International Journal of Technology Research and Management
ISSN (Online): 2348-9006
Vol 9 Issue 1 January 2022

Computer Vision Based Automatic Attendance System Using a Raspberry Pi 3 Computer

Anish Jain[1], Prof. Jagrati Nagadiya[2]
M. Tech Research Scholar, Information Technology, Mahakal Institute Technology, Ujjain, India[1]
Professor, Information Technology, Mahakal Institute Technology, Ujjain, India[2]
anish.jain.sym21@gmail.com[1]

Abstract: *The face of a person can be used to uniquely identify him/her. A face detection and face recognition system can be a very effective way to mark the attendance of students in a class. The use of such a system eliminates the use of paper and it also reduces the chances of errors made by humans while recording the attendance. Furthermore, since the attendance records are automatically stored in a digital format, it gets convenient to update, manage, and share the records with the concerned stakeholders. This system uses a Raspberry Pi 3 which is a credit card sized computer, as the main hardware. The system utilizes the computer vision library OpenCV for face detection and face recognition, and uses google sheets to store the records of the attendance. The system when integrated with other modules, is able to send automed notifications through e-mail and SMS. The system has the ability to replace conventional methods of recording attendance in classes, and with the use of this system, a lot of paper and valuable time of the class can be saved.*

Keywords: *Artificial Intelligence, Automatic Attendance System, Computer vision, Face Detection, Face Recognition, Machine learning, OpenCV, Raspberry Pi 3.*

1. INTRODUCTION

Accurate attendance recording for the students is an important task in every class. When done manually it generally wastes a lot of productive time of the class. As it is a repetitive process, student authentication can be automated using various methods available in the market like biometric attendance. This proposed solution for the current problem is through automation of attendance system using face detection and face recognition, since face is the primary identification for any human.

This project describes the method of detecting and recognizing the student's faces in real-time where a Raspberry Pi 3 model B is used for computation in the detection and recognition modules. This project describes an efficient algorithm using open-source image processing framework called OpenCV. This system is build using five modules, namely: – Face Detection, Face Preprocessing, Face Training, Face Recognition and Attendance Database. The face database is collected to recognize the faces of the students. The system is initially trained with the images of student's faces, which is collectively known as student database. The system uses user friendly User interface to maximize the user experience while both training and testing which are collecting student images and taking attendance with the system.

This project can be used for many other applications where face recognition can be used for authentication. Raspberry Pi usage helps in minimizing the cost of the product and the usability as it can be connected to any device to take the attendance.

This project uses OpenCV along with modified algorithm of Haar cascades proposed by Viola-Jones for face detection, Eigen faces, fisher faces with LBP histograms for face recognition, and uses Google Sheets and Google Cloud Server to update the database. The system will automatically update the student's presence in the class to the student's database.

Thus, with the use of this system, a lot of paper and overhead in maintaining student attendance would be minimized..

2. BACKGROUND WORK

2.1 Related Work

Detailed In the recent years, there has been a lot of research and development in the field of Artificial Intelligence and Machine learning. A lot of this research has been focused on Computer vision, specifically on Face recognition, Object recognition, and Optical character recognition. A huge example of this research and development is the car manufacturing company Tesla. They have devised computer vision algorithms focused on Face recognition, Object recognition to design autonomous vehicle (driver-less cars) which are very efficient.

Similar research and development has been carried out by multiple young as well as established organizations, with an aim of making life easier by automating tasks that are either too trivial to be performed by humans, or are very repetitive in nature.

