*Journal of Network Intelligence*
**Taiwan Ubiquitous Information**
 ISSN 2414-8105 (Online)
Volume **6**, Number **2**, May **2021**

# Face tracking based on convolutional neural network and kernel correlation filter

Yi-Jia Zhang

School of Information Science and Technology
Zhejiang Sci-Tech University
Hangzhou 310018, P. R. Chinaa
waiting@zstu.edu.cn

Kuncai Zhang and Zhe-Ming Lu*

School of Aeronautics and Astronautics
Zhejiang University
Hangzhou 310027, P.R. China
*Correponding Author: zheminglu@zju.edu.cn

Received January 2021;Revised March 2021

Abstract. *Face tracking is an important issue in the field of visual tracking and has great practical significance in real life. In this paper, a new face tracking framework based on detection, tracking and prediction is proposed. The framework detects the face based on an efficient convolutional neural network and starts tracking using the KCF (Kernel Correlation Filter) algorithm. Once the face tracking fails in some complex cases, convolutional neural network detects the face. If the target face is occluded by other objects that the detection fails, a Kalman filter is used to predict the position of the face. Thus the tracking framework guarantees the continuity of face tracking in a long time. The experimental results demonstrate high accuracy and efficiency of the proposed framework for robust face tracking.*
**Keywords:** Face tracking, Convolutional neural network, Kalman filter.

1. **Introduction.** In the field of computer vision, face tracking is a fundamental issue in visual tracking especially for the case of long-term continuous tracking. It can provide information of the facial position in video surveillance, face capture and many other real-life scenes.Face detection and tracking involve knowledge in the fields of pattern recognition, image processing, physiology and psychology. Up to now, face tracking is still a challenge in some extreme cases such as occlusion, illumination variation, size changes, rotation etc. And for the long-term face tracking, the efficiency and continuity are another two difficulties in the practical application. Three main challenges it faces can be summarized as follows:

(1) The diversity of the face in the image: Due to the different angles of the camera, the face in the image can be displayed from different angles (for example, only the side of the face is photographed); it may also be blocked by other objects or the human body itself , or the background of the face is more complicated, etc., which causes the features to be extracted during detection and tracking to be disturbed, or even invisible.

(2) Irregularity of the movement of the face: The movement of the person and the face has great autonomy, there is no law to follow, there are often sudden turns and drastic changes in the angle within a short period of time, which is a typical nonlinearity

movement, which cannot be expressed by a simple motion equation. This brings great difficulties to the design of tracking algorithms.

(3) Real-time requirements: In practical applications, such as security monitoring, video conferencing, etc., the system is required to have good real-time performance. For example, when someone walks into the monitoring range, the face needs to be detected and tracked in time, and the position, angle, size and other information of the face are fed back in real time until the target face goes out of the monitored range. In order to meet the real-time nature of tracking, the number of extracted target features cannot be too many or too complex; at the same time, in order to ensure the accuracy and robustness of tracking, multiple features of the target need to be considered comprehensively, and the amount of data in the image itself is very large. It is time consuming to calculate each feature. It is particularly important and difficult to find an appropriate compromise between the robustness of the algorithm and the real-time tracking.

Traditional algorithms of visual object tracking can be generally classified as generative-based and discriminative-based tracking methods. Tracking methods based on generative models try to orient the target object by representing its visual observations. Typical generative-based tracking methods include Meanshift [1], particle filter [2] and Kalman filter [3]. The Meanshift algorithm [1] tries to produce multiple centers of maximum density by moving multiple random centers towards the direction of maximum density. It uses the color distribution of the target object as a visual representation and computes the possible distribution on the next video frame. The particle filter [2] defines a measurement of similarity between a particle and the target object. It finds the possible orientation of the target object in assist of the similarity of these particles. The particles are produced according to the Gaussian distribution. The Kalman filter [3] simulates the motion model of the target object and utilizes the state information of current frame to predict the possible position of the target object in the next frame. The Kalman filter algorithm is often used to predict the position of the target in the cases of occlusions. For the discriminative-based tracking methods, they select robust features for binary classification and build a classifier that distinguishes between the background and the tracking target. The typical method based on discriminative models is ensemble tracking [4], which tracks the target using the AdaBoost classifier in conjunction with the color features and the local histogram. And another well-performed visual tracking algorithm is TLD [5] proposed in 2012. It divides the tracking task into three parts, i.e., tracking, learning, and detection.

