JETIR.ORG ISSN: 2349-5162 | ESTD Year : 2014 | Monthly Issue
JOURNAL OF EMERGING TECHNOLOGIES AND INNOVATIVE RESEARCH (JETIR)
An International Scholarly Open Access, Peer-reviewed, Refereed Journal

Drive Guard: Advanced Road Traffic Vehicle Detection and Tracking

G.Ruthvik1, G.Shashank2, M.Sai Teja3, K.Krishna Vamshi4, D.Kalpana5, Dr. S.Shivaprasad6
*1,2,3,4 Students of the Department of CSE - Data Science, Malla Reddy Engineering College, Maisammaguda, Medchal, Hyderabad, Telangana.
*5 Assistant Professor, Department of CSE - Data Science, Malla Reddy Engineering College, Maisammaguda, Medchal, Hyderabad, Telangana.
*6 Professor, Department of CSE - Data Science, Malla Reddy Engineering College, Maisammaguda, Medchal, Hyderabad, Telangana

Abstract : Deep learning is revolutionizing smart cities and societies, solving many longstanding problems. Transportation is continuing to cause unbelievable damages including 1.25 million deaths and trillions of dollars annually. This paper presents a study on the use of YOLOv4 for vehicle detection and DeepSORT for tracking the detected vehicles on roads. None of the earlier works have applied these models to road traffic in the Kingdom of Saudi Arabia (KSA). We have used three different variations of the deep learning models and compared their performance; a pre-trained model with the COCO dataset, and two custom-trained models with the Berkeley DeepDrive dataset, and our custom-developed dataset obtained by a Dash Cam installed onboard vehicle driven on KSA roads in five different traffic conditions; city traffic in day and night, highway traffic in day and night, and traffic in rain. We have used Google Colab platform to harness GPU power, CUDA and OpenCV. The results have been evaluated using precision and other metrics.

IndexTerms **- Vehicle detection, tracking-by-detection, YOLO, DeepSORT, roadtraffic data**

I. INTRODUCTION

Artificial intelligence (AI) has taken us by storm, helping us to make decisions in everything we do, even in finding our "true love" and the "significant other" [1]. Machine learning and deep learning are revolutionizing smart cities and societies [2–4] and solving many longstanding problems in sectors such as health [5], education [6], and others. Transportation is the backbone of modern societies and economies albeit continuing to cause unbelievable damages of the likes of over a million deaths, 20-50 million injuries to people, and trillions of dollars, each year, around the globe [7]. Traditional approaches of measuring and monitoring road traffic rely on inductive loops that give some basic information about average speed, vehicle occupancy, and traffic flow. The traditional technologies are unable to support real-time monitoring and management of road traffic.

Deep learning has the potential to revolutionize many fields and transportation is not an exception. Among the deep learning methods for road transportation, there are methods that involve image and video analysis for automated traffic monitoring with applications in detecting traffic congestion, road safety, and many more. This paper presents a study on the use of off the shelf, highly optimized, deep learning models for vehicle detection and vehicle tracking on roads. While the earlier works have focused on the same, none of the works have applied these models to road traffic in Kingdom of Saudi Arabia (KSA). The vehicles, roads, traffic, traffic conditions, etc. in KSA are different from other countries in several respects such as the mix of traffic involving various vehicles manufacturer and models, the driving culture, road infrastructure, language, and more. We use YOLOv4 for object detection and DeepSORT for tracking the detected vehicles. We have used three different variations of the deep learning models and compared their performance; a pre-trained model with the COCO dataset, and two custom-trained models with different datasets. We have used three different datasets; the COCO dataset [8], the Berkeley DeepDrive dataset [9], and our custom- developed dataset obtained by a Dash Cam installed onboard vehicle driven on city streets and highways in the Kingdom of Saudi Arabia (KSA). We have collected data in five different traffic conditions, city traffic in day and night, highway traffic in day and night, and traffic in rain. For experiments, we have used Google Colab platform to harness GPU power, CUDA and OpenCV. The results have been evaluated using precision and other metrics. The rest of the paper is organized as follows. Section 2 reviews the relevant works. Section 3 details our methodology, model design, and datasets. Section 4 presents results and analysis. Section 5 concludes the discussion.

2. LITERATURE REVIEW

We have mentioned earlier that road transportation while connecting people and economies causes major damages to people, their health and lives, the economies and the planet environment. Traffic congestion caused by weather conditions, construction work and other unforeseeable events along the roads, can hinder the efficiency of other services and cause possible damages in urban cities. As cities and societies grow larger, it is important to come up with more practical solutions supporting their infrastructure.

