BE EXTC – Project Evaluation

Automatic Attendance Management System using Face Detection and Spoof Detection

by

Sheen Liz Shaju : 301546
Pratik V. Shetty : 301548
Akhil R. Singh : 301550

External Project Under the Guidance of
External Supervisor : Mr. Hemant Meena (Senior Software Engineer, Zing HR)
Internal Supervisor : Ms. Smita S. Hande (Assistant Professor, EXTC, FCRIT)

Department of Electronics and Telecommunication Engineering
Fr. C. Rodrigues Institute of Technology, Vashi

2018-2019

Abstract

- To design an attendance management system for the on-field employees., where an image (a face) and the current location of the employee will be sent from a mobile application for updating the attendance.
- The system detects spoofs when recognizing the face, to mark the attendance.
- Deep metric learning and Support Vector Machine (SVM) classification algorithm are used to detect and recognize a face. A proposed method called average brightness method is used to differentiate between a genuine face and a spoofed image of a face.

Presentation Outline

1. **Introduction**
2. **Literature Review**
3. **Implementation**
4. **Results and Discussion**
5. **Summary and Conclusions**

1. Introduction

1.1 Project Overview

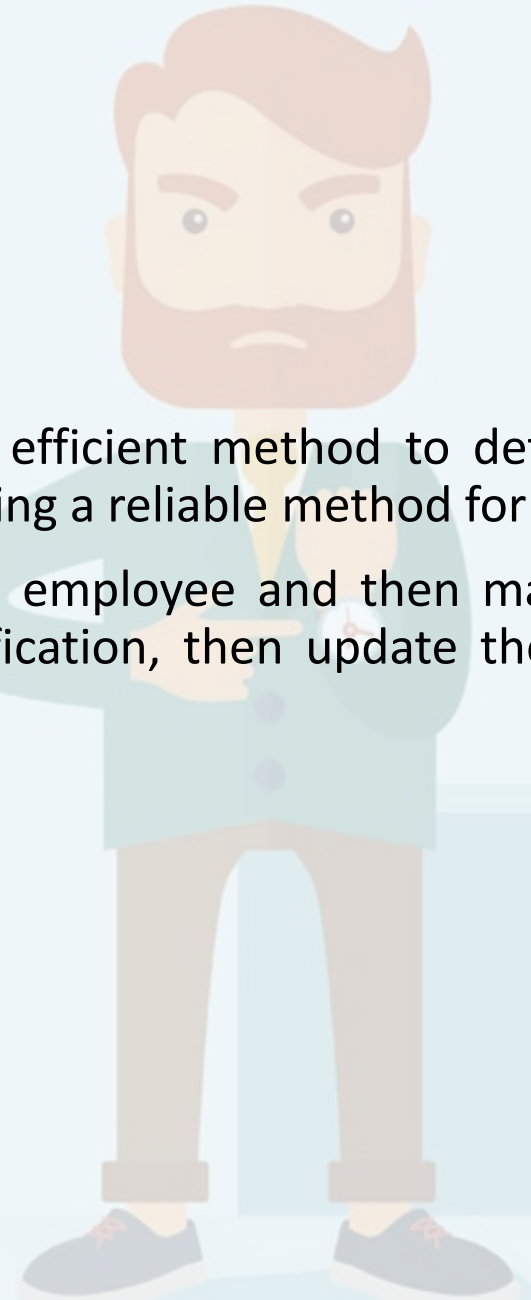
1.2 Project Aim

1.1 Project Overview

- The attendance management system uses the face recognition approach for the automatic attendance of employees in the office without a third person's intervention.

- This attendance is recorded by using a camera that captures image of the employee, detects the face in the image (detection), compares the detected face with the database (recognition), checks for the liveliness of the image (spoof detection) and then verifies the location and time at which the image is taken and accordingly marks the attendance onto the web portal.

1.2 Project Aim



- To implement an easy and efficient method to detect a face in an image and authenticate the same by using a reliable method for spoof detection.
- To recognize the face of an employee and then mark his/her attendance using Geo-location and time verification, then update the database by using android application.

2. Literature Review

2.1 Existing methods

2.2 Problem statement

2.1 Existing methods

- Pre-trained face detection classifiers:
  1. Haar (Cascade) Classifier [1]
  2. Local Binary Patterns (LBP) Classifier [2].
- Built-in face recognizers [3]:
  1. EigenFaces
  2. FisherFaces
  3. Local Binary Patterns Histogram (LBPH).
- Various existing method for spoof detection [4]:
  1. Depth analysis
  2. Texture analysis
  3. Image distortion analysis.

2.2 Problem statement

- Problems faced in the existing methods
- What is Spoof Attack?
- The methods used in the designed attendance management system
  1. Face detection: **HOG** (Histogram of Oriented Gradients) **face detector** [5].
  2. Face recognition: **dlib** and **face_recognition** tool [6].
  3. Spoof detection: a developed method called **average brightness method.**

