ResearchGate

See discussions, stats, and author profiles for this publication at: https://www.researchgate.net/publication/356137079

Vehicle Tracking based on an Improved DeepSORT Algorithm and the YOLOv4 Framework

Conference Paper · August 2021

DOI: 10.1109/ICIAfS52090.2021.9606052

CITATIONS
7

READS
1,164

9 authors, including:

Imalie Perera
University of Peradeniya

1 PUBLICATION **7** CITATIONS

SEE PROFILE

Aseni Jayarathne
University of Peradeniya

1 PUBLICATION **7** CITATIONS

SEE PROFILE

Shehan Kaushalya Senavirathna
Rensselaer Polytechnic Institute

3 PUBLICATIONS **8** CITATIONS

SEE PROFILE

Shamendra Egodawela
RMIT University

1 PUBLICATION **7** CITATIONS

SEE PROFILE

All content following this page was uploaded by Shehan Kaushalya Senavirathna on 11 June 2022.

The user has requested enhancement of the downloaded file.

Vehicle Tracking based on an Improved DeepSORT Algorithm and the YOLOv4 Framework

Imalie Perera[1], Shehan Senavirathna[1], Aseni Jayarathne[1], Shamendra Egodawela[1],
Roshan Godaliyadda[1], Parakrama Ekanayake[1], Janaka Wijayakulasooriya[1],
Vijitha Herath[1] and Sathindra Sathyaprasad[2]
[1]Department of Electrical and Electronic Engineering, University of Peradeniya, Peradeniya 20400, Sri Lanka
[2]Department of Civil Engineering, University of Peradeniya, Peradeniya 20400, Sri Lanka
Email: {imaliep101, shehan.senevirathna, chamandikajayarathne, shamendrae}@gmail.com,
{roshangodd, mpb.ekanayake, jan, vijitha}@ee.pdn.ac.lk, imss@eng.pdn.ac.lk

2021 10th International Conference on Information and Automation for Sustainability (ICIAfS) | 978-1-6654-4429-3/21/$31.00 ©2021 IEEE | DOI: 10.1109/ICIAfS52090.2021.9606052

***Abstract*—Vehicle tracking plays an important role in traffic surveillance systems in which efficient traffic management is the main objective. During the last several decades, with the rapid growth of the number of vehicles, the task of detecting and tracking vehicles efficiently and accurately has become challenging. Existing algorithms often fail to track vehicles continuously throughout the video stream due to the nonlinear nature of vehicular motion and vehicle occlusion in crowded scenarios. This paper proposes a vehicle tracking algorithm as an improvement on DeepSORT (Simple Online and Realtime Tracking with a Deep Association Metric), based on an optimized YOLOv4 (You Only Look Once version 4) detector, the Unscented Kalman filter, and AlexNet as the feature extraction network which ensures better performance in tracking the non-linear motion of vehicles and tracking through occlusions with a reduced number of ID switches compared to state-of-the-art vehicle trackers.**

Keywords—YOLOv4, Object Detection, Object Tracking, DeepSORT, Traffic Surveillance

I. Introduction

With the rapid advancements in multimedia capturing technology, video surveillance devices have emerged as valuable tools in traffic management applications. Over the years, researchers have introduced a wide array of vehicle detection and tracking algorithms based on video surveillance footage to facilitate efficient traffic management.

Many traditional approaches for object detection have been explored in the literature and blob extraction proposed by S. P. Dhole *et al.* in [1] is one such widely used technique in object detection aimed at detecting regions of images with consistent properties. This method of detection is limited by the requirement of a clear distinction between background and foreground as well as lighting conditions. One main drawback of blob extraction for object detection is the uncertainty in detections as shadows and nearby objects often tend to be grouped together with the object.

