Accredited by National Journal Accreditation (ARJUNA) Managed by
Ministry of Research, Technology, and Higher Education, Republic Indonesia with Second Grade (Peringkat 2, Sinta 2)
since year 2017 to 2021 according to the decree No. 10/E/KPT/2019.

Published online on the journal's webpage: **http://jurnal.iaii.or.id**

RESTI JOURNAL
(System Engineering and Information Technology)
Vol. 4 No. 2 (2020) 462 - 468 ISSN Electronic Media: 2580-0760

A Simple Vehicle Counting System Using Deep Learning with YOLOv3 Model

Muhammad Fachrie
Informatics Department, Faculty of Electrical and Information Technology, Universitas Teknologi Yogyakarta
muhammad.fachrie@staff.uty.ac.id

Abstract

Deep Learning is a popular Machine Learning algorithm that is widely used in many areas in current daily life. Its robust performance and ready-to-use frameworks and architectures enables many people to develop various Deep Learning-based software or systems to support human tasks and activities. Traffic monitoring is one area that utilizes Deep Learning for several purposes. By using cameras installed in some spots on the roads, many tasks such as vehicle counting, vehicle identification, traffic violation monitoring, vehicle speed monitoring, etc. can be realized. In this paper, we discuss a Deep Learning implementation to create a vehicle counting system without having to track the vehicles movements. To enhance the system performance and to reduce time in deploying Deep Learning architecture, hence pretrained model of YOLOv3 is used in this research due to its good performance and moderate computational time in object detection. This research aims to create a simple vehicle counting system to help human in classify and counting the vehicles that cross the street. The counting is based on four types of vehicle, i.e. car, motorcycle, bus, and truck, while previous research counts the car only. As the result, our proposed system capable to count the vehicles crossing the road based on video captured by camera with the highest accuracy of 97.72%.

Keywords: deep learning, yolov3, object detection, vehicle counting, traffic monitoring

1. Introduction

Deep Learning (DL) outperforms the conventional Machine Learning (ML) algorithms in many tasks, especially in Computer Vision (CV). ML plays important role in CV due to its capability to learn the pattern of objects or images and classify the object that is taken by camera. Previously, a CV system needs preprocessing and feature extraction step before it can detect, classify, or recognize objects within the image using ML algorithm [1]–[3]. Different objects or cases needs different techniques in preprocessing and feature extraction. Hence it makes the single model of conventional CV limited to detect or recognize certain objects only. While DL with its large and deep networks, automatically preprocess and extract the image features within its networks then classify the image class, even more it can detect the location of every single objects inside the image. Nevertheless, DL requires high specifications of machine and large amount of data to train the networks and optimize its performance [1], [4].

Traffic monitoring system is one area that implemented CV technology for some tasks, e.g. intelligent traffic light system [5], vehicle counting system [6]–[9], vehicle speed monitoring [8], parking lot monitoring [10]–[12], and traffic violation monitoring [13]. Every task mentioned before is started by detecting the position of each vehicle, i.e. car. Hence, object detection algorithm has crucial role in this part. Traditional Machine Learning approach needs preprocessing approach to complete this task, e.g. image gray scaling, image binarization, and background subtraction [6], [7], [14], or sometimes using edge detection [5]. Of course, this approach has limitations, e.g. when the vehicle's shadow exists in the image, the detection can be less precise. The inaccuracy of detection also occurs when some changes happen to the surface of the road, e.g. road repair, road damage, or any obstacles on the road, because those can disturb the image subtraction process. While Deep Learning (DL) approach gives more flexible performance without having to preprocess the image and extract the feature using several methods, even though it is computationally expensive, and it needs large amount of data to train the networks. Furthermore, there are now exist better DL architectures that has been trained with

Accepted by editor: 07-04-2020 | Final Revision: 25-05-2020 | Online Publication : 20-06-2020

millions of data, hence the development of CV system become easier.

