ISSN No:-2456-2165

Traffic Signal Violation Detection System

[1]Sanket S Sumptam (Author)
Computer Engineering (Final Year)
Government Polytechnic Gadchiroli
Gadchiroli, Maharashtra

[2]Hemant H Kamdi (Author)
Computer Engineering (Final Year)
Government Polytechnic Gadchiroli
Gadchiroli, Maharashtra

[3]Ritu P Dey (Author)
Computer Engineering (Final Year)
Government Polytechnic Gadchiroli,
Gadchiroli, Maharashtra

[4]Pratik W Junghare (Author)
Computer Engineering (Final Year)
Government Polytechnic Gadchiroli
Gadchiroli, Maharashtra

[5]M.V Lande I/C HOD
Priyanka S. Mallelwar (Lecturer)
Department of Computer Engineering
Government Polytechnic Gadchiroli, Maharashtra India

Abstract:- This study analyzed the impact of road traffic signal violation detection on vehicle speeds in an urban setting. Data was collected and analyzed to establish a vehicle speed probability density model before, within, and after the road traffic violation detection area. Results showed that the average and maximum speeds within the traffic violation detection area were significantly lower than those before and after it. Speeding vehicles were reduced from 70.1% to 15.9% when passing through the traffic signal violation detection had a limited effect in regulating driving behaviors and reducing speeding, but it was effective in the vicinity of the traffic signal violation detection system.

I. INTRODUCTION

I will be exploring a Traffic Signal Violation Detection System using the Python language. The main objective of the project is to create a system that can detect traffic violations using object detection algorithms and image processing techniques. The project will start by exploring the basics of object detection, such as the Viola-Jones algorithm, and how it can be used to detect objects in an image. After that, we will be exploring the image processing techniques that can be used to recognize and identify the objects. Then, we will be looking into how the system can be integrated with other systems, such as traffic cameras and automatic number plate recognition (ANPR) systems. Finally, we will be discussing the results of the project, and how the system can be used to improve the safety of drivers and pedestrians.

➢ *Ease of Use:*

The traffic signal violation detection system project is designed to be user-friendly and easy to use. The system utilizes a variety of technologies such as cameras, sensors, and computer vision to detect and record violations. The system also has an intuitive interface that allows users to easily monitor and manage the system. Additionally, the system is designed with security and privacy in mind, ensuring that only authorized personal can access the system.

- *Maintaining the Integrity of the Specifications:*

✓ Use consistent and precise language when describing the specifications.
✓ Ensure any changes to the code are thoroughly tested and reviewed before being implemented.
✓ Monitor and log all changes to the code to ensure accountability and traceability.
✓ Develop comprehensive unit tests to ensure the code meets the requirements.
✓ Ensure all code is properly documented and commented.
✓ Use coding standards and best practices when developing the system.
✓

The Traffic Signal Violation Detection System (TVDS) project is to improve the accuracy and efficiency of traffic signal violation enforcement. The objective of the TVDS is to create a system that can detect violations of traffic signals in real time, allowing police to respond quickly to violations and increasing the chances of apprehending the violators. Additionally, the project seeks to reduce the number of false positives generated by existing traffic signal enforcement systems.

- *Abbreviations and Acronyms:*

✓ TSVD: Traffic Signal Violation Detection
✓ ITS: Intelligent Transportation System
✓ CV: Computer Vision
✓ ML: Machine Learning
✓ AI: Artificial Intelligence
✓ SSD: Single Shot Detector

- *Units:*

✓ Previous models are not deployed
✓ Only for research purpose studies, not production.
✓ Previous models required huge amount of VRAM for proper functioning.

- ✓ Mostly implementation was in Tensor Flow, which is complex to understand and to work with, any may require multiple training cycle for better results.

