APPLICATION OF VARIOUS DEEP LEARNING MODELS FOR AUTOMATIC TRAFFIC VIOLATION DETECTION USING EDGE COMPUTING

Ramakanthkumar P, Chethan S, Pavithra H ,Prajwal K , Nehal Chakravarthy M D, Shivaraj B Karegera

Department of Computer Science & Engineering, RV College of Engineering, Bengaluru, India

ABSTRACT

A rapid growth in the population and economic growth has resulted in an increasing number of vehicles on road every year. Traffic congestion is a big problem in every metropolitan city. To reach their destination faster and to avoid traffic, some people are violating traffic rules and regulations. Violation of traffic rules puts everyone in danger. Maintaining traffic rules manually has become difficult over the time due to the rapid increase in the population. This alarming situation has be taken care of at the earliest. To overcome this, we need a real-time violation detection system to help maintain the traffic rules. The approach is to detect traffic violations in real-time using edge computing, which reduces the time to detect. Different machine learning models and algorithms were applied to detect traffic violations like traveling without a helmet, line crossing, parking violation detection, violating the one-way rule etc. The model implemented gave an accuracy of around 85%, due to memory constraints of the edge device in this case NVIDIA Jetson Nano, as the fps is quite low.

KEYWORDS

Traffic violation detection, Jetson Nano, MobileNet , YOLO, edge computing

1. INTRODUCTION

Nowadays with the increasing population, the world has seen a dramatic increase in traffic congestion. The main cause would be the increasing population and the failure to plan good roadways. It is affecting productivity, mobility, travel cost, and travel time. The travel time is affecting the work-life of many workers. People are getting annoyed by the waiting time in traffic for hours and hours. To avoid traffic congestion, they are trying to violate traffic rules. By doing so, they are putting everyone's life in danger and it is very difficult to detect so many people at once. Therefore, the situation demands an automatic detecting system that helps in regulating traffic rules and regulations. Hence, in this paper we have implemented various models and algorithms to detect traffic violations that includes signal jump, travelling without helmet, parking in no parking zone, and wrong way entry etc. To detect signal jump MobileNet algorithm is used which is a light weight deep neural network architecture designed for mobiles and embedded vision applications, for helmet detection YoloV4 is used, it is a good network design choice for an object detection task which is better than YOLOV3 model.

The parking violation detection was carried out by the YOLOV3 model, which is a convolutional neural network, which was used to extract the features of the input image. The wrong way entry was detected with the help of Haar Cascade classifier algorithm.

DOI:10.5121/ijitcs.2023.13201

To have a faster and efficient working model at the edge, the NVIDIA Jetson Nano 2 GB is used which is a powerful edge-computing device. This device is a powerful computer packed in a small for AI, IOT and embedded applications. It has the performance and capability to run workload in a fast and easy way. With the help of powerful tools and efficient detecting algorithms, we have develop a stand-alone working model.

2. LITERATURE REVIEW

The concept of object detection is recent and has come into existence due to the advent of machine learning and deep learning over the years [13]. The classical methods which were used to detect objects in the beginning were sliding window model method [11], frame-difference [7] method, background subtraction method [8], optical flow method [9, 10], Hough transform [6] method, sliding window model method [11] and deformable part model [12][13] method. However, there have emerged new ways to detect objects via deep learning [14]. The method of detecting the objects using the machine-learning model was adopted into our work since it is more efficient than other methods.

The recent works carried out in the field of deep learning to tackle object detection [13] with evaluation metrics and conveyed the light weighted models, which come in handy when running these models on the edge devices. The latest methods to perform object detection are RCNN, SPP-net, Fast R-CNN, R-FCN, which were based on region proposal, and YOLO SSD models, which were based on regression [14] also, compared these models based on evaluation parameters such as accuracy, FPS.

