

A Steady Change From RFID To Facial Recognition

Manjot Singh Randhawa, Rahul .O. Vaishya, Navneet Dua

Abstract: RFID (Radio Frequency Identification) is widely used in organisations for Access management, tracking of goods and persons, contactless payment etc. RFID system can be expensive and unreliable (sometimes) but recently, we have been hearing a lot about the new facial recognition techniques. Facial recognition can be faster and more secure than RFID. In general, these work by measuring the similarity or dissimilarity between facial features of given image with faces within a database. Therefore, we aim to develop a cheap but reliable and fast access management system capable of gradually upgrading from RFID to Facial recognition, with minimum obstruction to the daily activities within an organization.

Index Terms: Face Detection, Facial Landmarks Detection, Face pose estimation, Facial recognition, Intel Neural Compute Stick 2, Artificial Neural Networks, Python, Raspberry Pi, Radio Frequency Identification

1. INTRODUCTION

THE aim of this project was to build a solution for organizations to easily and gradually upgrade from RFID to facial recognition technology. One might ask, "What is the need to upgrade?" To answer this, we must look beyond the advantages of RFID and focus on correcting the drawbacks. Disadvantages of RFID are as follows:

- Costly setup
- Problems with metal and water
- Issues with scanning multiple RFID tags at once
- Data theft from unauthorized devices

To solve these problems, different biometrics such as fingerprint, iris recognition and facial recognition are available. Out of these, Facial recognition is widely adopted as it is contactless and a non-invasive process. [1] Facial recognition as an individual process is easy to install and deploy in an organization but we wanted the process to be non-disruptive. Depending upon the number of people working in an organization, it can take from a few hours to a couple of days to set up facial recognition technology which might cause some delay in the daily activities or even leave them vulnerable and without security. In addition, some people can be skeptical about the privacy in the facial recognition technology and might take some time before getting used to it. [2] This project is a compromise between processing speed, reliability and accuracy while trying to keep the cost at a minimum. Facial Recognition is already a difficult problem but when you introduce cost and ease of installation as well acting as a factor contributing to the quality of this product, one has to tread carefully to satisfy all criteria.

2 LIMITATIONS OF STUDY

The major limitations of this project are data, time and ethnic

- *Author: Manjot Singh Randhawa, currently pursuing bachelor of technology degree program in Production and Industrial engineering in Punjab Engineering College (formerly known as PEC University of technology), India, E-mail: msrandhawa9957@gmail.com*
- *Co-Author: Navneet Dua, currently pursuing bachelor of technology degree program in Production and Industrial engineering in Punjab Engineering College (formerly known as PEC University of technology), India, E-mail: navneetdua97@gmail.com*
- *Co-Author: Dr. Rahul.O.Vaishya, Assistant professor in Production and Industrial engineering in Punjab Engineering College (formerly known as PEC University of technology), India, E-mail: rahulvaishya83@gmail.com*

bias. For Facial detection, we need datasets for training Artificial Neural Networks to predict with good accuracy. The datasets available for research and the results produced are ethnically biased. [3] There is a solution to this, which required us to make a new and well-balanced dataset and re-train the facial detection model but limited time and resources prevented us from doing so. Another issue would be human aging with time. The facial recognition system will learn to recognize individuals with good accuracy but with time, as people age, the dataset will get outdated and their facial features might change leading to inaccurate/no recognition. To tackle this, the system needs to keep updating itself with more data. Instead, every time our system cannot recognize a person with acceptable accuracy, it asks the person to wait until it collects more data, which is an inefficient method. This data collection procedure can be better and more efficient.

3 DISCLOSURE

This project does not intend to commercialize this facial recognition technology. This is purely for research purposes and serves no financial gain to any of the authors or any institution/organization. All the codes/libraries/software were either self-made or used under proper license agreements.

