Investigating whether a Raspberry Pi executing facial recognition software can be used to securely identify a persons face and be integrated into a home to open a door lock without the need for a physical key.

Matthew Micallef
Institute Of ICT
Malta College Of Arts,
Science & Technology

Abstract – People across the world live in some form of house, from an article written by Architecture and Design, as of 2021 there are approximately 2.3 billion houses around the world [1]. Surely the owners of these houses have been in a situation where they have either lost their key or simply forgot it and where unable to unlock the door to enter their house. This study aims to investigate and implement a cost effective way of using a Raspberry Pi 4 in conjunction with the official Raspberry Pi camera to keep the cost as low as possible, as a means of training a model with the owners face, and executing a facial recognition algorithm to identify the face in front of the camera, if the algorithm returns a high confidence rate it will unlock the door and allow the user to enter his house. There are already products like this on the market but there are drawbacks from adopting these, firstly they are expensive they can range anywhere from €300 to €2500, secondly the adopter would need to thrust the brand and the implementation and assume that this solution is being done in a secure way.

I. Introduction

A dwelling is an individual's primary anchor in the environment. It may serve many functions, such as shelter, privacy, security and status [2]. Dwellings are one of the most important possessions to any individual in the world, the dwelling will contain all his possessions, might contain relatives' or family members and so much more. Having a secure dwelling has been of utmost importance with the inventions of locks, alarms, Closed-Circuit television (CCTV) cameras and many more devices aimed to keep the dwelling secure.

There are drawbacks to having conventional locks on the door to your house, this is that you always need a physical key, the drawback of this is the chance of loosing it or forgetting it and not being able to enter your house which in turn would create a problem for the homeowner.

This study aims to develop a low-cost solution to automatically train a model based on the owner's face and from their after automatically recognise his face and allow him to enter his house. By using this system people will no longer have to remember to carry their keys or have the problem of loosing them as their face will become the new key and will always be on their person.

II. Literature Review

The problem for this study arose when a relative was stuck outside their home and had to wait for another individual to use their key to open the door, hypothetically if the homeowner had the key and no other relative or individual had a key, the homeowner would have been locked out and would have had to call a locksmith to open the door, which begs the question, are traditional key lock doors secure? This is not a part of the scope of the research paper.

Using facial recognition for the purposes of authentication is not a new concept it has been used in multiple mobile devices and also laptop devices where they use either of the following 2 methods, the first method uses infrared to extract data from the iris which due to its complex pattern can contain many distinctive features such as arching ligaments, furrows, ridges, crypts, rings, corona, freckles, and a zigzag collarett. With the above mentioned features the iris has a great mathematical advantage that its pattern variability among different individuals is significant [3]. The second method is optical based which is what will be carried out in this research paper, where a stream of images are taken using a camera and when a face is detected it will then proceed to checking whether or not it is found in the previously trained model and if it is confident that it is the same face It will allow the user to be authenticated.

III. Research Methodology

The hypothesis for this implementation, would be attainable with the model being retrained each time a new user needs to be added to the authorized list of people who are given access to the household, given the hardware that will be used the confidence levels might be low, this is because certain image augmentation techniques like kernel filtering to sharpen blurry images cannot be feasibly implemented due to the lack of performance, so further images for training the model might be needed.

To address this hypothesis these research questions where identified.

1) What kind of data set is required for detecting faces?
2) What algorithm can be used to best distinguish between faces?
3) Is there any difference in the confidence levels when increasing the number of images taken for each new individual?
4) Is it safe to adopt this technology for the above mentioned use case?

The following phases will be used to implement the research paper.

Phase 1: Implement a method where the user enters his name, and several images are taken automatically using the PI camera.

Phase 2: Train the model with the previously taken photos of him.

Phase 3: Start using the model to recognize who is in front of the camera.

