by a matrix of 3x3:

\[
\begin{align*}
\mathbf{R} &= \begin{bmatrix} r_1 & r_2 & r_3 \\ r_4 & r_5 & r_6 \\ r_7 & r_8 & r_9 \end{bmatrix} \\
&= \begin{bmatrix} \cos \alpha \cos \beta \cos \gamma & -\cos \alpha \sin \gamma & \sin \gamma & -\sin \alpha \cos \beta \cos \gamma & \sin \alpha \sin \gamma & \cos \gamma \\ \sin \alpha \cos \beta \cos \gamma & \cos \alpha \cos \gamma & -\sin \gamma & \sin \alpha \sin \beta \cos \gamma & \cos \gamma & \sin \alpha \sin \gamma \\ -\sin \beta & \cos \beta & 0 & -\sin \beta \cos \gamma & \cos \beta \cos \gamma & \sin \beta \sin \gamma \end{bmatrix}
\end{align*}
\]

(15)

According to the relationship between rotation matrix and rotation angle, the head pose angles roll,
pitch, yaw can be obtained:
3. Test Results and Discussions

3.1. Data preparation

The wider face data set is selected as the face detection data set. The data set is composed of 393703 marked face bounding boxes in 3233 images, of which 50% is used to test three subsets according to image difficulty, 40% is used for training, and the rest is used for verification. The annotated facial landmarks in the wild (ALFW) dataset is selected as the facial feature point detection dataset. The aflw dataset contains 24386 facial feature point markers. Through the multi-scale and multi template processing method, the marked face image is randomly cut. According to the cut border and the real face frame, the IOU is calculated. The IOU > 0.65 is divided into positive sample data, the IOU < 0.3 is divided into negative sample data, and the IOU between 0.4 and 0.65 is divided into part data. The five feature points are divided into landmark face data. The purpose of these four types of data:

1. Negative and positive face classification
2. Positive and part faces are used for border regression
3. Landmark face for face feature point location (face alignment / key point detection)

3.2. Analysis of experimental results

After multi-scale and multi template processing, the number of training samples is obtained as follows:

|       | Positive | Negative | Part    | Landmark |
|-------|----------|----------|---------|----------|
| P-NET | 89251    | 36758547 | 623871  | 635847   |
| R-NET | 225176   | 674412   | 224713  | 312478   |
| O-NET | 283679   | 854178   | 289745  | 547894   |

The data set is sampled according to the ratio of negative / positive / part faces / landmark = 3:1:1:2, and then sent to each network for training. Finally, the recall rate is calculated according to the detection results. The equation is as follows:

\[
R = \frac{TP}{TP + FN} \quad (17)
\]

In the equation, TP is the number of correctly recognized faces, FN is the number of faces not correctly detected, that is, the system fails to report. Comparing the recall results with the multi-scale processing, we can see that the multi-scale and multi-template image preprocessing method has a certain improvement on the detection results.

Table 2. Comparison of recall rate results of different network layers.

|                | P-NET     | R-NET     | O-NET     |
|----------------|-----------|-----------|-----------|
| Multi scale    | 94.6%     | 95.4%     | 95.7%     |
| Multi scale and multi template | 94.8% | 97.7% | 95.9% |

ROC curves were drawn and compared with other algorithms in the wider face dataset, the results are shown in Figure 10:
The results show that the recall rate is higher than other algorithms, and the recall rate can reach 90.9% when the false alarm sample is 1000.

The performance of Deep_SORT is compared with other tracking algorithms, and the results are shown in Figure 11:

| Tracker | Detection | Tracking |
|---------|-----------|----------|
|         | Recall    | ID Sw    |
| ACF     | 36.3      | 220      |
| MDP     | Cascade_CNN | 243      |
| MTCNN   | 50.2      | 176      |

Table 3. Comparison of tracking performance of different detector elements.
| Method       | ACF  | Cascade_CNN | MTCNN |
|-------------|------|--------------|-------|
| Proposed    | 33.4 | 65.2         | 221   |
|              |      | 24.1         | 34.3  |

It can be seen from the table that the face detection and tracking method based on MTCNN-DeepSORT is better than the traditional ACF method and Cascade_CNN has the best detection and tracking performance on MDF and proposed.

4. Conclusion
In this paper, we propose a real-time pilot head detection and tracking method based on MTCNN-DeepSORT. Using MTCNN algorithm for face detection, the algorithm uses face detection and alignment in association, and outputs five facial key points while obtaining the face detection frame, which can be used for subsequent head pose calculation. Compared with other detection algorithms, it has better performance in detection accuracy and real-time performance, and is more suitable for engineering practice. At the same time, the MTCNN algorithm is optimized in this paper, and the accuracy of the output face detection box can be improved by 2.3% through multi-scale and multi-template processing of the input image. The Deep_SORT tracking algorithm is adopted, whose tracking quality is highly dependent on the accuracy of detection algorithm. Through comparative analysis combined with different detection algorithms, it can be found that MTCNN has the best performance. Meanwhile, we combine appearance information through pre-trained association method, so as to track and hide the target for a longer time. The proposed framework achieves the best performance in both speed and accuracy, satisfies the needs of cockpit simulation training, and achieves the real-time detection and tracking effect of the head-position. Since our experiments emphasize the importance of detection quality in tracking, future work will study a tightly coupled detection and tracking framework.

Acknowledgments
This work was financially supported by the Taishan Scholars project of Shandong Province (201511020).

References
[1] Feng Z. Face Detection Based on SVM and HOG[J]. Information Technology and Informatization, 2013.
[2] Li S Z , Zhu L, Zhang Z Q , et al. Statistical learning of multi-view face detection[J]. 2002.
[3] Mathias M, Benenson R, Pedersoli M, et al. Face detection without bells and whistles[J]. 2014.
[4] F Schroff, D Kalenichenko, J Philbin. Facenet: a unified embedding for face recognition and clustering[C]. IEEE, 2015.
[5] Li H, Lin Z, Shen X, et al. A convolutional neural network cascade for face detection[C] Computer Vision & Pattern Recognition. IEEE, 2015.
[6] Mathias M, Benenson R, Pedersoli M, et al. Face detection without bells and whistles[J]. 2014.
[7] F Schroff, D Kalenichenko, J Philbin. Facenet: a unified embedding for face recognition and clustering[C]. IEEE, 2015.
[8] Hosang J, Benenson R, Schiele B. Learning Non-maximum Suppression[J]. 2017.
[9] Hosang J, Benenson R, Schiele B. A convnet for non-maximum suppression[J]. 2015.
[10] Bodla N, Singh B, Chellappa R, et al. Soft-NMS -- Improving Object Detection With One Line of Code[J]. 2017.
[11] Wojke N, Bewley A, Paulus D. Simple Online and Realtime Tracking with a Deep Association Metric[J]. 2017:3645-3649.
[12] Bewley A, Ge Z, Ott L, et al. Simple Online and Realtime Tracking[J]. arXiv, 2016.
[13] Xiong G, Ding T H, Wang P. Perspective-n-point pose measurement with two line array cameras[J]. Optical Engineering, 2013, 52(1):177-182.
[14] Shujie L, Xiaoping L. An accurate and fast algorithm for camera pose estimation [J]. Journal of Image and Graphics, 2014.