A Face Liveness Detection Scheme to Combining Static and Dynamic Features

Lifang Wu(✉), Yaowen Xu, Xiao Xu, Wei Qi, and Meng Jian

School of Electronic Information and Control Engineering, Beijing University of Technology, 100124 Beijing, China
lfwu@bjut.edu.cn, xuyao_wen@126.com, {xuxiao2013, weiqi}@emails.bjut.edu.cn, jianmeng648@163.com

Abstract. Face liveness detection is an interesting research topic in face-based online authentication. The current face liveness detection algorithms utilize either static or dynamic features, but not both. In fact, the dynamic and static features have different advantages in face liveness detection. In this paper, we discuss a scheme to combine dynamic and static features that combines the strength of each. First, the dynamic maps are obtained from the inter frame motion in the video. Then, using a Convolutional Neural Network (CNN), the dynamic and static features are extracted from the dynamic maps and the images, respectively. Next, the fully connected layers from the CNN that include the dynamic and static features are connected to form the fused features. Finally, the fused features are used to train a two-value Support Vector Machine (SVM) classifier, which classify the images into two groups, images with real faces and images with fake faces. We conduct experiments to assess our algorithm that includes classifying images from two public databases. Experimental results demonstrate that our algorithm outperforms current state-of-the-art face liveness detection algorithms.

Keywords: Face liveness detection · Deep learning · Convolutional Neural Network (CNN) · Static features · Dynamic features

1 Introduction

In recent years, there are increasingly more internet-based applications, such as e-commerce and social networks. These applications require reliable online identity authentication. In the traditional identity authentication systems, either a password or a key is used for authentication. However, the password or key could be separated from the true user. For example, if user B steals the password of user A, user B could then perform any action on the internet by using the identity of user A. In comparison, biometric-based online authentication could preserve the consistence of a user's physical identity and the digital identity which could be extracted from the true user. However, biometrics-based online authentication could be fooled by fake biometrics of the true user, so it is important that these systems can determine the real biometrics.

Consequently, it proposes a new research topic of liveness detection. In this paper we focus on face liveness detection.

The objective of face liveness detection is to determine that a face image is captured from the real face rather than from a fake face such as photographs, videos [1] or 3D-generated faces of the true user [2]. Related methods of face liveness could be classified into dynamic-based and static-based algorithms [3]. In dynamic-based algorithms, local or global motion, such as blinking [4, 5], facial expression change [6], head movements or motion style [7], is used for liveness detection. Conversely, the main idea of static-based algorithms is to extract the texture features for liveness detection. Generally used algorithms include Local Binary Patterns (LBP) [8, 9], Gabor wavelets [9], Histogram of Oriented Gradient (HOG) [9], Local Graph Structures (LGS) [10], focus variation [11], and features learnt using deep learning [13].

Static and dynamic features have different strengths for liveness detection. The static features are generally used to represent texture differences, while dynamic features are used for motion differences. Our algorithm combines these features to gain the strengths from both features. The static features are extracted from the image frame-by-frame by using a trained Convolutional Neural Network (CNN), which includes four convolutional layers, two pooling layers, and a fully connected layer. For the dynamic features, we first obtained the dynamic maps frame-by-frame by extracting the horizontal and vertical optical flow by using the Lucas Kanade (LK) Pyramid method [14]. Then, the dynamic features are extracted from the dynamic maps by using the CNN. Next, the fully connected layers of two networks are connected to form the fused features. Using the fused features, a Support Vector Machine (SVM) classifier is trained to determine the face image.

The contribution of this paper includes the following: (1) Both dynamic and static features are used for liveness detection and (2) Static and dynamic features are extracted using a CNN, which is more efficient than other feature extraction algorithms such as LBP, SIFT and so on.

The remainders of the paper are organized as follows: In Sect. 2, the proposed face liveness detection scheme is described in details. In Sect. 3, the compared experimental results on two public databases (Print-Attack, Replay-Attack) are presented, and the experimental results confirm the efficiency of the proposed scheme; finally, the paper is concluded in Sect. 4.

