INTELLIGENT TRAFFIC MANAGEMENT SYSTEM USING YOLOv5

Polamarasetti Praneeth*(1), Kandregula Reshma(1), Mir Abdul Moinuddin(1), Moka Anupama(1), UG Scholar, Department of Computer Science & Engineering,
T.V.Madhusudhana Rao Professor, Department of Computer Science & Engineering,
Vignan's Institute of Information Technology (A), Duvvada, India
*polamarasetti.praneeth@gmail.com ; madhu11211@gmail.com

Abstract- Intelligent Traffic Management System using YOLOv5 is a system which detects the type of vehicle and uses number plate to identify the detected vehicles. The objective in our paper was to design an efficient automatic authorized vehicle identification system by using the vehicle number plate and vehicle type. The major focus of this project is to enable this system in villages to provide more security and control the traffic in the village more efficiently. After detecting the vehicle its number plate is identified using the model which was trained using YOLO algorithm. After analysing the image, the model will give specific information like the number plate, type of vehicle. The identified number plate is sent to Optical Character Recognition(OCR) which will help find the number inside the detected plate. By utilizing the license number extracted from the detected vehicle it can be used to create logs, keep a vigilant eye and also for various purposes.
Keywords- YOLOv5; number plate detection; vehicle detection; Indian vehicles; optical character recognition; custom model.

I. INTRODUCTION

Due to high mobility and ability of covering areas at different locations and altitudes, remotely piloted vehicles are increasingly used in traffic monitoring and surveillance. The major challenging is to use the aerial images precisely for detecting the vehicles and to count them for traffic monitoring purposes in real-time.

Based on convolution neural network (CNN), numerous deep learning techniques were recently proposed for recognition and classification in computer vision in real time. However, the performance of these algorithms depends on various scenarios in which where they were used.

In cities there will be many CCTV cameras are fit at some places and is monitored by security guards which provides security to the people living in cities and big town. When it is considered about the people in village installing these cameras, maintaining the people for constant monitoring is difficult. So, in order to provide the security for the people in village a system is developed which does not required any human effort for monitoring. From [1]-[5] it can be seen that to detect the license plate region-based approach, Vertical Edge Detection (VEDA). In our paper, a model was trained which will automatically detect the type of vehicle and the also save the number plate of the detected vehicle which can be used to find the owner of vehicle.

II. LITERATURE REVIEW

ANPR Framework utilizing OCR at the centre of the framework is the OCR (Optical Character Recognition framework) which is utilized to extricate the alphanumeric characters show on the number plate. To do this it to begin with employments an arrangement of picture control strategies to identify and normalize and improve the picture of the number plate. There are two components within the framework, the cameras which at the front-end and the inaccessible computers at the back-end. More often than not two cameras are used at a time to extend the productivity.

Intelligent Traffic Control system helps us detect type of vehicles which are widely used in India like auto-rickshaws, lorries etc. The objective is to plan a proficient programmed authorized vehicle recognizable proof system by utilizing the vehicle number plate and vehicle type. Since there is continuous traffic, we have to consider a model which have less inference time and decent accuracy. After observing different models, the YOLOv5 was considered because of the various features it offers:

- It was implemented in PyTorch which has simpler support and easier deployment.

- It has very less inference time which makes it suitable for this paper.
- It does not compromise on accuracy to achieve less inference time.
- It also has smaller weight file which helps us deploy our model within limited space.

After carefully studying [16]-[19] the features of You Only Look Once(YOLO) can be clearly seen . Since it consumes very less time for inference it was ideal for our paper as it can be implemented in real time and the inference time plays a vital role for the real time application.

It was found that the YOLOv5 uses CSPdarknet53 as a backbone. YOLO algorithm was used here for detecting the type of vehicle as well the number plate of the vehicle. Here in the system, it will be able to recognize the type of vehicle along with the number plate . Here two different custom models were trained, where one model was used to detect vehicles and classify them based on their types and another model was used to detect the license plate of detected vehicles. Since here the paper focuses on less inference time, YOLO algorithm is considered here.

III. METHODOLOGY

Here it can be observed that, there are not many models which are able to detect vehicles frequently used in India like auto-rickshaws, lorries etc. A single image contains number of vehicles which have to be detected so for less inference time Single Shot Multibox Detector neural network has to be considered which was discussed in detail in [20]. The YOLOv5 which uses SSD approach which was one of the reasons why this algorithm was considered. The dataset for training the haven been taken from various sources and labelled them so that it can be used for training the models used here. The implementation is done in Google Colab environment. The following are the steps used for training of the model.

