Vellore Institute of Technology
(Deemed to be University under section 3 of UGC Act, 1956)

PROJECT REPORT

DIGITAL IMAGE PROCESSING [SWE1010]

Vehicle Number Plate Detection and Speed Estimation

Team Members:

Registration Number	Name
18MIS1029	Vaishnavi S
18MIS1059	Balaji Chandramouli

Abstract

In a huge country like India with a high population on the roads and dangerous conditions, there is a need to automate a lot of functions to assist the traffic police. Hence, the function of calculating the speed of all vehicles that travel on the road becomes necessary. Due to the significant expense of radar and less precision, the radar system is not able to become popular in the traffic surveillance system. Therefore, we propose an image processing approach to calculate the speed of vehicles that are monitored through video cameras placed strategically.

Problem description and background

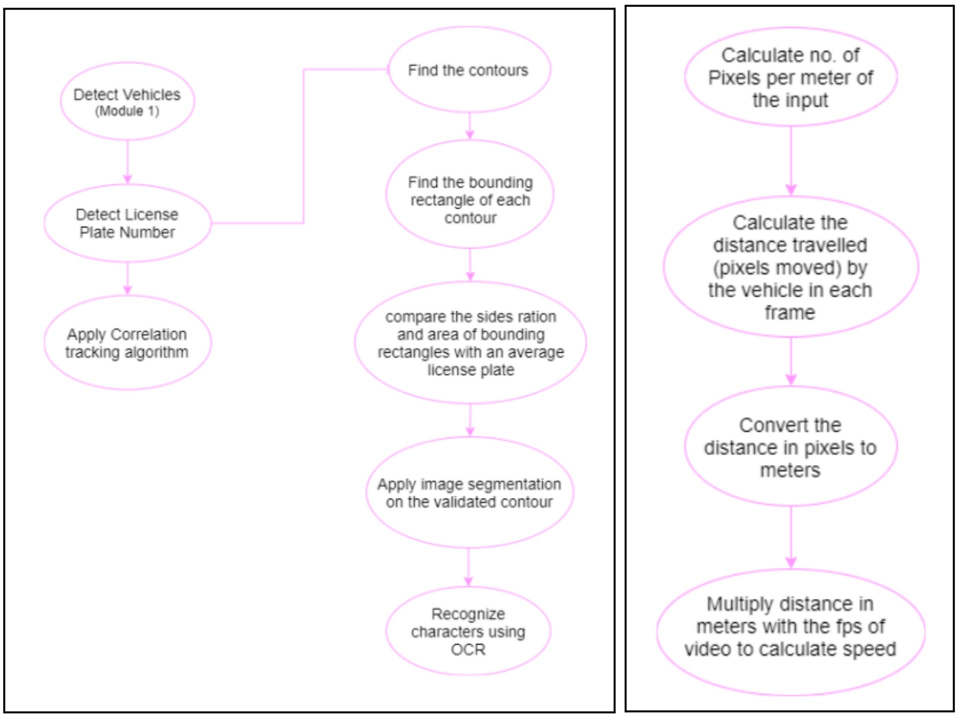
Most of us have experienced or witnessed unsafe and fast vehicles driven by distracted drivers who ignore speed limits, despite signs. In 2019, over 4 lakh cases of traffic accidents were reported, out of which a good number was due to overspeeding[1] and the number of road deaths has been increasing every year with more than 150,000 deaths in 2019 [2]. Therefore, vehicle speed monitoring and control is a very important issue for traffic management. In a large developing country with increasing urbanization like India, it is impossible to manually monitor the behaviour of vehicles across all roads. One method that was being used for this purpose was the Doppler radar. Doppler radar was employed to measure the speed of moving vehicles. It was a hand held device which sent a radio beam to a moving vehicle and then calculated the vehicle speed by measuring the change in reflected wave frequency. However, it had some problems such as Cosine error, radio interference, shadowing, etc. Hence, there is a need for more modern methods that rely on digital image and video processing to solve this problem.

