Real-Time Face Liveness Detection and Face Anti-spoofing Using Deep Learning

Ruchi Zawar(✉) and Vrishali Chakkarwar

Government Engineering College, Aurangabad, MS, India
ruchi.zawar21@gmail.com, vachakkarwar@geca.ac.in

Abstract. Face recognition biometrics is now widely employed, thanks to the rapid development of computer vision technology. However, since the facial recognition system cannot tell whether a face image is real or not, it is open to impersonation attempts. A face recognition system should be able to recognize not just people's faces but also spoofing attempts using printed images, videos, or 3D masks.

Examining facial liveness, such as landmark detection, eye blinking, and lip movement is a genuine strategy to avoid spoofing. However, when it comes to video-based replay attacks, this strategy is not sufficient. As a result, this study provides a face liveness detection approach that is integrated with a CNN (Convolutional Neural Network) classifier. The landmark detector module identifies the facial landmarks, the eye module analyses eye liveness, and the CNN classifier module makes up the anti-spoofing approach. We created an anti-spoofing model based on MobileNetV2, which was altered and retrained effectively using the LCC FASD dataset, which is freely available for this purpose. In an effort to get rapid inference time with satisfactory precision, a MobileNetV2's transfer learning is used as the classifier. We subsequently merged these landmark detection, eye liveness detection, and anti-spoofing modules and used the combined result to create a simple facial anti-spoofing and liveness detection application. The test results reveal that the built module can distinguish a variety of facial spoof attacks and has a high level of accuracy of 98%.

Keywords: Deep learning · convolution neural network · face liveness detection · face anti-spoofing · Keras and TensorFlow

1 Introduction

With the current growth in computer vision, and because of its simplicity and accuracy Face recognition systems are gaining more impetus. Face recognition biometrics is widely employed in access control systems, payment methods, and various other applications, where a high level of security is necessary. Face recognition functions similarly to how we recognize other people. This facial recognition system works by taking photographs of a person's face using a camera, then processing the image with a specific algorithm to determine whether the face is recognized from a database or not. But on the other hand, these systems cannot tell whether a face image is real or not, making them

R. Manza et al. (Eds.): ACVAIT 2022, AISR 176, pp. 626–636, 2023.
https://doi.org/10.2991/978-94-6463-196-8_47

open to forgery. Expedients to deceive these systems are getting more complicated, and antidote methods are necessary. Photo, video, and three-dimensional face model attacks are all common types of spoofing attacks. Attacks using photos and videos are more common as anyone can get your photos and videos quite easily from various social media sites or the internet. Face-based liveness detection technology arose as the times demanded in order to detect these forgery attacks. Several researchers have been working on developing a robust facial anti-spoofing solution for face recognition systems [1, 2].

Eye blink detection is a highly accurate method of determining whether or not the face in front of the camera is alive. Natural eye movement tracking is a simple method of determining whether or not a face is real. We also used facial landmarks detection along with eye motion detection for our analysis. However, depending on landmark detection and eye liveness detection are not adequate, as the system still remains vulnerable to video replay attacks using smartphones and tablets.

Active Anti-spoofing techniques include challenges and responses. This strategy relates to a one-of-a-kind activity known as a challenge. The system's purpose is to verify that the challenge has occurred during a video sequence. The user must complete these challenges in order to confirm its identity.

However, while successful, this technique necessitates more input and may have a substantial impact on the user experience [2].

Different texture patterns can be found in both real and spoof facial images. This simple truth is that rebuilding faces from camera photos degrades the clarity of facial features and creates gaps in reflectivity. Various prior research has attempted to capture the difference between real and fake facial images using designed color texture characteristics, such as RGB (Red, Green, and Blue) or LBP (Local Binary Pattern) variations. Fu-Mei Chen et al. use CNN to extract deep features and RI-LBP to extract color texture features from facial photos, based on the color information properties of the images [3]. The difference between high-definition color printed paper and high-definition recorded video, on the other hand, is difficult to tell. Color texture analysis was used by Boulkenafet et al. to develop a liveness detection system [4]. To represent the image, the LBP descriptor's combined color texture information (RGB, HSV, and YCbCr) was extracted and submitted to the SVM classifier for evaluating the authenticity.