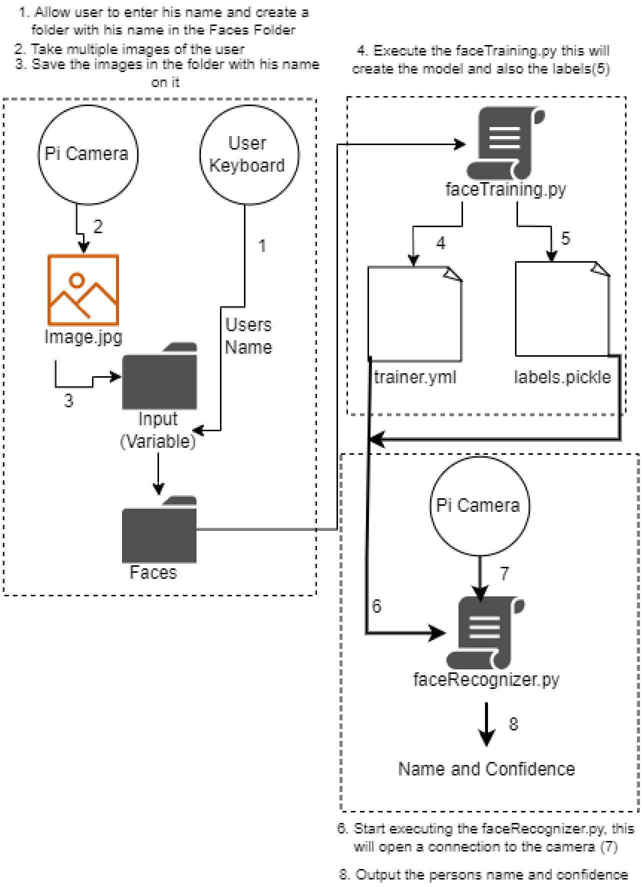
The flow for the development of the implementation can be seen on the other side of the column (Figure 1).

A) Data Set

A data set in the case of this implementation is not needed this is because the model will be trained on users that are allowed to enter the house. As stated in the above phase plan each time a new user is added a number of images are taken and saved based on whether a face is visible or not. Once the number of images taken of the user is reached the model will be re-trained on the new given images and trained on any other previous users given that the images for this person are not being deleted. This functionality can be easily implemented, when a user no longer has access to the household the users' images are deleted and the model is retrained and thus the deleted user will no longer have access as the model will no longer recognise the user. This ultimately means that the data set will increase and decrease each time a user is added or removed.

B) Face trainer.

Haar classifier will be used to identify whether the given image has a face in it or not, this algorithm was chosen because of its speed, high detection accuracy and low false positive rate. Haar classifier is an algorithm created by Paul Viola and Michael Jones, which was trained on various images containing faces and various images without containing faces [4]. Another approach would be to use additional classifiers in conjunction with the Frontal face Haar classifier [5], these would be based on skin hue histogram matching, eyes detection and mouth detection, this approach would be to further reduce the low false positive rate of the Frontal Face Haar Classifier, were the image would first go through this classifier and then be processed by the skin hue histogram matching classifier next the image would be processed by the eye detecting classifier and finally go through the mouth detection operation to continue decreasing as much as possible the false positive rate from the original Classifier.



1. Allow user to enter his name and create a folder with his name in the Faces Folder
2. Take multiple images of the user
3. Save the images in the folder with his name on it

4. Execute the faceTraining.py this will create the model and also the labels(5)

6. Start executing the faceRecognizer.py, this will open a connection to the camera (7)

8. Output the persons name and confidence

Figure 1: 3 Phase Plan

C) Facial recognition.

The Open CV library was chosen because of it being open source, other computer vision libraries include Microsoft Computer Vision API, Amazon Rekognition, Google Cloud Vision API and Azure Face API. Open CV has three built in facial recognition algorithms, these are the Eigenfaces, Fisherfaces and Local Binary Pattern Histogram (LBPH). Comparing the three algorithms together the LBPH algorithm can not only recognise the front of the face but can also recognise the side of the face which makes it slightly better than the others [6]. Thus, the LBPH algorithm will be handling the facial recognition for the proposed implementation.

