FACE ANTI-SPOOFING VIA DEEP LOCAL BINARY PATTERNS

Lei Li, Xiaoyi Feng*, Xiaoyue Jiang*, Zhaoqiang Xia*, Abdenour Hadid*+*

* Northwestern Polytechnical University, Shaanxi, China
+ Center for Machine Vision and Signal Analysis, University of Oulu, Finland

ABSTRACT

Convolutional neural networks (CNNs) have achieved excellent performance in the field of pattern recognition when huge amount of training data is available. However, training a CNN model is less obvious when only a limited amount of data is given such as in the case of face anti-spoofing problem. It is indeed not easy to collect very large sets of fake faces. Especially for the fully-connected layers, tens of thousands of parameters need to be learned. To tackle this problem of lack of training data in face anti-spoofing, we propose to explore the incorporation of hand-crafted features in the CNN framework. In our proposed approach, the color local binary patterns (LBP) features are extracted from the convolutional feature maps, which are fine tuned based on the VGG-face model. These features are then fed into support vector machine (SVM) classifier. Extensive experiments are conducted on two benchmark and publicly available databases showing very interesting performance compared to state-of-the-art methods.

1. INTRODUCTION

It is well known that most of existing face recognition systems are vulnerable to spoofing attacks. As the Internet evolves (such as Facebook, Twitter and Pinterest), people can easily download face images and learn how to spoof a biometric system. More specially, a face spoofing attack occurs when someone tries to bypass a face biometric system by presenting a fake face in front of the camera. For instance, in a live demonstration during the International Conference on Biometric (ICB 2013), a female intruder with a specific make-up succeeded in fooling a face recognition system. This example highlights the vulnerability of face recognition systems to spoofing attacks. Depending on spoofing materials, the attacks can be classified into print attacks, replay attacks, and 3D mask attacks. Print attacks can be launched with still face images of the clients (e.g. printed photos), whereas replay attacks use clients' face videos. 3D mask attacks use complex production processes e.g. targeting soft plastics simulating human skin.

The methods for face anti-spoofing can be classified into (i) shallow model based methods and (ii) deep learning based methods. The first category mainly focuses on extracting handcrafted features from face images and training a binary classifier to distinguish real from fake faces. Although methods based on shallow models can achieve satisfactory performance in some specific cases, they tend to suffer when generalizing to unknown spoofing materials and new scenarios. The deep learning methods, on the other hand, aim to build a general model from large amount of training data. As a matter of fact, the amount of fake faces in practice is limited and this is a serious challenge for using CNNs for face anti-spoofing. To tackle this problem, we propose to extract handcrafted features in the deep learning framework. Instead of learning tens of thousands of parameters in fully-connected layers, the local binary patterns (LBP) features are extracted from the convolutional layers and then fed to an SVM classifier.

Furthermore, real and fake faces can be very distinctive in the chrominance channels. For instance, Boulkenafet *et al.* [1] analyzed the impact of different color spaces on face anti-spoofing and presented a shallow model based on color features, achieving fairly good performance. To exploit and combine the effectiveness of color and deep learning, we utilize a deep learning framework combined with LBP features extracted from different color spaces (such as RGB, HSV and YCbCr).

Among the significant contributions of our present work, (i) While most previous methods based on deep learning suffer from the lack of face training samples, we introduce a new deep learning model by fine tuning the VGG-face model. (ii) We combine deep learning with handcrafted features by extracting the LBP descriptions from the convolutional feature maps. (iii) We explore how well different color spaces can be used for describing the intrinsic disparities in deep learning between genuine faces and fake ones. We also perform a fusion study to analyze the complementarity of different color spaces.

2. PROPOSED METHOD

We extract the color LBP features from the convolutional feature maps. The main architecture of our method is illustrated in Fig.1.

