



**CHRIST**  
**UNIVERSITY**  
B E N G A L U R U , I N D I A

Declared as Deemed to be University under Section 3 of UGC Act 1956

# Database Design Document

**Title: Feasible Real Time Helmet Detection Using  
Raspberry PI**

Submitted By:

Lynford Valentino D'souza 2047221

Yash Himmat Kataria 2047235

Sumi Thomas 2047259

## I) Block Diagram of Project:

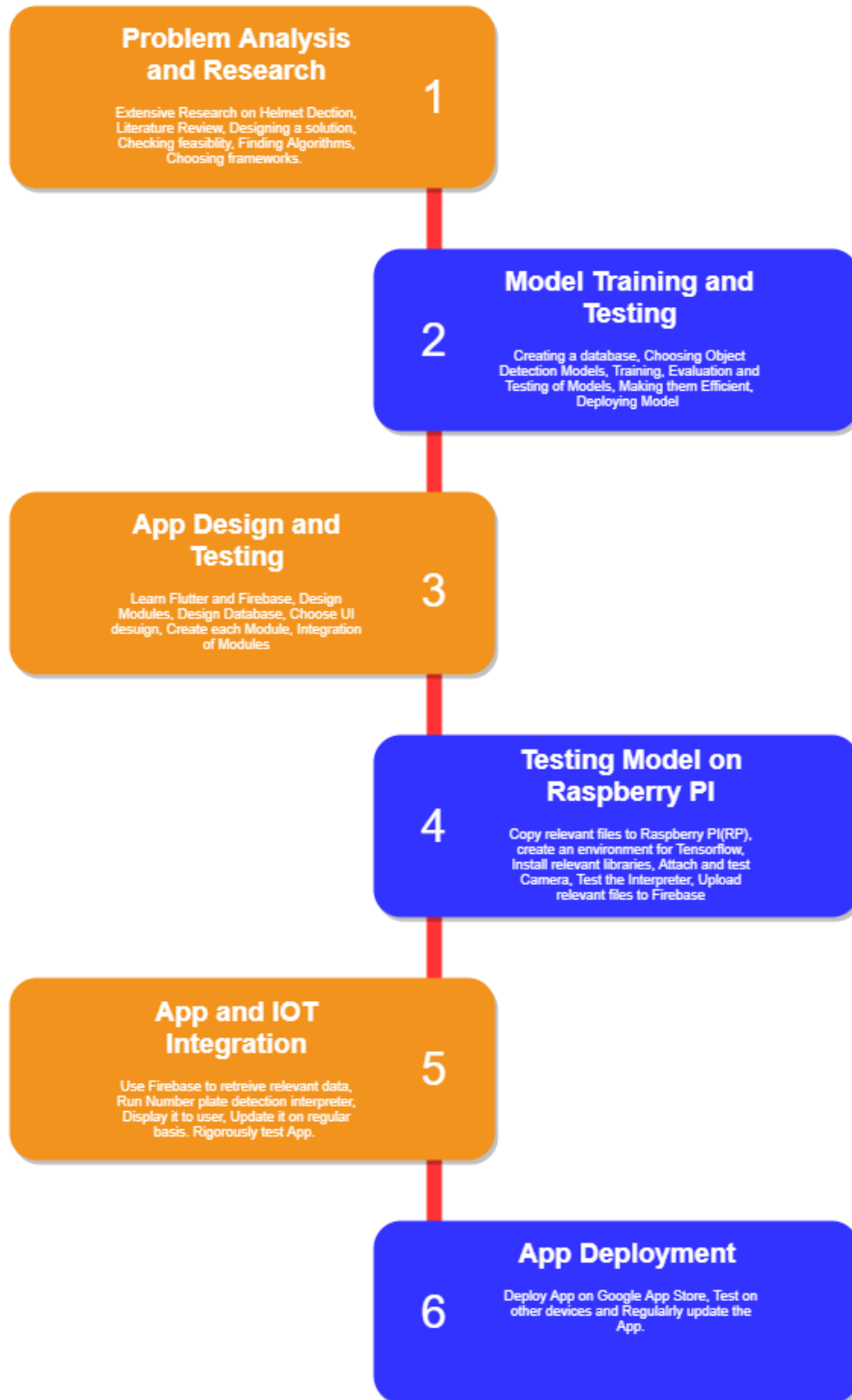


Figure 1: Main Project Block Diagram

## II) Data Collection and Refinement:

### A) Data Collection:

The search for a dataset for the problem i.e., Helmet Detection began online. Only one dataset was found which was relevant to the problem, which was from IIT Hyderabad. But downloading that dataset needed special permission and therefore it was not available to us. So, a custom dataset had to be made. In total 3 datasets were prepared, out of which dataset 3 was used due to the Object detection model's excellent performance after using it for training.

