International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 06 Issue: 05 | May - 2022 Impact Factor: 7.185 ISSN: 2582-3930

LIVENESS DETECTION WITH OPENCV

Aditya Richhariya, Amit Kumar Sachan, Aditi Gupta

Department of Computer Science and Engineering
B.Tech, Shri Ramswaroop Memorial College of Engineering and Management, Lucknow, Uttar Pradesh

---

Abstract - Biometric systems are commonly used to identify authorised individuals based on behavioural or physical features. However, various traits can be used to fake this. Spoofing is defined as the use of security features to attack or harm a biometric recognition system in order to use it without the authorization of an authorised user. Images or video of a person can readily be found on social media or shot from a safe distance. Consider what would happen if a malicious person tried to evade your face recognition system on purpose. A user like this could try to hold up a photo of someone else. Perhaps they have a snapshot or video on their smartphone that they might show to the camera that performs face recognition. We used OpenCV to detect "fake" versus "real/legitimate" faces in the project and to see how we could use anti-face spoofing techniques in our facial recognition applications by using liveness detection.

I. INTRODUCTION

We need to be able to detect fake/non-real faces in order to make facial recognition systems safer – liveness detection is the word for such methods.

There are a number of approaches to liveness detection, including:

- Texture analysis, which includes computing Local Binary Patterns (LBPs) over face regions and classifying the faces as real or faked using a support vector machine (SVM) [1].

- Frequency analysis, such as looking at the face's Fourier domain.

- Examining the variation of pixel values across two successive frames is an example of variable focusing analysis [2].

- Eye movement, lip movement, and blink detection are all reliant on heuristics. These algorithms try to track the user's eye movement and blinks to make sure they aren't holding up a photo of someone else (since a photo will not blink or move its lips)

- Algorithms for evaluating the differences and features of optical flow created by 3D objects and 2D planes, also known as optical flow algorithms.

- 3D face form, similar to what Apple's iPhone facial recognition technology does, allowing the system to discriminate between actual faces and printouts/photos/images of other people.

II. LITERATURE SURVEY

Movement of eye base analysis –

Hyung-Keun Jee et al. developed an embedded face recognition system using a technique based on eye movement analysis [3], [4]. The authors suggested a method for recognising eyeballs in successive input photographs, then calculating the variance of each eye area and determining whether or not the input face is real [5]. The main premise is that there should be large shape fluctuations in human eyes due to blinking and uncontrolled pupil movements. In the input facial image, the first centre point of both eyes is recognised. Face areas are normalised and eye regions are retrieved using both eyes that have been detected. After binarizing extracted eye areas, the variation between binarized eye regions is calculated [6]. If the result exceeds the threshold, the input image is identified as a live face; otherwise, it is classified as a snapshot. The authors employed the fact that the intensity of the eye region is lower than the rest of the face region when the image is evaluated as a 3D curve to detect the eye regions [7].

Face Shape based analysis –

Andrea Lagorio et al. offer a unique liveness detection approach based on the 3D structure of the face. The proposed method allows a biometric system to distinguish between a real face and a photograph, hence minimising vulnerability [8]. The proposed approach, according to the authors, can be used in two scenarios: as an anti-spoofing tool in conjunction with 2D face recognition systems, or as part of a 3D face recognition

system for early detection of spoofing attempts. The suggested algorithm uses the 3D properties of the acquired face data to identify whether or not there is a live face in front of the camera. The lack of surface variation in the scan is one of the primary indicators that the acquisition is from a 2D source, according to the scientists. It has a very low curvature on the surface. A simple and fast approach is created to compare the two 3D scans based on the estimation of the surface's mean curvature. The primary components of the Cartesian coordinates within a specified neighbourhood are used to calculate an approximation of the actual curvature value at each place. After that, the mean curvature of the 3D points on the face surface is calculated. Two experiments were devised by the authors. They employed FS and GVS sets in the first one. The False Rejection Rate (FRR) was computed as zero after separating the distributions of the mean curvature values for the two sets [9], [10].

