Research Article

Multifeature Fusion Detection Method for Fake Face Attack in Identity Authentication

Haiqing Liu, Shiqiang Zheng , Shuhua Hao , and Yuancheng Li

School of Control and Computer Engineering, North China Electric Power University, Beijing, China

Correspondence should be addressed to Yuancheng Li; ycli@ncepu.edu.cn

Received 2 September 2017; Revised 4 January 2018; Accepted 16 January 2018; Published 8 March 2018

Academic Editor: Jianping Fan

Copyright © 2018 Haiqing Liu et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

With the rise in biometric-based identity authentication, facial recognition software has already stimulated interesting research. However, facial recognition has also been subjected to criticism due to security concerns. The main attack methods include photo, video, and three-dimensional model attacks. In this paper, we propose a multifeature fusion scheme that combines dynamic and static joint analysis to detect fake face attacks. Since the texture differences between the real and the fake faces can be easily detected, LBP (local binary pattern) texture operators and optical flow algorithms are often merged. Basic LBP methods are also modified by considering the nearest neighbour binary computing method instead of the fixed centre pixel method; the traditional optical flow algorithm is also modified by applying the multifusion feature superposition method, which reduces the noise of the image. In the pyramid model, image processing is performed in each layer by using block calculations that form multiple block images. The features of the image are obtained via two fused algorithms (MOLF), which are then trained and tested separately by an SVM classifier. Experimental results show that this method can improve detection accuracy while also reducing computational complexity. In this paper, we use the CASIA, PRINT-ATTACK, and REPLAY-ATTACK database to compare the various LBP algorithms that incorporate optical flow and fusion algorithms.

1. Introduction

Facial recognition is a major research hotspot that has been widely applied to all areas of life, from the registration of certifications, transfers, and accounts of major financial institutions to identity verification in professional exams or mobile phone and entertainment APPs. These applications use a camera to obtain frame-by-frame images of the owner's face, by analysing image attributes to determine whether the user's identity is legitimate. Prompted by personal information attacks and threats, criminals are able to obtain pictures or videos from legitimate users to engage in fake face attacks. This behaviour is a serious threat to personal property security and public safety.

At present, there are three main types of attacks: legitimate users of high-definition photo attacks, video recording attacks, and three-dimensional face model attacks [1].

1.1. Photo Cheating [2]. Photo cheating is one of the most common and convenient ways to attack a biometric identity

authentication system. The use of high-definition photos in front of the system camera is utilized to properly bend, move, and create real face effects. Furthermore, removing the eye part in the photo and using face photos and real human eye rotation all spoof the detection system.

1.2. Video Deception [3]. Video deception [3] uses cameras, pinhole cameras, and other means to shoot videos of the legitimate user, which poses a great threat to the facial recognition system. Compared to the photo, the video can obtain head or facial movement and eye information from the blink of eye movement. Even though this is done via secondary imaging, this attack is still as realistic as a living body.

1.3. 3D Model Cheating [4]. By using the user's photos or videos, attackers can print a three-dimensional facial model. However, three-dimensional models with facial texture and expression can have large gaps when features are extracted from photos and videos. However, with the development of advanced printing technology, a three-dimensional model

may more closely resemble the human face, even to the extent that the human eye cannot identify any differences; therefore, such a facial recognition system poses a great threat.

2. Related Works

Today, there are numerous ways to identify fake faces from a variety of identification features. Many well-known journal and conference papers have carried out in-depth analysis of facial recognition software. The algorithms are divided into the following three categories.

2.1. Image Attribute Analysis. The image attributes that are analysed mainly include three-dimensional depth information, Fourier spectrum analysis, multispectral imaging technology, facial optical flow analysis, and the local binary model. De Marsico et al. proposed a method, based on geometric invariance, to detect three-dimensional objects [2]. They compared zigzag photographs and the geometry of the face. The results showed that the geometric shape of the picture is a “V” figure. However, because the face surface is rugged, it is easy to distinguish the face. Boulkenafet et al. introduced a new method that detects fake face attacks using colour texture analysis [5]. By using the luminance and chrominance channels of the combined colour texture information, complementary low-level feature descriptions are extracted from a variety of colour spaces; the feature descriptions from each image band are combined to produce a feature histogram. Garcia and De Queiroz proposed a detection algorithm based on a moiré search pattern [6]. When a photo or video attack occurs, the image, after digital media sampling, has higher overlapping pixels. A Gaussian filter, with an isolated high frequency mode, is used to extract a descriptor with less training data to obtain a higher recognition effect. Additionally, Määttä et al. researched analytical methods, based on multiscale and multiregional LBP features that were proposed at the 2011 International Symposium on Biometrics [7]. However, highly pixelated photos and video used for image attribute detection still pose a large threat.