Object tracking has also been an area of interest for many years. Among the traditional approaches of object tracking found in the literature, optical flow-based tracking methods which estimate the instantaneous velocity of pixel motion using spatial and temporal brightness variations as proposed by L. Kurnianggoro *et al.* in [2], have been widely implemented. This method of object tracking is constrained by the assumption of brightness constancy and spatial coherence. The main drawback of Optical flow as stated by S. M. K. C. S. B. Egodawela *et al.* is that it can only detect two orthogonal directions of motion. This is not a feasible solution for unstructured scenes where the high-level representation of the movement of objects in frames is not uniform and structured. It is also computationally intensive and therefore fails to meet the requirements of real-time tracking. These traditional vehicle detection and tracking techniques may fall short also due to the occlusion of vehicles by other vehicles in highly crowded scenarios or by background obstacles such as road signs and trees and due to the nonlinear motion patterns of vehicles as evident by the experimental results.

Now, deep learning-based frameworks although data-hungry, are gaining increased attention from researchers due to their promising results in modelling complex non-linearities which are patent in most practical scenarios. These algorithms are mostly based on Convolutional Neural Networks (CNNs) which are constructed using multiple building blocks such as convolution layers, pooling layers, and fully connected layers. In contrast, as mentioned earlier, traditional methods of object detection and tracking utilize traditional video processing techniques which perform poorly under ambient lighting changes, occlusions etc.

YOLO proposed by J. Redmon *et al.* in [4] is a state-of-the-art object detection algorithm based on a single CNN which is capable of predicting multiple bounding box coordinates and class probabilities efficiently to an acceptable accuracy given adequate training data. YOLO approaches object detection as a regression problem. It divides the input image into grid cells and predicts bounding boxes for each grid cell along with the confidence levels for those bounding boxes and conditional class probabilities. The confidence and the conditional class probabilities are used to obtain class-specific confidence scores for each box. Experimental results have shown that YOLO is more than twice as accurate as prior work on real-time detection [4]. YOLOv2 [5], YOLOv3 [6] and YOLOv4 [7] are built on the previous frameworks of YOLO, with each release surpassing the previous versions in terms of speed and accuracy.

SORT (Simple Online Realtime Tracking) proposed by A. Bewley *et al.* in [8] is an algorithm developed for real-time multiple object tracking. SORT tackles the problem of tracking under four stages, namely, detection, track estimation, association, and track identity creation and removal. The constant velocity Kalman filter is a key component of the SORT algorithm which is used to propagate detections from the current frame to the next. These predictions are then used to calculate the IoU (Intersection over Union) with the new detections for the association of tracks. It is also the mechanism through which objects are tracked during occlusions.

DeepSORT [9] which is an improved version of SORT is one of the most popular state-of-the-art object tracking frameworks today. DeepSORT has integrated a pre-trained

Authorized licensed use limited to: University of Wolverhampton. Downloaded on January 11,2022 at 09:01:14 UTC from IEEE Xplore. Restrictions apply.

neural network to generate feature vectors to be used as a deep association metric. Since DeepSORT was developed focusing on the Motion Analysis and Re-identification Set (MARS) dataset [10], which is a large-scale video-based human reidentification dataset, it uses a feature extractor trained on humans which does not perform well on vehicles.

Several state-of-the-art object detection and tracking algorithms including SORT and DeepSORT were deployed by V. Mandal and Y. Adu-Gyamfi in [11] to detect and track different classes of vehicles in their region of interest and it has been stated that the trackers did not perform ideally at predicting vehicle trajectories which resulted in ID switches during occlusions. A vehicle tracking method proposed by Z. Li, Y. Chen and Z. Yin [12] fuses the prior information of the Kalman filter to solve the problem of vehicle tracking under occlusion. But it has been stated that the proposed method does not perform well if the target is lost for a longer period.

This paper focuses on constructing a vehicle tracking algorithm based on the framework suggested in Deep SORT. In the proposed method, we intend to resolve the issues of tracking the non-linear motion of vehicles and tracking through occlusions with a reduced number of ID switches. A YOLOv4 detector was successfully trained with a sample vehicle dataset and the detector was optimized to obtain detections with a high level of accuracy. Different prediction models were exploited with the aim of selecting the best model for the prediction of nonlinear vehicular motion. In addition, this study incorporates a transfer learning-based approach for feature extraction using AlexNet [13], which is a CNN trained on more than a million images from the ImageNet database [14]. These approaches are discussed in detail in section II. The experimental results and their effectiveness are demonstrated under section III. Finally, the study is concluded in section IV.

