IOP Conference Series: Materials Science and Engineering

**PAPER** • **OPEN ACCESS**

# Visual-Based Motorcycle Detection using You Only Look Once (YOLO) Deep Network

# Visual-Based Motorcycle Detection using You Only Look Once (YOLO) Deep Network

**Fadhlan Hafizhelmi Kamaru Zaman.[1*], Syahrul Afzal Che Abdullah.[1], Noorfadzli Abdul Razak.[1], Juliana Johari.[1], Idnin Pasya.[2], Khairil Anwar Abu Kassim.[3]**

[1] Faculty of Electrical Engineering, Universiti Teknologi MARA, 40450 Shah Alam, Selangor.
[2] Microwave Research Institute, Universiti Teknologi MARA, 40450 Shah Alam, Selangor.
[3] Malaysian Institute of Road Safety Research (MIROS), 125-135, Jalan TKS 1, Taman Kajang Sentral, 43000 Kajang, Selangor.

*Corresponding author: fadhlan@uitm.edu.my

**Abstract**. In Malaysia and various Asian countries, use of motorcycle is growing in numbers, and the motorcycle is the dominating transport mode. In fact, the number of motorcycles per thousand people averaged over several major Asian cities is significantly higher than the average of the rest of the world. With growing use of motorcycle, blind spots have become one of major cause to road injuries and fatalities of motorcyclists in Malaysia. In this work, a visual-based detector is proposed to reduce the risk of blind spots in causing road injuries to motorcyclists. The objective of this detector is to provide alerts to the drivers of cars and other vehicles when there are motorcyclists in proximity to the vehicles especially around blind spot areas. In developing this solution, a variant of deep network called You Only Look Once (YOLO) is chosen as visual-based detector. This YOLO deep network is trained and tested using the 5811 collected images of motorcyclists. A benchmark with regards to its accuracy and speed is conducted by comparing this visual detector against several methods such as Aggregate Channel Features (ACF) and Faster Region Convolutional Neural Network (FRCNN). Results showed that YOLO detector is the most superior detector since it has the best average precision out of all detectors, and its inference time at 22.55 ms (44 fps) is able to provide real-time implementation. Besides, YOLO inference on machines without GPU still manage to achieve a commanding performance, which is on average, at 17 fps.

## 1. Introduction

For the urban community in Malaysia, motorcycles become a preferred vehicle for daily usage. This is because they are easier to park, use less gas, toll-free and also the best way to avoid traffic congestion. In 2018, there are 415,933 newly registered motorcycles within that year [1] and this number is increasing yearly to this day. However, with this number, motorcycles turn out to be the biggest contributor to accident numbers. The number of road accidents involving motorcycles are in 2018 alone is 113,288 [2]. There are more than 6000 fatalities and 25,000 injuries are recorded yearly for the past year of 2011 – 2016 [3]. Several factors that cause this matter to occur are identified. One of them comes from the road itself where the road is uneven, crack and contain potholes. Besides, the condition of the motorcycle where it poorly maintained or extremely modified also becomes the cause.

However, the major factor remains the motorcyclist attitude on the road. Impatience, careless, selfish and dangerous driving makes them involve an accident. Indeed, these data come to be evidence where action must be taken to reduce the number of motorcycle accidents in Malaysia. Another factor of road accident is due to distraction of vehicle drivers. Distraction is the diversion of attention away from navigating vehicle safely while responding to critical events towards competing activities. The critical activities for safe driving are impaired because when a driver is distracted, their attention is temporarily divided among the primary task of driving and secondary tasks that are not related to driving. For instance, the driver's cognitive (i.e. thinking) resource is being used to analyse both the driving situation (the primary task) and the conversation taking place (the secondary task) when making calls while driving [4, 5]

Mobile phone distracts driver in several ways: it causes physical distraction (e.g. taking their hands off the wheel to reach, dial or hold the phone), visual distraction (e.g. their eyes on the phone and off the road), cognitive distraction (e.g. instead of analysing the road situation, their mind is reflecting on the subject of conversation), and auditory distraction (e.g. the sound of the phone is loud and masks other sounds, such as ambulance sirens). As a result, the driver's situational awareness, decision-making and driving performance are affected [4, 5]. These increase the risk of accidents and the distraction may put the driver and fellow motorists in danger, as study by Malaysian Institute of Road Safety Research (MIROS) indicated that a vehicle's braking distance differed in accordance with the type of distraction faced by the driver whilst driving [6, 7] .