4 METHODOLOGY

4.1 Hardware used

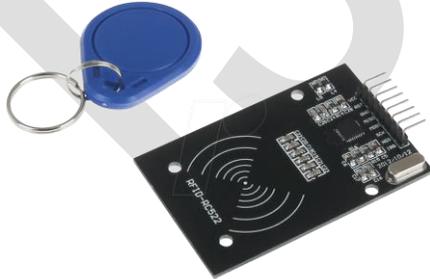
- **Raspberry Pi 3 (model B+):** The Raspberry Pi is a low cost, credit card sized computer that plugs into a computer monitor or TV, and uses a standard keyboard and mouse. [4] In our case, choosing a raspberry pi over a powerful computer was the ideal solution as it is cheap, mobile, small yet powerful device, which is versatile and reliable. Yes, it can be slow in case of image processing but the right optimizations can make all the difference.

Figure 1: Raspberry Pi 3 (model B+)

- Raspberry Pi Camera Module V2: The Raspberry Pi Camera v2 is a high quality 8-megapixel Sony IMX219 image sensor custom designed add-on board for Raspberry Pi, featuring a fixed focus lens. This camera provides good quality pictures.

Figure 2: Raspberry Pi Camera Module v2

- RFID Module based on NXP MFRC-522: MFRC522 is a highly integrated read and write chip, which is applicable to 13.56MHz contactless communications. It is a non-contact read and writer card chip, low voltage, low cost and small size is a better choice for smart meters and portable handheld devices development. [5] In our project, we use the RFID module as the sample technology that has to be upgraded as it is already in use with the organization.

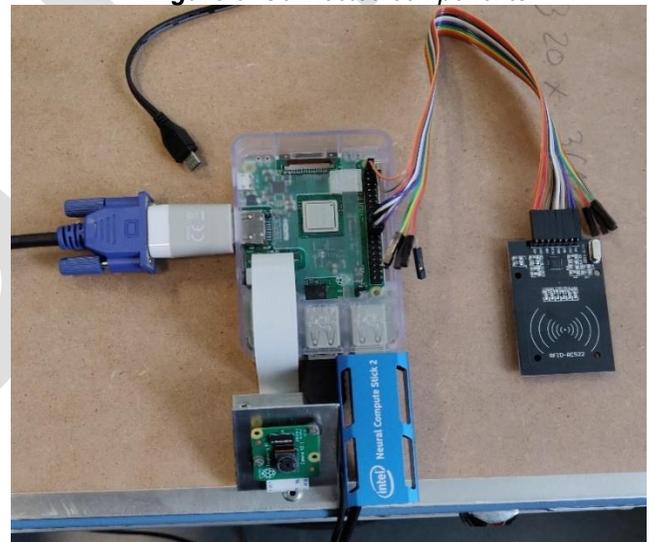
Figure 3: RFID Module

- Intel Neural Compute Stick 2: It is a USB stick that one can plug and use for accelerating machine-learning applications. [6] In our case, we use it to accelerate our Artificial Neural Networks that are run on the raspberry Pi.

Figure 4: Intel NCS 2

- Monitor to connect with raspberry pi in the initial development phase.
- Keyboard and Mouse for raspberry pi
- A remote computer available for basic neural network training and data storage (every organization in this modern age has a computer available)

4.2 Setup Preview

Figure 5: Connected components

4.3 Software/Language/Libraries/OS used

- Python 3.7.3 and its numerous libraries
- Dlib 19.17.0 (optimized for raspberry pi) [7], [8]
- Keras 2.2.4 [9]
- Tensorflow 1.13.1 [10]
- Openvino Toolkit 2019 R1
- Putty Secure Copy (PSCP)
- Raspbian Stretch and Windows 10

