ROBUST AND SCALABLE REAL-TIME VEHICLE CLASSIFICATION AND TRACKING: A CASE STUDY OF THAILAND

B. Neupane[1], T. Horanont[2,*], P. Pattarapongsin[3], A. Thapa[2]

[1]Advanced Geospatial Technology Research Unit, Sirindhorn International Institute of Technology, Pathum Thani 12000, Thailand - geomat.bipul@siit.tu.ac.th,

[2]School of Information, Computer, and Communication Technology (ICT), Sirindhorn International Institute of Technology, Pathum Thani 12000, Thailand - teerayut@siit.tu.ac.th, m6322041135@g.siit.tu.ac.th

[3]Mappico, Bangkok 10170, Thailand - phakawat.p@mappico.co.th

Commission IV, WG IV/9

KEY WORDS: Vehicle Classification, Multi-vehicle Tracking, Intelligent Transport Systems, Spatial Information, YOLOv5

ABSTRACT:

An accurate detection, classification, and tracking of vehicles are highly important for intelligent transport systems (ITS) and road maintenance. In recent years, the deep learning (DL)-based approach is highly regarded for real-time vehicle classification from surveillance cameras. However, the practical implementation of such an approach is affected by the adverse lighting conditions and positioning of the cameras. In this research, we develop a DL-based method for near real-time multi-vehicle counting, classifying, and tracking on individual lanes of the road. First, we train a DL network of the You Only Look Once (YOLO) family on a custom dataset that we have curated. The dataset consists of nearly 30000 training samples to classify the vehicles into seven classes, which is more than in the existing benchmark datasets. Second, we fine-tune the trained model into another small dataset collected from the surveillance cameras that are used during the implementation process. Third, we connect the trained model to a tracking algorithm that we have developed to produce a per-lane report with the calculation of the speed and mobility of the vehicles. We test the robustness of the system on different faces of the vehicles and in adverse lighting conditions. The overall accuracy (OA) of classification ranges from 91% to 99% in four faces of vehicles (back, front, driver side, and passenger side). Similarly, in an experiment on adverse lighting conditions, OA of 93.7% and 99.6% is observed in a noisy and clear lighting conditions respectively. The implications of these results will assist in road maintenance with spatial information management and sensing for intelligent transport planning.

1. INTRODUCTION

An assessment of road conditions is necessary to determine any maintenance program (Radopoulou and Brilakis, 2016) for a large road network. In a road assessment, a network of surveillance cameras (Baran et al., 2014), which are used for security and safety, can be used for observing numerous vehicles passing through an area of road covered by the cameras. Continuously moving vehicles on a road, especially heavy vehicles, causes damage to the pavements of the roads (Liu, 2015). Detection of these heavy vehicles and vehicle classification can provide strong support to ITS and road maintenance (Maungmai and Nuthong, 2019). DL methods are widely used for vehicle classification in recent years. However, the practical implementation, robustness, and scalability are the major concerns of such systems when it comes to the use of an existing network of surveillance cameras as the data input. In this study, we make use of an existing network of such cameras facilitated by the department of the rural road (DRR) of Thailand and develop a robust and scalable DL-based method to detect, classify, and track the vehicles in near real-time.

DL networks require training on a large dataset and suffer from a domain-shift problem due to the difference in train and test data. These problems often hinder the scalability of the DL methods. Implementing such method for real-time vehicle classification and tracking requires a powerful yet lightweight DL network and a multi-object tracking (MOT) algorithm. We use the current state-of-the-art (SOTA) DL network called YOLOv5 which belongs to the convolutional neural networks (CNNs) of the YOLO family. The network provides the optimal trade-off between the accuracy and speed of classification. To train the YOLOv5, we curate a dataset of nearly 30000 samples for seven classes of vehicles, some which are unavailable in the existing datasets. The trained model is further fine-tuned on a smaller dataset prepared from the cameras that are used during the implementation process. This fine-tuning increases the scalability of the DL model and minimizes the domain-shift problem by leveraging the knowledge from a large generic dataset and a smaller dataset collected from a practical environment. The fine-tuned model is then connected to a multi-vehicle tracking algorithm that we have engineered to calculate and report the count of vehicles, per car unit (PCU), speed of individual vehicles, per-lane average speed over a time interval, and the mobility of the vehicles to record the next destination the vehicle is headed to. The system is tested on different faces of vehicles and adverse lighting conditions for robustness-check and practical implementation.

The rest of the paper is structured as: Section 2. provides the existing literature on the research domain; Section 3. presents the overall method; Section 4. demonstrates the experiments and validation; the paper concludes with future remarks in Section 5..

