IOT PROJECT

Face Recognition Ai Robot Using Raspberry Pi

By -:

Suraj Garg(181500735)
Chirag Varshney (181500199)
Ankit Kumar (181500097)
Ishant Goyal (181500282)

Under the supervision of

Mr. Amir khan Sir

Department of Computer Engineering & Application

**GLA University Institute of Technology Mathura-281426,
Uttar Pradesh**

ACKNOWLEDGEMENT

On the submission of our report on **“Face Recognition Ai Robot using Raspberry Pi”,**

We would like to sincere thanks to our mentor **Mr. Amir khan Sir.**
for his constant motivation and support during the project in the last 4 month.

He is providing a solid background for our studies and research.

ABSTRACT

The Project –Face Recognition Ai Robot‖ is a IOT-based project. The objective of this project is to provide the security in homes, Offices and Universities.
This Ai Robot will help the people who want to use advance security system in home and business .

This project uses Raspberry pi 4 model B and python language to achieve high speed of operation. Pi Camera v2 module is used for capturing images. These hardware components are provided us from IOT Lab.

This project consist main part i.e. face Recognition and we use Har Cascade Classifier is used for face detection and Local Binary Pattern algorithm for recognition technology.

INDEX

Motivation and Overview

The one of the main reason behind making this project is to provide the better security to those who really want to secure their livelihood in home and business purpose.

Objective

To detect and recognize the faces from image files from Raspberry Pi.

Components Requirements

- Raspberry Pi 3 Model B
- Raspberry pi Camera Module V2
- 16x2 LCD I2C Display
- HC-SR501 PIR Sensor
- Micro SD Card with NOOBS pre-installed

Application Area

- Attendence using face detection.
- Home Security
- Business Security
- Hospital Monitoring System

WORKING PRINCIPLE

- First we Captures 10 images of 10 different person with the help of Pi camera module.
- The images of each subject is stored in specific directory with 30 images .
- Each image size is of about 100x130pixels.
- The viola Jones detection algorithm is used for face detection whereas
- PCA algorithm is used for feature extraction and

- then Adaboost classifier is used for face recognition.

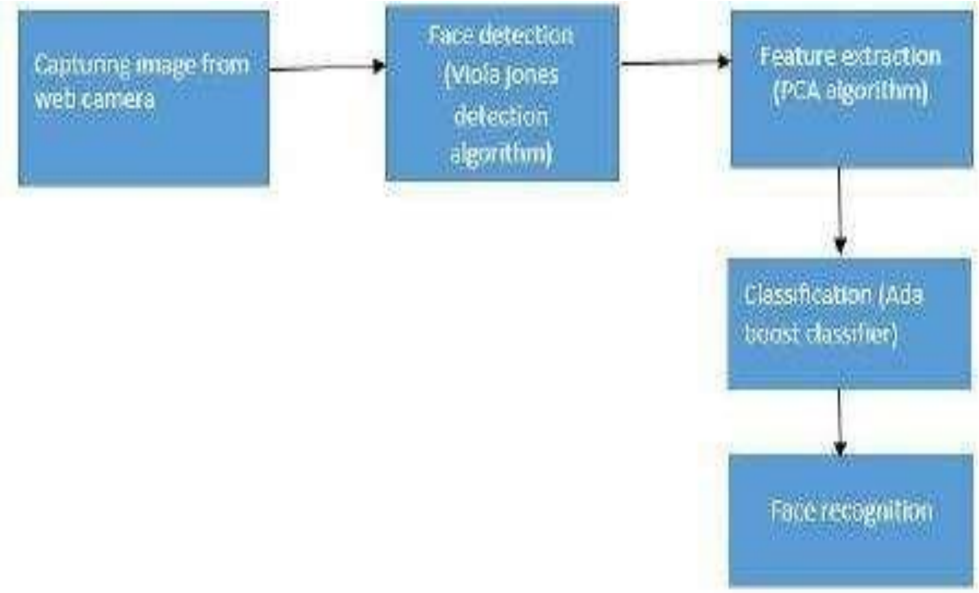

Capturing image from web camera
Face detection (Viola Jones detection algorithm)
Feature extraction (PCA algorithm)
Classification (Ada boost classifier)
Face recognition

Figure 1. Block diagram of face recognition

HARDWARE TOOLS

RASPBERRY PI 3 MODEL B

- The Raspberry Pi 3 is the third time Raspberry Pi.
- It has a 1.2GHz 64-bit quad-focus ARMv8 CPU, 802.11n Wireless LAN, Bluetooth 4.1, Bluetooth Low Energy (BLE),
- Also support 1GB RAM, 4 USB ports, 40 GPIO pins, Full HDMI port, Ethernet port, united 3.5mm sound jack and composite video, Camera interface (CSI), Display interface (DSI), Micro SD card opening and Video Core IV 3D outlines focus.

