**USER MANUAL**

**STEPS TO RECREATE**

In this section, I will run through how to recreate the output with the source code provided.

Please note, I believe that the code will work on any device, as long as, you have installed the camera first, and then OpenCV. I don’t believe this code is restricted to one version of python or OpenCV.

However, It is very, very temperamental and there are a lot of articles with 1000’s of comments with users with various issues.

It is a very fragile application to set up and get working.

As I tested several different ways, I don’t know if all the libraries I installed were needed to accomplish the output. I will not include the bulk of how to install OpenCV here, as this is not what my project is on.

To best recreate the results, I have a Raspberry Pi 3 B+ running NOOBS that I set up about 1 year ago and updated. I run python 3.53 and I used the pip install method for OpenCV.

I don’t know if method this will work on a newer version of Raspbian, but there are 100’s of tutorials for installing this. I will include what I found to be the most helpful tutorial for installation.

I can also provide my tracking document notes if requested. It is around 3600 words of notes broken down into steps and stages of me testing and my process – it is loosely formatted, but contains the whole sequence of each stage, but has around half of the document in me testing and progression for things that ended up not working. However, if more insight is needed, Happy to provide.

Now, onto the installation steps using the source code.

The testing phase and the brake down gives the overview of the process sequence that I followed. For this section. I will provide details on the scripts that you need to run, in order, and parts you need to change for your own set up.

**INSTALLATION STEPS**

• **Step 1 – initial set up**

Ensure you have your Raspberry Pi B 3+ uptodate.

Argon is connected to WIFI.

Connect the camera to the Pi, enable and install the camera module.

Install Particle CLI and curl on the Pi. Enable SSH and reboot.

Connect the 3 LED’s and servo to the breadboard and Argon pins as per the script (or change them to your liking)

Copy and flash the “Servo” code to the Argon.

The servo position of the hand (I used the propeller shaped one(dual ends)) should be 90 degrees to the actual servo to form a X and the red LED should be on.

**• Step 2 – Communication between devices**

Check the communication to the Argon with CURL.

Create a new API token for the Argon.

replace APIKEY with your new API token and DEVICEID with your Argon device id

In Pi terminal, run these 2 curl commands – one at a time and let the sequence play out before running the next curl command.

curl -H "Authorization:Bearer APIKEY" https://api.particle.io/v1/devices/ DEVICEID /door -d arg=granted

curl -H "Authorization:Bearer APIKEY " https://api.particle.io/v1/devices/DEVICEID/door -d arg=denied

The argon responds to the granted flag and should rotate the servo 90 degrees, turn the red LED off, the green on, for 10 seconds, orangs flash and then rotate back to locked and LED back to red.

The denied curl should flash the red LED.

Confirm this is working first.

**• Step 3 - Folder structure**

The folder structure for the project is important.

You should have a main folder “Facial Recognition”

this main folder will contain the 3 python scripts and the haarcascade\_frontalface\_default xml document.

Then 3 sub folders called.

“dataset”- contains the data set photos from script 1

“denied” – contains the image from denied results to emailed

“trainer”- contains the recognizer output from script 2

import the python code and save it into the main folder.

when running the scripts from the terminal. Run them from the main folder – ie “…/FacialRecognition/$ python3 faceRec.py”

Don’t run them yet as need to set up other things fist

**• Step 4 - Set up email**

To set up the email, set up a quick Gmail account ( if you do not want to use your own Gmail account(I did not test other email providers)

ensure that the password is ONLY numerical and characters NO special characters.. Pi doesn’t like this for some reason.

Change setting in Gmail to “allow less secure apps”

Install the following modules on the Pi

- mpack, ssmtp and mailutils

Modify the config file by running

sudo nano /etc/ssmtp/ssmtp.conf

and change the following:

root=postmaster

mailhub=smtp.gmail.com:587

hostname=raspberrypi

AuthUser= TheGmailEmail@gmail.com

AuthPass=TheGmailPassword

FromLineOverride=YES

UseSTARTTLS=YES

sudo nano /etc/ssmtp/revaliases

root: TheGmailEmail@gmail.com@gmail.com:smtp.gmail.com:587

pi: TheGmailEmail@gmail.com@gmail.com:smtp.gmail.com:587

After setting this up, create a quick text file ( or use any tile) and send a test email from the terminal using the command

mpack -s "Test" /home/pi/AA.txt deakinEmail@deakin.edu.au

“Test” is the subject line, and AA.txt is the file being sent – change this to suit your file and Deakin email

**• Step 5 – OpenCV installation**

This is a critical step and depends 100% on the hardware and software you are using. As stated above, this SHOULD work on any version of OpenCV installed.

Installing OpenCV is a project on its own, it is a long step to set this up and I can’t stress enough about the importance of reading and following the article in full to every detail before proceeding. This may take a couple hours to set up…. Or a few days..