2.2. Movement Attribute Analysis. Kollreider et al. proposed an interactive living detection method; this system requires the user to perform a set of simple actions, for example, reading words or verbally identifying images/figures, to analyse whether lip movement is a consistent feature to determine whether a user is a live person [8]. In addition, Arashloo et al. proposed multiscale dynamic texture descriptors based on binary statistical images, which feature three orthogonal planes (MBSIF-TOP) and are effective in detecting spoof attacks [9]. Next, by combining multiscale, local, phase quantization representation, the robustness of the spoofing attack detector can be further improved. Kim et al. proposed a real-time, noninterfering algorithm that uses the diffusion rate of single-frame images [10]. By calculating the diffusion speed between the true and false faces, a local velocity model was defined. The result is inputted to the linear SVM classifier by using the antispoofing and full flow scheme.

2.3. Static Dynamic Binding Analysis. In addition to the point mentioned above, Komulainen et al. improved the method of their predecessors and proposed the use of motion correlation analysis and face texture analysis to achieve in vivo detection [11]. Through the combination of dynamic and static analysis methods, this method can be more comprehensive as a face detection method. Experimental data of the article were collected using a camera to obtain video information; features from the adjacent frame contrast and face texture analysis were intercepted. Facial recognition technology has been increasingly applied to smart devices. Smith et al. identified a set of response mechanisms from a video sequence and digital watermarking in conjunction with the feature extraction test on smart terminal devices [12]. The method achieved very good test results.

In summary, the current fake face detection techniques have launched a wide range of research. Whether it is a feature of facial image extraction or human-computer interaction, all methods have achieved good results. However, highly pixelated images and video attacks still have many difficulties. Most methods require interpersonal interaction to identify fake image faces in video attacks. However, if you master the interactive content, video attacks can also be evaded.

In this paper, fake face analysis is made based on optical flow and a local binary model. This work is used to obtain facial motion characteristics and facial features rather than fusing multiple methods. This way, the method reduces its dimensionality and finally trains the SVM classifier to make accurate true and false judgements. Meanwhile, the public CASIA database will be used to identify potential test volunteers. The method proposed in this paper does not need the users to cooperate with one another, and it has some hidden mechanisms (optical flow) for obtaining dynamic information. Meanwhile, the extracted feature dimension is low using this algorithm, which reduces the computational cost and complexity of the algorithm. This article introduces the fake face detection method in detail throughout the second section. Meanwhile, the experimental results are compared in the third section. The fourth section summarizes the full paper.

3. Fusion of Fake Face Detection Algorithms Based on a Variety of Features

For high-definition photos and videos, our naked eye cannot easily distinguish between a real and fake face. We also find it difficult for the computer to analogously identify real and fake faces via image attributes. This paper proposes a fusion algorithm that can resolve high-resolution image and video attacks. First, the LBP feature is extracted from the facial part. LBP [16–18] is an effective texture description operator, which has significant advantages such as rotational invariance and grey-scale invariance. It plays an important role in the extraction of texture features from the video photo face and the real face. Simultaneously, it also uses the optical flow algorithm [19–21] to extract motion-based features on the facial features. This is largely because, when a person's face is in front of the camera, he will unconsciously blink, move lips, and have other facial movements. Figure 1 shows some

FIGURE 1: (a) Real face of the video image to capture the interval of 10 frames of the picture. (b) Face images taken from HD image attacks. (c) HD images of the human eye part with special treatment, adding eye movement. (d) Face images taken from video recordings.

real faces with the best picture and video contrast. Each video is 5 seconds long and randomly intercepts two pictures of the adjacent 10 frames.