2 The Proposed Algorithm

2.1 Overview

The framework of the proposed scheme is shown in Fig. 1. Using input video, the static data is obtained from each video frame and the dynamic maps are obtained from the horizontal and vertical optic flows. The CNN is trained using the dynamic maps and the static data respectively, which is used to extract the dynamic and static features. These features are then connected to form the fused features. Finally, the face liveness detection is implemented using a SVM 2-value classifier.

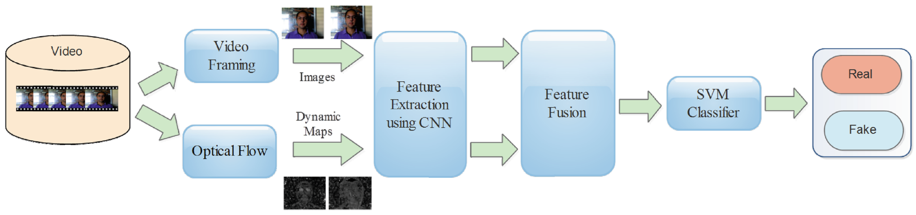


Fig. 1. The framework of the proposed scheme

2.2 Extracting the Dynamic Maps

We use the LK Pyramid optical flow method [14] to track the motion of objects in video. First, the current frame of the image is sampled, then the optical flow method is used to calculate the horizontal direction (F_x) and the vertical direction (F_y) of the current frame and the next frame image, and using the formula $D = \sqrt{F_x^2 + F_y^2}$ to calculate the displacement amplitude diagrams which form dynamic maps. Figure 2 shows dynamic maps from the real face and different fake faces. We can see that the motion style in the dynamic maps from real and fake faces is considerably different. In the dynamic map from the real face (Fig. 2a), there is distinct motion in the face region, and the motion in the region of the human eye is bigger than other regions. In the dynamic map from a photograph (Fig. 2b), there are very small motions in the region of face image. While in the dynamic map from video (Fig. 2c), there are uniform small motions in the face region.

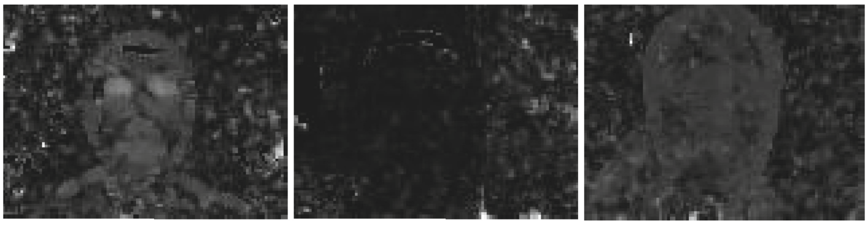

(a) From real face (b) From photograph (c) From Video

Fig. 2. Dynamic maps from real face and fake faces

2.3 Training the CNN Network

The CNN extracts the static features from the original image and dynamic features from the dynamic maps. The architecture of the network is shown in Fig. 3. Four convolutional layers are stacked after the input layer. The first two convolutional layers share the same weight, and they are followed by two max-pooling layers. The last two convolutional layers are locally connected and have independent weights. In the last fully connected layer, each neuron is connected to all other neurons in the fourth convolutional layer. The soft-max output layer includes two values.

The first convolutional layer and the following max-pooling layer have 64 convolution kernels with the size of each kernel being 5 × 5 pixels. The second convolutional layer and the following max-pooling layer also have 64 convolution kernels, with the kernel size being 3 × 3 pixels. The last two convolutional layers are locally connected layers with unshared weights, and every layer has 32 convolution kernels with each kernel being 3 × 3 pixels. There are 160 neurons in the last fully connected layer, and the soft-max layer has two neurons.