Intelligent Transportation System (ITS) is where ICT meets traffic management and transport with the goal of making smarter, more informed decisions and enhance the quality of services provide by city authorities.
Researchers have integrated many technologies in ITSs tackling issues like incident prediction, event detection, road detection, traffic analysis for diverse purposes from urban planning to autonomous driving. Within these innovative solutions, technologies like inductive loops, Bluetooth, machine learning, big data and HPC are deployed.

2.1 Inductive Loops & Wireless Sensor Networks

Inductive loops are among the earliest methods to measure road traffic. Ali et al. [10] used inductive loop sensors to detect and count diverse vehicles in lane-less roads. They developed a multiple loop system with a new structure for inductive loop sensors. Their solution was able to sense vehicles and divide them by type. During testing, the system provided accurate counting of vehicles despite the heterogenous traffic conditions. Jeng and Chu [11] combined inductive loop signature data with WIM data in their proposed solution. They aimed to track heavy vehicles by utilizing IDL locations spread all over the network hence not restricting the tracking process between two WIM stations. Bhaskar et al. [12] presented an indicative-loop based solution to controller traffic lights. The solution addresses scenarios like reducing congestion in a particular lane, utilizing radio transmitter-receivers to make ways for public service vehicles such as firefighting and ambulance. While Khoenkaw and Pramokchon [13] deployed inductive-loops and proposed a low-cost controller for sensing vehicles. Other researchers combined several technologies to complement infrastructure-based solutions. Caceres et al. [14] used cellular systems for detecting phones moving on the road which in turns, aid in estimating traffic volumes. The efforts are meant to cover limitations of systems that required installed hardware such as cameras and inductive loops. Whereas Laharotte et al. [15] used Bluetooth data for traffic monitoring. They introduced Bluetooth origin-destination (B-OD) matrix which describes the dynamics of vehicles between two Bluetooth detectors. After proper processing, they proved that Bluetooth data can be of use in traffic managements. Díaz et al. [16] identified the lake of a formal methodology for vehicle travel time estimation within Bluetooth Traffic Monitoring Systems. After analyzing the Bluetooth features effecting TT estimation, they proposed a methodology devised to address TT estimation based on vehicles information only. Gheorghiu et al. [17] addressed the possibility of deploying the infotainment systems within vehicles for providing information about traffic conditions. Usually such systems are equipped with wireless communication abilities to function. Therefore, it is logical to utilized such abilities for traffic monitoring purposes.

2.2 Big Data Approaches

Aqib et al. [18] combined big data, in-memory and GPU computing along with deep learning in an endeavor to develop an algorithm that predicts incidents for traffic incident management. Alomari et al. [19] detected traffic events in Saudi Arabia by utilizing machine learning algorithms and data mining technologies. Their work used twitter data in Saudi dialect and resulted in detection of traffic events with times and location without any prior knowledge. Iktishaf [20] is a big data tool developed over Apache Spark with the aim of detecting traffic-related events. While Suma et al. [21] detected patio-temporal events located in London city using twitter data. They used machine learning algorithms with apache Spark and Tableau along with HPC to increase scalability and computational intelligence. They succeeded at detecting three major events around the city during the time of study. In another work, real data for London Metro and technologies like big data, deep learning, in-memory and GPU have been used to model rapid transit systems [22]. They aimed to aid in enhancing the spatiotemporal planning urban transport systems. As for analyzing traffic characteristic, several studies considered traffic flow [23], [24] and traffic occupancy [25] to develop smarter traffic management systems.

2.3 Computer Vision Approaches

Computer vision-based solutions were developed to address various aspects of transportation and traffic management. Jiang et al. [26] addressed multiple lane-detection in structured highways in their work. They proposed an estimate and detect scheme for both straight and curve lanes. They used huge transformation with a simplified perspective transformation for straight lanes, and a complete transformation for detecting curve lanes. Their approach was able to detect lanes even when abstracted by vehicles. Seenouvong et al. [27] aimed to detect and count vehicles in videos. They used background subtraction technique to locate foreground objects within a video sequence. Combining techniques like thresholding and adoptive morphology operations, they were able to detect moving vehicles. As for vehicles counting, they deployed virtual detection zones. Around the same purpose of detecting and counting, Waranusast et al. [28] proposed an atomic system for detecting motorcycles and counting their riders. The detection is done using the K-Nearest Neighbor (KNN) classifier while counting is done using projection profiling. Zaho and Han [29] propose a fuzzy detection method to detect the state of fatigue that may affect the performance of logistics drivers on the road. Nizar et al [30] developed a system to detect and track vehicles and pedestrians. They deployed a Histogram Oriented Gradient (HOG) in their feature extraction method combined with a Support Vector Machine (SVM) classifier for detection. They used Kanade- Lucas-Tomasi (KLT) algorithm for tracking. Aziz et al. [31] presented an implementation of a vehicle detection algorithm that may serve as a part of an autonomous car systems. They deployed feature intuition, color spaces, and HOG to process images. Due to static parameters within their algorithm, they managed to detect vehicles during the day with good accuracy. TAAWUN [32] is an approach designed to enable autonomous vehicles and connected vehicles to share visual information. The motivation of this work was to enhance environment perception. It is the first work to exploit deep learning decision fusion in autonomous environments. Alam et al. [33] provided an in depth study that aims to compare the performance of two different methods for object classification in driving environments. Gao et al. [34] considered road safety as a general goal. They developed a Multi-Object Tracking (MOT) method that mimic attention of skilled drivers for autonomous driving.

