IJNRD.ORG ISSN : 2456-4184
INTERNATIONAL JOURNAL OF NOVEL RESEARCH AND DEVELOPMENT (IJNRD) | IJNRD.ORG
An International Open Access, Peer-reviewed, Refereed Journal

Face Swapping Detection using Machine Learning

Prof.Shubham Unhale[*1], Grishma R. Sakhare[*2], Gauri R. Chinkure[*3] ,Himanshi Khot[*4] , Pallavi Nimgade[*5], Divya Meshram[*6]

[*]Department of Computer Science & Engineering, NAGPUR INSTITUTE OF TECHNOLOGY
Mahurzari, Katol Road Nagpur-441501
Rashtrasant Tukadoji Maharaj Nagpur University, Nagpur

***Abstract*— Face has been used as one of the mainstream manners for user identification. However, with the popularity of face-swapping apps, it takes only a few seconds to change the faces between two facial images. Such swapped results, when using improperly or carelessly, might create some security issues in certain applications. This paper is the first work to address the importance of this issue and discusses the feasibility to achieve an automated face swapping detection through machine learning. Several approaches are tested on a face swapping database derived from a face benchmarking repository. The best solution in the experiments achieved a detection accuracy of over 92%.**

Keywords-face swap, face replacement, face changing, forgery detection, image forensics.

I. INTRODUCTION

Face recognition technology uses an image captured by a digital camera or live data feeds to identify or authenticate an individual, by analyzing the unique characteristics of face regions and comparing them with some known template(s). Such techniques have been widely used and developed in many applications [1, 2, 3] due to it being user friendly, non-intrusive, direct and cost effective, i.e., face images could be easily captured immediately with many available devices. However, such easy-to-be-collected nature also raises the risk that face information might easily fall into the wrong hand and is used in a malicious way.

Techniques attempting to attack such identification or authentication systems are named as face spoofing, which occurs when a person tries to masquerade as someone else by falsifying data and thereby gaining illegitimate access [4]. Specifically, current falsifying data are fake copies of an authorized client's face, in forms such as photos or videos [5, 6]. For example, the attacker can directly download the facial images through social network platform and use such information directly to break the recognition system. Thus, a key research problem here is face liveness detection which aims to identify if the source of the feeding to the system is live or fake. For example, Chingovska *et al*. inspected the

potential of texture features based on Local Binary Patterns (LBP) to detect the face liveness [7]. Extended from this, Pereira *et al*. [8] extended LBP approach into the spatio-temporal domain which achieves a 15% improvement. More recent work can also be found in a survey [9].

With the enormous development in the field of image processing, another new type of face spoofing attracts our attention: instead of directly using the victim's facial image, the attacker can swap the face between two facial images.

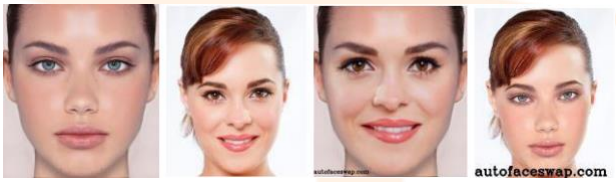

Figure 1. An example of face swapping results using an open software [12]. From left: original face 1, original face 2, swapped result 1, swapped result 2. The original images come from Google Image Search.

This cheap trick can be referred to the ancient Chinese dramatic art "Bian Lian" [10] that the actor changes from one face to another instantaneously. The magic is that the swapping operation is now automated by computer software including mobile and desktop applications [11, 12, 13] and it can be done in just a few seconds. In such a way, the generated image will be a new image which has never been used before and suitable for spoofing purpose. Figure 1 shows some example swapped results using a free face swapping software [12]. The results are quite realistic and sometimes are even impossible for human examiner to detect. To distinguish this scenario from the existing face spoofing techniques, we name this attack as *face swapping*. And in this paper, we refer to the images with and without swapped faces as *swapped images* and *innocent images*, respectively.