Although using mobile phone while driving carries a RM300 compound in Malaysia, most drivers in Klang Valley, despite being aware of the danger, they still committed the offence, i.e. by using their phone while driving. The use of the phone, whether handheld or hands-free, increases the risk of an accident because the driver's reaction time is slower, notably in braking reaction time, and also reaction to traffic signals [5]. Hands-free is not as safe as commonly believed, due to the cognitive distraction involved. Because of this biased sense of optimism, drivers continue to use their phone as they assume their risk of meeting with an accident is lower when compared to other drivers [4, 5].

In order to present countermeasures to tackle this growing problem, a technological solution, that can keep distracted driver safe, such as by providing feedback to the driver must be made available. We can decrease risk of driver losing focus while driving, especially towards motorcyclist, by means of a vision system that can help in providing alert to vehicle driver which commonly crashes with the motorcycle. When a person drives a car, this visual-based detection system is capable to detect presence of motorcycles around the vehicle especially in blind spot area. Later, the system can alert the driver to take necessary control to the car so that any accident can be prevented. With that regards, motorcyclist detection is chosen because fatalities involving motorcycle riders and passengers accounts for more than 60 percent of the 6,742 accidental road death cases in 2018 [8]. Thus, in this work, we propose a visual-based detector capable of detecting motorcyclists using a Deep Learning Neural Network (DLNN) approach called You Only Look Once (YOLO) Deep Network [9].

In the field of computer vision and artificial intelligence, the use of the Convolutional Neural Networks (CNN) for classification has dominated the field, ever more so since AlexNet [10] won the 2012 ILSVRC challenge. CNN has then been used as visual-based detector as well as classifiers that employed numerous strategies in describing and providing labels based on objects in an image. Various types of CNN architectures for object classification have been introduced since then, which includes GoogleNet [11], VGGNet [12], ResNet [13], DarkNet [9], and MobileNet [14]. Besides, several CNN region-based detectors are introduced as well which use these CNN architectures, such as Fast Region CNN (Fast RCNN) [15], Faster RCNN (FRCNN) [16], Single Shot Detector (SSD) [17], YOLO [9] and SNiPER [18]. The difference between these region-based detectors from conventional detector is conventional approach for object detection employs sliding window approach where it exhaustively slides windows from left and right, and from up to down to identify objects using classification. To detect different object types at various viewing distances, conventional detectors use windows of varied sizes and aspect ratios, which made this brute force method extremely slow and tedious. Several alternatives to this exhaustive search are proposed namely Selective Search, Region Proposal, Boundary Box Regressor, Region Proposal Network (RPN), and Anchor Boxes. These methods are proven to be more efficient and reduces time taken to perform visual object detection. For example, YOLO uses RPN and Anchor boxes approach in an end-to-end implementation and provides good performance in terms of its processing speed [19]

In this paper, we present a visual-based motorcyclist detector based on YOLO and ResNet50. The main objective of this proposed solution is to reduce the risk of drivers losing focus while driving due to distractions by providing effective warning which can help to redirect their attention to the driving task. This YOLO deep network is trained and tested using the 5811 collected images of motorcyclists. A benchmark with regards to its accuracy and speed is conducted by comparing this visual detector against several methods such as Aggregate Channel Features (ACF) [20] and FRCNN. The rest of this paper is arranged as follows: Section 2 outlines some related works while section 3 elaborate details of methodology used in this work. Section 4 analyses and discusses the results obtained from the experiments conducted. Subsequently in Section 5, the paper is concluded.

## 2. Methodology

YOLO treats object detection as a regression problem to spatially separated bounding boxes and associated class probabilities. YOLO employs a single neural network that predicts bounding boxes and class probabilities directly from full images in a single pipeline thus it can be optimized end-to-end directly on detection performance. The concept of YOLO is very simple, where a single convolutional network simultaneously predicts multiple bounding boxes and class probabilities for those boxes. YOLO resizes the input image to 448 × 448, and then runs a single convolutional network on the image, and finally thresholds the resulting detections by the model's confidence. This is shown in Figure 1.

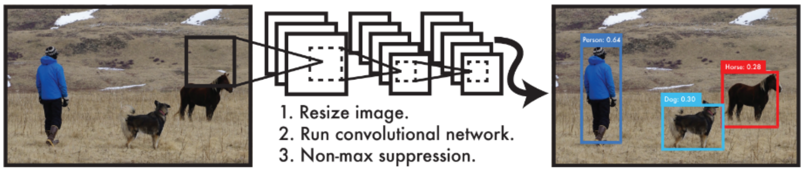


**Figure 1.** The YOLO Detection System [9].

In our implementation of YOLO in this paper, instead of using DarkNet as the base CNN, we use ResNet. ResNet is a deep residual learning framework where stacked layers are fit to a residual mapping. This type of network solves a dire issue in deep learning field where CNN has a notorious vanishing gradient problem. During training, the gradient is back propagated to earlier layers such that repeated multiplication may make the gradient infinitively small. As a result, as the network goes deeper, its performance gets saturated or even starts degrading rapidly. ResNet solves this problem by using a so-called "identity shortcut connection" that skips one or more layers [13]. In this paper, we specifically use a variant of ResNet called ResNet50, where it has 50 layers. ResNet50 has been used previously to solve image classification problems where it produces good performance and trained very quickly [21, 22].

