Live identification of vehicles with fake number plates using Machine Learning Algorithms

Regional Remote Sensing Center, West

Team members:

- Kshitij Upadhyay
- Akshat Agrawal
- Aman Sharma

Introduction to the problem statement

In India and most parts of the world, it is mandatory for all vehicles running in public places to have a number plate with their unique registration number. Unfortunately, number plates are often stolen or faked by people trying to conceal criminal activity. A system that can identify fake number plates on vehicles by processing the data from police cameras can be of great help for law enforcement in dealing with crime.
The system has to be able to read the number plate on a vehicle. The details of the vehicle with that number plate have to be obtained from the m-Parivahan API. At the same time, the system has to recognize the color and model of the vehicle. The details have to be matched, and an alarm has to be generated in case of discrepancy.
The system has to be fast enough to process data in real-time. In addition, since the amount of data to be processed is massive, accuracy must be reasonably high to avoid frequent false positives.

Proposed solution

We will read number plates using **Automatic number-plate recognition** (ANPR). ANPR uses a series of image manipulation techniques to detect the number plate and then optical character recognition (OCR) to extract the alphanumerics of the license plate. We will use a combination of TensorFlow Object Detection with OpenCV to detect the number plates and PyTorch and EasyOCR to extract the text out of it. We will also check the database of the RTO using m-Parivahan API to get the details of the vehicle associated with the number extracted from the number plate. Finally, we plan to use transfer learning, where a pre-trained model is used to solve the problem under consideration.
Meanwhile, we will also predict the model of the car using Deep Learning frameworks deployed on camera feeds, namely Tensorflow Object Detection Models (Tensorflow Model Zoo) and OpenCV. We are using a Kaggle dataset for training our model.

We will compare the two results, and in case of a mismatch, alert the nearby police station or headquarters for every mismatch.[Fig1]

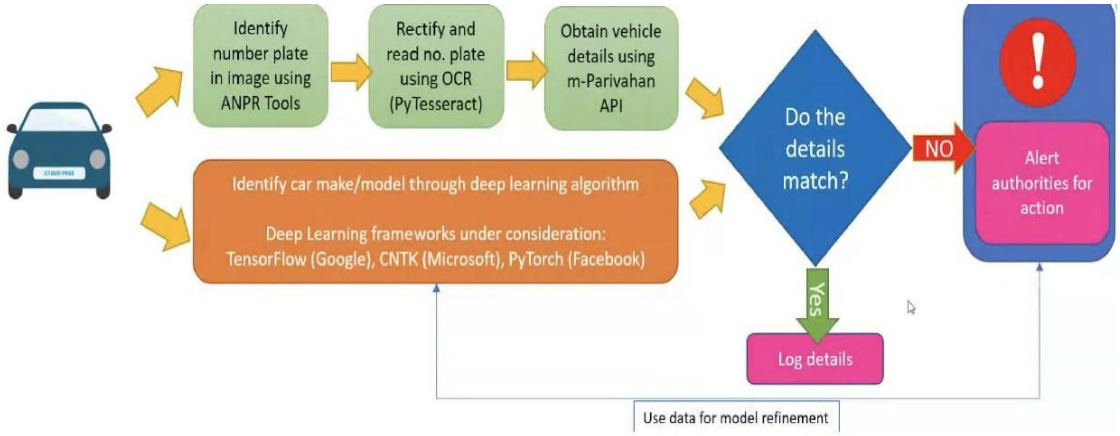

Fig 1: Methodology

Progress

We have gone through Youtube tutorials to learn about Tensorflow and its usage. We have also developed a hand gesture recognition model to implement the learning on a less demanding problem in terms of the training set. [Fig2] It recognizes the ‘thumbs up’ and ‘thumbs down’ gestures.

