International Journal of Recent Technology and Engineering (IJRTE)
ISSN: 2277-3878, Volume-8 Issue-3, September 2019

Intelligent Traffic Management System

Vedant Singh, Vyom Unadkat, Pratik Kanani

Abstract*: Urbanization has presented opportunities of progress which has attracted people from rural areas to the cities thus leading to mass migration. This migration has been going on for decades all around the globe and has reached a point of saturation. The area of the city remains the same but the population density has increased multiple times. Commuting for work is a scene of chaos on the roads. Though there are modes of public transport, roadway is the major mode of commute and the load on roadways is ever increasing due to the rise in population. There is hardly any scope to expand the area of the roadways. The rise in the number of vehicles each year has saturated the capacity the roads were built to carry. This leads to congestion and long hours of traffic on a daily basis which tests the patience of citizens. This provokes the daily commuters to violate the traffic rules which may sometimes amount to grave accidents. Even on Highways, the empty roads entice drivers to experience the thrill of speed overlooking the fact that they are putting themselves at risk. There have been regulations imposed to reduce the chance of an accidents by implementing rules and levying heavy fines on traffic violations. Traffic cameras have been installed all around the city to monitor for traffic violations and get hold of violators. With the technological advancements to store and process large chunks of data efficiently using techniques like Deep Learning and Computer Vision, this paper proposes an automated system to detect Traffic Violations using YOLOv3 to detect and track vehicles and save a snapshot in case a violation is committed.

***Keywords*: Computer Vision, Deep Learning, Object Detection, Traffic Violation, YOLOv3.**

I. INTRODUCTION

In Mumbai, each day, around 700 new vehicles are introduced on the road [1]. Yet, the length of the roads has remained constant at around 2000 kms. Congestion and development projects have resulted into heavy traffic. Traffic Violations are main reason for road accidents. Thousands of accidents occur each year where lives are lost and people get injured. To curb such events, it is imperative to follow the rules and avoid such mishaps from happening. With the technological advancement in monitoring traffic, cameras are now installed across the cities capturing violations all around. With the infrastructure laid out, it would be wise to take advantage of it to further improve the traffic conditions. Machine Learning and AI are some of the tools which will help us better understand and process the ton of data recorded through CCTV cameras.

Manuscript published on 30 September 2019
* Correspondence Author
Vedant Singh*, Information Technology student at Dwarkadas J. Sanghvi College of Engineering.
Vyom Unadkat Information Technology from the University of Mumbai
Pratik Kanani. Pratik Doctoral studies from The MS university of Baroda.

Artificial Intelligence is the main domain with Machine Learning and Deep Learning being subsets of it. Machine Learning can be defined as the ability of computers to learn without being explicitly programmed. Computer are here to assist humans and perform tasks with better accuracy. The computer is given instruction to perform a certain task.
The computer is trained on a certain amount of data so as to make it learn and then it improves by learning from its mistakes.

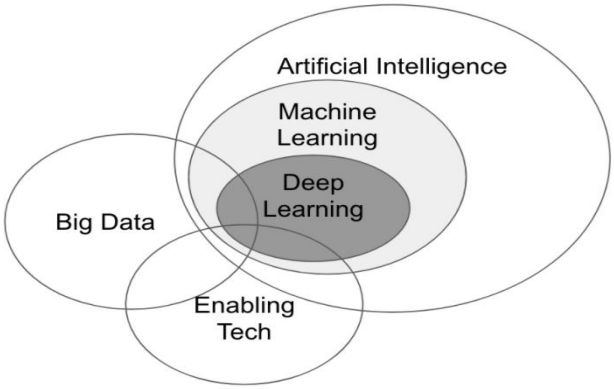

Fig. 1.Artificial Intelligence and subdomains

Monitoring a large number of cameras manually is a cumbersome, mundane and boring process. Using state of the art deep learning algorithms and computer vision, it is now possible to detect objects, vehicles, people and their activities. By this virtue, this paper proposes a system to automate the task of detecting traffic violations.