By examining Figure 1, the naked eye can identify the true face. However, it is still difficult to distinguish between faces found in some high-definition pictures and videos. Although careful analysis of the video images can identify local movement such as blinking, simply bending the photo can also create facial movement. However, there is no local movement that is relative to the overall vector movement. The movement in the video and the real face are very similar.

FIGURE 2: 3 * 3 window pixel arrangement that delineates the processing steps.

However, by analysing facial texture, the characteristics of the real face and the video face are very different. In summary, the combination of dynamic and static methods used in this paper is theoretically valid.

3.1. Local Binary Pattern (LBP). The local binary pattern (LBP) [22], as an operator of texture description, is often used to measure and extract an image with local texture information, which has the significant advantages of grey-scale and rotation invariance. Since the LBP-based 3 * 3 algorithm was proposed, it has enabled a resurgence of facial recognition research. For example, it extended 3 * 3 areas to arbitrary neighbourhoods and used circle neighbourhoods instead of square neighbourhoods. The basic LBP algorithm calculates nine pixels and then compares their weight. The computing method compares the periphery of the eight pixels; if the centre is greater than the others, the value is set to 1; otherwise, the value is 0. The LBP weight is calculated as follows:

$$\text{LBP}(x_c, y_c) = \sum_{p=0}^{p-1} 2^p s(i_p - i_c), \quad (1)$$

where (x_c, y_c) is the centre point, the pixel i_c . i_p is the pixel of the neighbour of (x_c, y_c) , and $s(x)$ is the threshold function:

$$s(x) = \begin{cases} 1 & \text{if } x \geq 0 \\ 0 & \text{else.} \end{cases} \quad (2)$$

The largest drawback of the basic LBP operator is that it encompasses only a small area of calculation and is not robust to light and noise. First, we extracted the face image from the video frame via our lab. Then, we treated the part of the image with the face in a greyness and noise reduction way. Finally, we extracted the necessary features via the LBP algorithm. This paper proposes a pixel-neighbour operator of relational grey pixels based on multiscale LBP features. Figure 2 shows a 3 * 3 window pixel arrangement.

The neighbouring-pixel-based LBP algorithm in this paper is processed by the following steps.

FIGURE 3: (a) Original face and the improved LBP features image. (b) Image attack face and the improved LBP features image. (c) Image attack with a face real eyes and the improved LBP features image. (d) Video attack face and the improved LBP features image.

Figure 2 reflects the nine pixels. The upper left corner pixels act as the starting point. Each pixel value is determined in clockwise order, that is, integration into a row: S9 S8 S7 S6 S5 S4 S3 S2 S1.

Next, the arrangement is converted into binary codes. Starting from the highest level, S9 is compared with the current value of pixels as well as with the next pixel. Moreover, the last pixel is compared with the first pixel. If the current pixel is greater, then the value is set to 1. Otherwise, the value is set to 0. The formula is expressed as follows:

$$s_{(x)} = \begin{cases} 1 & s_{(x)} - s_{(x+1)} > 0 \\ 0 & s_{(x)} - s_{(x+1)} \leq 0 \end{cases} \quad (3)$$

Finally, we determine the final contrast and compare it to the binary code of the original pixel. The average of 1 on behalf of the pixel minus the average of 0 represents the value of the pixel.

Figure 3 shows the output characteristics of a characteristic figure that used LBP.

We found that the attack face feature is obviously not as clear as the true face of the user. This is because the reflection of the real face is diffused. Since picture or video reflections have a fixed angle, this step will reduce the image characteristics that have been collected. In addition, the feature graph, extracted by the video, has some colour effects that are obviously different compared with other characteristics.

To reduce approximation errors caused by the local area and fixed radius of this kind of algorithm, the article also joins multiscale expressions with pyramid image resolution, which represents a series of expressions as image multiscale expressions. Every layer of the pyramid image has a unique size and resolution. The images in the floor in the pyramid have higher resolution. The lower layer's output is used as the next layer's input characteristics; this increases the

algorithmic complexity produced by multiple outputs, which is effectively avoided.

