Title – Detection of face mask and helmet using computationally efficient neural network.

Team Member and Student Id – Kunal Sanghvi, 011809708

Abstract – This is a company-sponsored project. I will be in touch with a company named UtopiaTech Pvt. Ltd. in India and solve a problem statement that will be a component of their final product. The motivation behind choosing this topic is that I want to develop some applications that can solve real-world problems. The unique advantage my project will provide is that it will identify the best neural network for face mask detection, which has an almost low computation cost and can be easily deployed on hardware such as the Raspberry Pi. This project will help me thoroughly understand every concept of neural networks.

Introduction – The development of this project will be from the perspective of a real-world application. This application is going to be installed in an ATM facility. The camera will always be at the top-end corner of the facility, so achieving a similar kind of dataset will be a big task. Yet, we expect to generate our dataset, which can be as close as possible to the real-world setting. I will be in touch with the co-founder, Mr. Mitesh Bajaria, regarding the project, which in turn will help me with any support required in terms of hardware or software.

There are multiple models available, such as Mobilenet SSD and Yolo, with several versions along with a tiny version to deploy on hardware. However, these models provide less accuracy at a high computational cost. Hence, my aim while developing the project will be to identify the best configuration of models that provide the maximum accuracy and have the least computational cost when deployed on hardware, thus giving us a fair FPS to process and work on.

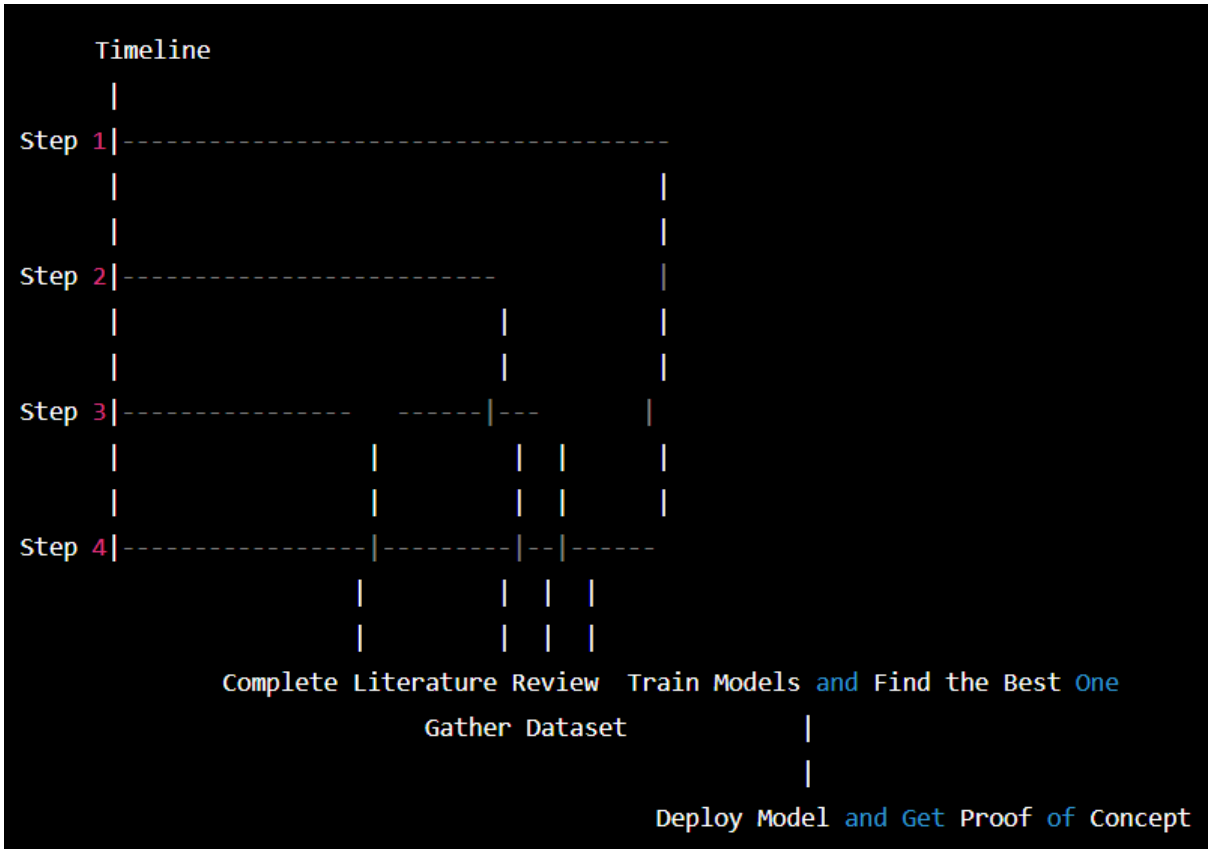
The plan is divided into four steps as follows:

1. Literature review to identify all the possible models which can be used for the application.
2. Generate and gather the dataset.
3. Train a bunch of models and find an accurate setting that has the most accuracy and least computational cost.
4. Try to deploy the model on a hardware setting and get a proof of concept of the application.

Once this application is developed, we can increase our dataset with multiple applications, train our model, and compare its efficiency. Thus, this will create a complete use case for a real-world application.

Literature Review and Technical Plan – Since I am in contact with the company, we are going to set up a couple of meetings where we will decide on our complete technical plan and also do some important literature reviews that will be key beneficiaries of the project. A basic outline of the plan is described in the introduction section, yet once the discussion takes place, a more in-depth plan might be formed.

Sample Technical Plan –



The horizontal axis represents the timeline of the project, and the vertical axis represents the four steps of the plan. Each step is represented by a horizontal bar, with the start and end points of the bar indicating the start and end of the step in the project timeline.

Step 1, Literature review, is the first step and is represented by a single bar. Step 2, Generate and gather the dataset, begins before step 1 is completed and ends before step 3 begins. Step 3, Train a bunch of models and find an accurate setting, overlaps with step 2 and starts before step 2 ends. Step 4, Try to deploy the model and get a proof of concept, which starts after step 3 is completed.

Intermediate Results – We had our first brainstorming meeting on 3rd March 2023. In the meeting, I got connected with the Co-Founder of the company along with the team with whom I was going to work. In the meeting, we discussed important points like:

- What do we want to achieve?
- What is our plan of action going to be?
- Setting GitHub repo to share and manage the code among the team.
- Reading literature based on the work divided.

We then decided on two approaches to take this project forward which are as follows:

- The first approach is to use pre-trained and defined models by TensorFlow, XNOR, PyTorch, Darknet, MXNet, and Caffe.
- The second approach is to develop your own model and train it with your custom dataset.

While exploring the first approach we found that the yolo v3 model with its tiny version has considerable accuracy in comparison to the others. But there is a big problem that we are still facing we are not able to detect accurately the helmet hence we feel there is an insufficient dataset and hence due to this the model is underfitting and not detecting it with enough accuracy. Also, the problem we are facing is the fps is between 1.5 to 2 while deploying it on raspberry pi. Hence, we are trying to solve both problems.

While exploring the second approach there is nothing significant success was found but we still haven't ruled this option out rather trying to develop something of our own, which I know will take a long time and extensive research but still, I am trying to speed up this process.

So, regarding the technical plan, which was shared earlier, we did go through Steps 1, 2, and 3. So currently we are at a phase where we are training models and trying to achieve maximum accuracy and eliminate the problems, we are facing one of which is false detection.

A sample output of model detecting helmet is shown in the video - https://emailwsu-my.sharepoint.com/:v:/g/personal/kunal_sanghvi_wsu_edu/Eetsbdc5Z6NDpVyugV94c9AB8HVbCYhGzsmA3pzpY5Ulvw?e=aOf5WE

This input to the model is a CCTV camera and its output is displayed on the television which is recorded by one of our onsite interns.

Future Work –

The plan now is to still explore models and train and test them to achieve better accuracy and FPS when deployed on raspberry pi and simultaneously work on creating our own model which will consume our maximum time. Yet the output which we are expected is two deploy our trained model on raspberry pi which is connected to an IP CCTV camera input feed and determine mask and helmet as the output and take necessary actions based on the findings.