Earlier, object detection was done by separating the background (static part) from the objects (dynamic/moving part). Background subtraction was a major method used to accomplish this task [2, 3, 4]. Gaussian Mixture Model based background subtraction [5], Hybrid support vector machine (SVM with extended Kalman filter) [6] and particle filter-based trackers [7] were also used. These methods did not fetch good results during heavy traffic conditions due to overlapping off vehicles and predicted wrong bounding boxes. These systems are better employed when the situation is ideal with a well-lit environment and few vehicles. But this is hardly the case. Sometimes, the angle of the camera may not be right, there might be shadows, shaking camera, complex background. The system proposed in this paper employs multiple object tracking in a video stream using Neural Network [8, 9, 10]. The algorithm can process live stream even if the camera is unstable and shaky. Vehicles of all category can be detected like bicycle, motorbike, car van, auto rickshaw, bus, truck, etc. These objects are detected in a live feed with IDs assigned to each vehicle. Traffic violations like jumping the red light and standing ahead the white line (beyond zebra crossing) can be easily detected.

A virtual line which overlaps the white line of the crossing is defined. If the centroid (point) of the bounding box crosses the line while the signal is red, the snapshot of the vehicle is saved along with the time and location. With the footage captured through a high definition camera, the number plate along with the face of the violator will clearly be visible and the fines can be charged accordingly.

II. DATA USED

There is no particular dataset available related to videos of traffic monitoring. There are a few videos online available publicly which have been used for testing. A number of videos were taken from a 12-megapixel mobile camera at 4k resolution. These handheld videos are captured at traffic signals and the algorithm shows promising results. Ideally, live feed from high resolution CCTV cameras installed at a certain elevation and angle should be used, thus showcasing the optimum capacity and accuracy of the algorithm.

III. MODEL

YOLOv3 – You Look Only Once is an Object Detection network which has performed better than other traditional algorithms [11]. Two major tasks have to be performed in Object detection, locating objects in an image and then classifying these objects. There were previous methods which R-CNN which were better than the traditional methods but could only be optimized to a certain level as they had a huge pipeline and each component had to be trained separately. YOLOv3[11] consists of a single neural network which accomplishes the job with better accuracy. It has fast performance and detects multiple objects in a single image which is an advantage.

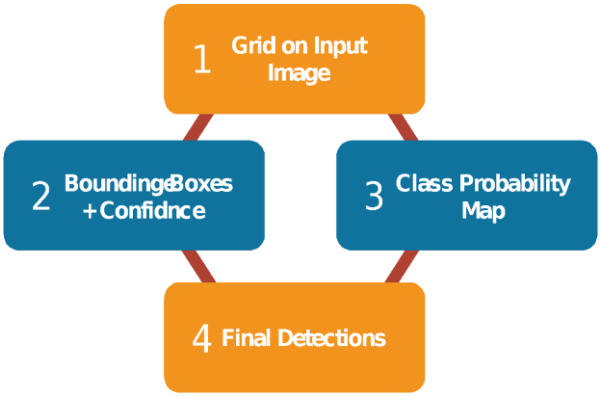

Fig. 2.YOLO detection process

The input image to the FCNN is divided into a grid of size S x S. Each cell in the grid is responsible to detect any object that lies in it. It detects the object with a bounding box and displays the confidence with which the model thinks the object is present. Each bounding box has 5 components:

- The (x,y) co-ordinates of the centre of the box relative to the grid cell.
- The width (w) and height (h) of the predicted object.
- The confidence of the predicted object. It is expressed in terms of Intersection over Union of the predicted bounding box and the ground truth.

Each grid cell also predicts the conditional class probability.