The LBPH algorithm is simple and very efficient, this algorithm works by labelling pixels of an image by thresholding the neighbouring of each pixel and considers the result as a binary number, then converts this number into a decimal number and that decimal number is now the new value of the centre pixel [7]. A representation of how the LBPH algorithm works can be seen on the below image (Figure 2) [8].

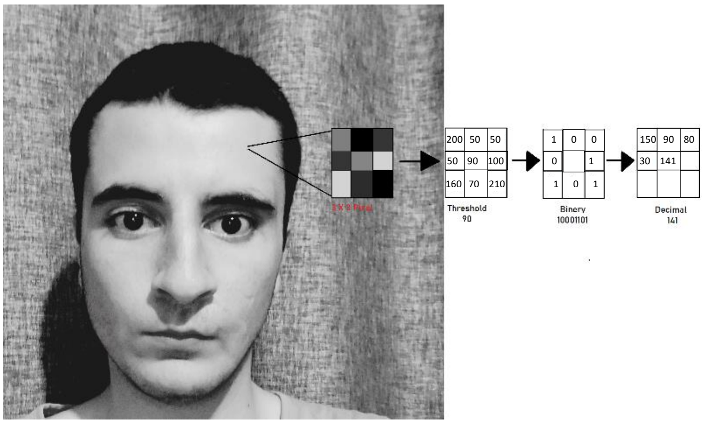


Figure 2: Conversion Of a Grayscale image to decimal

D) Number of images.

With regards to the number of images that will be taken of each user for the model to be trained on, A study will be conducted after both the trainer and the facial recogniser are implemented, to identify whether increasing the number of images that the model is trained on has an effect on how accurate the identification of a face is as the number of images can be altered by means of a variable, and the model can easily be retrained in a short period of time.

IV. Results.

The implementation followed the pipeline above where the user enters his name, a folder is created using his name and several images are taken and saved dependent on if the Haar classifier detects a face in the image. This algorithm worked flawlessly and from testing carried out by observing each image saved, no false positives were `observed, and all the images saved to the folder contained a face. During the implementation of the pipeline no other observations were seen given that a significant amount of research was carried out before hand, during the research methodology stage.

An observation was made that when the image is in grayscale both the Haar Classifier and the LBPH algorithm performed better. With this in mind the image was converted to grayscale before the Haar Classifier processed the image, and before the LBPH algorithm was trained on the images to produce the model, and even from the live feed when ultimately the facial recognizer was being used to recognize images that were being captured by the PI camera, through a video stream at the default 24 frames per second that the raspberry pi can capture.

After the images were taken the LBPH algorithm was used to train a model and create both a .yml file which is a highly efficient serialized document and a .pickle file, the .yml file contains the actual binary date that is used to identify a face and the .pickle file which allows objects to be serialized to files on disk and deserialized back into the executing program at run time, this file will contain a byte stream with all the names for all the faces while also containing some binary data, that is used to match the name to the image.

The research questions for this paper, where identified during the research methodology stage, with regards to what kind of data set was needed the implementation used a variable data set which means that a model will be built based on the user, so if for example only one user will be using this implementation the model will only recognise his face, if then the user had to add another user to be authorised to enter the household the model will be retrained based on the new images of the new user and also images of the old user. with regards to what algorithm is best suited to recognise faces the LBPH algorithm was chosen between the other algorithms after carrying research and analysis from previous studies carried out by other individuals, respectively cited in the research methodology phase.

With regards to testing the number of images that will be needed to train the model and have it accurately distinguish between the users, the following testing methodology will be carried out, the model will firstly be tested with one subject containing one image and an average of the confidence levels of 10 frames gathered from the video stream will it be tallied out and the result will be presented in table one attached below. Further testing would be carried out by increasing the number of images to 10 for one subject and this methodology would continue by increasing the number of images the model would be trained on in increments 10, 20, 40 for one subject. another case that the implementation would be tested on would be with five users containing the same number of images as the previous test case.