The research aims at Real time automatic helmet detection of bike riders [1] where they have used the YOLO model and developed the whole system. The major challenge in developing the system is to determine the regions of interest and while logically combining the COCO model and the developed model. The model used (YOLO) is heavy weight but the method used is fully appreciable and taken as the reference method for this work. Therefore, YOLO is used to custom train the Helmet detection model.

The research aims at Detection of traffic violations of road users based on convolutional neural networks [2]. The system was developed using R-CNN and R-FCN models, which are faster and accurate, compared to the YOLO object detection model. Detecting both pedestrians and the vehicles is the major challenge and highlight of this work. However, the algorithm designed to detect the violation is not optimal and hence could not detect traffic violation accurately. The work has three main objectives like detecting the signal violation, detecting the parking violation, and detecting the direction violation. The system used a pre-trained Mobile Net model since it is lightweight compared to any other object detection models. The idea of using a lightweight MobileNet model is for line cross violation detection.

The research focused on developing Gaussians based model, MeadianFlow algorithm for detection of one-way violation and speed violation detection. It was planned to develop a publish/subscribe distributed system model in which users can track infringements only in the type that they only want [5]. Hough transforms and bus cameras were used to monitor the road congestion information and detect violation of vehicles in order to achieve early warning and real-time monitoring. However, the camera position is at a very low level and hence long-range violation detection becomes impossible using this idea [6].

The advancements in IoT have led to opportunities in Edge computing. Nvidia Jetson Nano is the cheapest available single board device available in the market, which consumes low power and provides GPU, which were commonly used for high performance deep learning applications [15].

Paper [16] compares the performance of the Jetson Nano and its superior version, which is Jetson TX2 for openCV template matching method and it, was observed that the TX2 is 3 times faster than Jetson Nano is but still it is competitive enough to perform image processing. Hence, Jetson Nano was used in our proposed system.

Many Applications of image processing have been deployed and tested on the Jetson Nano device. A robust crosswalk violation detection application was proposed in paper [17] with an FPS of 33.1 with average F1 score being 94.83%. A Face and emotion recognition system that was implemented in paper [18]. A comprehensive traffic control system was proposed and developed in [19], which used the SSD model instead of YOLO and MQTT protocol for communication. In [19], utilization of pi cameras was done to capture the video feed, which had a success rate of 90% with respect to image processing.

An important research was done to exploit the characteristics of PTZ cameras [20]. These cameras allow motorized cover a wide field of view. A classic application of these cameras is to image mosaicing. However, they can also be used to track moving objects. An approach for performing the registration, adapted to the case of central projection and a background subtraction algorithm for these cameras. The background image is iteratively updated and only on the part "seen" by the camera. They have experimented with different segmentation algorithms using our background modeling technique and this approach makes it possible to track object tracking in real time for PTZ cameras. So, in order to make the system standalone the Nvidia Jetson Nano was used with PTZ camera connected to it for taking the real time input. Since the PTZ camera provides the liberty to adjust the focus at any given time.

3. METHODOLOGY

In order to implement the traffic violation detection system using the NVIDIA Jetson Nano, first need to set up the Jetson nano. This was done using the manual instructions provided by Nvidia. The traffic violation detection system works on a normal PC too. However, to make it a standalone system Jetson Nano is used. A Machine-learning model was used to detect the vehicles in a frame. So, a pre-trained machine-learning model was used since it has the highest accuracy and best adaptability. However, there is no pre-trained machine-learning model available for helmet detection. Therefore, a YOLOv4 model is custom trained with the custom dataset. In order to custom train, the model the dataset needs to be pre-processed and annotated.

The MobileNetV2 architecture was based on an inverted residual structure where the input and output of the residual block are thin bottleneck layers opposite to traditional residual models which use expanded representations in the input an MobileNetV2 uses lightweight depth wise convolutions to filter features in the intermediate expansion layer.