A. Architecture

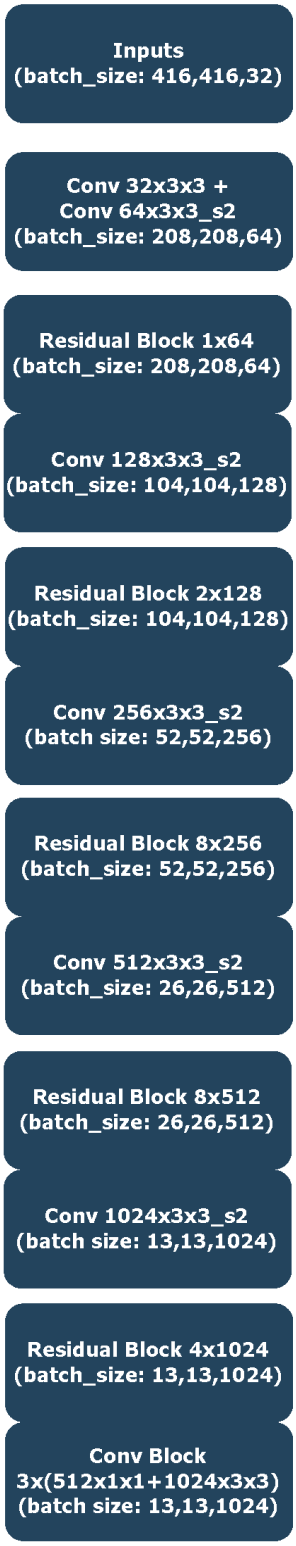

Fig. 3.YOLOv3 Network Architecture

- Conv: Convolutional layer
- _s2: stride of 2
- batch_size: output size of the block

Retrieval Number C6168098319/2019©BEIESP
DOI: 10.35940/ijrte.C6168.098319
Journal Website: www.ijrte.org

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

- Residual Block: multiple Convolutional layers with ResNet architecture

Earlier versions of YOLO used the darknet-19 architecture originally having 19 layers and later 11 more layers were added for object detection [12]. The major drawback was inaccuracy in detecting small objects. To overcome this issue, feature maps were concatenated, still there was no significant improvement.

The YOLOv3 architecture has 106 convolutional layers and is a feature-learning based network. This is a variant of Darknet architecture which has 53 layers trained on the Imagenet dataset [13]. 53 more layers are added for state-of-the-art image detection. It can process images irrespective of the size. No pooling layers are used to avoid missing out on minute features and details. To down sample the feature map, convolutional layers with stride of 2 are used. The ResNet architecture used in the Residual Blocks are key for accuracy and speed.

B. Detection at 3 Scales

The CNN layer takes an image as an input and returns a tensor as an output. The detection of an object is done on 3 scales i.e. on 3 different levels to detect smaller objects which was an issue in older YOLO versions. Output size of the predicted map have the same dimensions as the feature map. The depth of a feature map is defined as:

$$Depth = B \times (5 + C)$$

In equation 1, B represents the number of bounding boxes each grid cell can predict. (5 + C) are the attributes of the bounding box. These attributes are the centre co-ordinates, the height, width, objectness score and C which is the class confidence. Object score is the probability of the object being inside the bounding box and class confidence is the probability of the object belonging to a particular class.

The network is given images as inputs and scaling is done on 3 levels to detect objects of all sizes. Detections are made on the feature maps obtained after scaling images as per strides of 32, 16 and 8 respectively. For, e.g. if an image of size 416x416 is given as input, detections would be made on feature maps of 13x13, 26x26 and 52x52.

In YOLOv3, each cell can predict 3 bounding boxes. Hence, the value of B is 3. If YOLOv3 is trained on COCO, the number of classes is 80. So, C is 80. For 13x13 scale, the image is divided into 13x13 grid cells and each cell has a voxel value of 1x1x255. 255 comes from equation 1 (3 x (5 + 80)).

The 13x13 scale is used to detect large objects, 26x26 for medium and 52x52 for small objects.

C. No more Softmax layer

The features detected by the convolutional layer are then passed as an input to a classifier/regressor which detects the class of the object. The predictions are made by a convolutional layer with 1 x 1 convolutions with logistic regressor. Earlier versions used Softmax Layer. It applied Softmax function of all the class scores and selected the class with the maximum score. This implied that an object can belong to only one class and is mutually exclusive. If there are classes like Car and Vehicle in a dataset, then this assumption fails. Hence, using a logistic regressor (sigmoid) makes more sense [14]. Each class score is predicted using a logistic regressor and a threshold is set. The bounding box is assigned value with classes having a higher value than the threshold value.

D. Bounding Box Prediction [15]

$$b_x = \sigma(t_x) + c_x \quad (2)$$

$$b_y = \sigma(t_y) + c_y \quad (3)$$

$$b_w = p_w e^{t_w} \quad (4)$$

$$b_h = p_h e^{t_h} \quad (5)$$

b_x, b_y, b_w, b_h are the (x,y) co-ordinates and the height, width of the bounding box. t_x, t_y, t_w, t_h are predicted. c_x, c_y are the top-left co-ordinates of the grid cell. The anchor dimensions for the box are denoted by p_w and p_h.