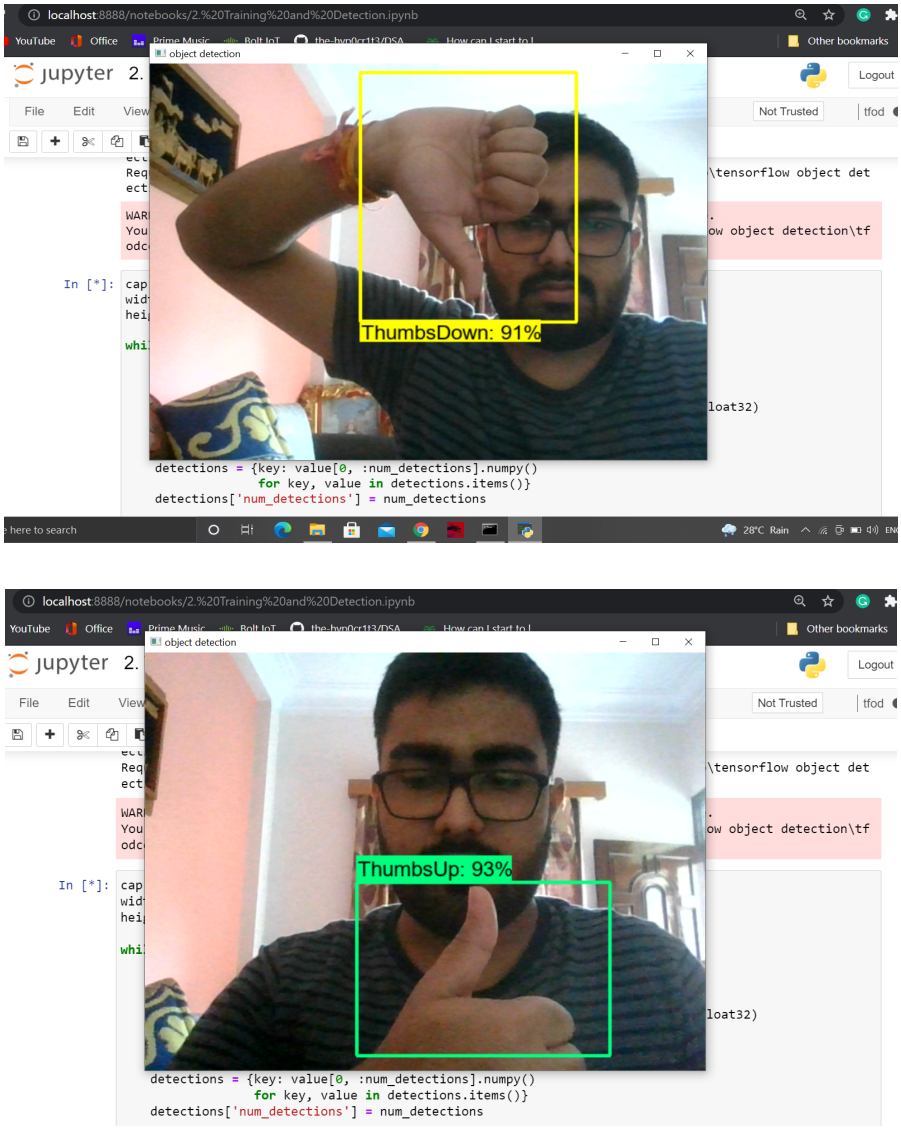

Fig 2: hand gesture prediction model

Later on, we have also extended the learning to the actual problem we have to solve. We have made a model that predicts the exact location of the number plates from the given image. We used the online available Kaggle dataset [1]. Attached are the pictures of the performance of the model on test images.[Fig 3]