Proposed system architecture

The high level design or system architecture involves two major modules:

1. License Plate Recognition and Detection
2. Speed Estimation

The input video is passed to the program which detects the moving vehicles in the frames. The program then scans for possible license plate structures and recognises it using an Optical Character Recognition algorithm. Finally, using the distance moved by the object(vehicle) in subsequent frames, the speed of the vehicle is calculated. The following flowchart illustrates the process of each module:



Module description

1. **License Plate Recognition and Detection :**

The first step here is detecting the number plate location from the input. This is done using a pre-trained Cascade Classifier.Cascade classifiers are used to detect specific regions or features in an image. Said feature in this case is the license plate. After applying the classifier, the contours in the license plate image are found and rectangles are created around them. Contours are basically a closed line/curve joining all the continuous points along the boundary and having the same intensity. Then, cropping and extraction of the characters will be performed.

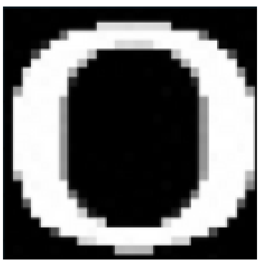
After the ROI is identified using the cascade classifiers, the process of Segmentation is performed to separate the characters in the preprocessed license plate image for further analysis. To achieve segmentation, the following processes are performed:

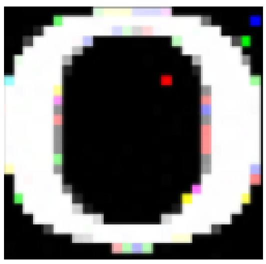
Cropping : Cropping can be achieved by resizing the image to display only the plates.
Thresholding : Thresholding is done by converting the BGR image into gray and then setting a threshold value for the pixel to be white.
Erosion : Erosion is the process of thinning the contents of the image. It is done to reduce the white noise generated by unwanted elements in the plate.
Dilation: After eroding the noise, we can use dilation to thicken the thinned contents. This will result in more accurate classification.

After Segmentation, OCR will be performed by a pre-trained CNN model. An augmented image data is generated from a dataset using Keras ImageDataGenerator and a sequential model is created. It is done by putting layers on top of each other in a sequence. conv2d() layers are added with the help of ReLU activation function. Other layers - MaxPooling, Dropout, Flatten, Dense and Compile. After training, the model is fit with parameters like epoch, callbacks, training and validation data.Finally, the OCR model will predict the number plate characters from the labels by trying and matching the characters. If the match is found, the character will be appended to the final output and displayed.

A file named indian_license_plate.xml file provides the neural network specifications to train the model. This data is taken from Kaggle.The Neural network problem specification in XML provides an easy-to-specify approach for describing the TensorFlow graph. For any change in problem parameters, only the XML needs to be changed. The core TensorFlow program remains untouched. This avoids problems caused due to depreciation (seen “initialize variables” depreciation warnings).
The Training data has 864 images and 36 classes (0-9)+(A-Z)
The Validation data has 216 images and 36 classes (0-9+(A-Z)
For classifying a license plate for training, videos recorded in a mobile phone (1080*720 at 30 fps) are used.

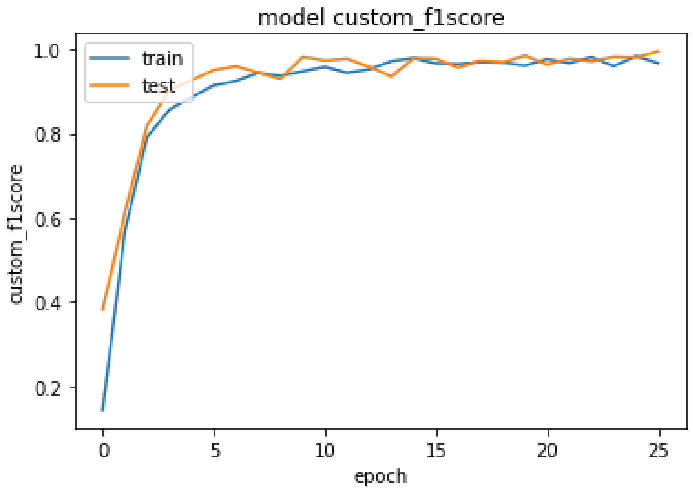
Data Augmentation : Data augmentation was performed to double the size of the dataset. Poisson noise was added to each data and the training of the model was done again. This time, it was observed that the convergence happened more quickly than the previous model. A sample of the original and augmented data is shown below.