2.1. Fine tuning pre-trained model

In previous works using CNNs for face anti-spoofing e.g. [2] and [3], the authors trained the networks with few face samples yielding in inadequate solutions easily over-fitting the da-

978-1-5090-2175-8/17/$31.00 ©2017 IEEE

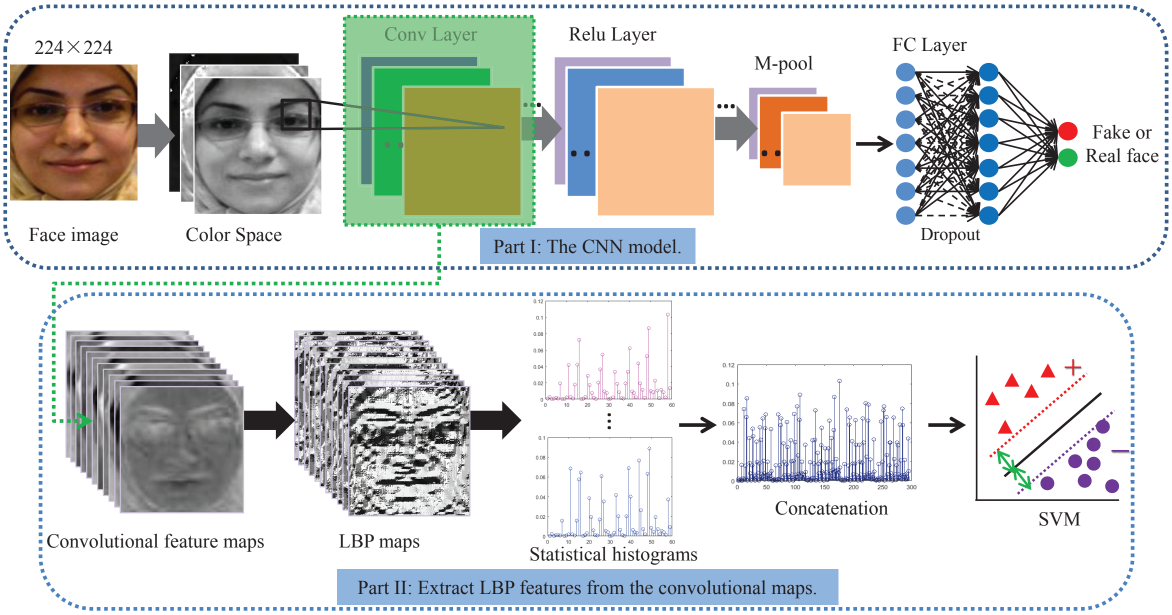

Fig. 1. Architecture of our method based on CNN and LBP. Part I is the CNN model fine-tuned on VGG-face. Part II illustrates is the idea of extracting LBP features from the convolutional feature maps and concatenating the features into one vector. In Part I, Conv Layer is convolutional layer, Relu Layer is rectified linear units layer, M-pool is pooling layer, and FC Layer is the fully connected layer.

ta. Unlike these works, we fine tune existing pre-trained deep learning model called VGG-face, which can reduce the influence of over-fitting problem. Note that the VGG-face model is originally designed for face recognition. Its architecture is given in [4]. VGG-face is trained with a large set of face images (2.6 M images). It has been evaluated on two challenging face recognition databases namnely Labeled Face in the World and Youtube face, yielding state-of-the-art results.

VGG-face is a popular deep learning model originally designed to recognize 2622 people. In order to adapt the model to our spoofing detection problem (a two class problem), we need to change the output of the last fully connected layer from 2622 to 2. The softmaxloss function, as illustrated in Eq.1, is used as the cost function when fine tuning the VGG-face model.

$$Cost = \sum_{i=1}^{n} \{log(e^{y_{i1}} + e^{y_{i2}} + ... + e^{y_{iv}}) + y_{ir}\} \tag{1}$$

where i is the index of training samples, and n is the number of training samples. $Y_i = [y_{i1}, y_{i2}, ..., y_{ir}, ..., y_{ik}]$ is the predict vector of the i_{th} sample, and v is the number of classes (in this paper, $v = 2$). It is noted that the y_{ir} is the predict value of the i_{th} sample.

2.2. Color spaces

RGB is perhaps the most used color space for sensing, representation and displaying of color images. However, it includes three color components (red, green and blue) and does consider (or codify) the information difference between luminance and chrominance. In our work, we explore two other color spaces: the HSV and the YCbCr. Both of these color spaces are based on the separation of the luminance and the chrominance information. In the HSV color space, the hue and the saturation dimensions define the chrominance of the image while the value dimension corresponds to the luminance. The YCbCr space separates the RGB components into luminance (Y), chrominance blue (Cb) and chrominance red (Cr).

