TRAFFIC RULE VIOLATION DETECTION SYSTEM

Presented by :

A. Manasa,
21251DB103.

Under the guidance of :

Dr. Renuka Devi S M,
Professor,
ECE Dept.

Contents

ABSTRACT

- Violation of traffic rules has become a critical issue for the majority of the developing countries and managing the traffic violations has become a tedious task.
- The proposed traffic rule violation detection system performs tasks
  - a) wrong way violation detection
  - b) over speed violation detection
- The YOLOv3 deep learning method is used for locating vehicles, Kalman filter with DeepSORT is used for tracking vehicles.
- After tracking each vehicle is assigned with an unique id, then speed violation detection algorithm and wrong way direction violation detection algorithm are used to identify the vehicles which violated the traffic rules.

OBJECTIVES

- Object detection using YOLOv3 deep learning method.
- Object tracking using Kalman filter and DeepSORT
- Design of Over Speed Violation Detection (OSVD) algorithm
- Wrong - way Traffic Violation Detection (WTVD) algorithm
- Comparing the results of proposed algorithms with existing algorithms, to showcase the better performance of proposed model.

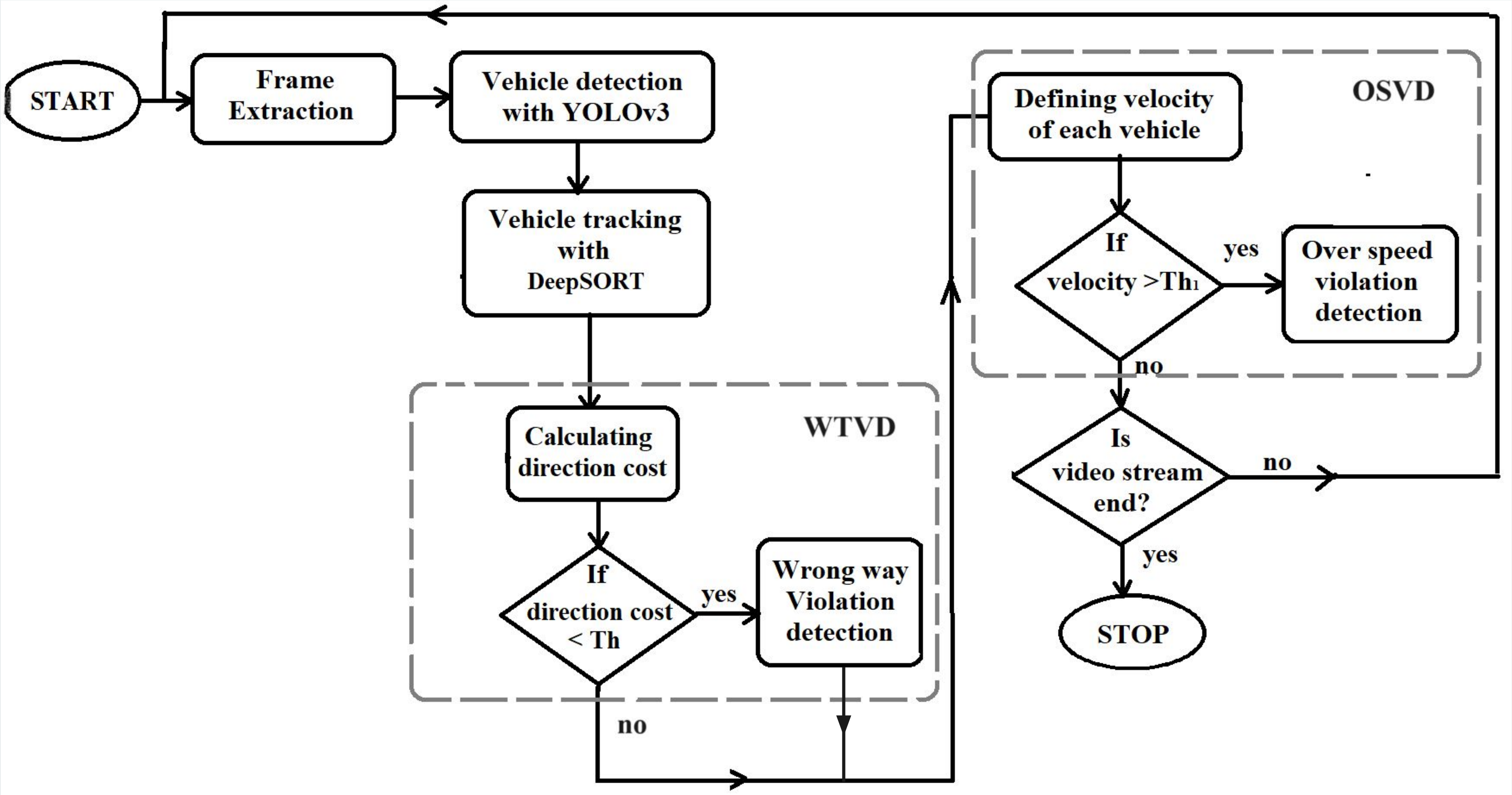


Fig. 1. Basic flowchart of the traffic violation detection system.

PROJECT METHODOLOGY

- The proposed model uses mainly 4 steps :

1. Vehicle detection with YOLOv3
2. Vehicle tracking with Kalman filter and DeepSORT
3. Wrong-way violation detection
4. Over-speed violation detection

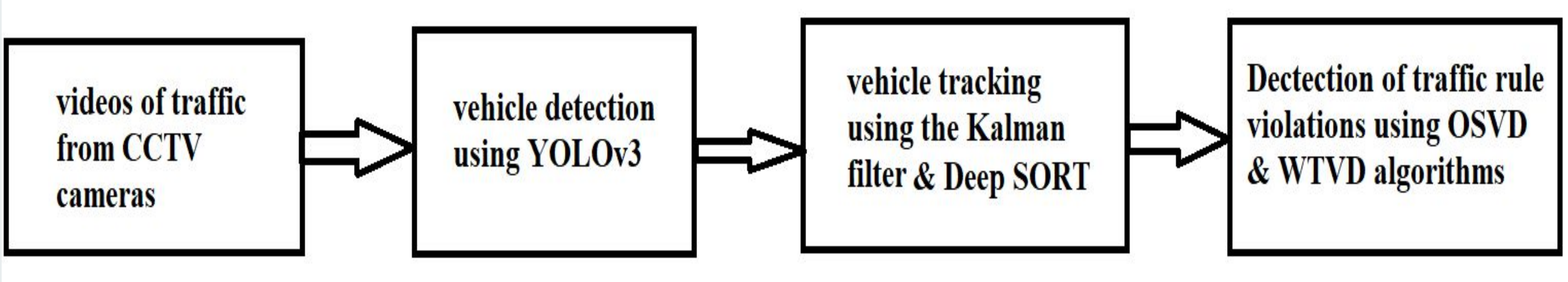


Fig. 2. Block diagram of traffic rule violation detection system.

1. Vehicle detection with YOLOv3

- The You Only Look Once version 3 (YOLOv3) algorithm is a Convolutional Neural Network used for object detection in real-time.
- Traditional methods such as background subtraction and optical flow can also be used to detect objects. However they are not efficient.
- **Drawbacks of traditional methods :**
  1. They provide false positive results when numerous cars are driving closely
  2. They are particularly sensitive in dim lighting, producing unreliable models
- To overcome these drawbacks YOLOv3 algorithm is selected for detection.
- YOLOv3 algorithm with input image size 416 x 416 is used which has processing speed of 29 ms and mean Average Precision (mAP) of 55.3 percent on COCO dataset [4].

YOLOv3 Network Architecture

- The architecture of the YOLOv3 network includes components like upsampling , skip connections and residual blocks.
- For detection YOLOv3 uses Darknet-53 along with additional 53 layers stacked onto it, with a total of 106 layers of fully convolutional architecture.
- It does not consist of any pooling layers. Instead of pooling, additional CNN layers with stride 2 are used to downsample the input image.
- This prevent loss of low level features which are useful in detecting small objects.
- YOLOv3 makes predictions at 3 different layers 82, 94 and 106 layers, which are used for detecting large, medium and small sized objects respectively.