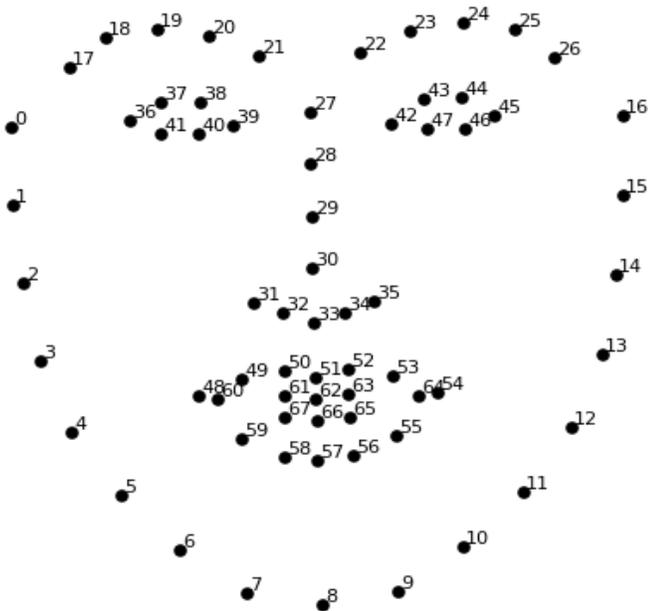
5 LITERATURE REVIEW

Object detection and recognition has been around for a few years but the human face poses even more problems than other objects since the human face is a dynamic object that comes in many forms and colors. [11] Therefore, the area of the image analyzed for a facial feature needs to be regionalized to the location with the highest probability of containing the feature. By regionalizing the detection area, false positives are eliminated and the speed of detection is increased due to the reduction of the area examined. [12] According to the author G.Hemalatha, the first step in face detection is preprocessing. They discuss the steps involved in converting an image to a pure facial image for feature extraction. These include detecting feature points, rotating to line up, Face Detection Feature Extraction, locating and cropping the face region using a rectangle, according to the face model. The various detection methods can be: [13]

- Knowledge based
- Feature based
- Texture based
- Skin color based
- Multiple features and many more.

In our case, we use a frontal face detector from Dlib based on histogram of oriented gradients (HOG) and linear SVM. Detecting a face is not enough information to create powerful applications like facial recognition. Much more information about the person's face, like position of eyes, lips, nose, eyebrows etc. is required. We use Dlib landmark 68-point detector.

Figure 6: Facial Landmark detection



A research paper by Yi Sun proposed a new approach for estimation of the positions of facial key points with three-level carefully designed convolutional networks. [14] They fused outputs of multiple networks for robust and accurate estimation. Facial landmark detection is impeded by the problems of occlusion and pose variation. Another research tries to improve detection robustness through multi-task

learning. Specifically, by combining facial landmark detection together with head pose estimation and facial attribute inference. [15] Feature extraction can be done using: [13]

- Feature Extraction using Discrete Cosine Transform (DCT)
- Facial Feature Extraction using Gabor Filter
- Facial Feature Extraction using Principle Component Analysis
- Facial Feature Extraction using Independent Component Analysis
- Facial Feature Extraction using Linear Discriminant Analysis

One way to recognize a face could be to train a neural network on the facial images of a person. However, this method would be slow for large organizations with large workforce. Even if we leave the size of an organization aside, maintaining the huge number of images or training an object detection system like YOLO (You only look once) would not guarantee good accuracy. Facial recognition cannot be solely based on an image because many factors like lighting, facial alignment, facial hair etc. can obstruct the process. Therefore, a better way would be to represent a face mathematically. The research done by Karen Simonyan on fisher vectors shows that a compact descriptor has a better recognition accuracy and is very well suited to large-scale identification tasks. [16] We use a trained ANN to convert a face into a list of 128 vectors and just save the vector file for each image. After we have sufficient training data, we train another ANN to classify between various persons by training on these vector files.