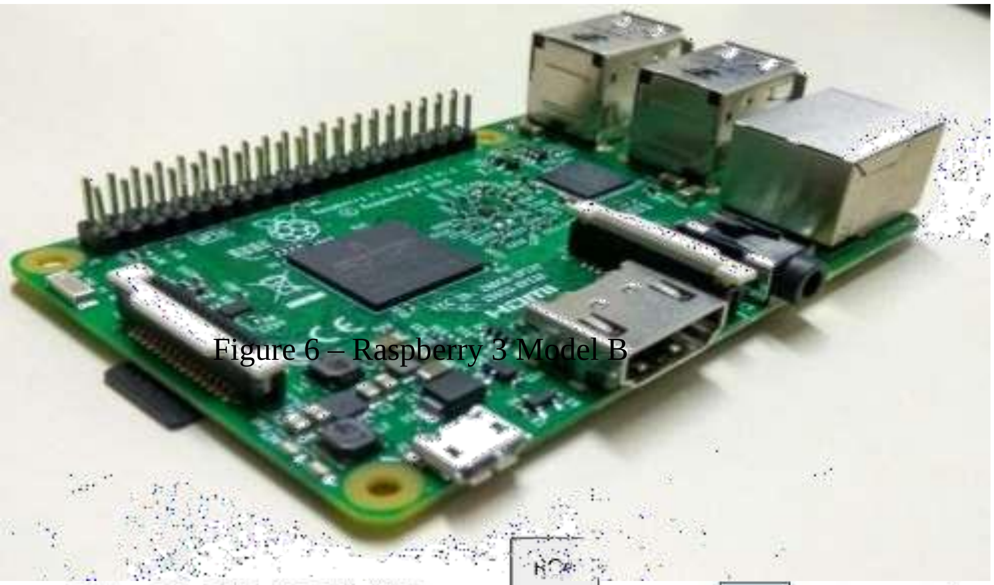


Figure 6 – Raspberry 3 Model B

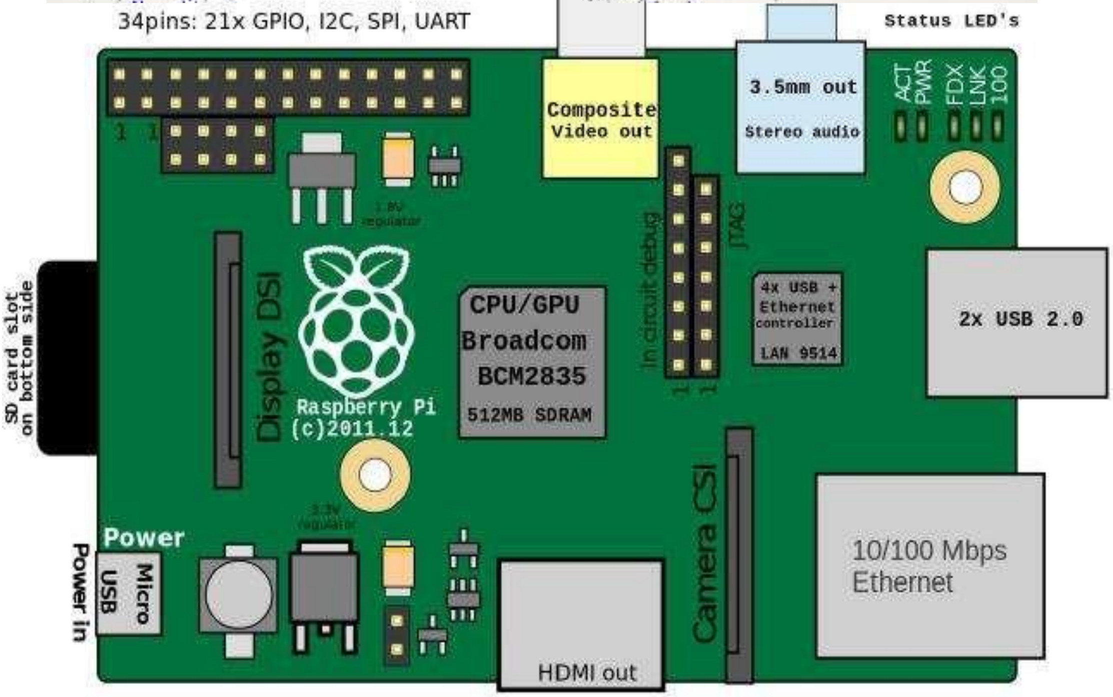


Figure 7. Layout of Raspberry Pi Model B

CAMERA MODULE V2

The v2 camera module has a Sony imx219 eight-megapixel sensor. The advanced camera module might be utilized to take superior quality video, and stills pix. It's smooth to use for fledglings, however has masses to offer propelled clients in the event that you're hoping to grow your know-how. There are masses of cases online of individuals the utilization of it for time-slip by, steady development, and diverse video astuteness. You may moreover utilize the libraries we bundle manage the computerized camera to make comes about. The digital cam works with all designs of raspberry pi 1, 2, and three. It can be gotten to by means of the mmal and v4l APIs, and there are different 0.33-festival libraries worked for it, alongside the picamera python library.

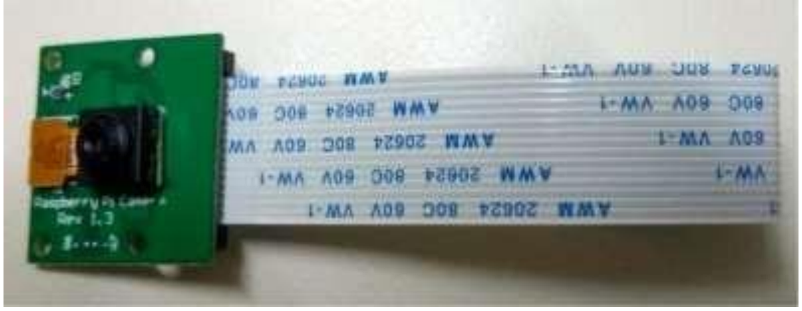


Figure 8– Camera Module v2

SD Card

A SD card is utilized to store the working arrangement of the Raspberry Pi. It likewise fills in as the capacity for all the bolster documents and programming for the face indicator and in addition stockpiling for info pictures to be tried.

