International Journal of Modern Agriculture, Volume 9, No. 4, 2020
ISSN: 2305-7246

LIVENESS DETECTION WITH OPEN CV

C.Anuradha[1], Deepak Kumar Singh[2]

Abstract

Biometric system is widely used to recognize the authorized person based on either behavioural characteristics or physical. But this can be spoofed using various traits. Spoofing attack is nothing but attacking or harming biometric recognition system using security features to use system without permission of authorized user. Images or video of person can be easily available from social media or can be easily captured from some distance. Consider what would happen if a nefarious user tried to purposely circumvent your face recognition system. Such a user could try to hold up a photo of another person. Maybe they even have a photo or video on their smartphone that they could hold up to the camera responsible for performing face recognition. In the project, we have tried spotting these "fake" versus "real/legitimate" faces and how we could apply anti-face spoofing algorithms into our facial recognition applications by applying liveness detection with OpenCV.

Introduction

In order to make face recognition systems more secure, we need to be able to detect such fake/non-real faces — liveness detection is the term used to refer to such algorithms.

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

- Texture analysis, including computing Local Binary Patterns (LBPs) over face regions and using an SVM to classify the faces as real or spoofed.
- Frequency analysis, such as examining the Fourier domain of the face.
- Variable focusing analysis, such as examining the variation of pixel values between two consecutive frames.
- Heuristic-based algorithms, including eye movement, lip movement, and blink detection. These set of algorithms attempt to track eye movement and blinks to ensure the user is not holding up a photo of another person (since a photo will not blink or move its lips).
- Optical Flow algorithms, namely examining the differences and properties of optical flow generated from 3D objects and 2D planes.
- 3D face shape, similar to what is used on Apple's iPhone face recognition system, enabling the face recognition system to distinguish between real faces and printouts/photos/images of another person.

Literature survey

Movement of eye based analysis

The technique based on the analysis of movement of eyes was introduced by Hyung-Keun Jee etal. for embedded face recognition system. The authors proposed a method for detecting eyes in sequential input images and then variation of each eye region is calculated and whether the input face is real or not is determined. The basic assumption is that because of blinking and uncontrolled movements of the pupils in human eyes, there should be big shape variations. First center point of both eyes is detected in the input face image. Using detected both eyes, face region are normalized and eye regions are extracted. After binarizing extracted eye regions, each binarized eye regions are compared and variation is calculated. If the result is bigger than threshold, the input image is recognized as live face, if not, it is discriminated to the photograph. For detection of the eye regions,

International Journal of Modern Agriculture, Volume 9, No. 4, 2020
ISSN: 2305-7246

the authors used the fact that the intensity of the eye region is lower than the rest of face region if the image is considered as a 3D curve.

Face Shape based analysis

The novel liveness detection method, based on 3D structure of the face is proposed by Andrea Lagorio et al. The proposed approach allows a biometric system to differentiate real face from a photo thus reducing the vulnerability. The authors suggested that the proposed approach can be implemented in different scenarios: either as an anti-spoofing tool, coupled with 2D face recognition systems; or can be integrated with a 3D face recognition system to perform an early detection of spoofing attacks. The proposed algorithm computes the 3D features of the captured face data to determine if there is a live face is presented in front of the camera or not. The authors show that the lack of surface variation in the scan is one of the key evidence that the acquisition comes from 2D source. It has a very low surface curvature. Based on the computation of the mean curvature of the surface, a simple and fast method is implemented to compare the two 3D scans. An approximation of the actual curvature value at each point is computed from the principal components of the Cartesian coordinates within a given neighbourhood. The mean curvature of the 3D points lying on the face surface is then computed. The authors designed two experiments. In the first one, they used FS and GVS sets. The distribution of the mean curvature values for the two sets was separated, and the value of the False Rejection Rate (FRR), was computed as zero.

Objectives

The aim of the dissertation " Liveness Detection using OpenCV" is to discuss liveness detection, including what it is and why we need it to improve our face recognition systems.

From there we'll review the dataset we'll be using to perform liveness detection, including:

- How to build to a dataset for liveness detection
- Our example real versus fake face images

We'll also review our project structure for the liveness detector project as well.

In order to create the liveness detector, we'll be training a deep neural network capable of distinguishing between real versus fake faces.

Problem statement

In this project we fill focus on How would we go about spotting these "fake" versus "real/legitimate" faces? How could we apply anti-face spoofing algorithms into your facial recognition applications? The answer is to apply liveness detection with OpenCV. We'll be treating liveness detection as a binary classification problem.

Existing system

The currently existing Face recognition technology generally involves

• Image capture

The first step is to acquire the facial image of user from the camera.

• Face detection

At the second step, face is detected from the acquired image. It can also be normalized or enhanced for further processing.

International Journal of Modern Agriculture, Volume 9, No. 4, 2020
ISSN: 2305-7246

• Feature Extraction

At the third step, face recognition process takes place in which the desired facial features are extracted.