1. Data Collection
2. Data Preparation
3. ML model Training
4. Model Evaluation
5. Model Updating if required
6. Deployment

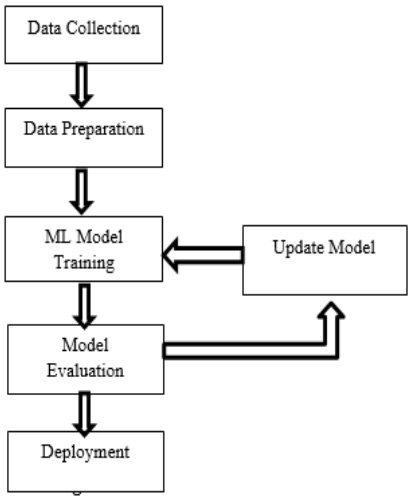

Fig 1: Flow Chart of Custom Model

The above Fig.1 Flow chart describes about the structure used when the custom YOLOv5 model used here is trained.

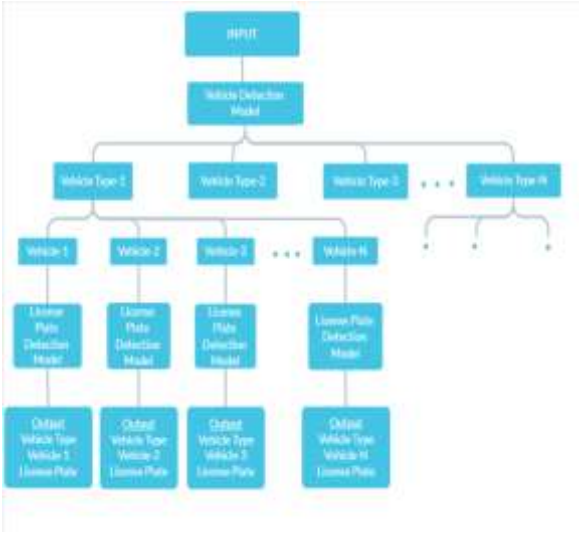

Fig 2: Flow Chart of the Project

From the Fig 2 it can be seen that the input image/frame is passed to the Vehicle Detection Model which detects the vehicles and classifies them into different type of vehicles. The detected part of the image is then sent to the License Plate Detection Model so that it can identify the license plate of that particular vehicle if it is detectable. After the license plate was detected, it was sent to Optical Character Recognition (OCR) so that it can recognize the characters present in the detected license plate. From [6]-[9] different approaches can be seen which were used to identify the characters in the plate. The EasyOCR package available in python was used in this paper to do the recognition of characters. The recognized characters will be saved along with the detected vehicles and the detected plate. From [10]-[15] it can be seen that different algorithms were used for object detection and classification. Here YOLOv5 pre-trained weights are used for training .

Vehicle Detection Model

For this model, a dataset of 2500 images was considered in which there are 10 classes and trained the model with 80 epochs.Using [1] and [2] a custom model was trained. Since there are not many models which were trained on vehicles commonly used in India, a custom model was trained. The vehicle types included in the dataset are as follows:

- Airplane
- Auto-rickshaw
- Bicycle
- Boat
- Bus
- Car
- Lorry
- Motorcycle
- Train
- Truck

A total of 183 images were considered for the validation of this model. The Validation results are as follows:

Fig 3: Validation Results of Vehicle Detection Model

From the above figure, it can be seen that the mean average precision(mAP) was very good at IOU 0.5 but when it is considered at mean Average Precision mAP_0.5:0.95 it was less.

Fig 4: Results of trained Vehicle Detection Model

From the above figure it can be observed that the trained model is predicting labels accurately.

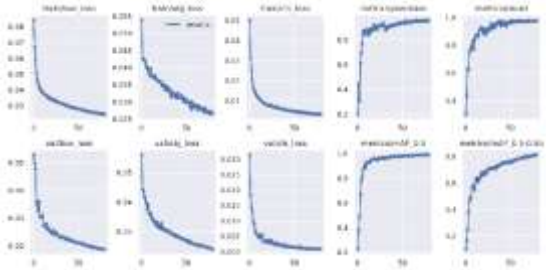

Fig 5: Training Results of Vehicle Detection Model

From the above figure it can be seen that the mean average precision(mAP) is very good at IOU 0.5. It can also be observed that there was some box loss, object loss and classification loss.

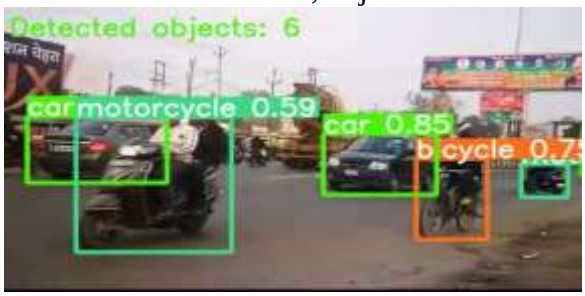

Fig 6: Results when the model was loaded using PyTorch

From the above figure it was clearly visible that the model was able to detect the different type of vehicles and the total number of objects detected.