3. Designing of the system

3.1 Face recognition

3.2 Spoof detection

3.1 Face recognition

1. Finding all the faces

- Histogram of Oriented Gradients (HOG)
- Face Detection

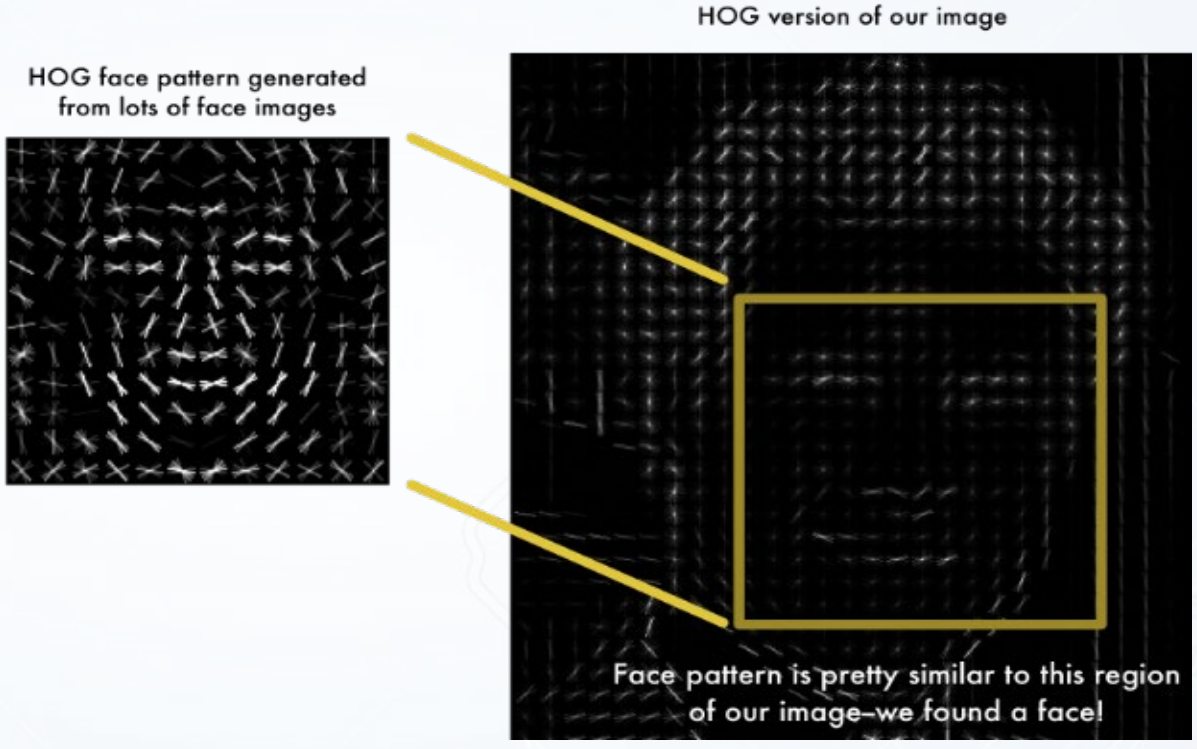


Fig. 1: HOG representation

3.1 Face recognition

2. Posing and projecting faces

- Face Landmark Estimation
- 68 points : landmarks

Fig. 2:The 68 landmarks marked on the face

3.1 Face recognition

3. Encoding Faces

- Deep Convolutional Neural Network
- 128 measurements: Embeddings
- Pre-trained network by OpenFace

4. Creating and predicting labels

- Support Vector Machine (SVM) Classifier
- Linear Support Vector Classifier
- Recognize the face, i.e. predict the label

