Efficient Transfer Learning for Robust Face Spoofing Detection

Gustavo B. Souza[1](✉), Daniel F. S. Santos[2], Rafael G. Pires[1],
Aparecido N. Marana[2], and João P. Papa[2]

[1] UFSCar - Federal University of São Carlos, São Carlos, SP, Brazil
gustavo.botelho@gmail.com, rafapires@gmail.com
[2] UNESP - São Paulo State University, Bauru, SP, Brazil
{nilceu,papa}@fc.unesp.br, danielfssantos1@gmail.com

Abstract. Biometric systems are synonym of security. However, nowadays, criminals are violating them by presenting forged traits, such as facial photographs, to fool their capture sensors (spoofing attacks). In order to detect such frauds, handcrafted methods have been proposed. However, by working with raw data, most of them present low accuracy in challenging scenarios. To overcome problems like this, deep neural networks have been proposed and presented great results in many tasks. Despite being able to work with more robust and high-level features, an issue with such deep approaches is the lack of data for training, given their huge amount of parameters. Transfer Learning emerged as an alternative to deal with such problem. In this work, we propose an accurate and efficient approach for face spoofing detection based on Transfer Learning, i.e., using the very deep VGG-Face network, previously trained on large face recognition datasets, to extract robust features of facial images from the Replay-Attack spoofing database. An SVM is trained based on the feature vectors extracted by VGG-Face from the training images of Replay database in order to detect spoofing. This allowed us to work with such 16-layered network, obtaining great results, without overfitting and saving time and processing.

1 Introduction

Despite the higher security of biometric systems, criminals are already violating them by presenting forged traits, e.g., facial photographs, to their capture sensors, process known as spoofing attack [1]. In this sense, countermeasures techniques must be integrated to the traditional biometric systems in order to prevent such frauds.

Antispoofing methods proposed so far use, in general, handcrafted features extracted at the moment of identification, e.g., presence of facial movements, to detect whether there is a real or synthetic biometric trait being presented to the

Authors are grateful to grants #2014/12236-1 and #2017/05522-6 from FAPESP, grant #306166/2014-3 from CNPq, and grant #88881.132647/2016-01 from CAPES.

M. Mendoza and S. Velastín (Eds.): CIARP 2017, LNCS 10657, pp. 643–651, 2018.
https://doi.org/10.1007/978-3-319-75193-1_77

sensor. Such handcrafted methods, however, are shown to be not good enough, especially in challenging scenarios, due to their raw data-based approaches.

Recently, deep neural networks have gained attention due to their greater results in many complex tasks [2]. However, despite of being more robust, the training of such architectures is computationally expensive and usually there are no public databases with enough images to avoid overfitting, especially when working with deeper architectures (which usually allow better results). To address such problem, Transfer Learning has been recently proposed, in which the deep architectures are trained on large datasets, even from other domains, and then a classifier, or the own network, is just fine-tuned based on the images of the smaller database of the problem being addressed [2].

In this work, we present a novel efficient method for spoofing detection in face-based biometric systems by means of high-level and robust features, extracted from images by the very deep VGG-Face [3] network, a well-referenced 16-layered Convolutional Neural Network (CNN) model, previously trained on large face recognition datasets. By using such trained network in face spoofing detection (smaller databases available), overfitting is avoided since only a Support Vector Machine is trained on the spoofing images, time is saved and accuracy tends to be preserved due to the similar domain of face recognition and face spoofing detection applications. Results on the traditional Replay-Attack [4] spoofing dataset show that the proposed method, besides of efficiency, presents great results, close to other state-of-the-art techniques, in terms of attack detection.

2 Face Spoofing Detection

In attacks to biometric systems, criminals usually generate synthetic samples of biometric traits of legal users, such as printed facial photographs or latex and gelatin fingers, to fool their capture sensors [5]. Figure 1 shows images of real and fake faces (protographs) presented to a real capture camera. After capture, it is difficult, even for humans, to differentiate between the real and fake ones.