To act against this new attack, this paper discusses the feasibility to leverage classic machine learning technique to detect the swapped images based on the visual information derived from the images themselves. To the best of our knowledge, this paper is the first work to address the importance of face swapping detection and introduces an automated solution. From the face images, we firstly detect their keypoints and represent each of them by a descriptor capturing the local information. As the number of such keypoints is not trivial and they are independent from each other, we apply a clustering among all descriptors, the centroids of which compose a codebook. Using such codebook, each image could be encoded to a compact version to represent the distribution of the codebooks' composing entities. This new feature is fed into either linear or non-linear based machine learning to predict its authenticity. This proposed solution can be taken as an additional function to some existing systems (e.g., face verification) to further enhance their security.

The paper is organized as follow: section II introduces our main approach including a part describing how to create a swapped image. Section III delivers the experiments based on a benchmark dataset. Section IV concludes the paper.

II. THE APPROACH

A. Face Swapping

To prepare a sufficiently large face swapping database for learning, we firstly achieve an automated face swapping among a batch of face images. We do not directly use the existing softwares as they can only swap two faces at a time and only through manual operations.

During face swapping, there are mainly four steps as follows (an example is shown in Fig. 2):

1) Determine the facial landmarks. In this work, we extract the 83 landmarks for each face. Two examples are shown in Fig. 2(c-d), from which we can see that the outer contour of these landmarks defines the region to be exchanged.
2) Align two faces using the detected landmarks based on the thin-plate spline model [14]. During this step, the replaced faces might be rotated and scaled to the original face (see Fig. 2(e)).
3) Color correction. To generate realistic faces, we should ensure the skin-tone and lighting between the two images is similar and will not cause obvious discontinuity around the edges of the overlaid region. For this purpose, we applied a Gaussian smooth filter on the mask, using which to blend two face images (see Fig. 2(g)).
4) Blend the second face to the original one using the smoothed mask (see Fig. 2(h)).

B. SURF Descriptor

The face images are usually described by features instead
of directly using the pixel values for the purpose of machine learning. In fact, local feature/descriptor describes a pixel in an image through its local neighborhood contents. Such feature representation should be distinctive and at the same time invariant to the environment changes such as the light conditions.
Speeded up robust features (SURF) [15] is one of the most well-known local descriptor which is robust to small image deformations. It is designed to address the spatial distribution of gradient information within a small neighborhood of a keypoint.
In our work, we choose to use SURF as it has a compact representation. The SURF feature vector of a keypoint only consists of 64 values which enables efficient computation. Moreover, a few approximations for the image response are applied by SURF for further acceleration.
The keypoints of an image could be identified in two different manners. The first one is through some interest point detection approaches, where the interest points are identified at different scales through Gaussian smoothing and sub-sampling. Alternatively, the keypoints can be detected from a regular grid within the image. The rationale behind is to ensure sufficient texture or illumination information

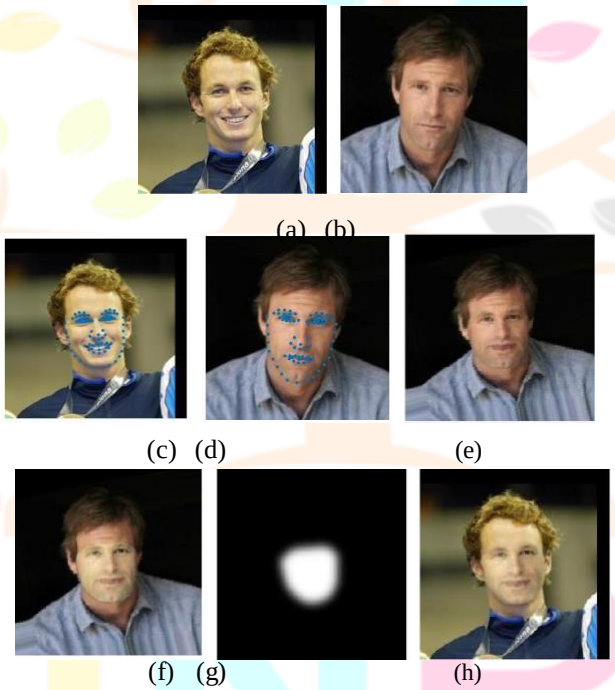

Figure 2. The procedure for face swapping: (a) original image. (b) image to be replaced with; (c) landmarks of (a); (d) landmarks of (b); (e) aligned face of (b); (f) the face region to be cropped from (e); (g) a smoothed mask; (h) swapping result.

within an image could be preserved, which might be missing in the first approach since it usually takes fewer keypoints. However, the grid approach might also include some less discriminative points, which is a tradeoff. In short, local descriptors are extracted at pre-defined grids so that they represent a dense description of the image contents. The two approaches are evaluated in our experiments.

