ResearchGate

See discussions, stats, and author profiles for this publication at: https://www.researchgate.net/publication/342853286

Design and Implementation of the Smart Door Lock System with Face Recognition Method using the Linux Platform Raspberry Pi

Article · December 2018

CITATIONS
8

READS
1,514

4 authors, including:

Sourav Roy
Khulna University of Engineering and Technology
45 PUBLICATIONS **387** CITATIONS
SEE PROFILE

Md Nasir Uddin
Southeast University (China)
4 PUBLICATIONS **21** CITATIONS
SEE PROFILE

Md. Jahidul Kabir
Jagannath University - Bangladesh
2 PUBLICATIONS **8** CITATIONS
SEE PROFILE

Some of the authors of this publication are also working on these related projects:

Software Defect Prediction based on Deep Learning Techniques View project

All content following this page was uploaded by Md Nasir Uddin on 23 February 2022.

The user has requested enhancement of the downloaded file.

IJCSN - International Journal of Computer Science and Network, Volume 7, Issue 6, December 2018
ISSN (Online): 2277-5420
www.IJCSN.org
Impact Factor: 1.5

Design and Implementation of the Smart Door Lock System with Face Recognition Method using the Linux Platform Raspberry Pi

[1] Sourav Roy; [2] Md Nasir Uddin; [3] Md Zahirul Haque; [4] Md Jahidul Kabir

[1, 2, 3, 4] School of Computer Science,
Nanjing University of Posts and Telecommunications,
Nanjing 210023, China

Abstract - Privacy and Security are two universal rights and, to ensure that in our daily life we are secure, a lot of research is going on in the field of home security, and IoT is the turning point for the industry, where we connect everyday objects to share data for our betterment. House security matters and people always try to make life easier at the same time. That's why we put up with this project, Face Recognition Door Lock System. Facial recognition is a well-established process in which the face is detected and identified out of the image. We aim to create a smart door, which secures the gateway on the basis of who we are. We want to develop this system based on Raspberry-pi 3, to make the house only accessible when your face is recognized by the recognition algorithms from Open CV library and meanwhile you are allowed in by the house owner, who could monitor entrance remotely. By doing so, the system is less likely to be deceived: since the owner can check each visitor in the remote console, getting recognized by the camera using a photo won't work. I want to add passcode function for entrance in case that face recognition part corrupts.

Keyword - Raspberry Pi, Facial Recognition Door, Home Security, IoT.

1. Introduction

In today's world of connectivity and smart devices there is an urgent need to modify our existing day to day objects and make them smart, also it is not the era when we can blindly trust the old and conventional security measures, specifically speaking is our door locks. To change and modernize any object we need to eliminate its existing drawbacks and add extra functionality.

The major drawbacks in a common door lock are that anyone can open a conventional door lock by duplicating or stealing the key and its simply impossible if we want our friends and family to enter our house, without being actually present over there. Thus why not just eliminate these problems. So, to simply convert this normal door lock into a smart lock, which can open the door whenever we turn up in front of the gate or when we want it to open up for someone else without being physically present, we need to modify the door. So an era has come where devices can interact with its users and at the same time ensure of their safety and keep improvising themselves.

Users could operate on a touchscreen to select entering the house by recognizing the face or motors and/or adding a digital number pad to take inputs from the user or adding Infra-Red or Bluetooth modules to operate these devices.
For face recognition, an image will be captured by a pi camera and pre-processed by Raspberry pi like converting, re-sizing and cropping. Then face detection and recognition are performed. Once the face is recognized by the classifier based on a pre-stored image library, the image will be sent to a remote console waiting for house owner's decision. For the passcode part, users could enter or reset passcode through a keypad.

2. Related Literature and Comparison with Existing Models

Since 2010 the industry has seen a dawn of work being done in fields of Artificial Intelligence, Machine Learning, Neural Networks, IoT, Big Data Analytic all with a common goal to make things easier, self-supervising and to interconnect all kinds of devices by making everyday objects interconnected and interoperable.