3.2 Spoof Detection

Average Brightness Algorithm

- Methods developed for spoof detection

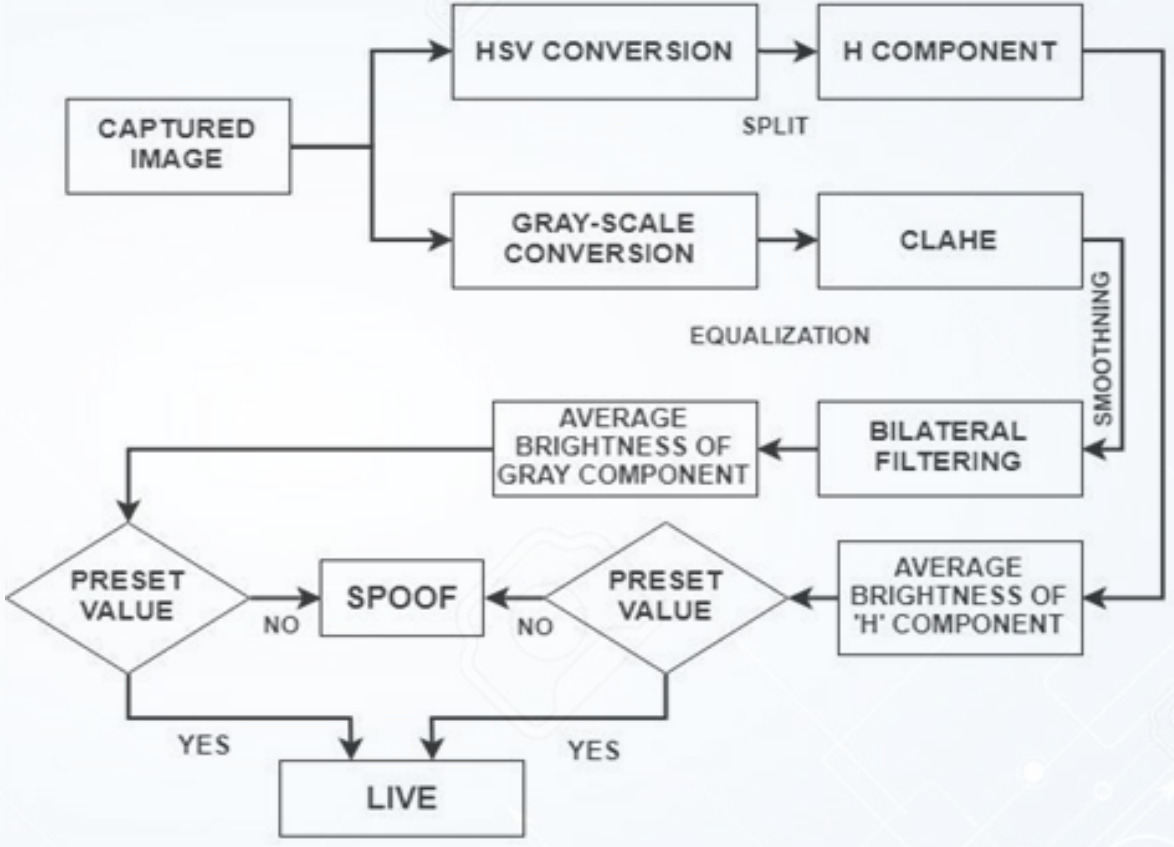


Fig. 3: Average brightness algorithm

4. Implementation

4.1 Volley and Flask framework

4.2 The developed algorithm

4.1 Volley and Flask framework

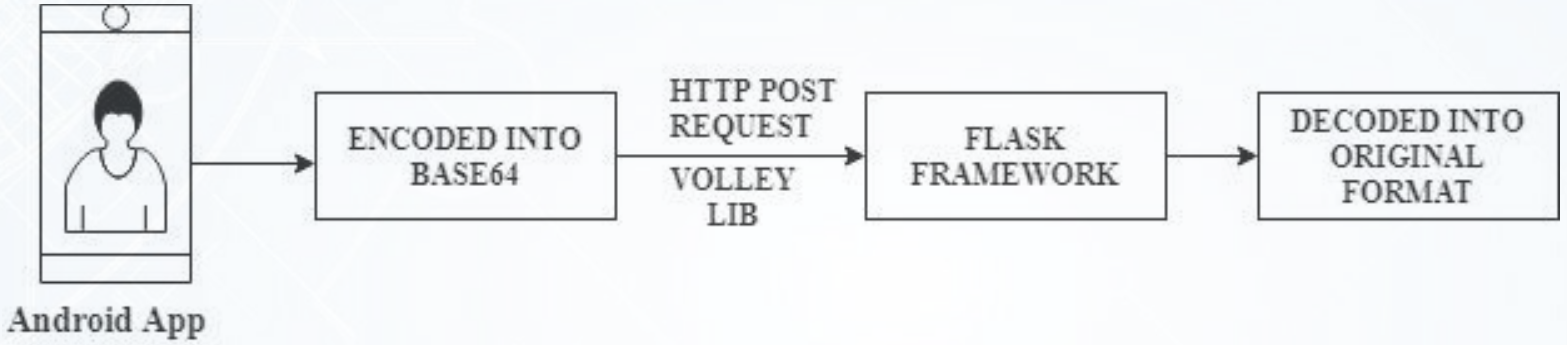


Fig. 5: Volley and HTTP request

- Uploading the face image to the flask micro web framework
- Base64 Format
- HTTP post requests

4.2 The developed algorithm

The basic pipeline for recognizing faces might work in the following way:

1. Find the face in the image.
2. Analyze facial features.
3. Compare against known faces.
4. Finally make a prediction.

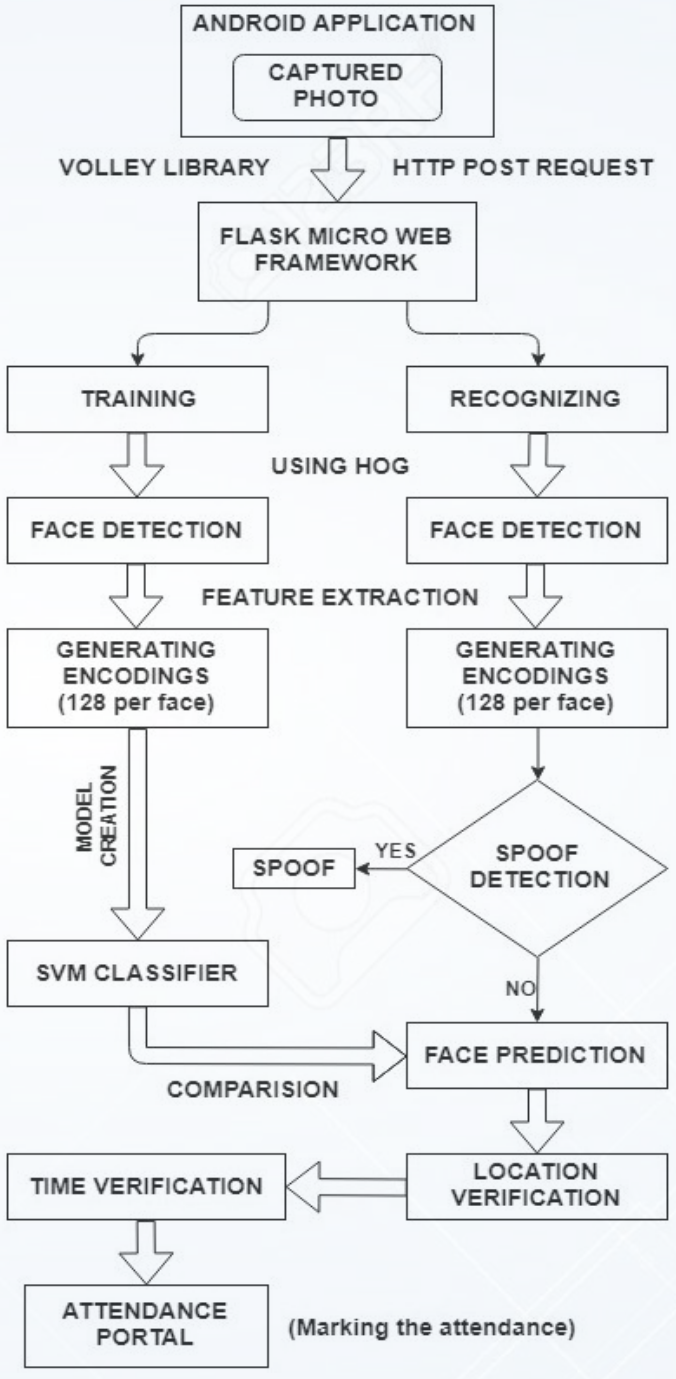


Fig. 4.:Block diagram of implemented algorithm

5. Result and Discussion

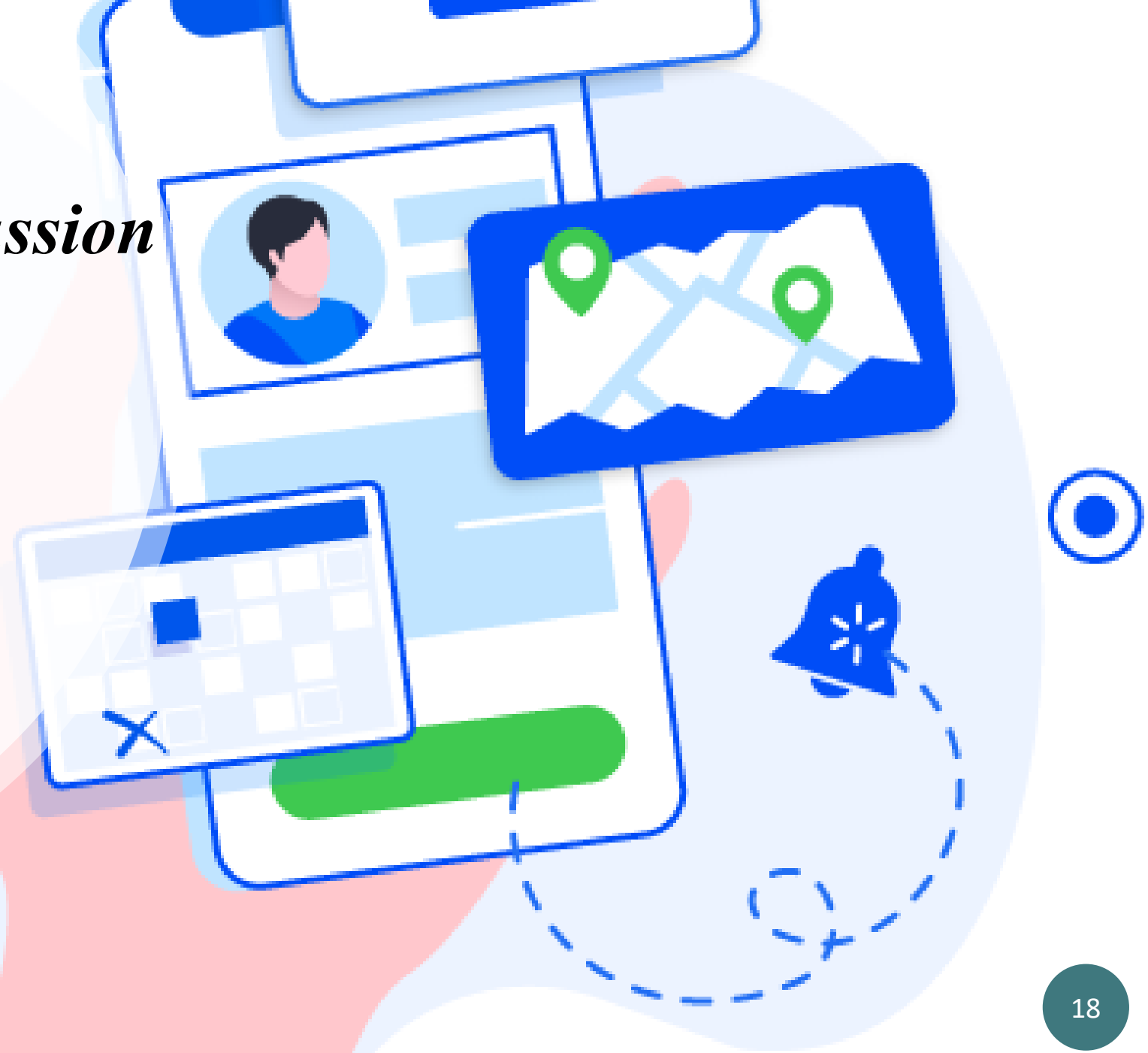
5.1 Face detection

5.2 Spoof detection

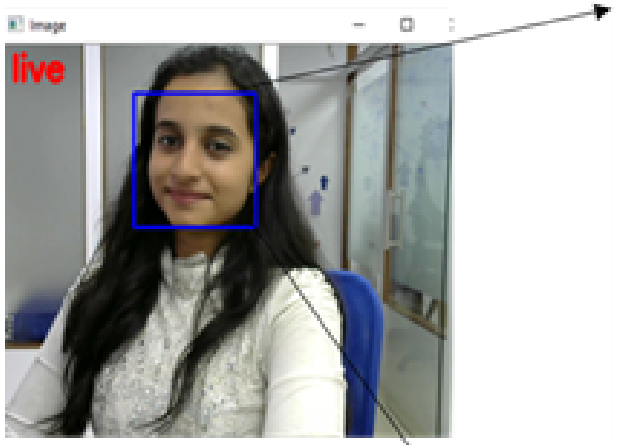
5.3 Face recognition

5.4 Android Application

5.5 Web Portal



5.1 Face Detection



-0.0997	0.018775	-0.09906	-0.07178	-0.05096	-0.06165
-0.07863	0.213641	-0.17732	0.119471	0.002605	0.014894
-0.09862	-0.02699	0.04281	0.14589	-0.1112	0.032075
-0.17849	-0.07635	-0.08327	-0.06056	0.040157	-0.01768
-0.01515	0.07004	-0.10793	-0.34908	-0.12126	-0.06192
-0.08356	0.027062	-0.00698	0.00971	0.122667	-0.01548
-0.17153	0.005036	0.066159	0.110263	-0.0537	-0.04248
-0.04466	0.233259	0.025771	-0.22544	-0.00017	0.13638
0.091873	0.242471	0.179985	0.037601	0.017147	0.253083
-0.00853	0.17208	-0.20858	0.081972	0.13375	0.01242
0.090352	0.036239	0.100655	-0.08411	0.043912	-0.12513
0.076792	-0.21565	0.028762	0.031189	-0.11369	0.141159
-0.06716	-0.05419	0.226776	0.159986	-0.08472	0.053369
-0.17221	0.238959	-0.20261	-0.06313	0.05852	-0.00902
-0.12922	-0.11997	-0.27372	-0.00604	0.397228	-0.01558
0.127833	-0.06961	0.112398	-1.99E-05	-0.07769	-0.22811
0.09633	-0.02636	0.096545	-0.02337	0.003356	0.214554
0.128434	-0.21768	0.16673	-0.04944	0.117631	-0.0674
-0.22784	0.017434	0.05432	0.000403	-0.05473	0.112582
0.085799	-0.12729	0.032315	-0.01168	-0.02115	0.001397
0.088353	-0.01918	0.028175	-0.00779	-0.03835	-0.02761