In terms of face tracking particularly in practical applications, the following requirements need to be met. First, the tracking process should be real-time. That means the tracking rate should be at least 25 30FPS, which is the common rate for playing a video [6]. Second, the tracking scheme is usually desired to be as accurate as possible. To some extent, the accuracy can reflect the overall effect of the tracking process. Last but not least, the continuity of the tracking process is desired to be realized in a large number of long-term tracking cases. That means once the tracking scheme fails in some complex cases, the tracking framework should be able to check whether it fails and correct in following frames if it fails.

Many face tracking schemes usually assume that the face can be quickly and accurately located in the initial frame for tracking. However, it takes a certain amount of time to locate the face in the initial frame, which may affect the real-time tracking. Secondly, many face tracking schemes assume that no other target faces will appear in the scene. If other faces appear during the tracking process, they cannot be detected in time. Finally, many face tracking algorithms cannot solve long-term nonlinear occlusion problem. This paper focuses on the face tracking problem in practical applications. As mentioned in the above paragraphs, this paper tries to put forward a face tracking framework that can

achieve a balance between the accuracy and the efficiency. In addition, the framework proposed by this paper can guarantee the continuity of face tracking. In addition, this framework accepts all kinds of new face detection schemes. The main contributions of this work are: (1) We propose an improved framework for face tracking;(2) This framework can guarantee the continuity of face tracking. In the case that the tracking fails in some complex conditions, detection can be executed to detect the face without interruption. When the detection fails in some extreme cases, the prediction submodule, i.e., Kalman filter, is applied to predict the position of the face; (3)The accuracy is improved with a good efficiency.

2. **General Framework.** Aimed at getting a balance between the accuracy and efficiency, this paper proposes a face tracking framework based on three submodules: detection, tracking and prediction. The detection submodule adopts a deep network fine-tuned from a lightweight and accurate CNN, i.e., YOLO [7]. The tracking submodule uses the efficient tracker KCF [8] to track the face after the detection is completed. When the tracking process fails in some hard cases, detection is executed again. Even if the detection failed in some occluded cases, the prediction submodule, i.e., Kalman filter, is applied to predict the face position. Thus the continuity of face tracking can be guaranteed.

The experimental results demonstrate that the proposed face tracking framework can meet real time requirement and has a relatively high accuracy for the long-term continuous face tracking. The overall pipeline of the proposed framework is shown in Fig. 1.

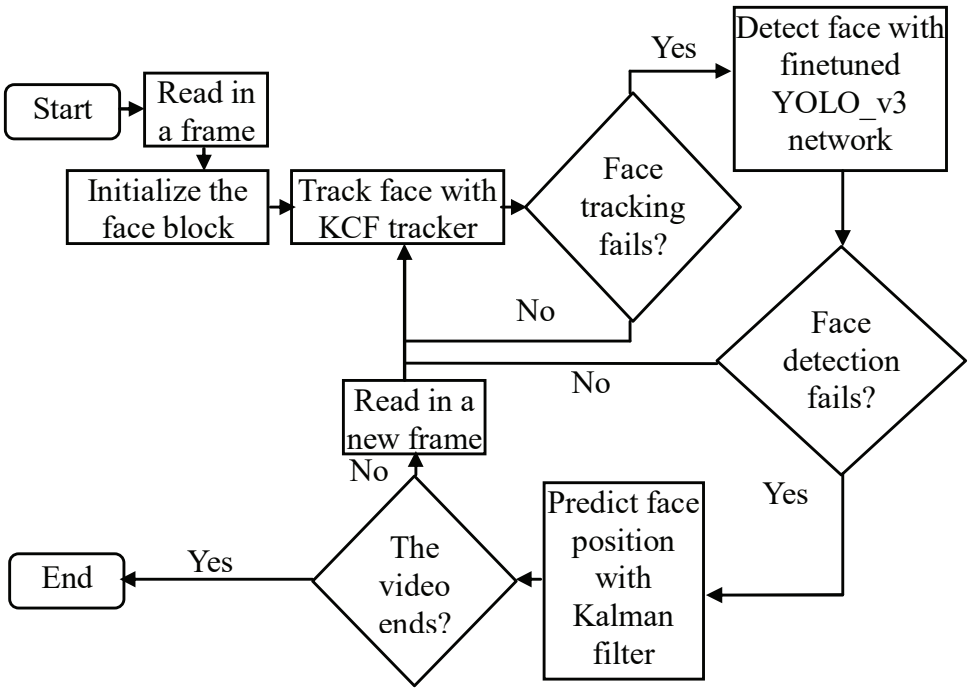


FIGURE 1. General framework of the proposed approach for continuous face tracking.