2.3. Local Binary Patterns (LBP)

The LBP descriptor [5] is a highly discriminative texture descriptor, especially for face analysis. For each pixel in an image, a binary code is computed by thresholding a circularly symmetric neighborhood with the value of the central pixel. For the sake of simplicity, we use X_i to denote the index of training samples, and X_i^{RGB}, X_i^{HSV} and X_i^{YCbCr} to denote the RGB, HSV and YCbCr color images of X_i respectively. For a face image X_i, the j_{th} convolutional layer's feature maps are given as $C_{X_i}^{j} = [C_{X_i}^{j1}, C_{X_i}^{j2}, ..., C_{X_i}^{jd}, ..., C_{X_i}^{jD}]$, where D is the number of the filters in the j_{th} convolutional layer.

For each pixel (x, y) in $C_{X_i}^{jd}$, the LBP code is computed as shown in Eq. 2.

$$LBP_{P,R}(x, y) = \sum_{v=1}^{P} \delta(r_v - r_c) \times 2^{v-1} \tag{2}$$

where $\delta(x) = 1$ if $x \geq 0$, otherwise $\delta(x) = 0$. r_c and

$r_\upsilon(\upsilon = 1, ..., P)$ denote the intensity values of the central pixel (x, y) and its P neighborhood pixels located at a circle of radius $R(R > 0)$, respectively. The occurrences of the different binary patterns are collected into a histogram to represent the image texture information. An LBP pattern is defined as uniform if its binary code contains at most two transitions from 0 to 1 or from 1 to 0. For example, 01110000 (2 transitions) and 00000000 (0 transitions) are uniform patterns.

To summarize, let $H_{C^{jd}_{X_i}}$ be the texture histogram extracted from $C^{jd}_{X_i}$. For the image X_i, all texture histograms are given by Eq 3.

$$H(X_i) = \{H_{C^1_{X_i}}, H_{C^2_{X_i}}, ..., H_{C^j_{X_i}}, ..., H_{C^J_{X_i}}\} \tag{3}$$

J is the number of convolutional layers in the fine tuned VGG-face model. So $H(X_i^{RGB})$, $H(X_i^{HSV})$ and $H(X_i^{YCbCr})$ are the color texture histograms from X_i^{RGB}, X_i^{HSV} and X_i^{YCbCr} respectively. In our work, we concatenate the texture histograms that belong to the same convolutional layer. So, for each convolutional layer, we obtain the concatenated LBP features as given in Eq. 4.

$$\begin{aligned}\mathbb{F}^j &= \begin{bmatrix} F^j_{X_1} & ... & F^j_{X_i} & ... & F^j_{X_n} \end{bmatrix} \\ &= \begin{bmatrix} H_{C^{j1}_{X_1}} & ... & H_{C^{jd}_{X_1}} & ... & H_{C^{jD}_{X_1}} \\ H_{C^{j1}_{X_2}} & ... & H_{C^{jd}_{X_2}} & ... & H_{C^{jD}_{X_2}} \\ ... & ... & ... & ... & ... \\ H_{C^{j1}_{X_i}} & ... & H_{C^{jd}_{X_i}} & ... & H_{C^{jD}_{X_i}} \\ ... & ... & ... & ... & ... \\ H_{C^{j1}_{X_n}} & ... & H_{C^{jd}_{X_n}} & ... & H_{C^{jD}_{X_n}} \end{bmatrix}\end{aligned} \tag{4}$$

2.4. Classification

After extracting the LBP descriptions from the convolutional feature maps, we use support vector machines (SVM) to learn the classifiers from the training set for face anti-spoofing. In all our experiments, the LIBLINEAR toolkit [6] is used.

3. EXPERIMENTS DATA AND SETUP

We validate our proposed method with extensive experiments on Replay-Attack and CASIA Face Anti-spoofing databases.

Replay-Attack: The IDIAP Replay-Attack database [1] [7] consists of 1300 video clips of real and attack attempts to 50 clients, which are divided into 3 subject-disjoint subsets for training, development and testing (15, 15 and 20, respectively). The genuine videos are recorded under two different lighting conditions: *controlled* and *adverse*. Two type of attacks are created: replay attacks and print attacks. In the replay attacks, high quality video and images of the real client are replayed on iPhone 3GS and iPad display devices. For the print attacks, a high quality images were printed on A4 papers and presented in front of the camera.