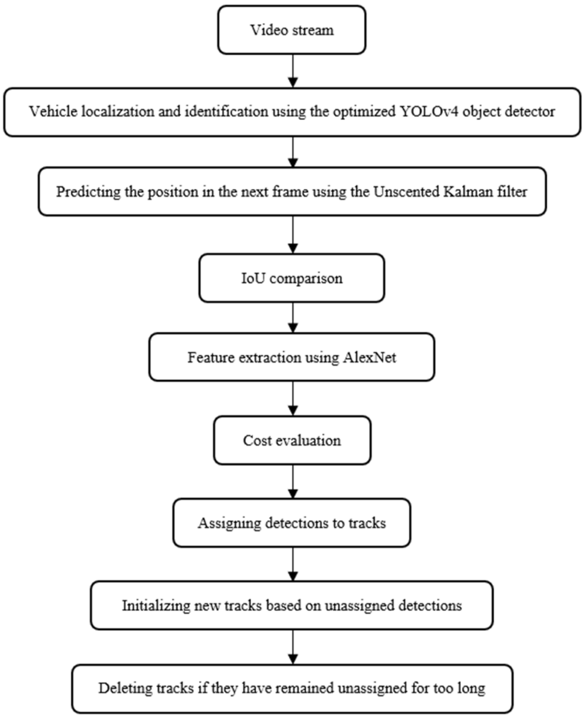

Video stream
Vehicle localization and identification using the optimized YOLOv4 object detector
Predicting the position in the next frame using the Unscented Kalman filter
IoU comparison
Feature extraction using AlexNet
Cost evaluation
Assigning detections to tracks
Initializing new tracks based on unassigned detections
Deleting tracks if they have remained unassigned for too long

Fig.1. Stages of the proposed algorithm

II. Methodology

In this section, stages of the proposed vehicle tracking algorithm are discussed as depicted in Fig.1. Video data were collected from a pole-mounted monocular camera and training and validation data sets were created by manually labelling a set of frames extracted from the video stream. YOLOv4 is used with Darknet [15], an open-source neural network framework written in C and CUDA, for vehicle localization and identification. This is followed by the prediction stage, IoU comparison, and feature extraction which are further discussed in the subsections to follow. Next, the cost matrix is evaluated using the weighted sum of the IoU and the cosine distance of feature vectors. Then, the vehicle detections are assigned to existing tracks using the Hungarian algorithm. Finally, new tracks are initialized based on unassigned detections and the tracks which have remained unassigned for a certain number of frames are deleted from the memory.

A. Training the Detector

A YOLOv4 detector was trained for vehicle localization and classification. To achieve the best training, this study focused on optimizing the training using the mAP (mean Average Precision) evaluation metric and the YOLOv4 training loss (YOLO loss). The size of the training set was improved by data augmentation techniques to make the system robust to ambient lighting changes, camera noise (grain noise, Gaussian noise, etc.), and camera shake.

Input resolution of images determines the accuracy as well as training and inference times. Larger pixel resolution may improve the accuracy while increasing the training and inference times. The resolution of the training images was set to 416x416 pixels to obtain reasonable accuracy.

The training was initiated using pre-trained weights (yolov4 conv.137) to reduce the time taken for training. This study consists of 6 classes of vehicles and the detector was trained with 2000 iterations/number of batches for each class. After 12000 iterations, an average loss of 0.24 was obtained. The mAP values at 0.5 IoU threshold for every 1000 iterations are represented in Fig.2 It can be observed that after 7000 iterations, the mAP shows only a slight variation. To avoid overfitting, the best weight file was selected by comparing the mAP and the Average Loss.

