Cloud Computing(CS6847)
Course Project

Face Recognition using Fog plus Cloud Computing
User Manual and Readme

By: Submitted to:

Gaurav Arora(CS18M021)   Dr. D. Janakiram
Kumar Vaibhav(CS18M030)
Niketan Shahapur(CS18M036)

Index

Prerequisite

Hardware

1. Raspberry Pi
2. USB mouse
3. USB Keyboard
4. Monitor
5. HDMI cable
6. Memory Card 64GB
7. LAN cable/WiFi network
8. Raspberry Pi Case
9. Webcam

Software

1. Ubuntu Mate OS
2. Mosquitto Broker
3. OpenCV
4. Dlib
5. Motion
6. Python

Programs/files

1. client.py
2. server.py
3. encode_face.py
4. c.xml
5. e.pickle

Setup Raspberry Pi

Hardware Connection

To start with the project, setup hardware requirements of Raspberry Pi:

1. Connect power adapter to micro USB type B port of Raspberry Pi
2. Connect Keyboard and Mouse to USB ports
3. Connect a monitor using HDMI cable or HDMI to VGA convertor is monitor only supports VGA to HDMI port of the Raspberry Pi
4. Insert memory card flashed with Ubuntu Mate OS (lookup OS setup section)
5. Connect webcam to the Raspberry Pi using USB port

OS Setup

Raspberry Pi needs specially compiled OS for it which are available in Raspberry Pi official website. You can choose between 10s of OS available for it or you can compile your own OS for it from scratch, in beginning we used Raspbian OS which is official OS for Raspbian. But it does not support opencv library due to architecture mismatch as Raspberry pi is manufactured with ARM Cortex-A53 and we cannot install using pip install opencv command so we compile Ubuntu Mate from scratch.

Steps to compile Ubuntu Mate from scratch to install opencv:

1. Update the packages using apt update and upgrade commands.
2. Install required OS libraries like libgtk,libatlas etc.
3. Install required python libraries like numpy etc.
4. Download opencv and opencv contrib.
5. Compile and install opencv with contrib module.

Once we have required OS we can configure Raspberry pi using steps:

1. Take an 64GB capacity micro SD card (memory card).
2. Insert SD Card into another machine to format it.
3. Format it using SD Association formatter tool (link) in either Windows or iOS. For Linux use gparted application, use this link to know procedure for Linux.
4. Flush the micro SD card with Ubuntu Mate.
5. Insert the card into Raspberry Pi and boot the RasPi (follow Hardware Connection setup for instructions).

For Mosquitto MQTT Broker we require MQTT broker and client libraries(suitable for lightweight publish/subscribe messaging transport) which we cannot install in Ubuntu Mate using apt-get install mosquitto command so we again compile Ubuntu Mate.

Steps to compile Ubuntu Mate from scratch to install MQTT:

1. Since we already have required OS and python library so just download MQTT and it's module.
2. Compile and install using MQTT-client module.
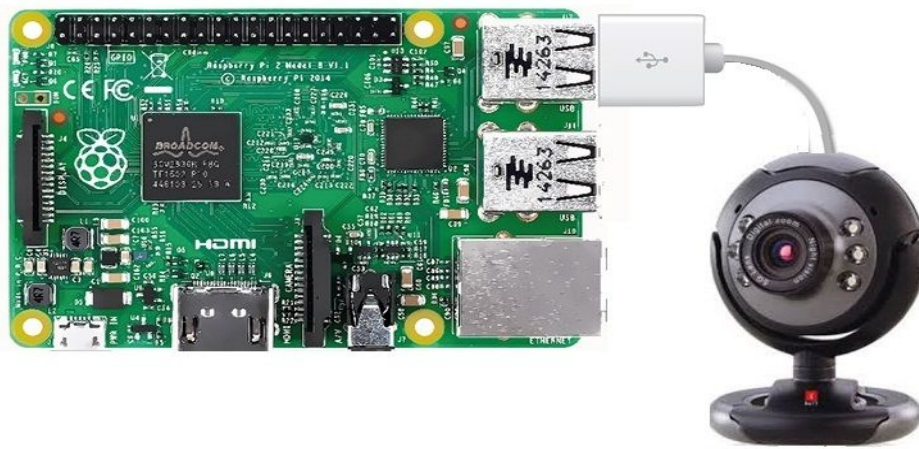3. Build once and during next boot up time just follow the installing mosquitto broker steps.