Some examples of these developments are:

•Computer vision and 3D depth-sensing technology

•AI tool can spot weapons and notify authorities of threats

•Computer vision platform that can identify pathologies

•Computer vision with (RFID), to allow retailers to track inventory in real time

•Uses lasers, mirrors and sensors to measure distances

•Capturing high-quality 3d images and generating virtual tours

•Using computer vision to determine the authenticity of government-issued ids to stop identity theft

•Scanning photographs and videos of disaster scenes to make an estimate of the amount of financial damage involved

•Unmanned drones for aerial surveillance

•Using image recognition to facilitate a checkout-free shopping experience for retail store customers

2.2 Research Gap

Although there has been very rapid growth in both Computer vision sector as well as Artificial Intelligence as a whole in the recent years, there hasn't been a lot of research and development in the niche of education and teaching. The world has seen developments in computer vision through the development of applications that can convert the text in an image to text and further converting the text into speech. But there haven't been many notable applications in the category of facial recognition when it comes to the education and teaching industry. A major reason for this exclusion could be the intent of focusing on broader issues, by solving the problems of masses which occur in everyday lives. Another reason could be not identifying the need of application of Artificial Intelligence and Computer vision in the industry. There have been some advancements in the application of facial recognition techniques in the education and teaching industry, but they have been limited to either individuals or small groups. Unlike commercial level developments in other industries like developing a surveillance camera capable of performing facial recognition and sending alerts to the users, there haven't been any developments on a commercial level in the education industry.

3. CHALLENGES

3.1 Introduction

Although Artificial Intelligence and Machine learning are not very new concepts, but the amount of research and development for Facial recognition techniques which is openly available in public domains is very limited. One can easily find the working and processes for Facial recognition techniques in books or on the internet, but the major challenges are faced when it comes to the implementation of those techniques and processes in real life applications. When those techniques are used in real life applications for practical use cases, the developers very often run into errors which have very limited troubleshooting information available on the internet.

The case with this project was fairly similar. Being a new and emerging technology which has limited open-source documentation and resources available in the public domain, there were multiple challenges which were hard to overcome. Even though most of the challenges were hardware related challenges, the majority of the challenges were in the software domain.

Some major challenges that were faced during the development of this project are mentioned below which have been classified into two categories based on the domain.

3.2 Hardware Challenges

3.2.1 A display screen for the system

The Raspberry Pi is a small computation unit. It has minimal components, which means that it has components which are absolutely essential, but misses some less essential but very important components such as a display screen. The challenge was to connect a display screen to the system which is responsive and portable. For this case, a computer screen was connected to the Raspberry Pi using SSH and X11 forwarding to ret the GUI of the Raspberry Pi displayed on the computer screen.

Challenge Complexity: High

3.2.2 Overheating of the system

Since the Raspberry Pi is a small computation unit, it has minimal components. This means that the Raspberry Pi does not have a cooling unit like an in-built fan. This leads to overheating of the system when the processor has a lot of load on it, usually while running heavy programs.

Challenge Complexity: Low

3.2.3 Limited memory of the system

A major challenge while creating this project was the selection of the hardware to be used. While Raspberry Pi fits most of the requirements of the project, it is a mini computing system with limited power and memory. For this case, all unnecessary programs were removed from the Raspberry Pi.

Challenge Complexity: High

3.2.4 Fragility of the system

Since the Raspberry Pi is a small computation unit in the form of a single circuit board, it lacks a protective covering that ensures that the system is protected from shocks and drops. For this case, we used a case for the Raspberry Pi that protects it from dust, water, shocks and other damages.

Challenge Complexity: Low

3.2.5 Portability of the system

Although the Raspberry Pi system is small in size and is designed to be portable, it needs to be connected with external devices such as a keyboard, mouse, webcam, display, and a power source. These connections reduce the scope of portability of the system. For this case, we used a computer screen as a display and to control the system, which eliminated the use of a keyboard, mouse, and, display, making the system more potable.

Challenge Complexity: High

- **3.3 Software Challenges**

3.3.1 Selecting an optimal Computer Vision platform

Since the Raspberry Pi has limited computational ability and memory, it was a challenge to select a Computer Vision platform that is not very heavy in terms of the computational load, while maintaining the reliability of the system. A platform that requires a lot of computation not only occupied a lot of space on the Raspberry Pi, but it also consumed a lot of resources and increased the time complexity of the system.