Based on the requirements and analysis three different machine-learning models were used for different use cases. The MobileNet SSD V2 model was used for line cross detection since it is lightweight and works with higher accuracy when objects are nearer to the camera. YoLov4 was used for parking detection and helmet detection since it can detect the objects until a long range of distance. Haar cascade was used for wrong way detection. The four different algorithms for four use cases are as below.

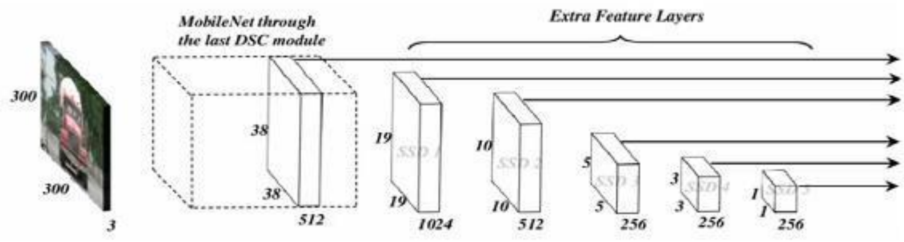

Figure 1. MobileNet Architecture

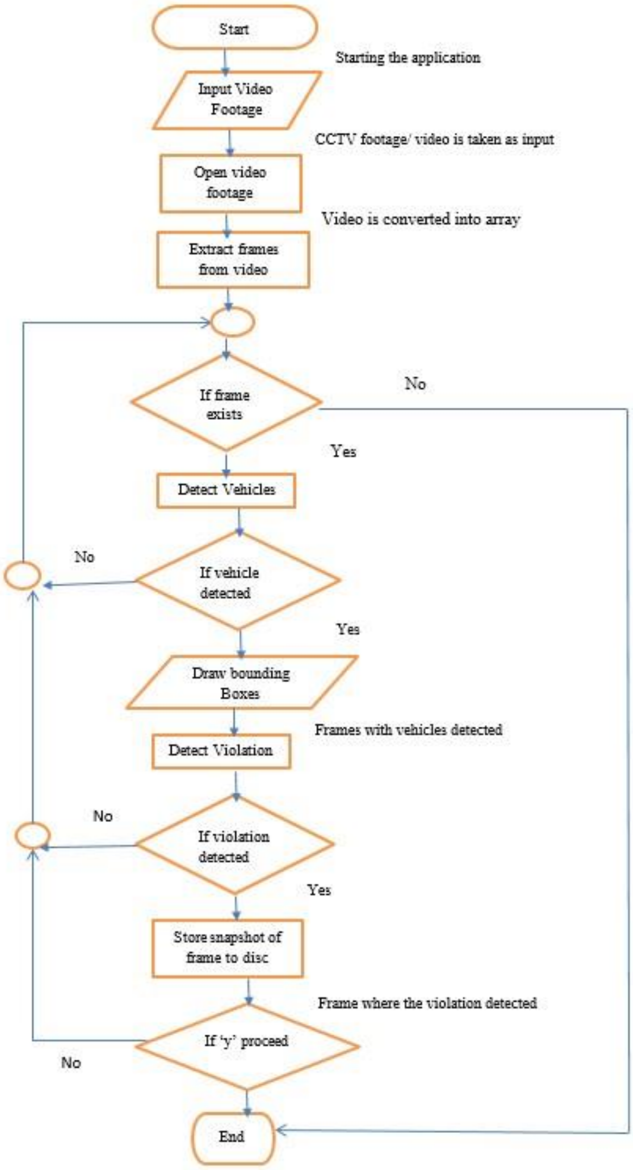

Figure 2. Flowchart for Automatic traffic violation detection

3.1. For Line Cross or Signal Jump Detection

Here the bounding boxes were drawn around only vehicles and the position of the bounding box are noted. Then when the position of the bounding box is inside our interested area then that means the vehicle has crossed the redline. Therefore, the frame with violation was captured and stored in the drive by creating a folder with timestamp as the folder name.