A need has been felt in the field of digitalizing conventional security tools and thus a lot of work has been modeled on making daily life locks smart by introducing locks movable with the help of stepper

Copyright (c) 2018 International Journal of Computer Science and Network. All Rights Reserve

An intensive study of literates implementing Smart Locks had been done and literature implementing Door Locks with the help of GSM phones [1] and stepper motors have been studied. Also, literature regarding smart display have been thoroughly reviewed [2]. The fault in existing models is a complexity of a system and unnecessarily relying on extra components. Our model is unique with its one of a kind combination of functionalities offered and the simplicity of the model. A major difference is in the overhead reduction by the application as it detects the face out of the image and sends it directly to application program interfaced with our application, which has not been provided in any existing model and the efficient use of solenoids, which also eliminates the use of stepper motors. So, we have avoided the use of unnecessary components like stepper motors and drivers as done in existing models and also we have given newer and unprecedented features of facial recognition as an access point control system with a combination of relay module with a solenoid to open the gate and unique and interactive User Interface. Also rather than using a low-quality Raspberry Pi Interfaced Camera we have used USB attachable HD WebCam to do efficient and reliable facial recognition.

The objectives of the proposed work is to implement a working model of a smart door and to give solutions to the problem faced by people in day to day incidents of burglary or losing the key and also to promote and ignite the work being done on IOT systems and implementing it with the help of key research areas of Neural Networks and IoT APIs and protocols.

This model is allowing people to add more functionality to it and thus induce more research work in the field of AI, Machine Learning, IoT and lot more.

3. System Design and Framework

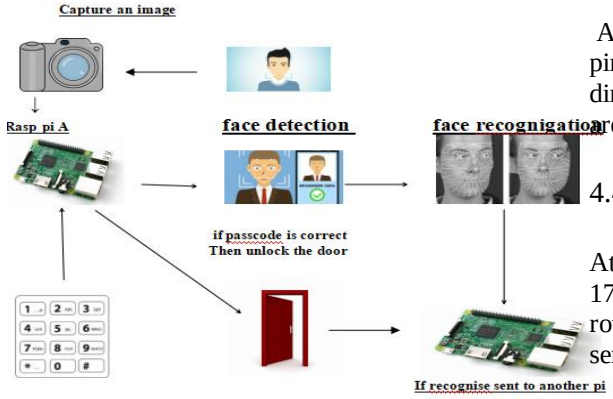

Figure1:Framework

4. Proposed Methodology

In order to implement the smart door model we need a list of materials which is briefly mentioned below:

4.1 Raspberry Pi

Raspberry Pi (RP) is an ARM-based single board computer. The Raspberry Pi 3 Model B is the third generation Raspberry Pi [3].It has Broadcom BCM2837 64bit ARM Cortex-A53 Quad Core Processor SoC running at 1.2GHz and 1GB RAM. The operating system used for Raspberry Pi is Raspbian as it is open source anyone can use. Raspbian is a Linux-based computer operating system. It has 40 pins in which 24 are GPIO pins these pins are used for general purpose, 8 ground pins, two of each 5V and 3V power pin. It has four USB-2 ports and a Micro USB power source. It runs on the 5V power supply. Additionally, it adds wireless LAN (BCM43143 WiFi on board (802.11 a/b/g/n)) and Bluetooth connectivity making it the ideal solution for powerfully connected designs.

4.2 PI CAMERA

Pi Camera Module is a custom designed add-on for Raspberry Pi. This interface uses the dedicated CSI interface, which was designed especially for interfacing to cameras [4] The CSI bus is capable of extremely high data rates, and it exclusively carries pixel data. The sensor itself has a native resolution of 5 megapixels and has a fixed focus lens on board. In terms of still images, the camera is capable of 2592 x 1944-pixel static images, and supports 1080p30, 720p60 and 640x480p60/90 video.

4.3 KEYPAD

Attach matrix 7-pin interfaces to 7 free GPIO pins, 3 column pins are set as output which are directly connected with GPIO, while 4 row pins are set as input with pull-up resistor.

4.4 SERVO

Attach the servo to a GPIO (we selected GPIO 17 here) of the Raspberry pi A and control its rotation utilizing pulse-width modulation. The servo is powered by a 6V-battery pack.

Copyright (c) 2018 International Journal of Computer Science and Network. All Rights Reserve

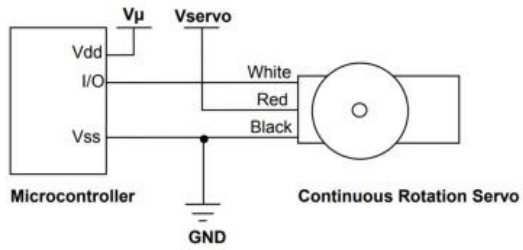

Figure 2: Servo

5. Software and Algorithm

5.1 OpenCV

OpenCV is an open source computer vision software library. The library has a lot of optimized algorithms, which can be used in many IOT related sectors including face detection and recognition. As the libraries of our project we liked to use the Haar classifier, LBPH (Lower Binary Pattern histogram) face recognizer.