III. OBJECTIVES

The dissertation "Liveness Detection using OpenCV" discusses liveness detection, including what it is and why we need it to better our face recognition systems.

Following that, we'll go over the dataset we'll be utilising to detect liveness, which includes:

- How to create a dataset for detecting liveness
- Real vs. artificial face images are our example.

For the liveness detector project, we'll also go over our project structure.

We'll train a deep neural network capable of differentiating between real and fake faces to construct the liveness detector [11].

IV. PROBLEM STATEMENT

In this project, we'll look at how to distinguish between "fake/spoofed" and "real/legitimate" faces. How could anti-face spoofing algorithms be included into your facial recognition software?
The solution is to use OpenCV to identify liveness. Liveness detection will be approached as a binary classification problem.

V. EXISTING SYSTEM

Face recognition technology that is currently available entails-

- Image Capture

The first step is to get a picture of the user's face from the camera [12].

- Face Detection

Face detection is performed in the second step using the acquired image. It can also be improved or normalised before being processed further.

- Feature Extraction

The third phase is a face recognition process in which the desired facial features are identified [13], [14].

- Matching

These extracted features are compared to the database's stored features.

- Determine Identity

Finally, the output of the facial recognition algorithm is used to determine the person's identification (whether there is a match or not) [15].

VI. PROPOSED METHODOLOGY

The main methodology used in this project is to train the model using CNN so that it can differentiate between real and fake images/videos i.e., face liveness detection using OpenCV. We'll train a Convolutional Neural Network capable of distinguishing real faces from fake/spoofed faces. The model is divided into four modules-

- Dataset (Real faces, Fake/Spoofed Faces)

This module deals with the building of the dataset on the basis of which our model will be trained. It will contain all the types of photoshot, videos, picture of both the types i.e., original, fake/spoofed and our model will be trained on this data set.

- LivenessNet

This module deals with the formation of Convolutional Neural Network. It is the base module which will help us in identifying real and spoofed/fake faces. The convolutional neural network will consist of four layers. The first layer will be the input layer which will have the image of ROI. The next two layers will be the hidden layers each comprising a set of CONV=> RELU=> CONV=> RELU=> POOL where ROI is the region of interest, CONV are the filters to detect edges, RELU is the activation function and POOL is MaxPooling and the last layer will be the output layer with softmax as classifier/activation function.

- Training Script

This module deals with the training of Convolutional Neural Network. In this module, the input will be the dataset which we had built in our previous modules. This module will convert the given dataset into the required format through various data manipulation techniques (like label-encoder, image-data-generator, train-test-split, etc.) so that the dataset gets fit for training convolutional neural network. The output of this module will a trained

convolutional neural network model to identify real and spoofed faces.

- Output Model

The last module will be the output layer which consist of a single neuron with softmax as the classifier/activation function. Softmax is an activation function that scales numbers/logits into probabilities. The output of a Softmax is a vector (say v) with probabilities of each possible outcome. The probabilities in vector v sums to one for all possible outcomes or classes.

VII. BLOCK DIAGRAM

VIII. IMPLEMENTATION

We will describe the details of the implementations in the work.

- We've used OpenCV to gain access of the system's web cam.
- For detecting faces/ROI, we've used a pre-trained model which detects faces that comes in-front on the web cam.
- We've created dataset with the help of two videos.
  - First video consists of me walking around in my home. (For real dataset)
  - Second video consists of the same video, this time facing my phone towards my desktop where I recorded the video replaying. (For fake/spoofed dataset)
- Deep Learning's CNN (Convolutional Neural Network) model is trained over the image's dataset of fake and real images.
- Flask is used to create a web application framework that enable us to write this python application without worrying about low-level details such as protocol, thread management, and so on.

IX. RESULT

The accuracy of the proposed system has been evaluated using Convolutional Neural Network (CNN) algorithm. The performance is shown in Table I.