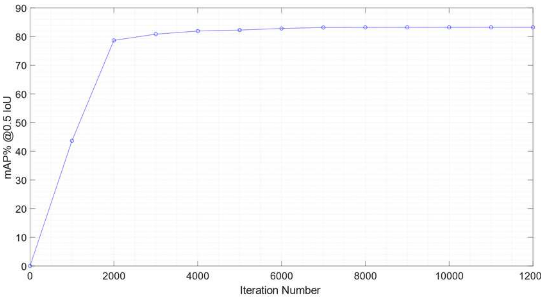


Fig.2. Plot of mAP% @0.5 IoU vs Iterations

B. Prediction stage

The Kalman filter is the most popular prediction algorithm used in state-of-the-art object trackers. But, through further analysis, it was evident that the Kalman filter performs poorly in predicting the position when it comes to nonlinear motion models (due to linear assumptions in state transitions). Therefore, as an improvement to the existing framework, an Unscented Kalman filter was selected as the best fit for the prediction of nonlinear vehicular motion.

Authorized licensed use limited to: University of Wolverhampton. Downloaded on January 11,2022 at 09:01:14 UTC from IEEE Xplore. Restrictions apply.

The unscented transform estimates a Gaussian distribution by applying the nonlinear transformation to a set of points called sigma points generated from the source gaussian distribution. The state used in the Unscented Kalman filter can be represented as $(\mathrm{cx}, \mathrm{vx}, \mathrm{cy}, \mathrm{vy}, \mathrm{w}, \mathrm{vw}, \mathrm{h}, \mathrm{vh})$ containing the bounding box centre position $(\mathrm{cx}, \mathrm{cy})$, aspect ratio w, height h, and their respective velocities.

The state transition matrix (1) is based on the constant velocity model where dt is equal to 1/frame rate. A set of 2n+1 (n - dimension of the state) sigma points is generated with the tuning parameters alpha, beta, kappa set to their optimum values. The sigma points are transformed into the vector $\hat{x}_k$ using the state transition matrix.

$$\begin{bmatrix} 1 & \mathrm{dt} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & \mathrm{dt} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & \mathrm{dt} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & \mathrm{dt} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \qquad (1)$$

Predicted mean:

$$x_k = \sum_{\mathrm{j=0}}^{\mathrm{2n}} W_j^m \hat{x}_k^{(j)}. \qquad (2)$$

Predicted covariance:

$$P_k^{\ x} = \sum_{\mathrm{j=0}}^{\mathrm{2n}} W_j^c\,(\hat{x}_k^{(j)} - x_k)\,(\hat{x}_k^{(j)} -)^T. \qquad (3)$$

Weights:

$$W_0^{\ m} = \frac{\mathrm{\gamma}^2 - \mathrm{n}}{\mathrm{\gamma}^2}\ , \qquad (4)$$

$$W_0^{\ c} = \frac{\mathrm{\gamma}^2 - \mathrm{n}}{\mathrm{\gamma}^2} + 1 - \alpha^2 + \beta, \qquad (5)$$

$$W_j^{\ m} = W_j^{\ c} = \frac{1}{2\mathrm{\gamma}^2}\ ; j \ \neq 0, \qquad (6)$$

$$\mathrm{\gamma} = \sqrt{\alpha^2(n+\kappa)}, \qquad (7)$$

$\alpha, \beta, \kappa - Tuning\ parameters.$

The sigma points are transformed into vector $\hat{y}_k^{\ (j)}$ using measurement function h:

$$\hat{y}_k^{\ (j)} = h(\hat{y}_{k-1}^{\ \ (j)}). \qquad (8)$$

Mean in measurement space:

$$\hat{y}_k = \sum_{\mathrm{j=0}}^{\mathrm{2n}} W_j^m \hat{y}_k^{(j)}. \qquad (9)$$

Covariance in measurement space:

$$P_k^{\ y} = \sum_{\mathrm{j=0}}^{\mathrm{2n}} W_j^c (\hat{y}_k^{(j)} - \hat{y}_k)(\hat{y}_k^{(j)} - \hat{y}_k)^T. \qquad (10)$$

The cross covariance between x_k and $\hat{y}_k$:

$$P_k^{\ xy} = \sum_{\mathrm{j=0}}^{\mathrm{2n}} W_j^c (\hat{x}_k^{(j)} - x_k)(\hat{y}_k^{(j)} - \hat{y}_k)^T. \qquad (11)$$