2.4 Deep learning Approaches

Given the breakthroughs in the field of deep learning algorithms, it is only logical to apply them on the persistent issues of transportations. AlexNet [35] was the first CNN to utilize GPU powers during training and winning the ImageNet Challenge

ILSVRS in 2012. In the years to follow, more deep learning networks appeared motivated by the design of AlexNet. VGGNet [36] is a convolutional network developed by Visual Geometry Group in the University of Oxford. Despite been a network with more layers than AlexNet, VGGNet achieved better accuracy with less parameters. GoogLeNet [37] is the winner of ILSVRS 2014. It was developed by Google and introduced the notion of Inception Blocks. Hence it is also known as the Inception Networks. The Residual Network [38] ResNet is develop by He et al. at Microsoft. As researchers attempted to add more layers to Neural Network design, they observed performance degradation. That is what developers of ResNet aimed to resolve. Such configuration motivated researchers to deploy its different variations in various computer vision problems including transportation. Lv et al. [39] aimed at detecting pedestrians using a Regional Proposal Network (RPN) after witnessing some advancement in pedestrian prediction by RPN. They combined VGGNet with ZFNet to extract pedestrians' features at different levels. Their proposed method proved to be superior to stat-of-the-art methods. Ren et al. [40] implemented a system to analyze vehicles' behaviors by extracting their trajectories. They used several deep learning detectors to detect vehicles including SSD-VGGNet. TrafficNet [41] is a classifier based on combining AlexNet and VGGNet and replacing the fully connected layers in both by SVM. TrafficNet attends to identify traffic congestion in various conditions. Whereas Zhou et al. [42] proposed an encoder-decoder architecture to understand road scenes. They based their encoder network on VGGNet.

2.5 Yolo Approaches for Detection

One of the most common methods for object detection is You Only Look Once (YOLO) [43]. The algorithm was introduced and outperformed other detection methods in speed and accuracy. Since then, it has been under continuous improvements to enhance its performance [44][45]. According to Bochkovskiy et al. [46] YOLOv4 achieved high performance compared with state-of-the-art object detection methods. Naturally, such performance encouraged researchers to exploit its potentials in transportation. Wang et al. [47] proposed an algorithm to detect abnormality in vehicles behavior such as stalled cars and cars speeding up or slowing down. They used YOLO algorithm for detection and Kalman filter for tracking. They tested their framework on videos from traffic cameras. The combination of YOLO and Kalman filter is applicable, despite some scenarios where farther contextual knowledge is needed to improve detection results. Zhao et al. [48] presented an abnormal event detection framework. They assumed that in case of an abnormal event cars would stop in the track video. Hence, such cars will be part of the background and can be detected by a background model using MOG2 algorithm. For detection, they used Faster R- CNN with some alteration on the ROI layer. While their MOT algorithm is based on CNN with emphasis on shape and position features. To farther enhance detection results, they applied a road mask model. Finally, the time of the anomaly occurrence is determined by a decision module. Their algorithm came in 7th place in the 2019 AI CITY CHALLENGE. Camacho et al. [49] aimed to design a detection method that can be part of a low cost traffic systems. They applied a cascade classifier combined with HAAR features for detecting vehicles. Kalman filter was their choice for tracking and counting. They used CNN as a classifier to determine the class for each vehicle. Chen et al. [50] developed a framework for detecting vehicles on highways. They used K-mean algorithm to prepare the data, feature fusion to address low-level and high-level features. They deployed VGG-16 network and replaced the last three fully connected layers with convolutional layers to improve detection speed while reducing parameters. The proposed method proved superior over Faster R-CNN and SSD tested on the same vehicle dataset (JSHD).
While Song et al. [51] considered small vehicles on highways in their proposed detection and counting system. They published a new high definition dataset containing annotations of small objects. They developed a segmentation method to extract and divide roads into remote and proximal areas. They used YOLOv3 as their detection method and the ORB algorithm as a feature extractor. They analyzed the trajectories of detected objects for counting purposes. Their proposed method provide good performance as can replace the traditional ways of counting vehicles without any new hardware equipment.
Chen et al. [52] aimed at detecting objects by generating 3D object proposals. Their proposed work utilized stereo imagery. They based the method on minimizing an energy function which encodes object size prior, object placement and some context depth information. Then, they used convolutional neural network to use appearance, context and depth information for object detection. The method predicted 3D bounding box coordinate and object pose. The approach proved to be superior to previously published object detection work on the KITTI benchmark.