SD CARD

SOFTWARE DEVELOPMENT

SETTING UP RASPBIAN

Downloaded the most recent adaptation of Raspbian. Required a picture author to compose the downloaded OS into the little scale SD card. So downloaded the "win32 circle imager‖. Embedded the SD card into the tablet/pc and run the picture essayist. When open, peruse and chose the downloaded Raspbian picture document. Chosen the right gadget that is the drive speaking to the SD card. On the off chance that the drive (or gadget) chose is unique in relation to the SD card, then the other chose drive will end up noticeably adulterated. So be watchful. Once the compose is finished, launch the SD card and insert it into the Raspberry Pi and turn it on. It should start booting up. in the wake of booting the Pi, there might be conditions when the customer capabilities like the "username" and mystery key will be asked. Raspberry Pi goes with a default customer name and mystery key therefore constantly use it at whatever point it is being asked. The accreditations are:

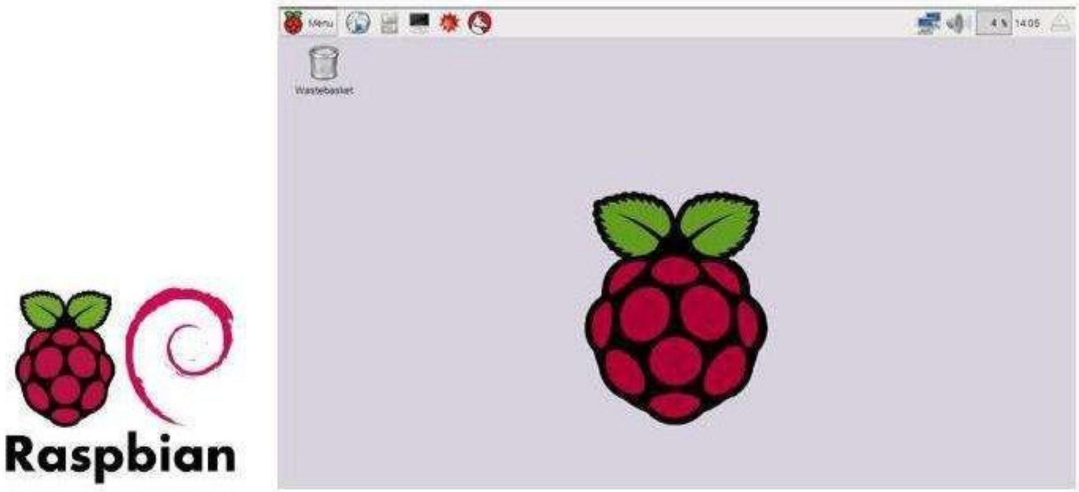


Figure 9. Logo and Desktop interface of Raspbian

```
login: pi
Password: raspberry
```

At the point when the Pi has been booted surprisingly, an arrangement screen called the "Setup Options" ought to show up and it will resemble the picture underneath. In the event that you have missed the "Setup Options" screen, it's not an issue, you can simply get it by writing the accompanying summon in the terminal.

```
$ sudo raspi-config
```

Since the Setup Options window is up, we set a couple of things. The main thing we did was to choose the principal choice in the rundown of the setup alternatives window, that is select the "Grow File system" choice and hit the enter key. We do this to make utilization of all the space introduce on the SD card as a full segment. This does is, grow the OS to fit the entire space on the SD card which can then be utilized as the capacity memory for the Pi. The second thing we did was to choose the third choice in the rundown of the setup alternatives window, that is select the "Empower Boot to Desktop/Scratch" choice and hit the enter key. It will take you to another window called the "pick boot alternative" window that resembles the picture beneath. In the "pick boot elective window", select the second decision, that is, "Desktop Log in as customer "pi" at the graphical desktop" and hit the enter get. e it at whatever point it is being asked. Once done you will be reclaimed to the "Setup Options" page, if not choose the "alright" catch at the base of this window and you will be reclaimed to the past window. We do this since we need to boot into the desktop condition which we know about. On the off chance that we don't do this progression, then the Raspberry Pi boots into a terminal each time with no GUI alternatives.

Once, both the means are done, select the "complete" catch at the base of the page and it ought to reboot consequently. On the off chance that it doesn't, then utilize the accompanying summon in the terminal to reboot. $ sudo reboot

After the reboot from the past walk, if everything went right, then you will end up on the desktop. When you are on the desktop, open a terminal and enter the going with charge to revive the firmware of the Pi.

```
$ sudo reboot
```

After the reboot from the past walk, if everything went right, then you will end up on the desktop. When you are on the desktop, open a terminal and enter the going with charge to revive the firmware of the Pi.**$ sudo rpi-update $ sudo apt-get update**

```
$ sudo apt-get upgrade
$ sudo reboot
```

The most recent firmware may have the settle to those bugs, in this way it's essential to refresh it in the first place itself.