- *Equations:*

  V = S * T (equation 1)
  Where,
  V = Traffic violation
  S = Speed of the vehicle
  T = Traffic rules

  C = I * F (equation 2)
  C = Captured image
  I = Input image
  F = Feature extraction algorithm

- *Some Common Mistakes:*
- ✓ Ignoring the need of accuracy: Accuracy is one of the most important elements of a successful traffic signal violation detection system. Without accurate data, the system is unlikely to be effective.
- ✓ Not considering the environment: Traffic signal violation detection systems must be designed to work in a variety of environments. This means that the system should be able to recognize different types of signals and adjust according to the conditions.
- ✓ Poorly designed user interface: A poorly designed user interface can be a major barrier to the effective use of a traffic signal violation detection system. The user interface must be intuitive and easy to use.
- ✓ Not taking into account the cost: Cost is an important consideration when it comes to the development of traffic signal violation detection system. It is important to consider the cost of the system, both in terms of development and in terms of ongoing maintenance and support.
- ✓ Inadequate testing: It is important to thoroughly test a traffic signal violation detection system before it is deployed in the field. This will ensure that the system is reliable and accurate.

II. LITERATURE SURVEY

10 violation, for this case, swerving and blocking the pedestrian lane. The threshold of white is to total pixel ratio in blocking the pedestrian lane is way more than that of the threshold of pedestrian lane possesses greater amount of white pixels, due to the multiple white lines that correspond to the pedestrian lane, than in swerving violation, which only has a single white line to be detected by the system. In fact, it was found out to be any ratio greater than 0.1 for blocking the pedestrian lane, and ratio in the middle of 0.01 to 0.1 for swerving. As the 0.01 threshold mean the noise level floor that was either created within the image due to some process made by the system, or other external effects. experiment results When the machine vision for traffic violation detection system was implemented, the process relating the image processing from the gathered photo from CCTV camera up to the output of the system is described as follows: First the system captures and takes a photo from the CCTV camera, then subtract from it the reference image, which gives the significant properties of the cars in the picture that is needed by the GA. The genetic algorithm is employed thereafter to know and detect whether each car in the scene violates or not. Finally, the system outputs cropped pictures of the car violators similar to the shown in the latter part. The internal process of the genetic algorithm was discussed in the preceding chapter. Shown in the proceeding discussions are the following different inputs and the output of the system was gathered and analyzed, since not only the cropped picture of the car violated was outputted but also the crossover performance and the best fitness plot was taken note of. The best fitness plot in shows the fitness value of the chromosomes every iterations for the detection of swerving, the shape of the curve slowly stabilized at the 2nd iteration at a value of between 0.01 and 0.1 as expected. This proves that the fitness value for swerving do lie between the said range, which in this case lied on 0.087. On the other hand, the crossover performance plot in shows a similar curve in that of blocking the pedestrian lane as explained earlier but converged at the 364th generation. The early convergence for swerving was reasonable because of the iterations took for detecting swerving was less than the blocking of pedestrian lane.

- ➢ *Shortcomings Based On Literature Survey:*
- Previous models are not deployed
- Only for research purpose studies, not production.
- Previous models required huge amount of VRAM for proper functioning.
- Mostly implementation was in Tensor Flow, which is complex to understand and to work with, any may require multiple training cycle for better results.

- ➢ *Problem Definition:*

Currently, traffic police and cameras are only deployed at traffic signals. This limits them to report violations only if they happen within the vicinity leaving room for large stretches of blind spots. Another constraint is the capacity of this personnel to quickly capture as many violation photographs as possible towards proof before the vehicle rushes away. Vehicle automatically identify and report traffic violations on the road around them as they move around thus acting as crowd sourced law enforcement.

- ➢ *Proposed Modules:*
- Data Collection
- Data Preparation / pre-processing
- Model Implementation
- Model Training
- Model evaluating
- Model Development

III. METHODOLOGY

From the preprocessed image moving objects are extracted. A vehicle classification model is used to classify those moving objects into three class - Car, Moto bike and Non-vehicle.0020. The classifier model is built with mobile net v1 neural network architecture. Mobile Net V1 is a variant of Mobile Net model which is specially designed for edge devices. We have explored the Mobile Net V1 architecture in depth. Convolutional Neural Networks (CNN) have become very popular in computer vision.

Mobile Net is a streamlined architecture that uses depth wise separable convolutions to construct lightweight deep convolutional neural networks and provides an efficient model for mobile and embedded vision applications. Mobile Net uses depth wise separable convolutions. It significantly reduces the number of parameters when compared to the network with regular convolutions with the same depth in the nets. This results in lightweight deep neural networks. A depth wise separable convolution is made from two operations.

Mobile Net is a simple but efficient and not very computationally intensive convolutional neural networks for mobile vision applications.

