Robust Face Tracking Using Siamese-VGG with Pre-training and Fine-tuning

Shuo Yuan
National Engineering Research Center for E-Learning
Central China Normal University
Wuhan, China
e-mail: yuanshuo98@mails.ccnu.edu.cn

Xinguo Yu
National Engineering Research Center for E-Learning
Central China Normal University
Wuhan, China
e-mail: 2429346468@qq.com

Abdul Majid
Wollongong Joint Institute Central China Normal University
Wuhan, China
e-mail: majid20205020@gmail.com

Abstract— **In order to analyze facial expression in human-computer interaction, real time face-tracking has become significant research problem. Traditional face tracking methods have achieved good results in some constrained environments (such as good illumination, no background interference, etc.) However,these methods require to design manual facial features depending on researcher's experience. In addition,lacking ability for generalization problems is worthy of study. The robustness of face tracking in complex scenes is challenging due to fast moving, multi-scale changes, rotation and occlusion, illumination changes, etc. In view of the above considerations, this paper proposes an improved method based on Siamese-Net to optimize for face tracking tasks. Our work mainly includes four aspects. First, the first two convolutional layers of deeper VGG-16 are used to extract feature.So we call our method Siamese-VGG. Second, we report experiment on face tracking using a pre-trained VGG-Face model which is trained by 2.6M images for face recognition and then fine-tuning to acceleration convergence.Third, in this research the same size crops are input to two branches in the framework and then the inner smaller template feature maps are extracted during training.The proposed method in this paper reduce offset losses by this way. Finally, L2 regularization add to the loss function to improve the generalization ability of the model. The experiment results show better robustness and generalization performance over the original algorithm. In complex scenes, the proposed improved method have achieved the great improvements by almost 11% on average overlap.But ,the frame rate of improved method is still 18.5fps on the Nvidia GTX1070Ti GPU.The improved method proposed in this paper is more practical in terms of speed and accuracy.**

Keywords-Siamese Net;VGG; face tracking;convolutional neural network;L2 regularization

I. INTRODUCTION

Face tracking is to predict the face's position and size of the next frame according to the initial face's bounding box of the first frame in a given video sequence. It has important significance in the research of analyzing facial expression in human-computer interaction.

Commonly used face tracking methods include: methods based on organ feature [1], methods based on skin color information [2, 3], methods based on motion information [4] and so on. These methods use relevant heuristic knowledge to define the search space and achieve the goal of fast tracking. However, most of these methods only use part of the face information, in some typical constraints (such as background features of video itself, in good light, etc.) to achieve better face tracking effect. However, face tracking performance is poor in complex dynamic scenes.

The traditional face tracking method requires manual design features. P.M.Antoszczyszyn in [1] uses organs features, Maricor Soriano in [2] uses the skin color features, and Zhong Rui in [3] uses the front and rear frame motion correlation to predict the next frame face position. Manual design features are relying on the experience of researchers, lack of generalization ability for problems, and problem do exist with this approach that requires more storage space to store manual features. In addition, in the case of complex changes, the traditional methods aren't adapted to the practical application.

The mentioned methods above have various deficiencies, such as facial contours, eyes, nose and other characteristics change with time goes by affected by occlusion and other factors, it will lead to incomplete features and it will make face tracking failure. Facial skin color is easily affected by light, observation angle, etc. The video frame picture is easy to blur during the fast movement of face, which results in the face feature is not obvious. The robustness of tracking faces in complex backgrounds remains to be improved. Lae-Kyoung [4] proposed to use the Camshift algorithm for face tracking. Poor tracking effect in skin-like interference and occlusion, fast motion. Eqi H [5] integrates the Camshift algorithm into the template matching method to solve the skin-like interference problem in the tracking process. Cao FC in [6] improved the face tracking performance of fast motion by using SCKF method, but still could not satisfy the robust tracking of human face in many complex scenes.