INSTALLING OPENCV 3 ON A RASPBERRY PI

OPENCV

OpenCV is released under a BSD allow and thus it's free for both insightful and business use. It has C++, C, Python and Java interfaces and support Windows, Linux, Mac OS, iOS and Android. OpenCV was proposed for computational capability and with a strong focus on continuous applications. Written in upgraded C/C++, the library can abuse multi-focus taking care of. Enabled with OpenCL, it can abuse the hardware accelerating of the fundamental heterogeneous process organize. Gotten all around the world, OpenCV has more than 47 thousand people of customer gathering and surveyed number of downloads outperforming 9 million. Usage ranges from instinctive craftsmanship, to mines examination, sewing maps on the web or through forefront robotics.

OpenCV Library

The open source PC vision library, OpenCV, started as an examination extends at Intel in 1998. It has been accessible since 2000 under the BSD open source permit. OpenCV is gone for giving the devices expected to tackle PC vision issues. It contains a blend of low-level picture handling capacities and abnormal state calculations, for example, confront location, person on foot identification, include coordinating, and following. OpenCV's GPU module incorporates countless, and numerous

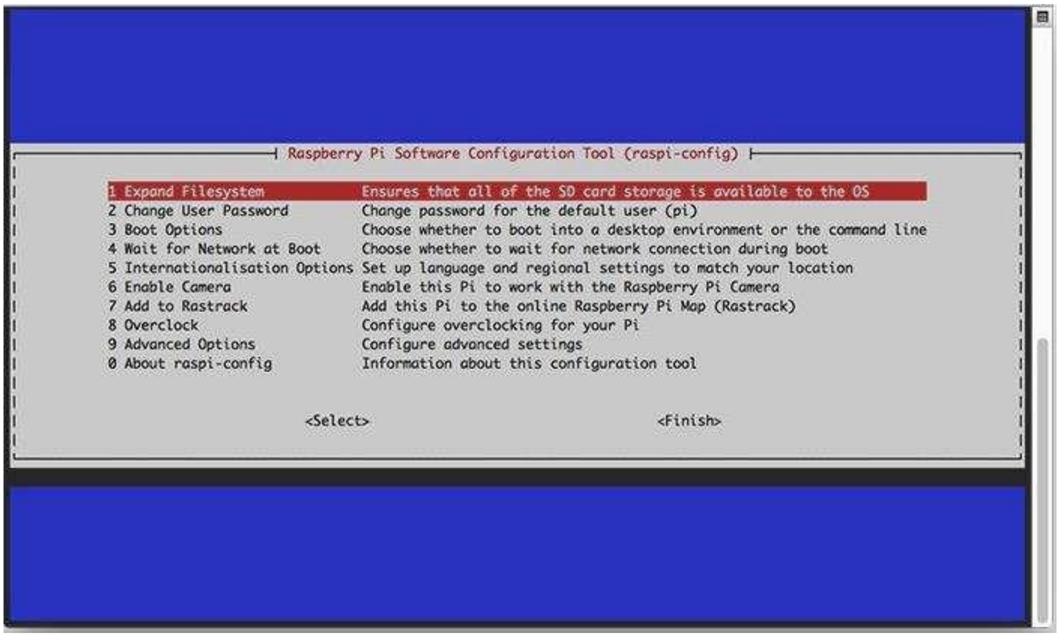
of them have been actualized in various forms, for example, the picture sorts (scorch, short, drift), number of channels, and outskirt extrapolation modes. This makes it trying to report correct execution numbers. An additional wellspring of trouble in refining the execution numbers down is the overhead of synchronizing and exchanging information. This implies best execution is acquired for extensive pictures where a great deal of preparing should be possible while the information lives on the GPU. To help the engineer make sense of the exchange offs; OpenCV incorporates an execution benchmarking suite that runs GPU capacities with various parameters and on various datasets. This gives a nitty gritty benchmark of how entirely different datasets are quickened on the client's equipment.

the main thing to do is to extend your file system to incorporate all accessible space on your small scale SD card.

$ sudo raspi-config

Once provoked, we chose the primary choice, "1. Extend File System", hit Enter on our console, bolt down to the "<Finish>" catch, and afterward reboot your Pi:

$ sudo reboot



```
┤ Raspberry Pi Software Configuration Tool (raspi-config) ├

1 Expand Filesystem            Ensures that all of the SD card storage is available to the OS
2 Change User Password         Change password for the default user (pi)
3 Boot Options                 Choose whether to boot into a desktop environment or the command line
4 Wait for Network at Boot     Choose whether to wait for network connection during boot
5 Internationalisation Options Set up language and regional settings to match your location
6 Enable Camera                Enable this Pi to work with the Raspberry Pi Camera
7 Add to Rastrack              Add this Pi to the online Raspberry Pi Map (Rastrack)
8 Overclock                    Configure overclocking for your Pi
9 Advanced Options             Configure advanced settings
0 About raspi-config           Information about this configuration tool

                 <Select>                                <Finish>
```

Figure 10 - Expanding the file system on your Raspberry Pi 3

In the wake of rebooting, your record structure should have been stretched out to join all available space on your littler scale SD card. You can watch that the plate has been reached out by executing df - h and taking a gander at the yield:

```
$ df -h
```

OpenCV, alongside every one of its conditions, will require a couple of gigabytes amid the gather, so you ought to erase the Wolfram motor to free up some space on your Pi

```
$ sudo apt-get purge wolfram-engine
```

The initial step is to refresh and update any current bundles:

```
$ sudo apt-get update
```

```
$ sudo apt-get upgrade
```