CASIA: The CASIA Face Anti-Spoofing Database (CASIA) [2] [8], consists of 600 video recordings of real and attack attempts to 50 clients, which are divided into two subject-disjoint subsets for training and testing (20 and 30, respectively). Three type of attacks are created: video replay attacks, warped attacks and cut attacks. The real and the attack attempts were recorded using three camera devices with: low, normal and high resolution.

For performance evaluation, we followed the overall protocol associated with the two databases. For each database, we used the training set to fine tune the VGG-face model and the testing set to evaluate the performance. On CASIA database, the results are evaluated in term of Equal Error Rate. The Replay-Attack database provides also a development set to tune the model parameters. Thus, the results are reported in term of Equal Error Rate (EER) on the development set and Half Total Error Rate (HTER) on the test set, illustrated in Eq.5.

$$HTER = \frac{FRR(\kappa, \mathcal{D}) + FAR(\kappa, \mathcal{D})}{2} \tag{5}$$

where $\mathcal{D}$ denotes the used database and the value of κ is estimated on the EER. $FRR(\kappa, \mathcal{D})$ means the false rejection rate for the real faces and $FAR(\kappa, \mathcal{D})$ means the false acceptance rate for the fake faces.

4. RESULTS AND DISCUSSION

We present in this section our obtained results. Firstly, we extract the LBP from different color spaces and explore the impacts of different convolutional layers on different databases. Secondly, we concatenate the LBPs extracted from different color spaces and analyze the effect of the concatenation. Finally, we compare the performance of our proposed method against state-of-the-art methods.

4.1. Impact of Different Color Spaces

Table 1 presents the results of the LBP descriptor applied on different convolutional feature maps and different color spaces. For Replay-Attack database, the EER and HTER obtained by RGB LBP are 0.3% and 0.9% respectively. This is much better than those obtained on the HSV and YCbCr color spaces. The best EER on CASIA database is 3.4%, and also obtained using RGB color space. It can be clearly seen that LBP features extracted from the RGB color space yields in the best performance compared to HSV and YCbCr color spaces. Furthermore, we can notice that the convolutional layers which give the best EER and HTER are the 15_{th} and 18_{th}, rather than the deepest layer (22_{th}). This indicates that the features should not necessary be extracted from the deepest convolutional feature maps.

[1] https://www.idiap.ch/dataset/replayattack/download-proc

[2] http://www.cbsr.ia.ac.cn/english/FaceAntiSpoofDatabases.asp

Table 1. The results of single color space from different convolutional feature maps.

Data-base			1 conv	3 conv	6 conv	8 conv	11 conv	13 conv	15 conv	18 conv	20 conv	22 conv
Replay _Attack	RGB	EER	17.8	5.2	7.9	7.2	2.8	1.5	**0.3**	2.7	2.8	4.7
		HTER	17.9	8.8	7.9	7.8	2.6	1.2	**0.9**	3.4	3.7	5.7
	HSV	EER	4.0	4.7	6.7	5.4	3.3	2.9	2.2	6.0	7.1	6.1
		HTER	3.1	5.2	8.1	5.3	2.7	3.5	3.4	7.2	7.4	8.6
	YCbCr	EER	20.5	24.9	6.7	10.9	5.4	4.2	1.6	4.6	5.4	4.8
		HTER	10.0	13.0	12.2	9.2	5.4	5.4	7.2	6.6	7.2	9.1
CASIA	RGB	EER	12.1	19.9	6.4	13.9	4.3	4.3	5.1	**3.4**	4	5.3
	HSV	EER	10.0	13.0	12.2	9.2	5.4	5.4	7.2	6.6	7.2	9.1
	YCbCr	EER	10.2	18.0	11.1	13.0	7.2	6.1	6.4	7.3	9.0	10.0

Table 2. The results of concatenated color spaces from different convolutional feature maps.