3. Methodology and Design

Figure1 describes the overall workflow of our work. In this section, we describe our methodology and design. First, we discuss the process of collecting data in Section 3.1. Section 3.2 describes the detection module of our model. Section 3.3 describes the tracking module. Subsection 3.4 gives the performance metrics used to evaluate the work. 3.1 Dataset Collection & Preparation
This section discusses our datasets that we are using for detection and tracking. We collected video streams from the streets of Kingdom of Saudi Arabia (KSA). We used Xiaomi 70mai Smart Dash Cam to capture videos. Each video is one-minute long and 30 fps. We trained the model two times and produced two custom weights files. The first custom training was done using our Saudi Arabia data only. We extracted frames from the collected videos and manually labeled them using the LabelImg tool [60]. We used two types of vehicles' labels 'Car' and 'Truck'. Our first training dataset (KSA dataset) contained 400 labeled images. During training, we divided the dataset into 80% training set and 20% validation set.

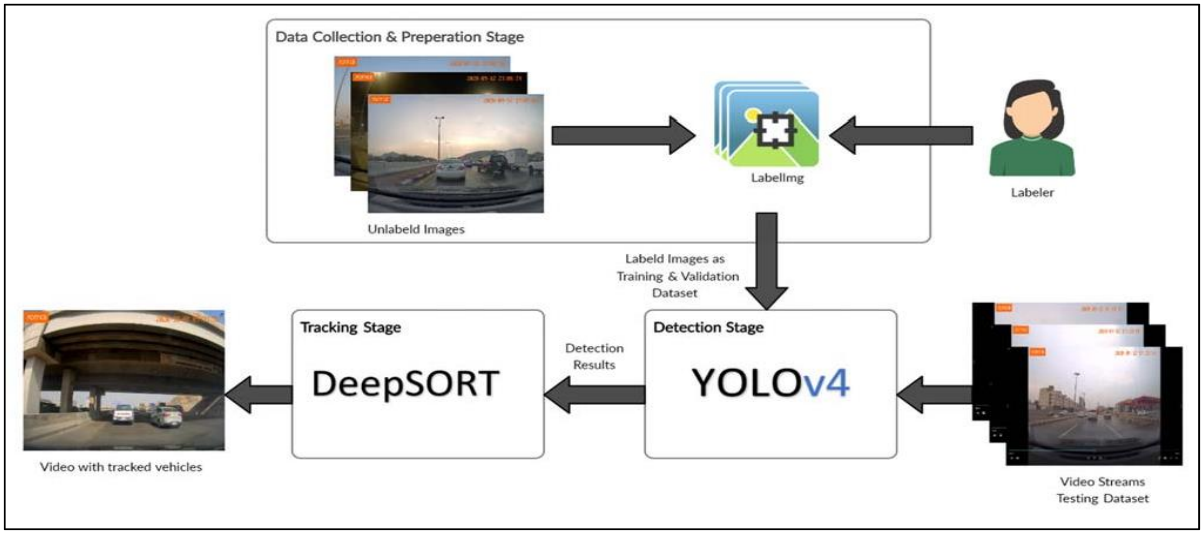


Figure 1: The Methodology Workflow

For the second custom training, we combined our KSA dataset with 400 images from the Berkeley DeepDrive dataset [9]. We call it the KSA_BDD dataset. We divided the dataset in the same proportions, 80% and 20%, for the training set and the validation set, respectively. We selected this dataset due to the resemblance of vehicles' shapes and models between the KSA dataset and BDD dataset. The KSA_BDD training dataset contained 800 labeled images and followed the same distribution for training and validation. As for the testing data, we choose five different videos to represent the diversity of driving conditions that a driver might face on the roads. They are as follow: City Daytime (CityD), City Night-time (CityN), Highway Daytime (HighD), Highway Night-time (HighN) and Rain during daytime (RainD). A snapshot of each of these videos is presented in Figure 2 to Figure 6.

Figure 7: Results of YOLOv4 detection

defined set of bounding boxes *B* is created. And for each bounding box the algorithm predicts four values: center, width and height of the box, the probability that a box contains an object, and the class of that object. The detection process end with outputting gives a bounding box surrounding the object and an object class see Fig. 7 for an example output.