C. Bag of Words

On the obtained SURF features, we adopt a bag of word (BoW) model [16] to represent the image in a more compact and effective manner. There are two stages to generate a BoW model: 1) quantize the descriptors into visual words to form a codebook (codebook generation); 2) find the occurrences in an image of each specific visual word in the codebook and represent them in a histogram (BoW feature construction).
To generate the codebook, we perform a k-means clustering over all selected SURF descriptors from the training data. These training data should include approximately same number of swapped and innocent face images to avoid bias. Denote the descriptors as X and the clusters as C, then we aim to find the N cluster centers by minimizing the sum of squared Euclidean distances between all points and their nearest cluster centers as shown in the equation below,

$$\text{istance}(X, C) = \sum_{i=1} \sum_{j=1} (-)_2, \quad j \quad i \qquad (1)$$

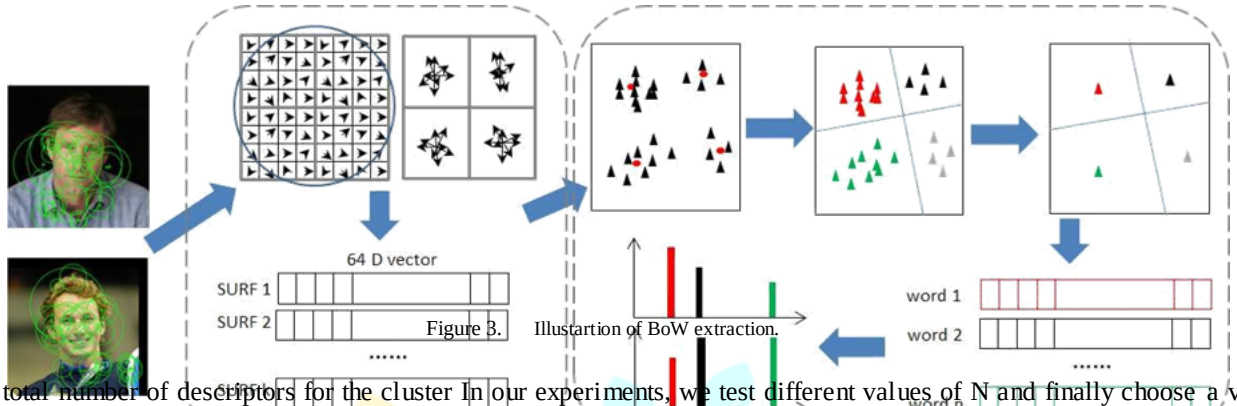

Figure 3. Illustartion of BoW extraction.

where is the total number of descriptors for the cluster In our experiments, we test different values of N and finally choose a value of N=512, using which the performance is sufficiently good. The minimization could be achieved by firstly randomly initializing N cluster centers, and then assigning each descriptor to their nearest cluster center. We re-compute the cluster center as the average of all its composed descriptors and update summed distance using Equation (1) until convergence.

Once we have generated the N visual words, they are composed to a codebook. The next step is to encode such visual words into each query image. Specifically, for a query image, each of its extracted descriptor is mapped to its nearest cluster centroid by,

should be able to predict the attribute of an image which is not included in the training set. Therefore, the performance of the trained classifiers is evaluated on a test set of images exclusive to the training set.

Compared to current popular deep learning tools that require large dateset of hundred-thousands of images for training [20], SVM [17] is a lite tool that can be easily applied to the relatively small-scale task in this paper. More importantly, the data represented by the feature descriptors may not be linearly separable in practice, and the SVM provides two different ways to measure the relationship between two feature vectors, i.e., the linear kernel and the nonlinear radial basis function (RBF) kernel. Formally, if we represent two feature vectors by and , the linear and RBF kernels can be expressed as following, respectively.