In recent years, convolutional neural networks have been successfully applied in the fields of object recognition and

978-1-7281-1593-1/19/$31.00 ©2019 IEEE

object detection, but they are rarely used in tracking tasks. Training deep convolutional neural networks needs a large number of supervised data sets, lack of data sets and real-time requirements of tracking limit the application of convolutional neural networks to object tracking tasks. In 2013, Wang N in [7] applied the deep learning method to the object tracking field, using the Tiny Images [8] data set. The knowledge transfer from offline training to online tracking process enables the tracker to adapt to the appearance changes of moving objects. In 2016, Bertinetto L in [9] proposed a fully-convolutional Siamese network trained end-to-end on the ILSVRC15 [10] data set for object detection in video. The method achieved advanced performance in VOT2017 [11]. In 2015, Omkar in [12] proposed that the VGG-Face network is used for face recognition tasks to obtain higher recognition accuracy.

The object tracking problem in any scenario can be seen as a visual task related to the target.In this paper, we propose a method to replace the traditional approach that requires researchers to manually design feature. A deep neural network is trained to solve the face similarity learning problem in the initial offline phase, and then the function is simply evaluated online during the tracking phase. This method can be extended to track face robustly in any complex backgrounds.

II. SIAMESE-VGG FRAMEWORK AND TRAINING

The face tracking fails in complex scenes due to changes in face features. If we continuously update the face feature template during the tracking process, the tracking effect in complex scenes may be greatly improved. This paper makes some improvements on the basis of the Siamese network to make the algorithm better meet the visual task. The algorithm is mainly divided into two stages, training similarity model and online tracking test.

Figure.1. Training image pairs: Fill the average RGB value beyond the picture position.

The training stage mainly extracts a large amount of data from the T-frames from the YouTube Faces[8](YTF) data set, as shown in Fig. 1, for learning a similarity function, which is used to measure the similarity between the face template samples and the larger search samples. If two sample images contain the same face, a high score is generated or a low score is generated. The network structure is shown in Fig. 2.

In the tracking stage, it is not necessary to store the appearance features of the previous face, and only the face feature map of the previous frame and the search area of the next frame nearly four times larger than the next frame are similarly evaluated to obtain the face center coordinate deviation. Then we can update the coordinates of the next frame face according to the coordinate deviation. We unify the template image and the search image size to the same size, and reduce the pooling layer step size in the convolutional neural network to increase the feature map, and then crop the template image online to reduce feature loss and improve tracking accuracy.

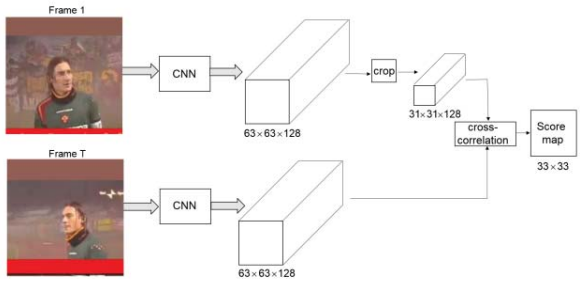

Figure.2. Siamese-VGG Framework

A. *Face Tracking Steps Using Siamese-VGG*

The entire face tracking algorithm flow is divided into the following five steps:

(1) Define visual tasks.

(2) Training data processing: As shown in Fig. 1, we cut the face bounding box and the context boundary offline and cut a large number of fixed-size template images and candidate search images for training the model. More specifically, if the face bounding box width is (w,h) and the context margin is p, the scaling factor s makes the area of the cropping rectangle equal to a constant A, as follows:

$$s(w+2p)\times s(h+2p)=A \tag{1}$$

In experiment, we set $A=255^2$, and the context margin $p=(w+h)/4$.

(3) Load the pre-trained VGG-Face model and initialize the network parameters.

(4) Fine-tune the Siamese-VGG model parameters using the YouTube Faces (YTF) data set. The training data uses the YTF data set, which contains 1595 video sequences of different people for a total of 3,425.

(5) Evaluate the model with test set in complex scenes and evaluate the performance of the tracker.

B. *Feature Extraction*

The original Siamese-Net algorithm uses AlexNet [13] to extract features. Similarly, AlexNet is used for feature extraction in RCNN [14]. In 2014, Simonyan K [15] proposed a network named VGG-16 with deeper depth but smaller convolution kernel size. The VGG-16 neural network model has achieved better results in object recognition . In 2015, Ross Girshick [16] proposed that the VGG-16-based Fast-Rcnn network used for object detection

to achieve a higher mAP than RCNN. Similarly, in 2015, Ren S et al.in [17] proposed Faster-rcnn, and in 2016, Liu W et al. In [18] used VGG-16 [15] for feature extraction, and these detection tasks' performance achieved state of art. Smaller convolution kernels and deeper networks significantly improve the accuracy of image classification and image recognition.