Data-base			1 conv	3 conv	6 conv	8 conv	11 conv	13 conv	15 conv	18 conv	20 conv	22 conv
Replay _Attack	RGB-HSV	EER	4.0	4.0	5.2	6.6	2.6	1.5	**0.1**	2.8	2.8	2.8
		HTER	4.4	5.4	4.9	4.9	2.6	1.6	**1.3**	3.5	4.4	4.3
	RGB-YCbCr	EER	13.7	6.1	7.5	6.6	1.5	2.6	0.3	2.8	3.1	3.1
		HTER	14.0	8.1	7.3	6.2	1.8	2.5	1.5	3.9	3.8	3.7
	HSV-YCbCr	EER	5.9	6.6	5.0	5.3	1.2	2.6	0.5	3.5	3.3	4.0
		HTER	5.8	7.9	4.0	6.4	1.1	2.9	2.9	3.3	3.6	4.9
	RGB-HSV-YCbCr	EER	5.3	4.6	4.0	6.4	1.5	1.5	**0.1**	2.8	2.8	2.8
		HTER	6.1	6.2	4.8	6.8	1.4	2.0	1.5	2.9	2.9	4.0
CASIA	RGB-HSV	EER	6.4	16.0	7.0	8.2	2.5	2.5	2.6	3.5	2.6	4.1
	RGB-YCbCr	EER	9.0	10.0	3.3	7.2	5.1	4.2	4.4	5.4	6.0	5.4
	HSV-YCbCr	EER	5.4	7.9	6.3	5.3	2.6	3.1	2.6	4.4	5.1	4.2
	RGB-HSV-YCbCr	EER	6.1	11.0	3.3	3.6	**2.3**	2.5	2.6	4.4	4.2	5.0

4.2. Concatenating Different Color Spaces

To explore the advantages of each color space, we concatenated the LBP features extracted from different color spaces. For example, we considered RGB-HSV, RGB-YCbCr, HSV-YCbCr and RGB-YCbCr-HSV color spaces. The obtained results are shown in Table 2. Comparing the results in Tables 2 and 1, we can see that the overall performance is significantly improved when concatenating the RGB and HSV color spaces. The best EER is reduced by more than 34%, even though the HTER on the Replay-Attack database is slighted increased to 1.3%.

Table 3. Comparison between the performance of our proposed method and state-of-the-art methods

Method	Replay-Attack		CASIA
	EER(%)	HTER(%)	EER(%)
Motion+LBP [9]	4.5	5.1	-
Motion [10]	11.6	11.7	26.6
LBP [7]	13.9	13.8	18.2
LBP-TOP [11]	7.8	7.6	10.6
Spectral cubes [12]	-	2.8	14.0
DMD [13]	5.3	3.8	21.8
Deep Learning [2]	6.1	2.1	7.3
Partial CNN [14]	2.9	4.3	4.5
LBP [15]	0.4	2.9	6.2
Color LBP[16]	**0.0**	3.5	3.2
Proposed Method	0.1	**0.9**	**2.3**

4.3. Comparison against State of the Art

Tables 3 provides a comparison against some state-of-the-art methods for face spoofing detection. It can be seen from this table that our proposed method (combining LBP and CNN) outperforms many state-of-the-art algorithms on the two challenging Replay-Attack and CASIA databases.

More specifically, Table 3 compares the performance with other methods. On the Replay-Attack database, we can see that the state of the art results are obtained in [16] with an EER of 0.0%. Our obtained results (0.1%) are very close. In terms of HTER, our obtained results are nearly four times better than those in [16]. On CASIA database, on the other hand, our proposed method yields in the best EER of only 2.3%. This demonstrates the validity of our proposed approach.

5. CONCLUSION

We proposed to approach the problem of face spoofing detection by combining deep learning and hand-crafted features. We extracted color texture from the convolutional feature maps and investigated how well different color image representations can be used for describing the intrinsic disparities between real and fake faces. The effectiveness of the different deep color texture representations was studied by concatenating different color spaces. Extensive experiments on Replay-Attack and CASIA databases showed interesting results. More importantly, our proposed method was able to achieve stable performance across the two databases unlike most of the methods proposed in the literature.

Acknowledgments: This paper is supported by the Academy of Finland, the Nokia foundation, National Natural Science Foundation of China (No.61502388) and the Fundamental Research Funds for the Central Universities (No.3102015BJ(II)ZS016).