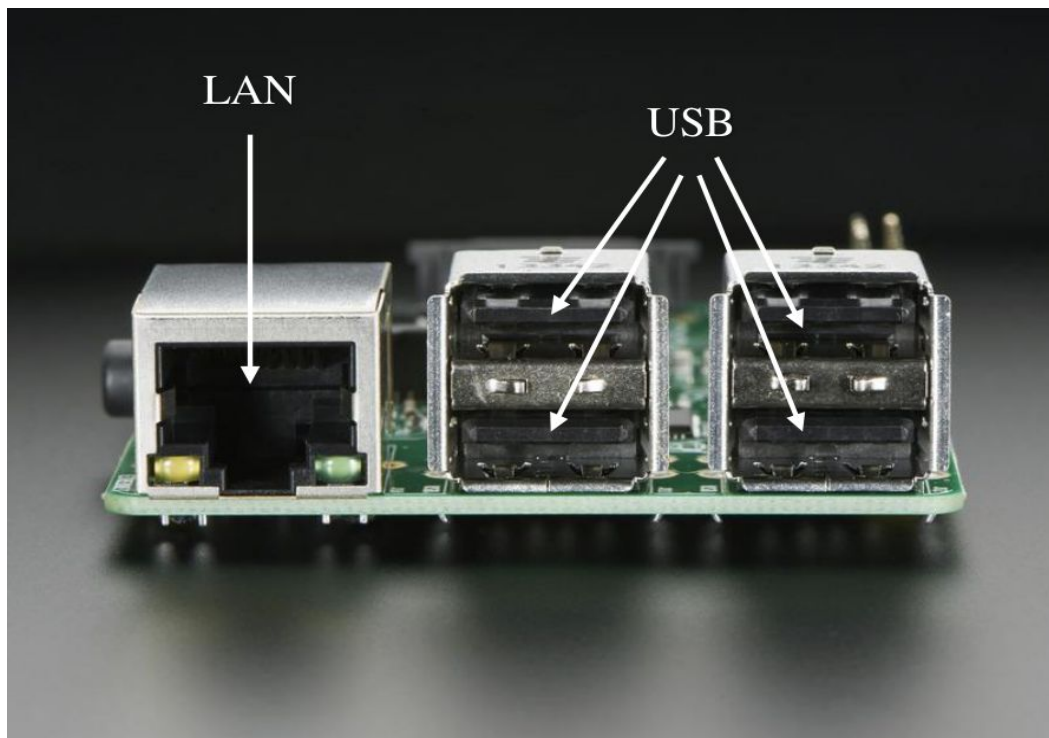Fig. 1 Raspberry Pi and webcam Connection sketch

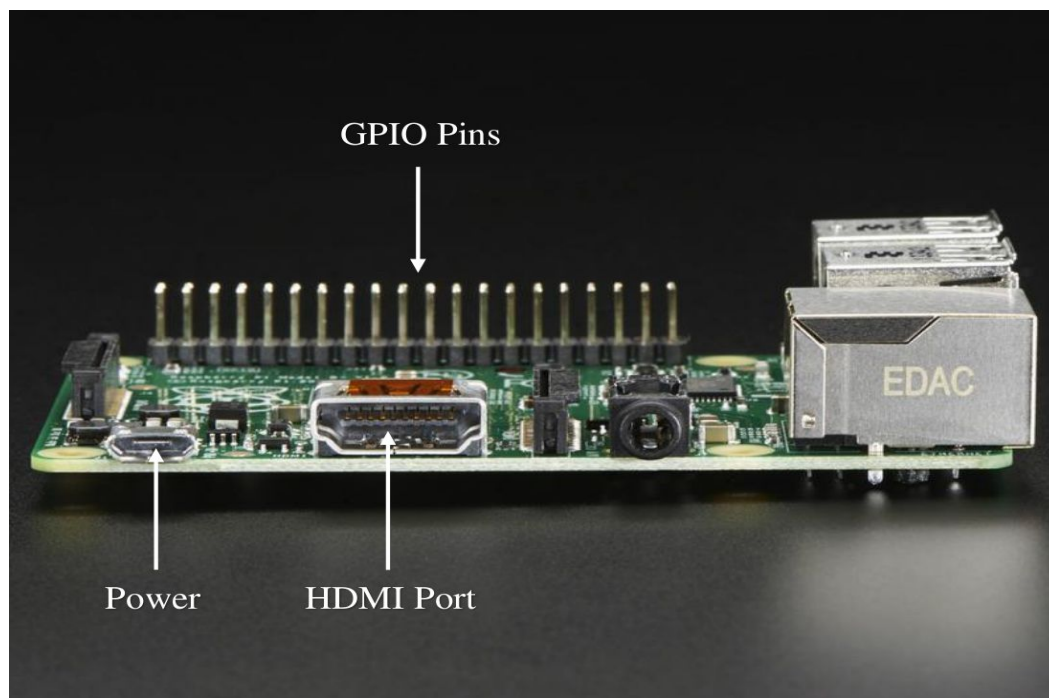
Fig. 2 Raspberry Pi Side View 1

Fig. 3 Raspberry Pi Side View 2

Installation of Packages

Installing Mosquitto Broker

Mosquitto Broker is needed to send and receive message using MQTT protocol. Follow below commands to install it using terminal.

1. *sudo apt-get install mosquito*
2. *sudo apt-get install mosquitto-clients*
3. *sudo apt-get install python-virtualenv*
4. *git clone https://github.com/roppert/mosquitto-python-example.git*
5. *cd mqtt-mosquitto-example*
6. *Virtualenv*
7. *source bin/activate*
8. *pip install -r requirements.txt*

Installing Motion

Motion driver is needed to use webcam in streaming mode. Follow below commands to install it using terminal.

1. *sudo apt-get update*
2. *sudo apt-get upgrade*
3. *sudo apt-get install motion*
4. *sudo vim /etc/motion/motion.conf*
5. Change the configuration accordingly.

Installing Opencv

1. OpenCV needs to be built from source since existing binaries are not available for Raspberry pi.
2. Refer https://www.pyimagesearch.com/2017/09/04/raspbian-stretch-install-opencv-3-python-on-your-raspberry-pi/ for detailed steps for installation on Raspberry pi.
3. To install on Ubuntu cloud server use pip3 install opencv-python

Running Program

Starting program

1. Run server program using python3 server.py -c c.xml -e e.pickle
2. Configure server address in client python script and run python3 client.py on client.
3. To encode new face dataset on server place the user name labeled folder inside the dataset directory and run python3 encode_faces.py -i dataset -e e.pickle -d hog