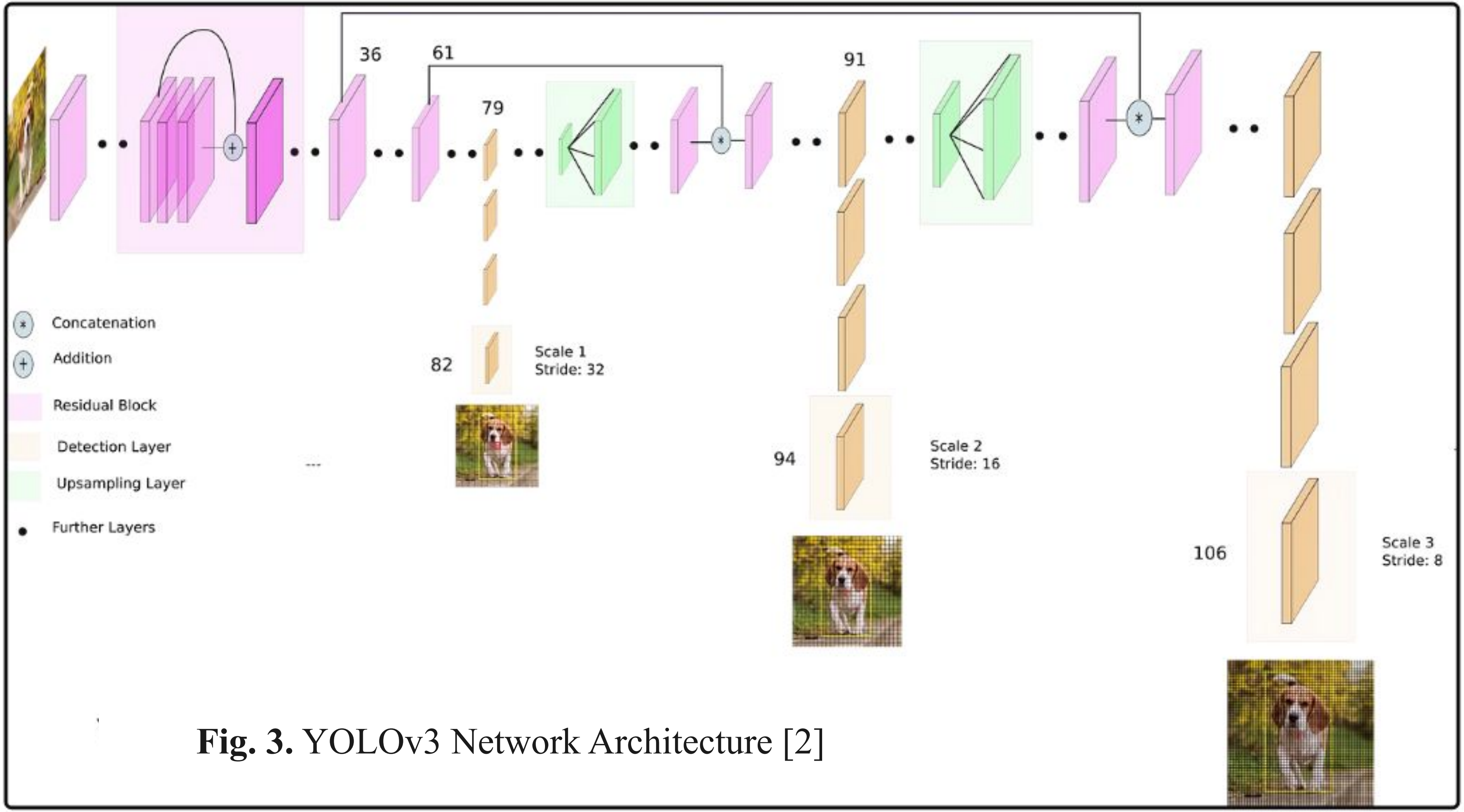


Fig. 3. YOLOv3 Network Architecture [2]

- The convolutional layers of YOLOv3 architecture learn various important features of input images and are then given to a regression for detection.
- These learned features have the bounding boxes coordinates, class labels, bounding boxes sizes ,dimensions and many more.
- The input image is first separated into grid cells by the YOLOv3 algorithm.
- The grid cell is responsible for detecting an object only when the center of that particular object falls into that grid cell .
- In YOLOv3, each cell of the prediction map predicts only a certain number of bounding boxes. YOLOv3 has three anchors and it predicts three bounding boxes.
- Each bounding box prediction consists of following five elements: (a, b, w, h, c). Where (a, b) is centroid coordinates, w is width, h is height and c is confidence score.

- A same object may be identified many times when more than one bounding box recognises it as a positive detection.
- To overcome this non-maximum suppression is used which only allows detections that haven't been seen before.

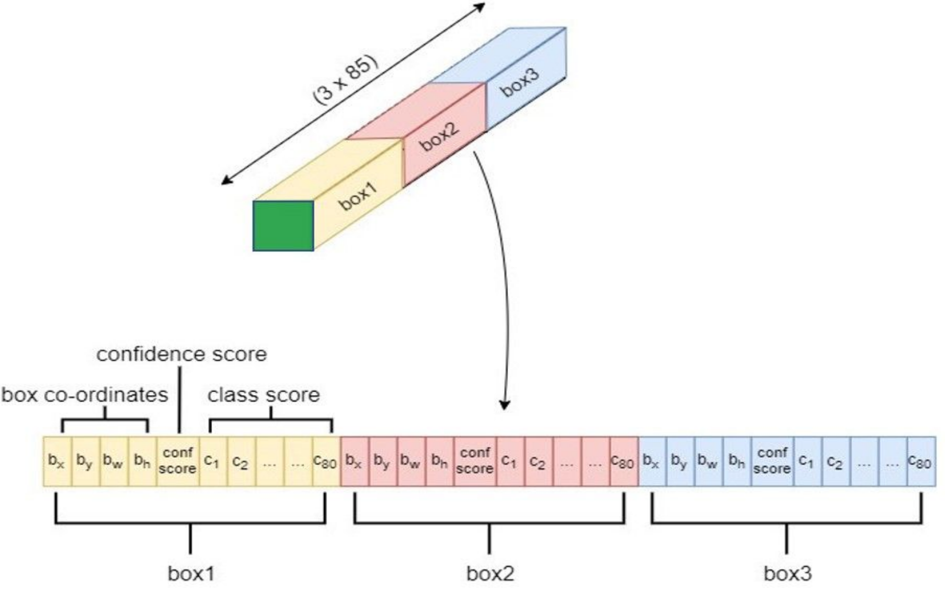


Fig. 4. Bounding box components

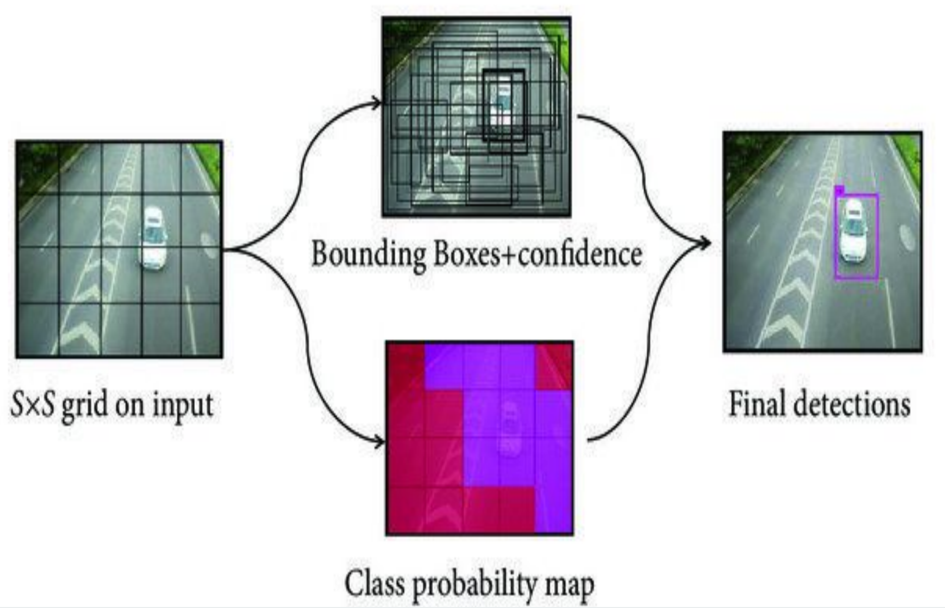
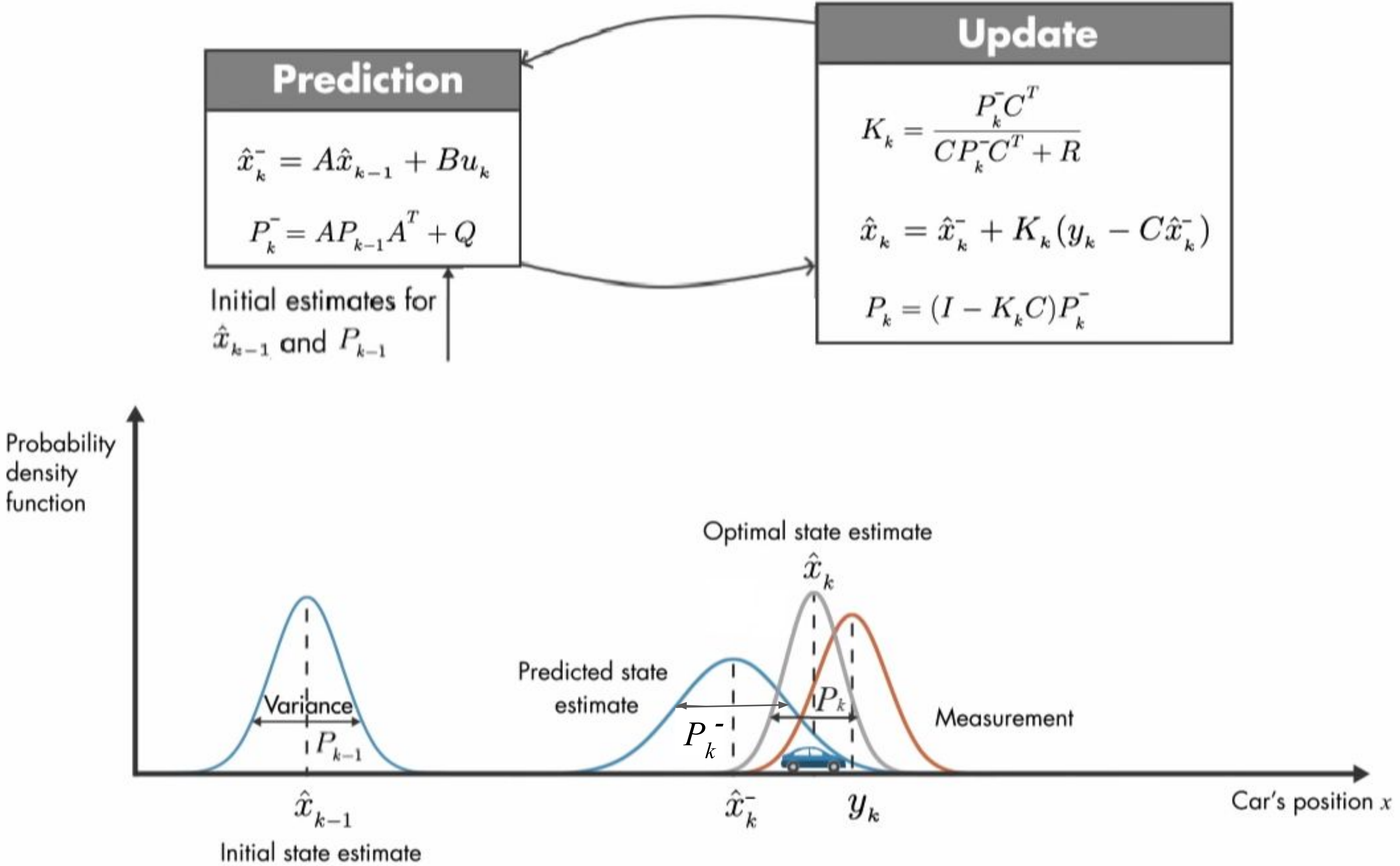