3.3 Tracking

For the purpose of tracking, we are deploying DeepSORT, the enhanced version of the algorithm where the association metric is substituted by an informed metric integrating motion and appearance information using Convolutional Neural Network. The algorithm takes the detection outputs from the previous stage and run tracking for each detected object. In tracking by detection scheme, the accuracy of tracking is based on the quality of detection results. As mentioned earlier, we have used YOLOv4 for the purpose.

Detection

The detection in our model is performed using YOLOv4 detection algorithm. We choose YOLO because of its speed and accuracy especially with relatively larger objects. Due to the nature of our data, we believe that YOLO is a suitable choice.

Performance Metrics

For the purpose of evaluating the detection results, we have used precision that is a well-known metrics to evaluate the detection models. Precision is defined as follow.

Precision = $TP/(TP + FP)$ (1)

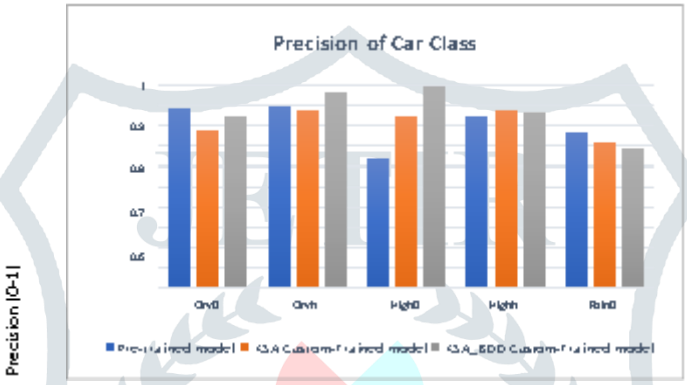
Where the terms TP and FP are defined as:

- True Positive (TP), which is the number of positive observations predicted by the model as positives.
- False Positive (FP), which is the number of negative observations predicted by the model as positives.

For evaluating the tracking results, we used Equation (2) below that measures the tracking success rate (TSR) . $tracked_objects$total is the total number of tracked objects and $ID_switches$total is the total number of ID Switches.

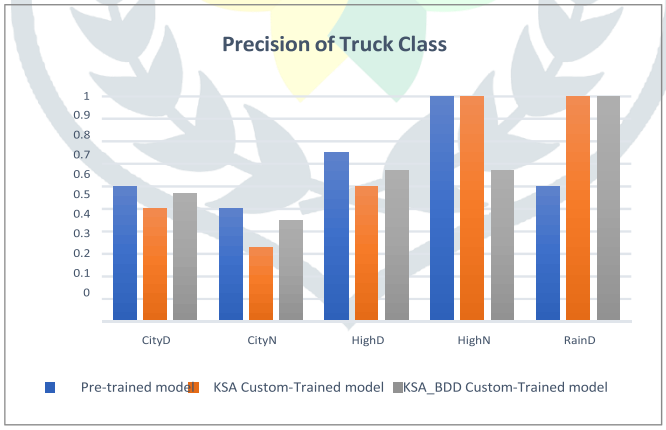
$TSR = tracked\_objectstotal – ID\_switches/ total\ tracked\_objectsto$ (2)

The YOLO family belongs to the one stage detectors category, which means that a target image is divided into a grid of $w\times h$. Then, for each segment of that grid, a Experiments and Analysis This section presents our experimental settings and the detection and tracking results. We provide details of our experimental settings for the training and testing phases in Section 4.1. In Section 4.2, we describe the three models -- Pre-Trained model, the KSA model and the KSA_BDD model – and present the detection results. Section 4.3 discusses the tracking results.



3.1 Experimental Environment

For training the custom models, we used Google Colab platform to harness GPU powers, CUDA and OpenCV. Then, we conducted the experiments on Windows 10 OS with NVIDIA GeForce GTX1060 using the produced weights files for each model. We experimented on our data using three variations of the YOLOv4 model. The pre- trained model available on [61], a custom trained model on KSA dataset, and a custom trained model on KSA_BDD dataset.



3.2 Detection Results

Pre-Trained Model: The pre-trained YOLOv4 model is obtained by training the model on the COCO dataset. After testing, we calculated precision for both classes Car and Truck under pre-trained model. The results are given in Fig. 8 and Fig. 9 and will be discussed later in this section

KSA Custom-Trained Model: The training of the KSA custom-trained model was conducted on capabilities needed for training the model. The overall mean average precision (mAP) of the model equals 90.34%. The model was able to detect cars and trucks according to the results given in Table

1. We tested the KSA Custom-Trained Model on the dataset described in Section 3.1. The results are presented in Fig. 8

Table 1: Mean Average Precision of KSA Custom-Trained model

Class	mAP	TP	FP
Car	93.62%	121	8
Truck	87.07%	22	4
Overall	90.34%	143	12

and Fig. 9 and will be discussed later in this section.