• Matching

These extracted features are matched against the features stored in the database.

• Determine identity

Finally, the output of face recognition process is used (if there is a match or not) to determine the identity of the person.

disadvantages of existing system

The problem with current system is, A face recognition system is also prone to the spoofing attacks. Our biometric facial data can be easily stolen from social sites and other personal web sites. Most common attack on face recognition system is the photograph attack i.e. placing photographs in front of camera. Other facial spoofing attacks are playing video of genuine user in front of camera and using 3D dummy faces or mask.

Proposed system

In order to minimize such problems, Liveness detection is integrated within the system. Method of liveness detection detect physiological signs of life from face ensuring that only live face samples are stored for enrolment or authentication. For the PROJECT, we'll be treating liveness detection as a binary classification problem. Given an input image, we'll train a Convolutional Neural Network capable of distinguishing real faces from fake/spoofed faces.

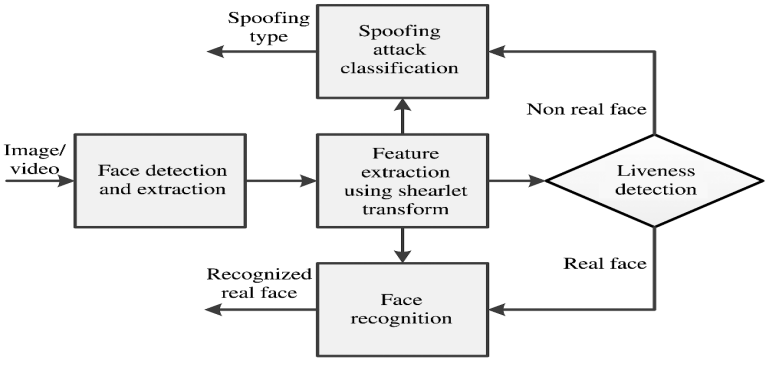
Architectural design specification

In order to build the liveness detection dataset, I:

- Took my Phone and put it in portrait/selfie mode.
- Recorded a ~25-second video of myself walking around my office.
- Replayed the same 25-second video, this time facing my iPhone towards my desktop where I recorded the video replaying.
- This resulted in two example videos, one for "real" faces and another for "fake/spoofed" faces.

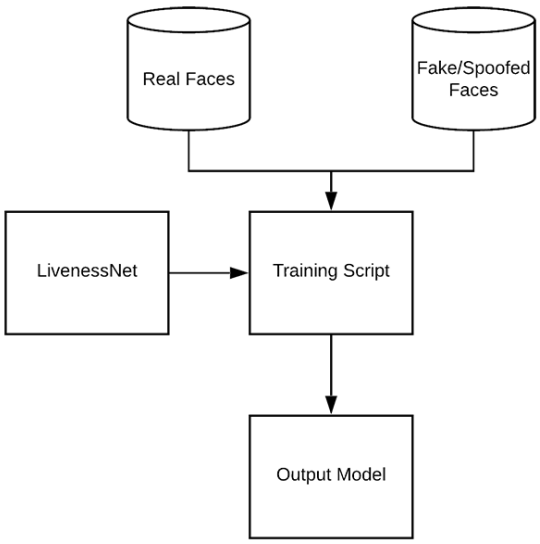
Finally, I applied face detection to both sets of videos to extract individual face ROIs for both classes.

Flowchart diagram

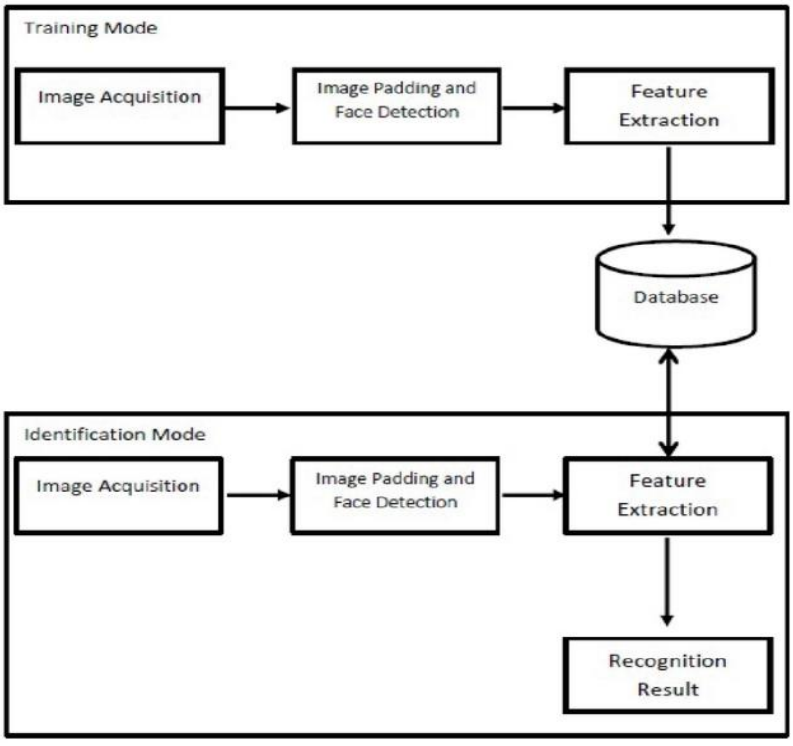


International Journal of Modern Agriculture, Volume 9, No. 4, 2020
ISSN: 2305-7246