The measurement update of the state estimate:

$$K_k = P_k^{\ xy}(P_k^{\ y})^{-1}, \qquad (12)$$

$$\hat{x}_k = x_k + K_k(y_k - \hat{y}_k). \qquad (13)$$

$$P_k = P_k^{\ x} - K_k P_k^{\ y} K_k^{\ T}, \qquad (14)$$

$K - Kalman\ gain.$

C. IoU Comparison

As the name implies, IoU represents the ratio of overlap to the union of a projection and a detection. High IoU captures the idea that the object is present at the predicted position and therefore IoU represents the accuracy of the prediction model. The IoU values of the Kalman predictions and the detections of the current frame are calculated to assign tracks to vehicles. With an accurate prediction model, the IoU of the same vehicle was observed to be a very high value. By observing these values, a threshold was set for IoU comparison.

D. Feature Extraction

A feature vector is a vector that contains information of an object's prominent features i.e., colour, shape, scale, etc. To extract features in detections every YOLOv4 detection is sent through a feature extractor. Here, as an improvement to the existing DeepSORT architecture, AlexNet is used as the feature extractor replacing the existing network trained on the MARS dataset.

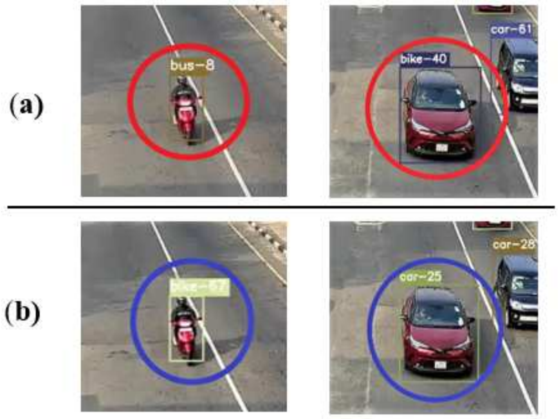
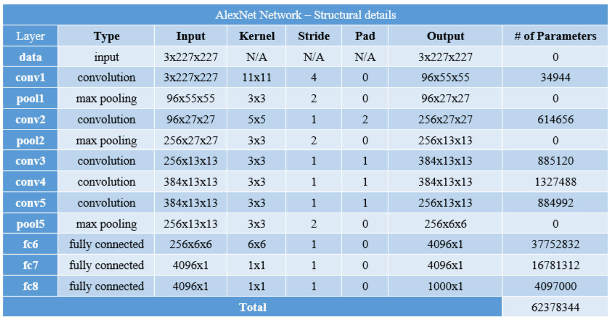


Fig.3. Results of the (a) Unoptimized, (b) Optimized detectors



AlexNet Network – Structural details							
Layer	Type	Input	Kernel	Stride	Pad	Output	# of Parameters
data	input	3x227x227	N/A	N/A	N/A	3x227x227	0
conv1	convolution	3x227x227	11x11	4	0	96x55x55	34944
pool1	max pooling	96x55x55	3x3	2	0	96x27x27	0
conv2	convolution	96x27x27	5x5	1	2	256x27x27	614656
pool2	max pooling	256x27x27	3x3	2	0	256x13x13	0
conv3	convolution	256x13x13	3x3	1	1	384x13x13	885120
conv4	convolution	384x13x13	3x3	1	1	384x13x13	1327488
conv5	convolution	384x13x13	3x3	1	1	256x13x13	884992
pool5	max pooling	256x13x13	3x3	2	0	256x6x6	0
fc6	fully connected	256x6x6	6x6	1	0	4096x1	37752832
fc7	fully connected	4096x1	1x1	1	0	4096x1	16781312
fc8	fully connected	4096x1	1x1	1	0	1000x1	4097000
Total							62378344

Fig.4. Structural details of the AlexNet network

The AlexNet CNN consists of 5 convolutional layers and the 3 fully connected layers as shown in Fig.4. The output of AlexNet is obtained from fc7 as a feature vector of size 4096x1 for each detection in a frame. This procedure is carried out for each frame and the cosine distances are calculated within a search area between detections in consecutive frames to check the similarity of existing tracks and detections. The cosine similarity measure is used to compare the similarity of feature vectors in high dimensional inner product space. The detections having a cosine distance greater than a threshold are considered for the association.