5.2 Image processing

Image processing is a mathematically intensive operation & one of the biggest areas of research for a big data field. It is a process on the image to convert it into desired looking for which the input is an image and the output may be an image or set of characters related to the particular image. It refers to a variety of techniques that are used to maximize the information yield from a picture.

5.3 Haar Cascade Classifier Algorithm

A Haar Cascade is basically a classifier which is used to detect the object for which it has been trained for, from the source.

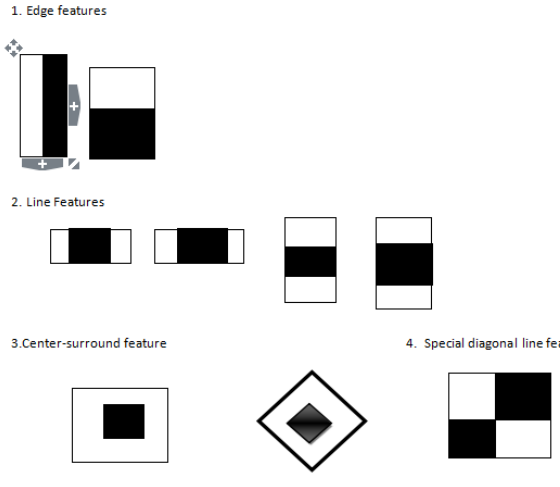

Figure 3: Cascade classifier

This proposed system uses Haar Cascades classifier as a face detection algorithm [5]. Firstly, the algorithm needs a lot of positive images and negative images to train the Haar cascades classifier. Positive images are images with clear faces where negative images are those without any faces. Haar cascades are similar to the convolutional kernel which are shown below in figure 3.

Each feature is represented as a single value obtained from the difference of the sums of pixels in white rectangle from the sum of all pixels in the black rectangle. All different possible sizes and locations of classifier is used for calculating of plenty of features. As the number of classifiers increase the arithmetic computations seems to take a long time. Instead of it, the concept of Integral Image has been used. Image Processing Integral image is a data structure which is a summed area table and algorithm for quickly and efficiently generating sum of values in a rectangular grid subset. Integral image is derived by using following equation,

$$I_{\sum}(x,y) = \sum_{x' \le x} i(x' + y')$$

To avoid the complexity of calculation we use Adaboost machine learning algorithm, which is inbuilt in OpenCV library that is cascade classifier, to eliminate the redundancy of the classifiers. Any classifier which has a probability of 50% of more in detection is treated as weak classifier. The Sum of all weak classifier gives a strong classifier which makes the decision about detection. Although it is very vague to classify with one strong classifier we use the cascade of classifiers. Classification takes place in stages, if the selected region fails in the first stage, we discard it. We don't use the classifiers on that region which is discarded. The region which passes all the stages i.e. all strong classifiers is treated as the detected face. Detected Faces are passed to the Face recognition phase. Local Binary Patterns histogram algorithm (LBPH) has been used for face recognition. Local binary patterns are simple at the same time very efficient texture operator which assigns the pixels of the image by comparing with the adjacent pixels as threshold and which results in a binary result. The detected integral image is subjected to this Local binary pattern. Face recognition is extremely vulnerable to the environment changes like brightness, facial expressions and position. Face preprocessing is the module which reduces the problems that makes the picture unclear to recognize the face

Copyright (c) 2018 International Journal of Computer Science and Network. All Rights Reserve

such as less brightness and contrast problems and noise in the image and make sure the facial features always be in a constant position. In this project we use histogram equalization for face preprocessing. For efficiency we use separate preprocessing which is histogram equalization for left and right face. So histogram equalization is done three times, firstly for the whole face and the other two for side faces [6].

