The facial recognition library we are using is one that has been maintained for many years by [Adam Geitgey](https://github.com/ageitgey/face_recognition). It contains many examples, including Python 3 bindings to make it really simple to build your own facial recognition applications.

What is not so easy is the number of dependencies that need to be installed first. There are way too many to list here, and you probably won’t want to type them out, so head over to [hsmag.cc/FacialRec](http://hsmag.cc/FacialRec) so that you can cut and paste the commands.

This step will take a while to complete on a Raspberry Pi 4, and significantly longer on a Model 3 or earlier.

**3**.**Install the libraries**  
We can install Adam’s applications and Python bindings with a simple, single command:

sudo pip3 install face\_recognition

Once installed, there are some examples we can download to try everything out.

cd  
git clone --single-branch https://github.com/ageitgey/face\_recognition.git

In this repository is a range of examples showing the different ways the software can be used, including live video recognition.

**5. Example images**

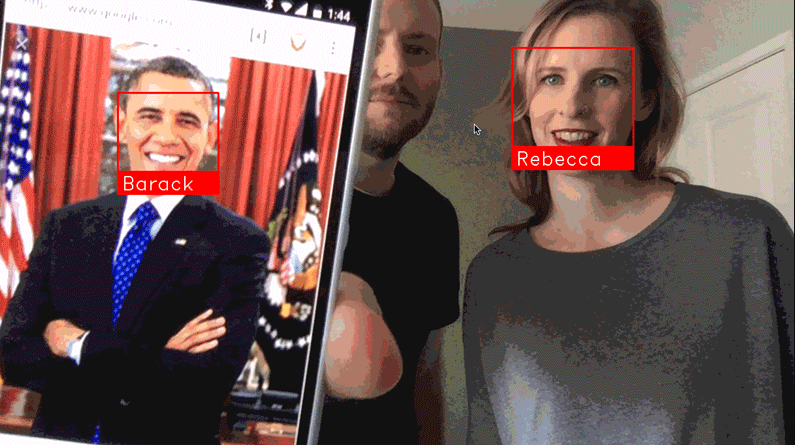
The examples come with a training image of Barack Obama. To run the example:

cd ./face\_recognition/examples  
python3 facerec\_on\_raspberry\_pi.py

On your smartphone, find an image of Obama and point it at the camera. Providing focus and light are good you will see:

“I see someone named Barack Obama!”

If you see a message saying it can’t recognise the face, then try a different image or try to improve the lighting if you can. Also, check the focus for the camera and make sure the distance between the image and camera is correct.



Who are you? What even is a name? Can a computer decide your identity?

**6. Training time**  
The final step is to start recognising your own faces.

Create a directory and, in it, place some good-quality passport-style photos of yourself or those you want to recognise.

You can then edit the facerec\_on\_raspberry\_pi.py script to use those files instead.

You’ve now got a robust prototype of face recognition.

This is just the beginning. These libraries can also identify ‘generic’ faces, meaning it can detect whether a person is there or not, and identify features such as the eyes, nose, and mouth.

There’s a world of possibilities available, starting with these simple scripts. Have fun!

Each month, HackSpace magazine brings you the best projects, tips, tricks and tutorials from the makersphere. You can get it from the [Raspberry Pi Press online store](https://store.rpipress.cc/), The Raspberry Pi store in Cambridge, or your local newsagents.

Each issue is free to download from the [HackSpace magazine website](https://hackspace.raspberrypi.org/issues).

**import face\_recognition**

**import picamera**

**import numpy as np**

# Get a reference to the Raspberry Pi camera.

# If this fails, make sure you have a camera connected to the RPi and that you

# enabled your camera in raspi-config and rebooted first.

camera = picamera.PiCamera()

camera.resolution = (320, 240)

output = np.empty((240, 320, 3), dtype=np.uint8)

name = "<Unknown Person>"

# Load a sample picture and learn how to recognize it.

TheImage = face\_recognition.load\_image\_file("obama\_small.jpg")

face\_encoding = face\_recognition.face\_encodings(TheImage)[0]

face\_locations = []

face\_encodings = []

while True:

print("Capturing image.")

# Grab a single frame of video from the RPi camera as a numpy array

camera.capture(output, format="rgb")

# Find all the faces and face encodings in the current frame of video

face\_locations = face\_recognition.face\_locations(output)

print("Found {} faces in image.".format(len(face\_locations)))

face\_encodings = face\_recognition.face\_encodings(output, face\_locations)

# Loop over each face found in the frame to see if it's someone we know.

for face\_encoding in face\_encodings:

# See if the face is a match for the known face(s)

match = face\_recognition.compare\_faces([face\_encoding], face\_encoding)

if match[0]:

name = "Barack Obama"

print("I found {}!".format(name))

= = = = = = = = = = = = =

OPCIONES PARA CARGAR

#CARGAR TODOS LOS ARCHIVOS EN DIRECTORIO FacesDir

import os

arr = os.listdir('.\FacesDir')

print(arr)

#EVALUAR CADA FOTO

for f1 in arr:

print(f1)

import pathlib

arr = pathlib.Path().glob("\*.py")

import glob

arr = glob.glob('145592\*.jpg')

import os

destdir = '/var/tmp/testdir'

files = [ f for f in os.listdir(destdir) if os.path.isfile(os.path.join(destdir,f)) ]

for file in arr:

print(file)

You can use [os.scandir()](https://docs.python.org/3/library/os.html" \l "os.scandir). New function in stdlib starts from Python 3.5.

import os

for entry in os.scandir('.'):

if entry.is\_file():

print(entry.name)

mas avanzado

Lo pasa a gris para ser mas facil la relacion

<https://www.hackster.io/mjrobot/real-time-face-recognition-an-end-to-end-project-a10826>

<https://www.pyimagesearch.com/2018/06/25/raspberry-pi-face-recognition/>