To ensure that the proposed YOLO and ResNet50 is the best solution for our motorcyclist detection system, it should perform well on a real-time system. Thus, we design a development framework to test its precision and speed by comparing them with another popular deep network region-based detector called FRCNN and image-gradient based method called ACF. ACF extends the image channel to diverse types like gradient magnitude and oriented gradient histograms and therefore encodes rich information. It was used successfully previously in face detection [20]. Our development framework for the proposed visual-based motorcyclist detector is shown in Figure 2.

According to Figure 2, we use separate training and test images of motorcyclists. After resizing the images according to the required input size, YOLO detector on ResNet50 is trained by supplying the training images with the ground truth label. The time required to complete the process is recorded.

Then, the training performance of the detector is measured in terms of precision. The trained YOLO detector is then tested using a set of test images. The images are resized accordingly, and the trained YOLO detector will predict the bounding boxes containing motorcyclists in these images. The predictions are then compared with ground truth label to evaluate its precision. The inference time required are also recorded. Finally, the precision of training and test as well as the time taken is compared to two variants of ACF and FRCNN to determine the best method.

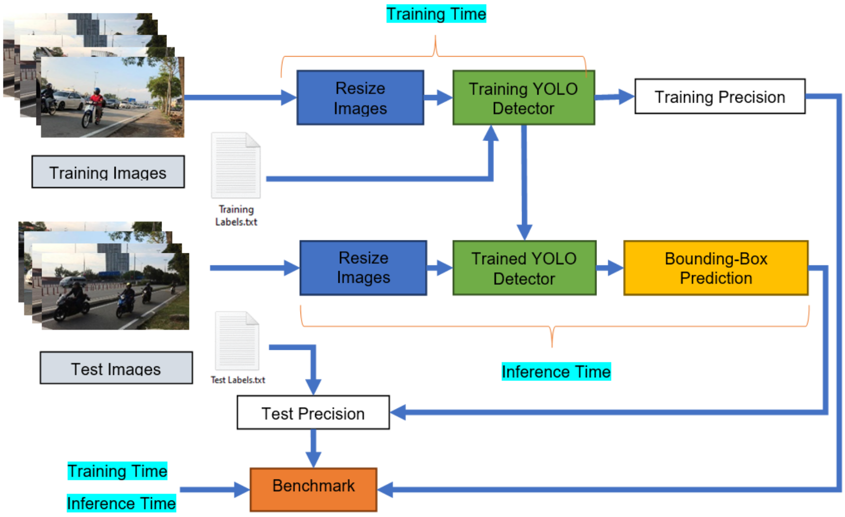


**Figure 2.** Development framework used in this paper

To ensure that the developed motor detector can be implemented effectively in vehicular environment, we need to evaluate its inference performance on several type of machines (computers). This should give a better understanding on how the detector would perform on a given machine with certain specifications. In this experiment, we are using 4 Machines denoted as Machine A, Machine B, Machine C and Machine D. Machine A and C are PC type, while Machine B is a laptop and Machine D is a Mini PC with a compact form-factor. Machine A and B has Graphics Processing Unit (GPU) to speed up computation while Machine C and D do not have any GPU on board. The specifications of these 4 machines are given in Table 1.

**Table 1.** Specifications of 4 different machines used in the experiment

| Machine | Type | Processor | Processor Frequency | RAM | GPU | Data Storage | Cost |
|---|---|---|---|---|---|---|---|
| A | PC | Intel Core i7-6700 | 3.40 GHz | 44GB | Nvidia GTX 1080Ti 11GB | External SSD 550 MB/s | $$$$ |
| B | Laptop | Intel Core i7-7700HQ | 2.80 GHz | 16 GB | Nvidia GTX 1080 Max Q 8GB | External SSD 550 MB/s | $$$$$$ |
| C | PC | Intel Core i5-9500 | 3.00 GHz | 8 GB | - | External SSD 550 MB/s | $$$ |
| D | Mini PC | Intel Core i7-5500U | 2.40 GHz | 16 GB | - | External SSD 550 MB/s | $$ |