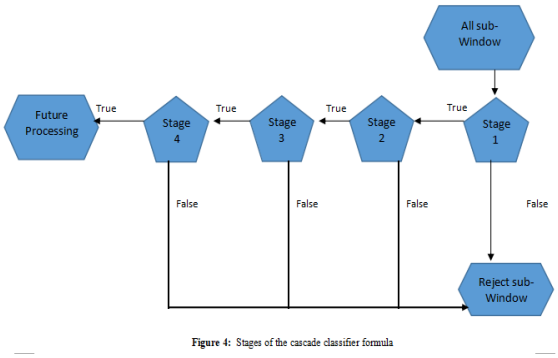

Figure 4: Stages of the cascade classifier formula

For better working, each stage of the cascade must have a low false negative rate, because if the actual object is classified as a non-object, then the classification of that branch stops, with no way to correct the mistake made. However, each stage can have a relatively high false positive rate, because even if the n-th stage classifies the non-object as actually being the object, then this mistake can be fixed in n+1-the and subsequent stages of the classifier [7].

6. TEST and Implementation

6.1 User Interfaces:

For user interfaces, we utilized python library 'pygame' to design the aesthetics of PiTFT touchscreen and detect any user input.

6.1.1 Rasp Pi A: The way we linked touchscreen buttons with system functions on Pi A is displayed in Figure 5.

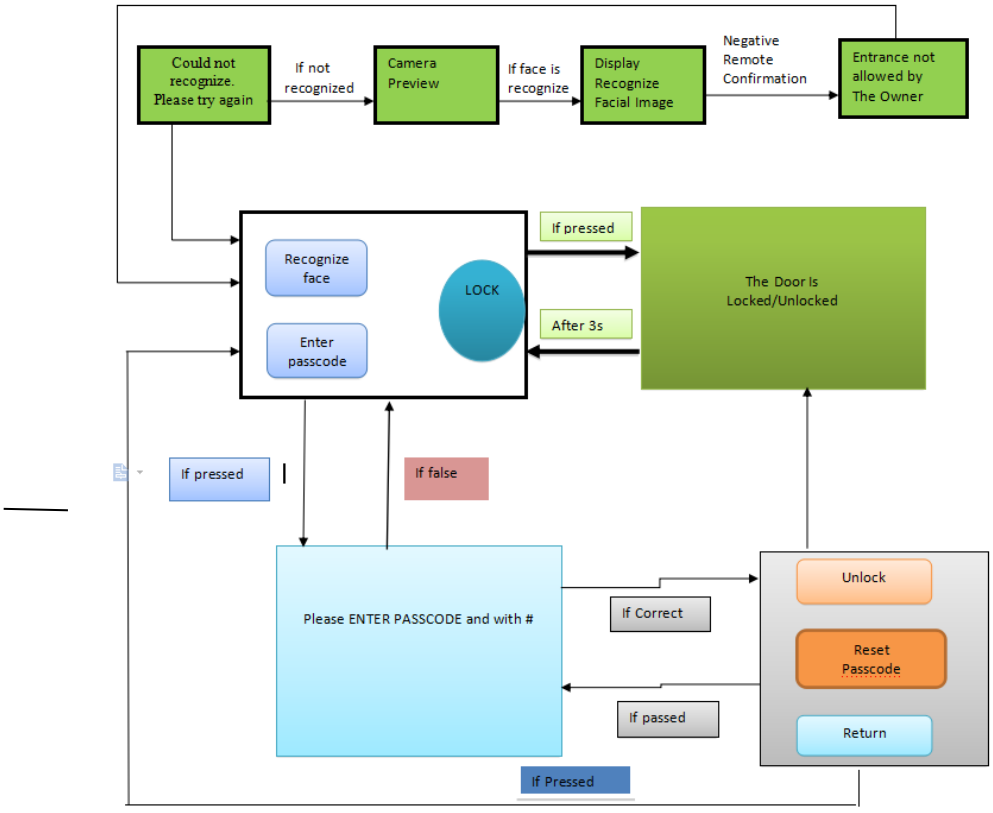

Figure 5: System Algorithm

Initially, on the PiTFT screen, there are 3 buttons in the 'main' level: 'Lock', 'Recognize Face' and 'Enter Passcode'. The 'Lock' button is designed for house owners to know whether the door is locked and lock the door manually. When the door is unlocked, the house owner could press the 'Lock' button to lock the door. If the door is already locked, nothing will happen when the 'Lock' button is pressed.