Mobile net is a model which does the same convolution as done by CNN to filter images but in a different way than those done by the previous CNN. It uses the idea of Depth convolution and point convolution which is different from the normal convolution as done by normal CNNs.

Counting depth wise and point wise convo- lotions as separate layers, Mobile Net has 28 layers. The Mobile Net model has only 13 million parameters with the usual 3 million for the body and 10 million for the final layer and 0.58 Million multi-adds. Mobile Net V2 model was developed at Google, pre-trained on the Image Net dataset with 1.4M images and 1000 classes of web images.

➢ *Violation Detection:*

- *After Detecting The Vehicles Three Violation Cases Arises:*

✓ *Signal Violation:* If a vehicle crosses a predefined line on the road while there is red signal, it is detected as a signal violation.

✓ *Parking Violation:* If a vehicle stands still in no parking zone for a predefined time, it is detected as a parking violation.

✓ *Direction Violation:* when a vehicle comes from a wrong direction, it is detected by tracking the vehicle. The direction of the vehicle is determined using its current position and previous few positions.

➢ *Implementation:*

- *Computer Vision:*

Open CV is an open source computer vision and machine learning software library which is used in this project for image processing purpose. Tensor flow is used for implementing the vehicle classifier with darknet-53.

- *Graphical User Interface (GUI):*

The graphical user interface has all the options needed for the software. The software serves administration and other debugging purposes. We don't need to edit code for any management. For example, if we need to open any video footage, we can do it with the Open item (Figure 2).

Fig 1 Initial User Interface View

Primarily, for the start of the project usage, the administrator needs to open video footage using the 'Open' item that can be found under 'File' (Figure 2). The administrator can open any video footage from the storage files (Figure 3).

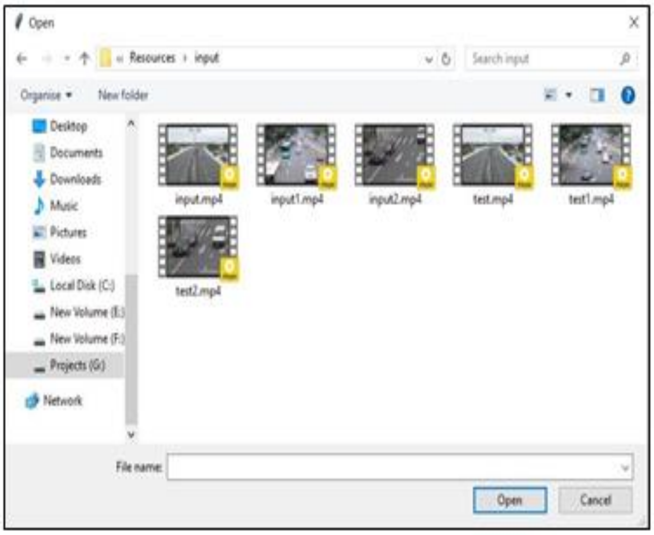
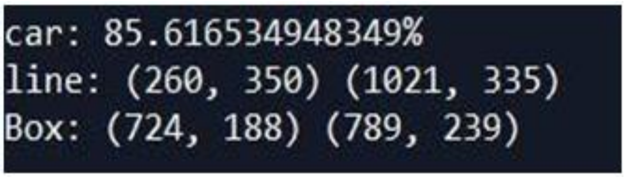


Fig 2 Opening Video Footage from Storage.

After opening a video footage from storage, the system will get a preview of the footage. The preview contains a frame from the given video footage. The preview is used to identify roads and draw a traffic line over the road. The traffic line drawn by administrator will act as a traffic signal line. To enable the line drawing feature, we need to select the 'Region of interest' item from the 'Analyze' option (Figure 3). After that administrator will need to select two points to draw a line that specifies the traffic signal.