On the other hand, the dependence on the lighting conditions of the room is a flaw in this texture analysis technology. In conditions where the room is dark, facial textures using illuminations are difficult to distinguish.

Meanwhile, some research uses different input domain names such as histogram or HSL (Hue, Saturation, and Lightness) to precisely extract the visual features that are more biased and counteract the influence of illumination. Because it is sensitive to noise and motion blur, this type of solution is better for pictures or photo attacks but bad for screen attacks.

3D cameras can distinguish between faces and flat objects and specific pixel depth recommendations could provide great precision against demonstration attacks. This would be the most reliable anti-spoofing method. But cameras, on the other hand, remain the most dependable anti-spoofing solution available.

New studies use deep learning and neural networks for face anti-spoofing. It involves training the neural networks to identify the difference between real and fake face images.

However, there are no uniform or defined features that CNN can see or understand and that is the issue.

That is why it is critical to use a combination of liveness detection methods and CNN analysis approaches for classification. We employ landmark detection (which includes brows, eyes, nose, lips, and jawline detection) combined with an eye movement liveness detection model for face liveness detection and a CNN classification model for spoof detection.

As a result, this article recommends using an improved face liveness detection algorithm with CNN to distinguish between legitimate and malicious faces. It is simple and it is more resistant to diverse attack techniques.

The following are some of the work's major contributions:

1. The proposed method is accurate since it reflects the properties of genuine and fake faces employed by CNN and deep transfer learning methods.
2. It can be implemented quickly and easily without the use of any additional hardware.
3. Our face anti-spoofing algorithms are reliable and detectable in real-time and can handle various spoofing techniques (print, replay, and mask).

2 Related Work

There are various approaches in anti-spoofing works that have been historically influential. The first is texture-based approaches, which use the feature descriptors like HOG (Histograms of oriented gradients) and LBP (local binary patterns), along with standard machine learning classifiers like Support vector machines to complete the task. Steps involved in classic machine learning algorithms for solving face spoofing problem includes processing, segmentation, feature extraction, and classifications [5].

On the other hand, to identify the real and fake faces the temporal methods use patterns generated from face motions like eye blinking, mouth movement, or use movements between the face and the image's background. They also use methods like optical flow to trace the facial movements [6].

Some methods use 3D sensors. These 3D sensors collect depth information from 2D images and check 3D face information and compare the 3D model of the input sample to that of a real face.

But this technique even though effective requires the use of 3D sensors which are not widely available and are likely to be costly [7].

Various studies have used deep learning and convolutional neural networks for training the model to distinguish real faces from false face photos. The author of the paper [8] used a deep learning architecture that includes CNN combined with LSTM for achieving face spoof detection in videos and images. The CNN identified local and dense features from the input sequences, whereas the LSTM captured temporal correlations [8]. Similarly, the paper [9] presents a hybrid architecture that combines CNN and LSTM for face anti-spoofing in video sequences by focusing on motion cues across video frames [9].

Active Anti-Spoof Methods entail the user's active participation in the data capture or enrolling process in order to detect spoofs. The user needs to perform a few tasks in front of the camera like smiling, nodding, blinking eyes, etc. to prove that the person is

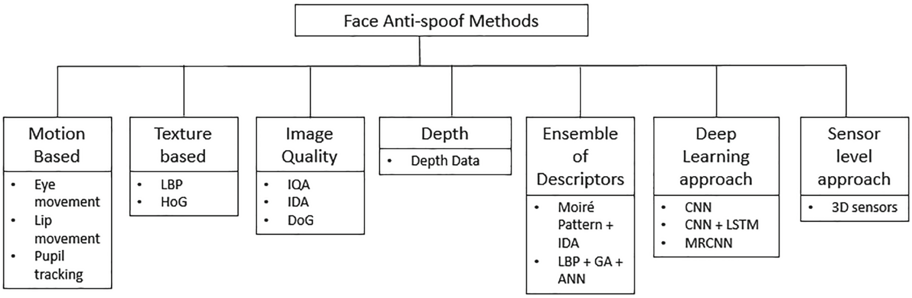

Fig. 1. Face Anti-spoofing methods

real and not fake. On the other hand, Anti-spoofing measures that aren't obtrusive are known as passive anti-spoofing approaches. Here user interaction is not required. The system takes care of everything without the user having to do anything (2). The various types of Face anti-spoof methods are classified as shown in Fig. 1.