The article I used to install OpenCV using the pip method is:

https://www.pyimagesearch.com/opencv-tutorials-resources-guides/

That is the main page with different installation methods and the pip article is

https://www.pyimagesearch.com/2018/09/19/pip-install-opencv/

Please note. Read these articles VERY carefully. The pi section is near the end of the article.

There are a lot of steps depending which version you are running.

Hopefully, you have a Pi 3B+ running NOOBS previously set up as, (after much trial and error with attempting the source code method)

The pip installation article works very well.

I installed the OpenCV-contrib-python package. Option 2 – which is also the recommended option.

I also did NOT run this in a virtual environment, which is also recommended.

BEFORE YOU INSTALL OPENCV! Make sure the camera module is installed and enabled.

The guide has specifics to your set up.

The main commands are\*\* Do not just follow these steps, please read the installation guide:

$ sudo apt-get install libhdf5-dev libhdf5-serial-dev libhdf5-100

$ sudo apt-get install libqtgui4 libqtwebkit4 libqt4-test python3-pyqt5

$ sudo apt-get install libatlas-base-dev

$ sudo apt-get install libjasper-dev

$ wget https://bootstrap.pypa.io/get-pip.py

$ sudo python3 get-pip.py

$ pip install imutils

I installed this to the python global site packages. It is RECCOMENDED NOT to do this and work from a virtual environment in the article.

I chose installation option A

Please also note that the Version of OpenCV may change.

You may have to play around with the == aspect of the below command. Or remove it.

$ sudo pip install opencv-contrib-python==4.1.0.25

If no error messages, the installation takes around 5-10mins.

This then should now be installed.

From the article you can download some test code – I did, to run an test to ensure it is working correctly. There are more steps to follow this and when running the script from the terminal with the flags, I removed the “/” to make it work.

You do need to enter an email address to obtain the code, but, if you want to trouble shoot, do this.

Run the code from the folder and it should show a live feed of the camera detecting your face.

**• Step 6 – Setting up the facial Recognition Dataset gathering**

Now that OpenCV is working, it is time to gather the data and train the recognizer.

\* You can also run both of these scripts through SSH.

Ensure good lighting for your camera, preferably stable so you don’t have to hold the camera.

Also, have a picture on your phone, like a selfie picture of someone else (or if you have someone else in the house with you)

Run from the main folder containing the script

python3 faceDataset.py

For each person you want to add ( I only did 2 1 for Granted and 1 for Denied) (I added more but 2 is all that is needed)

enter 1 at the prompt for the first person and look into the camera (for yourself)

then enter

2 for the next person and hold the picture on your phone to the camera.

It will take 30 photos of each – is fairly quick.

Check the photos (Saved in the dataset file) to ensure they are mainly clear and not blurry etc. They should all be in grey.

hit ESC to exit.

When you are happy with the pictures it is time to train them with machine learning into the recognizer.

Run the trainRecogniser.py script

It will automatically train the trainer on the dataset just captured. If you have used 2 faces, it will show an output of 60 photos trained and output the trainer.yml file to the trainer folder.

Now the dataset and recognizer are complete, and we can move onto the last parts.

**• Step 7 – Facial Recognition script modification**

In this last step, you will need to modify some lines of the faceRec.py script to include your changes.

Change the names[] list, leave ‘none’ and change ‘David’ to your name (this is for granted) and change ‘Neo’ to the other face you used, this will be for the Denied person.

Then update the names in the IF statements in the bulk of the script, where if id == ‘David’: - change this to your name – this is for the GRANTED access, which will unlock the door.

change the ‘Neo’ if statement to the denied person.

Change all of the curl commands to include your API and device IDs from before. There are 3 curl commands at each IF, ELIF and ELSE statements.

Change the email mpack -s command to your command as previously updated above. There are 2 email commands, in the ELIF and ELSE statements. MAKE sure you change the “Neo” in the subject field and for /denied/Denied.Neo.jpg in the email line. – the image saves as the ID for person 2 – the denied person.

This should be the only changes you will need to make to the script.

**• Step 8 – Running the program**

To run the program, either from the Pi, or SSH into the Pi (this is what I did) and traverse to the root folder for the scripts ie

FacialRecogniton/ python3 faceRec.py

The menu will prompt 1 to enable to program and 0 to quit

(If you miss enter a character just enter it in twice again)

the camera light will show red, indicating it is ready to go.

Start by showing the picture of the denied person, the light should flash red on the Argon, you will see the ‘Neo’ label in the program output, it waits for the Argon to return 1, and then will return to the menu. The email will be sent to your Deakin email after a few moments. Usually around 10 seconds.

Then, enter 1 and do an unknown person, the same thing should happen.

Lastly, show your face, the Argon should unlock the door.

Facial Recognition Door Access!