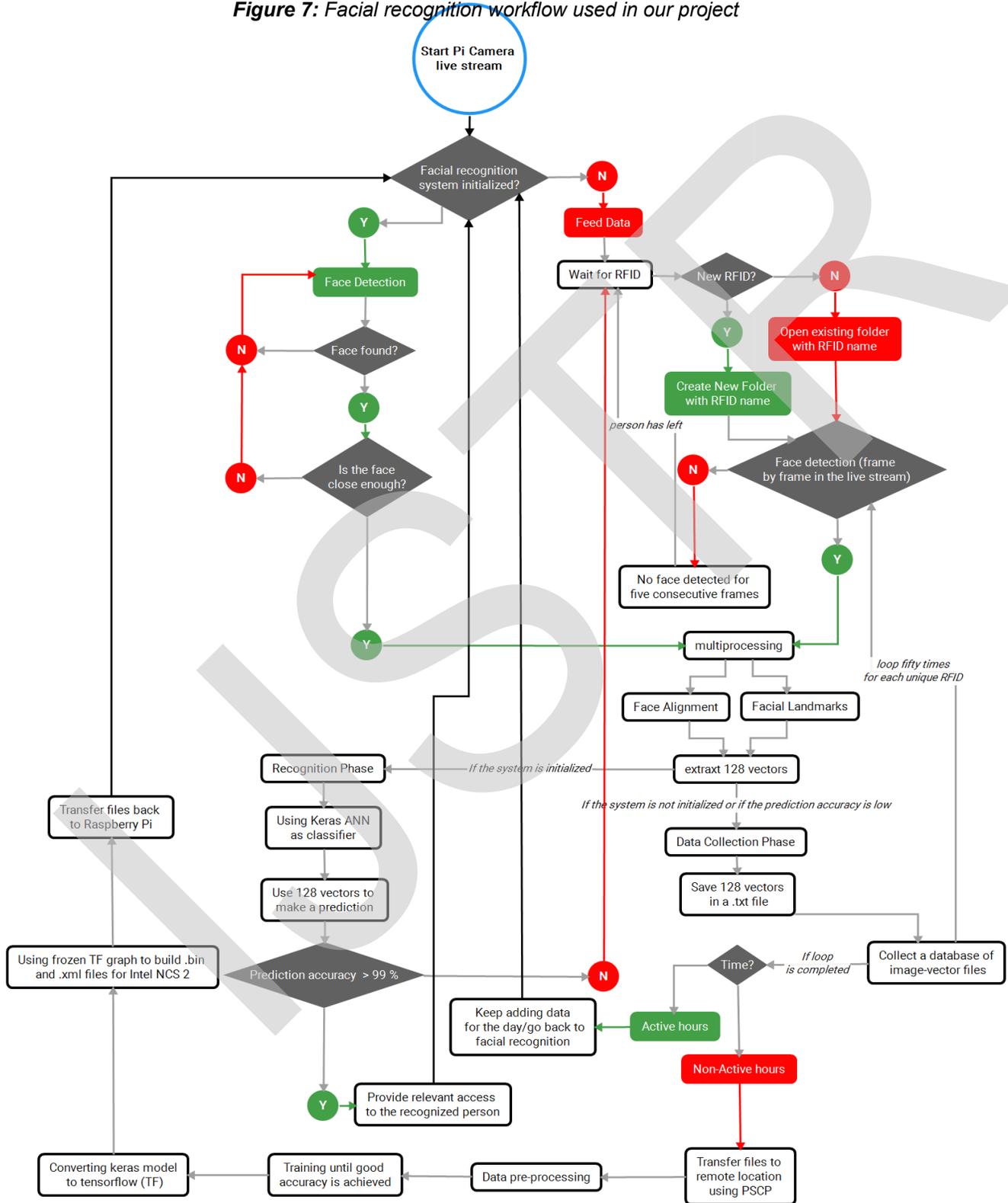
Classification methods: [13]

- Hidden Markov Model (HMM) as classifier
- Neural Networks as classifier
- Support Vector Machine as classifier
- AdaBoost as classifier

In our project, we have tried to incorporate multiple concepts at each step while optimizing them to increase processing speed on low-end machines. Many methods/platforms/libraries provide facial detection, feature extraction and facial recognition but a lot of them require high computing power as well. YOLO is a state of the art object recognition system and works well up to the detection part but is too complex for facial recognition and slow on a raspberry pi. Therefore, we have mainly used python and its libraries, considering their compatibility, ease of use and good results in our project. There is a lot of literature available on object detection and recognition and one can never read all of it.