Test case	Subject	Number of images per subject	Average of the confidence level for 10 frames
1	1	1	68.28
2	1	10	81.973
3	1	20	87.40
4	1	40	88.89
5	5	1	84.13
6	5	10	67.24
7	5	20	75.70
8	5	40	68.32

Table 1 Results from the test cases above

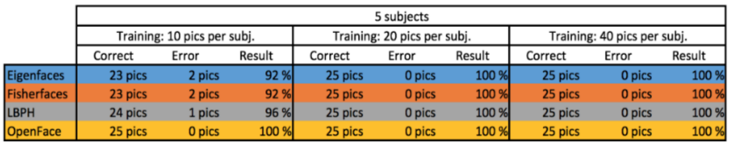
Carrying out test case one, the Haar cascade classifier worked flawlessly identifying only the face throughout all the test cases, comparing the confidence levels between test case one and test case two the confidence level can be seen to be increased by an average of 13, this means that increasing the number of images from one to ten is beneficial. The increase in confidence level can also be seen across test case three and test case four, the improvements between test case three and test case four are not significant give in that the confidence level only increased by 1.3. This means that having 40 images to train the model is not necessary and having 20 images is enough to confidently identify the face, this would help in decreasing the amount of storage space needed for each user while also decreasing the time the model takes to train.

The next four test cases were carried out by having an additional four people in conjunction with myself to train the model simulating five users, the results are as follows.

The highest confidence rate was at test case one some misidentification was present meaning that it thought that I was someone else, but the confidence rate was higher than that of test case one. Test case six shows that when each user has 10 images to be trained with, the confidence level drops significantly, there was also a significant increase in the number of misidentifications this result was not as expected and was executed again to see weather or not it was just an error in the implementation, but this did not change the result and the confidence level remained approximately the same.

Moving on to test case seven the confidence rate was higher than that of the previous test case and the misidentification was also lower, test case eight produced a result like that of test case six this, was also not expected, misidentification also increased again. The last three test cases six, seven and eight were re-executed and the same result prevailed. From the analysis of the test cases carried out having 20 images for each user is better than having more images as can be seen both by the first set of test cases and by the second set off test cases.

A study previously carried by Delbiaggio [9], similar testing comparing the error rate of the number of mis identified faces through the four most common facial recognition algorithms including the LBPH algorithm he concluded that having 10 images per subject has a higher misidentification rate then having 20 or even 40 images per subject. comparing the results from my study to this study carried out, a difference can be observed, Delbiaggio's study was tested using five users as can be seen from table two attached below his results from training the LBPH using 20 and 40 images per user produced no errors in identifying each user although I have no quantitative results comparing these similar studies but through, observation some misidentification was observed having the model predict that I am someone else.



	5 subjects								
	Training: 10 pics per subj.			Training: 20 pics per subj.			Training: 40 pics per subj.		
	Correct	Error	Result	Correct	Error	Result	Correct	Error	Result
Eigenfaces	23 pics	2 pics	92 %	25 pics	0 pics	100 %	25 pics	0 pics	100 %
Fisherfaces	23 pics	2 pics	92 %	25 pics	0 pics	100 %	25 pics	0 pics	100 %
LBPH	24 pics	1 pics	96 %	25 pics	0 pics	100 %	25 pics	0 pics	100 %
OpenFace	25 pics	0 pics	100 %	25 pics	0 pics	100 %	25 pics	0 pics	100 %

Table 2 Delbiaggio's results.

The difference in results Is most probably due to the fact that Delbiaggio used a very high quality camera the Nikon D3100, to take his images for each user. The same study does not specify what hardware that was used to execute the facial recognition software on, be it a computer or a system on chip (SOC) like the Raspberry Pi being used in this project.

From the two above mentioned anomalies I believe that this is where the difference in results came from, since I am using an SOC which is a low powered computer and an official Raspberry Pi camera which compared to the Nikon camera used in the cited study, does not compare with regards to quality of the images being taken. Also, the study was given still images to recognise and not a video stream like my implementation.