Challenge Complexity: High

3.3.2 Selecting an optimal Face Recognition mechanism

Selecting an optimal Face Recognition mechanism was a challenge because we had to make sure that the chosen mechanism did not slow down the system, while ensuring that the reliability and accuracy of the system are not compromised. We had to try out multiple mechanisms and then select the one that fit our requirements for the project.

Challenge Complexity: High

3.3.3 Ensuring version compatibility

Since a lot of libraries and dependencies had to be installed on the Raspberry Pi for the project, it became a challenge to ensure that the version of each of the libraries and dependencies was compatible with other libraries and dependencies. There was an instance when the python version was not compatible with the Gspread version, and hence OpenCV had to be installed all over again with a different version of python to ensure smooth functioning of the system.

Challenge Complexity: High

3.3.4 Optimizing the data pre-processing

To ensure that the accuracy of the system is high, the image data had to be pre-processed in order to reduce the load on the system and reduce the response time of the system. This was achieved by pre-processing the images in the database by cropping the images and converting them to grayscale. This was challenging because the specifics of the images in the database had to be matched with the specifics of the images recorded through the system.

Challenge Complexity: Low

3.3.5 Reducing the time complexity for the wireless display

It was essential to connect a display to the system in order to interact with the GUI and perform specific actions with ease. The display could have been connected to the system in two ways. First, connecting an external screen through the HDMI port and connecting a keyboard, and a mouse. But this would decrease the portability of the system. Hence we had to choose the other option of using a computer/laptop screen over SSH.

Challenge Complexity: Low

3.3.6 Reducing the time complexity for the system

OpenCV is a heavy library that requires a lot of resources and since the Raspberry Pi has limited computational ability and memory, it was a challenge to maintain a decent response time on the system.

Challenge Complexity: Low

3.3.7 Connecting a flexible database to the system

The hardest challenge faced through the entire course of this project was to select and connect a reliable, fast, and flexible database to the system. After a lot of research and trying multiple options, we decided to choose Google sheets for the project as it was fast, reliable and easy to share with other users. But the main task was to integrate it with the project without disturbing the existing functionalities.

Challenge Complexity: High

3.3.8 Reliability and Accuracy of the system

With a limited number of resources, it was a challenge to ensure reliability and accuracy of the system. Considering that the Raspberry Pi has limited computational ability and memory, the code of the system had to be kept clean and efficient without compromising on the reliability and accuracy of the system.

Challenge Complexity: High

4. PROPOSED SOLUTION

4.1 Product Overview

The present day attendance system is manual. It wastes a considerable amount of time both for teachers and students. There are still chances for proxies in the class when attendance is taken manually. Manual attendance always has a cost of human error. So automating the attendance process will increase the productivity of the class. Face identification separates faces from non-faces and those countenances that can be perceived. In this proposed system we take the attendance using face recognition which recognizes the face of each student in front of it while entering the class.

In this project, the objective is to develop an Automatic Attendance System, 'Auttendance', in which a Raspberry Pi 3 model B is used for the computation in the detection and recognition modules. A webcam is associated with the Raspberry Pi module. This project describes an efficient algorithm using open source image processing framework called OpenCV. This system is build using five modules namely: Face Detection, Face Preprocessing, Face Training, Face Recognition and Attendance Database. The programming language used in this project is Python. The face database is collected to recognize the faces of the students. The system is initially trained with the images of student's faces which is collectively known as student database. The system uses a user friendly User interface to maximize the user experience and ease of accessibility, while both training and testing which are collecting student images and taking attendance with the system.

This project uses the modified algorithm of haar's cascades proposed by viola-jones for face detection and Eigen faces,fisher faces with LBP histograms for face recognition.

