www.ijecs.in
International Journal Of Engineering And Computer Science ISSN:2319-7242
Volume 7 Issue 4 April 2018, Page No. 23820-23822
Index Copernicus Value (2015): 58.10, 76.25 (2016) DOI: 10.18535/ijecs/v7i4.04

Facial Recognition an Added Security Level To Mobiles And Laptops

K Sai Kiran[1], Divya M [2], B K Sree Chandana [3], Deepa R [4], Nagaveni B. Biradar [5]

[1, 2, 3, 4]B.E 4th Year, Dept. of CSE, RYMEC, Bellary, Karnataka(India)
[5]Associate Professor, Dept. of CSE, RYMEC, Bellary, Karnataka(India)

Abstract: We have been using mobiles and laptops for quite a long time, yet their security is poor. Any pattern or alpha-numeric lock can be forged by an un-authentic user. When someone wants to perform a major change in operating system's settings they are not asked to re-authenticate the user. Our application guards the system using facial recognition.

I. Introduction

The objective is to develop an application which increases the security of mobiles and laptops by adding facial authentication system. The proposed system enhances the security of mobiles and laptops. In mobiles and laptops the login consists of either a pattern lock or alphanumeric string both of which can be exploited/forged easily.

Our system uses the biometric style of authentication and enhances the security.

II. Releated Work

For face detection the framework used is Viola–Jones object detection framework and for face recognition the algorithm used was Linear Binary Patten Histogram (LBPH) algorithm. After performing extensive research among the available technologies, we had decided to use Python 3 (open-source, interpreted) programming language and to build the application interface we choose Kivy framework.

Viola–Jones Object Detection Framework

The Viola–Jones object detection framework is the first object detection framework to provide competitive object detection rates in real-time proposed in 2001 by Paul Viola and Michael Jones. Although it can be trained to detect a variety of object classes, it was motivated primarily by the problem of face detection.

Local Binary Patterns Histogram (LBPH)

A local binary pattern (LBP) is a type of visual descriptor used for classification in computer vision. LBP is the particular case of the Texture Spectrum model proposed in 1990. LBP was first described in 1994. It has since been found to be a powerful feature for texture classification; it has further been determined that when LBP is combined with the Histogram of oriented gradients (HOG) descriptor, it improves the detection performance considerably on some datasets. A comparison of several improvements of the original LBP in the field of background subtraction was made in 2015 by Silva et al. A full survey of the different versions of LBP can be found in Bouwmans et al.

Python 3

Python 3 is developed under an OSI-approved open source license, making it freely usable and distributable, even for commercial use. Python's license is administered by the Python Software Foundation.

Kivy Framework

Kivy is an open source Python library for developing mobile apps and other multitouch application software with a natural user interface. It can run on Android, iOS, Linux, OS X, and Windows.

III. Working

To have a facial recognition system we need to have two sub-systems in place, namely:

- **Face detection sub-system:** This detects the face from given sample image.
- **Face recognition sub-system:** This compares two faces using computer vision algorithm and returns prediction.

The face detection sub-system is implemented using Viola–Jones object detection framework which would work fast enough even on cheap cameras.

A typical facial recognition process involves three steps:

- **Training Data Gathering**: Gather face data (face images in this case) of the users.
- **Training of Recognizer**: Feed that face data (and respective names of each face) to the face recognizer so that it can learn.
- **Recognition:** Feed new faces of the persons.

The above mentioned steps will be done using Linear Binary Pattern Histogram (LBPH) algorithm. Once we have these underlying sub-systems functional, we create the user interface using Kivy framework. The interface consists of four screens namely Login, Home, Add-User, and Edit-User. The names are self-explanatory.

The Add-User screen is used to add a new authentic user to the system. The interface allows to input name and their sample image and they will be stored in database.

The Login screen is tied with both Face detection and recognition sub-system. The Login screen takes the input video feed from device's web camera. Although the camera might run at thirty frames per second (this is the minimum configuration for low end camera). But one fame for every half second is fed as input to the face detection sub-system to optimize computation. The face-detection sub-system isolates the face from the frame and returns its coordinates. The face recognition system compares the sample image from camera's feed with other images stored in its database and returns authorization status. If it was an authorized face then they will be redirected to Home screen.

Within the Home screen the interface allows the user to Logout, Add-User, Delete existing user, Edit-User profiles.

When the application is running, the Login screen will be the default current screen, when authentic user is in front of camera, they will be directed to Home screen as successful login. Otherwise the screen waits for successful login from authentic user.

IV. Accuracy

The tolerance level of face recognition sub-system at the time of testing the application was 60%. That is, the sample image received from Login screen and the face data in database must not have deviation lesser than 60%. With this configuration the application has proven to be very accurate in distinguishing faces at the time of testing with various subjects.

V. Time Efficiency

The face detection and face recognition are the most resource intensive sub-systems. But as have optimized the number of fames processed. Thus within one second the user will be recognized and authorization status will be displayed.

VI. Future Scope

Although the application works very well for normal user, but there is scope for improvements in the interface. We could incorporate material design standards and improve the UI. The database which stores the images can also be encrypted.

VII. Conclusion

Digital Security is important because it allows people to use social media and online baking and protects them from risks such as identity theft and fraud. Thus securing our regularly used devices with biometrics using built-in available hardware increases the security of these devices.

VIII. References

[1]. Alexey D. Egorov; Denis U. Divitskii; Anatolyi A. Dolgih; Grigoriy A. Mazurenko 2018 IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering (EIConRus).

[2]. Local Binary Patterns(LBP): https://globaljournals.org/GJCST_Volume13/1-Face-Recognition-using-Local.pdf

[3]. Face recognition with OpenCV: https://docs.opencv.org/3.3.0/da/d60/tutorial_face_main.html

[4]. Kivy developer documentation and guide https://kivy.org/docs/guide/basic.html#