3 Methodology

The two key components that we offer for building an anti-spoofing model include the liveness detector and the CNN classifier. The input data will be preprocessed before being sent to the liveness detection module, which will detect face landmarks and eye liveness. The input will then be passed to the CNN classifier module, which will assess if the face is real or fake. If the input passes through both modules, it is declared real. The phases in creating the CNN classifier module include 1) Data preprocessing, 2) model training, 3) model evaluation, and 4) testing (Fig. 2).

3.1 Liveness Detection Module

The Liveness detection module is further divided into two modules: 1) the landmark detection module, and 2) the eye liveness detection module. To find out whether the person in front of the camera is real and alive, more information about the person's face is needed, such as posture, whether the mouth is open or closed, if the eyes are open or closed, whether the person is looking up, and so on.

We use the Dlib library present in python, which provides face detection and landmark detection functions. Dlib employs histogram-oriented methods (HOG) for face

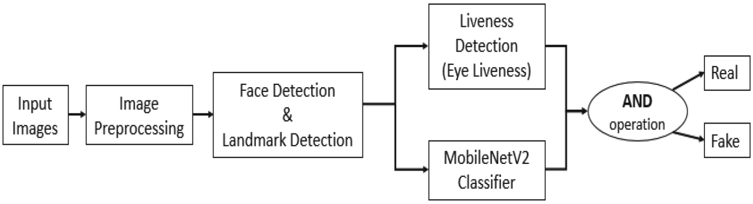

Fig. 2. Proposed Method Overview.

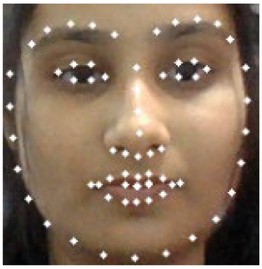

Fig. 3. Example of Landmark Detection on an image.

detection and Kazemi's model for landmark detection [10]. It provides 68 face points (face landmarks) in a swift and objective manner. The dlib's facial landmark detector uses pre-trained models to predict the position of sixty-eight coordinates (x, y) that identify the face features on a person's face as shown in the figure below.

The 68-point iBUG 300-W dataset was used to train the dlib face landmark predictor which detects areas around the eyes, brows, nose, mouth, and jawline. Dlib's landmark detector is fast and reliable and works well in real-time scenarios. It is a crucial element for extracting the movements of the eyes and mouth and detecting liveliness based on the features detected in Realistic scenarios. Also to identify the movement of the eyes of an individual the detected eye features are submitted to the trained KNN model. The model is trained on a labeled dataset of size 200 having different left and right eye ratios to identify whether the eyes are live or not. As input, the Network uses the eyes ratio for the next 20 frames. The eye ratio is defined as the height of the eye divided by the breadth of the eye. We have used the KNN algorithm for training the eye liveness detection model up to 3 epochs and it gives a test accuracy of 93%.

3.2 Classification Module

The second key component of the anti-spoofing model is our classifier model which classifies whether the input image is real or fake.

Dataset

We have used the LCC_FASD dataset which contains three subgroups: training, development, and evaluation. There are 1942 real photos and 16885 spoof faces in the database. The dataset consists of a wide range of high-quality images that are generated using 83 different capturing devices [11]. The table below shows the subject image statistics for each subgroup (Fig. 4 and Table 1).

Pre-processing

Data pre-processing is one of the most important steps that help in removing noisy and unwanted data and keeping only the important data for building our model. We scale, resize the images, perform image augmentation and convert them into array images which pulls the three-channel data from each image's array table. The data is then translated into an average mean value, which aids in extracting the image's best attributes. The

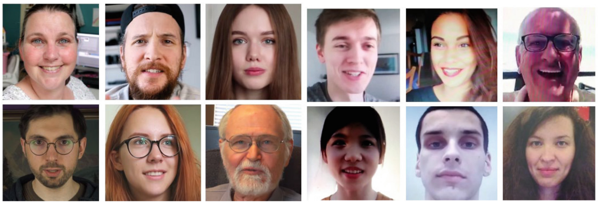

Fig. 4. Examples of real and spoof images.

