

## Asistencia – Software execution guide

To run the project, the following dependencies must be met:

OS:

- 64-bit Ubuntu 14.04+, Debian 8+, openSUSE 13.3+, or Fedora Linux 24+

**Linux Packages/Dependencies Required:**

- **openssh-client:** openssh-client package is used for sending all ssh requests to a server machine, by encrypting the requests to ensure maximum security.
- **openssh-server:** openssh-server package is used for reception and handling of ssh requests from a remote ssh client.
- **sshpass:** sshpass package is used for automation the authentication process to establish an ssh connection between a client machine and a server machine.
- **Anaconda distribution of python3.5+ and conda package:** python3.5+ version is to be used for execution of the project for compatibility issues. However anaconda distribution of python is advised for higher optimization.
- **OpenCV-python:** This package consists of all special wheel-binary modules required for running OpenCV algorithms using python.
- **Linux kernel 4.10+ for best performance:** Linux kernel 4.10+ is suggested for best compatibility with the Nvidia CUDA packages for GPU acceleration of machine learning algorithms used in this project. Linux kernel 4.10+ has better DKMS (Dynamic Kernel Module Support) system for Nvidia GPU related compatibility.

- **MySQL (MySQL-server):** MySQL-server package is required for the MySQL database used in the project.
- **python-pi-camera:** This package provides an interface to access the camera module of raspberry pi using python. This package supports various operations such as capturing still image, video, setting resolution and other aspects of the images and video etc.
- **CUDA package:** CUDA package enables Nvidia CUDA (Compute Unified Device Architecture) technology which is a parallel computing technology using Nvidia GPUs (Graphics Processing Unit), in order to accelerate the performance of various machine learning algorithms. Most machine learning algorithms require heavy amounts of vector processing and matrix operations, which can be processed parallelly by 1000s of cores in a GPU in a much faster pace compared to the traditional CPU.
- **cuDNN package:** cuDNN package is a collection of Deep Neural Network utility APIs, written in C++, which can directly run on GPU architecture. cuDNN package is heavily utilized by various Machine Learning Libraries such as TensorFlow, pytorch, theano, keras, etc.
- **libcupty-dev package:** This library consists of NVIDIA CUDA Profiler Tools Interface development files.

#### **Python Packages/Dependencies Required:**

- **Dlib:** Dlib is a collection of a repertoire of machine learning algorithms built using C++. Machine learning algorithms become extremely difficult and time taking to implement by an average developer, due to the heavy amounts of research work

that's required and requirement of access to machines with high computational capabilities. Dlib provides a comprehensive and efficient API of these complex machine learning algorithms, in an aim to solve real world problems. The usage of C++ has made Dlib one of the fastest libraries of machine learning, and its compatibility to run algorithms using GPU (Graphical Processing unit) has made it possible to improve the performance of the algorithms in real-time through Nvidia CUDA (Compute Unified Device Architecture) technology. Dlib is an open source library, allowing users to make any changes to the code free of charge and apply in customized scenarios.

- **Face-recognition:** Face-recognition is a python package, which abstracts the ResNet-32 Neural network mentioned in the algorithm section of this document, which was implemented in the Dlib library, inspired by a paper published from Cornell University. This particular neural network is used for face recognition task, by generating a 128-dimension vector for each inputted face. The package also consists of a CNN and a HOG algorithm as mentioned in the algorithms description, which can be used for the task of detecting multiple faces in an image, and finding their bounding box, so that they can be further sent to the face recognition ResNet-32 which performs the task of recognizing the face of the student in the bounding box.
- **Face-recognition-models:** This library consists of the trained weights and models used by the face-recognition library, and acts as a support to the face-recognition library.
- **Imutils:** This library is a collection of basic image processing functions such as rotation, translation, resizing images etc.
- **MySQL-connector:** This package acts as an interface for python code to access MySQL Database. It is similar to JDBC, which is used for Java, however it works

with python. It allows most operations such as insertion, updating, deleting, querying, creating new tables etc., via python code.