- **4.2 TOOLS AND TECHNIQUES**

We have used the following tools for the completion of the project:

•Raspberry Pi 3 model B :

The Raspberry Pi 3 is a small single board computer a Broadcom system on a chip (SoC), which includes an ARM compatible central processing unit (CPU) and an on-chip graphics processing unit (GPU, a Video Core IV). CPU speed is 1.2 GHz and on board memory is 1 GB RAM. Secure Digital (SD) cards are used to store the operating system and program memory in Micro SD sizes. It has four USB slots, HDMI and composite video output, and a 3.5 mm phone jack for audio. Lower level output is provided by a number of GPIO pins which support common protocols like I²C. It also has an 8P8C Ethernet port and an on board Wi-Fi 802.11n and Bluetooth.

•Python 3.5:

Python is a widely used high-level programming language for general-purpose programming. An interpreted language, Python has a design philosophy which emphasizes code readability and a syntax which allows programmers to express concepts in fewer lines of code than possible in languages such as C++ or Java.

•OpenCV 3.2.0:

OpenCV (Open Source Computer Vision Library) is an open source computer vision and machine learning software library. OpenCV was built to provide a common infrastructure for computer vision applications and to accelerate the use of machine perception in the commercial products. Being a BSD-licensed product, OpenCV makes it easy for businesses to utilize and modify the code.

Along with well-established companies like Google, Yahoo, Microsoft, Intel, IBM, Sony, Honda, Toyota that employ the library, there are many startups such as Applied Minds, VideoSurf, and Zeitera, that make extensive use of OpenCV.

Figure 1: OpenCV Diagram

Techniques which we have used in our project:

•Face Detection:
This project uses modified algorithm of haar cascades proposed by viola-jones for face detection and uses Eigen faces, fisher faces with LBP histograms for face recognition.

•Face Preprocessing:
The images of the faces are preprocessed in order to scale them so as to just keep the face in the image and convert it to gray scale.

•Face Training:
The system is trained using the dataset consisting of the images of the students whose faces are to be recognized and whose attendance has to marked.

•Face Recognition:
Then face detection algorithm is applied to the snapshot to detect the faces in the image. Images are subjected to face segmentation and then face recognition is done using the test image stored which tries to match with the student's database images

•Attendance Database:
When the recognition of the student is completed the attendance is automatically updated to the database.

- **4.3 Software Design Description**

System design is a solution on how to approach to the creation of the system. This is an important phase as it provides the understanding and procedural details necessary for implementing the system recommended in the feasibility study. The design step produces a data design, an architectural design and a procedural design. The data design transforms the information domain model created during analysis into the data structures that will be required to implement the software. The architectural design defines the relationship among major structural components into a procedural description of the software. Source code generated and testing is conducted to integrate and validate the software.

From a project management point of view, software design is conducted in two steps; Preliminary design is concerned with transformation of requirements into data and software architecture. Detailed design focuses on refinements to the architectural representation that leads to detailed data structure and algorithmic representations of software.

•Logical Design:
The logical design of an information system is analogous to an engineering blueprint or conceptual view of an automobile, It shows the major features and how they are related to one another. The detailed specifications are drawn on the basis of user requirements. The outputs, inputs and relationship between the variables are designed in this phase.

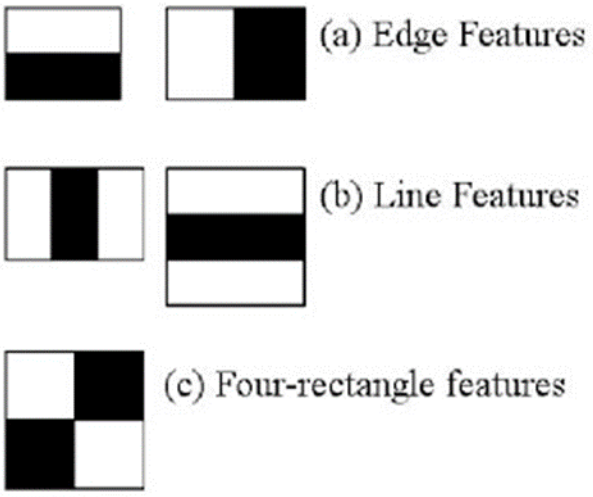
The objectives of database are accuracy, integrity and successful recovery from failure, privacy and security and overall good performance.