2. BACKGROUND

2.1 Vehicle detection and classification

Object detection is a fundamental problem in computer vision applications. It deals with locating and classifying the objects in an image. Several methods for vehicle detection and

*Corresponding author.

classification have been developed over the years due to advancements in machine learning and computer vision. Support Vector Machines (SVM) is used as a machine learning method for the classification of vehicles based on colour and type (Chen et al., 2009). The method suffers when the reflection of the surface and strong sunlight changes the colour of the vehicles. Histogram Orientation Gradients (HOG) feature-based method is less affected by the change in illumination to distinguish the appearance and shape of objects (Cao et al., 2011). This feature is also used to train an SVM to detect vehicles on videos collected from low-altitude airborne cameras and is also comparable to Haar-like features (Negri et al., 2008). *OpenCV development kits* is another method to detect moving vehicles on a camera for traffic count measurement (Uke and Thool, 2013). Other algorithms include Scale Invariant Feature Transform (SIFT), Speeded Up Robust Features (SURF) methods, and 3D models (Ferryman et al., 1995). Despite faster detection, these methods generate a high number of false negatives and false positives, especially during adverse lighting conditions, and fail during the task of vehicle classification.

In recent years, DL has produced a breakthrough performance in object detection and classification using "hidden layers" of convolutions. (Jung et al., 2017) propose the localization and classification of vehicles in traffic surveillance using ResNet50 with added dropping CNN (DropCNN), and fine-tuning the model to improve the classification. (Zhuo et al., 2017) use a GoogleNet (Szegedy et al., 2015) CNN to classify vehicles in large-scale traffic surveillance. They pre-train the GoogleNet on the ILSVRC-2012 dataset and fine-tune it with another dataset to improve accuracy. Even though these CNNs show good performance during object detection in their experiments, they require heavy computation and are sensitive to scale changes (Cai et al., 2016). The advancement of CNNs has developed sophisticated yet faster and more accurate CNNs such as YOLO, which we elaborate in the next section.

2.2 YOLO (You Only Look Once)

YOLO (Redmon et al., 2016) takes the task of object detection as a regression problem in a single neural network. The method has obtained the SOTA in object detection with an overwhelming performance. Since its introduction, five generations of YOLO have been produced by different authors: YOLOV2 (Redmon and Farhadi, 2017), YOLOv3 (Redmon and Farhadi, 2018), YOLOv4 (Bochkovskiy et al., 2020), and YOLOv5 (Jocher et al., 2020). The recent generations such as YOLOv4 and YOLOv5 have shown higher performance in terms of accuracy and speed among the YOLO family. YOLOv5 further has several versions with depths ranging from the smallest to the largest model size (eg. YOLOv5s, YOLOv5m, YOLOv5l, and YOLOv5x), providing different trade-offs between speed and accuracy of detection. The different versions are compared in terms of speed and accuracy in Figure 1. For our purpose, we use YOLOv5l after a comparison among the YOLOv5 family.

To perform vehicle detection and classification, (Sang et al., 2018) have improved the YOLOv2 by adding a k-means++ algorithm to cluster the bounding boxes of vehicles in the training dataset, removing some repeated convolution layers to improve feature extraction, and introducing normalization to improve the loss due to varying scales of vehicle bounding boxes. They use BIT-Vehicle (Sang et al., 2018) and CompCars (Yang et al., 2015) datasets. (Du et al., 2019) propose real-time detection of vehicles and traffic lights using YOLOv3 to improve detection in small objects with balanced speed and precision. They use the traffic light dataset (V-TLD) to train the YOLOv3. (Song et al., 2019) use YOLOv3 to detect and classify the vehicles and ORB algorithm (Rublee et al., 2011) to obtain driving directions. (Mahto et al., 2020) use fine-tuned Y OLOV4 for vehicle detection using the UA-DETRAC dataset. With this literature as background, next, we present our method design.

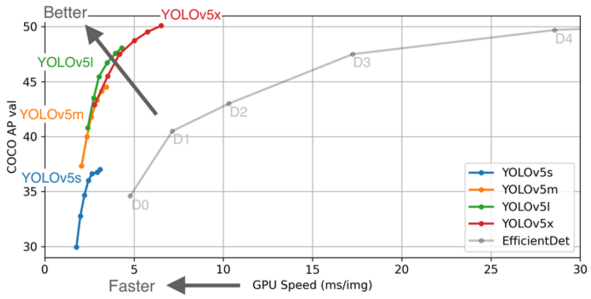


Figure 1: Comparison of various sized models of the YOLOv5 family in terms of speed and accuracy of detection (adapted from (Jocher et al., 2020)). The more the plots tend to the top-left corner, the better the performance of the model.

3. METHOD

3.1 Data Preparation

Training a DL model requires a large training dataset. The existing dataset like COCO (Lin et al., 2014), PASCAL VOC (Everingham et al., 2010), KITTI (Geiger et al., 2013), BIT-Vehicle and CompCars dataset do not cover the seven classes of vehicles – car, bus, taxi, bike, pickup, truck, and trailer – that the DRR needed to be classify. Therefore, we create a dataset called *Thai-Vehicle-Classification-Dataset* t hat w e have introduced in our previous study in (Neupane et al., 2022). The dataset is curated from 6.3 terabytes of surveillance videos, taken from 23 different cameras for 3 continuous days starting from 25-27 June, 2020. Training samples are manually annotated from carefully selected image frames of the videos to generate varying samples on adverse lighting conditions and different faces of vehicles. An open-source program called *labelimg* (Tzutalin, 2015) is used to annotate the vehicles into seven classes. To increase the samples for the class of *bus*, which is found to be less abundant in our dataset, we add 4431 samples of buses from a dataset of Hangzhou, China (Song et al., 2019). The total number of samples for each class is shown in Table 3.1. From all samples, the ratio of the train-validation samples is divided to be 90%-10%.