Table 1. Image statistics in LCC_FASD dataset.

Data	**Subject**	**Real Faces**	**Spoof Faces**
Training	118	1223	7076
Development	25	405	2543
Evaluation	100	314	7266
Total	243	1942	16885

attributes extracted are assigned with the labels which help encode the altered data. After the data is transformed and encoded, it is divided into training, validation, and test data. The train data in the main dataset folder is divided into two groups, with a train and validation ratio of 80–20%. The validation data is used to evaluate the loss during the training.

Model

We have used MobileNetV2 architecture for conducting the binary (real/fake) classification. MobileNet is a network architecture that is better suited for applications where the computational capability is little. In order to give rapid inference time with reasonable precision, a MobileNetV2's transfer learning is used as the classifier. The MobileNetV2 is pre-trained on the imagenet dataset [12]. The pre-trained model is frozen and in the final layers, the model is retrained on the LCC_FASD dataset and customized to give the binary output that is real or fake.

In contrast to traditional residual models, the MobileNetV2 design is based on an inverted residual structure, with the residual block's input and output being narrow bottleneck layers. MobileNetV2 filter features in the intermediate expansion layer with lightweight depth-wise convolutions. Non-linearities in the thin layers were eliminated to maintain representational power [13].

The size of the input has been reduced to 224 x 224 x 3 and the total number of parameters are 3504872. The model network used for training is shown in Fig. 5. The convolution is used to feature out the array of modified values to transmit into subsequent layers and evaluate the best feasible outcome for the categories to predict. The data is processed in MobileNet, with the features being sent to custom layers that aid in understanding the patterns of the picture array and updating its weights and bias using convolution layers. The use of depth-wise convolution, which performs the process in two

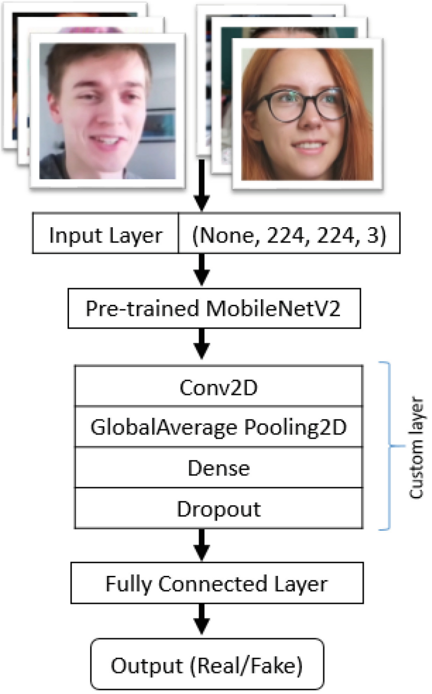

Fig. 5. Training Network Architecture Diagram

parts, first filtering and then combining the features, is one thing that distinguishes our approach. It executes spatial convolution, which is conductedindividually on each channel of a given input, followed by pointwise convolution, which is a 1x1 convolution. The relu as an activation function is used for connecting and processing data with the layers in order to update the bias from one convolution to the next. The Global Average Pooling method generates a median matrix value, which aids in the compression of the array. The dense layer helps to make a decision to evaluate the image's edge and corner array detection and aids in updating the weight parameters for the image pattern prediction process. We have used a dropout to prevent the model from over-fitting. We have used a learning rate of 0.0001, zero-gradient as the optimizer, and binary cross-entropy for calculating the loss. The accuracy is calculated as below.

Accuracy = (TP + TN) / (TP + TN + FP + FN).

The final output is obtained by performing the AND operation on the outputs from both the modules that are the liveness detection module and the classification module. If the output of the liveness module is true, that is the person in front of the web camera is alive, and the output of the CNN classifier is also true, that is the person is classified as real, then the final output (image of the person) is predicted as Real. If anyone of the module gives negative output, then the output (image of the person) is predicted as fake or a spoof.