The centre co-ordinates are passed through the sigmoid function as they are offset values relative to the top-left of the grid cell. For e.g. if the top-left of the cell is (8,8) and the centre co-ordinate is (0.3, 0.8), then the centre lies at (8.3, 8.8). As the centre co-ordinates are passed through sigmoid function, they always lie between 0 and 1.

The dimension, i.e. height and width of the bounding box is obtained by log-space transform of the output and then multiplication with the anchor. This output is then normalized to get a value between 0 and 1. If a feature map of size 13x13 is used and the values for b_x and b_y are 0.4 and 0.6, then the height and width of the bounding box are (13x04, 13x0.6).

The object score is the probability if an object lies inside the bounding box or no. The value is 1 if object centre lies in the grid cell and 0 if it lies at grid corner. The sigmoid function is applied on this score as well.

The class confidence denotes the probability that the object belongs to a particular class. Initially, softmax function was used. But as it made the assumption that an object can belong to a class and is mutually exclusive, v3 now makes use of sigmoid function.

E. Loss Function

This is the equation of loss function in versions used before YOLOv3. There were certain changes made in v3.

$$\begin{aligned}
&\lambda_{\text{coord}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{obj}} \left[(x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2 \right] \\
&+ \lambda_{\text{coord}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{obj}} \left[\left(\sqrt{w_i} - \sqrt{\hat{w}_i} \right)^2 + \left(\sqrt{h_i} - \sqrt{\hat{h}_i} \right)^2 \right] \\
&+ \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{obj}} \left(C_i - \hat{C}_i \right)^2 \\
&+ \lambda_{\text{noobj}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{noobj}} \left(C_i - \hat{C}_i \right)^2 \\
&+ \sum_{i=0}^{S^2} \mathbb{1}_{i}^{\text{obj}} \sum_{c \in \text{classes}} (p_i(c) - \hat{p}_i(c))^2
\end{aligned}$$

Fig. 4.Loss Function [16]

The first term deals with the error in the offset of the bounding box location i.e. (x,y) co-ordinates. The second term calculates the error of the width and height of the bounding box.

Third and fourth term deal with the object confidence error and the fifth term estimates the class probability error [17]. All these errors are the Sum of Squared Error (SSE). In YOLOv3, Cross-Entropy error is used instead of SSE. This implies Object Confidence and Class Predictions are made using Logistic Regression [18].

IV. RESULT AND DISCUSSION

The YOLOv3 model on COCO dataset is used as it is trained to detect 80 classes of objects. We are interested in motorcycles, bikes, cars, buses, trucks, etc. The model has also been trained to detect autorickshaw and the dataset was obtained from [19]. This training was done for 200 epochs which took around 48 hours on a machine with i5 processor and 8 GB of RAM attaining training loss of 0.0836. We detect these classes: 'bicycle', 'car', 'motorcycle', 'bus', 'truck' and 'auto' as these are of our interest.

We create a virtual line which falls over the white line ahead of the zebra crossing where the vehicles are supposed to stop. If the signal is red and, and if the vehicle is ahead of the white line, then a violation has to be registered. Also, if the vehicle jumps a red light, then an image of the car has to be captured.

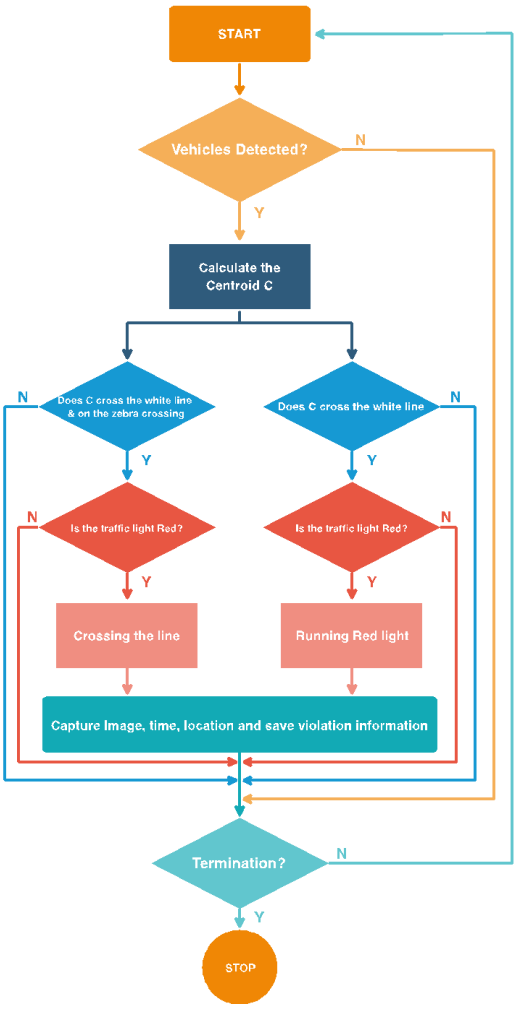

Fig. 5. Flowchart for Violation Detection

Some of the test videos were processed and the results obtained are given below:

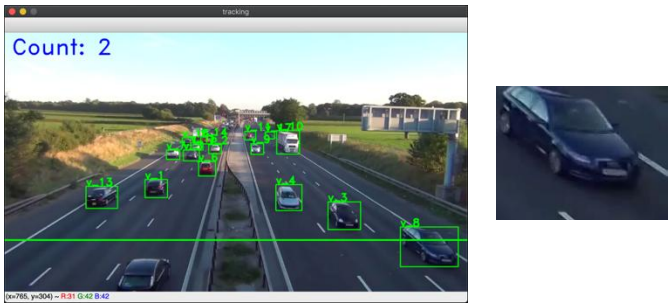

Fig. 6.Violation Detection through CCTV and saved Image of car crossing line

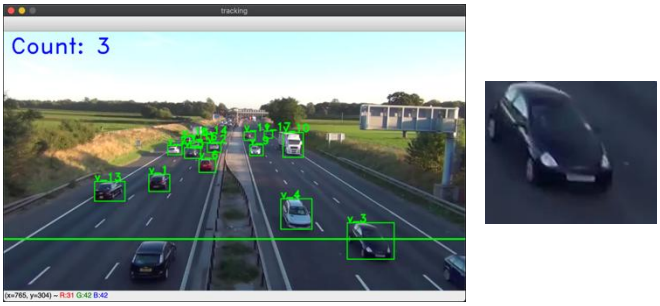

Fig. 7.Violation Detection through CCTV and saved image of car crossing line

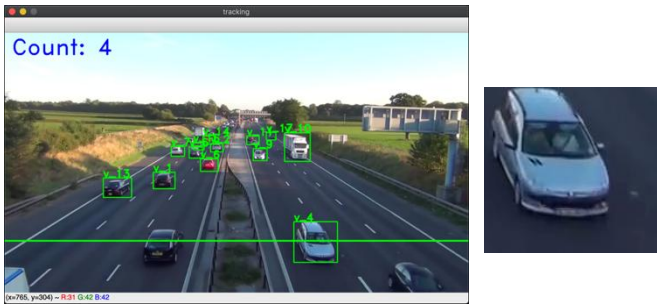

Fig. 8.Violation Detection through CCTV and saved image of car crossing line

Fig. 9.Violation Detection of motorbike ahead of white

Fig. 10. Violation Detection (Autorickshaw)

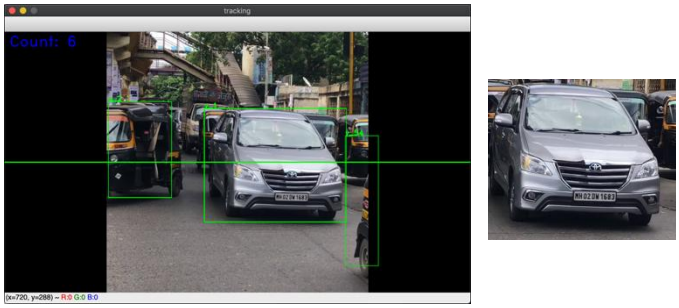

Fig. 11. Violation Detection (Car)

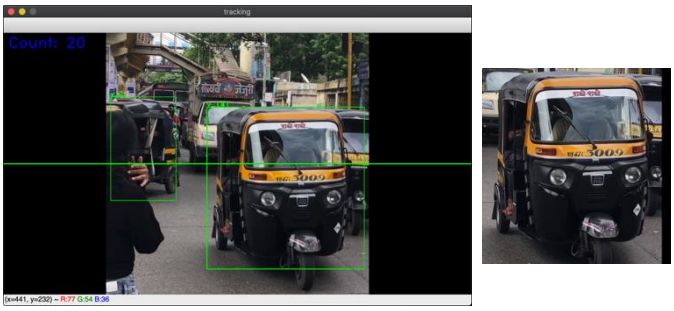

Fig. 12. Violation Detection (Autorickshaw)