Vehicle Type	Annotated Samples	Added from (Song et al., 2019)	Total Sample
Car	10478	0	10478
Bus	540	4431	4891
taxi	1605	0	1605
Bike	2572	0	2572
Pickup	6056	0	6056
Truck	2656	0	2656
Trailer	1179	0	1179

Table 1: The total number of samples collected to train the YOLOv5 network.

3.2 Training YOLOv5 and fine-tuning

The YOLOv5 network that we use follows a similar architecture as YOLOv4 and consists of a backbone of the Cross Stage Partial (CSP) network (Huang et al., 2017), (Wang et al., 2020), a neck of the Path Aggregation Network (PANet) (Liu et al., 2018)

with Spatial Pyramid Pooling (SPP) block (He et al., 2015) and a head of YOLOv3. The YOLOv5 integrates an automated anchor box selection process into the network, making it learn the best anchor boxes for the training dataset. This assembly of the backbone, neck, head, and anchor box selection process speeds up the space-to-depth conversion process, alleviates the gradient descent problem, strengthens the feature propagation, minimizes the network parameters, and generalizes the objects of different sizes and scales with increased precision. The network architecture is shown in Figure 2.

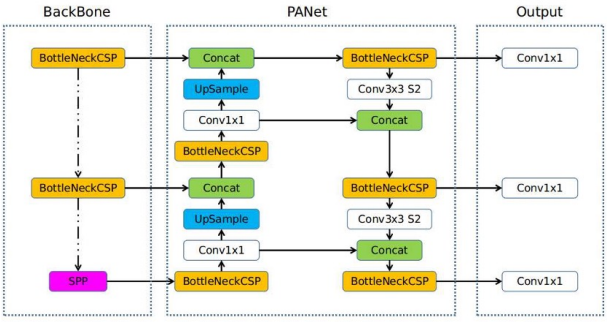


Figure 2: The network architecture of YOLOv5 (adapted from (Jocher et al., 2020)).

We train the large version of the YOLOv5 called YOLOv5large (abbr. YOLOv5l), which is wrapped in Pytorch framework. YOLOv5l has a more depth in the network layers than other smaller versions. To increase the accuracy of YOLOv5l, it is first trained on the Thai-Vehicle-Classification-Dataset without initializing weight. The trained model is then fine-tuned on a smaller dataset of approximately 5 times smaller samples (6612 samples) generated from the cameras that are used in the experimental settings. The small dataset contains 2585, 274, 323, 562, 2042, 666, and 160 samples for class of car, bus, taxi, bike, pickup, truck, and trailer respectively. The fine-tuning is based on transfer learning to leverage the knowledge from the larger dataset to the model fine-tuned on a smaller dataset. During the fine-tuning, the weights are initialized from the model that is previously trained on the larger dataset. Data augmentation is done during both training and fine-tuning, to increase the variability in the training dataset. The augmentation steps include random scaling, translation, a horizontal flip of 180 degrees, and hue–saturation–value (HSV) is randomly changed. The input images are resized to 640x640 pixels. Four anchor sizes are learned and derived using k-means clustering algorithm from the training dataset. The initial and final learning rate is set as 0.01 and 0.2, with a momentum of 0.937 and weight decay of 0.0005. An Adam optimizer is used to optimize the model. The model is trained in the batch size of 8 for 2000 epochs and fine-tuned for 300 epochs on a computer with 128GB of RAM, Intel(R) Xeon(R) Silver 4210 CPU, and two NVIDIA GeForce 2080 GPUs of 11GB memory each. The model is trained in approximately 3.4 days.

An improved Intersection of Union (IoU) loss called generalized intersection over union (GIoU) loss (Rezatofighi et al., 2019) is used to evaluate the YOLOv5 network, which is denoted by Eqn. 1.

$$L_{GIoU}(w) = 1 - IoU + \frac{|\,C(A \cup B)\,|}{|\,C\,|} \qquad (1)$$

where A and B are the bounding boxes of the ground truth and prediction respectively, C is the smallest rectangle circumscribed between A and B, and IoU is the intersection of A and B. The major improvement of GIoU compared to IoU is that it defines A minimum closed area C such that the borders of A and B are included in C. GIoU then calculates the area of A and B not included in C proportionate to the total area of C.

3.3 Tracking algorithm

The next step after training the YOLOv5l model is to use the final trained model to track individual vehicle classes on a real-time video stream. For this, we develop a multi-vehicle tracking algorithm that takes the predicted class and bounding box from any DL model and performs several tasks to track vehicles, count the number of vehicles of each class, and calculate the speed in each lane polygon of the road. The overall method is shown in Figure 3. This method shows superior performance in terms of computational power, speed, and matching costs.