=1,⋯,) (−)². (2)

Finally, the image is represented by a histogram which counts how many descriptors belong to each visual word,

$$i^{(.)} = \quad (4)$$

$$(\ ,\ ) = -\|\ -\|. \quad (5)$$

e (·) denotes transpose and ||·|| is the L2-norm of

(3) ector,is the RBF kernel parameter. We employ the

LIBSVM library [17] to tune and choose the parameters for

raining linear and nonlinear models on the training data.

(·) is a

Apart from the SVM classifiers, we also adopt RF [18] nd MLP [19] as comparisons. Specifically, the RF we used s an ensemble of many fisher linear discriminants (FLDs) s base learners. Its performance is presumed to be omparable to the linear SVM classifier. For the nonlinear VM classifier with the RBF kernel, a three-layer MLP is aken as a competitor since it uses the tanh function (i.e., the yperbolic tangent function) for nonlinear activation in the ingle hidden layer.

)W = ([n1, n2,⋯ , n]),

here n_i is the count for cluster, and

ormalization function to ensure no bias will be introduced or images with different number of descriptors. Figure 3 llustrates the conceptual procedure of extracting BoW from he face images.

). *Classifier*

The above process of BoW extraction can be applied to ither an innocent image or a swapped image after face wapping. The task of judging a given image's innocence is hen considered as a 2-class classification (or binary lassification) problem. Concretely, machine learning echniques such as support vector machines (SVM) [17], andom forest (RF) [18] and multi-layer perceptrons (MLP) 19] are adopted to train classifiers on some labeled images, .e., a training set of images. A decently trained classifier

II. EXPERIMENT

. *Dataset*

The LFW face database [21] has been used to create the ace swapping datasets. LFW is a well-known benchmark in

the task of face verification. Specifically, it contains 13,223 color face images each with resolution 250x250. As a preliminary test, we randomly generated 5000 swapped faces using the techniques described in Section II-A, and randomly selected 5000 innocent face images. Among these 10,000 images, 80% of them including half swapped and half innocent are used for training, and the rest 20% are for testing. Note that we have manually excluded some images with special marks among the face regions in the images, e.g., a hand is put on the face. Such processing is to ensure the generated images to be as realistic as possible.

B. Experimental Results

1) The selection of keypoints

As mentioned in Section II-B, we need to determine the keypoints in images using two methods. The first way is to conduct a grid division for each image and we took the grid-step as 8. The second solution is to detect the interest points of an image using an existing solution [15] and treats them as the keypoints. For either of the solutions, the descriptors of keypoints were extracted and were fed to a clustering system to generate the BoW features. Classification is then performed by the SVM classifiers. We reported the results of each in the following two tables (Table I and Table II). We can see that the latter detector-based BoW features are better classified through both the linear and nonlinear SVM classifiers. Comparing the linear and nonlinear kernels, the optimized RBF kernel obtains a significant performance gain than the linear and the default RBF settings.

TABLE I. USING GRID-BASED BOW FEATURE (DIMENSION OF BOW IS 512, RBF-OPT DENOTES SVM WITH OPTIMIZED RBF KERNEL)

	SVM		
	inear	***RBF***	***RBF-opt***
Precision	.6858	0.6408	**0.7831**
Recall	.8360	0.7010	**0.9100**
Accuracy	.7265	0.6540	**0.8290**
F1	.7535	0.6695	**0.8418**

'ABLE II. USING DETECTOR-BASED BOW FEATURE (DIMENSION OF BOW IS 512, RBF-OPT DENOTES SVM WITH OPTIMIZED RBF KERNEL)

	VM		
	inear	***RBF***	***RBF-opt***
Precision	.8263	0.7700	**0.9704**
Recall	.8280	0.8301	**0.8850**
.ccuracy	.8270	0.6790	**0.9290**
F1	.8272	0.7470	**0.9257**