3.2. Optical Flow Analysis. The definition of optical flow is the calculation of position changes between two consecutive frames [19]; it has important value in micromotion analysis. The front of the camera's facial movement belongs to the micromovement, and background and light do not change. Usually, a person blinks, moves his or her lips, or twitches his or her head. Photos or video structures are two-dimensional, so after the secondary imaging, the feature points for the movement and the true face are not consistent.

Assume a pixel point on the image is $A(x, y)$; the brightness at t time is $G(x_0, y_0, t_0)$, and the brightness at $t + 1$ time is $G(x + \Delta t, y + \Delta t)$. The movements of the point a vector are expressed by $u(x, y_0)$ and $v(x_0, y)$ on behalf of the vertical and horizontal movement vector, respectively.

$$\begin{aligned} u &= \frac{dx}{dt} \\ v &= \frac{dy}{dt} \end{aligned} \quad (4)$$

After Δt time, the point's brightness is $G(x_0 + \Delta t, y_0 + \Delta t, t_0 + \Delta t)$. The point brightness is the same as when the Δt infinite is close to zero. The brightness can be expanded using the Taylor formula:

$$\begin{aligned} &G(x_0 + \Delta t, y_0 + \Delta t, t_0 + \Delta t) \\ &= G(x_0, y_0, t_0) + \frac{\partial G}{\partial x} \Delta x + \frac{\partial G}{\partial y} \Delta y + \frac{\partial G}{\partial t} \Delta t + \varepsilon. \end{aligned} \quad (5)$$

When using the optical flow calculation image features, the brightness is the same as LBP, since they both use the pyramid structure to calculate the image in multiscale. Finally, the least squares method is applied to solve the basic

FIGURE 4: Real face extract from the video streaming random.

optical flow equation for all the pixels in the neighbourhood. The algorithm is not sensitive to noise and is not affected by noise and edge effects. When the image is stratified to calculate the optical flow, the feature points of the motion are more effectively differentiated. Generally, the establishment of the pyramid does not require too many layers, so this paper utilizes a three-tier pyramid structure. For the upper layer, the pixel coordinates are $A^L(x, y)$; the coordinate points calculated in the lower part of the pyramid are $A^{L+1} = A^L/2^{L+1}$.

Each layer of the image will output the optical flow and transformation matrix. The transformation matrix is the smallest grey difference matrix vector between adjacent frames. The iterative calculation of the optical flow is delivered via the upper layer of the optical flow matrix U ; then, the transformation matrix A is delivered to the next layer. Finally, the optical flow and superposition are determined.

Figure 4 indicates the two images extracted by our lab at random. Our Optical Flow algorithm needs a window size that can be used to classify a pixel; thus, it has an impact on our feature extraction processing.

Figure 5 shows the feature using this algorithm, which shows the size effect of different windows on the image. The first line of the picture shows the direction of motion between two different frames, via our optical flow algorithm. When the window size is 29, the number of the feature vectors meet our needs and the characteristics of the facial movements are more apparent. The second line reflects the corresponding heat map, which has different colours to indicate different features.

To select the best size, we calculated many window sizes in our lab (Figure 6). The experiment indicates that oversized windows increase the complexity of the subsequent calculation and reduce the subsequent accuracy. Thus, our experiment selected a window size of 29.

As the pyramid structure from top to bottom shrinks, the image size and its resolution lead to reduced optical flow information. In this paper, the top layer is defined as the beginning. Then, calculating to the bottom cumulatively, so each layer of the light flow is kept relatively small, but final calculation of the optical flow increases.

3.3. Multifeature Fusion Detection Method (MLOF) Based on Static and Dynamic Combination. In this paper, a multifeature fusion scheme is proposed that combines the texture operator LBP and the optical flow algorithm, while also

classifying the true and false face by SVM. The flow chart is shown as Figure 7.

Step 1. First step is face extraction and background extraction. Two adjacent frames are taken as input images from recorded real face and fake face videos. Then, the optical flow is used to compare the two images to determine whether there is a motion vector while the LBP uses one of the computing features. The extracted image is converted into the grey level, and then the partial image is extracted.