Fig. 6: Face detection and the generated encodings of a test image

5.2 Spoof Detection

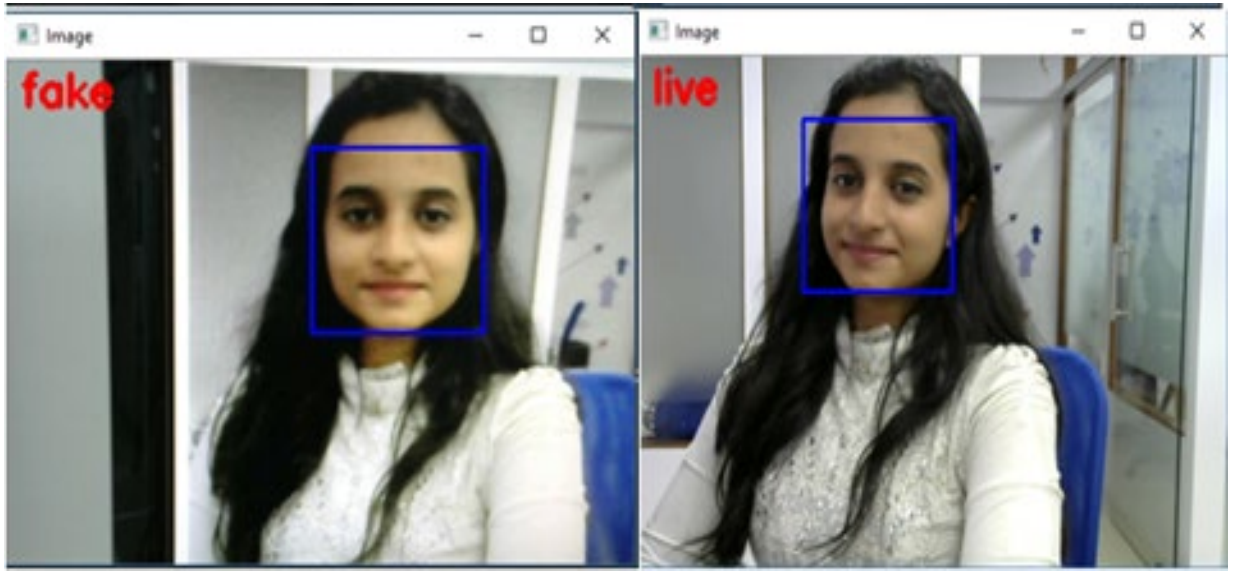


Fig. 7: Spoof detection on a spoofed image and a genuine image

5.3 Face Recognition

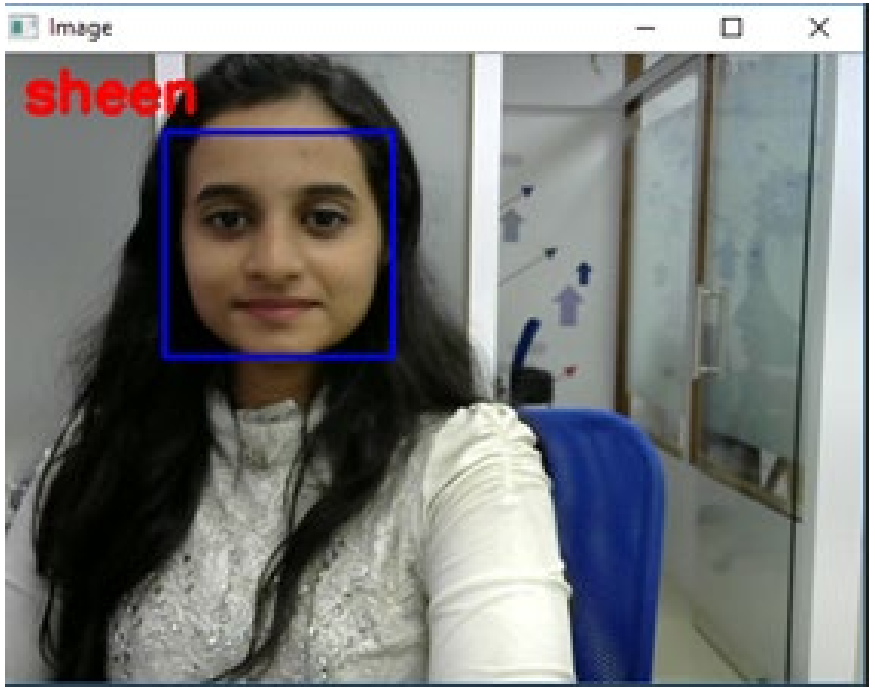


Fig. 8: Face recognition (prediction)