## 3. Results and Discussions

The performance of YOLO motorcycle detector is measured to indicate the suitability of the method to be used as a visual-based motorcycle detector to be equipped in cars. These performance measures namely Average Precision (AP) and Inference time will indicate the preciseness and the inference speed of the method, which is critical for the real-time implementation. AP is a measure that combines recall and precision for ranked retrieval results where it computes the average precision value for recall value over 0 to 1. Precision measures how accurate is the predictions. i.e. the percentage of predictions are correct while recall measures how well the detector finds all the positives. In another word, precision describes how good a model is at predicting the positive class. On the other hand, recall or sensitivity measures the proportion of actual positives that are correctly identified. Meanwhile, inference time measures the time taken by the algorithm to detect all motorcycles in a single 2D image. Inference time can also be indicated by the number of images processed per second. The equation for AP, precision, and recall are given as the following:

$$AP = \frac{\sum_{r=1}^{R} P_r}{R} \quad (1)$$

$$Precision = \frac{TP}{TP+FP} = \frac{TP}{Total\ Positive\ Results} \quad (2)$$

$$Recall = \frac{TP}{TP+FN} = \frac{TP}{Total\ Relevant\ Samples} \quad (3)$$

where r is the rank of each relevant document, R is the total number of ranks, $P_r$ is the precision of the top-r positive results, TP denotes number of true positives, FP denotes number of false positives, and FN denotes number of false negatives. Additionally, the inference time measured in seconds can be computed from (4):

$$Inference\ time = \frac{Test\ processing\ time}{Number\ of\ test\ images}$$

In order to validate the performance of the proposed motorcycle detector, in this work, we collected 26 videos of frontal views of moving motorcycles, taken at a few locations, during different times of the day. The angle at which the videos are captured also varies. The captured videos are then sampled into images where 5811 images are obtained as a result. These collections of images are henceforth known as Motorcycle Dataset. Some images from this Motorcycle dataset are shown in Figure 3.

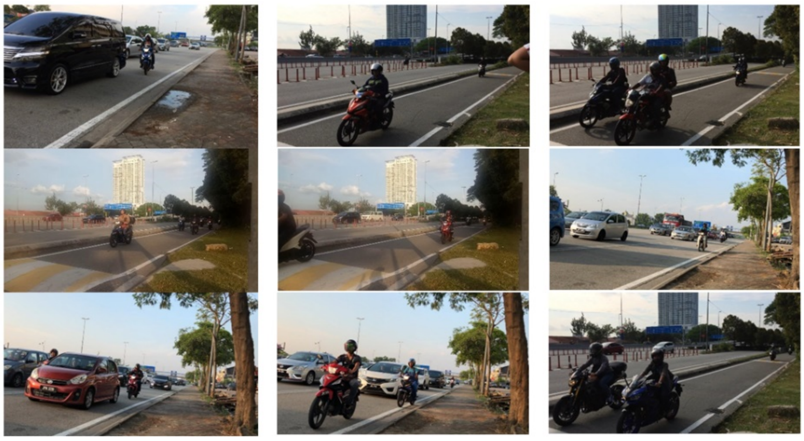


**Figure 3.** Several sample images in Motorcycle Dataset

### *3.1. Training performance of YOLO motorcycle detector*

For the experiment in this work, the Motorcycle Dataset is randomly divided into training and test set. 4067 images are used as training set, while 1744 images are used as test set. All images are resized from original dimension of 1280x720 (720p) into two different sizes, namely 640x360 (360p) and 224x224 dimension. In this experiment, the training time and Average Precision (AP) of YOLO method is compared against FRCNN and ACF. Two variants of ACF are compared here that is ACF224 and ACF360. Since in our implementation, YOLO and FRCNN uses ResNet50, the input images for YOLO and FRCNN are 224x224 dimension, while ACF360 uses images of 640x360 dimension. On the other hand, ACF224 uses images of 224x224 dimension. YOLO network is trained for 10 epochs with minibatch size of 10, while FRCNN network is trained for 10 epochs with minibatch size of 3 which is due to GPU memory size limitation. ACF224 and ACF360 are trained for 4 epochs with number of negative samples factor is set to 5. The training for all methods in this work are carried out on Machine A, and all images from Motorcycle Dataset are stored on the same External SSD having a read/write speed of 550 MB/s to eliminate read/write delay factor difference. The training time and the training AP of YOLO and other methods are shown in Table 2.