3.2. For Parking Violation Detection

Vehicles were detected using the pre-trained YOLO model and bounding boxes were drawn around the detected vehicles. Then the coordinates of the bounding boxes are noted. If the vehicle is Not Parked, then it should move a minimum distance in a given amount of time. Therefore, accordingly the position of the bounding box also changes. If the position of the bounding box is not changed then it was concluded that it was parked. To differentiate the vehicles, ID was assigned to every vehicle. Then the alert message was printed with the duration the vehicle was parked in that place.

3.3. For Wrong Way Entry Detection

Pass the video frame through the Haar-cascade model to obtain the classes present in that frame, their corresponding bounding boxes and probabilities. Track the vehicle using centroid-tracking method. Now that we have tracked a vehicle, we can find out the direction in which the vehicle is travelling. By the direction in which the vehicle is travelling, we can infer if the vehicle is moving in the wrong direction. Once a vehicle travelling in wrong direction was detected, we can crop the image of that vehicle using the bounding box got from YOLO model and save that image along with timestamp for further analysis.

3.4. For Helmet Detection

The custom-trained model was used to detect the bike riders with and without helmet and then bounding boxes were drawn around the detected objects. If there exists any rider without a helmet then the alert message was printed on the frame.

For detecting the wrong way and parking the centroid, tracking mechanism was used. The mechanism or methodology for centroid tracking is as follows.

Step 01- Accept bounding box coordinates and compute centroids using the formula given below:

$$(x\,,y) = (x + \left(\frac{w}{2}\right)\,, y + \left(\frac{h}{2}\right))$$

(x, y) => Coordinates of top left corner of bounding box
w => width of bounding box   h => height of bounding box

Step 02 - Compute Euclidean distance between new bounding boxes and existing objects.

Euclidean distance between two points was calculated using the formula given below.

$$d(p,q) = \sqrt{\sum_{i=1}^{n} (qi - pi)^2}$$

p, q = Two points in Euclidean n space. qi, pi = Euclidean Vectors, starting from the origin. n = n-space.

Step 03 - Update (x, y)-coordinates of existing vehicle
Step 04 - Register a new vehicle
Step 05 - Deregister old vehicle

It is not enough to detect the violations in a recorded video feed. So, in order to make this work useful the system was made to work on real-time video feed. Once the software was developed for traffic violation detection, the PTZ camera or USB camera or IP camera was used to take the input. Therefore, the developed system is dynamic, realistic and stand-alone.

4. EXPERIMENTAL ANALYSIS

Software performance analysis looks at how a specific program is performing on a daily basis and chronicles what slows down performance and causes errors now and what could pose a problem into the future. Performance issues aren't always built into software in a way that can easily be spotted through the QA process. Instead, it is something that can emerge over time after the project has been deployed.

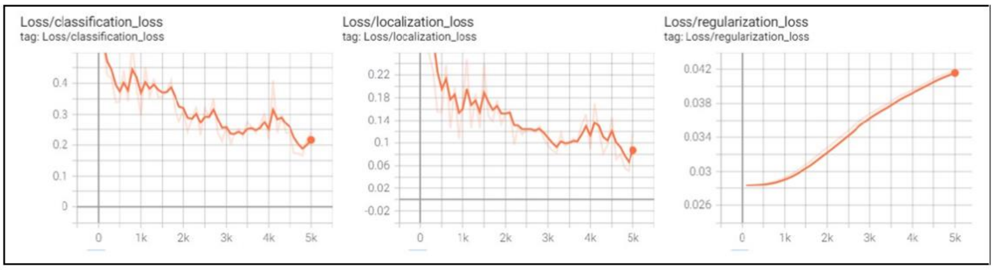

Figure 3. Graph showing model train result

In the above figures, we have plotted the amount of loss on the y-axis and the training epochs on the x-axis it can be observed that the classification loss and localization loss function keep decreasing with training the model with more epochs which is a good sign that our model is performing as expected. The localization loss is very negligible since its value ranges only between (0.026,.043).