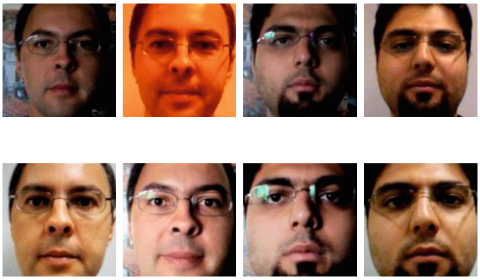

Fig. 1. Real (top) and fake (bottom) faces from the Replay-Attack [4] spoofing dataset.

Among the main biometric traits, face is a promising one especially due to its convenience and low cost of acquisition, being very suitable to a wide variety of environments. However, despite all these advantages, face recognition systems are the ones that most suffer with spoofing attacks since they can be easily fooled even with common printed photographs obtained in the worldwide network. In this context, attack detection methods for face recognition systems are essential.

Antispoofing techniques for face recognition systems have been proposed based on different principles. Nevertheless, spoofing detection is still an open question [6]. As mentioned, most of techniques are based on simple rules (handcrafted features) in order to detect attacks, e.g., facial movements. However, criminals can easily identify these rules and improve attacks: moving the facial photographs in a specific way. In this sense, algorithms able to work with high-level and non trivially generated features are necessary. Among them, the deep learning-based methods simulate the deep structures of neurons in human brain and have outperformed state-of-the-art techniques in many areas [2].

3 Convolutional Neural Networks (CNN)

Convolutional Neural Networks (CNN) [7] are deep learning architectures constituted of layers in which different kind of filters (convolution and sampling) are applied to the input data, initially two-dimensional images. The result of a given layer serves as input to the above one until the top of the network is reached. As shown in Fig. 2, besides convolution and sampling, layers with neurons fully connected can be included at the top of the network for classification, usually performing signal rectification or applying the softmax normalization function, in this case, transforming their input values into output probabilities. At the top layers of the network, high-level representation of the original image are obtained, more robust than the raw pixels information for many applications [1].

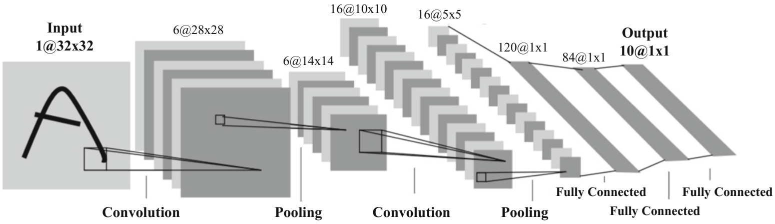

Fig. 2. Illustration of the Lenet-5 [7] model of CNN, with convolution, pooling and fully connected layers. The last one presents softmax neurons for classification of the input images. Sizes of the feature maps generated in each layer are shown at top.

Despite of obtaining high level of abstraction over the input data and, usually, more accurate results, a common problem in working with deep neural networks consists in the fact that, due to their huge amount of free parameters to be optimized (weights of connections, biases, etc.), a huge amount of samples are required to train the architecture in an effective way and avoid overfitting [8].

Besides of demanding time and processing, for many problems there are no public databases with such dimensions.

In order to deal with such issues, still preserving the robustness of the very deep neural networks in problems with lack of training data, Transfer Learning, which consists in applying trained models on large datasets to extract high-level features from the images of other databases (usually smaller), was proposed [2]. The weights of the trained deep network are just fine-tuned based on the training set of the given problem. Other more efficient approach consists in extracting the high-level features from the samples of the target dataset using the trained deep network and training only a classifier based on such features, which is less computationally expensive and also allows great results [9].

An interesting and public available trained model of a very deep CNN, as denoted by the own authors, which can be used in Transfer Learning, is VGG-Face [3] network. It presents 16 layers with convolution, pooling, linear rectification and softmax operations. Each operation has its own set of parameters which are already optimized over the Labeled Faces in the Wild (LFW) [10] and YouTube Face [11] databases (huge datasets) for the face recognition task, problem similar to ours. Details of VGG-Face architecture can be found in [3].