In this work, we developed a simple vehicle counting system using Deep Learning algorithm. Pretrained YOLOv3 is used as the DL architecture that is well known with its good accuracy in object detection and its moderate computation compared to other DL architectures [15]–[17]. Moreover, YOLOv3 has been used in several vehicle detection systems as in [15], [18]–[21]. This work is motivated by previous research that was mostly tested using videos in the highways that are only passed by cars, bus, or truck, and there are no motorcycles. Besides, any buses or trucks are simply considered as 'car' without classifying them into more detail classes as 'bus' or 'truck' when counting. While some traffic monitoring system may need more detail information about the type of the vehicles whether it is a car, truck, bus, or motorcycle. The previous studies as in [6]–[9] were also mostly tested in good traffic condition with good driving manner, so that the counting gives accurate result. Hence, in this work we focus on developing a system that count the number of vehicles crossing the road where the counting is based on the type of the vehicle itself, i.e. car, bus, truck, and motorcycle based on Deep Learning algorithm with YOLOv3 architecture.

2. Research Method

This research was conducted in several phases, starting with some literature reviews to explore the result from previous related studies in order to find what problems that should be solved in the current research, also to decide what methods should be implemented. The system is designed once the problem and the methods are clearly decided. As mentioned in Introduction section, YOLOv3 is used as the algorithm to detect the vehicles that cross the road.

To test the system, a set of data was collected in form of videos recorded in Full HD resolution (1080p). The performance of the system is measured by its accuracy in counting the vehicles. It is compared to the real number of vehicles counted by human.

2.1. System Architecture

The vehicle counting system which was built in this work has two main modules, i.e. Object Detection Module and Counting Module as given in Figure 1. The first module reads every single frame from the video and doing vehicle detection using YOLOv3 algorithm. This module results the location of every detected vehicles, i.e. the bounding box coordinates. Then, the second module counts the number of vehicles that crossing the road based on the coordinates or location of the vehicles. So, the result of object detection module plays significant role in this system, because once the vehicle is not detected, then it will not be counted.

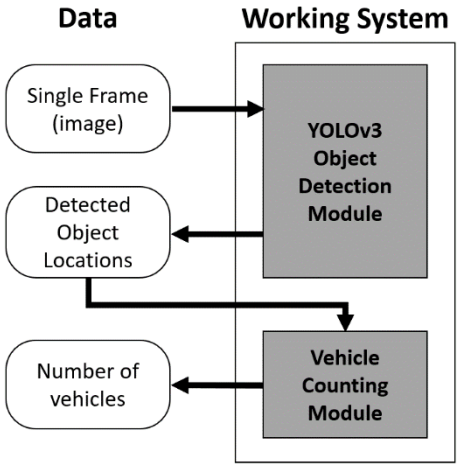


Figure 1. General architecture of the vehicle counting system.

The system is developed and tested on machine with mobile version GPU from Nvidia, i.e. Nvidia Geforce MX150 that has 384 cuda cores with 1.5GHz clock speed and 4 GB of VRAM. Although this engine is not as fast as the GTX version from Nvidia, but it is quite enough to run the Deep Learning-based system. From our observation, it can detect the objects for about 0.2 – 0.3 seconds for a single frame.

2.2. Data Acquisition

In order to test the system's performance, several videos in Full HD resolution (1080p) were successfully recorded using 8 megapixels smartphone camera in 30 fps. The video was taken manually from the top of pedestrian bridge in one of big city in Indonesia. The distance between the surface of the road to the top of the bridge is approximately 5 meters high as illustrated in Figure 2.

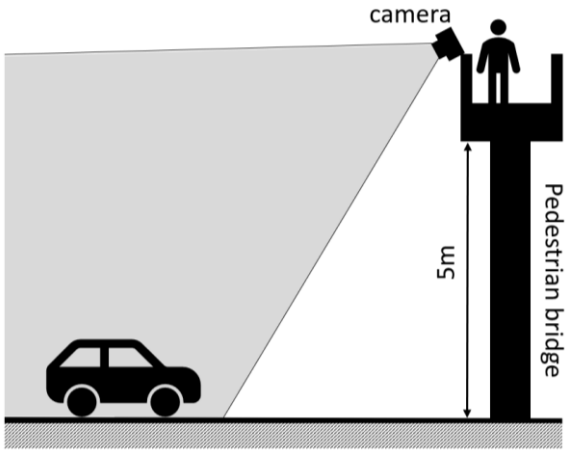
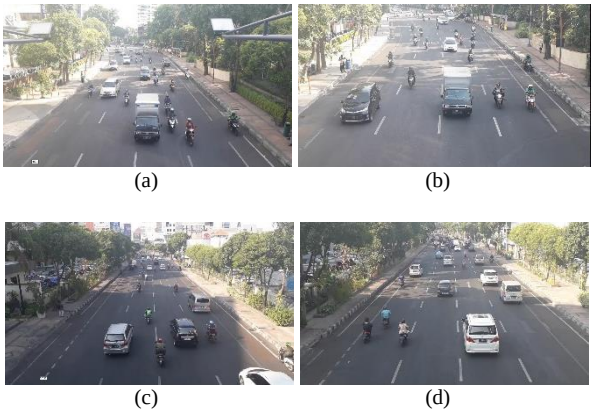