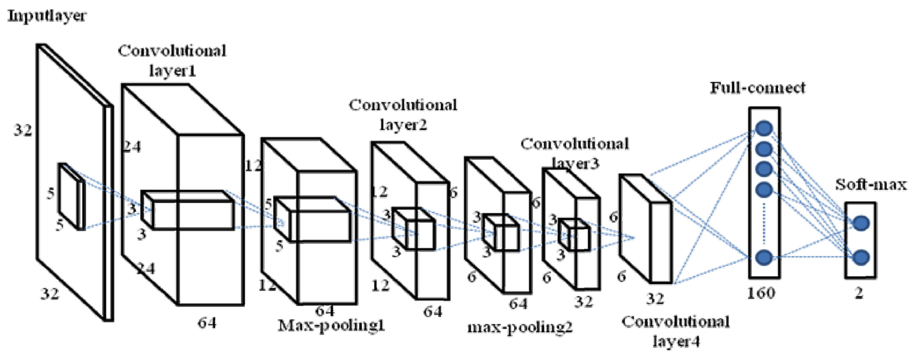


Fig. 3. The structure of CNN

The convolution function could be represented as follows:

$$y_j^{\text{cov}} = \max\{0, \sum_i W_{i,j} * x_i^{\text{cov}} + b_j\} \tag{1}$$

where x_i^{cov} is the i^{th} input and y_j^{cov} is the j^{th} output. W_{ij} is the convolution kernel between the i^{th} input x_i^{cov} and the j^{th} output y_j^{cov}. The symbol * denotes the operation of convolution. b_j is the bias of the j^{th} output. The hidden neuron that we use is a rectified linear unit (ReLU) $f(x) = \max(0, x)$. It is proven to have better fitting abilities than the functions $f(x) = \tanh(x)$ and $f(x) = (1 + e^{-x})^{-1}$[15].

The max-pooling function is formulated as:

$$y_j^{pool} = \max_{k \in D}\{x_i^k\} \tag{2}$$

where D is the non-overlapping local region in the i^{th} input map, y_j^{pool} is the max neuron in D.

The last fully connected layer is fully connected to the fourth convolutional layer, and the function can be formulated as:

$$y_j^{full} = \max\{0, \sum_i x_i^{full} \cdot W_{i,j}^{full} + b_j^{full}\} \tag{3}$$

where x_i^{full} is the i^{th} input of the fully connected layer, which corresponds the i^{th} output of the fourth convolutional layer. Moreover, y_j^{full} is the j^{th} output of the fully connected layer.

The soft-max layer is an n-value output that predicts the probability distribution over n different classes. Our algorithm uses a two-value output, and the soft-max function can be formulated as:

$$y_j^{sm} = \frac{e^{y_j'}}{\sum_m e^{y_m'}}, \tag{4}$$

Where

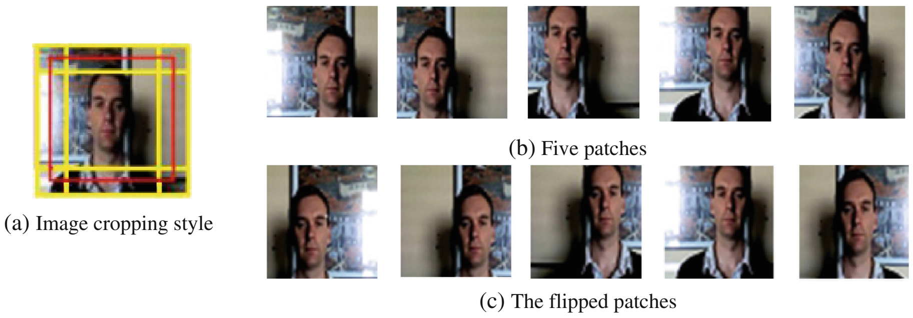
$$y_j' = \sum_{i=1}^{n} y_i^{full} \cdot w_{i,j}^{sm} + b_j^{sm}, \tag{5}$$

where y_i^{full} is the i^{th} vector of the fully connected layer.