4 Proposed Approach

In most of cases, the very deep neural networks, already trained and used to extract features on other smaller datasets, are trained on data of a different domain of the problem being addressed. In this work, as said, we use a pre-trained model, the very deep VGG-Face [3] network, on a similar domain problem (face recognition), tending to avoid at maximum a decrease in performance. The proposed approach for face spoofing detection presents the following training steps: (i) Face Detection and Normalization; (ii) Deep Features Extraction; (iii) Support Vector Machine (SVM) Training; and (iv) Threshold Calibration.

4.1 Face Detection, Normalization and Deep Features Extraction

After loading VGG-Face [3] model, each frame of the training videos of the assessed Replay-Attack [4] dataset passed through a preprocessing step: the face in each frame was cropped from the original image following the coordinates provided by the own authors of the dataset in order to avoid segmentation errors. We incremented or decremented, by few pixels, the size of the provided rectangles in order to crop only 112×112-sized squares, to facilitate their resizing to 224×224 pixels, dimensions required to feed the VGG-Face [3] network.

Actually, we regularly sampled and worked with only 1 of each group of 8 frames of the videos in order to speed up the feature extraction and SVM Training, obtaining almost $11,650$ training facial images. Such resultant 224×224 facial images were also normalized (subtraction of the mean image of the databases on which the VGG-Face was trained) and presented to VGG-Face, performing a foward pass until the layer "Pool5", last layer before the fully connected ones in the VGG-Face [3] model (almost at the top of the network).

Given an input image presented to the network, the resultant 512 feature maps with size of 7 × 7 generated as output of the layer "Pool5" were also regularly sampled (a quarter of them) and their values, after concatenation, were taken as the feature vector of such training image.

4.2 SVM Training and Threshold Calibration

Given all high-level feature vectors extracted by VGG-Face [3] from the training images (from the real and fake training videos of the Replay [4] dataset), and their respective labels, their values were normalized to the range $[0; 1]$ and a radial basis function SVM classifier was trained. Given all training vectors, we performed the grid-search for finding the parameters c and g of the SVM kernel using a 5-fold cross-validation scheme. After then, the SVM was trained with such parameters and the whole set of training vectors.

Actually, a probabilistic version of the SVM classifier [12] was used. After training, given a facial image from a video being analyzed, the classifier outputed a probability indicating the similarity of such face with the real and fake classes. In order to calibrate the threshold τ, used to determine whether a test face is real or fake given the probability generated by the SVM to it, we used the videos of the development set of the Replay-Attack [4] dataset, varying τ from 0 to 1.

Given each development video, we also sampled it in regularly spaced positions, using the ratio of 1 frame in each 8, also obtaining almost $11,650$ facial images, which feeded the VGG-Face [3] model, after face cropping and normalization, in order to extract their feature vectors and generate their probabilities of belonging to each class. For each facial image, if its output probability was higher than threshold τ being considered, we classified it as real, fake otherwise.

Finally a votation scheme was applied to determine the final class of each development video, i.e., given its classified frames, if at least half of them were considered fake, the video was classified as attack, otherwise the video was taken as real. We varied τ in the interval $[0; 1]$, in steps of 0.01, and calculated the False Acceptance Rate (FAR) and False Rejection Rate (FRR) of the method based on the incorrect classified development videos. We fixed the τ parameter when obtained the same values for FAR and FRR (EER - Equal Error Rate).

4.3 Testing

The same process done for the training and development videos was applied to the test videos after fixing τ. For each test video we also sampled its frames (considering all training videos, a total of $15,500$ frames were sampled), cropped and normalized the faces in them, and passed such images through the VGG-Face [3] network in order to extract their feature vectors and also classify them by comparing their SVM output probabilities with the τ value. If greater than τ, the face was considered as real, and fake otherwise. A votation scheme also was applied to determine the final class of the test videos: real or fakes. Again, if at least half of the classified frames from a given video were considered as fake, the video was classified as attack, and real otherwise.

5 Experiments, Results and Discussion

The proposed approach for face spoofing detection was assessed on the traditional Replay-Attack [4] dataset using a computer with an Intel i7-4790K CPU (8 cores), 32 GB of RAM memory and an NVIDIA GTX980 GPU with 4 GB of RAM (our best computer). Such database contains color videos with frames with 320×240-sized frames of real and fake faces, divided in training (360 videos), development (other 360 videos, only for threshold estimation) and testing sets (with 480 videos). The real videos from all sets present 375 frames while the attack videos present 240 or 230 frames, depending on the type of attack.