Once the 'Face Recognize' button is pressed, you could preview pi camera on the screen and the image will be captured in 5 seconds. After a series of image processing steps, if the face cannot be recognized, a message 'Could not

Copyright (c) 2018 International Journal of Computer Science and Network. All Rights Reserve

recognize, please try again' will appear on the screen for 3 seconds before screen return to 'main' level. Otherwise, the recognized image will be displayed until remote confirmation is made. If Pi A receives a positive response from Pi B, that is, the one is allowed in by the house owner, the door will be unlocked by the servo and 'The door is unlocked' will show up on the screen. If Pi A receives a negative response from Pi B, you will be told that 'Entrance not allowed by the house owner'. At last, after 3 seconds, the system will return to its 'main' level.

When a user presses the 'Enter Passcode' button on the screen, the text 'Please enter the passcode, end with #' can be seen on the touchscreen and the user needs to enter the passcode using the 3x4 matrix keypad. Each pressed number on the keypad will be displayed on the screen and then be covered by a star if the next key has been pressed. If the passcode entered is wrong, then we have the PiTFT display a message, 'Password does not match, please try again', which will last for 3 seconds before going back to 'main' level.

On the other hand, if the passcode entered is correct, the system will get into the second-level buttons: 'Unlock', 'Reset Passcode' and 'Return'. Users could unlock the door by pressing 'Unlock' button and the servo will be driven to open the door. The password could be reset by pressing the 'Reset Passcode' button. The new passcode needs to be entered through the keyboard too and will take effect next time. The 'Return' button is designed to return to the previous level. For security reason, we set the upper limit for times of entering the wrong passcode. If the upper constraint is reached, the whole system will be locked for a certain amount of time, such as 15 minutes and screen will tell the user that 'Maximum failed passcode attempts in restriction. Please try again after 5 minutes.

6.1.2 Rasp Pi B

Figure 6: Permission

Pi B works as the server waiting for the connection from Pi A and its user interfaces only show up when Pi B receives an image. The image will be displayed on its PiTFT screen plus 2 buttons 'Yes' or 'No', by pressing which the corresponding response will be sent to Pi A.

6.1.3 Test

Since there are many levels of touch buttons and we need to display various messages in different circumstances, we observed pygame window on the computer monitor and debugged our user interfaces layer by layer. We tested the sensitivity and reaction time of each touch button and experimented with different sleep time to make sure buttons all work well. Besides these, we also did some corner tests to see what would happen.

7. Face Detection and Haar Cascade Classifier

After preprocessing, like resizing and cropping, the image will be used as input of Haar Cascade Classifier to detect whether there is a single face detected in this image.

Face detection is a process of finding out the face area in the image. In the project, we use Haar Cascade to detect faces. Haar-like features such as edge features, line features, and center-surround features are used and they are inputs of classifiers. Cascade classifiers test the image by cascade features. Since the amount of features is large, instead of applying all features on the window, the features are divided into different stages. The window will be tested stage by stage and initial stages usually have less haar-like features. If the window fails in a stage, it will be discarded and the following stages won't be tested. The window which successfully passes all the stages is considered to be face image. Haar cascade classifiers has an advantage of its fast detection speed compared to other classifiers.

7.1 Test

We tried to resize captured images to different sizes and different classifier tuning parameters like scale factor, a minimum number of neighbors, to find out the best configuration for face detection. See details in 'face.py' in code appendix.

8. Face Recognition and Eigenface Classifier

If a single face is detected by Haar Cascade classifier, the face will be cropped out of the scene. Then the Eigenface classifier that has already been trained by pre-stored image library,

Copyright (c) 2018 International Journal of Computer Science and Network. All Rights Reserve

will try to recognize the cropped face and return the confidence of its prediction at the same time. By setting a threshold for the predicting confidence, we can determine when to acknowledge that the face is truly recognized.
Face recognition is matching the input signal with the pre-stored library. Though the input signal is noisy due to the different angle, position and intensity of light, the image could be recognized according to the position of eyes, face, and mouth in the face, and their relative distances between each other. These features are called eigenfaces and they could be extracted from original image data by principal component analysis. Each face is represented by a subset of eigenfaces and the face could be reconstructed if eigenface could be correctly calculated for each proportion. The new image will be recognized based upon eigenvectors and the Euclidean distance between eigenvectors.

8.1 Test

We tested the eigenface classifier with different confidence threshold to get the best recognition accuracy. After the system could recognize one face successfully, we also considered making the system recognize multiple faces by setting up several classifiers and let them do recognition work in turns or in multi-thread way. As a result, due to the limited time, we continued on the former choice which is simpler than multi-thread.