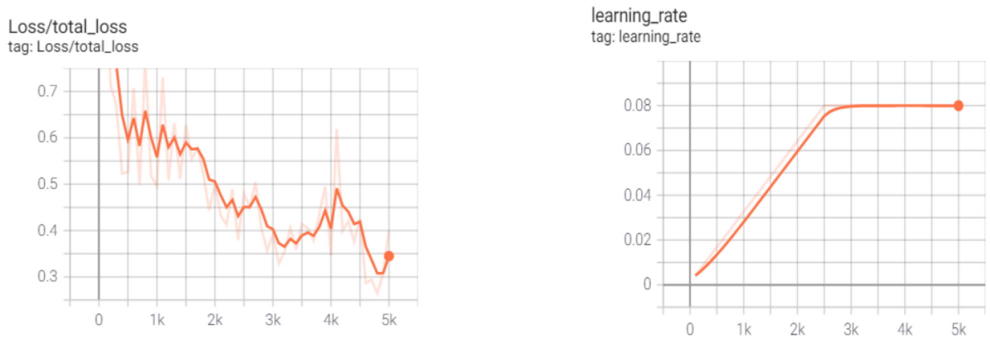

Figure 4. Graph showing the loss

As seen in the above Figure 4, the total loss keeps decreasing with training time and epochs and the final lower bound is about 0.3, which is acceptable. The learning rate saturates after a point of 2,500 epochs, which means that further training of the model will not give any reasonable results, which can also be observed in total loss graph as well.

4.1. Evaluating Mobilenet Model for Signal Jumping

In this section, the results of the work carried out were discussed. The number of true positives for signal jump detection was high with 18 correct detections out of a 24-violations.The model is capable of detecting signal jumping when vehicle density in the frame is low. However, it could not detect all the violations when the vehicle density in the frame is high. The performance of the model was depicted in Figure 5 below.

Table 1: Model Evaluation data for MobileNet

No. of vehicles	No. of vehicles crossing line	Detected Violation
4	2	2
8	4	4
12	8	6
16	14	10
20	16	13
24	20	18

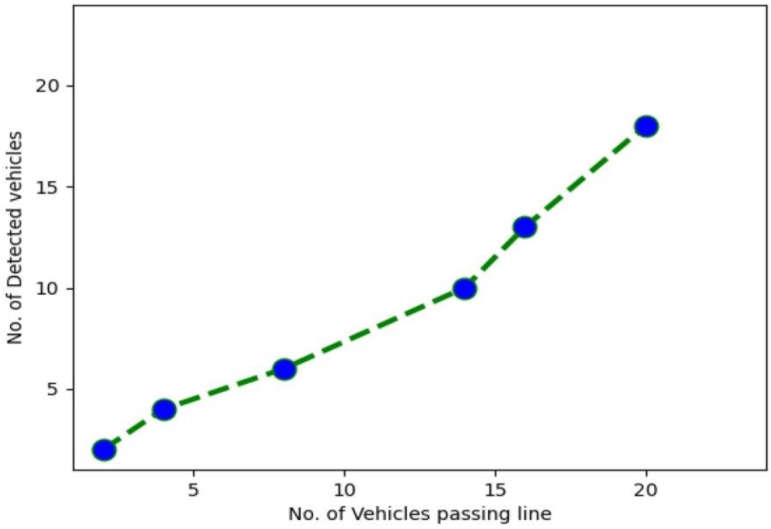

Figure 5. Graph of performance of signal jump violation detection

The YOLO model for parking detection in no-parking zones performs a very good job of detecting parked vehicles. The model detects parked vehicles with an accuracy of 90%, with a small number of false positives. Occasionally, vehicles moving with low speed were falsely detected as parked. The confusion matrix for no parking violation detection is as shown in Figure 6 below.