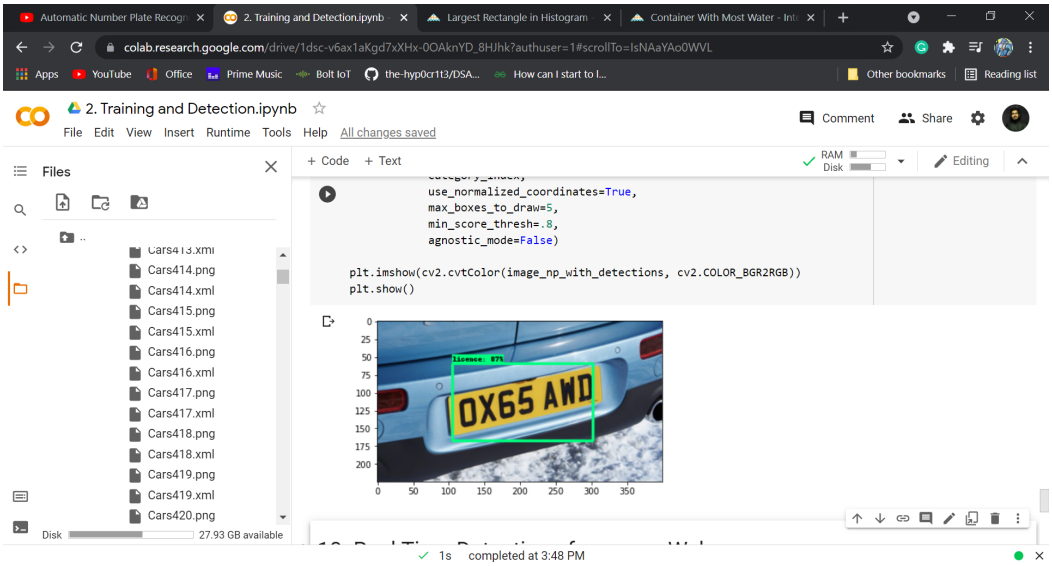

Fig 3(b): Detecting number plates on test images

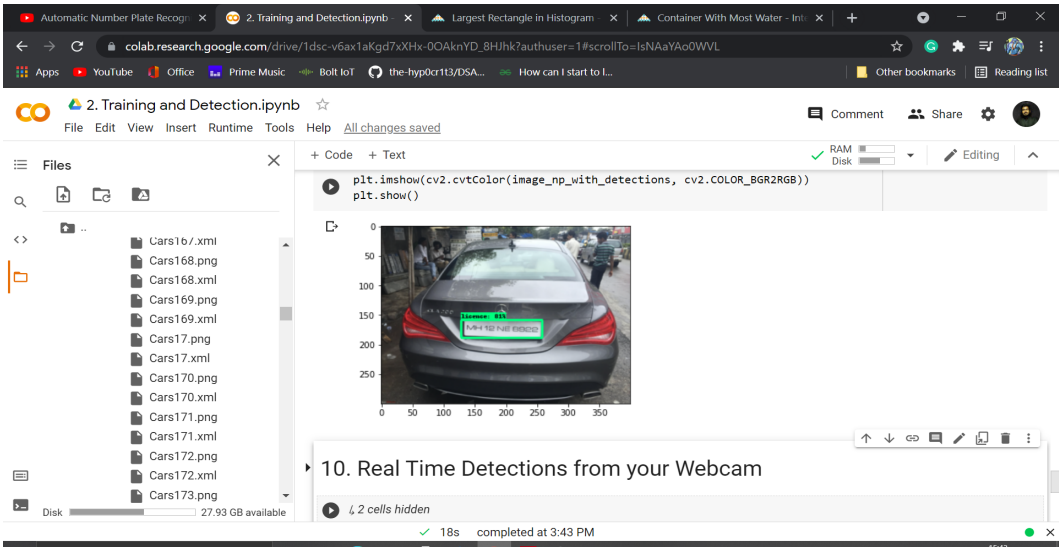

Fig 3(b): Detecting number plates on test images

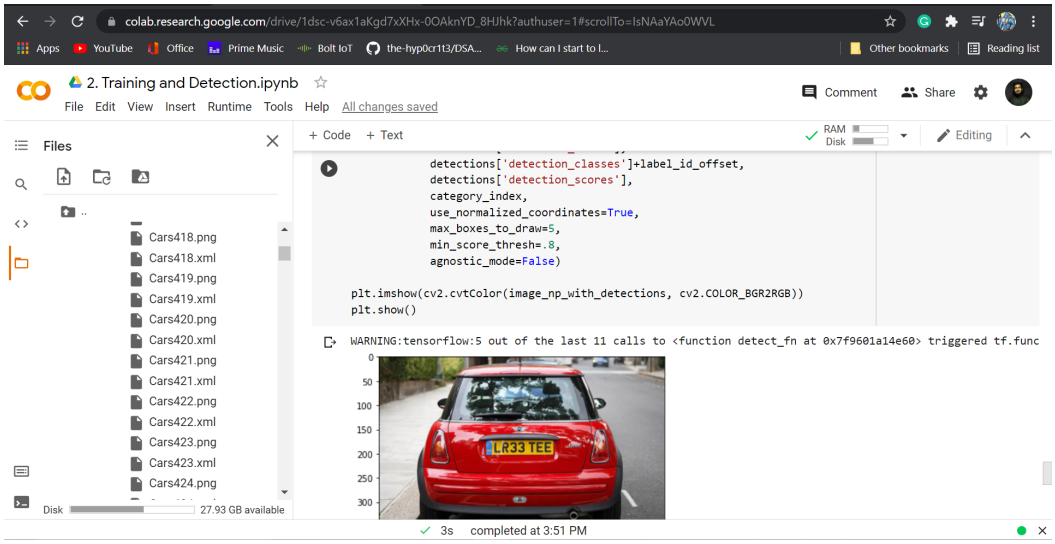

Fig 4: Detecting number plates on test images

Challenges

The biggest challenge so far has been to train our model. We do not have the latest GPUs in our laptops, and if we are trying to train it locally, it takes about 4 hours to train the model with 10000 steps.
Using online platforms was the option we have restored to, and the results mentioned above are run on Google Colaboratory. However, they come with their restrictions. For example, we cannot test our model on the live feed in these platforms, which fails the project's purpose. Moreover, these platforms have their time limits and clear the progress once the boundary is crossed.

Also, our Model performs poorly on taxi number plates. [Fig 3] Lack of training examples with images of yellow number plates is the reason for the same. To overcome this problem, we are trying to find datasets with more images with taxis, i.e., yellow number plates.

Plans for the remaining four weeks

We have started with an object detection course where we detected hand gestures (live, when using Jupyter Notebook, and with images when using Google Colaboratory), then used that model to detect where the number plate is in the image. Next, we will extract text/license numbers from the number plate using Optical Character Recognition(OCR). Finally, after successfully pulling the license number, we will recognize the car's model and then match the license data with the RTO data using m-Parivahan API.
After we have all the data, we will report the mismatch to the corresponding stations.
If time permits, we also plan to develop a website where people can upload their car's image and verify with our model, giving better results and avoiding their hassle due to an error that misclassification by the model might cause.

Limitations

Challenges that the model might face:

- Non-standard number plates.
- Vehicles with damaged/modified bodies might be misclassified
- Weather and poor lighting conditions might substantially reduce the quality of the camera feed, making it difficult for our model to classify correctly.

References

1. https://www.kaggle.com/andrewmvd/car-plate-detection
2. Tensorflow 2.0 crash course
3. https://www.coursera.org/learn/machine-learning