Fig. 5. S * S grid cells of YOLO system

2. Vehicle tracking with Kalman filter

- Kalman filter is an optimal state estimate algorithm used to estimate a system state.
- It is used to track the detected vehicles in given video. It consists of two steps:
  a. Prediction
  b. Update
- The Kalman filter determines the vehicles location by predicting its state at the current video frame, using detected location of previous video frame.
- The filter then uses the newly detected location of current frame to update the state, giving resultant vehicle location.
- If the vehicle is missing in current frame, then it relies only on its previous state to predict the current location of video frame.


Prediction
$\hat{x}_k^- = A\hat{x}_{k-1} + Bu_k$
$P_k^- = AP_{k-1}A^T + Q$
Update
$K_k = \frac{P_k^- C^T}{CP_k^- C^T + R}$
$\hat{x}_k = \hat{x}_k^- + K_k(y_k - C\hat{x}_k^-)$
$P_k = (I - K_k C)P_k^-$
Initial estimates for $\hat{x}_{k-1}$ and P_{k-1}
Probability density function
Optimal state estimate
$\hat{x}_k$
Predicted state estimate
Variance
P_{k-1}
P_k^-
P_k
Measurement
$\hat{x}_{k-1}$
Initial state estimate
$\hat{x}_k^-$
y_k
Car's position x

- **Inputs :** u_k is the input vector
  y_k is the measurement vector
- **Outputs :** x_k is the state estimations in update phase
  P_k is the error covariance in update phase
- **Variables :** x_{k-1} is the initial state estimate
  P_{k-1} is the initial state covariance
  x_k^- is the state estimations in prediction phase
  P_k^- is the error covariance in prediction phase
  K_k is the Kalman gain
- **Constants :** A is the state transition matrix
  B is the control matrix
  C is the observation matrix
  Q is the process noise covariance
  R is the measurement covariance

- The calculation of optimal state estimate x_k depends on the Kalman gain K_k .
- It decides whether to rely on the prediction state estimate or measurement state and gives final state estimate x_k.
- For every detection an unique ID is assigned and tracks are created respectively.
- The Hungarian algorithm is used to assign each detection to its corresponding unique ID.
- Hungarian algorithm tells us the correlation between detected vehicles from different frames.
- This process runs in loop for further frames and updates the assigned tracks and deletes lost tracks till the end of the video.

RESULTS OF KALMAN FILTER

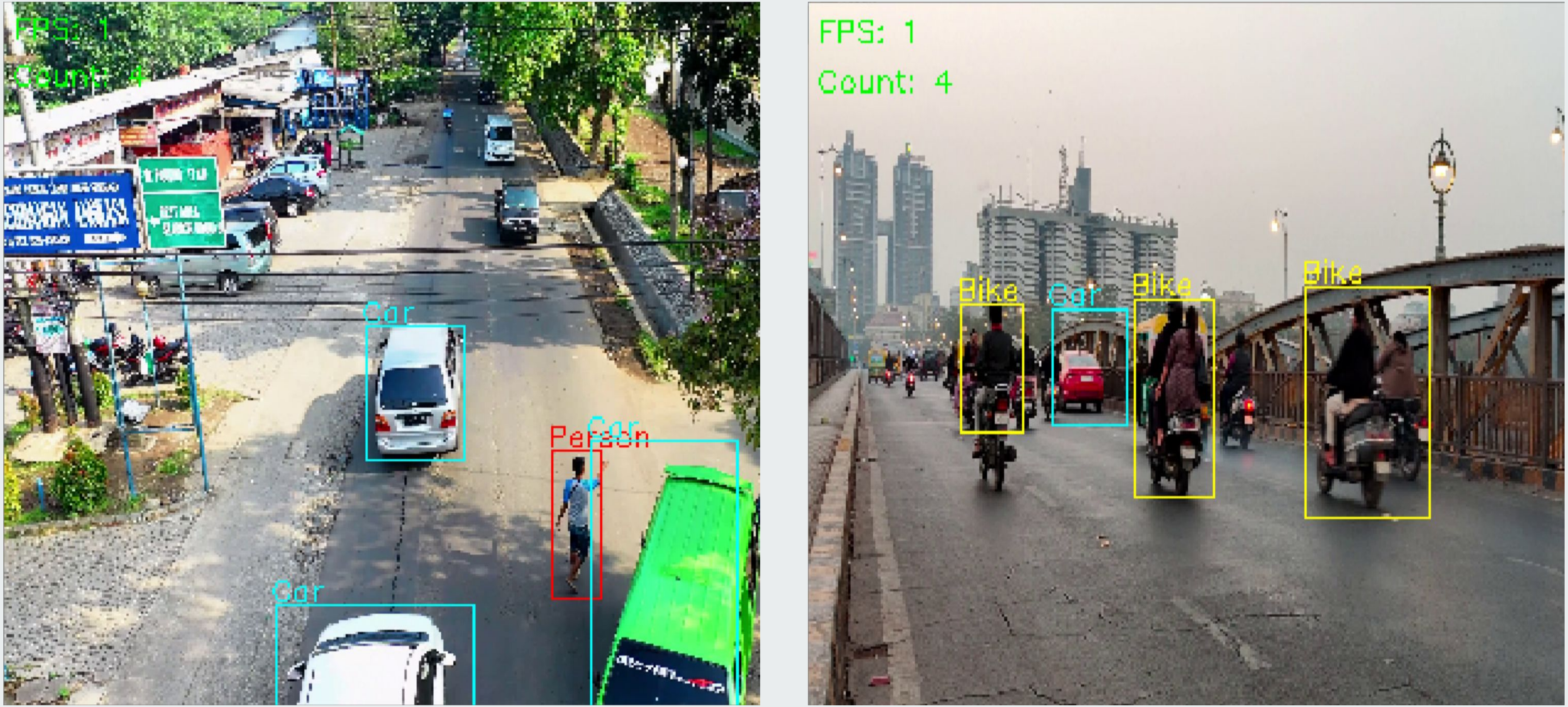


Fig. 6. Detection and tracking of cars, bikes and persons.

Drawbacks of Kalman filter approach

1. Kalman filter returns tracks with higher number of ID switches and is not suitable for occlusion.
2. In low light conditions the existing methodology does not give accurate detection and tracking results .
3. It is very susceptible to illumination and weather conditions and is not robust enough to get accurate results.
4. This system approach has more processing time and may not be suitable for real time scenarios.
5. Kalman filter cannot track vehicles when there is movement in camera, it can only work well for videos taken from static cameras.

Proposed Tracking Algorithm : DeepSORT

Simple Online Real-time Tracking (SORT)

1. Detection
2. Estimation
3. Association &
4. Track Identity creation and destruction.

- ❑ YOLOv3 object detector provides vehicle detections, Kalman filter creates tracks for vehicle detections and the Hungarian algorithm associates detections to tracked objects.
- ❑ When vehicles enter and leave in a frame, unique identities need to be created or destroyed accordingly.

- In terms of tracking accuracy and precision SORT works admirably.
- However, SORT fails in cases of occlusion, as it returns tracks with a more number of ID switches. The association matrix used is to be blamed for this.
- A more effective association metric is used by DeepSORT, which integrates both motion and appearance descriptors.

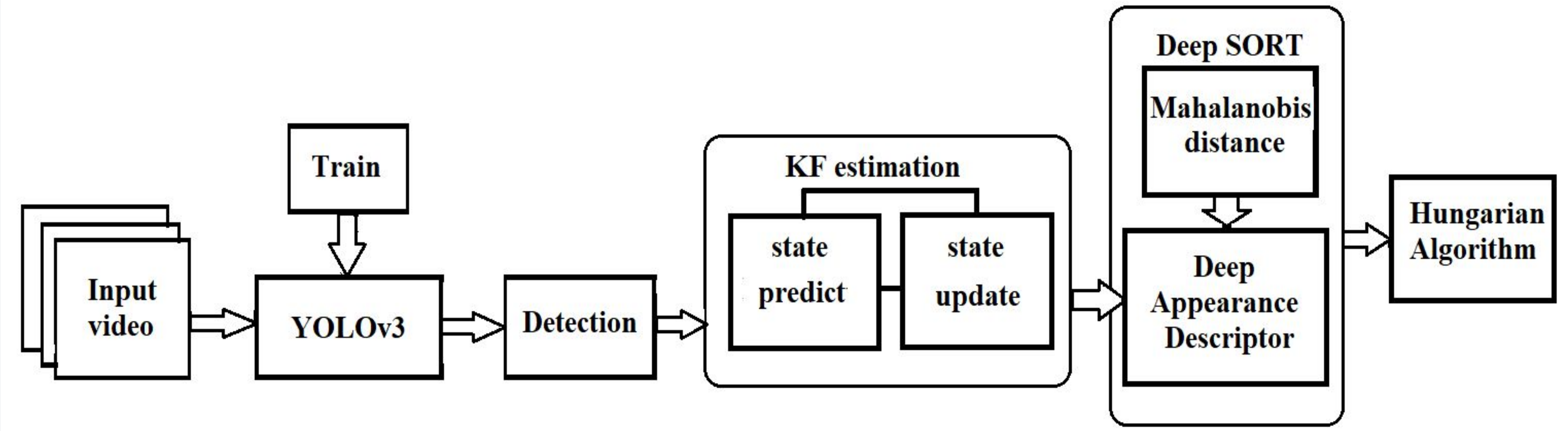


Fig. 7. Block diagram of DeepSORT

- DeepSORT is a tracking algorithm that follows things not only based on their motion and velocity but also on their outward appearance.
- A classifier is created using dataset and trained until it reaches a respectable level of accuracy, giving a single feature vector known as the appearance descriptor.
- The Appearance feature Vector is used to track and assign the vehicles to tracks.