5.3 Android Application

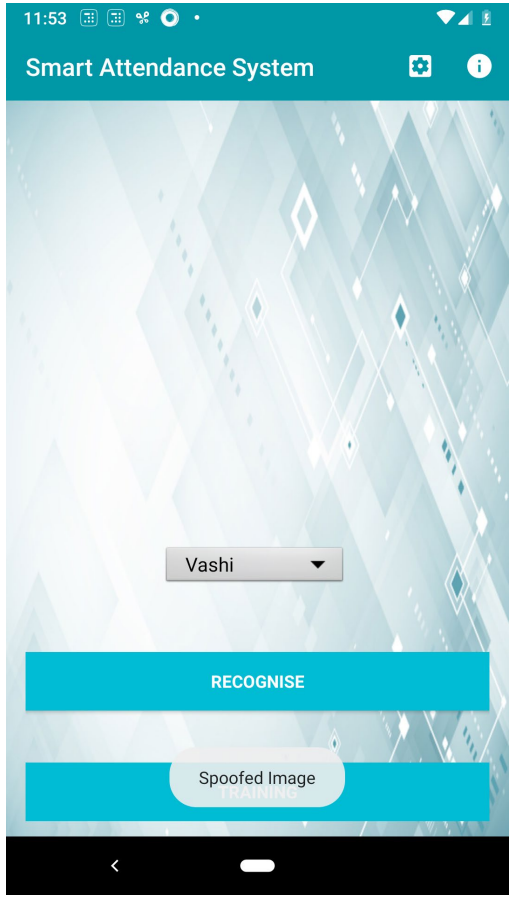


Fig. 9:Screenshot showing a toast message when a spoof is detected

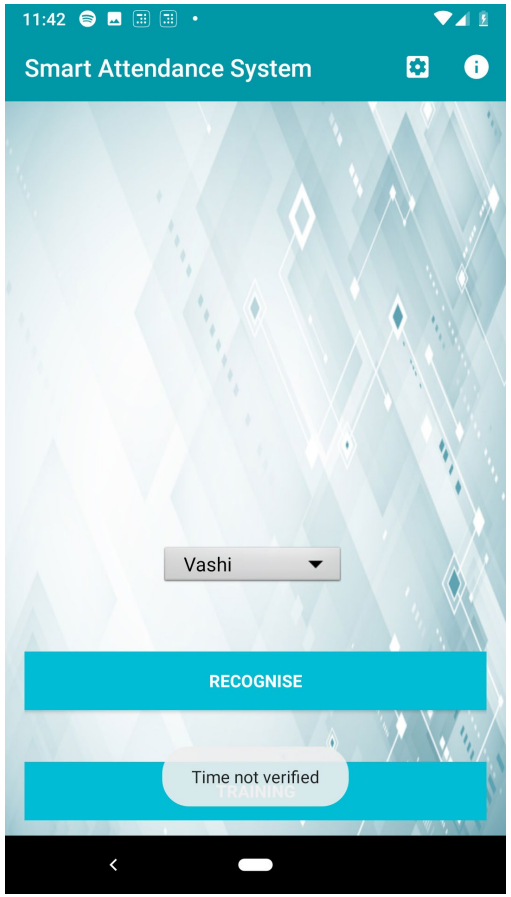


Fig. 10: Screenshot showing the toast message when time is not verified

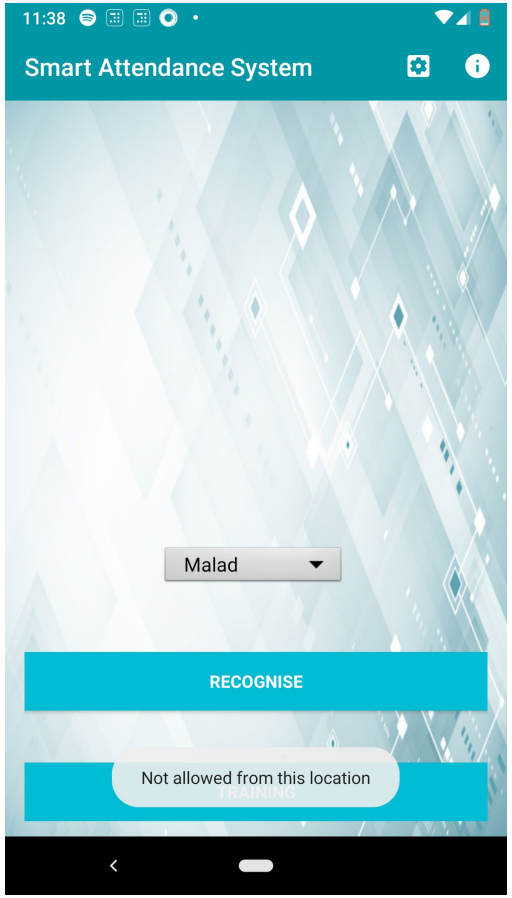


Fig. 11: Screenshot showing the toast message when location is not verified

5.3 Android Application

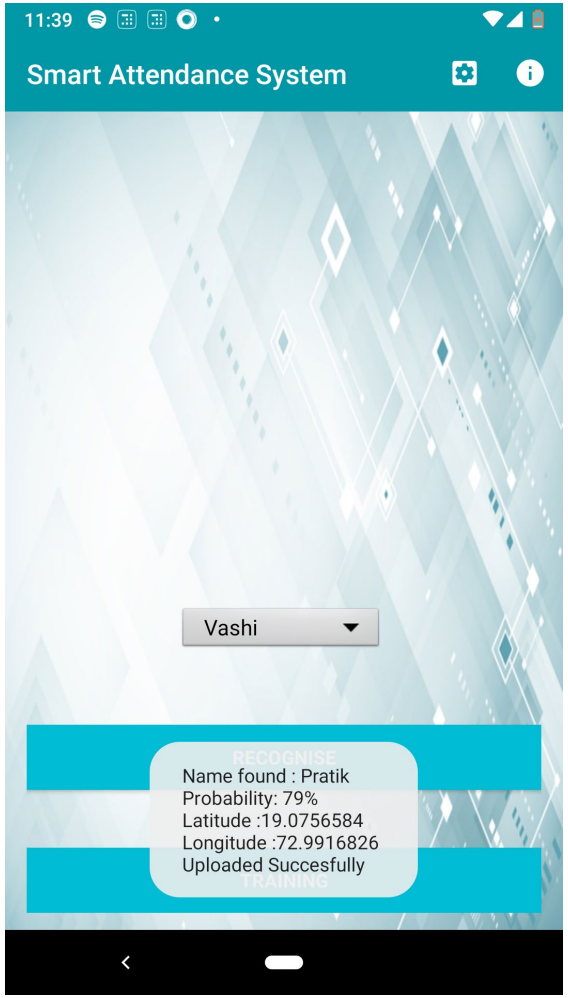


Fig. 12: Screenshot of application showing the toast message after successful authentication and marking the attendance