2) The size of bag of word

When compressing the SURF features in the form of BoW, it is necessary to choose a proper number of visual

words. A small size would bring significant degradation to the classification performance but a large size may slow down the computation. To investigate the effect of the BoW size, we set it to 128, 256 and 512, respectively and presented the results as following:

'ABLE III. 'HE EFFECT OF THE BOW SIZE (USING SVM WITH PTIMIZED RBF KERNEL)

	ize of BoW		
	128	***256***	***512***
Precision	0.8249	0.9190	**0.9704**
Recall	0.7820	**0.8850**	**0.8850**
Accuracy	0.8080	0.9035	**0.9290**
F1	0.8029	0.9017	**0.9257**

From the results in Table III, we observed that generally the larger the size of BoW, the better the classification. But the improvement gain from 256 to 521 is much less than that from 128 to 256. Therefore, considering the balance between classification performance and computational efficiency, 512 should be sufficiently good for detecting face swapping.

C. Classifier Performance

We also investigated the performance by adopting other classifiers, which include a linear candidate RF and a nonlinear competitor MLP. The RF takes the default setting in [18] and the hidden layer size of the MLP is set to 800. We also reported the equal error rate (EER) of the three methods as below.

'ABLE IV. ERFORMANCE COMPARISON USING DIFFERENT LASSIFIERS

	inear		***onlinear***	
	SVM-linear	***RF***	***VM-RBF***	***MLP***
Precision	0.8263	0.8113	**0.9704**	0.9342
Recall	0.8280	0.8640	0.885	0.9370
Accuracy	0.8270	0.8315	0.929	0.9355
F1	0.8272	0.8368	0.9257	**0.9356**
1-EER	0.8270	0.8280	**0.9410**	0.9350

We plotted the receiver operating characteristic (ROC) curves to explicitly compare the different classifiers in Figure 4 with the horizontal axis representing the false positive rate and the vertical is the true positive rate. Visually, the closer the curve is to the upper left corner, the better the corresponding method works. In the figure, we can clearly observe the performance difference among the four methods. RF performs similarly to MLP-linear and MLP performs similarly to SVM-RBF, which is as expected as approaches belonging to the same category (linear or nonlinear) work similarly. However, there is a big gap

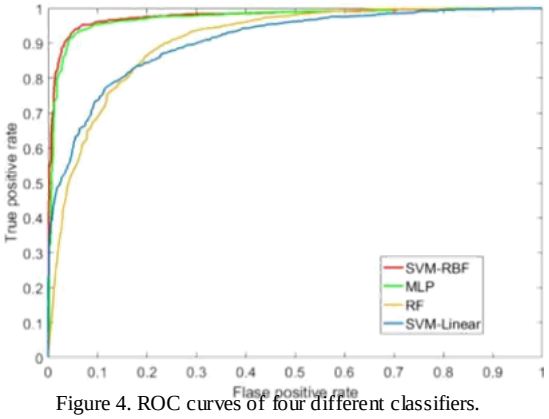

Figure 4. ROC curves of four different classifiers.

between linear and nonlinear classifiers where the later ones work much better. Specifically, SVM-RBF works slightly better than MLP. This can also be observed by their EER values in Table IV.

IV. CONCLUSION

This paper presents a preliminary study on the detection of face swapping attacks. We briefly introduce the process of swapping the face regions between face images of different persons. We establish a dataset including both swapped face images and innocent face images. A feature set of BoW was demonstrated as an effective image representation for describing the face features and providing distinguishable information for the different classifiers to identify the swapped images from the innocent ones. Extending the current work of face swapping detection from still images to live videos would be an interesting direction of our future work.

ACKNOWLEDGMENT

This material is based on research work supported by the Singapore National Research Foundation under NCR Award No. NRF2014NCR-NCR001-034.