•Data Design:
Data design is the first of the three design activities that are conducted during software engineering. The impact of data structure on program structure and procedural complexity causes data design to have a profound influence on software quality. The concepts of information hiding and abstraction provide the foundation for an approach to data design.

- **4.4 Detailed Description of Components**

This Automatic Attendance System can be broken down into two major components; viz. Teacher Module and Student Module. These two components work parallel to give the desired outcomes or meet the goals.

•Component 1: Face Detection Module
The first module is the Face Detection module. This proposed system uses viola jones algorithm for face detection which uses modified haar cascades for detection. Current face detection systems are quite reliable but face recognition systems are less accurate compared when we use them in real-life applications. OpenCV has preinstalled cascade classifiers. In this project we use two cascades one for detecting Faces and one for detecting eyes.



•Component 2: Face Recognition Module

Detected Faces are passed to the Face recognition phase. In this phase we use Local Binary Patterns algorithm 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 which results in decimals is represented as histogram for every integral image. Face recognition is extremely vulnerable to the environment changes like brightness, facial expressions and position.

5. CONCLUSION AND FUTURE SCOPE

This section includes the conclusion reached after developing the Automatic Attendance System 'Auttendance'. The comparison is done between the system that was built and original requirements that were designed at the beginning of the project. It also describes the future work that is intended to be accomplished with later versions of the application.

5.1 Conclusion:

•We come to realize that there are extensive varieties of methods, for example, biometric, RFID based and so on which are tedious and non-productive. So to defeat this above framework is the better and solid arrangement from each keen of time and security. Hence we have accomplished to build up a solid and productive participation framework to actualize an image handling algorithm to identify faces in classroom and to perceive the confronts precisely to check the attendance.

•The risk management process should not be compromised at any point, if ignored can lead to detrimental effects. The entire management team of the organization should be aware of the project risk management methodologies and techniques. Enhanced education and frequent risk assessments are the best way to minimize the damage from risks.

5.2 Future scope:

•The same project can be utilized for several security applications where authentication is needed to access the privileges of the respective system. It can be used in recognizing guilty parties involving in unauthorized business. Face recognition algorithm can be improved with respect to the utilization of resources so that the project can recognize more number of faces at a time which can make the system far better. Many variants of the project can be developed and utilized for home security and personal or organizational benefits.

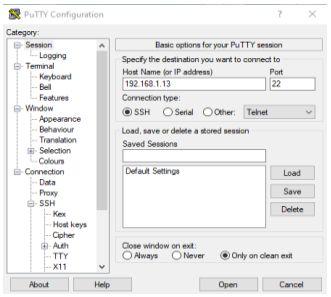
•By making some modifications in the project, it can be used as a security surveillance system, monitoring system etc.

•OpenCV is a very powerful tool which is also capable of Object Recognition. Hence, by making few amendments in our project, the resulting system would be able to perform – Face Detection, Face Recognition, Object Recognition, Optical Character Recognition, Barcode detection and Recognition, and much more. Therefore, the future scope of the project is as immense as the imagination of the developer.

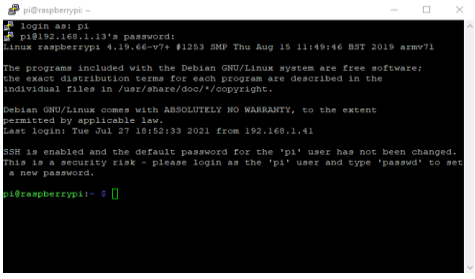
List of Visuals

Visual 1 – A Raspberry Pi 3 model B

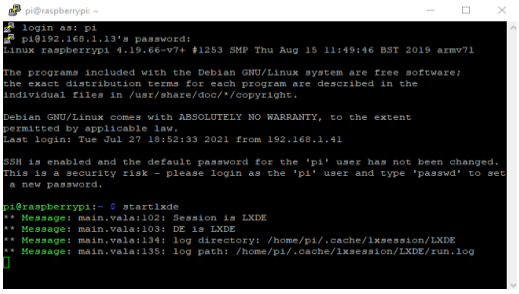
Visual 2 – Project setup with minimum equipment