Figure 2. Illustration of data acquisition process

The video was recorded in fine weather about 03:00 pm. They are four scenarios in taking the video, i.e. frontside recording with 1x zoom, frontside recording with 2x zoom, backside recording with 1x zoom and backside recording with 2x zoom. Therefore, there are four different videos which is three minutes long for each.

The road is one-way direction with four different lanes. Various vehicles are passing by the road, e.g. cars, trucks, buses, motorcycles, bicycles, etc. Figure 3 shows the sample frames of each recording scenario.



(a) (b)

(c) (d)

Figure 3. Screenshots of videos from each recording scenario: (a) frontside 1x zoom, (b) frontside 2x zoom, (c) backside 1x zoom, and (d) backside 2x zoom.

2.3. YOLOv3

Deep Learning (DL) with its Convolutional Neural Networks (CNN) architecture is well-known to have very good performance in Computer Vision (CV), especially in object detection and classification. There are several CNN-based architectures that is used in CV for object detection and classification, e.g. Regional-based CNN (R-CNN) [22], Fast R-CNN [23], Faster R-CNN [24], Region-based Fully CNN (R-FCN) [25], YOLO [26], Single Shot Detector (SSD) [27], YOLO9000 [28], YOLOv2 [28], Mask R-CNN [29], and YOLOv3 [17]. Among these object detections techniques, YOLOv3 is considered as the most suitable model to be used in this experiment due to its good accuracy and real time speed of computation.

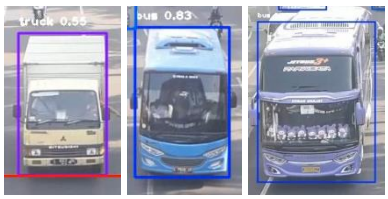
YOLOv3 is a Deep Learning model with Convolutional Neural Networks (CNN) architecture, which is an improvement from the previous version, i.e. YOLOv2 - where YOLOv2 itself is the improvement from YOLO. Based on [17], YOLOv3 has 53 convolutional layers as described in Figure 4, hence the network itself is named Darknet-53. With this architecture, beside the improvement of the detection accuracy, it also optimizes the GPU utilization and makes this network more efficient in computation [17]. YOLO is also invariant to the size of the input image, so that it makes the implementation of YOLO is easier and more practice.

	Type	Filters	Size	Output
	Convolutional	32	3 × 3	256 × 256
	Convolutional	64	3 × 3 / 2	128 × 128
1×	Convolutional	32	1 × 1	
	Convolutional	64	3 × 3	
	Residual			128 × 128
	Convolutional	128	3 × 3 / 2	64 × 64
2×	Convolutional	64	1 × 1	
	Convolutional	128	3 × 3	
	Residual			64 × 64
	Convolutional	256	3 × 3 / 2	32 × 32
8×	Convolutional	128	1 × 1	
	Convolutional	256	3 × 3	
	Residual			32 × 32
	Convolutional	512	3 × 3 / 2	16 × 16
8×	Convolutional	256	1 × 1	
	Convolutional	512	3 × 3	
	Residual			16 × 16
	Convolutional	1024	3 × 3 / 2	8 × 8
4×	Convolutional	512	1 × 1	
	Convolutional	1024	3 × 3	
	Residual			8 × 8
	Avgpool		Global	
	Connected		1000	
	Softmax			

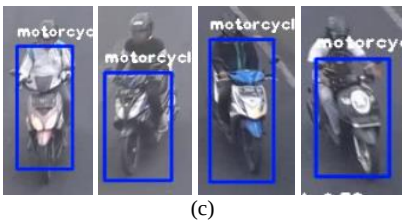
Figure 4. Architecture of Darknet-53 [17].