REFERENCES

[1] Tao Hu, Honggang Qi, Qingming Huang, and Yan Lu.
[2] See better before looking closer: Weakly supervised data
[3] augmentation network for fine-grained visual classification.
[4] arXiv preprint arXiv:1901.09891, 2019.
[5] [13] Dong Huang and Fernando de la Torre. Facial action trans fer with personalized bilinear regression. In Proc. of the Eu ropean Conference on Computer Vision (ECCV). Springer Verlag, 2012.
[6] Liming Jiang, Wayne Wu, Ren Li, Chen Qian, and
[7] Chen Change Loy. DeeperForensics-1.0: A Large-Scale
[8] Dataset for Real-World Face Forgery Detection. In Proc.
[9] of IEEE Conference on Computer Vision and Pattern Recog nition (CVPR), 2020.
[10] Martyn Jolly. Fake photographs: making truths in photogra phy. 2003.
[11] Tero Karras, Samuli Laine, and Timo Aila. A style-based
[12] generator architecture for generative adversarial networks.
[13] Proc. of IEEE Conference on Computer Vision and Pattern
[14] Recognition (CVPR), 2018.
[15] Pavel Korshunov and Sebastien Marcel. DeepFakes: a New
[16] Threat to Face Recognition? Assessment and Detection.
[17] arXiv preprint arXiv:1812.08685, 2018.
[18] Yuezun Li, Xin Yang, Pu Sun, Honggang Qi, and Siwei Lyu.
[19] Celeb-DF: A Large-scale Challenging Dataset for DeepFake
[20] Forensics. arXiv preprint arXiv:1909.12962, 2019.
[21] Rayhane Mama and Sam Shi. Towards deepfake detection
[22] that actually works. Dessa, Nov 2019.
[23] Yuval Nirkin, Yosi Keller, and Tal Hassner. FSGAN: Sub ject agnostic face swapping and reenactment. In Proc. of the
[24] IEEE International Conference on Computer Vision (ICCV),2019.
[25] Britt Paris and Joan Donovan. Deepfakes and cheapfakes.
[26] United States of America: Data & Society, 2019.
[27] Adam Polyak, Lior Wolf, and Yaniv Taigman. TTS

[28] skins: Speaker conversion via asr. arXiv preprint.

[29] Face Swap Booth – photo faceswap & face changer, https://itunes.apple.com/us/app/face-swap-booth-photo-faceswap-face-changer/id826921329?mt=8

[30] Auto Face Swap, https://www.microsoft.com/en-us/store/p/auto-face-swap/9nblggh3m5nq

[31] Face Swap Live, http://faceswaplive.com/

[32] J. Duchon, 1976, Splines minimizing rotation invariant semi-norms in Sobolev spaces. pp 85–100, In: Constructive Theory of Functions of Several Variables, Oberwolfach 1976, W. Schempp and K. Zeller, eds., Lecture Notes in Math., Vol. 571, Springer, Berlin.

[33] Bay H, Tuytelaars T, Van Gool L. Surf: Speeded up robust features[C]//European conference on computer vision. Springer Berlin Heidelberg, 2006: 404-417.

[34] Qiu G. Indexing chromatic and achromatic patterns for content-based colour image retrieval[J]. Pattern Recognition, 2002, 35(8): 1675-1686.

[35] Chang C C, Lin C J. LIBSVM: a library for support vector machines[J]. ACM Transactions on Intelligent Systems and Technology (TIST), 2011, 2(3): 27.

[36] Kodovsky J, Fridrich J, Holub V. Ensemble classifiers for steganalysis of digital media[J]. IEEE Transactions on Information Forensics and Security, 2012, 7(2): 432-444.

[37] Zheng L, Duffner S, Idrissi K, et al. Siamese multi-layer perceptrons for dimensionality reduction and face identification[J]. Multimedia Tools and Applications, 2016, 75(9): 5055-5073.

[38] Dong Yi, Zhen Lei, Shengcai Liao, and Stan Z Li, "Learning face representation from scratch," arXiv preprint arXiv:1411.7923, 2014.

[39] Hanqing Zhao, Hao Cui, and Wenbo Zhou.] Egor Zakharov, Aliaksandra Shysheya, Egor Burkov, and

[40] Victor Lempitsky. Few-shot adversarial learning of realistic

[41] neural talking head models. In Proc. of the IEEE Interna tional Conference on Computer Vision (ICCV), 2019.