KSA_BDD Custom-Trained Model: After training the model on the SA_BDD dataset, we chose a suitable. weights file to custom the model. The overall mAP is 81.02 %. As for detecting cars and trucks, the model achieved the following numbers:

Table 2:Mean Average Precision of KSA_BDD Custom-Trained model

Class	mAP	TP	FP
Car	87.78%	439	78
Truck	74.25%	48	21
Overall	81.02%	487	99

CityD: The CityD video represents a city scene in the daytime of a four-lane street including the service lane. We considered both the vehicles in front (in the driving direction), as well as the vehicles coming from the opposite direction on all four lanes on the other side of the road. In CityD, while the pre-trained model is the model with most detections (TPs) among the three models, it registered most of the misclassification cases (FP) for trucks (it classified trucks as cars). Note that the higher Precision values for the Pre-Trained model in Fig. 8 and Fig. 9 are due to higher number of TPs. We can safely assume that the COCO dataset does not represent truck models in KSA hence a higher misclassification cases for trucks. Regarding custom models, we can conclude that the undetected numbers of vehicles, are due to the fact that both KSA dataset and KSA_BDD dataset are small (have 400 and 800 images compared to a total 120,000 in COCO used for training and validation for the Pre-Trained model). To improve the detection for custom-trained models, we need to increase the size of both the datasets, KSA and KSA_BDD. Fig. 10

scenarios for the Car class are due to white SUVs being identified in the faraway lane as trucks.

This might be due to the resemblance between them and the faraway small freight trucks usually used for logistics services like Aramex and FedEx. Note that the pre-trained model among the three models had the most TPs but also the most FPs. The custom-trained models, on the other hand, detected less (a relatively smaller number of TPs) also with a fewer misclassification cases (FPs). See Fig. 12 for a snapshot of the detection. shows a snapshot of some detected cars from the city scene.

CityN: The CityN video is a city traffic video on a two-lane street where the parked cars are also considered because it is a narrow road and with no service lane. In CityN, the Pre- trained model detected the least of the trucks in the video (see Fig. 9) that points out to the lack of training data regarding certain truck models. Yet, the model was able to detect vehicles from farther distance unlike both custom models that detected the same vehicles from a closer distance. Due to the nature of the street, we noticed that even when from a side view trucks were detected by both custom models. Fig. 8 shows that the precision of the Car class was similar with small differences for all three models. Fig. 11 presents a snapshot of the nighttime city scene with some detected vehicles.

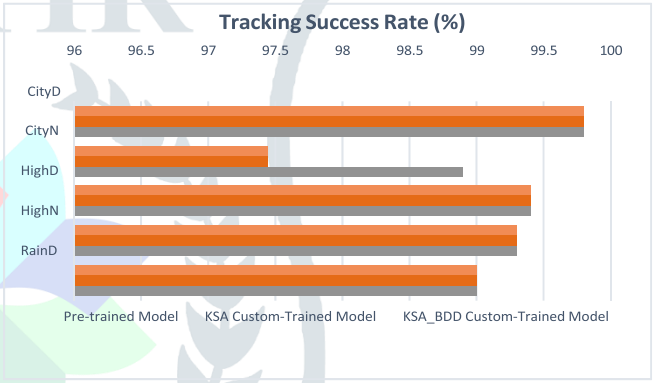
HighD: Fig. 8 shows that the precision results for the Pre- Trained model are relatively poorer than the two custom- trained models. The misclassification in the HighD

HighN: We noticed in HighN traffic conditions that with crowded lanes, cars on the third lane went undetected by the custom models as seen in Fig. 13. It could be due to the lighting conditions which would be further investigated in the future by adding more nighttime highway data to both the training datasets, KSA and KSA_BDD datasets. Note the results for HighN in Fig. 8 and Fig. 9. As before, while both custom models detected less than the pre-trained model, they had less misclassifications especially for truck class. This is due to the fact that certain pickup trucks have different models in Saudi Arabia. Once the model was custom trained to recognize such trucks, it is able to detect them. This in turns, supports the need to create a dataset for vehicles in Saudi Arabia to address traffic analysis properly. For both HighD and HighN scenarios, we excluded the opposite lanes from the test as well as vehicles that are parked on the right side of the road, the pre-trained model doesn't detect most of them while both custom models did not detect any. A snapshot of the detection in HighN is given in Fig. 13. RainD: In Rain, we considered all the lanes on the driving side as well as the opposite side of the road. Note the results for RainD in Fig. 8 and Fig. 9. We observed misclassification of vehicles in the case of the same model of pickup trucks. The rain also affected the detection process when the wind shield has water drops. The detection improves when the viper clears the water drops. Fig. 14 presents a snapshot of the detection results during the rain.