The first dataset was basically a set of 1080p videos at 30fps which were recorded on the Oppo Reno Smart Phone. The footage was recorded at a single location from 2 different angles. After recording the footage, screenshots were clicked which contained the required information as shown in Figure 2. This made up the first dataset.



Figure 2: Sample from Oppo Reno Video of Dataset 1

The second dataset was made using the Nikon D5500 DSLR. Each photo has a resolution of 24.2 MP. Photos were clicked from 3 different locations. Only on the first location photos were clicked from 2 different angles. From the other 2 angles photos were clicked from one angle. A sample image is shown in Figure 3. For the third dataset, pictures were clicked from a single location and 2 different angles a sample of which is shown in Figure 4.



Figure 3: Sample from Dataset 2



Figure 4: Sample from Dataset 3

For the final working model 302 training images and 71 test images were used.



## B) Dataset Labelling

For this step a python script developed by Darrenl tzutalin [1] was used. It's free and open-sourced code designed to label Image data for Object Detection. It has a user-friendly UI with which we need to tag the objects in the image as shown in Figure 5 and 6. Using this software around 800 images were labelled. Each image having anywhere from 1 to 20 labels depending on the approach used. LabelImg saves a .xml file containing the label data for each image. These .xml files will be later used to generate TFRecords, which are one of the inputs to the TensorFlow trainer.