6 WORKFLOW

Figure 7: Facial recognition workflow used in our project



6.1 Explanation of basic steps

You might have noticed Facebook's uncanny ability to automatically recognize your friends in the pictures you upload. Earlier, it used to ask you to tag your friends. Facebook does this by using facial recognition. As a human, you are good at recognizing faces and are able to do it instantly. When it comes to facial recognition using computers, we have to teach each step separately from detecting a face to recognizing a person. We build a pipeline to solve each step separately and chain multiple machine-learning algorithms together.

Steps involved in facial recognition:

- **Facial Detection:** We use the dlib Histogram of Oriented Gradients (HOG) to detect faces in an image. Before this, we remove color data from the image, as it is not required for detection. [17]

- **Posing and Projecting Faces:** Another big problem is that faces in the wild are turned in different directions, which makes the faces look different to a computer. To solve this, we use an algorithm called facial landmark

estimation to warp each image such that the eyes and lips are always in the same place. A machine-learning algorithm is trained to find sixty-eight specific points on a face; the top of the chin, the outside edge of each eye, the inner edge of each eyebrow, etc. to use for face alignment.



Figure 8: Facial Detection and Facial Landmark estimation

- **Encoding Faces:** This is the step where we can actually tell different faces apart. The easiest approach would be to compare the unknown face with the entire database of faces. This seems easy but it would take a lot of time and computing power. It is impossible to imagine a company like Facebook to recognize an unknown face by comparing it with millions of images in its database. Facial recognition needs to be done in milliseconds to be of any use in

real time. The other method could be to extract measurements from a face and use them to compare faces. A computer does not understand measurements the same way a human does. Therefore, according to many researchers, it is best to let a computer figure out the measurements itself using a deep convolutional network. [18] The 128 measurements (vectors) extracted using this algorithm are used to differentiate between faces.

- **Classification:** This is the easiest step in this project. A feed-forward ANN that was built in keras was used to train on the 128 measurements and predict the identity of the person with at least 99 percent accuracy.

6.2 Explanation of workflow in our project

- As soon as we start our facial recognition system, raspberry pi camera live stream is started. This stream supplies us with the required images for facial detection. This live stream is used via OpenCV, which can be slow and lag at times. So, to counter this, we use stream in async mode along with multithreading to increase frames per second (FPS).
- Each frame captured by our webcam is deprived of color data to reduce computations, and because color data is useless for facial detection.
- Next, we check if our system is initialized, i.e. check if this is the first time, the facial recognition system is running or if it has run before. We check this by searching for weight files in our folder. This works because in order to predict the identity of a person using a ANN, weight files are used. These files are present only if our system has been initialized and run a full cycle. Otherwise, we can safely assume that the system is not initialized.
- If the system is not initialized, we ask the user for data (pictures of the face). As soon as the person uses his RFID, the data extracted from RFID is used to see if the person already exists in the database or not. After making either a new folder or opening an existing folder for data storage, we perform face detection.
- Face detection is done on each incoming frame. If a face is detected, then we utilize all the four cores of raspberry pi (multiprocessing) to speed up the process of facial landmark estimation and facial