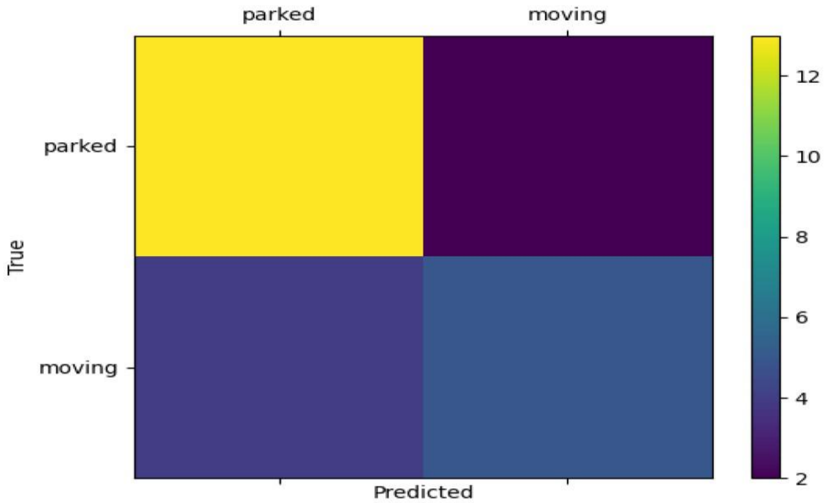

Figure 6. Confusion matrix of no parking violation detection

Similar to the signal jump violation detection model, the wrong way detection model performs well with low density of vehicles in the frame. The performance dips slightly with the increase in vehicle density in the frame. The overall accuracy of the model is 80%. The confusion matrix for wrong way violation detection is as shown in Figure 7 below.

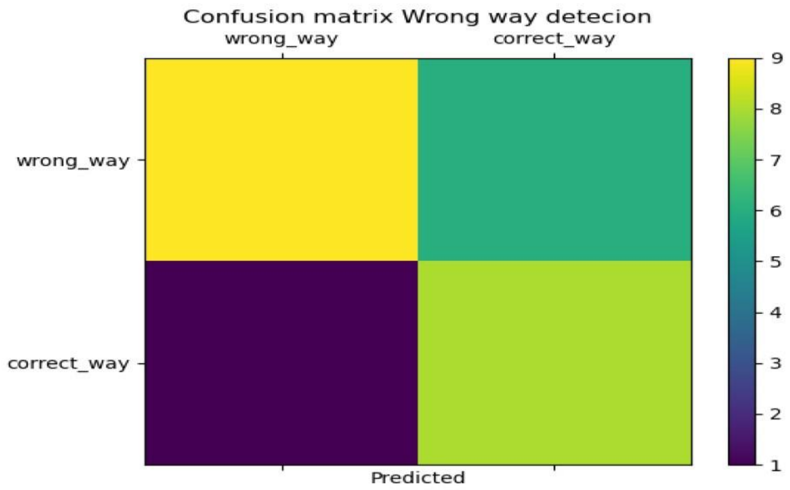

Figure 7. Confusion matrix of wrong way violation detection

The helmet detection model performs well in low speed traffic with good lighting, with an accuracy of 85%. With the increase in speed of motorbikes, the model struggles to accurately detect violations. This can be further improved by training the model on a more robust dataset. Similarly, with dim lighting conditions, the model struggles to differentiate between riders wearing a helmet and those not wearing.

Further, a common observation across all the models is that the rate at which frames are processed is quite slow. The number of frames processed per second becomes a crucial factor in real time applications. The frame rate can be improved by optimising the model to consume less memory to process each frame, to work on embedded devices like the NVIDIA Jetson Nano.

5. CONCLUSION

In this paper, we have discussed the design and implementation of a standalone system for traffic violation detection using NVIDIA Jetson Nano, interfaced with different cameras for capturing real time video feed. Four important traffic violations, namely, signal jump violation, no parking violation, wrong way violation and no helmet violation were addressed in our work.