We then need to introduce some designer devices, including CMake, which helps us arrange the OpenCV assemble handle:

```
$ sudo apt-get install build-essential cmake pkg-config
```

Next, we need to present some photo I/O packages that empower us to stack diverse picture archive outlines from circle. Instances of such record bunches fuse JPEG, PNG, TIFF, et cetera.

```
$ sudo apt-get install libjpeg-dev libtiff5-dev libjasper-dev libpng12-dev
```

Essentially as we need picture I/O groups, we similarly require video I/O packs. These libraries empower us to scrutinize diverse video record bunches from hover and likewise work particularly with video streams:

```
$ sudo apt-get install libavcodec-dev libavformat-dev libswscale-dev libv4l-dev
```

```
$ sudo apt-get install libxvidcore-dev libx264-dev
```

The OpenCV library goes with a sub-module named high-up which is used to show pictures to our screen and make fundamental GUIs. Remembering the ultimate objective to arrange the highgui module, we need to present the GTK change library:

```
$ sudo apt-get install libgtk2.0-dev
```

Numerous operations within OpenCV (to be specific grid operations) can be enhanced further by introducing a couple of additional conditions:

```
$ sudo apt-get install libatlas-base-dev gfortran
```

These upgrade libraries are especially crucial for resource obliged contraptions, for instance, the Raspberry Pi.

Eventually, we should present both the Python 2.7 and Python 3 header records so we can fuse OpenCV with Python ties:

```
$ sudo apt-get install python2.7-dev python3-dev
```

On the off chance that you skirt this progression, you may see a mistake identified with the Python.h header record not being found when running make to accumulate OpenCV.

Since we have our conditions introduced, how about we snatch the 3.1.0 file of OpenCV from the authority OpenCV archive. (Note: As future renditions of openCV are discharged, you can supplant 3.1.0 with the most recent adaptation number):

```
$ cd ~
```

```
$ wget -O opencv.zip https://github.com/Itseez/opencv/archive/3.1.0.zip
```

```
$ unzip opencv.zip
```

We'll need the full introduce of OpenCV 3 (to have entry to components, for example, SIFT and SURF, for example), so we likewise need to snatch the opencv_contribution storehouse too:

```
$ wget -O opencv_contrib.zip https://github.com/Itseez/opencv_contrib/archive/3.1.0.zip
```

```
$ unzip opencv_contrib.zip
```

You may need to broaden the summon above using the "<=>" get in the midst of your copy and paste. The .speed in the 3.1.0.zip may have every one of the reserves of being cutoff in a couple programs. The full URL of the OpenCV 3.1.0 report is:

```
$ https://github.com/Itseez/opencv_contrib/archive/3.1.0.zip
```

Note: Make beyond any doubt your OpenCV and opencv_contrib renditions are the same (for this situation, 3.1.0). On the off chance that the renditions numbers don't coordinate, then you'll likely keep running into either order time or runtime.

Before we can begin accumulating OpenCV on our Raspberry Pi 3, we initially need to introduce pip, a Python bundle chief:

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

At first, it's basic to appreciate that a virtual circumstance is a one of a kind gadget used to keep the conditions required by different exercises in parceled puts by making isolated, free Python circumstances for each of them.

Basically, it disentangles the "Amplify X depends on upon adjustment 1.x, yet Project Y needs 4.x" circumstance. It moreover keeps your overall site-groups flawless, clean, and free from chaos.

If you may need a full illumination on why Python virtual circumstances are awesome practice, absolutely give this shocking web journal section on Real Python a read.

It's standard practice in the Python society to use virtual circumstances or something to that effect, so I extremely endorse that you do in like manner:

```
$ sudo pip install virtualenv virtualenvwrapper
$ sudo rm -rf ~/.cache/pip
```

Since both virtualenv andvirtualenvwrapper have been presented, we need to invigorate our ~/.profile report to fuse the going with lines at the base of

```
# virtualenv and virtualenvwrapper
export WORKON_HOME=$HOME/.virtualenvs
source /usr/local/bin/virtualenvwrapper.sh
```

In past instructional exercises, I've prescribed utilizing your most loved terminal-based content tool, for example, vim,emacs, or nano to refresh the ~/.profile document. In case you're alright with these editors, simply ahead and refresh the document to mirror the progressions specified previously. Else, you ought to just utilize feline and yield redirection to deal with refreshing.

```
$ echo -e "\n# virtualenv and virtualenvwrapper" >> ~/.profile

$ echo "export WORKON_HOME=$HOME/.virtualenvs" >> ~/.profile $

echo "source /usr/local/bin/virtualenvwrapper.sh" >> ~/.profile
```

Since we have our ~/.profile refreshed, we have to reload it to ensure the progressions produce results. You can compel a reload of your ~/.profile document by:

Logging out and afterward logging back in.

Shutting a terminal occasion and opening up another one

Or, on the other hand my undisputed top choice, simply utilize the source order

```
$ Source ~/.profile
```

Next, we should make the Python virtual condition that we'll use for PC vision advancement:

```
$ mkvirtualenv cv -p python2
```

This order will make another Python virtual condition named cv utilizing Python 2.7.

In the event that you rather need to utilize Python 3, you'll need to utilize this charge:

```
$ mkvirtualenv cv -p python3
```

If you ever reboot your Raspberry Pi; log out and log back in; or open up another terminal, you'll need to use the workon request to re-get to the cv virtual condition. In past blog passages, I've seen perusers use the mkvirtualenv summon — this is totally unneeded! The mkvirtualenv request is planned to be executed only once: to truly make the virtual condition.