This paper proposes to replace the CNN layer in the original algorithm with the VGG-16 network. Due to the GPU memory limitation, the evaluation of the tracking performance using two to four layers of convolution is experimentally determined. The final feature extraction structure is shown in Figure 3.

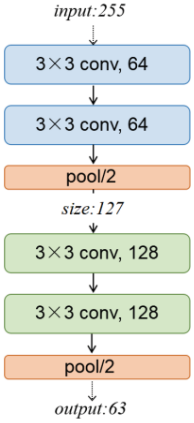

Figure.3. Convolution network structure and corresponding template image size

C. Pre-training and Fine-tuning

Dumitru Erh et al. [19] demonstrated in a large number of experiments that the gain obtained by unsupervised pre-training is more obvious as the number of network layers increases, and it is found that the network without pre-training adopts the penalty of L1/L2 regularization. Not pre-trained. Dimitrios Marmanis et al. [20] used Image-Net Pre-trained Networks to solve the Earth Observation Classification problem, and the classification accuracy was greatly improved. In this study, a novel pre-training and fine-tuning pipeline strategy is introduced in the face tracking task. In the first stage, unsupervised pre-training, the feature extraction layer is initialized by the first two layers of VGG-FACE convolution layer trained on a large number of data sets. In the second phase, fine-tuning, end-to-end fine-tuning training on the YTF data set.

D. Learning Procedure

The face region can be defined by the horizontal and vertical coordinates of the centre point, the width and height of the area, and our task is to use the Siamese-VGG network to predict the next frame of the face frame based on the first frame of the face bounding box coordinates. As shown in Fig. 2, the Siamese-VGG network is divided into two branches. It is assumed that the size of the face frame of the first frame is (w,h), and the centre of the first frame is surrounded by the centre point (x,y). Crop the image of size A (such as formula (1)) and input example images to the first line through the convolutional neural network f_p to obtain the feature map, and then crop out a smaller example feature map z', while at the Tth frame (x,y) off-line cropping candidate search area image input the second line which is processed by the convolutional neural network f_p to obtain the feature map x', and the convolutional neural network used here is shared with the convolutional neural network used by the first line. Parameters, and finally calculate the similarity of the feature maps extracted by the two lines online to obtain a similarity score map:

$$g_p(z',x') = f_p(z') * f_p(x') \tag{2}$$

E. Bounding Box Prediction

In the tracking test phase, in order to adapt to the multi-scale variation of the face, we cut the candidate search area of three different scales S' to the uniform size A near the centre point of the second line, and scale the step size of the candidate search area. For S Finally, three similarity score maps (the left side of the equation (2) equal sign) are used to calculate the offset between the highest score and the centre point, and then map back to the original T-frame image to obtain the T-frame face. Centre point coordinates (x',y'). We calculate the size (w',h') of the face bracketing frame of the Tth frame according to formula (3).

$$(w',h') = (1-s\_LR)\times(w,h) + s\_LR\times s(w,h) \tag{3}$$

Here, s_Step and s_LR are hyper parameters. To compensate for the positional offset caused by multi-scale changes, we add a Hamming window to the score map.

F. L2 Regularized Logistic Regression

Ian's "Deep Learning" [20] proposed adding L2 regular terms to the objective function to make the weight closer to the origin, which means that the weight has a smaller effect, and the model complexity is reduced, according to the Occam razor principle [21] In turn, the model can be better fitted.

We add the L2 regular (as in Equation 5) to the defined Logistic objective function (as shown in Equation 4). Experiments have shown that it can improve the generalization ability of the model and achieve better tracking performance.

$$L_0 = \ell(g_p(z'_i, x'_i), y_i) \tag{4}$$

$$L = L_o + \frac{\lambda}{2n}\sum_{\omega}\omega^2 \tag{5}$$

Here, λ is a hyper parameter, in the experiment $\lambda = 0.3$.