File Edit Format View Help

```
[-0.08808450,0.07164350,0.0805757,-0.0616803,-0.13
0.0373954,-0.09089330,0.20283,-0.1473310,0.152384
0.00118178,-0.03645920,0.166395,-0.161866,-0.139
0.0286843,-0.02779560,0.07041320,0.0650831,-0.085
0.0699758,-0.06628450,0.103468,-0.061148,-0.0606
0.04670420,0.004589780,0.0721582,-0.0736932,-0.10
-0.256772,-0.05397080,0.2116280,0.06634160,0.1848
0.1494440,0.114355,-0.3160480,0.05697710,0.1340450
0.0885895,-0.198954,-0.01259920,0.216519,-0.1782
0.0221611,-0.0659539,-0.0432650,0.1711920,0.12428
0.260494,-0.1636860,0.04192310,0.180948,-0.022359
0.009356240,0.046190,0.141723,-0.1198450,0.013044
```

Figure 9: Sample, 128 measurements

alignment. If we are unable to detect a face for five consecutive frames, we assume that the person has left and go back to the step where we wait for RFID.

- We now extract 128 measurements from each incoming frame and save these measurements in text file in their respective folders.
- Up until now, we detected a face in a single frame, estimated facial landmarks, aligned that picture and save measurements. However, to train a neural network to recognize a single face, we need more data. Therefore, all the steps from detection to measurement extraction are done for at least 50 times so we have enough data.
- After collecting data, we again go to the initial step and wait for RFID to gather data from other individuals.
- After we reach non-active hours in an organization (time when that organization is least active), we transfer all the measurements to a remote computer using PSCP.
- We pre-process the data and make a single file containing all the data (MS excel) in a format that is acceptable to the neural network.
- A feed forward ANN is trained until good accuracy is achieved on the data.
- After the completion of training, we save the model and weights obtained from keras ANN. (.h5 format)
- Next, we use the saved weights and model files to build a frozen tensorflow graph. (.pb format)
- Now, the frozen graph is given as input to the model optimizer and the output is a bin, mapping and an xml file that is transferred back to raspberry pi using PSCP. These files are used to predict the identity of the person.
- It is possible to use the weights and model files from keras ANN to make predictions but we go through this complicated procedure of converting them to bin, mapping and xml files so that we can make predictions using Intel NCS 2. Intel NCS 2 is much faster at making predictions and also decreases the load on raspberry pi.
- Now, the next time facial recognition system starts, it is initialized and we start detecting faces in the live stream.
- We need to detect faces that are within a certain range and not every face that passes by. We put a limit on the range by varying the resolution and region of interest. Decreasing the resolution decreases the range up to which a face can be detected along with increasing computation speed. Every time a face is detected, we draw region of interest around the face. We extract the coordinates of the rectangle, find the length and breadth of the rectangle. If the length and breadth satisfy the minimum criteria manually set by us, we assume that the face is within range for face detection and move on to further steps. Else, we keep looking for a face.
- The next step is facial landmark estimation and alignment by utilizing all the cores on a raspberry pi. The 128 measurements extracted now are not saved, instead used to make a prediction for the unknown face with the help of Intel NCS 2.
- If the prediction accuracy is more than 99%, we

consider the prediction to be good, provide relevant access.

- In case, the accuracy is not good, it could be that this is a new person or the data obtained from an existing person was not good enough. Therefore, we ask the user to use his RFID and provide more data.
- Every day, we transfer the new data to remote pc, create new prediction files and transfer them back to the raspberry pi.

6.3 Results

All the above-mentioned steps are automated. You just have to start the system once, the remaining steps are automatic, and no human intervention is required. We were able to detect a face every other second and recognize a face every two seconds on a raspberry pi. This speed on a small device like raspberry pi is remarkable and barely satisfies the criteria for its implementation in organizations.

6.4 Future Improvements

The major improvement could be addition of security features to differentiate between a real human and a video/image. This could be done by using a thermal camera, detecting blinking, detecting skin texture etc. Another improvement could be to do memory-consuming processes like face detection, landmark estimation, and measurement extraction on Intel NCS 2. This would speed up the entire process a lot.