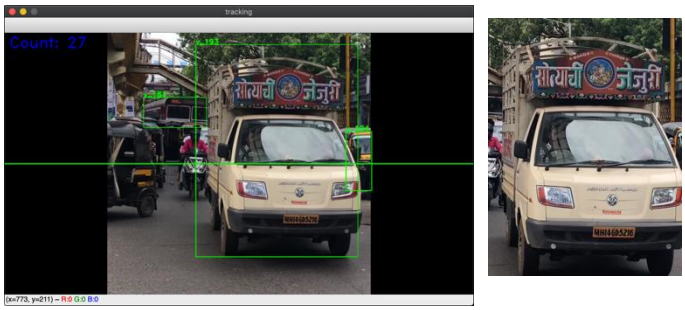

Fig. 13. Violation Detection (Truck)

These results prove the accuracy and efficiency with which the model can detect vehicles and track them. Each vehicle detected is given a unique id and is tracked so as to avoid redundancy in detecting same vehicle twice. As it is clear, a count is maintained of all the vehicles that cross the line. As seen in Fig. 6,7 and 8, the video is captured from a CCTV camera and the model has proved its accuracy in detecting vehicles even from that distance. As soon as a vehicle crosses the line, the image is saved which is the Region of Interest (ROI). If the video is captured from a high-resolution CCTV camera and the same model is run on the feed, then the number plate of the vehicle will be clearly visible in the saved image. In Fig. 9, the motorcycle has crossed the white line and is halted on the zebra crossing. As the centroid of the bounding box has crossed the white line and on the zebra crossing, this is a case of crossing the line. With a good camera, the ROI saved will have a better resolution than the camera used and the number plate would be easily visible. Fig. 10 and 12 demonstrate the case of detecting the autorickshaw after training the model on the dataset. This is a clear showcase of the power of the model as the number plates of the vehicles are easily captured in the ROI. Car and Truck detection is clearly visible in Fig. 11 and 13 respectively. Even the number plates are easily readable. A closer look at Fig. 13 shows bus detection even from partial visibility and obstructions.

V. CONCLUSION

This paper presents the detection of traffic violation in real time using YOLOv3. This neural network used pretrained weights on COCO dataset and then was trained on a separate dataset to detect autorickshaws. Vehicle classes are used to detect objects as other classes are of no use in this application. The uniqueness of this proposed solution is real time detection of the entire spectrum of vehicles that operate on roads. This system also gives the statistics of the number of violators captured at a particular signal by keeping a count of the number of violators. This proposed method has fetched respectable results even in cases where the density of vehicles is large and has detected vehicles even when there were obstructions in vision. The snapshot of the violator saved is stored in a central repository maintaining proof of violation. The only major concern is running the model as running on a CPU on a real-time video can take a long time. Currently, violations of jumping the red light and crossing the white line are detected but later on, violations of speeding and detection of two-wheeler without helmet can also be implemented.