The analysis of the work carried out depicts that the models perform well generally with a good accuracy of over 85% in detecting the violations. However, the performance is slightly under par in difficult conditions such as high-density traffic, fast moving traffic, dim lighting conditions, etc. Thus, these models can be made more robust by training them on a large robust dataset.

Further, there is scope for improvement in the frame processing speed of the models. The models can be optimized to suit low resource edge commuting devices like the NVIDIA Jetson Nano, by reducing the memory consumption for processing each frame and adopting parallel computing techniques to process frames faster. In future, research work can be carried out to add more violation detections to the standalone system created to assist traffic personnel in maintaining traffic law and reducing the number of violations that put everybody in danger.

ACKNOWLEDGEMENTS

Thanks to ARTPARK Minigrant for providing financial support for the project.

REFERENCES

[1] Ruben J Franklin, Mohana, *"Traffic Signal Violation Detection using Artificial Intelligence and Deep Learning," presented at Fifth International Conference on Communication and Electronics Systems, 2020*

[2] Arun Mathew, Athul Raj A, S Devakanth, Vyshnav B L, Ancy S Anselam, *"Real Time Road Surveillance and Vehicle Detection using Deep Learning," published in International Journal of Engineering Research & Technology, 2020*

[3] Helen Rose Mampilayil, Rahamathullah K, *"Deep learning-based Detection of One-Way Traffic Rule Violation of Three-Wheeler Vehicles", presented at conference on Intelligent computing and control systems, 2019*

[4] Mukremin Ozkul, Ilir Capuni, *"Police-less multiparty traffic violation detection and reporting system with privacy preservation", (2018) IET Intelligent Transport Systems, Vol. 12 No. 5, pp. 351-358.*

[5] Ali Şentaş, Seda Kul , Ahmet Sayar, *"Real-Time Traffic Rules Infringing Determination Over the Video Stream: Wrong Way and Clearway Violation Detection", IEEE, 2018*

[6] Xiaopeng Ji, Zhiqiang Wei, lei huang, Yun Gao, *"Violation vehicle automated snap and road congestion detection ", Proceedings of CCIS2016.*

[7] A. Kurniawan, A. Ramadlan, E. M. Yuniarno,*"Speed Monitoring for Multiple Vehicle Using Closed Circuit Television (CCTV) Camera ", international conference on computer Engineering, network and intelligent multimedia,2018.*

[8] Zhihan Jiang, Longbiao Chen, Member, IEEE, Binbin Zhou, Jinchun Huang, Tianqi Xie,Xiaoliang Fan, Member, IEEE, and Cheng Wang,*"iTV: Inferring Traffic Violation-Prone Locations With Vehicle Trajectories and Road Environment Data", 2017, IEEE SYSTEMS JOURNAL*

[9] Voronin V.V., Gapon N.V., Sizyakin R.A., Ibadov R.R., Ibadov S.R., Semenishchev E.A. *In Proc. XI International scientific-technical forum / DSTU (2014).*

[10] Ren, Shaoqing, et al. *Advances in Neural Information Processing Systems (2015)*

[11] M. Takatoo, T. Kitamura, Y. Okuyama, Y. Kobayashi, K. Kikuchi, H. Nakanishi & T. Shibata, *Traffic flow measuring system using image processing. Proc. SPIE, 1989, Vol. 1197, 172-180.*

[12] R.R. Goldberg & M. Roth, Parallel algorithms for real time tracking. Proc. SPIE, 1996, Vol. 2908, 167-174

[13] Zaidi, S., Ansari, M., Aslam, A., Kanwal, N., Asghar, M. and Lee, B., *"A survey of modern deep learning based object detection models", Digital Signal Processing, [online] 126, p.103514. Available at: https://doi.org/10.1016/j.dsp.2022.103514*

[14] Y. Zhao, H. Shi, X. Chen, X. Li and C. Wang, *"An overview of object detection and tracking," 2015 IEEE International Conference on Information and Automation, 2015, pp. 280-286, doi: 10.1109/ICInfA.2015.7279299*