In this work, we used pretrained YOLOv3 model which has been trained using Ms. COCO dataset that makes this model can detects and classifies 80 different objects. But we just use YOLOv3 to detect three types of vehicles, i.e. car, bus, truck, and motorcycle. Some detected vehicles are presented in Figure 5.

(a)



(b)



(c)

Figure 5. Samples of detected vehicles: (a) car, (b) truck and bus, and (c) motorcycle.

2.4. Non-Tracking Vehicle Counting

Different with previous works that used object tracking to count the number of vehicles, we propose other simple strategy to count the vehicles without having to track its movement from frame to frame. As described in Figure 6, our method just simply evaluates the distance between the vehicle's centroid to the border line. If the distance is less or equal to threshold value which is defined before, hence it is counted as one vehicle. In this experiment, the threshold value is decided as 1.5% from the resolution of the video after several observations.

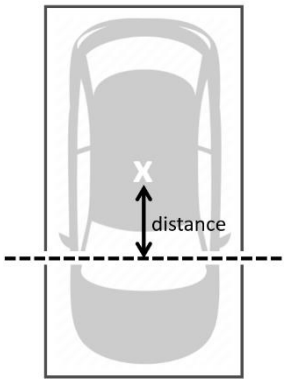


Figure 6. The vehicle is counted based on its centroid distance to the border line.

But this simple strategy has weakness if in several frames, the same vehicle has more than one close positions to the border line, then the system will count it as two or even three vehicles. Therefore, we applied additional strategy as shown in Figure 7 by analyzing the three consecutive frames and evaluating the position of the same vehicles to the border line. The same vehicles are identified by measuring their centroid to the previous frame (the same vehicles in different frame must have the closest distance among the other vehicles). This strategy can decrease the counting error. We also observed that the minimum distance of the same vehicles is 4% from the video resolution.

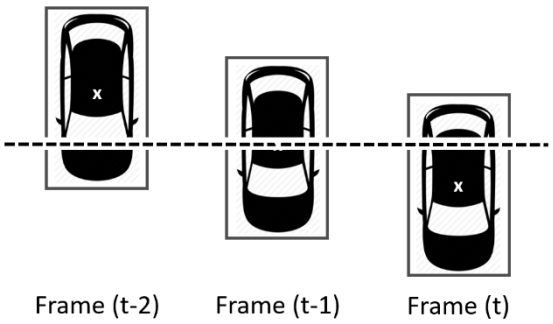


Figure 7. Illustration of non-tracking vehicle counting system.

2.5. Performance Measurement

The performance of the system is measured by comparing the difference between real number of vehicles and number of counted vehicles by the system. The percentage of accuracy is calculated using equation (1).

$$Accuracy = \left(1 - \frac{|RC - SC|}{RC}\right) \times 100\% \quad (1)$$

where *RC* is real number of vehicles counted by human and *SC* is number of vehicles counted by the system.

3. Result and Discussion

The vehicle counting system developed in this work is tested using four different videos as described in previous section. Originally, the videos are in 1080p of resolution with 30 fps. But, to explore the performance of the system, we used two different scenarios in the testing phase. Since we used a pretrained YOLOv3, so there is no training phase in this work instead.

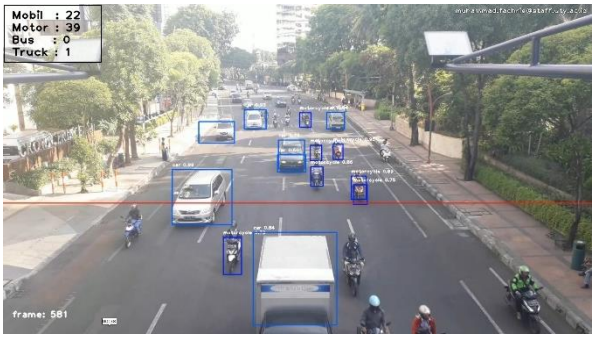


Figure 8. Sample of detected frame from the testing scenario.

In the first testing scenario, we ran the system using all four videos with 1080p Full HD resolution and 30 fps. This totally has about 5400 frames. The sample of the detected frame is shown in Figure 8.