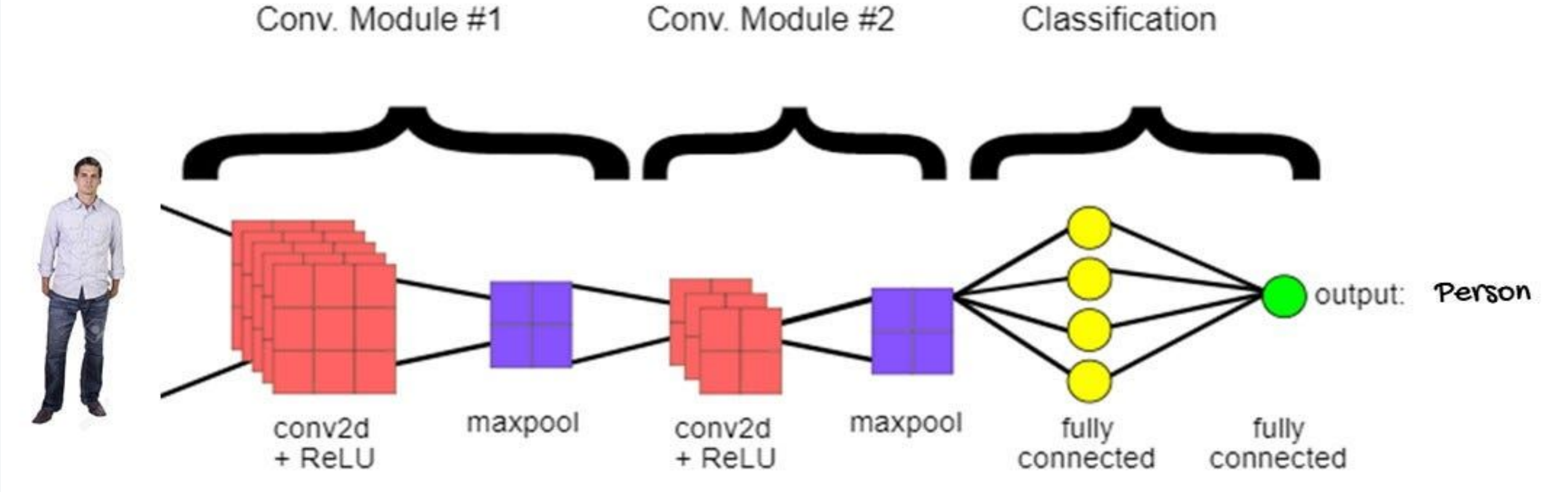


Fig. 8. Network architecture for appearance factor of DeepSORT [5]

- After the appearance descriptor is obtained the Measurement-to-track association (MTA) is established which determines the relation between a measurement and an existing track.
- Deep sort uses Mahalanobis distance instead of Euclidean distance for MTA, to evaluate the predicted Kalman state and the new state as shown in the following formula :

$$d^{(1)}(i,j) = (d_j - y_i)^T \times S_i^{-1} \times (d_j - y_i)^T \qquad (1)$$

- Representing the motion matching degree between the j_{th} target and the i_{th} track, where S_i is the covariance matrix of the trajectory predicted by Kalman filter at the current time, y_i is the predicted observation of the trajectory at the current time, d_j is the state of the j_{th} target.

- How far a point is from the center of a multivariate normal distribution can be determined statistically using the Mahalanobis distance.

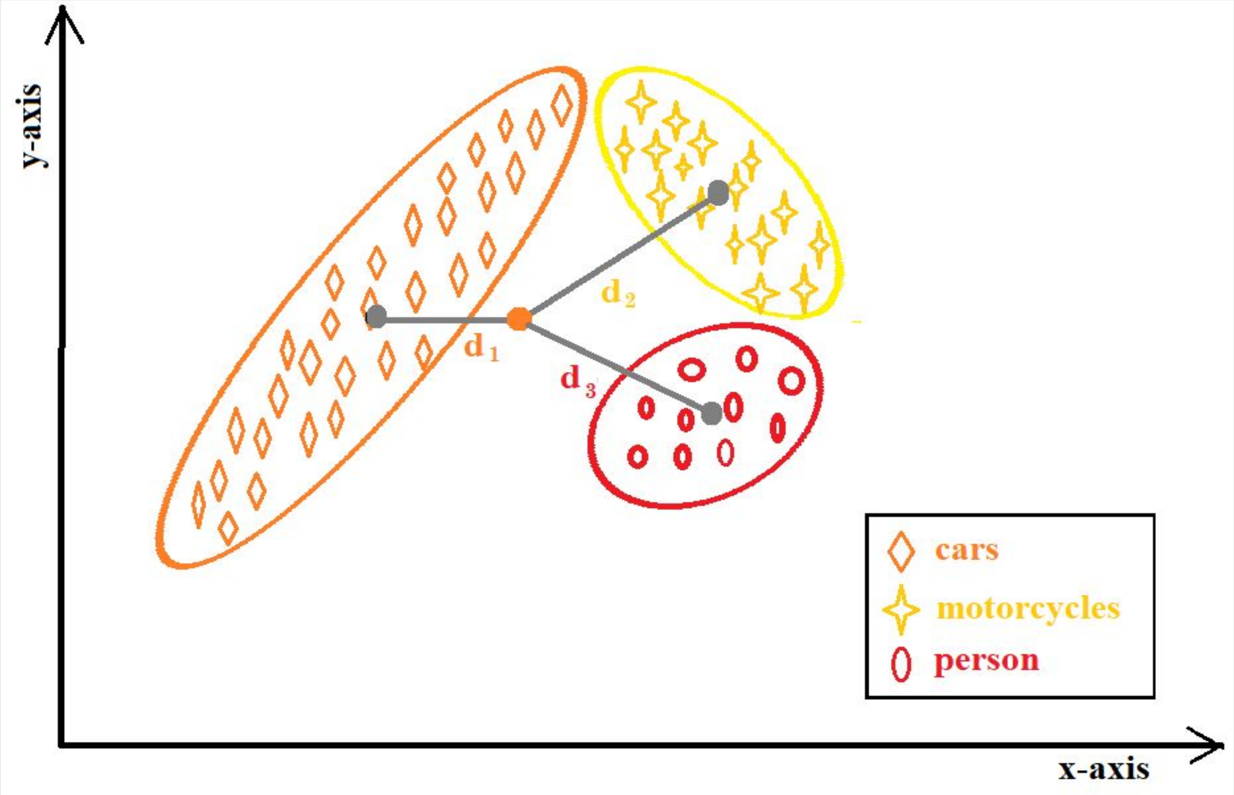


Fig. 9. Mahalanobis distance for d_1, d_2, d_3 three classes

EXPERIMENTAL RESULTS

- To obtain a dataset of traffic videos, different traffic videos are taken from different locations and from Google [6].
- These videos are of resolution 1920*1080 with 30 fps. Then all frames are extracted from every video and split into a 7:3 ratio and used as training dataset and testing dataset.
- The YOLOv3-416 model is then trained with 120 epochs. It took around seven hours in total to finish training of the custom model.
- The python software with Tensorflow and OpenCV libraries are used to implement the proposed model.