2.4 Data Preparation

Prior to feature extraction, the original image and dynamic map are normalized to an image of size 32 × 32 pixels, and five overlapping patches of 24 × 24 pixels are cropped from the 32 × 32 input images. These patches correspond to the four corners and central region in the input image. Then, the five patches are flipped horizontally, resulting in a total of ten patches, as shown in Fig. 4.

For static feature extraction, the three components (R, G, B) of the original image are utilized as the input of CNN network, which consists of 32 × 32 × 3 images. For dynamic feature extraction, the dynamic maps including horizontal and vertical motion components as the input of CNN network, which consist of 32 × 32 × 1 images.


(a) Image cropping style

(b) Five patches

(c) The flipped patches

Fig. 4. Illustration of data preparation

2.5 Training the SVM Classifier

In the stage, the fully connected layers of two CNN networks of size 160 nodes are connected to form 320 dimensional fused features. The fused features are utilized to train the SVM classifier.

In the testing stage, only the central patch of size 24 × 24, marked in red color in Fig. 4(a), is cropped as the input of the CNN network for the static image and dynamic maps. The output of face liveness detection is obtained from the output of the SVM classifier.

3 Experimental Results

The algorithm is tested using two databases: Print-Attack [16] and Replay-Attack [17]. These databases are publicly available and there are numerous challenging benchmarks for them. Each database contains a training set, a testing set, and a development set. In our experiments, the training set is used to train the CNN network, the development set is used to train the SVM classifier, and the algorithm is tested using the testing set. The experimental results are evaluated using the Detection Rate and Half-Total Error Rate (HTER). Detection Rate is the ratio of the number of correctly classified videos to the total number of videos. HTER is defined as half of the sum of the False Rejection Rate (FRR) and the False Acceptance Rate (FAR).

3.1 Experiments on Idiap Print-Attack Database

The Print-Attack Database consists of 200 valid access videos and 200 print attack video attempts for 50 clients, which were captured in controlled and uncontrolled imagining conditions by using a webcam at 25 fps with a resolution of 320 × 240 pixels. The 400 video clips were divided into three groups: the training set (60 valid access videos and 60 print attack videos), the development set (60 valid access videos and 60 print attack videos), and the testing set (80 valid access and 80 print attack videos). This database includes two different scenarios: (i) controlled background (i.e., a uniform background), and (ii) adverse background (i.e., a non-uniform background). Example images from the database are shown in Fig. 5. In the experiments, we only use 200 frames of each video.

We compare our scheme with DMD + LBP + SVM^E [7], DMD + LBP + SVM^F [7], Partial Least Squares (PLS) [20], Non-rigid Motion Analysis [19], Face-Background Consistency Analysis [19], Image Banding Analysis [19], and Fusion of Multiple Clues [19]. The experimental results are shown in Table 1.

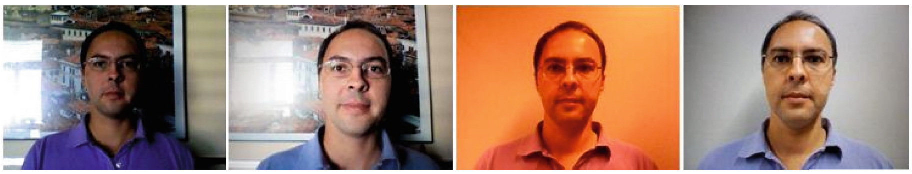


Fig. 5. Example images from Print-Attack Database

Table 1. Compared performance on Print-Attack Database

Method	Detection rate (%)	HTER (%)
DMD + LBP + SVM^E [7]	–	0
DMD + LBP + SVM^F [7]	–	0
PLS [20]	99.375	–
Non-rigid Motion Analysis [19]	90	–
Face-Background Consistency Analysis [19]	97.5	–
Image Banding Analysis [19]	97.5	–
Fusion of Multiple Clues [19]	100	–
The proposed scheme	100	0

From Table 1, we can see that our algorithm obtained the best performance. In [19], three clues are extracted for liveness detection, (1) Non-rigid Motion Analysis, (2) Face-Background Consistency, and (3) Image Banding Analysis. When these three clues are used independently, the detection rates are 90 %, 97.5 %, and 97.5 % respectively. The performances of these three algorithms are lower than our algorithm. When fused three clues, the detection rate can reached 100 %, however, it needs to make a choice according to the different backgrounds. In Ref [20], the features from HSC, CF, GLCM, and HOG are fused for face liveness detection and they could obtain the accuracy of 99.375 %.