Visual 3 – SSH login from laptop into the Raspberry Pi

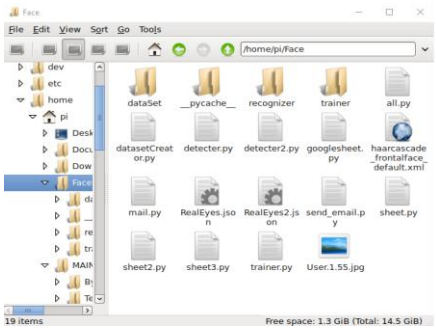


Visual 4 – Logging into the Raspberry Pi through wireless display

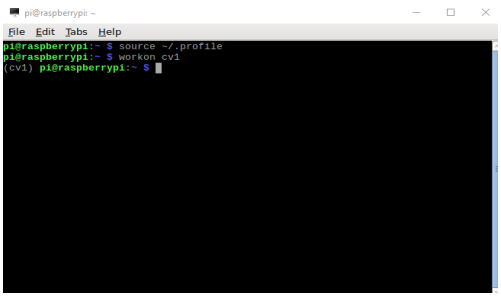


Visual 5 – Loading the GUI through command line interface

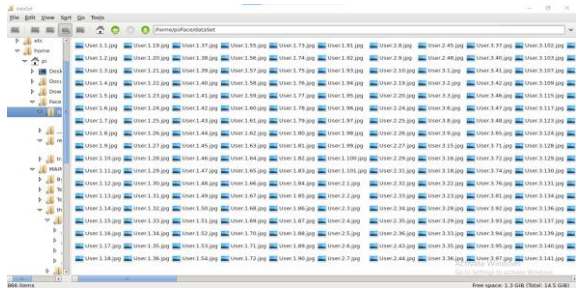
Visual 6 – GUI after loading on the wireless display



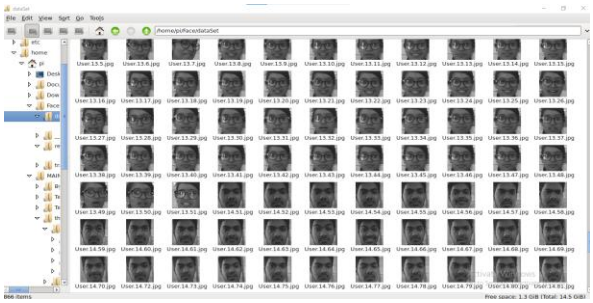
Visual 7 – The project files on Raspberry Pi



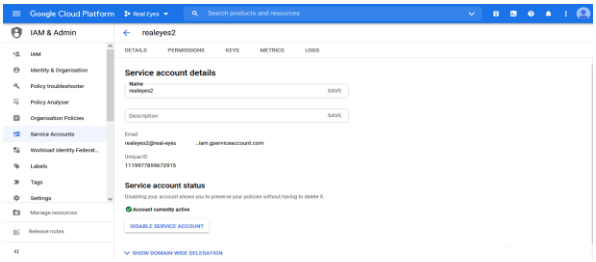
Visual 8 – Entering the virtual environment