License Plate Detection Model

For this model a dataset of 1156 images were gathered from various sources and the model was trained with 80 epochs. The Validation results are as follows:

Fig 7: Validation Results of License Plate Detection Model

From the above figure it can be seen that the mAP of the model is not very high.

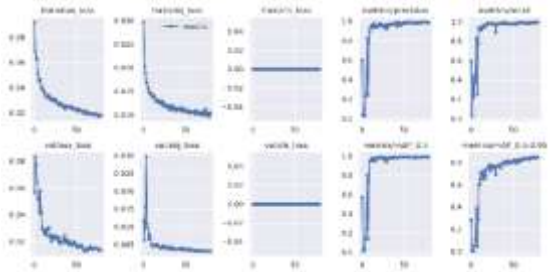

Fig 8: Training Results of License Plate Detection Model

From the above figure it can be seen that the mean average precision(mAP) is very good at IOU 0.5. It can also be observed that there was there is no classification loss since there is only one class.

Fig 9: Recognized characters using easyOCR

From the above figure it can be seen that the characters are extracted from the detected license plate using the easyOCR package available for Python.

IV. RESULTS

Both the models are deployed here by using PyTorch and were used in the same module which increase the time taken to process a single image. The inference time for each image depends on number of vehicles in the image since the detected vehicles are sent as input to license plate model one after another.

If GPU is available, the inference time greatly decreases which will allow the module to run faster.

Fig 10: Input Page

The above Figure shows the input image which is given to the vehicle detection model. The detecting the vehicles the result is as follows

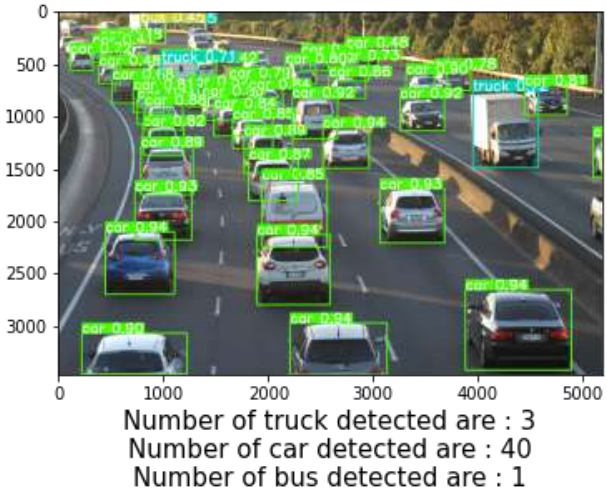

Fig 11: Output Page

From the above figure it can be seen that same vehicle is being detected as bus and truck. It was due to a smaller number of epochs when the training of the model is done. As we can see it is also showing the number of vehicles detected in each type of the vehicles detected. All the detected vehicles are sent to the license plate detection model individually.
The detected vehicles are stored based on vehicle type and the detected plates are also saved next to them along with the time when the vehicle is detected.

Fig 12: Saved Images

From the above figure it can be seen that the 3 trucks and the 1 bus detected by the vehicle model are saved in the file explorer of colab environment.

Fig 13: Sample Image of a detected vehicle

From the above figure it can be seen that it was the car which is located at the right-bottom of the input image. This is one of the images which was sent to the license plate detection model.
The output for the above input is as follows

Fig 14: Detected License plate when Fig 13 is given as input to License Plate Detection Model

From the above figure it can be seen that the license plate was detected from the detected vehicle which will be sent to optical character recognition so that the characters in the license plate can be obtained.

Fig 15: Output when the model couldn't recognize license plate

The above figure shows the starting part of the output when figure 10 is given as input to the vehicle detection model. From the metrics provided in the figure it can be seen that to detect 44 vehicles it took almost 327.5 ms time to pre-process and 315.8ms for inference.

Since all the 44 detected vehicles will be sent to license plate detection model the total time taken will be equal to the time taken by the vehicle type detection model and 44 times the time taken by the license plate detection model per image.

Fig 16: Sample Output-1

The above figure is part of the output where it can be seen that the plate number is detected from the detected vehicle and using OCR it detected the characters which are displayed.

Fig 17: Sample Output-2

Similarly, the to the above figure we can see the detected vehicle and the detected license plate along with characters recognized by easyOCR package available in python.

V. CONCLUSION

Thus, a system is developed which helps to detect the vehicle type and license plate to reduce the effort of handling it manually. The system can be further improved by considering a larger dataset while training the model with more epochs so that its accuracy will be increased. When a video is given as the input, then a vehicle in both the frames is sent to the license plate model both the times since all the detected vehicles are sent to it. To minimize it deep sort techniques can be used so that there is no need to send every vehicle in every frame to the plate detection model.