3.2 Experiments on Idiap Replay-Attack Database

The Replay-Attack database consists of 1200 videos which include 200 valid access videos and 1000 attack videos. The attack videos were generated using three techniques: (1) print attack, (2) mobile attack, and (3) high-definition attack. The 1200 video clips were divided into three groups: the training set (60 valid access videos and 300 attack videos), the development set (60 valid access videos and 300 attack videos), and the testing set (80 valid access videos and 400 attack videos). The imaging conditions for the Replay-Attack database are similar to those for the Print-Attack database.

We compared our scheme with DMD + LBP + SVM^E (entire video as input) [7], DMD + LBP + SVM^F (face region as input) [7], AO + Random [13], Spoofnet + Random [13], LBP-TOP [21], LBP + LDA [22], HOOF + LDA (thresholding) [18], and HOOF + LDA (NN) [18], as shown in Table 2. LBP-TOP and HOOF + LDA are designed based on only the motion features and obtained the highest HTER 1.25 %. DMD + LBP + SVM, AO + Random and Spoofnet + Random used only the images as input. DMD + LBP + SVM obtained HTER 3.75 % with the entire video and HTER 0 % with the face region. AO + Random and Spoofnet + Random both are deep learning based algorithms. AO + Random algorithm is designed based on hyperopt-convnet. AO + Random is better and obtains a detection rate of 98.75 % and HTER of 0.75 %. Our algorithm obtains HTER 0 % and detection rate 100 %.

In our viewpoints, the proposed scheme performs better than other state-of-the-art algorithms because the following two reasons: First, we utilize both the static and

Table 2. Compared performance on Replay-Attack database.

Method	Detection rate (%)	HTER (%)
DMD + LBP + SVM^E [7]	–	0
DMD + LBP + SVM^F [7]	–	3.75
AO + Random [13]	98.75	0.75
Spoofnet + Random [13]	–	3.5
LBP-TOP [21]	–	8.51
LBP + LDA [22]	–	13.87
HOOF + LDA (thresholding) [18]	–	4.38
HOOF + LDA (NN) [18]	–	1.25
The proposed scheme	100	0

dynamic data. Second, the CNN framework could extract the dynamic and static features more efficiently.

4 Conclusion

In this paper, we discuss a face liveness detection algorithm that combines static and dynamic features. The static features are extracted directly from images by using the CNN network, and the dynamic features are extracted from dynamic maps by using the CNN network. The dynamic maps are obtained using LK optical flow. Finally, the static and dynamic features are connected to form fused features. The fused features are input to the two-value SVM classifier for liveness detection. The compared experimental results with the state-of-the-art algorithms show that the proposed algorithm achieved significantly better performance. From these experimental results, we could draw the following conclusions: (1) Both static and dynamic features are useful for face liveness detection; (2) Deep learning is an efficient method for feature extraction. But whether the fusion of static and dynamic features or the CNN framework is most important, this problem will be studied in our future research.

References

1. Pan, G., Sun, L., Wu, Z., Lao, S.: Eyeblink-based anti-spoofing in face recognition from a generic webcamera. In: Proceedings of the Computer Vision, IEEE ICCV, pp. 1–8 (2007)
2. Erdogmus, N., Marcel, S.: Spoofing face recognition with 3D masks. IEEE Trans. Inf. Forensics Secur. **9**(7), 1084–1097 (2014)
3. Wu, L., Xu, X., Cao, Yu., Hou, Y., Qi, W.: Live face detection by combining the fourier statistics and LBP. In: Sun, Z., Shan, S., Sang, H., Zhou, J., Wang, Y., Yuan, W. (eds.) CCBR 2014. LNCS, vol. 8833, pp. 173–181. Springer, Heidelberg (2014)
4. Bharadwaj, S., Dhamecha, T.I., Vatsa, M., et al.: Computationally efficient face spoofing detection with motion magnification. In: 2013 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), pp. 105–110. IEEE (2013)