5.4 Web Portal

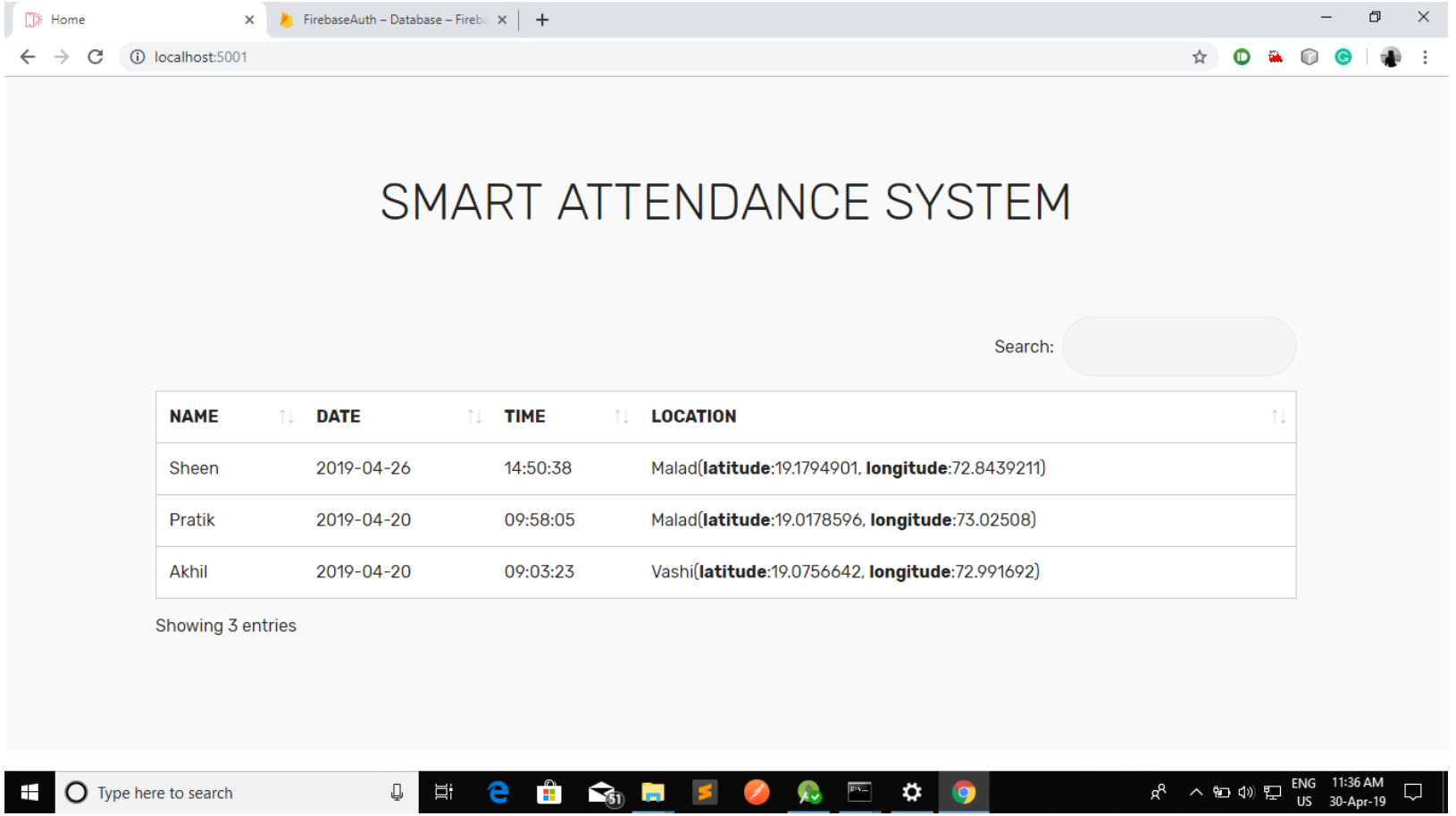


Fig. 13: Screenshot of the web portal after making 3 entries

5.4 Web Portal

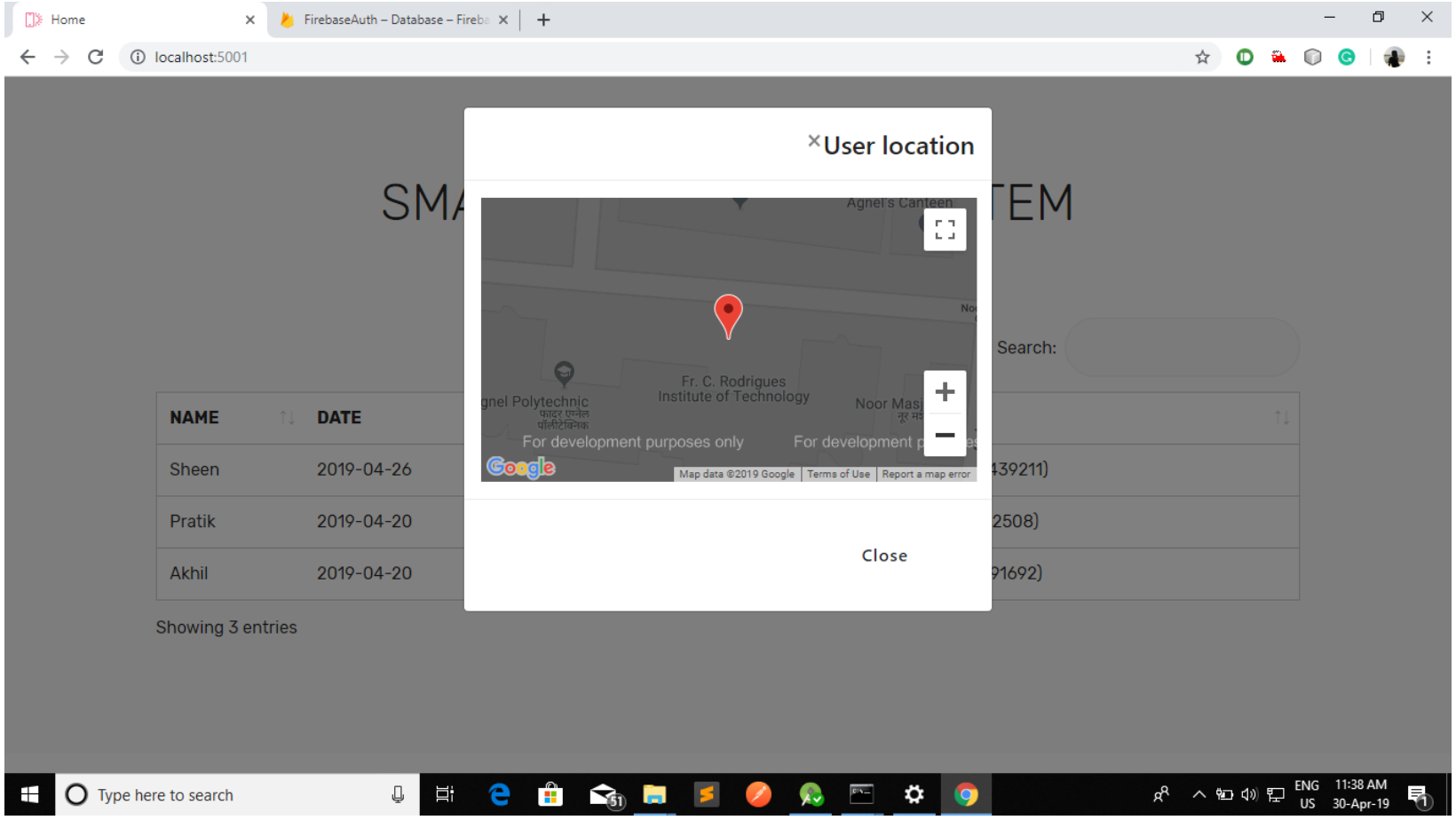


Fig. 14: Screenshot of the web portal showing the map of the location from where the attendance was marked

6. Conclusion

6.1 Conclusion

6.2 Future Scope

6.1 Conclusion

- Face detection is achieved using HOG algorithm.
- The developed method for spoof detection is based on Average Brightness level, which have certain limitations.
- Face recognition is achieved by using dlib face recognition library of OpenCV, which is accurate up to 99.38% of the time.
- The attendance management system is reliable and works in real time.

6.2 Future Scope

Implement indoor localization to improve the area restriction.

Improve spoof detection by making the threshold dynamic and also by using depth analysis.

With further improvement and modification, the system can be used in:

1. In medical fields
2. For law enforcement and security

CO-PO Mapping Table

	Program Outcomes												
Course Outcomes	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	Attainment Level
CO-1	3	-	-	-	-	-	-	-	-	-	-	-	3
CO-2	-	3	-	-	-	-	-	-	-	-	-	-	3
CO-3	1	-	3	-	-	-	-	-	-	-	-	-	4
CO-4	-	-	-	2	-	-	-	-	-	-	-	-	2
CO-5	-	2	-	-	3	-	-	-	-	-	-	-	5
CO-6	-	-	-	-	-	2	-	-	-	-	-	-	2
CO-7	-	-	-	-	-	-	2	-	-	-	-	-	2
CO-8	-	-	-	-	-	-	-	3	-	-	-	-	3