REFERENCES

1. Somit Sen, “Every day, 700 new vehicles hit Mumbai roads”. [Online]. Available: https://timesofindia.indiatimes.com/city/mumbai/every-day-700-new-vehicles-hit-mumbai-roads/articleshow/61908535.cms
2. Pornpanomchai, C., Liamsanguan, T., & Vannakosit, V. (2008). Vehicle detection and counting from a video frame. *2008 International Conference on Wavelet Analysis and Pattern Recognition, 1*, 356-361.
3. Danelljan, Martin & Häger, G. & Khan, Fahad. (2014). Accurate scale estimation for robust visual tracking. British Machine Vision Conference. 1-11.
4. Yang, Y., & Bilodeau, G. (2016). Multiple Object Tracking with Kernelized Correlation Filters in Urban Mixed Traffic. *2017*.
5. P. K. Bhaskar and S. Yong, "Image processing based vehicle detection and tracking method," *2014 International Conference on Computer and Information Sciences (ICCOINS)*, Kuala Lumpur, 2014, pp. 1-5.
6. Vishnu, Maha & Rajalakshmi, M. & Nedunchezhian, R.. (2018). Intelligent traffic video surveillance and accident detection system with dynamic traffic signal control. Cluster Computing. 21. 10.1007/s10586-017-0974-5.
7. C. Bouvié, J. Scharcanski, P. Barcellos and F. L. Escouto, "Tracking and counting vehicles in traffic video sequences using particle filtering," *2013 IEEE International Instrumentation and Measurement Technology Conference (I2MTC)*, Minneapolis, MN, 2013, pp. 812-815.
8. V. Unadkat, P. Sayani, P. Kanani and P. Doshi, "Deep Learning for Financial Prediction," *2018 International Conference on Circuits and Systems in Digital Enterprise Technology (ICCSDET)*, Kottayam, India, 2018, pp. 1-6.
9. V. Unadkat, P. Sayani, H. Kapadia, P. Shah and H. Dalvi, "Automated System for Detecting Distracted Driver," *2018 4th International Conference on Computing Communication and Automation (ICCCA)*, Greater Noida, India, 2018, pp. 1-4.
10. Unadkat V., Gajiwala S., Doshi P., D’silva M. (2020) Intelligent System for Weather Prediction. In: Abraham A., Cherukuri A., Melin P., Gandhi N. (eds) Intelligent Systems Design and Applications. ISDA 2018 2018. Advances in Intelligent Systems and Computing, vol 940. Springer, Cham.
11. Redmon, Joseph & Farhadi, Ali. (2018). YOLOv3: An Incremental Improvement. [Online]. Available: arXiv:1804.02767v1 [cs.CV].
12. Sik-Ho Tsang, ‘Review: YOLOv2 & YOLO9000 – You Only Look Once (Object Detection)’. [Online]. Available: https://towardsdatascience.com/review-yolov2-yolo9000-you-only-look-once-object-detection-7883d2b02a65.

13. Sik-Ho Tsang, 'Review: YOLOv3 – You Only Look Once (Object Detection)'. [Online]. Available: https://towardsdatascience.com/review-yolov3-you-only-look-once-object-detection-eab75d7a1ba6.
14. CyberAILab, 'A Closer Look at YOLOv3'. [Online]. Available: https://www.cyberailab.com/home/a-closer-look-at-yolov3.
15. Ayoosh Kathuria, 'How to implement a YOLO (v3) object detector from scratch in PyTorch'. [Online]. Available: https://blog.paperspace.com/how-to-implement-a-yolo-object-detector-in-pytorch/.
16. Redmon, Joseph & Divvala, Santosh & Girshick, Ross & Farhadi, Ali. (2016). You Only Look Once: Unified, Real-Time Object Detection. [Online]. Available: arXiv:1506.02640v5 [cs.CV].
17. Sik-Ho Tsang, 'Review: YOLOv1 – You Only Look Once (Object Detection)'. [Online]. Available: https://towardsdatascience.com/yolov1-you-only-look-once-object-detection-e1f3ffec8a89.
18. Ayoosh Kathuria, 'What's new in YOLO v3?'. [Online]. Available: https://towardsdatascience.com/yolo-v3-object-detection-53fb7d3bfe6b
19. IIIT, 'AUTORIKSHAW DETECTION CHALLENGE'. [Online]. Available: http://cvit.iiit.ac.in/autorickshaw_detection/

AUTHORS PROFILE

Vedant Singh is currently a final year Information Technology student at Dwarkadas J. Sanghvi College of Engineering. He has interests in the fields of Machine Learning, Deep Learning, Artificial Intelligence, Android & iOS Development, Full-stack Development. He has research papers in IEEE, SPRINGER and other reputed international journals.

Vyom Unadkat has completed his Bachelor of Engineering in Information Technology from the University of Mumbai and will be going to the US for Masters. Just at the age of 12, Vyom fathomed the deep waters of computer programming and Animation and maintained a balance between both of them. Having gained proficiency in programming languages, Machine Learning, Deep learning, Computer Vision and Data Science, he has also developed numerous commercial projects as well. After having gained professional experience through 3 internships, he has also authored 5 research papers published and indexed in SCOPUS, IEEE, SPRINGER and other scientific journals.

Pratik Kanani. Pratik received his Bachelors and Master's Degree from University of Mumbai. He has more than 25 research contributions in International and National Journals as well as Conferences. He is an active researcher in the field of Network Security, IoT, Machine Learning and Fog Computing. Currently he is pursuing his Doctoral studies from The MS university of Baroda.