3. **Robust and Efficient Face Tracking.** In consideration of the balance between accuracy and efficiency, a face tracking framework based on detection, tracking, and prediction is proposed. The three submodules of the framework are depicted as follows.

3.1. **Detection.** The first submodule is the finetuned YOLO_v3 network. YOLO is a deep learning based framework for real time object detection proposed by Redmon et al. in 2016 [7]. YOLO develops a new way of thinking for target detection aiming at the problem that real-time detection cannot be achieved in target detection. The main training mechanism of this method is different from the traditional method, which regards detection as a regression problem. It developed from the original version 1 to present version 3 through continuous improvement. While satisfying detection accuracy, the model can realize real-time object detection, reaching the detection rate of more than 45 FPS.

The main idea of YOLO is to input the image into an evenly divided grid. If the center of the target falls into one of the grid cells, that grid cell is responsible for detecting that target. Bounding boxes and confidence scores for those boxes are predicted by each grid cell. These confidence scores reflect the possibility that the box contains an object and also the accuracy of the box that it predicts.

As shown in Fig. 2, this paper uses the framework of YOLO_v3 as a reference and makes some improvements based on it. The finetuned YOLO network proposed in this paper adds two convolutional layers behind the original eighth layer. The filter sizes of the added convolutional layers are $1 \times 1$ and $3 \times 3$ respectively. Fig. 2 shows the schematic diagram of the finetuned YOLO network framework. It can realize cross-channel interaction and information integration in assist of the $1 \times 1$ convolutional layer and achieve a slightly higher accuracy. Here, the $1 \times 1$ convolution adds non-linear activation to the learning representation of the previous layer to improve the expressive ability of the network. Using a $1 \times 1$ convolution kernel, the operation of achieving dimensionality reduction and dimension upgrading is actually a linear combination of information between channels. Add a $1 \times 1$, 28 channels convolution kernel in front of the $3 \times 3$, 64 channels convolution kernel, and it becomes With the $3 \times 3$, 28 channels convolution kernel, the original 64 channels can be understood as a cross-channel linear combination into 28 channels, which is the information exchange between channels.

3.2. **Tracking.** The second submodule is the KCF tracker. For the tracking part of the proposed face tracking framework, an effective algorithm recognized as KCF is applied. KCF is a discriminant tracking method, which usually trains a target detector in the tracking process, using the target detector to detect whether the predicted position is a target in the next frame. KCF uses the new detection result to update the training set and then update the target detector. When training the target detector, the target region is generally selected as a positive sample and the surrounding region is a negative sample.

The core idea of KCF is to use the cyclic matrix surrounding the target area to collect the positive and negative samples. It uses ridge regression [9] to train target detector and successfully applies cyclic matrix diagonolization in the Fourier space to convert the matrix calculations to the Hadamard product [10] of vectors, namely multiplication dot. The ridge regression of the linear space is mapped to the nonlinear space by a kernel function. In the nonlinear space, it simplifies the calculation by diagonalizing the cyclic matrix in the Fourier space and solving a dual problem with some common constraints. It greatly reduces the computational complexity and improves the operation speed, making the algorithm meet the real-time requirements.