Testing
We put a variety of face spoofing scenarios to the test, including printed images, digital photos, and digital videos from smartphones. Our model performs quite well in real-time scenarios.

4 Results and Discussion

Deep learning-based techniques integrated with various other cues for life-sign detection give a good outcome for resolving almost all spoof scenarios. The detected landmarks give the points that denote the different facial features which are the eye, brows, nose, lips, and jaw as shown in Fig. 3. The classification module is a neural network-based system that uses images as input to determine if a face is real or not and gives an accuracy of 98% after training on 60 epochs. Photographs, phone photos, video frames, and masks are all examples of fake images. The eye movement liveness module gives accuracy of 97%. The CNN classification module gives a good accuracy of 98%. We have trained our pre-trained neural network model using the LCC_FASD dataset that contains a total of 18827 images, out of which 1942 are real images and 16885 are spoof images. We have used the transfer learning technique for our model to adapt as per the new dataset and customized it to provide binary classification as output. The precision, recall, and f1 score are shown in Table 2 below. The training accuracy, loss and validation accuracy, loss is as depicted in Figs. 6 and 7 respectively. The real-time results of our model are shown in Fig. 8.

The development environment settings used during this research are Windows 11, Intel Core i7-7500U (10th Gen), 6 GB NVIDIA graphics card, and 24 GB RAM. We used Python as the programming language and Keras-OpenCV, Tensorflow framework.

5 Conclusion and Future Work

The aim of this project is to improve the face recognition systems by improving their reliability and protecting it against various types of spoofing attacks. The proposed model performs well on the real and fake discrimination objectives, allowing it to be

Table 2. Precision, Recall, f1-score

	precision	recall	f1-score
spoof	0.94	0.98	0.96
real	0.99	0.98	0.99
accuracy			0.98
Macro avg	0.96	0.98	0.97
weighted avg	0.98	0.98	0.98

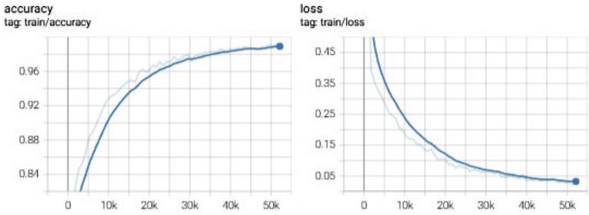

Fig. 6. Training accuracy and loss

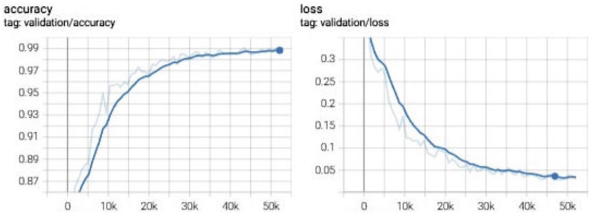

Fig. 7. Validation accuracy and loss

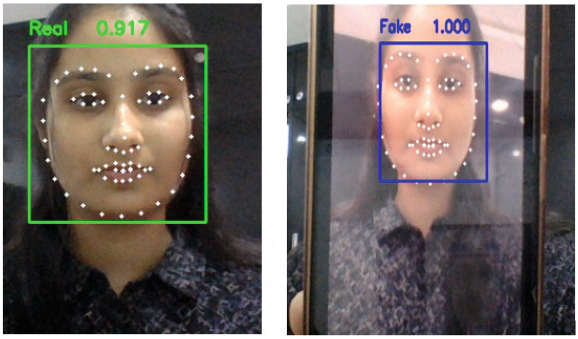

Fig. 8. Examples of real-time results

employed effectively in a variety of applications at low cost and with great accuracy without requiring any new hardware. We suggest in this paper that CNN combined with liveness detection performs remarkably well in real-time scenarios for face anti-spoofing. Various attacks have been tried and tested. The system can efficiently handle printed photos, digital photos, and video replay attacks. We have also discussed various types of spoofing attacks and various facial anti-spoofing approaches that have been proposed in different studies so far. The goal is to give a simple roadmap for the design of more dependable, user-friendly, and effective systems.