7 CONCLUSION

We have explained and built a solution for organizations to gradually upgrade from RFID to facial technology. This project still lacks security features as it cannot differentiate between a picture and real human. Necessary optimizations were used to increase the speed of the entire process. There is scope for a lot of improvement.

ACKNOWLEDGMENT

The authors wish to thank Davis King, for providing dlib as a tool to allow researchers like us to use it. In addition, we are grateful for the guidance of our professor, Dr. Rahul. O. Vaishya. We value the efforts done by our college, Punjab Engineering College (deemed to be university) to motivate students like us to pursue research.

REFERENCES

- [1] Lei Zhang, Meng Yang, Xiangchu Feng, Yi Ma, David Zhang, "Collaborative Representation based Classification for Face Recognition", Lancaster: DEStech Publications
- [2] K.W. Bowyer, Dept. of Comput. Sci. & Eng., Notre Dama Univ., USA, "Face recognition technology: security versus privacy", Publisher: IEEE, Volume: 23, Issue: 1
- [3] Adam Kortylewski, Bernhard Egger, Andreas Schneider, Thomas Gerig, Andreas Morel-Forster, Thomas Vetter, "Analyzing and Reducing the Damage of Dataset Bias to Face Recognition with Synthetic Data", Publisher: IEEE
- [4] Website (Official Raspberry pi website), link: <https://www.raspberrypi.org/help/what-%20is-a-raspberry-pi/>
- [5] Instruction Manual, RFID Module based on NXP MFRC-522, link: <http://anleitung.joy-it.net/wp-content/uploads/2016/09/SBC-RFID-RC522-Manual.pdf>
- [6] Website (Intel NCS 2), link: <https://software.intel.com/en-us/neural-compute-stick>

- [7] Website (Dlib C++ Library), link: <http://dlib.net/>
- [8] Website (Dlib Github), Link: <https://github.com/davisking/dlib>
- [9] Documentation (Keras), Link: <https://keras.io/>
- [10] Documentaton (Tensorflow), Link: https://www.tensorflow.org/versions/r1.13/api_docs/python/tf
- [11] Muller, N., Magaia, L. and Herbst B.M. Singular value decomposition, eigenfaces, and 3D reconstructions. SIAM Review, Vol. 46 Issue 3, pp. 518–545. Dec. 2004.
- [12] Phillip Ian Wilson, Jon Fernandez, “Facial feature detection using Haar classifiers”, Published in Journal of Computing Sciences in Colleges archive, Volume 21 Issue 4, April 2006 , Pages 127-133
- [13] G.Hemalatha , C.P. Sumathi, “A Study of Techniques for Facial Detection and Expression Classification”, Published in International Journal of Computer Science & Engineering Survey (JCSES) Vol.5, No.2, April 2014
- [14] Yi Sun, Xiaogang Wang, Xiaoou Tang, “Deep Convolutional Network Cascade for Facial Point Detection”, The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2013, pp. 3476-3483
- [15] Zhan Peng Zhang, Ping Luo, Chen Change Loy, Xiaoou Tang, “Facial Landmark Detection by Deep Multi-task Learning”, European Conference on Computer Vision (ECCV) 2014: Computer Vision – ECCV 2014 pp 94-108
- [16] Karen Simonyan, Omkar M. Parkhi, Andrea Vedaldi, Andrew Zisserman, “Fisher Vector Faces in the Wild”, Conference: British Machine Vision Conference 2013
- [17] O.Déniz, F. De la Torre, G. Bueno, J. Salido, “Face recognition using Histograms of Oriented Gradients”, Pattern Recognition Letters: Volume 32, Issue 12, 1 September 2011, Pages 1598-1603
- [18] I.Craw, H.Ellis, J.R.Lishman, “Automatic extraction of face-features”, Pattern Recognition Letters: Volume 5, Issue 2, February 1987, Pages 183-187