Detection Results - Summary: As noted in the discussion above, the precision results of the custom trained models are comparable to the Pre-Trained model. A large number of misclassification cases suggest the need of further experimentations. A large number of misclassification cases suggest that our models are not over-fitted. This leaves us with the consideration of adding more data for training and testing. Meaning that, in order to further improve the performance of our custom models, we should consider training on bigger custom datasets.

3.3 Tracking Results

This section presents the tracking results obtained using DeepSORT for the three models – Pre-Trained, KSA Model, and KSA_BDD Model and five scenarios, CityD, CityN, HighD, HighN, and RainD. The results are calculated using Equation (2) defined in Section 3.4. Fig.15 plots these results.



The results show a mix of performance from the three models across the five traffic scenarios. None of the models provides a dominating performance. Looking closely at Fig. 15, for the CityD scenario, the KSA Custom_Trained Model provides the best results followed by first the KSA_BDD Custom_Trained Model and then the Pre_Trained Model. This we believe is due to the Pre_Trained Model being trained on a dataset that did not describe the nature of the vehicles in Saudi Arabia. For CityN, the Pre-Trained Model provides the best tracking success rate. This could be due to the fact that both custom datasets contained less night time images of the class Car compared to the COCO dataset while most of the vehicles in CityN are of the class Car. In Highway scenario, the pre-trained model performed better during the daytime (HighD) and worst during the nighttime (HighN). This might be due to the nature of highway traffic when combined with lighting conditions. For RainD, we note that both the Custom_Trained models performed better benefiting from custom training on the data describing the nature of traffic in Saudi Arabia. This could also be due to the specific means of collecting the data via Dash Cam, which is different from the COCO dataset.

5. Conclusion

Deep learning is revolutionizing all spheres of our life, smart cities and societies, Industry 4.0, and much more. Transportation is continuing to cause unbelievable damages including 1.25 million deaths and trillions of dollars annually. This paper has presented a study on the use of YOLOv4 for vehicle detection and DeepSORT for tracking the detected vehicles on roads. None of the earlier works have applied these models to road traffic in KSA. We used three different variations of the deep learning models and compared their performance; a pre-trained model with the COCO dataset, and two custom-trained models with the Berkeley DeepDrive dataset, and our custom-developed dataset obtained by a Dash Cam installed onboard vehicle driven on KSA roads in five different traffic conditions. The five traffic scenarios included city traffic in day and night, highway traffic in day and night, and traffic in the rain. We used the Google Colab platform to harness GPU power, CUDA and OpenCV. The results have been evaluated using precision and tracking success rate and show a mix of performance for the pre-trained and custom-trained models. The pre-trained model was unable to deliver consistently good performance across all five scenarios both in terms of precision and tracking success rate. The results of the custom trained models are comparable to the Pre-Trained model. A large number of misclassification cases by all three models suggest the need of further experimentations. A large number of misclassification cases also suggest that our models are not over-fitted. This shows that there is a need to add more data for training and testing for all three models particularly the custom-trained models. An important finding of this work is that pre-trained models cannot work

in KSA environments without retraining due to the differences in the language, driving culture, driving environments, and vehicle models. Future work will look into building larger datasets for vehicle detection, tracking, and other problems in road transportation, and developing highly accurate deep learning models optimized for the environment.

REFERENCES

[1]N. Janbi, I. Katib, A. Albeshri, and R. Mehmood, "Distributed Artificial Intelligence-as-a-Service (DAIaaS) for Smarter IoE and 6G Environments," Sensors, vol. 20, no. 20, p. 5796, Oct. 2020.

[2] Battula, B., et al. "Prediction of vehicle safety system using internet of things." Journal of Green Engineering, 10 (4) (1798).

[3] T. Yigitcanlar, L. Butler, E. Windle, K. C. Desouza, R. Mehmood, and J. M. Corchado, "Can Building 'Artificially Intelligent Cities' Safeguard Humanity from Natural Disasters, Pandemics, and Other Catastrophes? An Urban Scholar's Perspective," Sensors, vol. 20, no. 10,pp. 2988, May 2020.

[4] Y.M.M. Babu, Optimized performance and utilization analysis of real-time multi spectral data/image categorization algorithms for computer vision applications, Turk. J.Comput. Math.Educ.(TURCOMAT), 12 (9) (2021), pp. 2212-2227

[5] P. A. H. Vardhini, V. K. R. Yasa and G. J. Raju, "Raspberry Pi Vehicle Gateway System with Image Processing based Authorization Detection using IoT," 2021 IEEE International Conference on Electronics, Computing and Communication Technologies (CONECCT), Bangalore, India, 2021, pp. 1-5, doi: 10.1109/CONECCT52877.2021.9622528.