**Table 2**. Training time for YOLO compared to other detection methods

| Method | Training Time (s) | Average Precision (Training) |
|---|---|---|
| ACF224 | 621.1 | 0.6374 |
| ACF360 | 5754.7 | 0.8103 |
| YOLO | 1683.9 | 0.976 |
| FRCNN | 3239.1 | 0.7421 |

According to Table 2, processing time for YOLO network training is 1683.9 seconds which is faster than FRCNN and ACF360. ACF224 is the fastest with only 621.1 seconds taken to train its detector. This is expected since ACF224 uses smaller images than ACF360 and has less complexity than deep network methods such as YOLO and FRCNN. In terms of AP, YOLO training delivers the best AP at 0.976 which surpasses ACF360, FRCNN and ACF224. To further analyse the performance of the detector, we employ Precision-Recall curve where Precision-Recall curves summarize the trade-off between the TP rate and the positive predictive value for a predictive model using different probability thresholds. We thus vary the threshold of the detectors used and record their precision and recall across the varied threshold. This is shown in Figure 4. Based on Fig. 4, YOLO detector gives the best trade-off between precision and recall. From the curve, it is also observed that YOLO has the highest Area Under Curve (AUC) as compared to other methods. This highlights that on training data, YOLO detector delivers the best performance in term of its precision and recall.

### *3.2. Precision of YOLO Motorcycle Detector Tested on Test Data*

The trained detectors are then tested on the test dataset to measure its performance. All tests are performed on Machine A and the results of the test are shown in Table 3. Based on Table 3, YOLO detector gives the best AP of 0.8231, which is better than ACF360 at 0.7799 AP and followed by FRCNN and ACF224 at 0.6944 AP and 0.5908 AP respectively.

**Table 3.** Test precision for YOLO compared to other detection methods

| Method | Average Precision (Test) |
|---|---|
| ACF224 | 0.5908 |
| ACF360 | 0.7799 |
| YOLO | 0.8231 |
| FRCNN | 0.6944 |

Detailed analysis on the AP performance is shown on the Precision-Recall curve given in Figure 5. According to Figure 5, YOLO detector has better curve with greater AUC than ACF360, FRCNN and ACF224. ACF360 and FRCNN has similar curve except towards higher recall rate where ACF360 performs slightly better, and surprisingly despite its much simpler approach, ACF360 surpasses the FRCNN performance on the test set. ACF224 performs poorly where the obtained precision is very low in order to achieve high recall rate. Some results of YOLO detector compared with other detectors are shown in Figure 6. Based on Figure 6, the trained YOLO detector shows its superiority in detecting motorcycles correctly, especially when compared against ACF224 and FRCNN detectors. From the image samples, FRCNN and ACF360 has several errors in detection such as detecting motorcycles when there is none present (false positives) and missing the motorcycle when it is actually present (false negatives). However, YOLO detector is also not a perfect detector and it has several errors and flaws in detection as well. Some of its common error are shown in Figure 7. In this figure, YOLO has missed several motorcycles in the image, falsely detects other object as motorcycle, gives multiple detections on one motorcycle, and detects several motorcycles as a single motorcycle (overlapping detection). Despite these errors, on average from all test images, YOLO still delivers the best AP of 0.8231, which surpasses all other tested detectors.