The fake videos were made by presenting many kinds of synthetic facial images to the capture cameras: high-definition printed photographs, videos or even photographs exhibited on displays of mobile devices, etc. Figure 1, in Sect. 2, shows examples of real and fake faces present in the frames of the training set of videos from such database. As said, it is difficult to differentiate between them, especially due to high intraclass variability and high interclass similarity.

In our experiments we used the Caffe [13] framework with Matlab R2014b interface (MatCaffe) in order to load the trained VGG-Face model and to perform the foward pass of the detected and normalized facial image from the videos. Given the feature vectors of the faces in the training frames (from all training videos), we used the LibSVM [12] library in order to normalize their values in the range $[0; 1]$ and train a radial basis function kernel SVM. As said, the grid-search was performed in a 5-folds cross-validation scheme and the SVM was trained with the best found kernel parameters ($c = 32.0$ and $g = 0.0078$). After training the SVM (probabilistic version), we used the videos from the delvelopment set to optimize the τ threshold.

After finding the optimal threshold $\tau = 0.18$, we classified the sampled frames from the testing videos and used a votation scheme, as said, to find a final classification for each of them (based on the classes of their sampled frames). As in literature, the results in terms of Half-Total Error Rate (HTER) of the proposed approach and of other state-of-the-art methods are shown in Table 1. The lower the HTER, the better the method. We also provided our Accuracy (ACC) result, measure of performance not mentioned in the other works.

It is important to observe in Table 1 that our efficient approach, based on an already trained very deep CNN on similar domain databases, presents an HTER result close to the state-of-the-art methods: with an HTER of 16.62%, it outperforms the LBP+LDA technique and is close to the other methods. Such methods are computationally expensive: LBP-TOP, e.g., extracts multiple LBP histogramas for lots of small sets of frames of the videos (a slide window is deslocated over all the frames) and concatenates such histograms generating large feature vectors to be processed by an SVM. In such method, in order to classify each video, a votation scheme also is performed.

Table 1. Results (%) on the Replay-Attack database regarding Half-Total Error Rate (HTER) of the proposed method and other state-of-the-art techniques. We also provided our Accuracy (ACC) result. The best values are highlighted.

Method	ACC	HTER
Proposed approach	**93.95**	16.62
Diffusion speed model [14]	-	12.50
LBP+LDA [4]	-	17.17
LBP+SVM [4]	-	16.16
LBP-TOP+SVM [15]	-	**7.60**
Non-linear diffusion [16]	-	10.00

Our architecture only requires to train an SVM based on robust feature vectors directly extracted by the already trained very deep VGG-Face [3] model, being efficient and saving processing time and hardware consumption: the feature extraction, based on the layer "Pool5" of VGG-Face, took about only 0.4 s per frame. It is important to note that the obtained HTER and ACC results of our method can be considerably improved by considering more frames of the videos instead of 1 in each 8 of them, as well as using all the 7×7 output feature maps from layer "Pool5" to represent the images. Also, as done in other works, we could use the top fully connected layers of the VGG-Face [3] network or even a combination of feature maps from many of its layers, including the bottom ones, in order to better represent faces (which present larger dimensions).

The other mentioned approach for Transfer Learning, i.e., fine-tune of the own deep network in the database of the problem, changing the size of the softmax layer according to the number o classes existent, in our case fake and real, also can be performed to improve even more the results. Despite this, it is important to note that all these mentioned approaches require more processing time and computer resources (especially memory space) than the proposed architecture, and, due to this reason, it was not possible to execute them given our hardware restrictions. However, having more powerfull equipaments, they can be performed easily.