Step 2. Second step is optical flow analysis and LBP analysis. Take two images for optical flow calculation. The steps are as follows:

- (1) Different images are used to reduce the impact of noise.
- (2) Establish the pyramid of image and calculate the size and resolution of each image in the Pyramid. In this experiment, we use a Gaussian filter to process the image, which is to say, the $K + 1$ -layer image can be obtained by smoothing and subsampling the K -layer image, and the pyramid contains a series of low-pass filters. To obtain the image of the first $K + 1$ layer, the K layer image needs to be described by the convolution of the Gauss kernel. Then, all the even rows and columns are removed. The resulting graph is 1/4 of the original image, which reduces the image processing. By iterating over the input image, the whole pyramid appears. The convolution function is as follows:

$$g_{k(i,j)} = \sum_{n=-2}^2 \sum_{m=-2}^2 w(m,n) g_{l-1}(2i+m, 2j+n). \quad (6)$$

$w(m,n) = w(m) * w(n)$ is the Gaussian convolution kernel of length 5. Then, the image is expanded in each layer. The expansion parameter is set to 14 pixels. Then, the horizontal and vertical features of the upper image are superimposed. Figure 8 shows the size of $32 * 16, 64 * 32, 128 * 64$.

- (3) Using optical flow function to determine the characteristics of this layer,
 - (a) obtain two frames of the current pixel matrix R and obtain the difference matrix with the two-matrix subtraction;
 - (b) four feature maps are for the R square and gradient matrix multiplication. Image pixel difference is obtained by the gradient matrix;
 - (c) comparing the value of each difference over the specified threshold, calculate the point moving vector over the threshold.
 - (d) By calculating the sum of the vectors, we obtained two channel pixels. The two-channel pixels are merged into a single channel pixel, and the characteristics of the current image are outputted.

FIGURE 5: Feature abstracted by optical flow with window sizes 3, 9, and 29.

FIGURE 6: Performance evaluation with different window sizes.

(4) Extracting LBP feature: the image is divided into blocks ($16 * 16$ pixels in the experiment).

- The $3 * 3$ pixels of the block are arranged clockwise, and the centre pixel is placed in the last bit or the first position. According to the arrangement of adjacent pixels, a 01 code is obtained.
- The decimal number and contrast LBP/C are obtained based on the obtained codes.
- The histogram for each block is calculated, and the normalization of histogram is obtained.
- The histograms of every block are connected to obtain the features of the current image.

To compare the MLOF algorithm for feature extraction, we compared our MLOF algorithm's histogram of the image feature to real faces and three different attacks. Figure 9 indicates that the characteristics of the real face are stronger

FIGURE 7: Algorithm flowchart.

than the fake face and photo and video attacks significantly reduce the facial features.

Step 3. Third step is analysis and classification of the two characteristics above. The Gaussian kernel function SVM method is used in this study. The two features are distinguished by finding a separating hyper plane. However, it cannot be guaranteed that a hyperplane can divide all the data linearly. In this experiment, we introduced the Gauss kernel

FIGURE 8: Pyramid hierarchical structure.

TABLE 1: Number of pictures on the CASIA dataset.

	Train	Develop	Test	Total
Training				
Real face	1033	995	0	2028
Fake face	1020	987	0	2007
Test				
Real face	0	0	3452	3452
Fake face	0	0	7051	7051

function, which can be mapped into a higher dimensional space. The features obtained by the above algorithms are trained, including the features of the training set and test set. The ROC curves for classification results are drawn.

4. Experimental Result

4.1. Experimental Preparation. To detect the accuracy and efficiency of the experiment, this experiment was tested on the public face video database. All video data provided by the database are collected in an uncontrolled environment. The background region is complex and the light conditions are changeable. To consider different ways of attacking, high-definition image targets are displayed on different mediums, including print on ordinary A4 paper, print on photo paper, and display on a high-resolution screen. In addition, the glasses area of the face of the A4 paper or photo is removed to resist the living detection method based on the blink model.

The subsequent experiments compared the CASIA database, the PRINT-ATTACK database, and the REPLAY-ATTACK database. The data collection of the above datasets are, respectively, shown as Tables 1, 2, and 3.

4.2. Experimental Comparison. To test the effectiveness of the algorithm, this paper extracts the LBP and optical flow features to both the true and false faces for the classification training via SVM. The detection accuracy is shown in Table 4, and the accuracy of experimental algorithm and other commonly used algorithms to the human face are shown in Table 5.