III. Experiment

A. Experimental Environment and Parameter Details

The experiment was completed under the Windows 10 system using the deep learning toolkit MatConvnet based on Matlab. NVidia GTX1070Ti GPU was used to accelerate the calculation and measure the frame rate of racker.

In the experiment, the batch size is set 5 when training model, learning rate is annealed geometrically at each epoch from $10^{-3}$to 10^{-5}.

The setting of hyper parameters has an important influence on the accuracy of the tracker evaluation. How to choose the optimal hyper parameter is also a difficult problem. We randomly select some data points for evaluation and choose the best tracking effect in these limited data as shown in Table I. Hyper parameters, but do not represent the most perfect combination.

TABLE I. THE HYPERPARAMETER WITH THE HIGHEST AVERAGE OVERLAP AMONG THE 30 RANDOM DATA UNDER THE TEST SET

Parameter name	Value
scaleStep	1.039
scalePenalty	0.9785
scaleLR	0.6807
wInfluence	0.18
zLR	0.0102

B. Test Set

Since there is no standard test set in the face tracking field, we select 6 face video sequences (No.1-6) from OTB50 [10], OTB100 [11] and annotate a video sequence (No. 7). A total of 7 video sequences (shown in Table 2) were used to evaluate the performance of the tracker.

TABLE II. TEST VIDEO SEQUENCE

No	Number of frames	Resolution ratio	Challenge
1	493	640×480	Motion blur
2	602	640×480	Fast motion, rotation
3	892	352×288	Foreign object occlusion
4	500	128×96	Similar face occlusion, rotation
5	134	241×193	Light change
6	569	320×240	Light change,Janus face
7	171	640×480	Multi-scale changes, expression changes

C. Online Assessment Results

In this research, all the experiments are under the accelerated calculation of NVidia GTX1070Ti GPU with 8GB of memory, the similarity measurement model is obtained by training for about 80h. The average overlap is the average overlap ratio of the face bounding box in all frames. The average overlaps lager, the tracker will more robust. The frame rate reflects the real-time performance of the tracker.

1) How many layers of convolution are used?

VGG-16 has five convolutional layers and three fully connected layers. In this research, VGG-16 is only used for feature extraction. It is necessary to remove the fully connected layer. However, it is still necessary to discuss how many layers of convolution should be used. Due to GPU memory limitations, we compared the performance of face tracking using 2 to 4 layers of convolution. The evaluation results are shown in Table 3. The experimental data proves that the tracking performance is best when using two layers of convolution, which is better than the accuracy of the original algorithm. There is an almost 11% improvement in the average overlap of our model used 2 convolution layers, but the frame rate is still 18.5fps.

In addition, only 20 epochs are required when we use the YTB data set to fine-tune the training model that is initialized by pre-trained VGG-Face model. The method saves a lot of training time but achieves better tracking performance.

TABLE III. TRACKING PERFORMANCE EVALUATION RESULTS FOR DIFFERENT NETWORKS MODEL UNDER THE TEST SETS

Method		Avg. overlap	fps
Baseline-conv5		70.91	35.8
Baseline+YTB		75.32	34
Siamese-VGG	2layers	**81.76**	18.5
	3layers	80.83	12.0
	4layers	75.60	15.7

2) Comparison of tracking performance in complex scenes

In the Siamese-VGG network with two layers of convolutional layer, the algorithm and the original algorithm are compared and measured under different backgrounds. The experimental comparison results are shown in table 4.

TABLE IV. TEST EXPERIMENT RESULTS

No	Avg.overlap	
	Baseline-conv5	***Siamese-VGG***
1	85.98	87.66
2	82.42	83.49
3	59.11	79.05
4	75.39	79.95
5	62.53	86.98
6	59.40	78.87
7	80.28	83.66

The average overlap of our face tracking algorithm in different complex backgrounds up to 80%, which is much better than the original algorithm. The performance is improved by about 20% under conditions of light changes, face occlusion, Janus face, etc. Other performances such as rotation, motion blur, and similar face interference are also improved. The experimental video result refer to https://github.com/sarah3598/facetracking.git

IV. CONCLUSION

In order to tackle face tracking problem in the background of complex dynamic changes, this paper proposes a face tracking algorithm based on Siamese-VGG, which solves the limitation of the feature extraction template of traditional face tracking method, and the algorithm is used in the tracking process. Light changes, motion blur, and occlusion have strong adaptation and processing capabilities, which improves tracking accuracy compared to the original baseline algorithm.It is more suitable for practical applications in terms of speed and time.