Authorized licensed use limited to: University of Wolverhampton. Downloaded on January 11,2022 at 09:01:14 UTC from IEEE Xplore. Restrictions apply.

III. Experiments

A. *YOLOv4 detector*

A large number of false labels and label switches were present prior to the optimization of the detector. These errors were minimized by the optimized detector as shown in Fig.3.

B. *Prediction model*

Tracking data were analyzed to find the best prediction model for the nonlinear motion of vehicles. A plot of the position measurement along the y-axis of the image for a single vehicle and the prediction of each model with the frame number is shown in Fig.5. It can be observed by the experimental results that a significant error is present in the position estimate of the constant velocity Kalman filter. This is mainly due to the fact that the Kalman filter is a linear estimation algorithm and therefore it fails to capture the nonlinearities of vehicular motion. To get a better prediction model, a constant acceleration Kalman filter was applied and while it was fairly accurate in very low levels of nonlinearities, it also failed to capture the higher levels of nonlinearities. Thus, two more prediction algorithms were tested out, the Extended Kalman Filter and the Unscented Kalman filter. It is evident from the analysis that these models are a much better fit for the prediction of nonlinear motion.

The plot in Fig. 6 was obtained for another vehicle with a nonlinear motion pattern. It is evident from this that the Unscented Kalman filter produces the closest approximation for the position measurement of a vehicle. The mean square errors of each prediction model are tabulated in TABLE I.

Based on the results, the Unscented Kalman filter was selected as the best prediction model for the vehicle tracking algorithm.

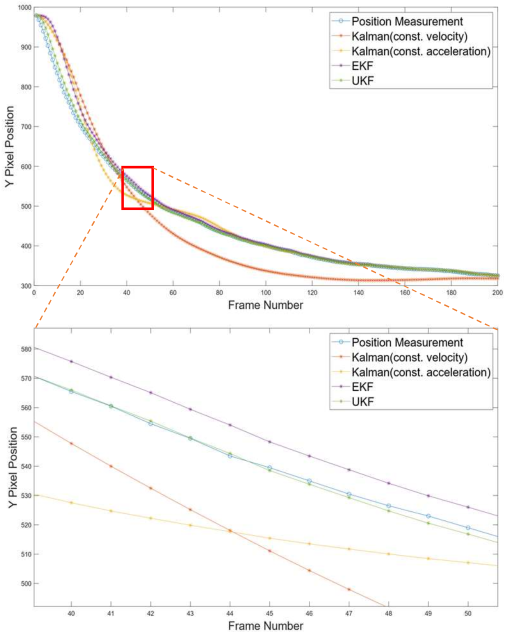


Fig.5. Motion estimation of a single vehicle using constant velocity, constant acceleration, Extended, and Unscented Kalman Filters

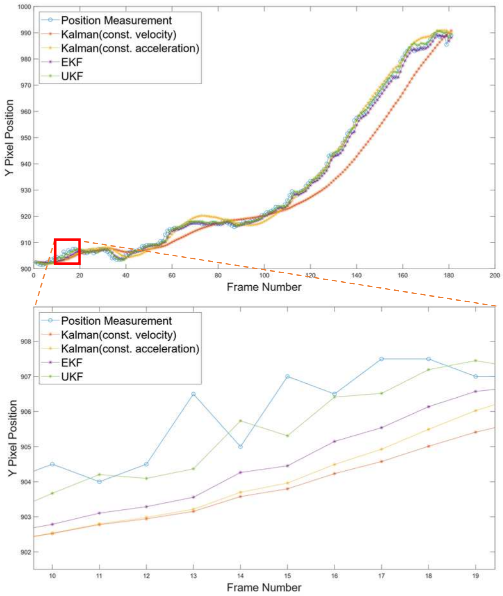


Fig.6. Motion estimation of a single vehicle using constant velocity, constant acceleration, Extended, and Unscented Kalman Filters