TABLE 2: Number of pictures on the PRINT dataset.

	Train	Develop	Test	Total
Training				
Real face	185	394	0	579
Fake face	193	380	0	573
Test				
Real face	0	0	387	387
Fake face	0	0	355	355

TABLE 3: Number of pictures on the REPLAY dataset.

	Train	Develop	Test	Total
Training				
Real face	232	562	0	794
Fake face	351	580	0	931
Test				
Real face	0	0	421	421
Fake face	0	0	323	323

TABLE 4: Compared the fusion algorithm of this experiment to NLBP and LK optical flow.

Sample	TP	TN	Accuracy	Dimension
NLBP	92.78%	93.34%	93.12%	59
LK optical flow	94.33%	91.17%	95.19%	128
Fusion feature	97.03%	96.88%	97.56%	24

TP, the real input is accepted; TN, the fake input is refused.

TABLE 5: Compared the fusion algorithm of this experiment to LBP, HOG, and optical flow.

Sample	FAR	FRR	HTER
LBP	15.79%	47.37%	22.81%
HOG	15.79%	56.03%	35.91%
Optical flow	15.79%	29.82%	40.00%
MLOF	13.35%	25.26%	46.88%

FAR, the fake input is accepted; FRR, the real input is refused; HTER, half of the sum of the FRR and FAR.

Through the data in both Tables 4 and 5, the MLOF algorithm shows a strong ability to distinguish the real and fake faces. The LBP algorithm can detect the face texture, and the optical flow algorithm can detect whether there is movement in the face part. Moreover, the optical flow vector of different attack modes is obviously different. The difference in lighting and the environment of the photo and video secondary imaging also has a great impact. Therefore, the combination of the two to increase the accuracy rate is feasible. In Table 4, the accuracy rate of NLBP is 93.12%, and the accuracy of LK optical flow algorithm is increased to 97.56%. Table 5 shows the comparison results of various feature extraction algorithms. The results indicate that the correct rate and characteristic dimension in this experiment are obviously improved. Tables 2 and 3 show that the experiment can reduce the complexity of the algorithm and reduce

FIGURE 9: Histogram with different attack methods. The top image shows the features with the real face and the image attack face. The bottom image is the feature with the video attack face and image face with real eyes.

the feature dimension while also improving the accuracy rate, whether it is a separate test or a single algorithm.

4.3. Experimental Results. The ROC curves of the LBP, optical flow, HOG, and improved experimental algorithm are as Figure 10(a). Figure 10(b) shows the algorithm's ability to recognize photo attack, video attack, image with real eye attack, and the total three-way attack.

The operator accepts that the ROC curve is used to evaluate the algorithm performance. Through the ROC curve and the relationship between the positions of the axis, the operator can visually examine the algorithm's ability to distinguish between the real and fake face. The closer to the line $y = 1$ the ROC curve is, the better the performance of the algorithm is. As shown in Figure 10, each point on the ROC curve represents the corresponding FRR and FAR values at different thresholds. The performance evaluation of the entire ROC curve for the algorithm has nothing to do with the selection of the specific threshold. This feature also makes the ROC curve better reflect the robustness and can be applied to compare the use of the various algorithms. It can be seen from the ROC diagram in Figure 10 that the fusion algorithm of this experiment is superior to other algorithms and has good recognition performance and higher accuracy and robustness for both photo and video attacks.

In Figure 11, we describe, in more detail, the accuracy of our proposed method utilizing the CASIA dataset. MLOF represents the algorithm proposed in this paper, as well as a variety of feature fusion algorithms. This article is used

TABLE 6: Compared the fusion algorithm of this experiment on different dataset.

Dataset	Accuracy		
	LBP	DOG	NLOF
PRINT-ATTACK	85.243%	83.354%	92.762%
REPLAY-ATTACK	87.122%	86.281%	93.734%
CASIA	88.651%	87.593%	96.543%

to use the linear SVM classifier to distinguish true and false faces. Compared to the accuracy of LBP and the DOG algorithm, we can see that our proposed algorithm is superior in accuracy compared to the other two kinds of algorithms.