Since the algorithm needs to crop 3 scales template images for similarity measurement in the tracking phase, the calculation amount is large, so, in the future we can improve the tracking speed of the algorithm in this aspect.

REFERENCES

[1] X. Xing, K. Q. Wang, and L. S. Shen, “A Real-time Algorithm for Tracking Human Faces Based on Organ Tracking,” Acta Electronica Sinica, vol. 28, no. 6, pp. 24-28, 2000.

[2] C. X. Wang og Z. Y. Li, "A new face tracking algorithm based on local binary pattern and skin color information," in International Symposium on Computer Science & Computational Technology, 2008.

[3] S. Xia, J. Li og L. Xia, "Robust face tracking using self-skin color segmentation," in International Conference on Signal Processing, 2006.

[4] Lae-Kyoung, Lee, Su-Yong, An, and Se-Young, "Efficient face detection and tracking with extended CAMSHIFT and haar-like features," in International Conference on Mechatronics & Automation, 2011.

[5] Y. Q. Huang, X. C. Dong, L. I. Hao, and H. T. Wang, "Improved CamShift Algorithm for Face Tracking," Computer Engineering, vol. 37, no. 2, pp. 180-182, 2011.

[6] F. C. Cao and X. X. Xing, "A Face Tracking Method Based on Camshift and SCKF in Rapid Moving Process," Applied Mechanics & Materials, vol. 668-669, pp. 1025-1028, 2014.

[7] N. Wang and D. Y. Yeung, "Learning a Deep Compact Image Representation for Visual Tracking," in International Conference on Neural Information Processing Systems, 2013.

[8] T. Antonio, F. Rob, and W. T. Freeman, "80 million tiny images: a large data set for nonparametric object and scene recognition," IEEE Transactions on Pattern Analysis & Machine Intelligence, vol. 30, no. 11, pp. 1958-1970, 2008.

[9] L. Bertinetto, J. Valmadre, J. F. Henriques, A. Vedaldi, and P. H. S. Torr, "Fully-Convolutional Siamese Networks for Object Tracking," in European Conference on Computer Vision, 2016.

[10] O. Russakovsky et al., "ImageNet Large Scale Visual Recognition Challenge," International Journal of Computer Vision, vol. 115, no. 3, pp. 211-252, 2015.

[11] M. Kristan et al., "The Visual Object Tracking VOT2017 Challenge Results," in IEEE International Conference on Computer Vision Workshop, 2017.

[12] O. M. Parkhi, A. Vedaldi, and A. Zisserman," Deep face recognition", CA: BMVC, 2015.

[13] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," in International Conference on Neural Information Processing Systems, 2012.

[14] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation," in IEEE Conference on Computer Vision & Pattern Recognition, 2014.

[15] K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition," Computer Science, 2014.

[16] R. Girshick, "Fast R-CNN," Computer Science, 2015.

[17] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: towards real-time object detection with region proposal networks," in International Conference on Neural Information Processing Systems, 2015.

[18] W. Liu et al., "SSD: Single Shot MultiBox Detector," 2015.

[19] D. Erhan, Y. Bengio, A. C. Courville, P. A. Manzagol, and S. Bengio, "Why Does Unsupervised Pre-training Help Deep Learning?," Journal of Machine Learning Research, vol. 11, no. 3, pp. 625-660, 2010.

[20] D. Marmanis, M. Datcu, T. Esch, and U. Stilla, "Deep Learning Earth Observation Classification Using ImageNet Pretrained Networks," IEEE Geoscience & Remote Sensing Letters, vol. 13, no. 1, pp. 105-109, 2016.

[21] Goodfellow, I., Bengio, Y., Courville, A., & Bengio, Y, "Deep learning"Cambridge,CA: MIT , 2016.

[22] Hang L,"Statistical learning method", Beijing,CA:Tsinghua University , 2012, pp. 80-87.