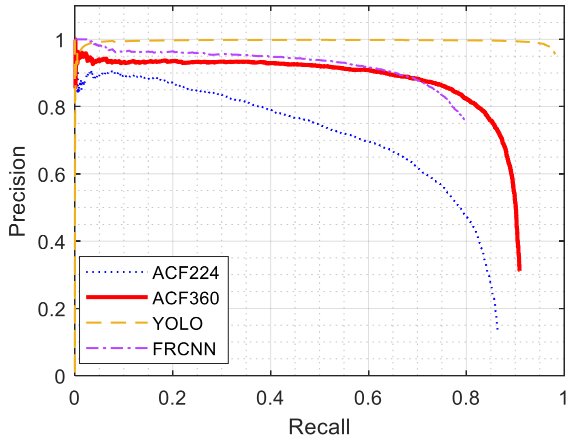


**Figure 4.** Training performance measured by Precision-Recall curve for YOLO and other tested methods

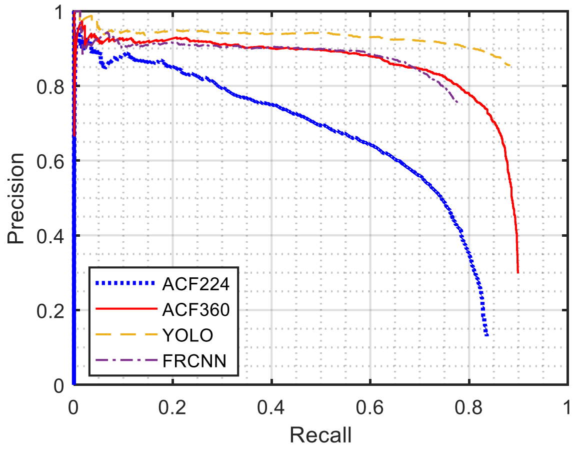


**Figure 5.** Test performance measured by Precision-Recall curve for YOLO and other tested methods

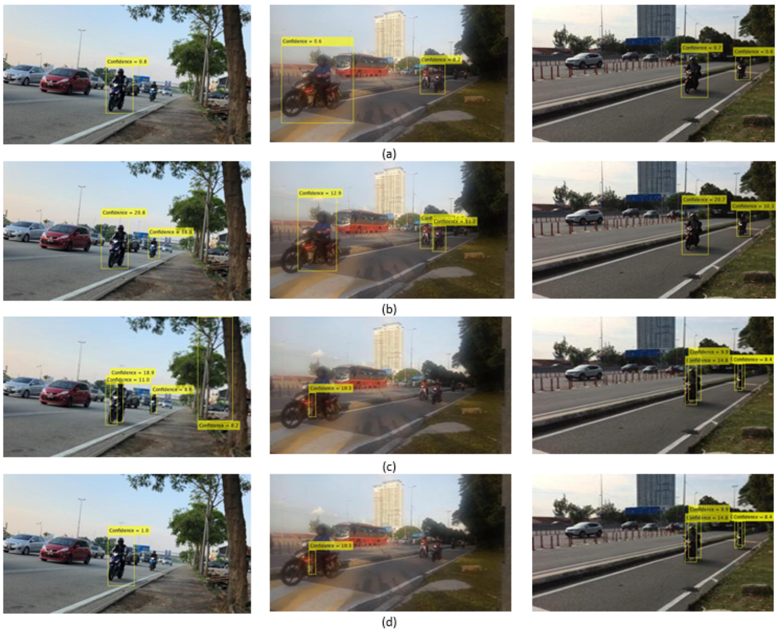
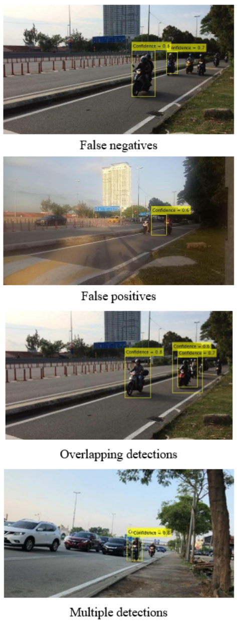


**Figure 6.** Several detection results of (a) YOLO, (b) ACF360, (c) ACF224 and (d) FRCNN are shown.


False negatives

False positives

Overlapping detections

Multiple detections

**Figure 7.** Several examples of error in detection by YOLO detector

### 3.3. *Inference Time of YOLO Motorcycle Detector*

Besides AP performance, another factor that will determine the effectiveness of the motorcycle detector is its computation speed that will ultimately determine the number of images from video streams that it can process. Higher number of images processed per second will ensure that all moving motorcycles can be detected in real-time and the detector can deliver real-time performance. One limitation is computing systems available on vehicle is usually limited in processing power, due to its limited power resource available and its cost. Thus, it is important for a detector system to be able to perform accurately while at the same time not too resource hungry. For a 30fps camera, real time processing should require the detector to be able to process 30 images in one second. In this experiment, we evaluate the inference time of YOLO motorcycle detector and other detectors using 4 different machines. We perform the detection on 1000 images and the time taken to process all images are recorded and repeated 10 times. The performance of these detectors on all machines is summarized in Figure 8.

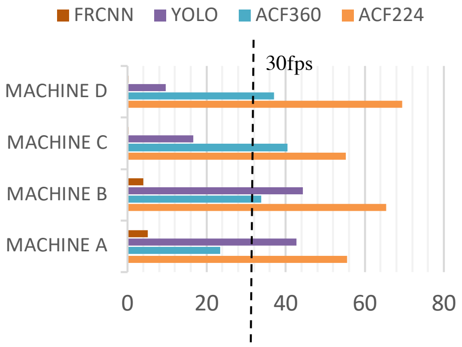
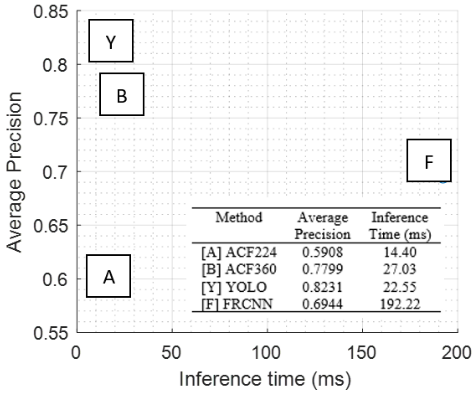