Visual 9 – The image database used for training the system

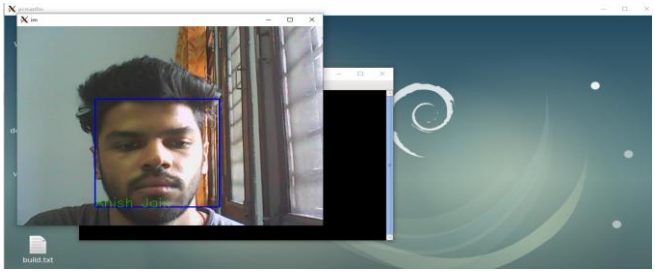


Visual 10 – The images used for training the system

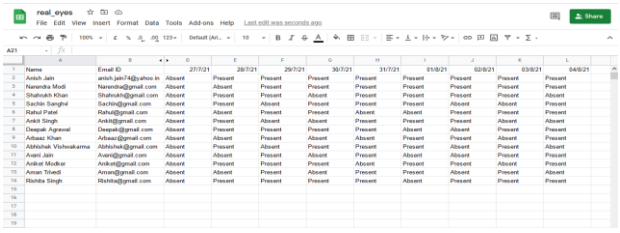


Visual 11 – Connecting Google sheets to the system

Visual 12 –Google cloud connection to the system



Visual 13 – Face recognition on the system



Visual 14 – Updating of the database

REFERENCES

[1] https://datasheets.raspberrypi.org/cm/cm3-plus-datasheet.pdf
[2] https://www.raspberrypi.org/blog/bring-on-the-documentation/
[3] https://pimylifeup.com/raspberry-pi-opencv/
[4] https://www.raspberrypi.org/blog/bring-on-the-documentation/
[5] https://www.pyimagesearch.com/2021/05/10/opencv-eigenfaces-for-face-recognition/
[6] https://www.pyimagesearch.com/2021/05/03/face-recognition-with-local-binary-patterns-lbps-and-opencv/
[7] https://docs.opencv.org/4.5.2/d3/df2/tutorial_py_basic_ops.html
[8] https://docs.opencv.org/4.5.2/d0/de3/tutorial_py_intro.html
[9] https://spin.atomicobject.com/2019/06/09/raspberry-pi-laptop-display/
[10] https://maker.pro/raspberry-pi/projects/how-to-connect-a-raspberry-pi-to-a-laptop-display
[11] https://www.raspberrypi.org/blog/use-your-desktop-or-laptop-screen-and-keyboard-with-your-pi/
[12] https://www.raspberrypi.org/forums/viewtopic.php?t=200882
[13] https://www.pyimagesearch.com/2021/04/05/opencv-face-detection-with-haar-cascades/
[14] https://www.pyimagesearch.com/2021/04/12/opencv-haar-cascades/
[15] https://www.pyimagesearch.com/2017/10/09/optimizing-opencv-on-the-raspberry-pi/
[16] https://www.pyimagesearch.com/2017/09/04/raspbian-stretch-install-opencv-3-python-on-your-raspberry-pi/
[17] https://www.pyimagesearch.com/2016/04/18/install-guide-raspberry-pi-3-raspbian-jessie-opencv-3/

[18] https://www.pyimagesearch.com/2015/07/27/installing-opencv-3-0-for-both-python-2-7-and-python-3-on-your-raspberry-pi-2/
[19] https://www.raspberrypi.org/forums/viewtopic.php?t=262144
[20] https://docs.opencv.org/3.4/db/d28/tutorial_cascade_classifier.html
[21] https://docs.opencv.org/3.4/d7/d8b/tutorial_py_face_detection.html
[22] https://en.wikipedia.org/wiki/Raspberry_Pi
[23] https://www.researchgate.net/publication/335219198_FACE_DETECTION_AND_RECOGNITION_USING_RASPBERRY_PI
[24] https://www.researchgate.net/publication/334167675_Facial_Detection_and_Recognition_using_OpenCV_on_Raspberry_Pi_Zero
[25] https://ieeexplore.ieee.org/document/8934485
[26] https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3557027
[27] https://www.geeksforgeeks.org/python-haar-cascades-for-object-detection/
[28] https://www.raspberrypi.org/documentation/remote-access/ssh/
[29] https://magpi.raspberrypi.org/articles/ssh-remote-control-raspberry-pi
[30] https://www.raspberrypi.org/documentation/remote-access/ssh/unix.md.