RESULTS

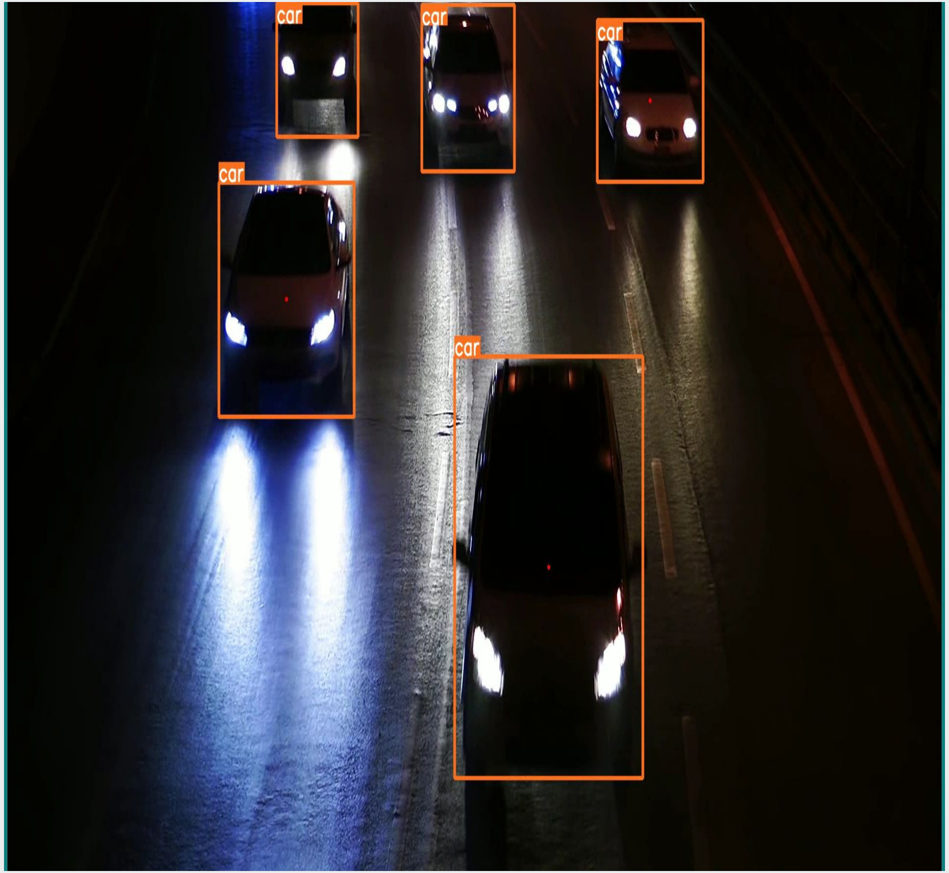


Fig 10 (a). Detection and tracking in low light

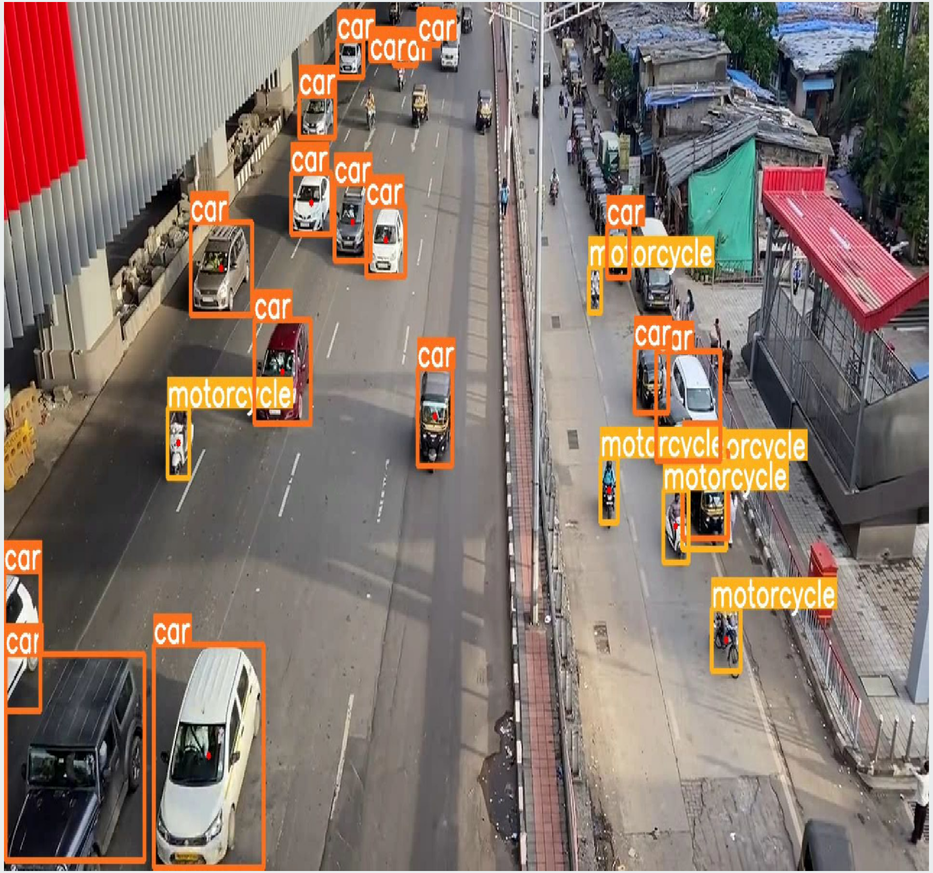


Fig 10 (b). Detection and tracking in daylight

3. Wrong way Traffic Violation Detection (WTVD) Algorithm

- In a one-way road, if vehicles come from two opposite directions, then it is called wrong-way traffic violation.
- WTVD algorithm defines the direction of movement of the vehicle.
- Direction cost ie., the difference between centroids of a detected vehicle is calculated between frames and direction is assigned to them.
- The difference is calculated as $\Delta A = A_5 - A_1$ and $\Delta B = B_5 - B_1$, ΔA is negative it is WEST and if ΔB is negative it is NORTH
- Vehicles moving in direction WEST and NORTH are considered as correct way.
- Vehicles moving in direction EAST and SOUTH are considered as violation.

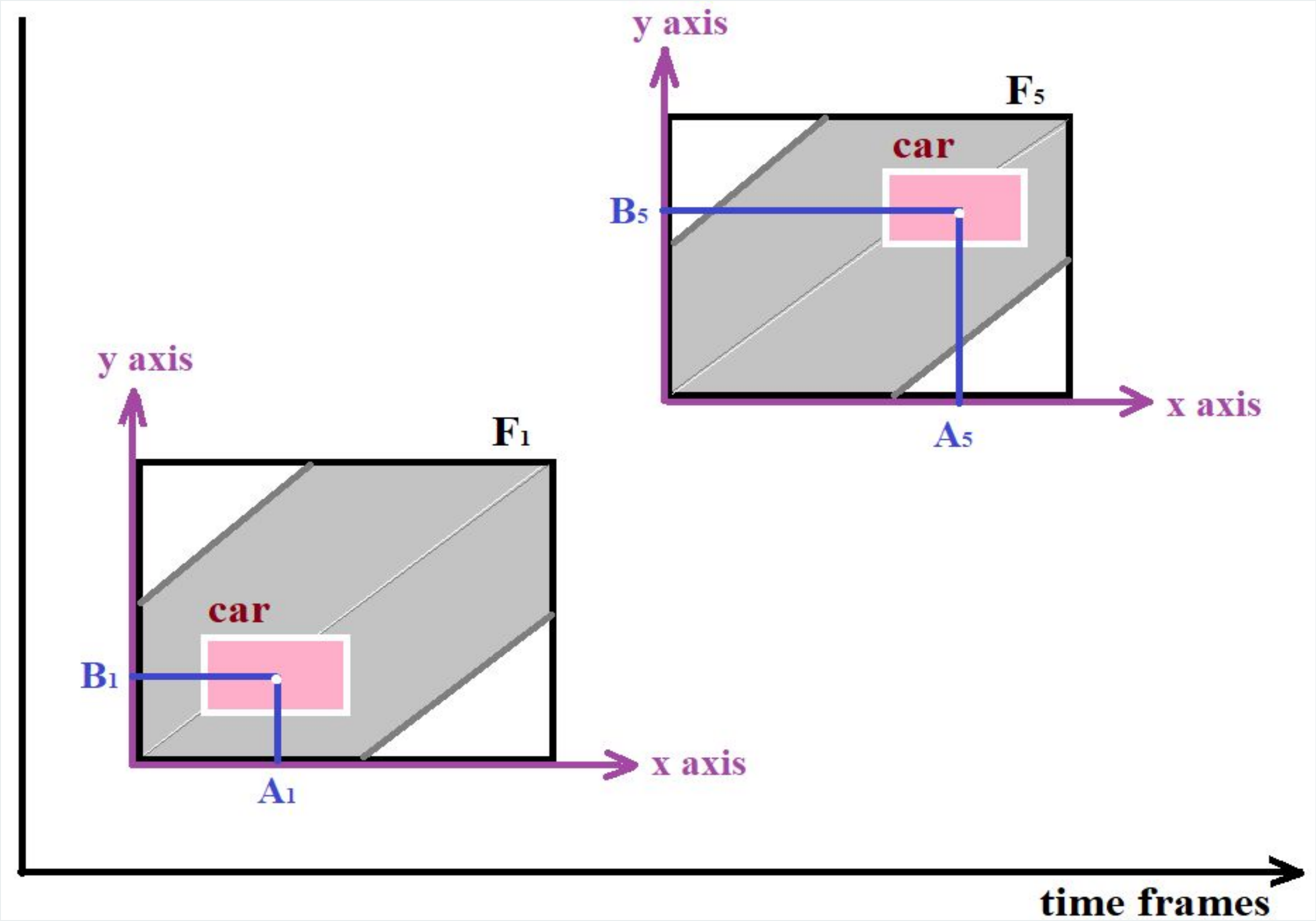


Fig. 11. Cars moving through the frames with centroids (A_1, B_1) and (A_5, B_5)

Fig. 12. Vehicles moving in wrong direction are detected by DeepSORT

4. Over Speed Violation Detection (OSVD) Algorithm

- Dynamic feature, such as velocity is extracted from the detected object and it is used for the detection of speed violation.
- The centroids of the vehicles are used to compute the velocity.
- During the detection and tracking task, bounding-box is fitted over the detected object.
- Coordinates of the centroid of such box is used to obtain the velocity of the detected object. The velocity is initially calculated in terms of pixels, and then it is transformed into Km/hr.
- The OSVD algorithm determines whether the speed is below the threshold value Th_1 or not. (Th_1 = 60 km/hr)

- By using the formulas listed below, the centroids of each vehicle (A_1, B_1) and (A_2, B_2) are utilized to determine the velocity.

$$Velocity = d\_met * fps * 3.6 \tag{2}$$

$$dpix = \sqrt{((A_2 - A_1)^2 + (B_2 - B_1)^2)} \tag{3}$$

$$d\_met = dpix \div ppm \tag{4}$$

- where *ppm* is pixel per meter, *fps* is frames per second, 3.6 is the factor used to convert speed into Km/hr, *d_met* is pixel to meter conversion, *dpix* is pixel distance travelled.
- **Note** : ppm is calculated by dividing car width in pixels by car width or road width in pixels by road width.