Table 1. Result of the first testing scenario (1080p and 30fps)

Video	Vehicle	Counting		Counting Error	Overall Accuracy
		Real	System		
Video 1 (frontside 1x zoom)	Car	149	150	+1	96.96%
	Motor	244	246	+2	
	Bus	1	2	+1	
	Truck	1	9	+8	
Video 2 (frontside 2x zoom)	Car	128	114	-14	87.09%
	Motor	232	219	-13	
	Bus	3	4	+1	
	Truck	1	20	+19	
Video 3 (backside 1x zoom)	Car	122	114	-8	90.24%
	Motor	173	170	-3	
	Bus	1	3	+2	
	Truck	1	17	+16	
Video 4 (backside 2x zoom)	Car	99	96	-3	94.52%
	Motor	247	246	-1	
	Bus	0	2	+2	
	Truck	1	14	+13	
				Average Accuracy	92.20%

In the first testing scenario as described in Table 1, the system achieved the highest accuracy in the first video that is recorded from frontside with 1x zoom with 96.96% of accuracy. Type 'motorcycle' and 'car' has the most accurate counting with only 2.14% and 5.3% of error in average respectively among all videos, while type 'truck' got the worst accuracy followed by the type 'bus'.

The second scenario was run using 15 fps videos with the same resolution. As described in Table 2, the counting accuracy decreases 5% up to 14% for each video. But still, the first video got the highest accuracy with 91.14%. Type 'car' got a relatively constant average error for both testing scenarios, while 'motorcycle' got more error up to 11% than the previous one in the second scenario. 'Truck' rather better in the second scenario but still the worst among other vehicles.

Table 2. Result of the second testing scenario (1080p and 15fps)

Video	Vehicle	Counting		Counting Error	Overall Accuracy
		Real	System		
Video 1 (frontside 1x zoom)	Car	149	148	-1	91.14%
	Motor	244	218	-6	
	Bus	1	2	+1	
	Truck	1	8	+7	
Video 2 (frontside 2x zoom)	Car	128	116	-12	85.71%
	Motor	232	212	-20	
	Bus	3	5	+2	
	Truck	1	19	+18	
Video 3 (backside 1x zoom)	Car	122	113	-9	76.09%
	Motor	173	128	-45	
	Bus	1	3	+2	
	Truck	1	16	+15	
Video 4 (backside 2x zoom)	Car	99	94	-5	84.15%
	Motor	247	209	-38	
	Bus	0	2	+2	
	Truck	1	11	+10	
Average Accuracy					84.27%

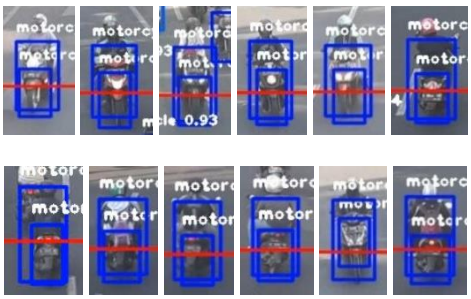
Based on these two scenarios, the video from frontside with 1x zoom is the most suitable to be used in the proposed system, even though other videos also give good results. Higher frame rate gives better performance because there is more information processed by the system, while decrease of fps ignores some or even most of information.

'Car' has the most stable counting accuracy among the others, even in lower fps video. This because the object of car is well-recognized by YOLOv3 model, so that every 'car' object is always detected in every frame. YOLOv3 is also capable to detect small objects of car that are located far from the camera as can be seen in Figure 8. After some observations, we realized that there is a drawback occurs, i.e. some cars are detected as two different types of vehicles at the same time as shown in Figure 9. It happens to the car whose shape is similar to 'truck' or 'bus'. Therefore, a single vehicle could be double counted as two different types of vehicle. This explains why 'truck' always get overcounted in both scenarios.

Figure 9. Sample of vehicle that is detected as two different types of vehicle (marked by two bounding boxes) at the same time while crossing the border line.

Type 'motorcycle' is also well-detected by YOLOv3, even though some motorcycles are not detected even when the location is close to the camera. But fortunately, this miss-detection does not decrease the counting accuracy because some other motorcycles are also double counted due to double detection on the same object as in Figure 10. The double detection on motorcycle is caused by YOLOv3 recognized two different kinds of motorcycle, i.e. a motorcycle with its driver and a motorcycle itself.

(a)



(b)

Figure 10. Sample of motorcycle objects: (a) undetected motorcycle when crossing border line, (b) double detected motorcycle when crossing the border line.