**Figure 8.** Average of number of images per second processed by YOLO and other detectors

According to Figure 8, number of images processed by ACF224 detector on all machines surpassed the 30fps whereas ACF360 achieves more than 30 fps on all machines except for Machine A. The result shows that due to simplicity of ACF method, its performance in term of number of images processed per second is acceptable on all machines, including those who do not has any GPU. On the other hand, YOLO detector only achieves 30 fps on Machine A and B that has GPU, while it struggles to achieve real-time performance on Machine D, due to lack of GPU. However, despite its deep network architecture, YOLO still shows good performance on Machine C, which do not have GPU. YOLO achieves on average, 17 fps on this machine which is quite good considering lack of GPU. Meanwhile, FRCNN failed to achieve real-time performance on all tested machines, where the highest performance is achieved on Machine A, that is slightly over 5 fps. Machine-wise, YOLO and ACF224 manage to achieve real-time performance on Machine A, while on Machine B, ACF224, ACF360 and YOLO manages to achieve real-time performance. On Machine C and D, only ACF224 and ACF360 achieve real-time performance which is due to lack of GPU on these 2 machines. Due to form factor, only Machine B (a laptop) and D (Mini PC) is suitable to be used in vehicular implementation. If we can use Machine B, YOLO detector can be deployed and will deliver real-time performance, But, looking further at the cost and power consumption, Machine D is most likely to be used on vehicular implementation. In that case, YOLO will suffer some penalty on performance if it were to be deployed on this machine, and the better choice is to use ACF360 detector to achieve real-time performance.

### *3.4. Best Inference Time vs Average Precision for YOLO Motorcycle Detector*

We compare the best inference time vs AP for all detectors used in this work and plot the result on Figure 9. Based on the result, YOLO detector is the most superior detector since it has the best AP out of all detectors, and its inference time at 22.55 ms (44 fps) is comparable to the fastest method, that is ACF 224 at 14.40 ms (69 fps). Despite being the fastest, ACF224 has very poor AP, which is the lowest of all tested detectors at 0.5908 AP. FRCNN is the slowest detector at 192.2 ms (5 fps) and it delivers worse AP than YOLO and ACF360, thus it is not a suitable choice for implementation. ACF360 on the other hand has good AP of 0.7799 while being one of the detectors that can achieve real-time performance at 27.03 ms (37 fps).



| Method | Average Precision | Inference Time (ms) |
|---|---|---|
| [A] ACF224 | 0.5908 | 14.40 |
| [B] ACF360 | 0.7799 | 27.03 |
| [Y] YOLO | 0.8231 | 22.55 |
| [F] FRCNN | 0.6944 | 192.22 |

**Figure 9.** The best inference time vs AP for all detectors used in this work

## 4. Conclusions

In this work, we propose a visual-based motorcycle detector that can be used in vehicular implementation to reduce the risk of accidents due to blind spots. A region-based detector called YOLO is proposed to perform visual-based motorcycle detection. However, in the real-world implementation, aside from the precision of the detector, the speed at which the detection can be performed is also critical, due to the real-time processing needs, and limitation of processing power that is available for such mobile implementation. In this work, we compared the performance of YOLO detector against several other detectors namely ACF and FRCNN detectors. A dataset consists of moving motorcycle images are collected for this purpose and several experiments are performed to evaluate the performance. Based on results obtained, YOLO detectors deliver superior precision and recall in detecting motorcycle at 0.8231 AP, surpassing all tested variants of ACF detectors and FRCNN detector. It is also one of the fastest detectors, capable of detecting at 44 fps. However, YOLO detector requires machine with GPU in order to achieve real-time performance. In many cases, such machine is not available due to its cost, size, and power consumption. In that case, the best alternative is ACF360 detector which can deliver real-time performance at 37 fps with good AP of 0.7799 on machine that is smaller in size, less power-hungry and less costly. However, if machine with GPU is not available at all, YOLO still achieves on average, 17 fps. One way to improve the YOLO detector speed so that it can be used on other machines, is by simplifying the CNN network that it is based on. In our case, we can consider the use lightweight MobileNet instead.

## Acknowledgements

This research is funded by Society of Automotive Engineers Malaysia (SAE Malaysia) and Universiti Teknologi MARA (UiTM), Shah Alam, Selangor, Malaysia under the Research Grant Scheme No: 100-IRMI/INT 16/6/2 (003/2019).

## References

[1] (2020, 4 March). *Malaysia Number of Motor Vehicle: Newly Registered: Motorcycles*. Available: https://www.ceicdata.com/en/malaysia/motor-vehicles-registration/number-of-motor-vehicle-newly-registered-motorcycles

[2] "Transport Statistics Malaysia 2018," Ministry of Transport Malaysia2019, Available: http://www.mot.gov.my/en/Statistik%20Tahunan%20Pengangkutan/Transport%20Statistics%20Malaysia%202018.pdf.