TABLE I: Performance analysis

Algorithm	Accuracy (%)
CNN	95.61%

X. CONCLUSION AND FUTURE SCOPE

In most of the authentication system, face plays an important role. In face recognition systems, the face features of these system are extracted to develop a system. But this system may recognize the person from photos. This is a system failure. Hence to improve the Face recognition system, face liveness detection plays a vital role. In this method frame is extracted from the input stream. After that, image is converted into gray scale, to detect frontal face. Now different features are collected from extracted face image. Now the extracted face goes through a deep neural network model to check whether the extracted face is real or spoofed. The proposed system is implemented using OpenCV. The results of the detection can be improved by the following ways:

- The dataset collected can be increased by taking more and more varieties of faces from all around the world.
- The dataset was created with the help of screen/monitor, can upgrade the dataset with photos/images.
- The front-end can be made in a more appropriate way.

XI. REFERENCES

[1] Boneh, D., and Boyen, X. Short signatures without random oracles and the sdh assumption in bilinear groups. Journal of Cryptology 21, 2 (2008), 149–177.

[2] Boyen, X., and Waters, B. Full-domain subgroup hiding and constant-size group signatures. In Lecture Notes in Computer Science, PKC 2007 (2007), pp. 1–15.

[3] Chai, Z., Cao, Z., and Lu, R. Efficient password-based authentication and key exchange scheme preserving user privacy. In Lecture Notes in Computer Science, Wireless Algorithms, Systems, and Applications (2006), vol. 4138, pp. 467–477.

[4] Erdogmus, H. Cloud computing: Does nirvana hide behind the nebula? IEEE Software 11, 2 (MarchApril 2009), 4–6.

[5] Foster, I., Zhao, Y., Raicu, I., and Lu, S. Cloud computing and grid computing 360-degree compared. In Proceedings of Grid Computing Environments Workshop, GCE'08 (Austin, TX, 2008), pp. 1–10.

[6] Gellman, R. Privacy in the clouds: Risks to privacy and confidentiality from cloud computing. Tech. rep., http://www.worldprivacyforum.org.

[7] Goldwasser, S., Micali, S., and Rivest, R. A digital signature scheme secure against adaptive chosenmessage attacks. SIAM Journal of Computing 17, 2 (1988), 281–308.

[8] Hartig, K. What is cloud computing? website, April 2009. http://kevinhartig.sys-con.com/.

[9] Hasan, R., Sion, R., and Winslett, M. Introducing secure provenance: problems and challenges. In Proceedings of ACM workshop onStorage security and survivabilit, StorageSS'07 (Alexandria, Virginia, USA, October 2007),pp. 13–18.

[10] Kaufman, L. M. Data security in the world of cloud computing. IEEE Security & Privacy 7, 4 (JulyAug.2009), 61–64.

[11] Liang, X., Cao, Z., Shao, J., and Lin, H. Short group signature without random oracles. In Lecture Notes in Computer Science, ICICS 2007 (2007), pp. 69–82.
[12] Lin, X., Sun, X., Ho, P.-H., and Shen, X. GSIS: a secure and privacy-preserving protocol for vehicular communication. IEEE Transactions on Vehicular Technology 56, 6 (2007), 3442–3456.
[13] Lu, R., Lin, X., and Shen, X. Spring: Asocial-based privacy-preserving packet forwarding protocol for vehicular delay tolerant networks. In The 29th IEEE International Conference on Computer Communications (INFOCOM 2010) (San Diego, California, USA, March 2010).
[14] Lu, R., Lin, X., Zhu, H., Ho, P.-H., and Shen, X.ECPP: Efficient conditional privacy preservation protocol for secure vehicular communications. In The 27th Conference on Computer Communications (INFOCOM 2008) (Phoenix, Arizona, USA, April 2008), pp. 1229–1237.
[15] Lynch, C. A. When documents deceive: Trust and provenance as new factors for information retrieval in a tangled web. Journal of the American Society for Information Sciene and Technology 52, 1 (2001), 12–17.