9. Passcode Implementation

In the implementation of the keypad, the approach for deciding the pressed button is firstly setting outputs (column pins) and inputs (row pins) as high. When a button is pressed, it will produce a low signal. Then each column is scanned in a 'for' loop. If the low signal is detected, we could know which button is pressed. See details in 'passcode.py' in code appendix.

9.1 Test

In the test of passcode function, the system worked well on the monitor screen at the beginning. But in the condition that all buttons and layers were displayed on the PiTFT screen, the PiTFT screen would display sliding messy codes when we entered the password ending with '#'. The TA helped us debugging and told us the problem was caused by wrong circuit connection. We checked our circuit and found the reason is that we attached one of the keypad interface pins to GPIO 25 which is used for special function. We changed the connection to GPIO 21 and the problem was solved.

10. Communication over TCP Sockets

The two Raspberry is in the project are set to communicate using TCP sockets via wifi modular and a wifi router. The Raspberry Pi A, which displays main buttons, is the client and the other Raspberry Pi B, for remote confirmation, is the server. With specific IP addresses, the client could connect to the server and send files like images over TCP socket. Then the server will send a response back to the client and once the response received, the client is going to close the connection. We developed a python script code for the above communication routine by leveraging python library 'socket'. See details in 'server.py' and 'client.py' & 'client2.py'.

10.1 Test

When we started this communication part, there were 2 choices when deciding which Pi worked as a server or client. We tested on both 2 methods and found that Pi B needs to wait for images sent by Pi A at any time, so it'd better be a server. By doing so, whenever Pi A, the client, is trying to send an image to PiB, it could be connected to Pi B successfully since the server keeps on all the time.

11. Conclusion

We successfully finished a face recognition door lock system as we planned. The passcode and face recognition works well. There is high accuracy in recognizing house owner faces and it could realize sending the matched face image to another Raspberry Pi in time and give a good output. And it takes a little bit time to recognize. At all, we all are satisfied to build it.

Acknowledgement
First of all, we are thankful to all our team members for working hard around two months on this project and gave their great approach on it. At beginning of this project we were in trouble for starting it with the appropriate way. We all were confused from where can start, at that fortunately we got an opportunity to discuss this matter to our honorable lecturer Professor Xu he who was representing about a project related to IoT. By getting that chance we told him at once about our problem. Then he gave the suggestion and guided us for the correct way. Whenever we faced difficulty to work on it then we never felt hesitate to go through him and every time he made our work to overcome all the

Copyright (c) 2018 International Journal of Computer Science and Network. All Rights Reserve

difficulty. So we all give him a great credit and thanks as well. Finally, we are also delighted to complete this project as our demands eventually outweigh any negative challenges.
gave the suggestion and guided us for the correct way. Whenever we faced difficulty to work on it then we never felt hesitate to go through him and every time he made our work to overcome all the difficulty. So we all give him a great credit and thanks as well. Finally, we are also delighted to complete this project as our demands eventually outweigh any negative challenges.

References

[1] Jie-Ci Yang et. all An Intelligent Automated Door Control System Based on a Smart Camera.
[2] S. Nazeem Basha et all An Intelligent Door system using Raspberry pi and Amazon Web services IOT.
[3] Richard Grimmett, Raspberry Pi Robotic Projects. Packt Publishing.
[4] W. F. Abaya, J. Basa, M. Sy, A. C. Abad and E. P. Dadios, "Low cost smart security camera with night vision capability using Raspberry Pi and OpenCV," 2014 International Conference on Humanoid, Nanotechnology, Information Technology, Communication and Control, Environment and Management (HNICEM), Palawan, 2014, pp. 1-6
[5] Priya Pasumarti1, P. Purna Sekhar "Classroom Attendance Using Face Detection and Raspberry-Pi" International Research journal of Engineering and Technology (IRJET), Volume: 05 Issue: 03. p3-p5,Mar-2018
[6] S Rajkumar, J Prakash, "Automated attendance using Raspberry pi", International Journal Of Pharmacy & Technology (IJPT), Vol. 8, No. 3, pp. 16214-16221, September 2016.
[7] T. M. Inc., "Train a Cascade Object Detector," [Online]. Available: http://www.mathworks.se/help/vision/ug/train-a-cascadeobject-d

Copyright (c) 2018 International Journal of Computer Science and Network. All Rights Reserve

View publication stats