To further reduce the load on the main system a combination of two systems can be used so that one will handle the task of detecting vehicles and the other one can be used to detect the license plates and recognize the characters in that plate.

VI. REFERENCES

[1] N. Palanivelap, T. Vigneshwaran, M. Srivarappradhan, R. Madhanraj" Automatic Number Plate Detection in Vehicl,es using Faster R-CNN", 2020 International Conference on System , Computation , Automation and Networking, doi:10.1109/icscan49426.2020.926

[2] A.Kashyap, B.Suresh, Ankul Patil, Ankit Jaiswal, "Automatic Number Plate Recognition", 2018 International Conference on Advances in Computing, Communication Control and Networking (ICACCCN), 10.1109/ICACCCN.2018.8748287 .

[3] Du S, Ibrahim M, Shehata M et al (2013) Automatic license plate recognition (ALPR): a state-of-the-art review, IEEE Transaction on Circuits and Systems for video Technology , 23(2),311-325.doi:10.1109/tcsvt.2012.2203741

[4] M.A. Rafique, W. Pedcryz, M. Jeon, "Vehicle license plate detection using region-based convolutional neural networks", Soft Comput., vol. 22, no. 19, pp. 6429- 6440, 2018

[5] K. K. Kumar, V. Sailaja, S. Khadeer, K. Vishwajit, "Automatic Number Plate Recognition System", Indian Journal of Science and Technology, vol. 10, no. 4, Jan. 2017.

[6] Zhai X, Bensaali F, Sotudeh R. OCR-based neural network for ANPR. In 2012 IEEE International Conference on Imaging Systems and Techniques Proceedings 2012 Jul 16 (pp. 393-397). IEEE.

[7] Mori S, Suen CY, Yamamoto K. Historical review of OCR research and development. Proceedings of the IEEE. 1992 Jul;80(7):1029-58.

[8] Mantas J. An overview of character recognition methodologies. Pattern recognition. 1986 Dec 31;19(6):425-30.

[9] Karez Hamad, Mehmet Kaya, "A DetailedAnalysis of Optical Character Recognition Technology ", (2016) International Journal of Applied Mathematics Electronics and Computers , DOI:10.18100/ijamec.270374

[10] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan, "Object detection with discriminatively trained part-based models," IEEE Trans. Pattern Anal. Mach. Intell., vol. 32, pp. 1627–1645, 2010.

[11] Zhao, P. Zheng , Shou-Tao Xu , X.Wu, "Object Detection with Deep Learning Systems,1-21,doi:10.1109/tnnls.2018.2876865

[12] 9012Tang, Y., Zhang, C., GU, R. et al. "Vehicle detection and recognition for intelligent traffic surveillance system." Multimed Tools Appl 76, 5817–5832 (2017).

[13] J. Deng, X.Uan, W.Wang, Z.Li, H.Yao, Z.WANG, "A review of research based o deep learning ", (2020) Journal of Physics: Conference series, 1684,012028,doi:10.1088/1742-6596/1684/1/012028.

[14] Apoorva Raghunandan, Mohana, Pakala Raghav and H. V. Ravish Aradhya,"Object Detection Algorithms for video survilllance applications" International conference on communication and signal processing (ICCSP), India, 2018, pp. 0570-0575

[15] P. Prabhakar, P. Anupama and S. R. Resmi, "Automatic vehicle number plate detection and recognition," 2014 International Conference on Control, Instrumentation, Communication and Computational Technologies (ICCICCT), Kanyakumari, 2014, pp. 185-190.

[16] Arka Prava Jana, Abhiraj Biswas, Mohana, "YOLO Based Detection and Classification of Objects in video records", (2018)3rd IEEE International Conference on Recent trends in electronics , Information and Communication (RTEICT) , doi : 10.1109/rteict4201.2018.
[17] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in CVPR, 2016.
[18] Hassan, N. I., Tahir, N. M., Zaman, F. H. K., & Hashim, H, People Detection System Using YOLOv3 Algorithm ,2020 10th IEEE International Conference on Control System, Computing and Engineering (ICCSCE). doi:10.1109/iccsce50387.2020.9204925
[19] Glenn, J., Alex, S., Jirka, B., NanoCode012, Ayush, C., Tao, X., . . . Ayush, C. (2021). ultralytics/yolov5: v5.0 - YOLOv5-P6 1280 models, AWS, Supervise.ly and YouTube integrations (v5.0). Zenodo. doi:10.5281/zenodo.3908559
[20] Liu Wei, Anguelov D, Erhan D, 2016. SSD: Single shot multibox detector. Proceedings of the European Conference on Computer Vision (ECCV2016). Springer, Amsterdam, AMS, Netherlands, 21-37. https://doi.org/10.1007/978-3-319-46448-0_2