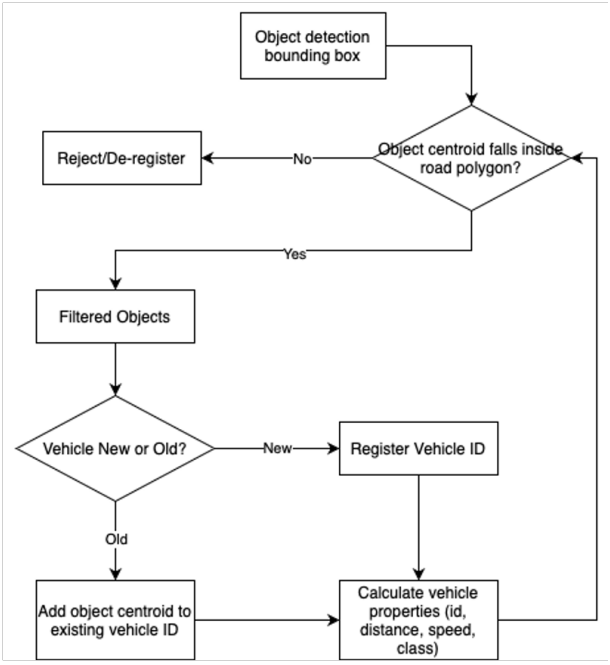


Figure 3: Multi-vehicle Tracking Algorithm for a lane-based count and speed detection of vehicles.

To explain the overall tracking method, the centroid of the detected object's bounding box is first cross-checked if it falls inside the lane polygon drawn over the video frame. These polygons are pre-defined by the video surveillance team over an image frame coming from the video stream from the surveillance camera. If the centroid does not fall into the polygon, it is "de-registered" meaning that the vehicle class and bounding box are stored in the database but do not pass through the tracking process. If the centroid falls inside the defined road polygon, then these "filtered objects" go through the registration process. If the vehicle is new, then it is first registered and passed to the "vehicle property calculation process". If it is an older vehicle but being tracked, the updated bounding box and centroid are added to the vehicle ID's array and passed to the "vehicle property calculation process". In the "vehicle property calculation process", the *distance* between the current position of the centroid of the object and the point in the line of the road polygon through which the vehicle passed is calculated. This distance is used to calculate the speed of the vehicle using the general formula of $speed = distance/time$. The *time* variable is the difference in time between when the object is first recorded within the road polygon and the current time recorded. Finally, the vehicle ID, speed, and class are saved into the database.

4. EXPERIMENT AND RESULTS

In this section, we describe the performance of the trained model and the accuracy of the overall method. First, the training and validation accuracy of the YOLOv5l model is presented. Then several video streams collected from different camera positions and with adverse lighting conditions are used to validate the overall method. Three accuracy metrics are used for the experiments:

$$Recall(R) = \frac{TP}{TP + FN} \tag{2}$$

$$Precision(P) = \frac{TP}{TP + FP} \tag{3}$$

$$OverallAccuracy(OA) = \frac{R + P}{2} \tag{4}$$

where *TP*, *FP*, and *FN* are the number of true positives, false positives, and false negatives, respectively.

4.1 Training YOLOv5l

As mentioned before, the total image dataset is divided into training and validation images in a ratio of 90:10. The selected YOLOv5l model is trained until the GIoU loss on the training dataset decreased to 0.027, and the best model is chosen based on GIoU such that it is minimum on the validation dataset. The minimum GIoU of 0.025 is obtained in the validation dataset on epoch 1442, approximately after 2.6 days of training. Even though the network is trained until the 2000 epochs, the best model at the 1442 epoch is saved and used as a trained model for our method. The change in GIoU throughout training is shown in Figure 4. The validation of count, detection, and classification are shown in the next section.

4.2 Effects of different faces of vehicles

The overall method of vehicle detection, classification, and tracking is validated on an experimental setup of four cameras each facing the four different sides of the vehicles on the road as shown in Figure 5. The four cameras Cam 1, Cam 2, Cam 3, and Cam 4 face on the back, driver-side, passenger-side, and front of the vehicles respectively. The cameras are streamed from the highway of Ratchapruek, Pathum Thani, Thailand to the computing server using 4G internet broadband and Real-Time Streaming Protocol (RTSP). A ratio of frame per second (FPS) is used to measure the rate of the real-time stream. If the computing server can process the video stream with the same FPS that arrives from the RTSP, real-time is achieved. A total of 10 frames are provided per second via RTSP to the server. Our method could process up to 38 FPS of image frames with some loss in image frames due to broadband connection, therefore achieving near real-time speed.