3.3. **Prediction.** The third submodule is the Kalman filter based predictor. For the purpose of dealing with the problem of detection failure on account of illumination variation, occlusion or other negative factors, the Kalman filtering feedback mechanism is used to predict the position of the target in the next frame. Kalman filter is a recursive filter for time-varying linear systems. The system can be described by a differential equation

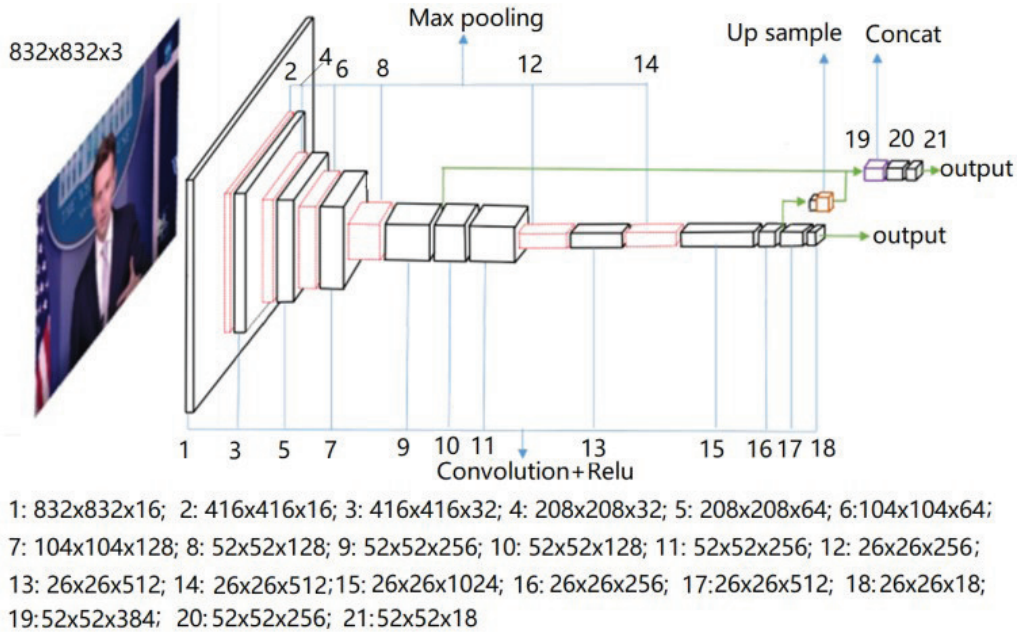


FIGURE 2. Schematic diagram of the finetuned YOLO network for face detection.

model with orthogonal state variables. This filter combines past estimation errors into new measurement errors to estimate future errors. Kalman filtering can be used in any dynamic system with uncertain information to make educated predictions about the next step of the system. It can always point out what is really going on, even with all kinds of disturbances. It is ideal to use the Kalman filter in continuously changing systems, which has the advantage of small memory footprint since no other historical data needs to be retained other than the previous state variable. Kalman filtering is fast and well suited for real-time problems and embedded systems.

The process equation of the discrete system is

$$X(k) = AX(k-1) + BU(k-1) + W(k) \tag{1}$$

The measurement equation of the discrete system is

$$Z(k) = HX(k) + V(k) \tag{2}$$

In the above equations, $X(k)$ and $X(k-1)$ represent the system states at times $k$ and $k-1$ respectively. $A$ and $B$ are system parameters. $U(k)$ is the control amount of the system at time $k$. $V(k)$ and $W(k)$ are the measured and process noises. $H$ is the parameter of the measurement system.

According to the system model, five classical equations of the discrete Kalman filter can be given as follows:

(1) The state equation

$$X(k|k-1) = AX(k-1|k-1) + BU(k-1) \tag{3}$$

(2) Error prediction of the covariance matrix equation

$$P(k|k-1) = AP(k-1|k-1)A^{\mathrm{T}} + Q \tag{4}$$

(3) The filter gain equation

$$G(k) = P(k|k-1)H^{\mathrm{T}}(HP(k|k-1)H^{\mathrm{T}} + R)^{-1} \tag{5}$$

(4) The estimation equation

$$X(k|k) = X(k|k-1) + G(k)(Z(k) - HX(k|k-1)) \tag{6}$$

(5) The updating equation of the covariance matrix's error prediction

$$P(k|k) = (I - G(k)H) + P(k|k-1) \tag{7}$$

The parameters mentioned in above equations are defined as follows. $U$ is an optional control input and $Q$ is the process noise covariance. $R$ is the covariance of $V(k)$. $G(k)$ is the Kalman gain during the measurement update. $P(k|k-1)$ and $P(k-1|k-1)$ are the covariances of $X(k|k-1)$ and $X(k-1|k-1)$ respectively. $H$ is a matrix to relate the state with the measurement.