6. REFERENCES

[1] Z. Boulkenafet, J. Komulainen, and A. Hadid, "Face anti-spoofing based on color texture analysis," in *Image Processing (ICIP), 2015 IEEE International Conference on*, Sept 2015, pp. 2636–2640. 1

[2] Jianwei Yang, Zhen Lei, and Stan Z. Li, "Learn convolutional neural network for face anti-spoofing," *Eprint Arxiv*, vol. 9218, pp. 373–384, 2014. 1, 4

[3] David Menotti, Giovani Chiachia, Allan Pinto, William Robson Schwartz, Helio Pedrini, Alexandre Xavier Falcao, and Anderson Rocha, "Deep representations for iris, face, and fingerprint spoofing detection," *Information Forensics and Security, IEEE Transactions on*, vol. 10, no. 4, pp. 864–879, 2015. 1

[4] O. M. Parkhi, A. Vedaldi, and A. Zisserman, "Deep face recognition," in *British Machine Vision Conference*, 2015. 2

[5] T. Ojala, M. Pietikainen, and T. Maenpaa, "Multiresolution gray-scale and rotation invariant texture classification with local binary patterns," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 7, pp. 971–987, 2002. 2

[6] Rong En Fan, Kai Wei Chang, Cho Jui Hsieh, Xiang Rui Wang, and Chih Jen Lin, "Liblinear: A library for large linear classification," *Journal of Machine Learning Research*, vol. 9, no. 12, pp. 1871–1874, 2010. 3

[7] I. Chingovska, A. Anjos, and S. Marcel, "On the effectiveness of local binary patterns in face anti-spoofing," in *International Conference of the Biometrics Special Interest Group (BIOSIG)*, Sept 2012, pp. 1–7. 3, 4

[8] Zhiwei Zhang, Junjie Yan, Sifei Liu, Zhen Lei, Dong Yi, and S.Z. Li, "A face antispoofing database with diverse attacks," in *International Conference on Biometrics (ICB)*, March 2012, pp. 26–31. 3

[9] J. Komulainen, A. Hadid, M. Pietikainen, A. Anjos, and S. Marcel, "Complementary countermeasures for detecting scenic face spoofing attacks," in *International Conference on Biometrics (ICB)*, June 2013, pp. 1–7. 4

[10] T. de Freitas Pereira, A. Anjos, J.M. De Martino, and S. Marcel, "Can face anti-spoofing countermeasures work in a real world scenario?," in *International Conference on Biometrics (ICB)*, June 2013, pp. 1–8. 4

[11] Tiago De Freitas Pereira, Andr Anjos, Jos Mario De Martino, and Sbastien Marcel, "Lbp - top based countermeasure against face spoofing attacks," in *Proceedings of the 11th international conference on Computer Vision - Volume Part I*, 2012, pp. 121–132. 4

[12] A Pinto, H Pedrini, W. R. Schwartz, and A Rocha, "Face spoofing detection through visual codebooks of spectral temporal cubes.," *IEEE Transactions on Image Processing A Publication of the IEEE Signal Processing Society*, vol. 24, no. 12, pp. 4726–40, 2015. 4

[13] Santosh Tirunagari, Norman Poh, David Windridge, Aamo Iorliam, Nik Suki, and Anthony T. S. Ho, "Detection of face spoofing using visual dynamics," *IEEE Transactions on Information Forensics and Security*, vol. 10, no. 4, pp. 762–777, 2015. 4

[14] L. Li, X. Feng, Z. Boulkenafet, Z. Xia, M. Li, and A. Hadid, "An original face anti-spoofing approach using partial convolutional neural network," in *2016 Sixth International Conference on Image Processing Theory, Tools and Applications (IPTA)*, Dec 2016, pp. 1–6. 4

[15] Zinelabidine Boulkenafet, Jukka Komulainen, and Abdenour Hadid, "Face anti-spoofing based on color texture analysis," in *IEEE International Conference on Image Processing (ICIP2015)*, 2015. 4

[16] Zinelabidine Boulkenafet, Jukka Komulainen, and Abdenour Hadid, "Face spoofing detection using colour texture analysis," *IEEE Transactions on Information Forensics and Security*, vol. 11, no. 8, pp. 1–1, 2016. 4