When it comes to real-time applications, insufficient light has an impact on classification accuracy. Biases among datasets are inescapable due to varying capture conditions. The future scope for this project could be to develop a mechanism to adapt the learned model to various lighting conditions, and shadowing effects. Another approach is to incorporate deep learning with other cues such as textures, movements, and shapes.

Also, there is a need for an easily available large-size dataset that contains all types of spoofing scenarios and various lighting conditions for further research.

Acknowledgments. I would like to thank Vrishali Chakkarwar ma'am and Dr. Vivek Kshirsagar for their guidance and support. I would like to thank the Department of Computer Science and Information technology, Government college of Engineering, Aurangabad, Maharashtra for providing me with research support. Lastly, I would like to thank Dr. Babasaheb Ambedkar Marathwada University, Aurangabad, Maharashtra, India for providing the publication support.

References

1. R. B. Hadiprakoso, H. Setiawan and Girinoto, "Face Anti-Spoofing Using CNN Classifier & Face liveness Detection," 2020 3rd International Conference on Information and Communications Technology (ICOIACT), 2020, pp. 143–147, https://doi.org/10.1109/ICOIACT50329.2020.9331977.
2. P. Anthony, B. Ay and G. Aydin, "A Review of Face Anti-spoofing Methods for Face Recognition Systems," 2021 International Conference on INnovations in Intelligent SysTems and Applications (INISTA), 2021, pp. 1-9.
3. Wen C, Chen F M, Xie K, et al. Face liveness detection: fusing color texture feature and deep feature[J]. IET Biometrics, 2019, 8(6):369-377.
4. Boulkenafet Z, Komulainen J, Hadid A. face anti-spoofing based on color texture analysis[C]//Image Processing (ICIP), 2015 IEEE International Conference on. IEEE, 2015: 2636-2640.
5. J. Komulainen and M. Pietikainen, "Face spoofing detection from single images using texture and local shape analysis," International Conference on Computing, Communication and Automation (ICCCA), vol. 1, pp. 3– 10, 2012.
6. Singh, P. Joshi, and G. Nandi, "Face recognition with liveness detection using eye and mouth movement," 07 2014.
7. J. Zhou, C. Ge, J. Yang, H. Yao, X. Qiao, and P. Deng, "Research and application of face anti-spoofing based on depth camera," in 2019 2nd China Symposium on Cognitive Computing and Hybrid Intelligence (CCHI), Sep. 2019, pp. 225–229.
8. Z. Xu, S. Li, and W. Deng, "Learning temporal features using LSTM-CNN architecture for face anti-spoofing," in 2015 3rd IAPR Asian Conference on Pattern Recognition (ACPR), Kuala Lumpur, Malaysia, 3- 6 November 2015, pp. 141–145.
9. X. Tu, H. Zhang, M. Zie, Y. Luo, Y. Zhang, and Z. Ma, "Enhance the motion cues for face anti-spoofing using CNN-LSTM Architecture," retrieved from https://arxiv.org/pdf/1901.05635.pdf (accessed on 07/16/2019).
10. Kazemi, Vahid and Josephine Sullivan. "One millisecond face alignment with an ensemble of regression trees." 2014 IEEE Conference on Computer Vision and Pattern Recognition (2014): 1867–1874.
11. D. Timoshenko, K. Simonchik, V. Shutov, P. Zhelezneva and V. Grishkin, "Large Crowdcollected Facial Anti-Spoofing Dataset," 2019 Computer Science and Information Technologies (CSIT), 2019, pp. 123-126, https://doi.org/10.1109/CSITechnol.2019.8895208.

12. Ghofrani, Ali, Rahil Mahdian Toroghi, and Seyed Mojtaba Tabatabaie. “Attention-Based Face AntiSpoofing of RGB Images, using a Minimal End-2-End Neural Network.“ arXiv preprint arXiv:1912.08870 (2019).
13. M. Sandler, A. Howard, M. Zhu, A. Zhmoginov and L. -C. Chen, “MobileNetV2: Inverted Residuals and Linear Bottlenecks,” 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2018, pp. 4510-4520, https://doi.org/10.1109/CVPR.2018.00474.