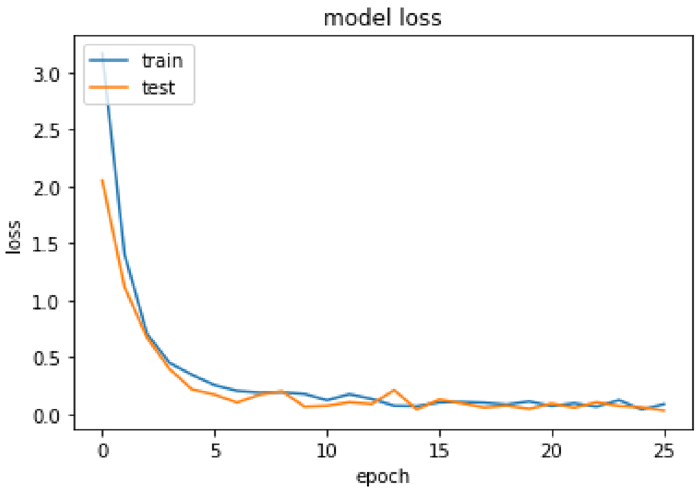


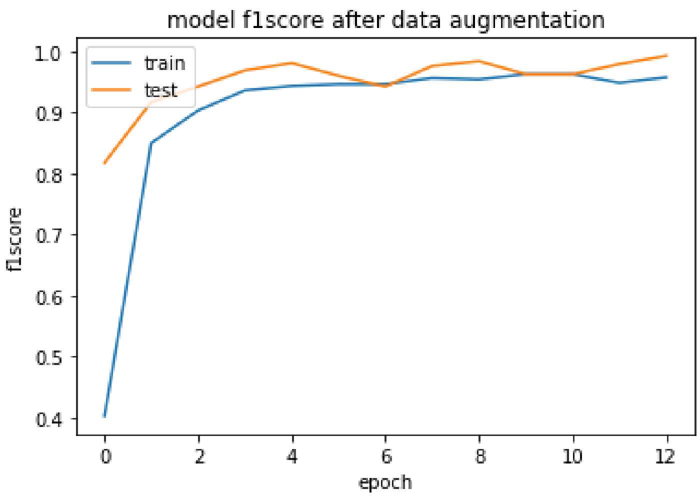


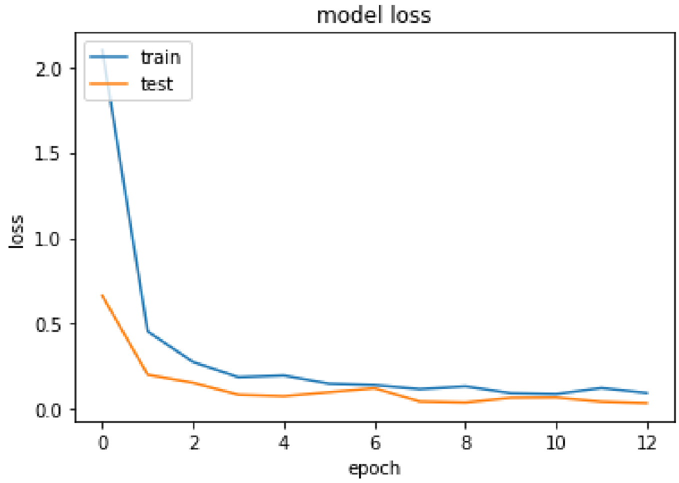
Comparison of the models before and after augmentation with various metrics

From the plots below we can see that the convergence happens quickly after data augmentation at the 13th epoch while the unaugmented data model takes 26 epochs. The metric used for the plots below is f1 score.