RESULTS

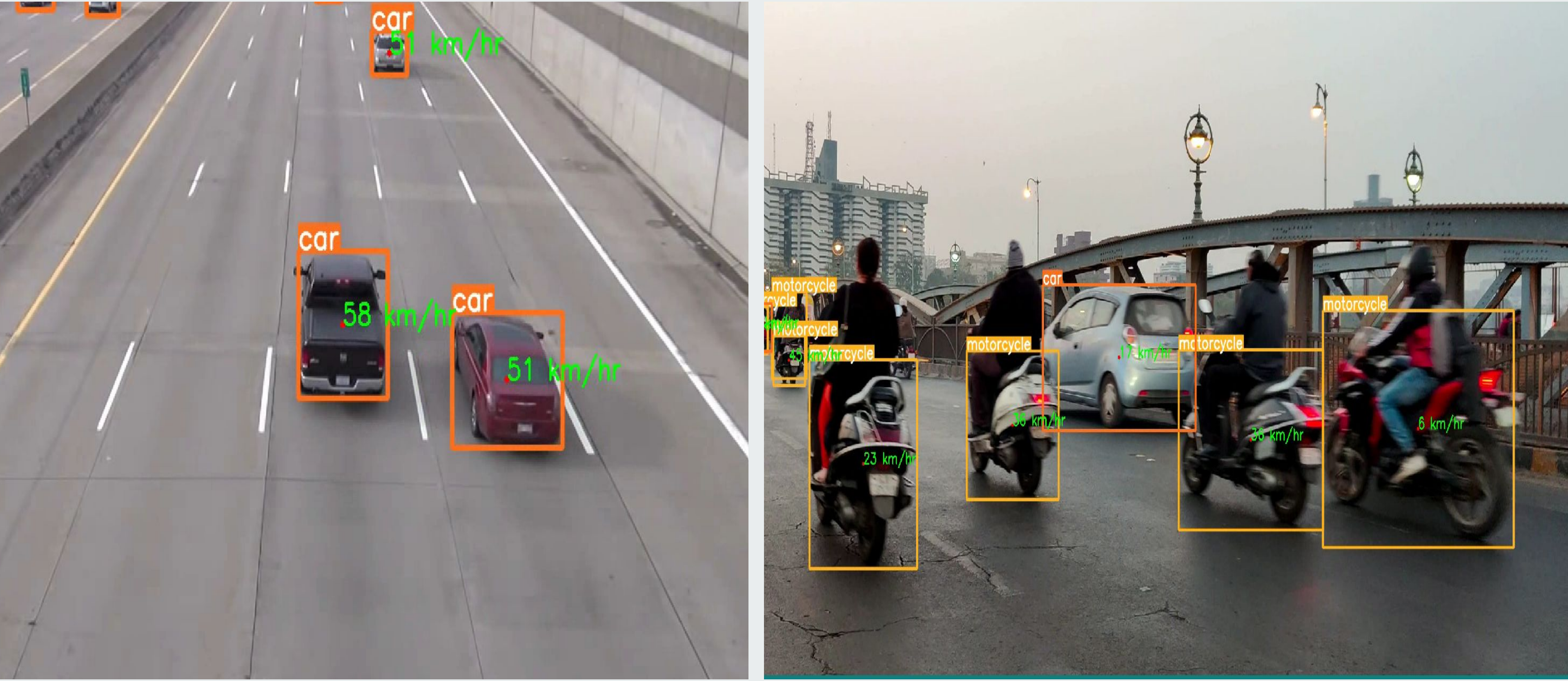


Fig. 13. Speed estimation of vehicles by using DeepSORT

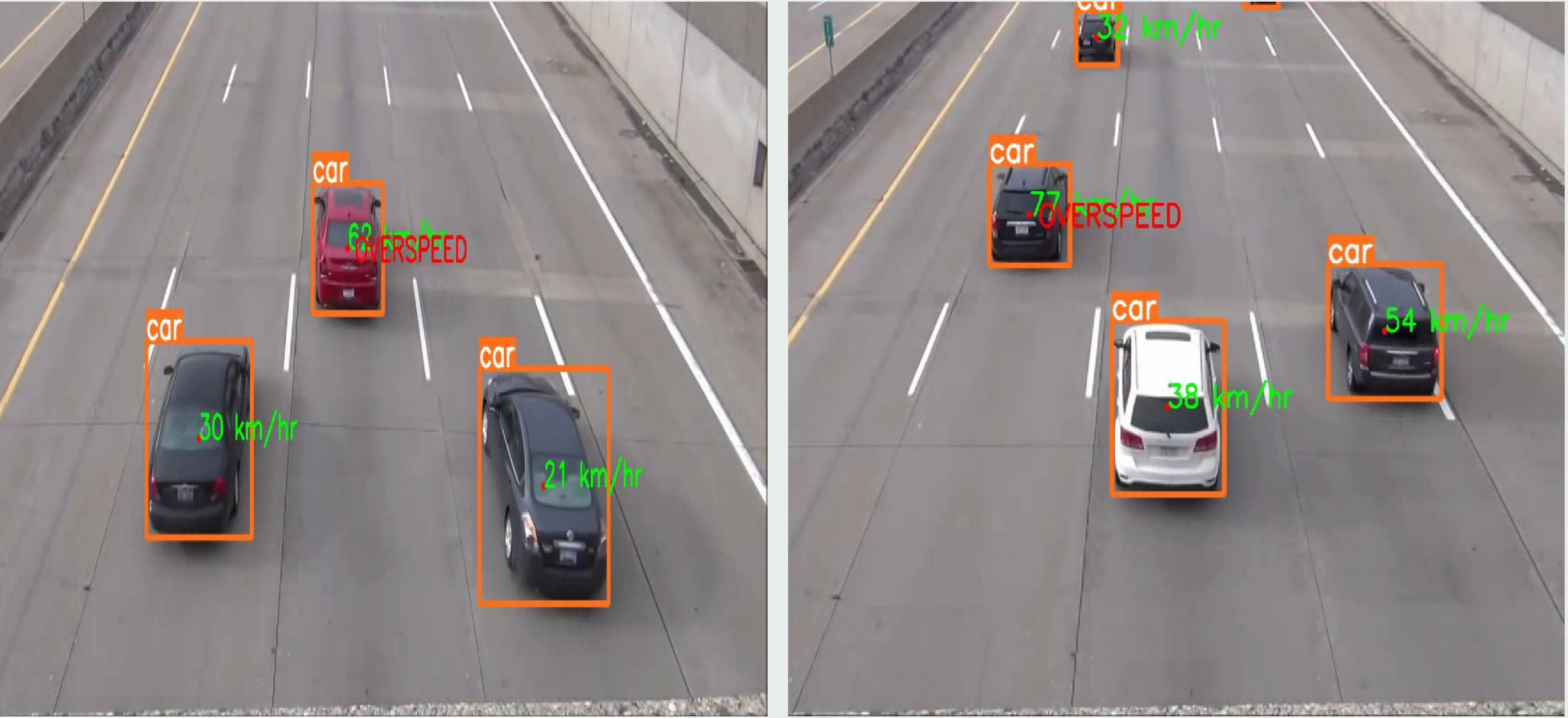


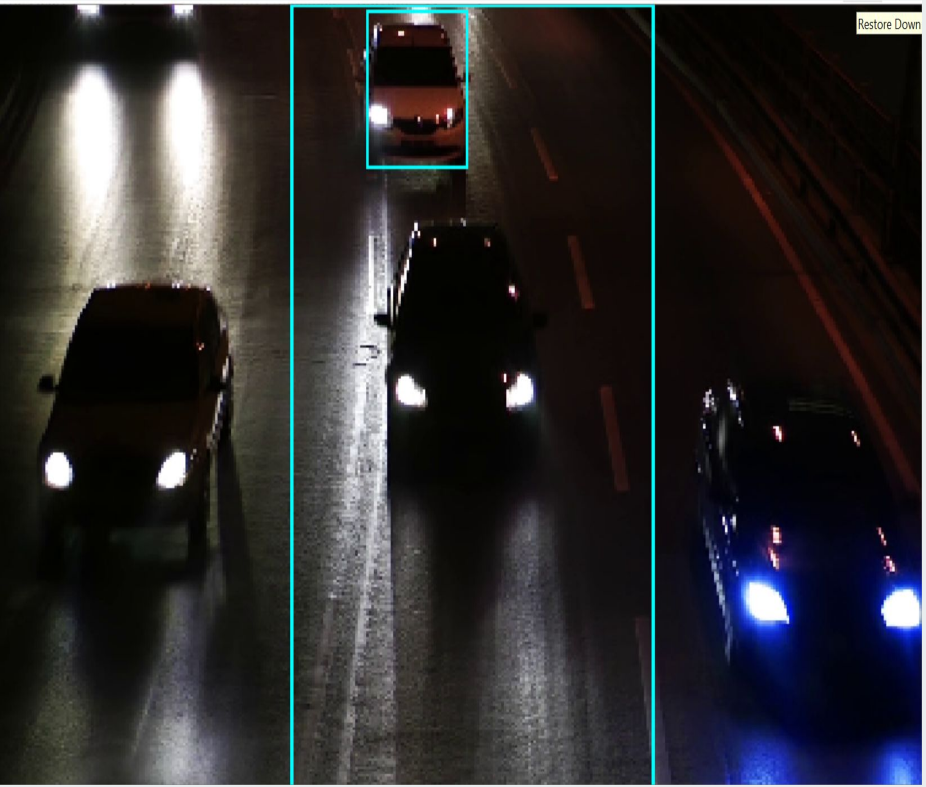
Fig. 14. Detection of cars with over speed by DeepSORT