We also compared the accuracy of the algorithm on multiple databases, and the accuracy result is displayed in Table 6. The results show that the algorithm proposed in this paper has high accuracy in multiple database tests.

Table 7 provides a comparison with the state-of-the-art face spoofing detection technique proposed in the literature. As we can see from Table 7, our proposed fusion analysis approach achieves very competitive performance in multiple datasets compared with other advanced algorithms. Most importantly, the essay's approach is able to reflect the stable performance across all three benchmark datasets.

The computation cost of most of the published methods on face spoof detection is unknown. In this paragraph, we verify the efficiency of our proposed approach. In order to analyse the processing time in detail, we show a comparison

FIGURE 10: (a) ROC curves of the four algorithms; (b) ROC curve of three ways of attack.

FIGURE 11: The accuracy curve of different methods.

of the processing time of the proposed method with that of other methods in Table 8. As we can see from Table 8, we can conclude that the difference between the processing time of the proposed and other approaches is negligible, while our approach significantly outperforms previous ones. Currently, the proposed approach is implemented in MATLAB, likely allowing for further optimizations.

TABLE 7: Comparison between the proposed countermeasure and other literature methods on the three benchmark datasets.

	PRINT-ATTACK		CASIA	MSU
	EER	HTER	EER	EER
CTA [5]	0.8	2.8	3.1	4.9
CNN [13]	6.1	2.1	7.4	-
IQA [14]	-	15.2	32.4	-
Proposed method	0.8	2.0	2.9	3.6

TABLE 8: Comparison of the processing time.

Method	DOG-based [15]	LBP-based [7]	Proposed method
Time	30.7 msec	33.2 msec	34.3 msec

5. Conclusion

Facial recognition technology is user-friendly and direct and has many characteristics to identity authentication, which has been widely used in the financial and banking fields as well as other fields. However, the attacks on facial recognition software have become more sophisticated. The main methods of photo attacks and video attacks along with other types of attacks are always performing perfectly. This paper aims to improve the existing methods of facial recognition technology based on photo and video attacks. The complexity of the algorithm and the dimension of the feature values are reduced by using a pyramid structure. Because the output optical flow algorithm in this paper possesses dual-channel characteristics when merging the dual channel characteristics into a single channel characteristic, directly calculating their square root (horizontal optical vector and vertical optical vector) may lead to lower accuracy. We will continue to

study a better fusion method. Meanwhile, the classification algorithm of SVM with high accuracy and the depth learning algorithm will both be used to improve the accuracy and computational efficiency. Although the samples for training and testing in all images are people with yellow skin tones, we will verify more experiments with people of other skin tones in our sample library in the future.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported by the Fundamental Research Funds for the Central Universities (2018ZD06).