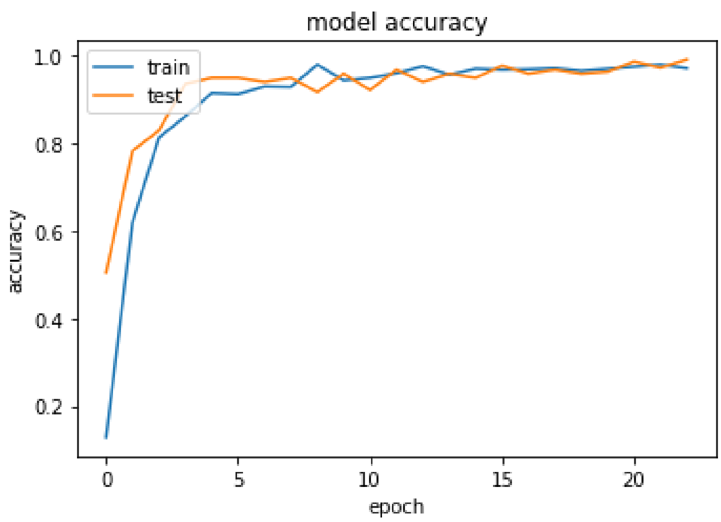


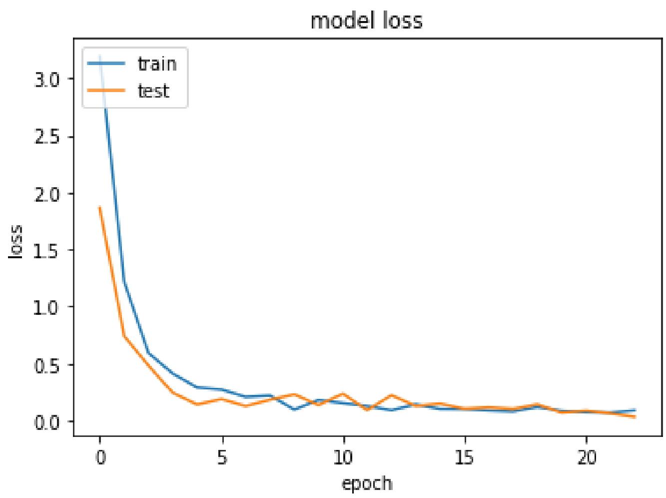


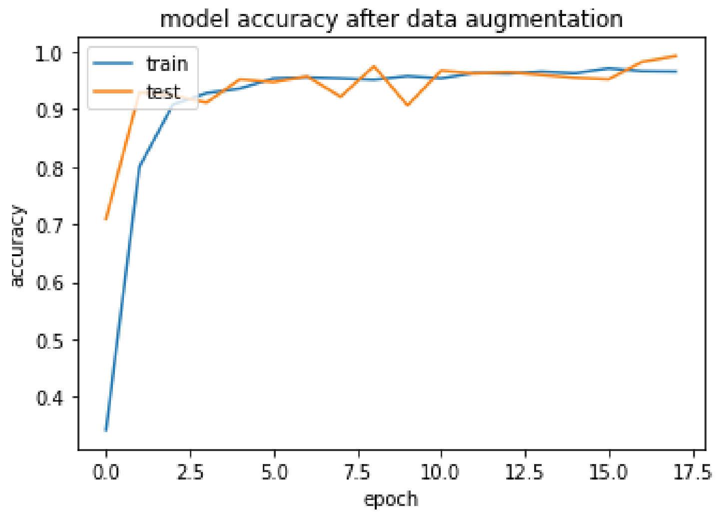


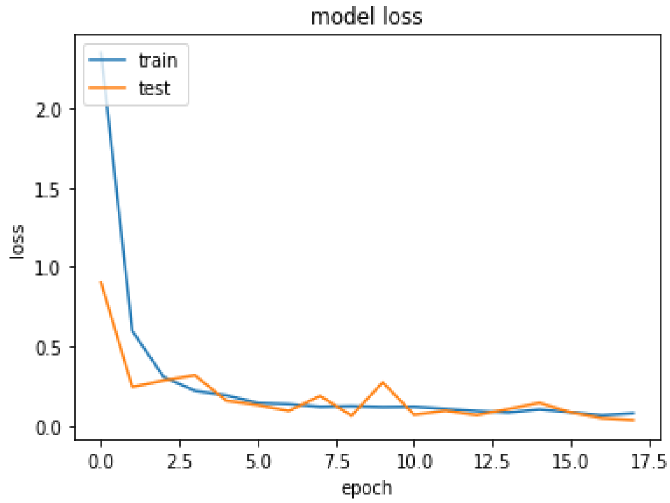


Similar plots were obtained for accuracy where we can see that the convergence for the augmented model took 18 epochs while the unaugmented model took 23 epochs.









Speed Detection

Unlike detection of license plates, speed estimation requires more than one image(frame) to compute. A Haar Cascader will be used to identify vehicles in the frame. After the vehicle is detected using the contour, a correlation tracker is set for each contour in the corresponding frames to track the vehicle's movement. This is done using an inbuilt function.