TABLE I. MEAN SQUARE ERROR

	Constant velocity Kalman Filter	**Constant acceleration Kalman Filter**	**Extended Kalman Filter**	**Unscented Kalman Filter**
Fig.4	2365.81	597.66	480.62	50.89
Fig.5	40.51	3.41	1.99	0.99

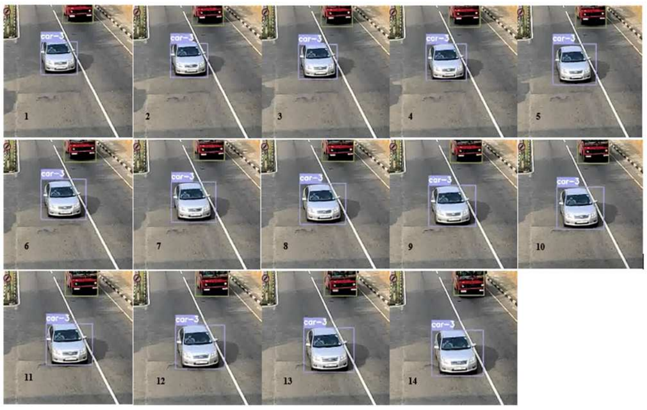


Fig.7. The appearance of a vehicle through 14 consecutive frames

Authorized licensed use limited to: University of Wolverhampton. Downloaded on January 11,2022 at 09:01:14 UTC from IEEE Xplore. Restrictions apply.

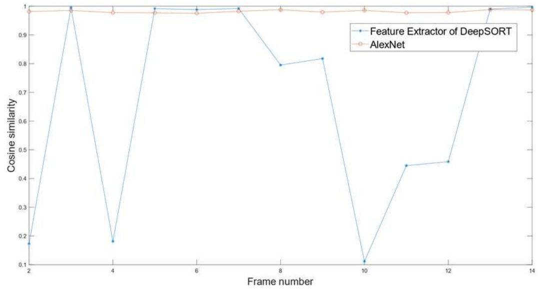


Fig.8. Comparison of the cosine similarity obtained from the DeepSORT Feature Extractor and the AlexNet Feature Extractor

C. Appearance Descriptor

Fig.8 is a comparison between the cosine similarity of the vehicle shown in Fig.7 obtained using the integrated feature extractor of DeepSORT and the Alexnet feature extractor. It can be observed that The output of the Alexnet feature extractor is much more consistent in terms of accuracy. In consecutive frames, The cosine distances evaluated using the AlexNet feature vectors are close to 1 throughout the video whereas the cosine distances of the DeepSORT feature vectors fluctuate from frame to frame. These fluctuations are in direct relation to the large number of ID switches in the experiments conducted for vehicle tracking using DeepSORT. Therefore, in this study, as an improvement to the existing DeepSORT architecture, AlexNet is used as the feature extractor replacing the existing network trained on the MARS dataset

D. Summary of experiments

In summary, the detector was improved and false labels and label switches were minimized by optimizing the training of YOLOv4. Nonlinear motion of vehicles could be predicted accurately with the addition of the Unscented Kalman filter and in turn, prediction errors were minimized and vehicles could be tracked accurately through occlusions. Furthermore, ID switches and tracking errors were reduced by the addition of the Unscented Kalman filter and the AlexNet feature extraction network as shown in Fig.9 and Fig.10.

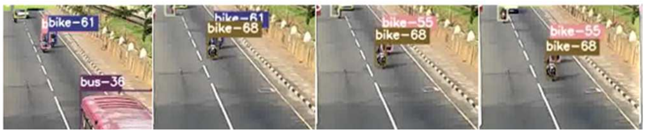


Fig.9. ID switches before the addition of the Unscented Kalman Filter and AlexNet Feature Extractor

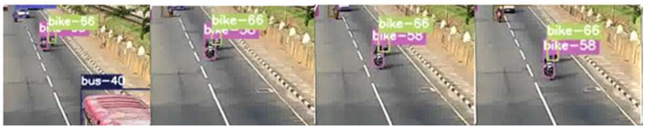


Fig.10. Improvements made by the Unscented Kalman Filter and AlexNet Feature Extractor