CO-9	-	-	-	-	-	-	-	-	3	-	-	-	3
CO-10	-	-	-	-	-	-	-	-	-	3	-	-	3
CO-11	-	-	-	-	-	-	-	-	-	-	1	-	1
CO-12	-	-	-	-	-	-	-	-	-	-	-	3	3

Reference

[1] S. Emami and V. P. Suciu, “Facial Recognition using OpenCV” *Journal of Mobile, Embedded and Distributed Systems*, vol. 4, no. 1, pp. 38-43, March 2012. [Online]. Available: http://www.jmeds.eu/index.php/jmeds/article/download/Facial _Recognition_using_OpenCV. [Accessed March 11, 2019].

[2] Laura Sánchez López. *Local Binary Patterns applied to Face Detection and Recognition.* Final research project. Universitat politècnica de catalunya; 2010.

[3] L. Dinalankara, “Face Detection & Face Recognition Using Open Computer Vision Classifies,” *ResearchGate*, 2017. [Online]. Available: https://www.researchgate.net/publication/318900718_Face_De tection_Face_Recognition_Using_Open_Computer_Vision_Cl assifies. [Accessed: 14- Mar- 2019].

[4] Zinelabidine Boulkenafet, Jukka Komulainen, and Abdenour Hadid,” FACE ANTI- SPOOFING BASED ON COLOR TEXTURE ANALYSIS,” in IEEE International Conference on Image Processing (ICIP), Quebec City, CAN, September 27-30, 2015.

[5] V. Gupta, “Home,” *Learn OpenCV*, 22-Oct-2018. [Online]. Available: https://www.learnopencv.com/face-detection-opencv-dlib-and- deep-learning-c-python/. [Accessed: 09-Feb-2019].

[6] *dlib C Library*. [Online]. http://www.dlib.net/face_detector.py.html. 09-Feb-2019].


Thank you