## 4. **Experimental Results.**

4.1. **Datasets.** To evaluate the robustness and efficiency of the proposed pipeline for continuous face tracking, we train the YOLO detector with the WIDER FACE dataset [11] and compare the performance of the proposed pipeline with other traditional approaches on the 300-VW dataset [12].

In our experiments, we use the WIDER FACE dataset to train the deep learning based YOLO detector. The WIDER FACE dataset is a famous open dataset for academic research, of which images are selected from the publicly available. The dataset contains 32203 images and 393703 annotated faces with a high degree of variability in scale, pose and occlusion. We train the YOLO detector on a single machine with a GeForce 1080TI GPU. The training process costs about three hours.

For the evaluation of the proposed pipeline for long-term continuous face tracking, we use the 300-VW dataset to make comparing experiments with other traditional approaches. The 300-VW dataset contains 50 training videos and 64 test videos which annotates faces from different people. This dataset aims at testing the ability of current systems for fitting unseen subjects, which is independent of variations in pose, expression, illumination, background, occlusion, and image quality. The test set is divided into three categories, which respectively include the face videos under three different constraints. As shown in Fig. 3, the 64 test videos are classified into three categories according to the scenarios of different complexities. The first category contains 31 videos captured in advantageous environments while has occlusions such as glasses and the person displaying different expressions in various head poses. The second category includes 19 videos captured under adverse illumination conditions such as dark rooms, overexposed shots, etc. The third category consists of 14 video sequences captured under completely extreme conditions (including severe occlusions and extreme illuminations).


FIGURE 3. Three categories of the 300-VW datasets.

TABLE 1. Success rate of different approaches.

| Category Number | Approaches | | | | |
|---|---|---|---|---|---|
| | Meanshift | Particle Filter | TLD | KCF | Our |
| 1 | 0.860 | 0.929 | 0.945 | 0.962 | **0.991** |
| 2 | 0.839 | 0.925 | 0.940 | 0.959 | **0.989** |
| 3 | 0.537 | 0.786 | 0.823 | 0.847 | **0.878** |

TABLE 2. Frame rate of different approaches (measured by FPS).

| Category Number | Approaches | | | | |
|---|---|---|---|---|---|
| | Meanshift | Particle Filter | TLD | KCF | Our |
| 1 | 47.9 | 35.6 | 32.6 | 46.5 | 44.8 |
| 2 | 43.6 | 33.7 | 28.6 | 43.6 | 42.6 |
| 3 | 42.8 | 31.4 | 23.9 | 40.8 | 38.9 |

4.2. **Results.** Our face tracking framework is compared with other approaches including Meanshift, Particle filter, TLD and KCF on the 300-VW dataset in terms of both accuracy and efficiency. We use the Intersection over Union (IoU) index to evaluate the accuracy of face tracking. IoU is the overlap ratio of the model's generated target area and the annotated ground truth area, as defined in Eq. 8, where $Area^{GT}$ is the area of the ground truth face and $Area^{GE}$ is the generated target area.

$$IoU = \frac{Area^{GT} \bigcap Area^{GE}}{Area^{GT} \bigcup Area^{GE}} \tag{8}$$

In this paper, the hardware environment of the positioning system includes an equipment box (blue), an industrial HD CCD camera and a Mega pixel camera (on the top of the blue box body), an Advantech industrial PC (with a panel in the middle), a machine vision ring light source, transmission and positioning mechanisms, and a PLC control system and other ancillary equipments, On the left side of Fig. 1, there are the working interface and the identification system box, and on the right side, it shows the wheel hub is lining up into the drilling machine.

Similar to other works [9,10], we set the IoU threshold to be 0.75, which means that if $IoU \geq 0.75$, face tracking succeeds. If $IoU < 0.75$, the face tracking fails. We do experiments on all of the video frames and compute the average success rate of different approaches for videos in each category. Table 1 shows the experimental results.

As shown in Table 1, our approach made an evident improvement on the accuracy of face tracking. The Meanshift algorithm displays a worse performance than the other approaches with the success rate lower than 0.9 on all of the three categories. Our framework that combines KCF with detection and prediction modules gets a higher success rate than the single KCF algorithm. This demonstrates the superiority of our approach on the accuracy for face tracking.