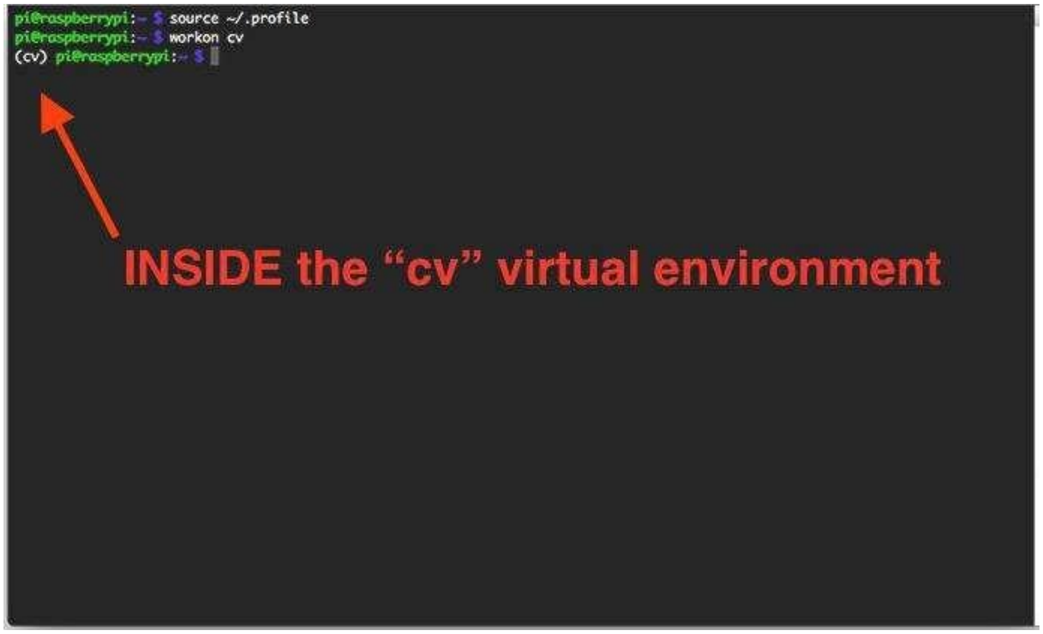
Starting there ahead, you can use deal with and you'll be dropped down into your virtual condition:

e of the record:

$ Source ~/.profile

$ workon cv

To approve and guarantee you are in the cv virtual condition, look at your summon line — in the event that you see the content (cv) going before your incite, then you are in the cv virtual condition:



```
pi@raspberrypi:~ $ source ~/.profile
pi@raspberrypi:~ $ workon cv
(cv) pi@raspberrypi:~ $
```

Figure 11 - The "(cv)" message on your provoke, showing that you are in the cv virtual condition. Our exclusive Python reliance is NumPy, a Python bundle utilized for numerical handling:

$ pip install numpy

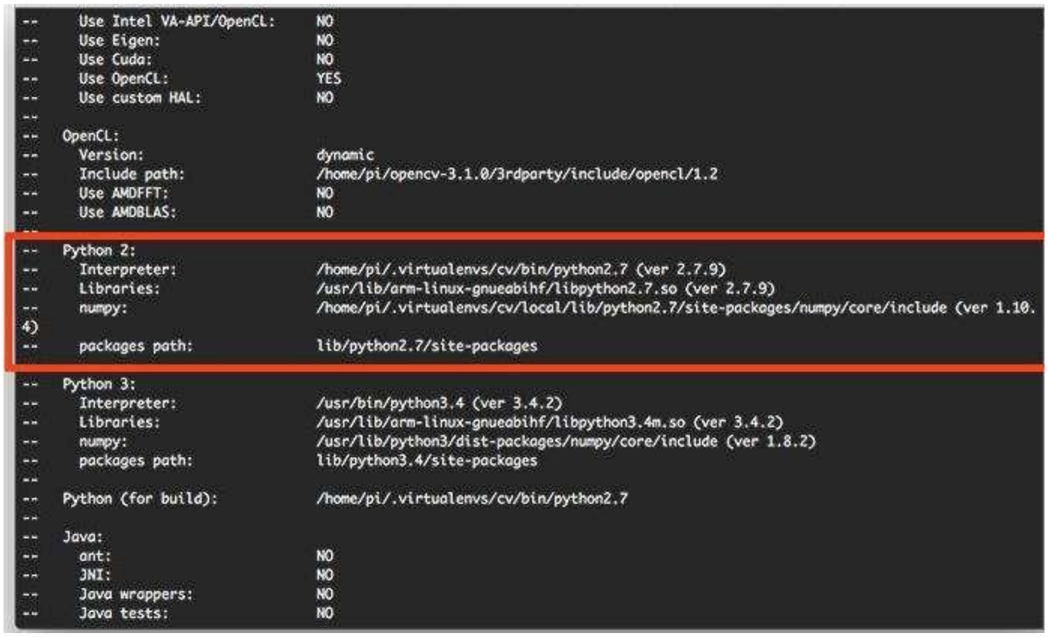
We are as of now arranged to arrange and present OpenCV! Twofold watch that you are in the cv virtual condition by taking a gander at your incite (you should see the (cv) content going before it), and if not, simply execute take a shot at:

$ workon cv

When you have promised you are in the cv virtual condition, we can setup our produce using CMake:

```
 $ cd ~/opencv-3.1.0/
$ mkdir build
$ cd build
$ cmake -D CMAKE_BUILD_TYPE=RELEASE
   \ -D CMAKE_INSTALL_PREFIX=/usr/local \
   -D INSTALL_PYTHON_EXAMPLES=ON \
   -D OPENCV_EXTRA_MODULES_PATH=~/opencv_contrib-3.1.0/modules \
   -D BUILD_EXAMPLES=ON
```

In the event that you are incorporating OpenCV 3 for Python 2.7, then ensure your Python 2 area incorporates substantial ways to the Interpreter, Libraries, numpy and bundles way, like my screenshot underneath:



```
--     Use Intel VA-API/OpenCL:     NO
--     Use Eigen:                   NO
--     Use Cuda:                    NO
--     Use OpenCL:                  YES
--     Use custom HAL:              NO
--
--   OpenCL:
--     Version:                     dynamic
--     Include path:                /home/pi/opencv-3.1.0/3rdparty/include/opencl/1.2
--     Use AMDFFT:                  NO
--     Use AMDBLAS:                 NO
--
--   Python 2:
--     Interpreter:                 /home/pi/.virtualenvs/cv/bin/python2.7 (ver 2.7.9)
--     Libraries:                   /usr/lib/arm-linux-gnueabihf/libpython2.7.so (ver 2.7.9)
--     numpy:                       /home/pi/.virtualenvs/cv/local/lib/python2.7/site-packages/numpy/core/include (ver 1.10.
4)
--     packages path:               lib/python2.7/site-packages
--   Python 3:
--     Interpreter:                 /usr/bin/python3.4 (ver 3.4.2)
--     Libraries:                   /usr/lib/arm-linux-gnueabihf/libpython3.4m.so (ver 3.4.2)
--     numpy:                       /usr/lib/python3/dist-packages/numpy/core/include (ver 1.8.2)
--     packages path:               lib/python3.4/site-packages
--
--   Python (for build):            /home/pi/.virtualenvs/cv/bin/python2.7
--
--   Java:
--     ant:                         NO
--     JNI:                         NO
--     Java wrappers:               NO
--     Java tests:                  NO
```

Figure 12 - Ensuring that Python 2.7 will be utilized when gathering OpenCV 3 for Raspbian Jessie on the Raspberry Pi 3.

Provided that this is true, get to the cv virtual condition using workon cv and re-run the cmake arrange plot above.Finally, we are directly arranged to accumulate OpenCV:

```
$ make -j4
```

The - j4 charge controls the amount of focuses to utilize when organizing OpenCV 3. The Raspberry Pi 3 has four focuses; along these lines we supply an estimation of 4 to empower OpenCV to total speedier. Notwithstanding, due to race conditions, there are times when make mistakes out when utilizing different centres. In the event that this transpires, I recommend beginning the aggregation once again and utilizing just a single centre::

```
$ make clean
$ make
```

From that point, you should simply introduce OpenCV 3 on your Raspberry Pi 3

```
$ sudo make install
$ sudo ldconfig
```

OpenCV ought to now be introduced in/usr/neighborhood/lib/python2.7/site-pacakges. You can confirm this utilizing the ls charge:

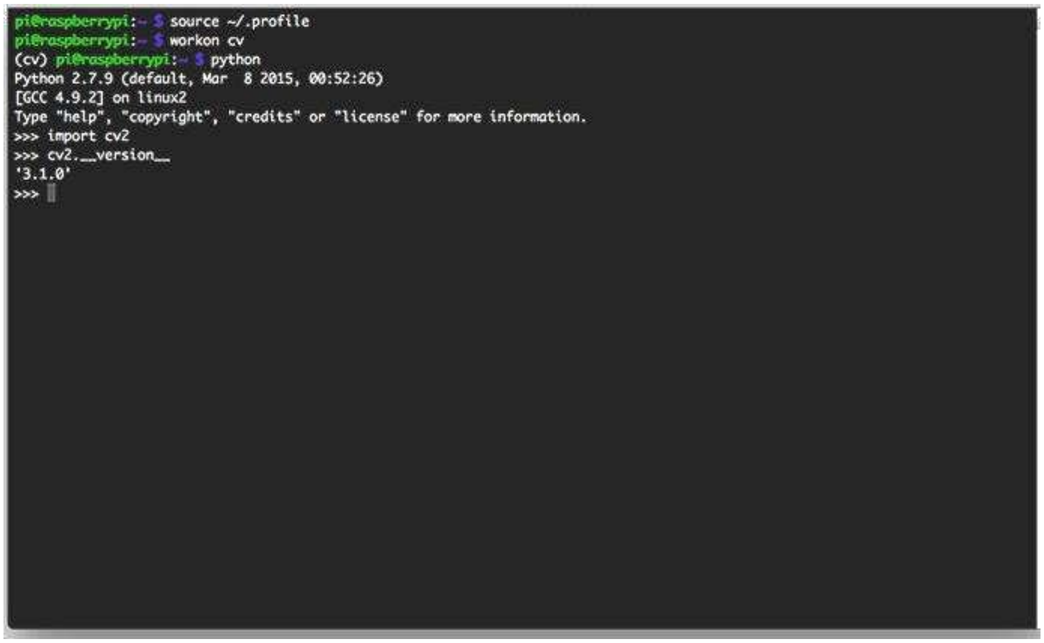
```
$ ls -l /usr/local/lib/python2.7/site-packages/
Total 1852
```