[3] Z. Sultan, N. I. Ngadiman, F. D. A.Kadir, N. F. Roslan, and M. Moeinaddini, "FACTOR ANALYSIS OF MOTORCYCLE CRASHES IN MALAYSIA," *Journal of the Malaysian Institute of Planners,* vol. SPECIAL ISSUE IV, pp. 135-146, 2016.

[4] "Overview of the National Highway Traffic Safety Administration's Driver Distraction Program," United States Department of Transportation 2010, Available: https://www.nhtsa.gov/sites/nhtsa.dot.gov/files/811299.pdf.

[5] "Mobile Phones Use: A Growing Problem of Driver Distractions," Belgium2011, Available: https://www.who.int/violence_injury_prevention/publications/road_traffic/distracted_driving_en.pdf?ua=1.

[6] K. Kamarudin. (2011, 4 March). *Fikirlah: Drivers Oblivious To Danger of Using Phone While Driving*. Available: http://youth.bernama.com/v2/news.php?id=1688201&c=7

[7] M. Lum. (2019, 4 March). *We have the third highest death rate from road accidents*. Available: https://www.thestar.com.my/lifestyle/health/2019/05/14/we-have-the-third-highest-death-rate-from-road-accidents

[8] (2016, 4 March). *Index of Accidental Road Death in Malaysia from 2011 to 2016*. Available: http://www.mot.gov.my/en/lands/public%20safety-roads/statistics-accidents-jalan-raya-index-death

[9] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You Only Look Once: Unified, Real-Time Object Detection," *arXiv e-prints,* p. arXiv:1506.02640Accessed on: June 01, 2015Available: https://ui.adsabs.harvard.edu/abs/2015arXiv150602640R

[10] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," *Advances in Neural Information Processing Systems,* 2012.

[11] C. Szegedy *et al.*, "Going deeper with convolutions," in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015, pp. 1-9.

[12] K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition," *arXiv e-prints,* p. arXiv:1409.1556Accessed on: September 01, 2014Available: https://ui.adsabs.harvard.edu/abs/2014arXiv1409.1556S

[13] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," *arXiv e-prints,* p. arXiv:1512.03385Accessed on: December 01, 2015Available: https://ui.adsabs.harvard.edu/abs/2015arXiv151203385H

[14] A. G. Howard *et al.*, "MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications," *arXiv e-prints,* p. arXiv:1704.04861Accessed on: April 01, 2017Available: https://ui.adsabs.harvard.edu/abs/2017arXiv170404861H

[15] R. Girshick, "Fast R-CNN," *arXiv e-prints,* p. arXiv:1504.08083Accessed on: April 01, 2015Available: https://ui.adsabs.harvard.edu/abs/2015arXiv150408083G

[16] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," *arXiv e-prints,* p. arXiv:1506.01497Accessed on: June 01, 2015Available: https://ui.adsabs.harvard.edu/abs/2015arXiv150601497R

[17] W. Liu *et al.*, "SSD: Single Shot MultiBox Detector," *arXiv e-prints,* p. arXiv:1512.02325Accessed on: December 01, 2015Available: https://ui.adsabs.harvard.edu/abs/2015arXiv151202325L

[18] B. Singh, M. Najibi, and L. S. Davis, "SNIPER: efficient multi-scale training," presented at the Proceedings of the 32nd International Conference on Neural Information Processing Systems, Montréal, Canada, 2018.

[19] J. Redmon and A. Farhadi, "YOLO9000: Better, Faster, Stronger," *arXiv e-prints,* p. arXiv:1612.08242Accessed on: December 01, 2016Available: https://ui.adsabs.harvard.edu/abs/2016arXiv161208242R

[20] B. Yang, J. Yan, Z. Lei, and S. Z. Li, "Aggregate channel features for multi-view face detection," *arXiv e-prints,* p. arXiv:1407.4023Accessed on: July 01, 2014Available: https://ui.adsabs.harvard.edu/abs/2014arXiv1407.4023Y

[21] H. Mikami, H. Suganuma, P. U-chupala, Y. Tanaka, and Y. Kageyama, "Massively Distributed SGD: ImageNet/ResNet-50 Training in a Flash," *arXiv e-prints,* p. arXiv:1811.05233Accessed on: November 01, 2018Available: https://ui.adsabs.harvard.edu/abs/2018arXiv181105233M

[22] M. Yamazaki *et al*., "Yet Another Accelerated SGD: ResNet-50 Training on ImageNet in 74.7 seconds," *arXiv e-prints,* p. arXiv:1903.12650Accessed on: March 01, 2019Available: https://ui.adsabs.harvard.edu/abs/2019arXiv190312650Y