For the evaluation of the efficiency of our approach, we compute the average frame rate on the test videos and make a comparison with other methods. The experimental results are shown in Table 2. As illustrated in the table, the frame rates of Meanshift and KCF trackers are similar and higher than the TLD tracker. Our approach achieves a comparable average frame rate to the KCF tracker on the three categories of test videos. The average frame rate of our approach is about 40 FPS and can meet the requirements of real-time face tracking. In the future, we consider adopt some new network framework [13,14] to further improve the tracking efficiency and accuracy.

5. **Conclusions.** This paper proposed a robust and efficient framework for face tracking based on detection, tracking and prediction submodules. The detection submodule adopts a deep network fine-tuned from a lightweight and accurate CNN, i.e., YOLO_v3. The tracking submodule uses the KCF tracker to track the face after the detection is completed. In case that the tracking fails in some complex conditions, detection is executed to detect the face without interruption. When the detection fails in some extreme cases, the prediction submodule, i.e., Kalman filter, is applied to predict the position of faces, thus the continuity of face tracking is guaranteed. Comparison results on the open dataset demonstrate the robustness and efficiency of our face tracking approach.

## REFERENCES

[1] Y. Dhassi, and A. Aarab, Visual tracking based on adaptive mean shift multiple appearance models, *Pattern Recognition and Image Analysis*, vol. 28, no.3, pp. 439–449, 2018.

[2] X. Kong, Q. Chen, G. Gu, K. Ren, W. Qian, and Z. Liu, Particle filter-based vehicle tracking via HOG features after image stabilisation in intelligent drive system, *IET Intelligent Transport Systems*, vol. 13, no. 6, pp. 942–949, 2018.

[3] J. Yang, W. Zhao, Y. Han, C. Ji, B. Jiang, Z. Zheng, and H. Song, Aircraft tracking based on fully conventional network and Kalman filter, *IET Image Processing*, vol. 13, no.8, pp. 1259–1265, 2019.

[4] J. Guo, and T. Xu, Deep ensemble tracking, *IEEE Signal Processing Letters*, vol. 24, no. 10, pp. 1562—1566, 2017.

[5] Z. Kalal, K. Mikolajczyk, and J. Matas, Tracking-learning-detection, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no.7, pp. 1409–1422, 2012.

[6] W. Song, D. Tjondronegoro, and M. Docherty, Exploration and optimization of user experience in viewing videos on a mobile phone, *International Journal of Software Engineering and Knowledge Engineering*, vol. 20, no. 8, pp. 1–26, 2010.

[7] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, You Only Look Once: unified, real-time object detection, *in IEEE Conference on Computer Vision and Pattern Recognition*, pp.779–788, June 2016.

[8] J. F. Henriques, R. Caseiro, P. Martins, and J. Batista, High-speed tracking with kernelized correlation filters, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 3, pp. 583–596, 2014.

[9] D. Hsu, S. M. Kakade, and T. Zhang, Random design analysis of ridge regression, *Foundations of Computational Mathematics*, vol. 14, no. 3, pp. 569–600, 2014.

[10] C. Bocci, G. Calussi, G. Fatabbi, and A. Lorenzini, The hilbert function of some hadamard products, *Collectanea Mathematica*, vol. 12, pp. 1–16, 2017.

[11] S. Yang, P. Luo, C. C. Loy, and X. Tang, WIDER FACE: A Face detection benchmark, *in IEEE Conference on Computer Vision and Pattern Recognition*, November 2015.

[12] J. Shen, S. Zafeiriou, G. G. Chrysos, J. Kossaifi, and M. Pantic, The first facial landmark tracking in-the-wild challenge: benchmark and results, *in IEEE International Conference on Computer Vision Workshops*, December 2015.

[13] K. Wang, P. Xu, C.-M. Chen, S. Kumari, M. Shojafar, and M. Alazab, Neural architecture search for robust networks in 6G-enabled massive IoT domain, *IEEE Internet of Things Journal*, Early Access, November 2020.

[14] E. K. Wang, S. P. Xu, C.-M. Chen, and N. Kumar, Neural-architecture-search-based multiobjective cognitive automation system, *IEEE Systems Journal*,Early Access, June 2020.