COMPARISON OF KALMAN FILTER AND DEEPSORT

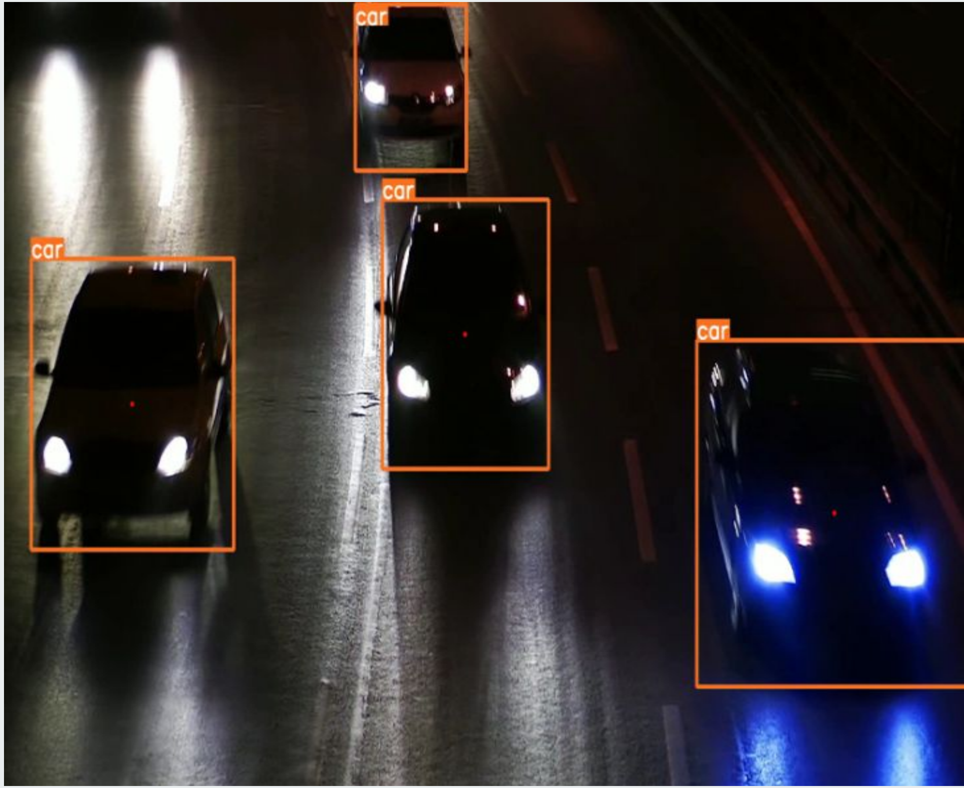
- In heavy traffic conditions vehicles are detected as shown in comparative Fig. 15.

Fig. 15. Comparison of number of car detections in heavy traffic conditions

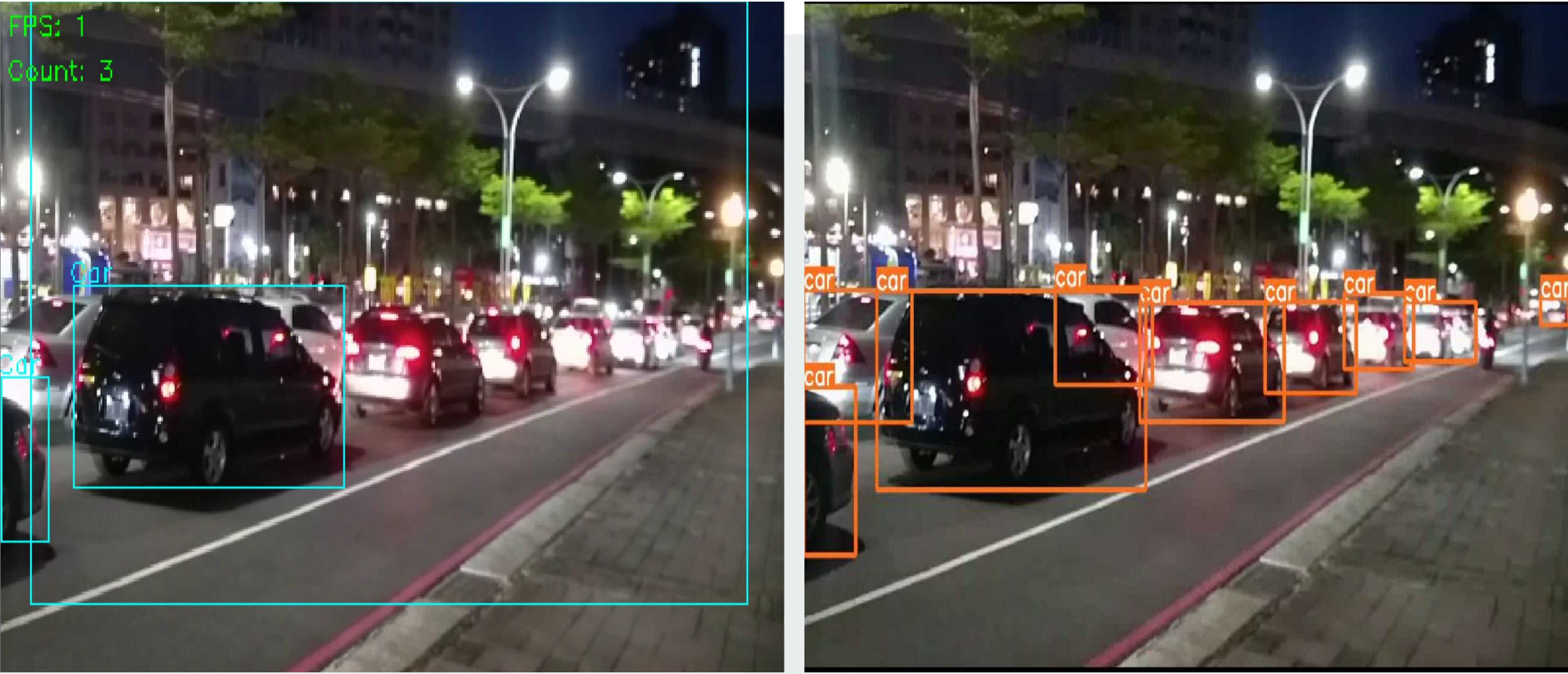
❑ In low light conditions the existing algorithm fails to detect and track the vehicles whereas with new improved DeepSORT it can be overcomed easily as shown in Fig. 16.



16 (a). Kalman filter



16 (b). DeepSORT



16 (c). Kalman filter

16 (d). DeepSORT

Fig. 16. Comparison of car detections in low light conditions

- In situations where cars are moving very close to each other ,existing methodology faces overlapping and incorrect ID assessment issues as shown in below Fig. 17.

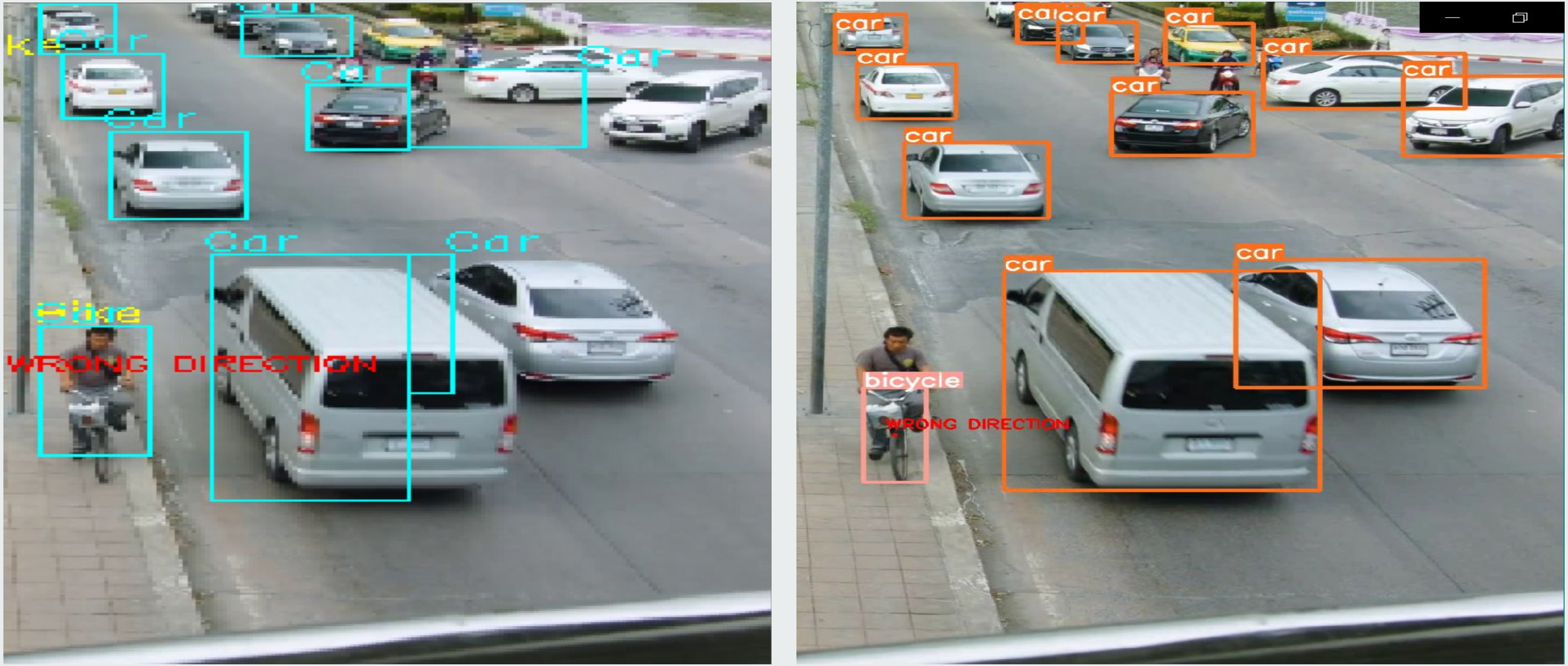
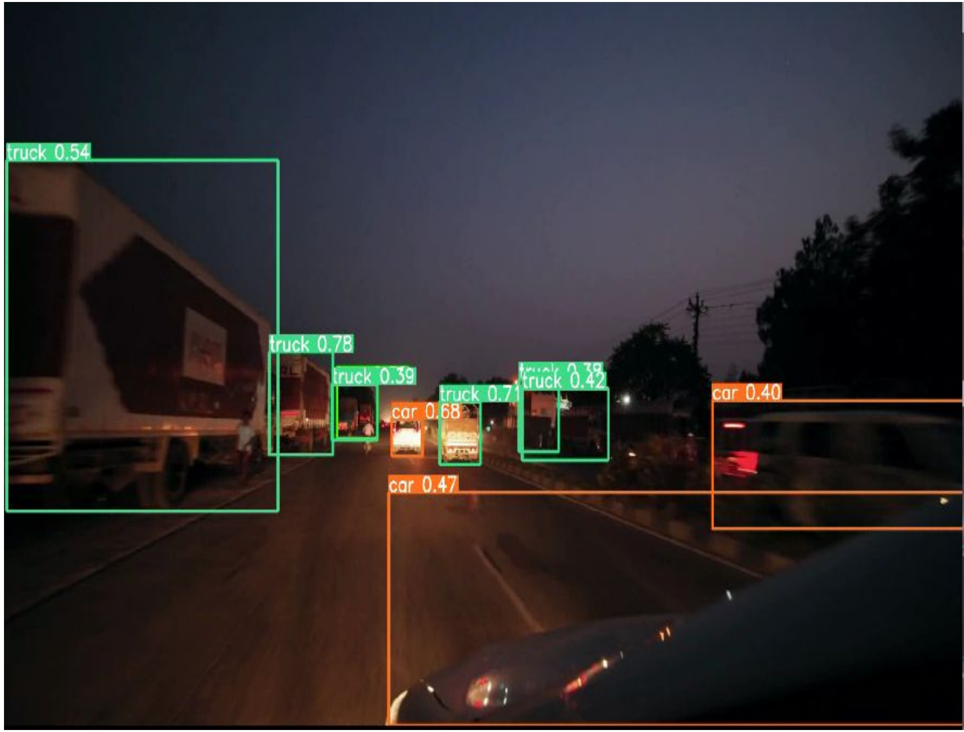