A total of approximately 35000 image frames are used for validation with 7500, 9060, 8800, and 9575 image frames from Cam 1, Cam 2, Cam 3, and Cam 4 respectively. The recall (R), precision (P), and overall accuracy (OA) of count and classification of individual classes of vehicles are shown in Table 4.3. OA for the class *car* is the highest (97%) and *truck* is the lowest (91%). The smaller vehicles such as cars, taxis, bikes, and pickups are classified with higher OA of 96% to 97%. However, the larger vehicles such as buses, trucks, and trailers are classified

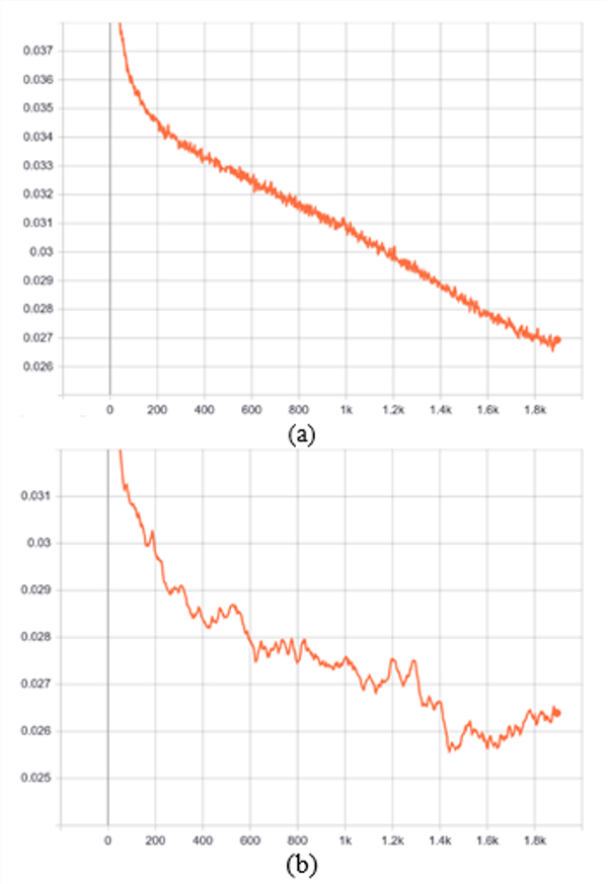


Figure 4: A plot to show GIoU loss during training of the YOLOv5l model on training and validation samples of Thai-Vehicle-Classification-Dataset. (a) On training samples. (b) On validation samples.

with relatively lower OA from 90% to 94%. The reasons are associated with the placement height of the camera, fewer training samples for truck and trailer, and fewer test samples for validation. The camera is placed 5.5m high from the road surface, which allows better classification of vehicles that are contained in small bounding boxes during prediction. However, for the larger vehicles, the bounding box of the prediction is larger when the vehicles get closer to the camera, leading to false classifications in different image frames. On average, OA of 94% is obtained for the classification of vehicles from the experimental setup of four cameras. Some vehicles like buses and trailers were in low abundance in the experimental setup, which will be increased in future works.

4.3 Effects of adverse lighting conditions

The performance of the method is highly affected by the adverse lighting conditions. Four videos are clipped from the surveillance cameras that are different from the experimental setup in the previous section to show the limitation of our method in the adverse settings. The image frame samples of the four videos (Vid 1, ..., Vid 4) are shown in Figure 6. Vid 1 contains noise from the dust on the camera. Vid 2 and Vid 3 are taken from a newly installed camera at 3 PM and 5 PM respectively of a day. The intensity of light at 3 PM is more than at 5 PM. Vid 4 is taken in the nighttime at 3 AM. The four experimental videos provide a practical environment to test the robustness of the method in adverse lighting conditions. The results of the count (detection) of vehicles in the four videos are shown in Table 4.3.