6 Conclusion

In this work, we presented an efficient and also robust approach for face spoofing detection based on Transfer Learning by applying an already trained model of a very deep Convolutional Neutal Network (CNN), the VGG-Face network, on the Replay-Attack spoofing database. The proposed approach, due to its work with a deep architecture, is very robust, presenting results close to the state-of-the art techniques on the traditional Replay dataset. Besides, since we just need to train an SVM based on the high-level features extracted by VGG-Face from the training facial images, just performing a foward pass of such images into the trained model, it saves much processing time and also hardware resources, especially in training. Also, since such VGG-Face was trained on images of a similar domain (facial images for face recognition) than our given problem (facial images for face spoofing detection), robustness of the high-level features extracted are strongly preserved. Overfitting is also avoided given the lack of huge amounts of training data for face spoofing detection, since we do not need to train the deep VGG-Face model for robust feature extraction. Based on all this, the proposed approach is very suitable for real applications or in cases where there are not available high-tech computers servers, as ours.

References

1. Menotti, D., Chiachia, G., Pinto, A., Schwartz, W., Pedrini, H., Falcao, A., Rocha, A.: Deep representations for iris, face, and fingerprint spoofing attack detection. IEEE Trans. Inf. Forensics Secur. **10**(4), 864–879 (2015)
2. Bengio, Y.: Deep learning of representations: looking forward. Stat. Lang. Speech Process. **7978**, 1–37 (2013)
3. Parkhi, O.M., Vedaldi, A., Zisserman, A.: Deep face recognition. In: Proceedings of the British Machine Vision Conference (2015)
4. Chingovska, I., Anjos, A., Marcel, S.: On the effectiveness of local binary patterns in face anti-spoofing. In: Proceedings of International Conference of BIOSIG, USA, pp. 1–7 (2012)
5. Ratha, N.K., Connell, J.H., Bolle, R.M.: An analysis of minutiae matching strength. In: Bigun, J., Smeraldi, F. (eds.) AVBPA 2001. LNCS, vol. 2091, pp. 223–228. Springer, Heidelberg (2001). https://doi.org/10.1007/3-540-45344-X_32
6. Patel, K., Han, H., Jain, A.K.: Cross-database face antispoofing with robust feature representation. In: You, Z., Zhou, J., Wang, Y., Sun, Z., Shan, S., Zheng, W., Feng, J., Zhao, Q. (eds.) CCBR 2016. LNCS, vol. 9967, pp. 611–619. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46654-5_67
7. LeCun, Y., Bottou, L., Bengio, Y., Haffner, P.: Gradient-based learning applied to document recognition. Proc. IEEE **86**(11), 2278–2324 (1998)
8. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: Advances in Neural Information Processing Systems, pp. 1106–1114 (2012)
9. Montavon, G., Orr, G.B., Müller, K.-R. (eds.): Neural Networks: Tricks of the Trade. LNCS, vol. 7700, 2nd edn. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-35289-8
10. Huang, G., Ramesh, M., Berg, T., Miller, E.: Labeled faces in the wild: a database for studying face recognition in unconstrained environments. Technical report 07–49, University of Massachusetts, Amherst (2007)
11. Wolf, L., Hassner, T., Maoz, I.: Face recognition in unconstrained videos with matched background similarity. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2011)
12. Chang, C., Lin, C.: LIBSVM: a library for support vector machines. ACM Trans. Intell. Syst. Technol. **2**, 1–27 (2011)
13. Jia, Y., Shelhamer, E., Donahue, J., Karayev, S., Long, J., Girshick, R., Guadarrama, S., Darrell, T.: Caffe: convolutional architecture for fast feature embedding. arXiv preprint arXiv:1408.5093 (2014)

14. Kim, W., Suh, S., Han, J.: Face liveness detection from a single image via diffusion speed model. IEEE Trans. Image Process. **24**(8), 2456–2465 (2015)
15. de Freitas Pereira, T., Anjos, A., De Martino, J.M., Marcel, S.: *LBP−TOP* based countermeasure against face spoofing attacks. In: Park, J.-I., Kim, J. (eds.) ACCV 2012. LNCS, vol. 7728, pp. 121–132. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-37410-4_11
16. Alotaibi, A., Mahmood, A.: Deep face liveness detection based on nonlinear diffusion using convolution neural network. Sig. Image Video Process. **11**(4), 713–720 (2017)