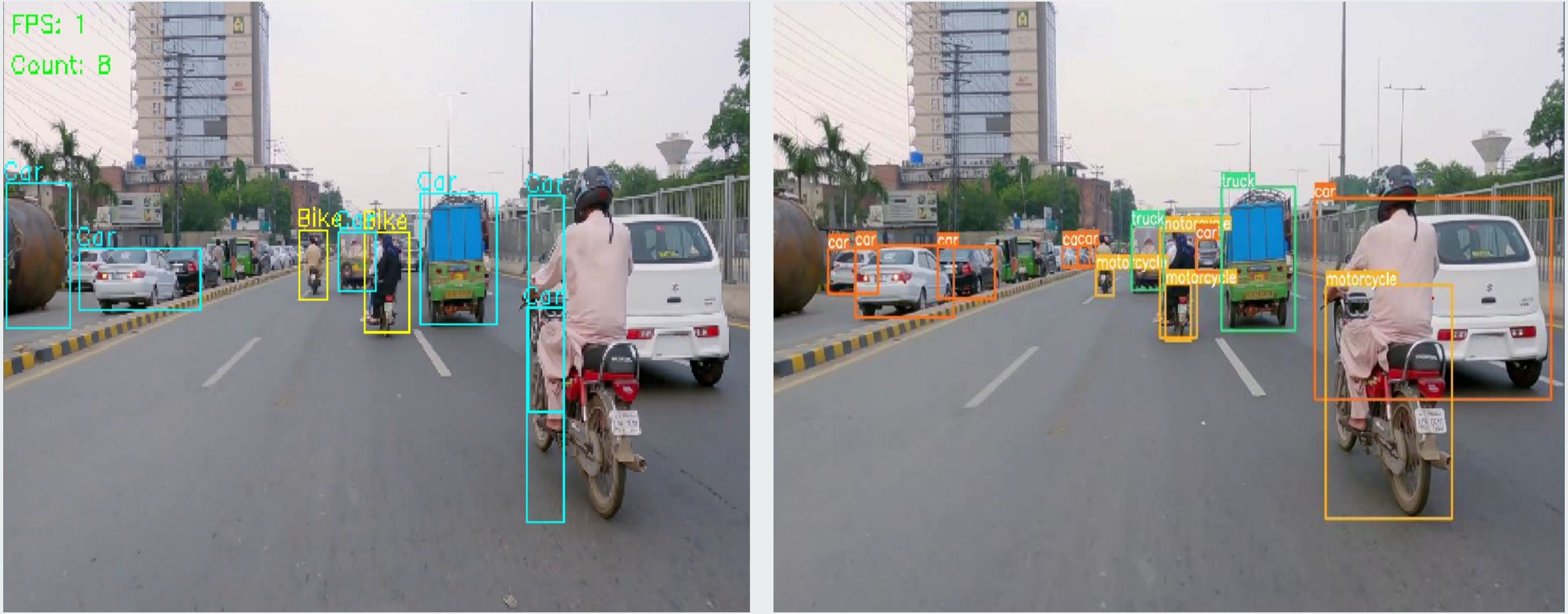
Fig. 17. Comparison of car detections in overlapping scenarios.

- When the camera is not static ,the Kalman filter does not work properly ,to overcome this DeepSORT is used ,a comparison of results is shown in Fig. 18.

18 (a)



18 (b)



18 (c)

18 (d)

Fig. 18. Comparison of car detections in non-static scenarios.

- DeepSORT performs much better than Kalman filter in terms of both preprocessing time and detection speed as illustrated in the below Table 1.

Table 1. Comparison table of Kalman filter and DeepSORT with different input videos

Input Video	**Time Duration**	**Preprocessing Time**		**Detection Speed**	
		Kalman filter	*DeepSORT*	*Kalman filter*	*DeepSORT*
video 1	40 s	6.15 s	1.20 ms	2.91 s	0.78 s
video 2	44 s	6.21 s	1.22 ms	2.98 s	0.73 s
video 3	55 s	6.87 s	1.31 ms	3.56 s	0.81 s
video 4	68 s	8.09 s	1.40 ms	4.10 s	0.84 s

APPLICATIONS

1. This can be used to detect traffic rule violations and sending warnings to police in real time without any human interaction.
2. It is very useful for self-driving vehicles and in exploring unnatural traffic scenarios.
3. It can be implemented in real time by feeding a realtime CCTV footage path as input.

ADVANTAGES

1. This model detects the over speeding and wrong way traffic violations with single integrated system i.e., it can detect multiple traffic rule violations.
2. The proposed algorithms have greater speed and greater accuracy compared to other existing algorithms.
3. DeepSORT used to track the vehicles is a very accurate deep learning algorithm which can overcome difficulties of low light conditions,occlusions and harsh weather conditions.

CONCLUSION

- The proposed system detects and tracks vehicles successfully using YOLOv3 and DeepSORT.
- The designed algorithm, detects traffic rule violations like overspeeding and wrong way traffic violation successfully.
- When compared to existing method, proposed system has improved performance in speed and accuracy .
- It also shows greater performance improvement in low light conditions.
- The proposed model has an accuracy of 98.2% in detecting vehicles moving in the wrong direction and accuracy of 96.4% in detecting vehicles with over speed.

FUTURE SCOPE

- The proposed approach exclusively concentrates on the two types of traffic violations which are wrong way driving and over speed, despite the fact that several traffic violations and offenses are to blame for traffic accidents and fatalities.
- A framework that can address other traffic violations and offenses that cause traffic accidents and fatalities can be developed as an objective for the future.
- Designing and implementing an integrated system with different algorithms which can detect : lane changing violation, parking violation, overtaking violations, can be seen as the main future scope of this project.

PAPER PUBLICATIONS

[1] A. Manasa and Renuka Devi S M.: An Enhanced Real-Time System For Wrong-way and Over Speed Violation Detection Using Deep Learning. In : 4th International Conference on Image Processing and Capsule Networks (ICIPCN), Bangkok, Thailand , August 2023.

REFERENCES

1. R. J. Franklin and Mohana, "Traffic Signal Violation Detection using Artificial Intelligence and Deep Learning," 2020 5th International Conference on Communication and Electronics Systems (ICCES), Coimbatore, India, 2020, pp. 839-844, doi: 10.1109/ICCES48766.2020.9137873.

2. Anima Pramanik, Sobhan Sarkar, J. Maiti, A real-time video surveillance system for traffic pre-events detection, Accident Analysis & Prevention,Volume 154,2021,106019,ISSN 0001-4575, https://doi.org/10.1016/j.aap.2021.106019.

3. H. R. Mampilayil and R. K., "Deep learning based Detection of One Way Traffic Rule Violation of Three Wheeler Vehicles," 2019 International Conference on Intelligent Computing and Control Systems (ICCS), Madurai, India, 2019, pp. 1453-1457, doi: 10.1109/ICCS45141.2019.9065638.

4. Usmankhujaev S, Bay Dadaev S, Woo KJ. Real-Time, Deep Learning Based Wrong Direction Detection. Applied Sciences. 2020; 10(7):2453. https://doi.org/10.3390/app10072453

5. https://medium.com/augmented-startups/deepsort-deep-learning-applied-to-object-tracking

6. https://www.videezy.com/free-video/car

Thank you