Data flow diagram



Use case diagram



Algorithm specification

In gather_examples.py:

Lines 2-5 import our required packages. This script only requires OpenCV and NumPy in addition to built-in Python modules.

From there Lines 8-19 parse our command line arguments:

International Journal of Modern Agriculture, Volume 9, No. 4, 2020
ISSN: 2305-7246

--input : The path to our input video file.

--output : The path to the output directory where each of the cropped faces will be stored.

--detector : The path to the face detector. We'll be using OpenCV's deep learning face detector.

--confidence : The minimum probability to filter weak face detections. By default, this value is 50%.

--skip : We don't need to detect and store every image because adjacent frames will be similar. Instead, we'll skip N frames between detections. We can alter the default of 16 using this argument.

Now load the face detector and initialize our video stream.

Lines 23-26 load OpenCV's deep learning face detector.

From there we open our video stream on Line 30.

We also initialize two variables for the number of frames read as well as the number of frames saved while our loop executes (Lines 31 and 32).

Now create a loop to process the frames:

Our while loop begins on Lines 35.

Conclusion

Using this liveness detector we can now spot fake fakes and perform anti-face spoofing in your own face recognition systems.

To create our liveness detector we utilized OpenCV, Deep Learning, and Python.

The first step was to gather our real vs. fake dataset. To accomplish this task, for this :

1>First recorded a video of ourselves using our smartphone (i.e., "real" faces).

2>Held the smartphone up to our laptop/desktop, replayed the same video, and then recorded the replaying using our webcam (i.e., "fake" faces).

3>Applied face detection to both sets of videos to form our final liveness detection dataset.

After building our dataset we implemented, "LivenessNet", a Keras + Deep Learning CNN.

This network is purposely shallow, ensuring that:

1>We reduce the chances of overfitting on our small dataset.

2>The model itself is capable of running in real-time (including on the Raspberry Pi).

Overall, our liveness detector was able to obtain 99% accuracy on our validation set.

To demonstrate the full liveness detection pipeline in action a Python + OpenCV was created, script that loaded our liveness detector and applied it to real-time video streams.

As demo showed, the liveness detector was capable of distinguishing between real and fake faces.

International Journal of Modern Agriculture, Volume 9, No. 4, 2020
ISSN: 2305-7246

Reference

1. Boneh, D., and Boyen, X. Short signatures without random oracles and

2. the sdh assumption in bilinear groups. Journal of Cryptology 21, 2 (2008), 149–177.

3. 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.

4. 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.

5. Erdogmus, H. Cloud computing: Does nirvana hide behind the nebula? IEEE Software 11, 2 (March-April 2009), 4–6.

6. 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.

7. Gellman, R. Privacy in the clouds: Risks to privacy and confidentiality from cloud computing. Tech. rep., February 2009. http://www.worldprivacyforum.org.

8. Goldwasser, S., Micali, S., and Rivest, R. A digital signature scheme secure against adaptive chosen-message attacks. SIAM Journal of Computing 17, 2 (1988), 281–308.

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

10. Hasan, R., Sion, R., and Winslett, M.

11. 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.

12. Kaufman, L. M. Data security in the world of cloud computing. IEEE Security & Privacy 7, 4 (July-Aug.2009), 61–64.

13. 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.

14. 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.

15. 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).

16. 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.

17. Lu, R., Lin, X., Zhu, H., and Shen, X. Spark: a new vanet-based smart parking scheme for large parking lots. In The 28th Conference on Computer Communications (INFOCOM 2009) (Rio de Janeiro, Brazil, April 2009).Lynch, C. A. When documents deceive: Trust and provenance as new factors for

18. information retrieval in a tangled web. Journal of the American Society for Information Sciene and Technology 52, 1 (2001), 12–17.

International Journal of Modern Agriculture, Volume 9, No. 4, 2020
ISSN: 2305-7246

19. Mei, L., Chan, W., and Tse, T. A tale of clouds:Paradigm comparisons and some thoughts on research issues. In Proceedings of Asia-Pacific Services Computing Conference, APSCC'08 (Yilan, Dec.2008), pp. 464–469.

20. Shoup, V. Oaep reconsidered. Journal of Cryptology 15, 4 (2002), 223–249.