Fig 3 Region of Interest (Drawing Signal Line)

Selecting the region of interest will start violation detection system. The coordinates of the line drawn will be shown on console. The violation detection system will start immediately after the line is drawn. At first the weights will be loaded. Then the system will detect objects and check for violations. The output will be shown frame by frame from the GUI (figure 6).



```
car: 85.616534948349%
line: (260, 350) (1021, 335)
Box: (724, 188) (789, 239)
```

Fig 4 Line Coordinate (from Console)

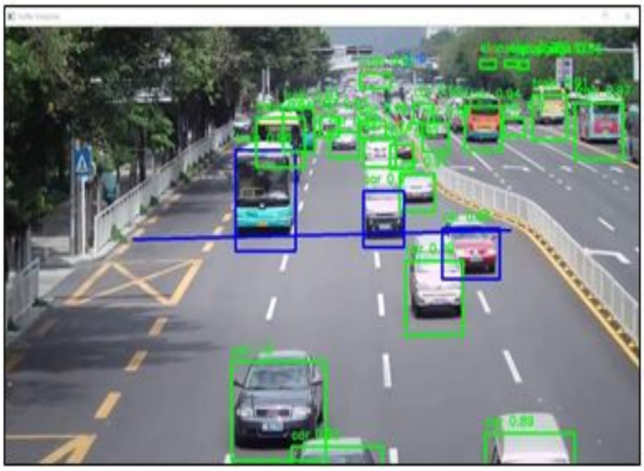


Fig 5: Final Output (on Each Frame)

ACKNOWLEDGMENT

With immense please we, Sanket Sumptam, Ritu Dey, Hemant Kamdi, Pratik junghare presenting a project report "Traffic Signal Violation Detection System" as part of the curriculum of "CSE 3200: System Development Project".

We wish to thank for the people who gave us to unending support.

We express our profound thanks to our guide Ms. P.S. Mallelwar who has guided us and helped us in the development of this system project.

➢ *Libraries used for Graphical User Interface:*

- Tkinter:

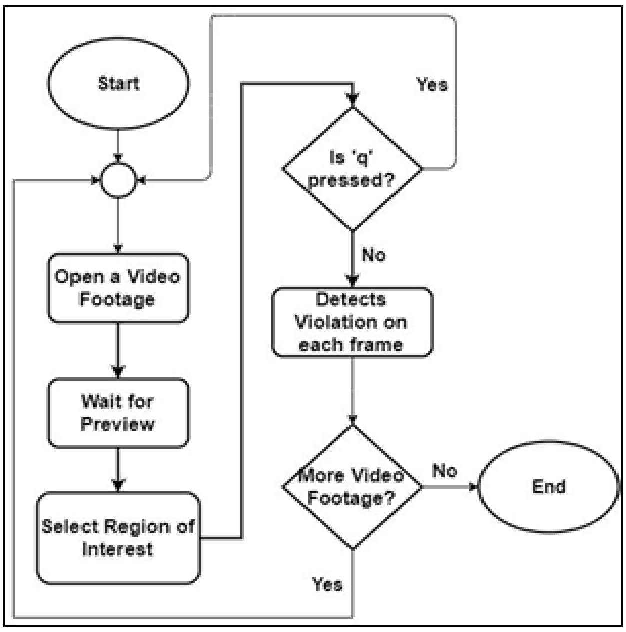


Fig 6 Overall Flow of Software

➢ *Model Training:*

The most optimized way for reconstruction was by minimizing the voxel wise difference such as absolute difference or mean square error. If L1 and L2 loss were not minimized this will result in over-blurred and features at the neighbor or patch level. For better optimization, we used GAN based loss-function. GAN has two networks as discussed earlier, a generator and discriminator. Hence training of GAN has its own challenges. If either of them becomes too strong, the training will fail, and generator will learn nothing from discriminator. The model training would be the difficult of all the modules, we have implemented till now. We cannot train this model all at once, the factors being its huge size added with the large size dataset we used for training. We have tried different platforms available to us for training this model but result is not equivalent to our prior results there are various techniques for training a model with huge model size.

Multi- processing, data-parallel, model-parallel, distributed data-parallel. Multi- processing and data-parallel can help for training model of larger size, but fails to train the model when there is a huge dataset. We have used model-parallel technique for training this model at different machines, resulting the best possible output we can attain. Model-parallel and distributed data-parallel are optimized techniques for training model with huge datasets. For model training using this technique, we have to create a training script which will split the model onto different GPUs, where each GPU consumes a different partitions of input data. The high-level idea of model parallel is to place

ISSN No:-2456-2165

different sub-networks of a model onto different devices, and implement the forward method accordingly to move intermediate outputs across devices. As only part of a model operates on any individual device, a set of devices can collectively serve a larger model. The model was implemented in Py Torch on a workstation with NVidia RTX 2060. The batch size was set to 4. We followed a similar process of patching and data augmentation as in Chen et al. (2018), except, the patch size during training was set as 32 x 32.