Now, we can estimate the speed of a vehicle using the following steps:

- Calculate the width/length of the road in real life. (Estimation through observation)
- Calculate the same in digital video in the form of the number of pixels.
- Divide the number of pixels by the actual width to find the pixels per metre (ppm).
- Subtract the pixel difference in the consecutive frames and multiply it by the ppm value.
- Since we know the distance travelled and time taken (in the form of frames per second), we can use the formula: Speed = Distance/Time

After calculating the speed, the output will be displayed near the contour box drawn around the moving vehicle. Since the speed of the vehicle might change over time, the final output will be given by the mode of the various speeds (maximum occurring element) that occurred during the span of the video.

Note that the speed calculation depends on a number of factors such as:

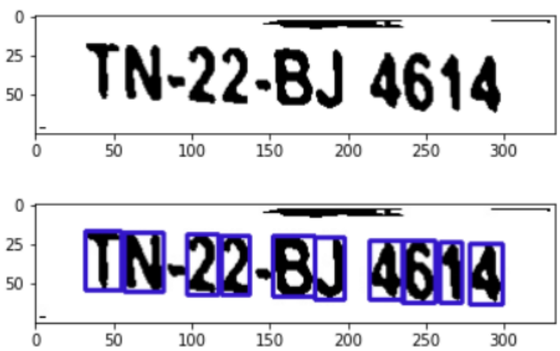
- Dimensions of the road
    - We have to make sure that the dimensions of the road are calculated accurately so that the pixel per meter calculation is precise.
    - Since the ppm value depends on the length/width, this process needs to be performed uniquely for every road where the speed calculation takes place.
    - The width of the road used in this project is around 9 meters.
- Angle of the camera
    - The angle at which the camera focuses on the vehicles/road is another important factor. When the position of the camera gets lower relative to the position of the ground, there will be some cosine error associated with it.

- The perfect angle will be either 90 degrees on top of the road which would give us a two dimensional structure, or sideways position with respect to the movement of the vehicles.
    - However, neither case would allow us to perform the number plate detection process.
    - Therefore, we had chosen to take the input video at about equal height above the ground compared to the movement of the vehicle, and we had appropriately tweaked the pixel per frame value.
- Quality of the input video
    - Finally, the quality of the input video should be high.
    - No disturbances or noise should persist throughout the course of the video.
    - The input video used in our demonstration has resolution 1920 x 1080, and is captured at 60 frames per second.

Results and discussions

- The number plate of the vehicle has been correctly identified and printed in five frames. It is also the mostly likely occurrence, therefore it is presented as the final result.

The number plate is detected in the video



TN228J4614

After preprocessing, the contours are segmented and passed to OCR and the output is predicted.

- The speed of the vehicle is approximately calculated, and is also the most likely occurrence in the video. It is also presented as the final result.

The moving vehicle is detected and the speed is predicted

- The final results are thus printed:

```
Number plate of vehicle is identified as: TN22BJ4614
Speed of vehicle is calculated to be: 5.236363636363636km/hr
```

Some notable observations from the results are:

- Both the license plate tracker and the speed estimator have detected more than a few false positive segments. This might be due to the inherent settings in the multiscale detectors of the corresponding cascaders. The cascaders used were pre made and taken from the internet.
- The actual speed of the vehicle in the input video is around 5 kilometers per hour. This would imply a deviation of 5% from the predicted value.
- The process is still susceptible to external disturbance/noise generated. This can be improved by preventing occlusion.

Conclusion

The process of speed monitoring is now done by fully automated intelligent systems with improved algorithms like yolo, however this project uses various concepts of image processing like segmentation, thresholding, morphological transformations etc. The usage of data augmentation and the impact of it on training the model has also been discussed. Since roads and number plates vary in sizes, this algorithm can be further improved to be used for all kinds of environments.

References

[1] https://ncrb.gov.in/sites/default/files/Chapter-1A-Traffic-Accidents_2019.pdf

[2] Nagaratna M Raikar , Dr. Megha P Arakeri, 2020, Development of Vehicle Speed Estimation Technique using Image Processing, INTERNATIONAL JOURNAL OF ENGINEERING RESEARCH & TECHNOLOGY (IJERT) Volume 09, Issue 09 (September 2020)

[3] The International Journal of Engineering and Science (IJES) || Volume || 7 || Issue || 9 Ver.III || Pages || PP 64-72|| 2018 || ISSN (e): 2319 – 1813 ISSN (p): 23-19 – 1805 - Vehicle Speed detection by using Camera and image processing software

[4] https://www.statista.com/statistics/746887/india-number-of-fatalities- in-road-accidents/