An observation that came from a vulnerability given that this application would be used as a security feature to allow homeowners into their home. The following test case was carried out, A picture was taken off myself using a mobile device the image was then displayed to the camera of the Raspberry Pi while it was executing the facial recognition software end two observations were made, the first observation was that it still thought that it was a person even though it was a photo displayed from on the phone, the second observation was that the confidence rates were significantly higher averaging from around 100 to 110. This would prove a significant security risk given that anyone can take a picture of you or pull it off social media and use it to enter your house. Answering research question four unfortunately the implementation as it is, is not secure enough to be adopted for the aforementioned use case.

More research would be needed, to identify if there is a way to distinguish between a real face in front of the camera or an image of the face displayed through a screen or printed photo. If the vulnerability was to be fixed and the implementation is accurate as can be, with no false positives and working as intended more testing would be needed to identify weather the ambient light would affect the recognition, given that the user can enter his house at night or the user can enter his house with the bright sun beaming behind him. These tests would need to be carried out before implementing the system as to eliminate any security problems and to also eliminate the chance of the user still being stuck outside of his house due to the facial recogniser not recognising his face because of the surrounding conditions.

V. Conclusion

The implementation was carried out following the phase plan mentioned during the research methodology stage, the implementation was implemented successfully although as discussed above the results were not as expected, this could have been due to several reasons with regards to limitations, better hardware would have been chosen although the chosen hardware was chosen for the reason of keeping the implementation as cheap as possible, but mainly due to the camera the results were skewed. Time constraints also played a significant role, due to the limited amount of time that was present other algorithms could not have been tested.

Regarding future improvements apart from the ones mentioned in the results section of this research paper a better camera could have been used, also research would be carried out to identify whether a deep convolutional neural network would have been better suited for this use case.

Although the implementation as is cannot be implemented, future improvements could be made, and a cheap and affordable product using this software could be released to market due to the potential it has amongst other more highly priced products.

V. References

[1] Architecture & Design. 2022. How Many Houses are in the World? | Architecture & Design. [online] Available at: <https://www.architectureanddesign.com.au/features/list/how-many-houses-are-in-the-world> [Accessed 13 May 2022].

[2] Henny Coolen (2006) The Meaning of Dwellings: an Ecological Perspective, Housing, Theory and Society, 23:4, 185-201.

[3] Daugman, J., 2009. How Iris Recognition Works. *The Essential Guide to Image Processing*, pp.715-739.

[4] A. Priadana and M. Habibi, "Face Detection using Haar Cascades to Filter Selfie Face Image on Instagram," 2019 International Conference of Artificial Intelligence and Information Technology (ICAIIT), 2019, pp. 6-9.

[5] L. Cuimei, Q. Zhiliang, J. Nan and W. Jianhua, "Human face detection algorithm via Haar cascade classifier combined with three additional classifiers," 2017 13th IEEE International Conference on Electronic Measurement & Instruments (ICEMI), 2017, pp. 483-487,

[6] XueMei Zhao and ChenBing Wei, “A Real-time Face Recognition System Based on the Improved KBPH Algorithm”, 2017 IEEE 2nd International Conference on Signal and Image Processing.

[7] Suma, S.L. and Raga, S., 2018. Real time face recognition of human faces by using LBPH and Viola Jones algorithm. International Journal of Scientific Research in Computer Science and Engineering, 6(5), pp.6-10.

[8] A. M. Jagtap, V. Kangale, K. Unune and P. Gosavi, "A Study of LBPH, Eigenface, Fisherface and Haar-like features for Face recognition using OpenCV," 2019 International Conference on Intelligent Sustainable Systems (ICISS), 2019, pp. 219-224.

[9] Delbiaggio, N., 2022. *A comparison of facial recognition’s algorithms*. [online] Theseus.fi. Available at: <https://www.theseus.fi/handle/10024/132808>