Our last stride is to sym-interface the OpenCV ties into our cv virtual condition for Python 2.7:

```
$ cd ~/. virtualenvs/cv/lib/python2.7/site-packages/
$ ln -s /usr/local/lib/python2.7/site-packages/cv2.so cv2.so
```

How about we initially confirm that your OpenCV establishment is working appropriately. Open up another terminal; execute the source andwork on charges, and after that at last endeavour to import the Python + OpenCV ties:

```
$ source ~/.profile
$ workon cv
```



```
pi@raspberrypi:~ $ source ~/.profile
pi@raspberrypi:~ $ workon cv
(cv) pi@raspberrypi:~ $ python
Python 2.7.9 (default, Mar  8 2015, 00:52:26)
[GCC 4.9.2] on linux2
Type "help", "copyright", "credits" or "license" for more information.
>>> import cv2
>>> cv2.__version__
'3.1.0'
>>>
```

Figure 13 - Affirming OpenCV 3 has been effectively introduced on my Raspberry Pi 3 running Raspbian

FACE DETECTION ALGORITHM

A Haar-like component considers neighboring rectangular areas at a specific region in an area window, adds up to up the pixel controls in each locale and figures the differentiation between these totals. This refinement is then used to request subsections of a picture. An instance of this would be the revelation of human appearances. For the most part, the districts around the eyes are darker than the extents on the cheeks. One instance of a Haar-like component for face acknowledgment is thus a plan of two neighboring rectangular locales over the eye and cheek ranges.

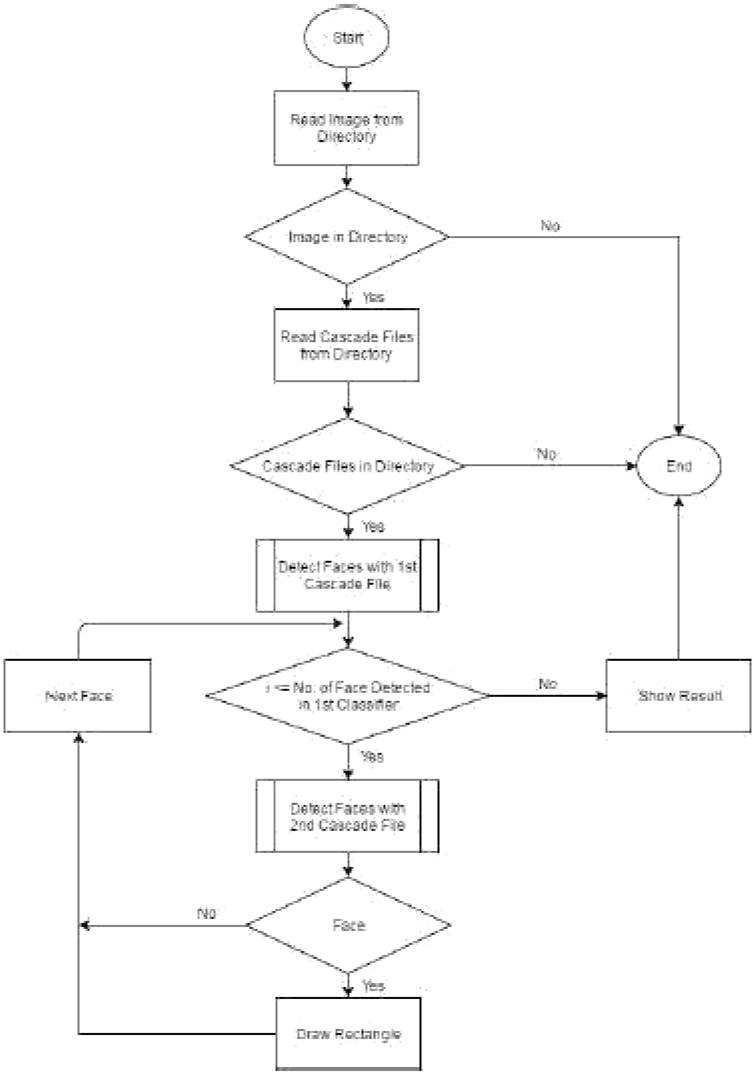

Start
Read Image from Directory
Image in Directory
No
Yes
Read Cascade Files from Directory
Cascade Files in Directory
No
End
Yes
Detect Faces with 1st Cascade File
Next Face
i <= No. of Face Detected in 1st Classifier
No
Show Result
Yes
Detect Faces with 2nd Cascade File
No
Face
Yes
Draw Rectangle

Figure 14. Flowchart for face detection for image files

CONCLUSION

This iot based project is the fruit of continuous work of our Team. We use python language, xml and shell scripting language for further development of this project . Raspberry pi model 3 helps for Faster respose, CameraV2 module used for capturing images and store in bucket then using viola jones detection algorithm and PCA algorithm for face Recognition.

` REFERENCES

1. You-tube- www.youtube.com
2. Coursera- www.coursera.org
3. www.howtoforge.com
4. www.pyimagesearch.com
5. https://www.raspberrypi.org/documentation/

GITHUB REPOSITORY LINK :

https://github.com/suraj-garg/Mini_Project_2