[15] S. Cass, *"Nvidia makes it easy to embed AI: The Jetson nano packs a lot of machine-learning power into DIY projects - [Hands on]," in IEEE Spectrum, vol. 57, no. 7, pp. 14-16, July 2020, doi: 10.1109/MSPEC.2020.9126102*

[16] A. A. Süzen, B. Duman and B. Şen, *"Benchmark Analysis of Jetson TX2, Jetson Nano and Raspberry PI using Deep-CNN," 2020 International Congress on Human-Computer Interaction, Optimization and Robotic Applications (HORA), 2020, pp. 1-5, doi: 10.1109/HORA49412.2020.9152915*

[17] Zhang, ZD., Tan, ML., Lan, ZC. et al. *"CDNet: a real-time and robust crosswalk detection network on Jetson nano based on YOLOv5." Neural Comput & Applic 34, 10719–10730, 2022. https://doi.org/10.1007/s00521-022-07007-9*

[18] Sati, Vishwani, Sergio Márquez Sánchez, Niloufar Shoeibi, Ashish Arora, and Juan M. Corchado. *"Face Detection and Recognition, Face Emotion Recognition Through NVIDIA Jetson Nano." ,International Symposium on Ambient Intelligence, pp. 177-185. Springer, Cham, 2020*

[19] M. I. Uddin, M. S. Alamgir, M. M. Rahman, M. S. Bhuiyan and M. A. Moral, *"AI Traffic Control System Based on Deepstream and IoT Using NVIDIA Jetson Nano," 2021 2nd International Conference on Robotics, Electrical and Signal Processing Techniques (ICREST), 2021, pp. 115-119, doi: 10.1109/ICREST51555.2021.9331256.*

[20] Lionel Robinault, Stéphane Bres, Serge Miguet "Real Time Foreground Object Detection using PTZ Camera", International Conference on Computer Vision Theory and Applications, 2009

[21] X. He et al., " A Driving Warning Method based on YOLOV3 and Neural Network", IEEE International Conference on ServiceOperations and Logistics, and Informatics (SOLI), Zhengzhou, China,2019.

[22] H. Qu et al., " A Pedestrian Detection Method Based on YOLOv3Model and Image Enhanced by Retinex," 11th International Congresson Image and Signal Processing, Bio Medical Engineering andInformatics (CISP-BMEI), Beijing, China, 2018.

[23] J. Won et al., " An Improved YOLOv3-based Neural Network for De-identification Technology," 34th International Technical Conferenceon Circuits/Systems, Computers and Communications (IT C-CSCC),Korea, 2019.

[24] P. T umas et al., "Automated Image Annotation based onYOLOv3", IEEE 6th Workshop on Advances in Information,Electronic and ElectricalEngineering (AIEEE), Vilnius, 2018.

[25] A . Corovic et al., "The Real-T ime Detection of Traffic ParticipantsUsing YOLO Algorithm," 26th Telecommunications Forum(T ELFOR), Belgrade, 2018.

AUTHORS

Dr Ramakanth Kumar P is a Professor and HOD in the Computer Science and Engineering department at RVCE. His research interests are Digital Image Processing, Pattern Recognition and Natural Language processing.

Chethan S, graduated from R V College of engineering. Working as an GET in Mercedes Benz Research and Development India (MBRDI). Bangalore.

Prof H Pavithra is Assistant Professor in the Computer Science and Engineering department at RVCE.Her research interests are Software Defined Networks, Machine Learning, Deep Learning, Software Engineering.

Prajwal K, undergraduate from R V College of Engineering, SDE at Dish Network technologies, Bangalore

Mr. Nehal Chakravarthy M D is a Computer Science Engineering graduate from R V College f Engineering, Bangalore. He is working as a Software Engineer at HPE.

Shivaraj B karegera, graduated from R V College of engineering. Working as SDE at sixt R&D, Bangalore