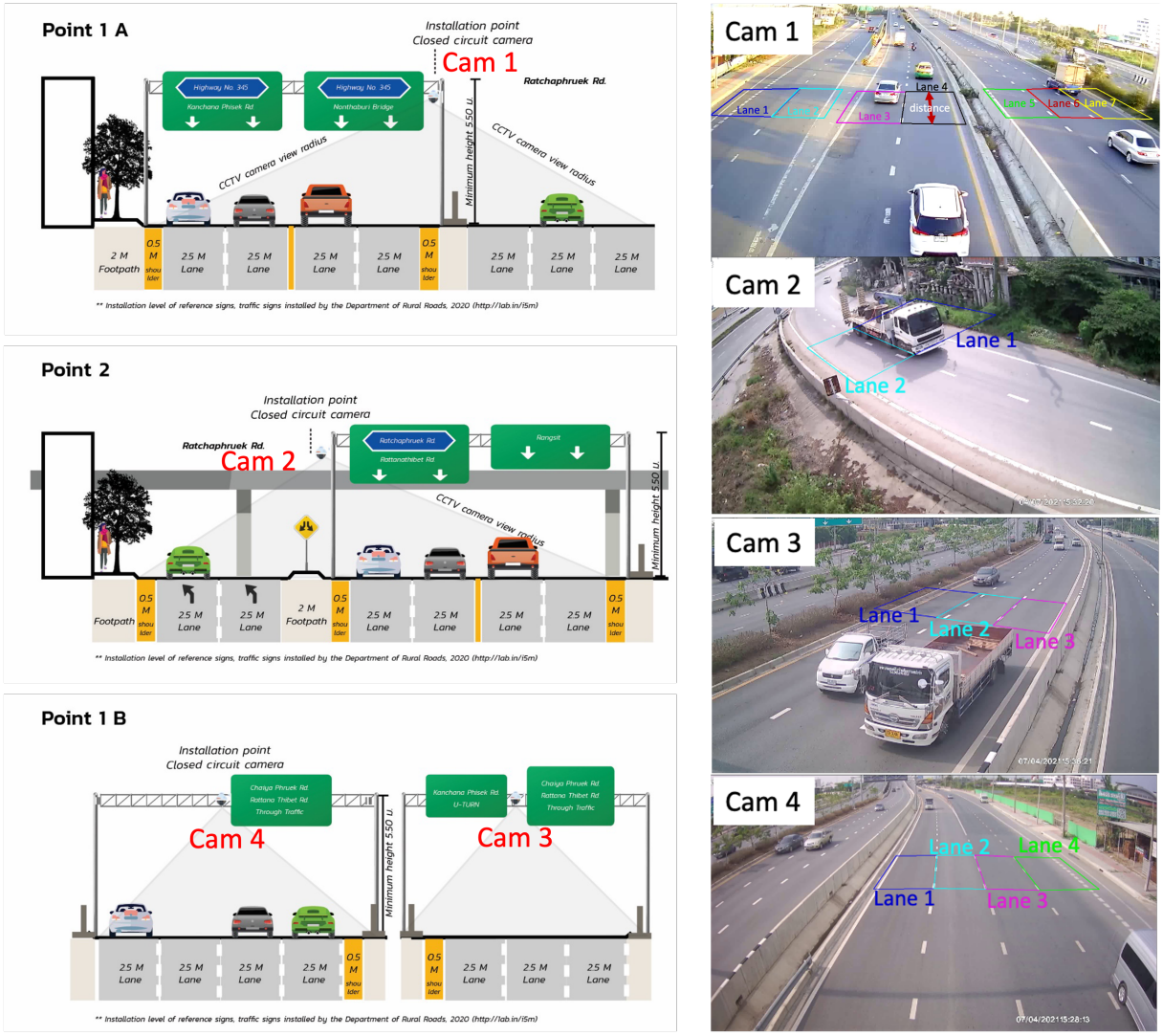
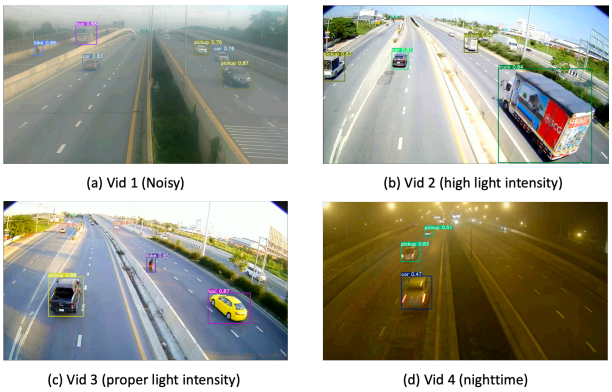


Figure 5: Experimental setup of four cameras (Cam 1, ..., Cam 4) for the validation of vehicle count and classification.



(a) Vid 1 (Noisy)

(b) Vid 2 (high light intensity)

(c) Vid 3 (proper light intensity)

(d) Vid 4 (nighttime)

Figure 6: Sample of vehicle classification in the four experimental videos to test the robustness of the method in adverse lighting conditions.

As seen in Table 4.3, with the proper light intensity, the performance of vehicle detection and the count is most effective in the daytime at 5 PM and the least effective during the nighttime. There are several factors associated with the inefficacy of the method at nighttime, such as the black and white (B/W) image stream provided by our experimental cameras during the nighttime, the light from the headlight of the vehicles, and the switching between B/W and color during a sudden change in light intensity from vehicle's headlight. A maximum OA of 99.6% is obtained at 5 PM.

For the validation of classification, experiment is carried-out on Vid 1 (noisy video), Vid 2 (video with high light intensity), and Vid 3 (video with proper light intensity at 5 PM). The result is shown in Table 4.3, where the classification is better in the video with better light intensity. We do not include the performance at night-time, because the classification is highly affected by several factors mentioned before. We use the YOLOv5l only for detection and without classification during the nighttime.

5. CONCLUSION

In this study of vehicle detection, classification using deep learning, we create a DL-based method to detect, classify, and track the vehicles on the roads in near real-time. The method reports the count, per car unit, classification, speed of individual vehicles, per-lane average speed over different time intervals, and the mobility of the vehicles that record the next destination of the vehicle. The system ensures practical use of the method for the maintenance and monitoring of the highways of Thailand. A specific focus is on developing a scalable and robust system for practical implementation. The scalability of the method