- **NumPy:** NumPy is a collection of abstract data structures with methods to operate on the abstract data structures, used for scientific computation. Most vectoral and matrix operations in python are done using this library, owing to its extremely efficient and optimized data structures and algorithms. NumPy is a very powerful library and is extensively used by a large number of scientific researchers and engineers.
- **OpenCV-python:** This package consists of all special wheel-binary modules required for running OpenCV algorithms using python. It consists over 30,000 computer vision related algorithms.
- **OpenCV-contrib-python:** This package is a support package for the OpenCV-python package and provides constants, weights and support data for algorithms in OpenCV.
- **Pickle:** Pickle is a python package, which allows storage of pythonic objects such as lists, tuples, dictionaries, strings, python objects, etc. in the ROM (Read Only Memory) for permanent storage. The object or set of objects are written as a file specified in a path. The file needs to be read to retrieve the objects again.
- **Os:** The os python library is a cross-platform library, that supports primitive os level operation abstract at a level high enough to access through python. Several operations such as creation of file, directory, copying files, moving files, listing files in a directory, conversion of absolute paths to relative and vice-versa, appending paths in file systems and much more.
- **Pillow:** Pillow, also known as PIL is a Python Image Library which consists of various basic image processing utilities.

- **Argparser:** Argparser is a python library which enables command line arguments in python, in a traditional UNIX/LINUX pattern, ie, switch followed by value which allows more flexible command line parsing of arguments, allowing the user to provide the command line arguments in any order required. It also provides an extensive help support, which displays help messages when required to the user and also notifies when illegal switches are used or necessary switches are omitted.
- **Django:** Django is a high-level Python Web framework that supports rapid development with a clean and structured design. It makes web development a clean and straight forward process and also allows very low-level customization, making it a very powerful tool. It is an open source project and also hosts an internal server to host websites and web-apps developed in real time and for demonstration.
- **Django-crispy-forms:** Django-crispy-forms provides with a crispy filter and {% crispy %} tag that will let you control the rendering behavior of your Django forms in a very elegant and DRY way. Very powerful custom form templates can be made.

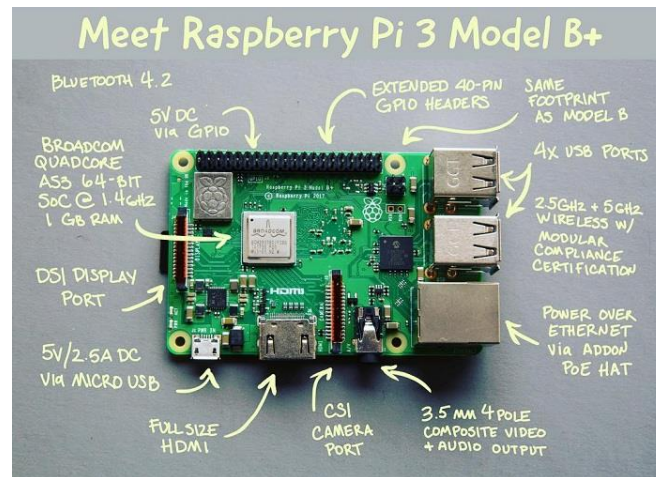
### 5.1.2 Hardware Requirements

- **Server Machine** with the following minimum configuration:
  - 8 GB of RAM, 16 GB recommended.
  - Processor: Intel i7 6<sup>th</sup> gen and above is suggested.
  - Nvidia GPU with at least 4 GB of VRAM.
- **Graphics Processing Unit** must conform to Nvidia CUDA support and must be a high-performance GPU such as GTX 960m for mobile GPUs or GTX 1080 Ti for desktop variants with core count of at least a 1000.