Table 3. Result of the first and the second testing scenario after improvement

Video	Vehicle	Real Count.	Counting (30 fps)		Accuracy (30 fps)		Counting (15 fps)		Accuracy (15 fps)	
			Prev.	New	Prev.	New	Prev.	New	Prev.	New
Video 1 (frontside 1x zoom)	Car	149	150	150	96.96%	**97.72%**	148	148	91.14%	**91.90%**
	Motor	244	246	246			218	218		
	Bus	1	2	2			2	2		
	Truck	1	9	6			8	5		
Video 2 (frontside 2x zoom)	Car	128	114	114	87.09%	**89.83%**	116	116	85.71%	**88.46%**
	Motor	232	219	219			212	212		
	Bus	3	4	4			5	5		
	Truck	1	20	10			19	9		
Video 3 (backside 1x zoom)	Car	122	114	113	90.24%	**91.25%**	113	112	76.09%	**77.44%**
	Motor	173	170	170			128	128		
	Bus	1	3	2			3	2		
	Truck	1	17	14			16	12		
Video 4 (backside 2x zoom)	Car	99	96	96	94.52%	**97.41%**	94	94	84.15%	**86.17%**
	Motor	247	246	246			209	209		
	Bus	0	2	2			2	2		
	Truck	1	14	4			11	4		

Since the counting accuracy of type 'truck' is the worst, then we tried to improve the system by adding some lines of code to solve the double detection on car objects by ignoring the 'truck' or 'bus' label when it is detected together with 'car' in the same object. This improvement gives better result to both testing scenario as can be seen in Table 3. However, 'truck' is still rather overcounted due to misclassification of vehicles. Some objects of 'car' are miss-classified as 'truck' or 'bus' by YOLOv3. But overall, the proposed system can perform well to count the vehicles on the road.

4. Conclusion

A vehicle counting system has been developed using YOLOv3 without tracking the vehicle movements. The counting is simply executed by evaluating the distance between the vehicle's centroid to the border line. It successfully achieved the highest accuracy of 97.72% when using frontside-1x zoom video. YOLOv3 plays significant role in detecting the vehicles since the counting is object to the detected object only. Object 'car' has the highest counting accuracy followed by 'motorcycle' and 'bus', while 'truck' is the worst. Frame rate of the video also give impact to the performance since it represents the integrity of information that is processed by the system. All in all, this work has been successfully completed with good result. In the future, any improvement should be made to get a better system.

References

[1] N. K. Chauhan and K. Singh, "A Review on Conventional Machine Learning vs Deep Learning," *2018 Int. Conf. Comput. Power Commun. Technol.*, pp. 347–352, 2018.

[2] N. O. Mahony *et al.*, "Deep Learning vs . Traditional Computer Vision," no. Cv.

[3] M. F. Rachmadi, M. C. Valdés-hernández, M. Leonora, F. Agan, and T. Komura, "Deep Learning vs . Conventional Machine Learning : Pilot Study of WMH Segmentation in Brain MRI with Absence or Mild Vascular Pathology †," pp. 1–19.

[4] J. Hagerty, R. J. Stanley, and W. V Stoecker, "Medical Image Processing in the Age of Deep Learning Is There Still Room for Conventional Medical Image Processing Techniques ?," no. Visigrapp, pp. 306–311, 2017.

[5] J. T. G. Nodado, M. A. P. Abugan, A. C. Aralar, and H. C. P. Morales, "Intelligent Traffic Light System Using Computer Vision with Android Monitoring and Control," *TENCON 2018 - 2018 IEEE Reg. 10 Conf.*, no. October, pp. 2461–2466, 2018.

[6] A. J. Kun and Z. Vámossy, "Traffic Monitoring with Computer Vision," in *7th Int'l Symposium on Applied Machine Intelligence and Informatics*, 2009, pp. 131–134.

[7] Z. Iftikhar, P. Dissanayake, and P. Vial, "Computer Vision Based Traffic Monitoring System for Multi-track Freeways," in *2014 Int'l. Conf. on Intelligent Computing*, 2014, pp. 339–349.

[8] Krishna, M. Poddar, M. K. Giridhar, and A. S. Prabhu, "Automated Traffic Monitoring System Using Computer Vision," in *2016 Int'l. Conf. n ICT in Business Industry & Governance*, 2016.