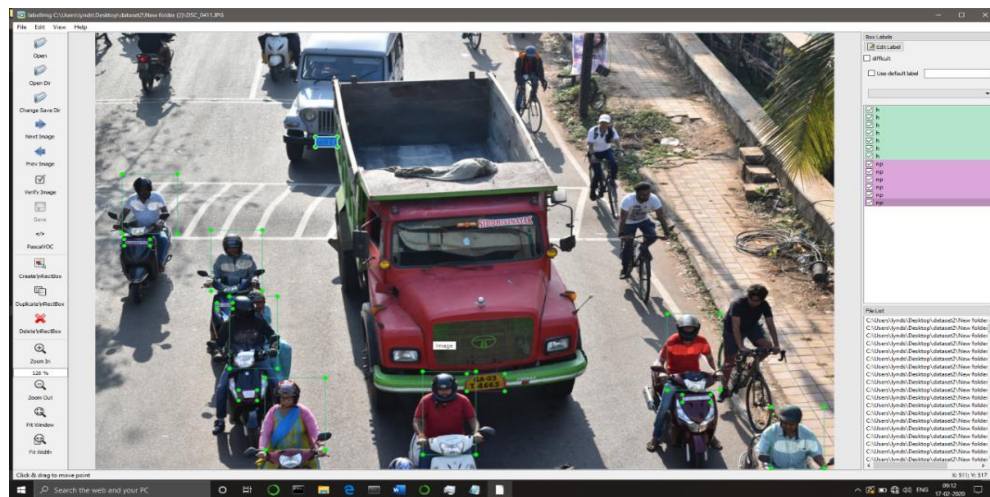


Figure 5: Labelling a sample from a Dataset 2

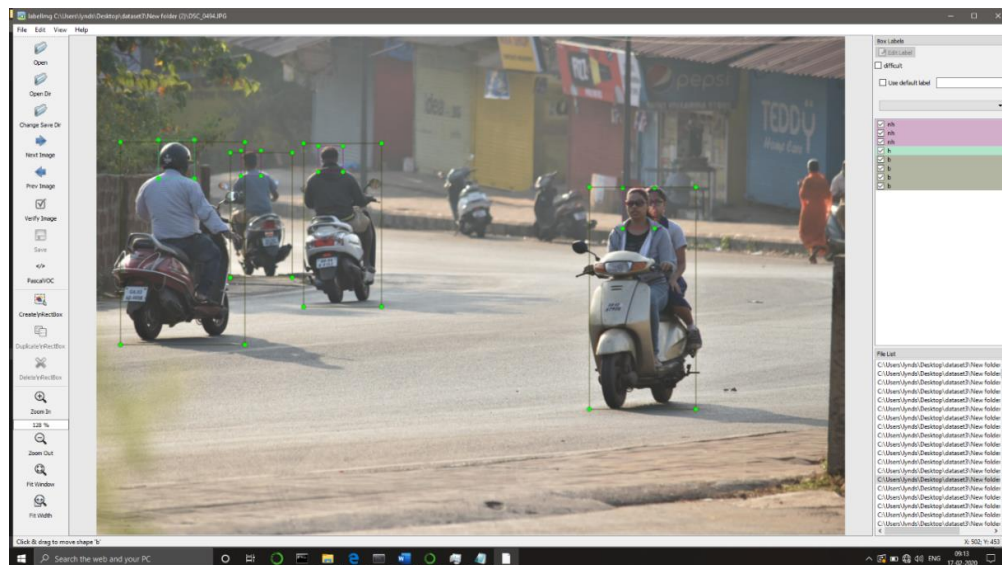


Figure 6: Labelling a sample from Dataset 3

## C) Dataset Pre-processing

First the Images were resized using `resizer.py` written by Evan [2] so training becomes much easier and all the data is normalized. To generate the TFRecords that serve as input data to the TensorFlow training model, we use `xml_to_csv.py` and `generate_tfrecord.py` scripts from Dat Tran's Raccoon Detector [3]. The image .xml data will be used to create .csv files containing all the data for the train and test images. The `generate_tfrecord.py` was then edited with the required labels and the appropriate ids. After running the script, we should have all the necessary TFRecords files for training the model.