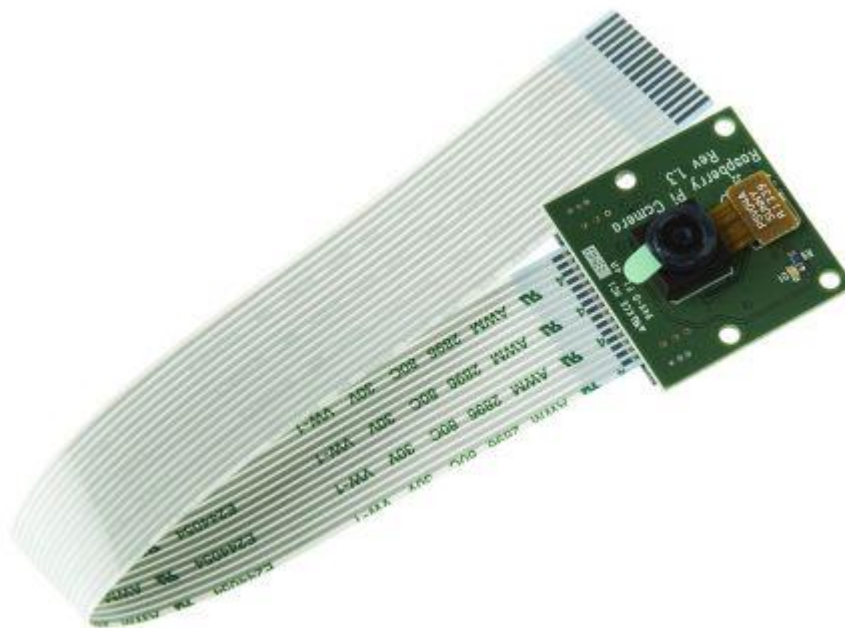
**Fig 5.1 Graphic Card**

- **Raspberry pi 3B+ and above:** Raspberry pi is a mini computer, which was initially designed and implemented to teach programming to school kids. However later realizing its potential as an IoT device, it has evolved into a great client system for performing small and simple tasks in an automated way, being placed at remote areas. Raspberry pi can be combined with various other electronic equipment such as cameras, infra-red sensors, sonar sensors for distance calculation, smoke detectors, etc and can also be connected to various actuators such as servo motors, dc motors, speakers etc. However, in the current project, the need for heavy image processing in the client side, the most modern and latest version of raspberry pi is needed to manage such computationally expensive tasks.



**Fig 5.2 Rasp berry pi**

- Raspberry pi camera module:** The Raspberry Pi Camera Module v2 is a medium quality 8 megapixel Sony IMX219 image sensor custom designed with add-on board for Raspberry Pi, with a fixed focus lens. It's capable of producing 3280 x 2464 pixel static images, and also supports 1080p30, 720p60 and 640x480p90 video recording. It is connected to Raspberry Pi by way of one of the small sockets on the board upper surface and uses the dedicated CSI interface, designed especially for interfacing to cameras.



**Fig 5.3 Raspberry pi camera module**

## To execute the project:

1. Ensure that all the dependencies mentioned above are met and the Nvidia GPU is active and configured to run neural network computations.
2. Deploy the Server code at server side and raspberry pi code at raspberry pi side.
3. Connect a server machine (The computer with Linux and the GPU) and the raspberry pi to the same wifi network.
4. Set the paths, system passwords, raspberry pi passwords and username, mysql password mentioned in main.py in raspberry pi and django\_school/classroom/MarkAttendance.py file in the server end.
5. Set the mysql password in the settings.py file.
6. Open mysql using the login credentials through command line, and create a database asistencia.
7. Find out the ip address of raspberry pi and set it in the django\_school/classroom/MarkAttendance.py
8. Ensure SSH permissions between raspberry pi and the computer are set up, and test it by executing from server system:  

```
ssh user_name_of_raspberrypi@ipAddressOfRaspberryPi
```
9. Ensure that the virtual environment is active of using virtual environments for packages.
10. Navigate to the **Django\_school** directory in terminal and execute the following commands to initiate database and populate sample data (Note that all operations may work on class 4CSE3 in sample data, due to limited data available for that class id):
  - a. `python3 manage.py makemigrations classroom`
  - b. `python3 manage.py sqlmigrate classroom 0001_initial`
  - c. `python3 manage.py migrate classroom`
  - d. `python3 manage.py migrate`
  - e. `python3 manage.py shell`
  - f. After a shell opens execute:
    - i. `from django.contrib.contenttypes.models import ContentType`
    - ii. `ContentType.objects.all().delete()`
    - iii. `quit()`



g. `python manage.py loaddata datadump_1.json`

11. Execute command “python3.5 manage.py runserver 0.0.0.0:8000”
12. Open the browser and go to the url 0.0.0.0:8000 from server machine, or the server-machine-ip:8000 on any device connected to the same network to access the website.
13. The website is intuitive and self-explanatory and can be operated with ease.
14. To train for newer images, please use the train.py file under Django\_school/classroom directory, and place atleast 10 – 15 images **of a student's face only** in Django\_School/classroom/classId/RollnumberOfStudent, and run the command, with the directory in terminal as /Django\_school/classroom/:  
`Python3 train.py -dataset [classId]`

**Note** that while marking attendance, after entering class id and submitting, the raspberry pi is called for photo capturing and face recognition also happens, and may take upto a minute to finish the operation till it moves to the next page where students accidentally marked absent and other exceptional case students can be given attendance.

**Note** that, for elective subject such as IRS, mapped to 4CSE3 and 4CSE4 where professor is GnanaPrakash, the class id can be given as subject code ie IRS, and the system automatically checks for 4CSE3 and 4CSE4 students. It can also be done manually for each student.

**Note** that, to add new users (professors or students), the respective tables in mysql have to be manually updated by the admin. Simply signing up in the website without the database records of the user in the student tables and professor tables will not authenticate the user account to access the functionality, to prevent any person to access the website features.