Classes	Cam 1 (Back)			Cam 2 (Driver side)			Cam 3 (Passenger side)			Cam 4 (Front)			Avg. OA
	P	R	OA	P	R	OA	P	R	OA	P	R	OA	
car	0.99	0.97	0.98	1.00	0.94	0.97	0.98	0.94	0.96	0.95	0.97	0.96	0.97
bus	1.00	0.80	0.90	-	-	-	-	-	-	-	-	-	0.90
taxi	0.75	1.00	0.88	-	-	-	1.00	1.00	1.00	1.00	1.00	1.00	0.96
bike	0.95	1.00	0.98	1.00	1.00	1.00	1.00	0.83	0.92	1.00	1.00	1.00	0.97
pickup	0.98	0.96	0.97	0.92	1.00	0.96	0.96	0.98	0.97	0.94	0.90	0.92	0.96
truck	0.60	0.75	0.68	1.00	1.00	1.00	0.92	1.00	0.96	1.00	1.00	1.00	0.91
trailer	1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.67	0.83	-	-	-	0.94
Avg.	0.90	0.93	0.91	0.98	0.99	0.99	0.98	0.90	0.94	0.98	0.98	0.98	0.94

Table 2: Vehicle count and classification on the experimental setup of four cameras facing on different sides of the vehicles as shown in Figure 5.

Video	Frames	Manual Count	Method Count	OA
Vid 1 (noisy)	90000	3617	3390	93.7
Vid 2 (3PM)	7500	552	545	98.7
Vid 3 (5PM)	7500	514	512	99.6
Vid 4 (3AM)	7500	14	12	85.7

Table 3: Validation of detection of vehicles (count) in adverse lighting conditions.

Video	Frames	Correct	False	R	P	OA
Vid 1	90000	2851	766	78.8	84.1	81.4
Vid 2	15000	598	39	91.6	93.9	92.7
Vid 3	15000	470	30	94.0	94.0	94.0

Table 4: Validation of classification in terms of quality of the video stream.

is maximized by training the DL model on a custom large dataset that we develop and further fine-tuning on a small dataset from the test cameras. To test the robustness, the experiments, therefore, investigate the performance of the method on four faces of the vehicles and four adverse lighting conditions. The experiments report 91%, 95%, 99%, and 94% OA for the back, front, driver side, and passenger side of the vehicles. The smaller vehicles are classified with higher accuracy and larger vehicles are classified with relatively lower accuracy due to the placement height of the cameras and lower number of training and validation samples for larger vehicles. In an experiment of adverse lighting conditions, an OA of 93.7%, 98.7%, and 99.6% is observed while counting the vehicles on a noisy camera, camera with high light intensity at 3 PM, and camera with normal lighting condition at 5 PM respectively. Similarly, the OA of classification on noisy, 3 PM, and 5 PM video streams is 81.4%, 92.7%, and 94% respectively. The classification in the noisy video is highly affected, and at nighttime, a reasonable accuracy could not be reported. To conclude, a small yet powerful CNN such as YOLOv5 that efficiently makes a trade-off between the accuracy and speed of detection can be used for real-world implementation of multi-object classification in near real-time. The domain-shift problem can be minimized by fine-tuning the model on a small dataset, avoiding the need to frequently train a CNN on a large training dataset. Furthermore, we recommend the researchers in the domain of vehicle classification train their model on the proposed *Thai-Vehicle-Classification-Dataset* to improve upon the research problems. In future work, the effects of loads of the vehicles on the road conditions will be studied to monitor and map the maintenance works, and further support the spatial information management and sensing for intelligent transport planning.

ACKNOWLEDGEMENTS

This work is partially supported by the Center of Excellence in Intelligent Informatics, Speech and Language Technology and Service Innovation (CILS), Thammasat University Research Fund under TSRI, Contract No. TUFF19/2564 and TUFF24/2565 for the project of "AI Ready City Networking in RUN", based on the RUN Digital Cluster collaboration scheme.

REFERENCES

Baran, R., Ruść, T. and Rychlik, M., 2014. A smart camera for traffic surveillance. In: *International Conference on Multimedia Communications, Services and Security*, Springer, pp. 1–15.

Bochkovskiy, A., Wang, C.-Y. and Liao, H.-Y. M., 2020. Yolov4: Optimal speed and accuracy of object detection. *arXiv preprint arXiv:2004.10934*.

Cai, Z., Fan, Q., Feris, R. S. and Vasconcelos, N., 2016. A unified multi-scale deep convolutional neural network for fast object detection. In: *European conference on computer vision*, Springer, pp. 354–370.

Cao, X., Wu, C., Yan, P. and Li, X., 2011. Linear svm classification using boosting hog features for vehicle detection in low-altitude airborne videos. In: *2011 18th IEEE International Conference on Image Processing*, IEEE, pp. 2421–2424.

Chen, Z., Pears, N., Freeman, M. and Austin, J., 2009. Road vehicle classification using support vector machines. In: *2009 IEEE International Conference on Intelligent Computing and Intelligent Systems*, Vol. 4, IEEE, pp. 214–218.

Du, L., Chen, W., Fu, S., Kong, H., Li, C. and Pei, Z., 2019. Real-time detection of vehicle and traffic light for intelligent and connected vehicles based on yolov3 network. In: *2019 5th International Conference on Transportation Information and Safety (ICTIS)*, IEEE, pp. 388–392.