## III) Module Block Diagrams:

### A) Object Detection Model and Image Capture:

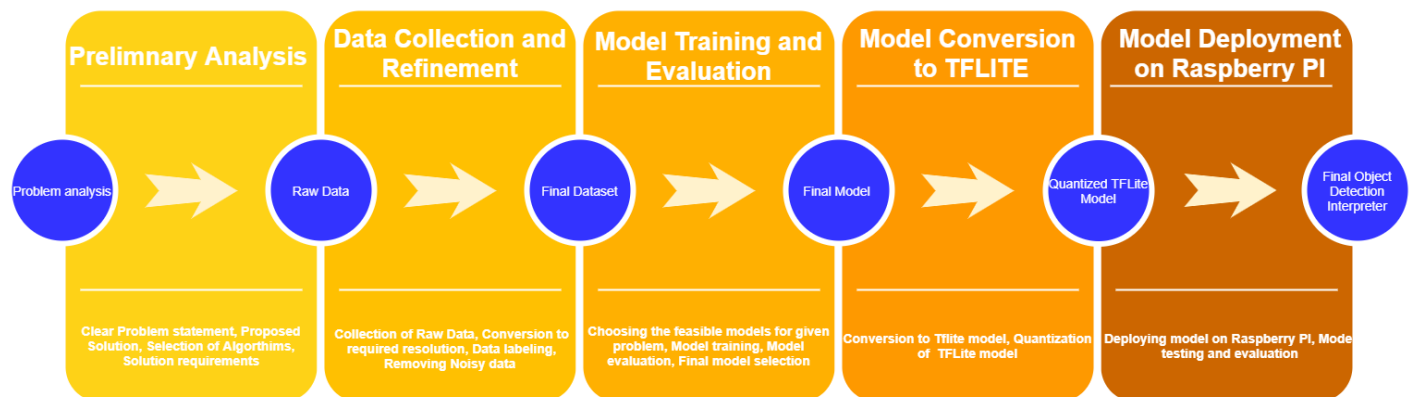


Figure 7: Object Detection Process

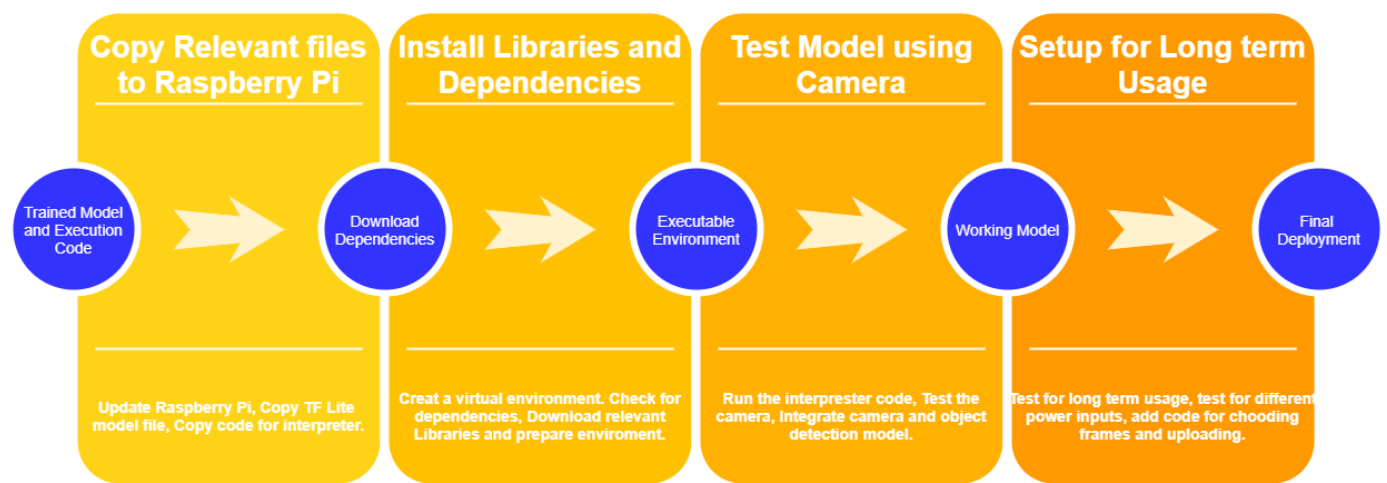


Figure 8: Image Capture Process

## B) Main Mobile Application:

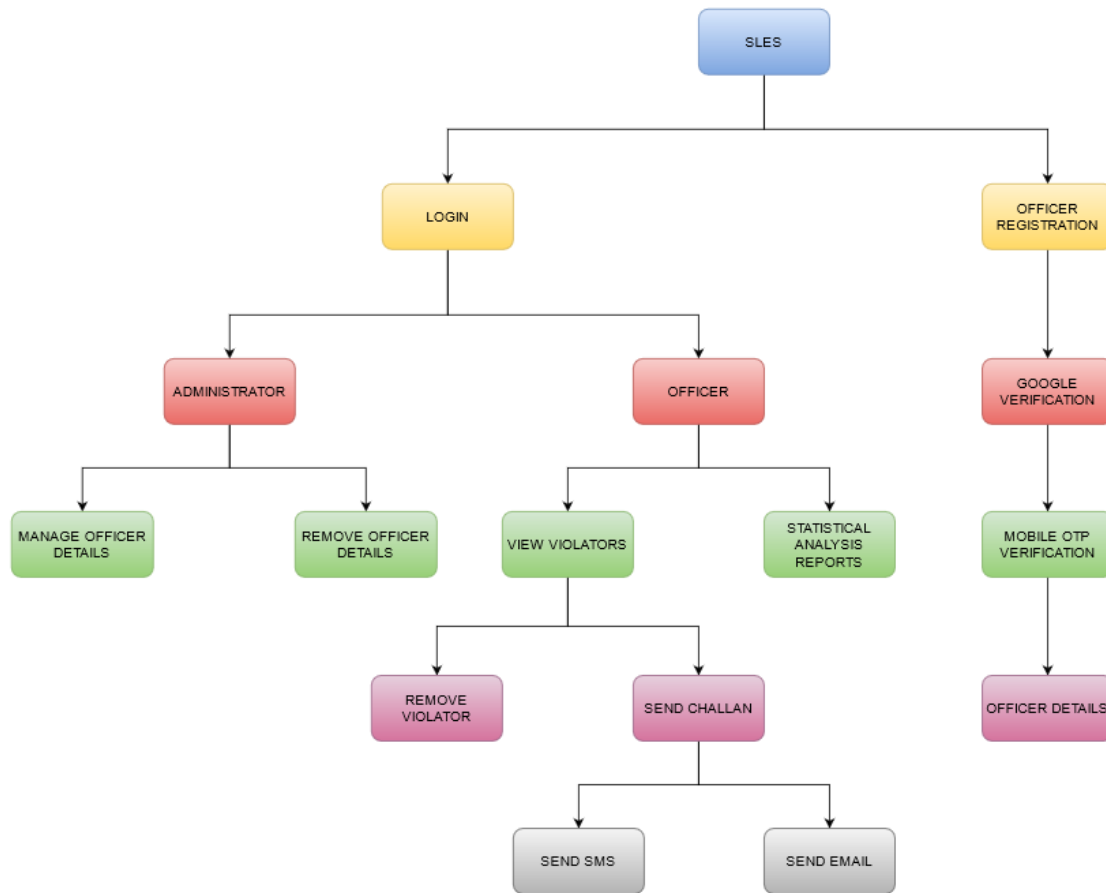


Figure 9: Main Application Flow Diagram

## References:

- [1] <https://github.com/tzutalin/labelImg>
- [2] <https://github.com/EdgeElectronics> by Evan
- [3] <https://github.com/datitran>