References

- [1] X. Zhao, Y. Lin, and J. Heikkilä, "Dynamic Texture Recognition Using Volume Local Binary Count Patterns With an Application to 2D Face Spoofing Detection," *IEEE Transactions on Multimedia*, vol. 20, no. 3, pp. 552–566, 2018.
- [2] M. De Marsico, M. Nappi, D. Riccio, and J.-L. Dugelay, "Moving face spoofing detection via 3D projective invariants," in *Proceedings of the 2012 5th IAPR International Conference on Biometrics, ICB 2012*, pp. 73–78, India, April 2012.
- [3] E. Thabet, F. Khalid, P. Suhaiza Sulaiman, and R. Yaakob, "Low Cost Skin Segmentation Scheme in Videos Using Two Alternative Methods for Dynamic Hand Gesture Detection Method," *Advances in Multimedia*, vol. 2017, Article ID 7645189, 9 pages, 2017.
- [4] Y. Tang and L. Chen, "Shape analysis based anti-spoofing 3D face recognition with mask attacks," *Communications in Computer and Information Science*, vol. 684, pp. 41–55, 2017.
- [5] Z. Boulkenafet, J. Komulainen, and A. Hadid, "Face Spoofing Detection Using Colour Texture Analysis," *IEEE Transactions on Information Forensics and Security*, vol. 11, no. 8, pp. 1818–1830, 2016.
- [6] D. C. Garcia and R. L. De Queiroz, "Face-Spoofing 2D-Detection Based on Moiré-Pattern Analysis," *IEEE Transactions on Information Forensics and Security*, vol. 10, no. 4, pp. 778–786, 2015.
- [7] J. Määttä, A. Hadid, and M. Pietikäinen, "Face spoofing detection from single images using micro-texture analysis," in *Proceedings of the 2011 International Joint Conference on Biometrics, IJCB 2011*, USA, October 2011.
- [8] K. Kollreider, H. Fronthaler, M. I. Faraj, and J. Bigun, "Real-time face detection and motion analysis with application in "liveness" assessment," *IEEE Transactions on Information Forensics and Security*, vol. 2, no. 3, pp. 548–558, 2007.
- [9] S. R. Arashloo, J. Kittler, and W. Christmas, "Face Spoofing Detection Based on Multiple Descriptor Fusion Using Multiscale Dynamic Binarized Statistical Image Features," *IEEE Transactions on Information Forensics and Security*, vol. 10, no. 11, pp. 2396–2407, 2015.
- [10] W. Kim, S. Suh, and J.-J. Han, "Face liveness detection from a single image via diffusion speed model," *IEEE Transactions on Image Processing*, vol. 24, no. 8, pp. 2456–2465, 2015.
- [11] J. Komulainen, A. Hadid, M. Pietikäinen, A. Anjos, and S. Marcel, "Complementary countermeasures for detecting scenic face spoofing attacks," in *Proceedings of the International Conference on Biometrics (ICB '13)*, pp. 1–7, Madrid, Spain, June 2013.
- [12] D. F. Smith, A. Wiliem, and B. C. Lovell, "Face recognition on consumer devices: Reflections on replay attacks," *IEEE Transactions on Information Forensics and Security*, vol. 10, no. 4, pp. 736–745, 2015.
- [13] J. Yang, Z. Lei, and S. Z. Li, "Learn convolutional neural network for face anti-spoofing," *Computer Science*, vol. 9218, pp. 373–384, 2014.
- [14] J. Galbally and S. Marcel, "Face anti-spoofing based on general image quality assessment," in *Proceedings of the 22nd International Conference on Pattern Recognition (ICPR '14)*, pp. 1173–1178, Stockholm, Sweden, August 2014.
- [15] Z. Zhang, J. Yan, S. Liu, Z. Lei, D. Yi, and S. Z. Li, "A face anti-spoofing database with diverse attacks," in *Proceedings of the 5th IAPR International Conference on Biometrics (ICB '12)*, pp. 26–31, IEEE, New Delhi, India, April 2012.
- [16] H. Li, S. Xiong, P. Duan, and X. Kong, "Multitarget tracking of pedestrians in video sequences based on particle filters," *Advances in Multimedia*, vol. 2012, Article ID 343724, 14 pages, 2012.
- [17] C. Riess, "Illumination analysis in physics-based image forensics: A joint discussion of illumination direction and color," *Communications in Computer and Information Science*, vol. 766, pp. 95–108, 2017.
- [18] R. Sutoyo, J. Harefa, Alexander, and A. Chowanda, "Unlock screen application design using face expression on android smartphone," in *Proceedings of the 2016 7th International Conference on Mechanical, Industrial, and Manufacturing Technologies, MIMT 2016*, South Africa, February 2016.
- [19] A. Anjos, M. M. Chakka, and S. Marcel, "Motion-based counter-measures to photo attacks in face recognition," *IET Biometrics*, vol. 3, no. 3, pp. 147–158, 2014.
- [20] R. Raghavendra, K. B. Raja, and C. Busch, "Presentation attack detection for face recognition using light field camera," *IEEE Transactions on Image Processing*, vol. 24, no. 3, pp. 1060–1075, 2015.
- [21] P. Motlicek, S. Duffner, D. Korchagin et al., "Real-time audio-visual analysis for multiperson videoconferencing," *Advances in Multimedia*, vol. 2013, Article ID 175745, 2013.
- [22] J. Galbally, S. Marcel, and J. Fierrez, "Biometric antispoofing methods: A survey in face recognition," *IEEE Access*, vol. 2, pp. 1530–1552, 2014.