Everingham, M., Van Gool, L., Williams, C. K., Winn, J. and Zisserman, A., 2010. The pascal visual object classes (voc) challenge. *International journal of computer vision* 88(2), pp. 303–338.

Ferryman, J. M., Worrall, A. D., Sullivan, G. D., Baker, K. D. et al., 1995. A generic deformable model for vehicle recognition. In: *BMVC*, Vol. 1, Citeseer, p. 2.

Geiger, A., Lenz, P., Stiller, C. and Urtasun, R., 2013. Vision meets robotics: The kitti dataset. *The International Journal of Robotics Research* 32(11), pp. 1231–1237.

He, K., Zhang, X., Ren, S. and Sun, J., 2015. Spatial pyramid pooling in deep convolutional networks for visual recognition. *IEEE transactions on pattern analysis and machine intelligence* 37(9), pp. 1904–1916.

Huang, G., Liu, Z., Van Der Maaten, L. and Weinberger, K. Q., 2017. Densely connected convolutional networks. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4700–4708.

Jocher, G., Nishimura, K., Mineeva, T. and Vilariño, R., 2020. yolov5. *Code repository https://github. com/ultralytics/yolov5*.

Jung, H., Choi, M.-K., Jung, J., Lee, J.-H., Kwon, S. and Young Jung, W., 2017. Resnet-based vehicle classification and localization in traffic surveillance systems. In: *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pp. 61–67.

Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P. and Zitnick, C. L., 2014. Microsoft coco: Common objects in context. In: *European conference on computer vision*, Springer, pp. 740–755.

Liu, J., 2015. Research on the damage of heavy vehicles to the pavement. In: *2015 International Conference on Management, Education, Information and Control*, Atlantis Press, pp. 649–655.

Liu, S., Qi, L., Qin, H., Shi, J. and Jia, J., 2018. Path aggregation network for instance segmentation. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 8759–8768.

Mahto, P., Garg, P., Seth, P. and Panda, J., 2020. Refining yolov4 for vehicle detection. *International Journal of Advanced Research in Engineering and Technology (IJARET)*.

Maungmai, W. and Nuthong, C., 2019. Vehicle classification with deep learning. In: *2019 IEEE 4th International Conference on Computer and Communication Systems (ICCCS)*, IEEE, pp. 294–298.

Negri, P., Clady, X., Hanif, S. M. and Prevost, L., 2008. A cascade of boosted generative and discriminative classifiers for vehicle detection. *EURASIP Journal on Advances in Signal Processing* 2008, pp. 1–12.

Neupane, B., Horanont, T. and Aryal, J., 2022. Real-time vehicle classification and tracking using a transfer learning-improved deep learning network. *Sensors* 22(10), pp. 3813.

Radopoulou, S. C. and Brilakis, I., 2016. Improving road asset condition monitoring. *Transportation Research Procedia* 14, pp. 3004–3012.

Redmon, J. and Farhadi, A., 2017. Yolo9000: better, faster, stronger. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 7263–7271.

Redmon, J. and Farhadi, A., 2018. Yolov3: An incremental improvement. *arXiv preprint arXiv:1804.02767*.

Redmon, J., Divvala, S., Girshick, R. and Farhadi, A., 2016. You only look once: Unified, real-time object detection. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 779–788.

Rezatofighi, H., Tsoi, N., Gwak, J., Sadeghian, A., Reid, I. and Savarese, S., 2019. Generalized intersection over union: A metric and a loss for bounding box regression. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 658–666.

Rublee, E., Rabaud, V., Konolige, K. and Bradski, G., 2011. Orb: An efficient alternative to sift or surf. In: *2011 International conference on computer vision*, Ieee, pp. 2564–2571.

Sang, J., Wu, Z., Guo, P., Hu, H., Xiang, H., Zhang, Q. and Cai, B., 2018. An improved yolov2 for vehicle detection. *Sensors* 18(12), pp. 4272.

Song, H., Liang, H., Li, H., Dai, Z. and Yun, X., 2019. Vision-based vehicle detection and counting system using deep learning in highway scenes. *European Transport Research Review* 11(1), pp. 1–16.

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V. and Rabinovich, A., 2015. Going deeper with convolutions. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1–9.

Tzutalin, D., 2015. Labelimg. *GitHub Repository*.

Uke, N. and Thool, R., 2013. Moving vehicle detection for measuring traffic count using opencv. *Journal of Automation and Control Engineering*.

Wang, C.-Y., Liao, H.-Y. M., Wu, Y.-H., Chen, P.-Y., Hsieh, J.-W. and Yeh, I.-H., 2020. Cspnet: A new backbone that can enhance learning capability of cnn. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*, pp. 390–391.

Yang, L., Luo, P., Change Loy, C. and Tang, X., 2015. A large-scale car dataset for fine-grained categorization and verification. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3973–3981.

Zhuo, L., Jiang, L., Zhu, Z., Li, J., Zhang, J. and Long, H., 2017. Vehicle classification for large-scale traffic surveillance videos using convolutional neural networks. *Machine Vision and Applications* 28(7), pp. 793–802.