5. Jee, H.K., Jung, S.U., Yoo, J.H.: Liveness detection for embedded face recognition system. Enformatika, 235–238 (2011)
6. Komulainen, J., Hadid, A., Pietikäinen, M.: Face spoofing detection using dynamic texture. In: Park, J.-I., Kim, J. (eds.) ACCV Workshops 2012, Part I. LNCS, vol. 7728, pp. 146–157. Springer, Heidelberg (2013)
7. Tirunagari, S., Poh, N., Windridge, D., et al.: Detection of face spoofing using visual dynamics. IEEE Trans. Inf. Forensics Secur. **10**(4), 762–777 (2015)
8. Määttä, J., Hadid, A., Pietikainen, M.: Face spoofing detection from single images using micro-texture analysis. In: 2011 International Joint Conference on Biometrics (IJCB), pp. 1–7. IEEE (2011)
9. Määttä, J., Hadid, A., Pietikainen, M.: Face spoofing detection from single images using texture and local shape analysis. Biometrics, IET **1**(1), 3–10 (2012)
10. Housam, K.B., Lau, S.H., Pang, Y.H., et al.: Face spoofing detection based on improved local graph structure. In: 2014 International Conference on Information Science and Applications (ICISA), pp. 1–4. IEEE (2014)
11. Kim, S., Yu, S., Kim, K., et al.: Face liveness detection using variable focusing. In: 2013 International Conference on Biometrics (ICB), pp. 1–6. IEEE (2013)
12. Yang, J., Lei, Z., Liao, S., et al.: Face liveness detection with component dependent descriptor. In: 2013 International Conference on Biometrics (ICB), pp. 1–6. IEEE (2013)
13. Menotti, D., Chiachia, G., Pinto, A., et al.: Deep representations for iris, face, and fingerprint spoofing attack detection, EprintArxiv (2014)
14. Bouguet, J.Y.: Pyramidal implementation of the lucas kanade feature tracker description of the algorithm. Acta Pathologica Japonica **22**(2), 363–381 (2000)
15. Krizhevsky, A., Sutskever, I., Hinton, G.: Imagenet classification with deep convolutional neural networks. In: Proceedings of NIPS (2012)
16. Anjos, A., Chakka, M.M., Marcel, S.: Motion-based counter-measures to photo attacks in face recognition. IET Biometrics **3**(3), 147–158 (2014)
17. Chingovska, I., Anjos, A., Marcel, S.: On the effectiveness of local binary patterns in face anti-spoofing. In: Proceedings of International Conference on Biometrics Special Interest Group (BIOSIG), pp. 1–7, September 2012
18. Bharadwaj, S., Dhamecha, T.I., Vatsa, M., et al.: Computationally efficient face spoofing detection with motion magnification. In: 2013 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), pp. 105–110. IEEE Computer Society (2013)
19. Yan, J., Zhang, Z., Lei, Z., et al.: Face liveness detection by exploring multiple scenic clues. In: 2012 12th International Conference on Control Automation Robotics and Vision (ICARCV), pp. 188–193. IEEE (2012)
20. Schwartz, W., Rocha, A., Pedrini, H.: Face spoofing detection through partial least squares and low-level descriptors. In: IJCB. IEEE (2011)
21. de Freitas Pereira, T., Anjos, A., De Martino, J.M., Marcel, S.: Can face anti-spoofing countermeasures work in a real world scenario? In: Proceedings of International Conference on Biometrics (ICB), pp. 1–8, June 2013
22. Chingovska, I., Anjos, A., Marcel, S.: On the effectiveness of local binary patterns in face anti-spoofing. In: Proceedings of International Conference on Biometrics Special Interest Group (BIOSIG), pp. 1–7, September 2012