IV. Conclusion

In this paper, we have proposed a vehicle tracking algorithm based on the framework suggested in DeepSORT which is capable of tracking the nonlinear motion of vehicles with a high level of accuracy. The proposed algorithm utilizes YOLOv4 with Darknet, an open-source neural network framework, for vehicle localization and identification. The number of detection errors was minimized by optimizing the training of the detector through hyperparameter optimization and data augmentation. As a modification to the DeepSORT implementation which is incapable of capturing nonlinear motion, the unscented Kalman filter is used to obtain highly accurate track predictions which in turn reduces the errors in track association significantly. AlexNet, a pre-trained convolutional neural network, is used to perform feature extraction which is an integral part of the tracking algorithm aimed at further reducing the errors in track association. The experimental results demonstrate that the proposed vehicle tracking algorithm ensures better performance in tracking the non-linear motion of vehicles and tracking through occlusions with a reduced number of ID switches compared to the state-of-the-art object trackers.

V. References

[1] S. P. Dhole, V. R. Gadewar, D. Raut, R. Chandwadkar, "Object Detection Using Blob Extraction", International Journal of Emerging Technologies and Innovative Research, ISSN:2349-5162, Vol.1, Issue 6, November-2014, pp. 396-399.

[2] L. Kurnianggoro, A. Shahbaz, K.H. Jo, "Dense optical flow in stabilized scenes for moving object detection from a moving camera[C]", International Conference on Control, Automation and Systems. IEEE, 2016, pp. 704-708.

[3] S. M. K. C. S. B. Egodawela *et al*., "Vehicle Detection and Localization for Autonomous Traffic Monitoring Systems in Unstructured Crowded Scenes", 2020 IEEE 15th International Conference on Industrial and Information Systems (ICIIS), RUPNAGAR, India, 2020, pp. 192-197, doi: 10.1109/ICIIS51140.2020.9342663.

[4] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You Only Look Once: Unified, Real-Time Object Detection", in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegan, NV, USA, 2016, pp. 779–788.

[5] J. Redmon, A. Farhadi, "YOLO9000: better, faster, stronger", arXiv preprint. 2017.

[6] J. Redmon, A. Farhadi, "YOLOv3: An incremental improvement", arXiv:1804.02767, 2018.

[7] A. Bochkovskiy, C. Wang, H.M. Liao "YOLOv4: Optimal Speed and Accuracy of Object Detection", Apr 2020.

[8] A. Bewley, Z. Ge, L. Ott, F. Ramos and B. Upcroft, "Simple online and realtime tracking", 2016 IEEE International Conference on Image Processing, Phoenix, AZ, 2016, pp. 3464-3468.

[9] N. Wojke, A. Bewley, D. Paulus, "Simple online and realtime tracking with a deep association metric", 2017 IEEE International Conference on Image Processing, 2017, pp.3645-3649.

[10] L. Zheng, Z. Bie, Y. Sun, J. Wang, C. Su, S. Wang, and Q. Tian, "Mars: A video benchmark for large-scale person re-identification", European Conference on Computer Vision, 2016.

[11] V. Mandal, Y. Adu-Gyamfi, "Object Detection and Tracking Algorithms for Vehicle Counting: A Comparative Analysis", J. Big Data Anal. Transp. 2, 251–261, 2020.

[12] Z. Li, Y. Chen and Z. Yin, "Vehicle tracking fusing the prior information of Kalman filter under occlusion conditions". SN Appl. Sci. 1, 822, 2019.

[13] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks", Advances in neural information processing systems, 2012, pp. 1097–1105.

[14] J. Deng, W. Dong, R. Socher, L. J. Li, K. Li and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database", Computer Vision and Pattern Recognition 2009. CVPR 2009. IEEE Conference on, pp. 248-255, June 2009.

[15] J. Redmon. Darknet: Open source neural networks in c. http://pjreddie.com/darknet/, 2013–2016.

Authorized licensed use limited to: University of Wolverhampton. Downloaded on January 11,2022 at 09:01:14 UTC from IEEE Xplore. Restrictions apply.

View publication stats