IV. CONCLUSION

The designed algorithm was effectively able to detect the type of violation specified on this project which are denying traffic signal. The convergence of detection for the traffic violation mentioned is dissimilar, since it has a different threshold condition. The system provides detection for traffic signal violation. Further, the system is able to process one data at a time. Also, the program runtime is somewhat slow, and can be improved by using a computer with high speed processor specifications or GPU. Future research about the application of the designed algorithm other advanced image processing techniques. Since, this may improve the program runtime of the system by neglecting other unnecessary steps done in a background difference method. A computer vision algorithm may be done instead to provide more intelligence in the system. Our future plan is to implement the number plate detection with OCR support to make this system more robust.

REFERENCES

[1]. Monitoring System," in 2012 IEEE/WIC/ACM, Beijing, China 2012.

[2]. Horry, W.J.; Leach, M.F. Driver-Initiated Distractions: Examining Strategic Adaptation for In-Vehicle Task Initiation. Acid Anal.Prev. 2009, 41, 115–122.

[3]. Dong, Y.; Hu, Z.; Chimera, K.; Murayama, N. Driver Inattention Monitoring System for Intelligent Vehicles: A Review. IEEE Intel. Transp. Syst. 2011, 12, 596–614.

[4]. Horry, W.J.; Leach, M.F Garbed, A. Assessing the Awareness of Performance Decrements in Distracted Drivers. Acid. Anal. Prev. 2008, 41, 675–682.

[5]. Eriksson, M.; Panicle pools, P.N. Driver Fatigue: A Vision-Based Approach to Automatic Diagnosis. Transp. Res. Part C Emerge. Technol. 2001, 9, 399–413. Sensors 2014, 14 22126

[6]. Bahaman, C.; Ying, Z.; Ramesh, M.; Koehler, T. A system for traffic sign detection, tracking, and recognition using color, shape, and motion information. In Proceedings of Intelligent Vehicles Symposium, Las Vegas, NV, USA, 6–8 June 2005; pp. 255–260.

[7]. Bergama, L.M.; Nuevo, J.; Sotelo, M.A.; Berea, R.; Lopez, M.E. Real-Time system for monitoring driver vigilance. IEEE Intell. Transp. Syst. 2006, 7, 63–77.

[8]. McCall, J.C.; Trivedi, M.M. Video-Based Lane Estimation and Tracking for Driver Assistance: Survey, System, and Evaluation. IEEE INTAL. Transp. Syst. 2006, 7, 20–37.

[9]. Manu Bhai, M.; Trivedi, T.G.; McCall, J.C. Looking-In and Looking-Out of a Vehicle: Computer-Vision-Based Enhanced Vehicle Safety. IEEE in tell. Transp. Syst. 2007, 8, 108–120.

[10]. Handan, U.; Kalie, T.; C.; Werner and, M.; Snellen, W.V. An Image Processing System for Driver Assistance. Image Vis. Comput. 2000, 18, 367–376.

[11]. Bergama, L.M.; Nuevo, J.; Sotelo, M.A.; Berea, R.; Lopez, M.E. Real-Time system for monitoring driver vigilance. IEEE Intell. Transp. Syst. 2006, 7, 63–77.

[12]. Escalera, A.; Armingol, J.M.; Mata, M. Traffic sign recognition and analysis for intelligent vehicles. Image Vis. Comput. 2003, 12, 247–258.

[13]. Broggi, A.; Cerri, P.; Ghidoni, S.; Grisleri, P.; Jung, H.G. A New Approach to Urban Pedestrian Detection for Automatic Braking. IEEE Intell. Transp. Syst. 2009, 10, 594–605.

[14]. Jiménez, F.; Naranjo, J.E.; Gómez, O. Autonomous maneuvering systems for collision avoidance on single carriageway roads. Sensors 2012, 12, 16498–16521. [14] Keller, C.G.; Dang, T.; Fritz, H.; Jools, A.; Rabe, C.; Gavrila, D.M. Active pedestrian safety by automatic braking and evasive steering. IEEE Intell. Transp. Syst. 2011, 12, 1292–1304.