[9] S. Alghyaline, N. K. T. El-Omari, R. M. Al-Khatib, and H. Y. Al-Kharbshh, "RT-VC: An Efficient Real Time Vehicle Counting Approach," *J. Theor. Appl. Inf. Technol.*, vol. 97, no. 7, pp. 2062–2075, 2019.

[10] T. Paula, C. Florina, R. Brad, L. Br, and M. Greavu, "An Image Feature-Based Method for Parking Lot Occupancy," *Futur. Internet*, vol. 11, no. 169, pp. 1–17, 2019.

[11] T. Fabian, "A Vision-Based Algorithm for Parking Lot Utilization Evaluation Using Conditional Random Fields," in *2013 Int'l Symposium on Visual Computing*, 2013, pp. 222–233.

[12] B. Y. Cai, R. Alvarez, M. Sit, F. Duarte, and C. Ratti, "Deep Learning Based Video System for Accurate and Real-Time Parking Measurement," *IEEE Internet Things J.*, vol. 6, no. 5, pp. 7693–7701, 2019.

[13] W. Wu, O. Bulan, E. A. Bernal, and R. P. Loce, "Detection of Moving Violations," *Comput. Vis. Imaging Intell. Transp. Syst.*, vol. 1, pp. 101–130, 2017.

[14] N. Seenouvong and U. Watchareeruetai, "A Computer Vision Based Vehicle Detection and Counting System," in *8th International Conference on Knowledge and Smart Technology*, 2016, pp. 224–227.

[15] B. Benjdira, T. Khurseed, A. Koubaa, A. Ammar, and K. Ouni, "Car Detection using Unmanned Aerial Vehicles: Comparison between Faster R-CNN and YOLOv3," in *1st Unmanned Vehicle Systems Conference*, 2018, pp. 1–6.

[16] M. Bugeja, A. Dingli, M. Attard, D. Seychell, and T. I. S. Roma, "Comparison of Vehicle Detection Techniques applied to IP Camera Video Feeds for use in Intelligent Transport Systems," *Transp. Res. Procedia*, vol. 45, pp. 971–978, 2020.

[17] J. Redmon, "YOLOv3: An Incremental Improvement."

[18] L. Ouyang and H. Wang, "Vehicle target detection in complex

scenes based on YOLOv3 algorithm," *IOP Conf. Ser. Mater. Sci. Eng.*, vol. 569, pp. 1–7, 2019.

[19] H. Song and H. Liang, "Vision-based vehicle detection and counting system using deep learning in highway scenes," *Song Eur. Transp. Res. Rev.*, vol. 11, no. 51, pp. 1–16, 2019.

[20] J. Uus and T. Krilavi˘, "Detection of different types of vehicles from aerial imagery," vol. 3, 2019.

[21] X. Ding and R. Yang, "Vehicle and Parking Space Detection Based on Improved YOLO Network Model Vehicle and Parking Space Detection Based on Improved YOLO Network Model," *J. Phys. Conf. Ser.*, 2019.

[22] R. Girshick, J. Donahue, S. Member, T. Darrell, and J. Malik, "Region-based Convolutional Networks for Accurate Object Detection and Segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 1, pp. 1–16, 2016.

[23] R. Girshick, "Fast R-CNN," in *2015 IEEE International Conference on Computer Vision (ICCV)*, 2015.

[24] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN : Towards Real-Time Object Detection with Region Proposal Networks," in *Neural Information Processing Systems (NIPS) 2015*, 2015, pp. 1–14.

[25] R. F. C. Networks and J. Dai, "R-FCN: Object Detection via Region-based Fully Convolutional Networks," in *Neural Information Processing Systems (NIPS) 2016*, 2016.

[26] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You Only Look Once: Unified, Real-Time Object Detection," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[27] W. Liu *et al.*, "SSD : Single Shot MultiBox Detector," in *European Conference on Computer Vision (ECCV) 2016*, 2016.

[28] J. Redmon and A. Farhadi, "YOLO9000: Better, Faster, Stronger," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 1–9.

[29] K. He, G. Gkioxari, P. Dollar, and R. Girshick, "Mask R-CNN," in *IEEE International Conference on Computer Vision (ICCV)*, 2017.