[6] R. Mehmood, S. See, I. Katib, and I. Chlamtac, Eds., Smart Infrastructure and Applications: Foundations for Smarter Cities and Societies. EAI/Springer Innovations in Communication and Computing, Springer Nature Switzerland AG, pp. 692, 2020.

[7] R. Mehmood, B. Bhaduri, I. Katib, and I. Chlamtac, Eds., Smart Societies, Infrastructure, Technologies and Applications, vol. 224. Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering (LNICST), Springer, pp. 367, 2018.

[8] Ranjith, E., Parthiban, L., Latchoumi, T.P. et al. An Effective Content Based Image Retrieval System Using Deep Learning Based Inception Model. Wireless Pers Commun 133, 811–829 (2023). https://doi.org/10.1007/s11277-023-10792-8

[9] Tushara, D. Bindu, PA Harsha Vardhini, and N. Ranganadh. "Implementation of Compressed Image Using DWT on FPGA." International Journal of VLSI and Embedded Systems-IJVES: 1420-1423.

[10] H Vardhini and Raju G. Janardhana, "Design of Internet of Things Based Smart and Efficient Water Distribution System for Urban and Agriculture Areas", Journal of Computational and Theoretical Nanoscience, vol. 17, no. 9–10, pp. 4688-91, Jul 2020.

[11] Prakasam, V., P. Sandeep, and P. A. Harsha Vardhini. "Snappy and video st ream edge detection using labview." International Journal of Advanced Science and Technology 28.19 (2019): 197-203.

[12] T. Muhammed, R. Mehmood, A. Albeshri, and I. Katib, "UbeHealth: A personalized ubiquitous cloud and edge- enabled networked healthcare system for smart cities," IEEE Access, vol. 6, pp. 32258–32285, 2018.

[13] R. Mehmood, F. Alam, N. N. Albogami, I. Katib, A. Albeshri, and S. M. Altowaijri, "UTiLearn: A Personalised Ubiquitous Teaching and Learning System for Smart Societies," IEEE Access, vol. 5, pp. 2615–2635, 2017.

[14] J. Balsa-Barreiro, P. M. Valero-Mora, M. Menéndez, and R. Mehmood, "Extraction of Naturalistic Driving Patterns with Geographic Information Systems," Mob. Networks Appl., pp. 1–17, Oct. 2020.

[15] Vardhini, PA Harsha, and Y. MuraliMohanBabu. "FPGA based Energy-aware image compression and transmission with single board computers." Journal of green engineering 10.5 (2020).

[16] T.-Y. Lin et al., "Microsoft COCO: Common Objects in Context BT - Computer Vision – ECCV 2014," 2014, pp. 740–755.

[17] L. Bhaskar, A. Sahai, D. Sinha, G. Varshney, and T. Jain, "Intelligent traffic light controller using inductive loops for vehicle detection," Proc. 2015 1st Int. Conf. Next Gener. Comput. Technol. NGCT 2015, no. September, pp. 518–522, 2016.

[18] F. Yu et al., BDD100K: A Diverse Driving Dataset for Heterogeneous Multitask Learning. 2020.

[19] P. Khoenkaw and P. Pramokchon, "An implementation of automatic inductive-loop vehicle sensor using low-cost microcontroller," ECTI DAMT-NCON 2019 - 4th Int. Conf. Digit. Arts, Media Technol. 2nd ECTI North. Sect. Conf. Electr. Electron. Comput. Telecommun. Eng., pp. 331–334, 2019.

[20]S. Sheik Mohammed Ali, B. George, L. Vanajakshi, and J. Venkatraman, "A multiple inductive loop vehicle detection system for heterogeneous and lane-less traffic," IEEE Trans. Instrum. Meas., vol. 61, no. 5, pp. 1353– 1360, 2012.

[21] V Prakasam, PA Harsha Vardini , "Analysis and Applications of Continuous Wavelet Transform", pp. 192-195, Suraj Punj Journal For Multidisciplinary Research, Volume 9, Issue 4, 2019.

[22] S. T. Jeng and L. Chu, "Tracking Heavy Vehicles Based on Weigh-In-Motion and Inductive Loop Signature Technologies," IEEE Trans. Intell. Transp. Syst., vol. 16, no. 2, pp. 632–641, 2015.

[23] D Bindu Tushara and P.A. Harsha Vardhini, "Performance of Efficient Image Transmission Using Zigbee/I2C/Beagle Board Through FPGA", Innovations in Computer Science and Engineering. Lecture Notes in Networks and Systems, vol. 8, 2017, [online] Available: https://doi.org/10.1007/978-981-10-3818-1_27.

[24] Shivaprasad, S., et al. "Real time CNN based detection of face mask using mobilenetv2 to prevent Covid-19." Annals of the Romanian Society for Cell Biology 25.6 (2021): 12958-12969.