Proceedings of the Seventh International Conference on Research in Intelligent and Computing in Engineering pp. 279–284 DOI: 10.15439/2022R05 ISSN 2300-5963 ACSIS, Vol. 33

Safety-Assisted Driving Technology Based on Artificial Intelligence and Machine Learning for Moving Vehicles in Vietnam

Hong-Son Vu[1,*] , Van-Hien Nguyen[1]
[1]Faculty of Electrical and Electronics Engineering, Hung Yen University of Technology and Education
* Email: hongson.ute@gmail.com

***Abstract*—ADAS (Advanced Driver Assistance Systems) plays an important role in building a safe and modern traffic system. For these systems, precise detection performance and response speed are critical. However, the detection of mobile vehicles is facing many difficulties due to the density of vehicles, the complex background scene in the city, etc. In addition, the detection and identification requirements respond in real time is also a challenge for current systems. This paper proposes a model using deep learning algorithms and artificial intelligence to increase accuracy and improve response speed for intelligent driving assistance systems. Accordingly, this paper proposes the YOLO (You Only Look One) model together with a sample data set collected and classified separately suitable for Vietnam traffic and our training algorithm. The experimental results were then performed on an NVIDIA Jetson TX2 embedded computer. The experimental results show that, the proposed method has increased the speed by at least 1.5 times with the detection rate reaching 79% for the static camera system; and speed up at least 1.5x with a detection rate of 89% for the dynamic camera system at 1280x720px high resolution images.**

***Index Terms*—ADAS Advanced Driver Assistance Systems, YOLO model, Deep learning algorithm and Artificial Intelligence.**

1. Introduction

ADAS - Advanced Driver Assistance Systems plays an important role in building a safe and modern traffic system. In addition to building infrastructure, the development of new technologies for smart mobile vehicles is one of the most important elements that make up this system. Detecting and recognizing surrounding vehicles and giving warnings to users will help make driving safer. Today, with the hardware platform are embedded computers with compact size, high performance, integrated with many powerful supporting technologies for image processing, along with machine learning models developed by leading technology companies to train deep learning neural networks, this makes it easy for researchers to choose the right machine learning model to test the proposed methods.

The research results of applying deep learning algorithms and artificial intelligence to detect and recognize moving vehicles for ADAS intelligent driving assistance systems have been pointed out by the authors in [1] [2] [3]. This paper proposes a model using deep learning algorithms and artificial intelligence to increase accuracy and improve response speed for Advanced Driver Assistance Systems. Accordingly, we first propose the YOLO model [4] along with a sample data set collected and classified separately suitable for Vietnamese traffic and our training algorithm. The experimental results were then performed on an NVIDIA Jetson TX2 embedded computer. The experimental results show that, the proposed method has increased the speed by at least 1.5 times with the detection rate reaching 90% for the static camera system; and speed up at least 1.5x with a detection rate of 67% for dynamic camera systems at 1280x720px high resolution images.

2. Related research

2.1. YOLO model

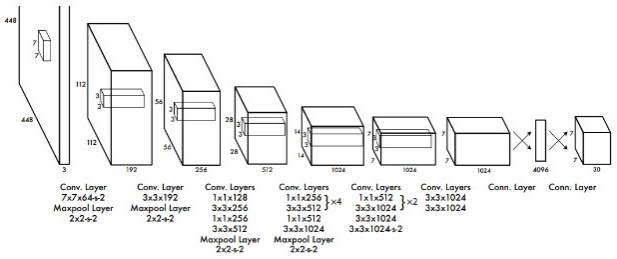

Figure 1: YOLO algorithm model.

YOLO [5] is a convolutional neural network (CNN) model that enables object detection and recognition. YOLO is created from a combination of convolutional layers and connected layers. In which the convolutional layers will extract the features of the image, while the full-connected layers will predict 8the probability and coordinates of the object.

In algorithms, indexes are always evaluated against the ground truth (which is predefined from the dataset via coordinates (cx, cy, w, h) to help identify the object). Ground truth is the location of the provided objects in the training, validated and tested dataset. For object detection systems, the ground truth includes the image, the classes of objects in it, and the frames surrounding the ground truth (ground truth box) of each object in the image.

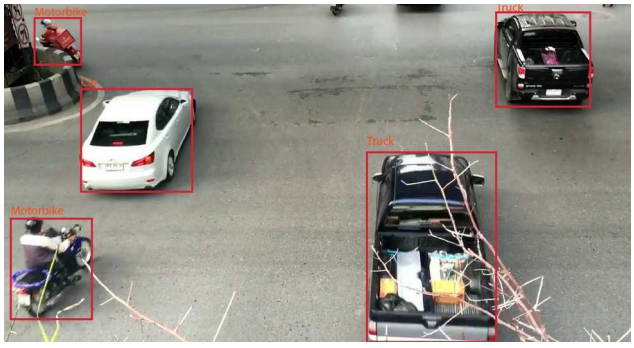

Figure 2: Define ground truth for objects

As shown in Figure 2, the ground truth boxes defined on the trained and tested images. Assume the original image and the caption for the ground truth are the same as the image above, training and test data all images are annotated in the same way, the model will return a lot of predictions, but most of them have very low confidence, so only those predictions above a certain confidence are considered. The original image through the model and the object detection algorithm will return the results of the location of the object in the image according to a threshold of confidence.

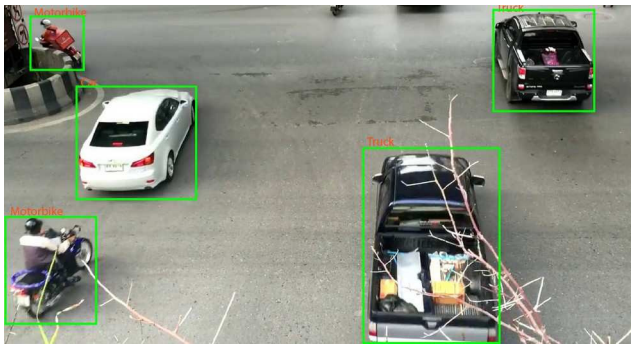

Figure 3: The model's prediction results

To evaluate the accuracy, it is first necessary to evaluate the accuracy of each prediction on the image. To calculate the accuracy of a bounding box, it is necessary to use the IoU measure - Intersection over Union: a ratio that measures the degree of intersection between two frames (usually the prediction frame and the ground truth frame) to aim determine if 2 frames overlap or not.

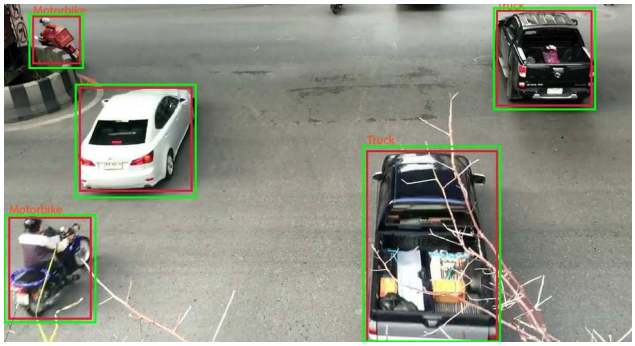

Figure 4: Compare the prediction results with the object's ground truth.

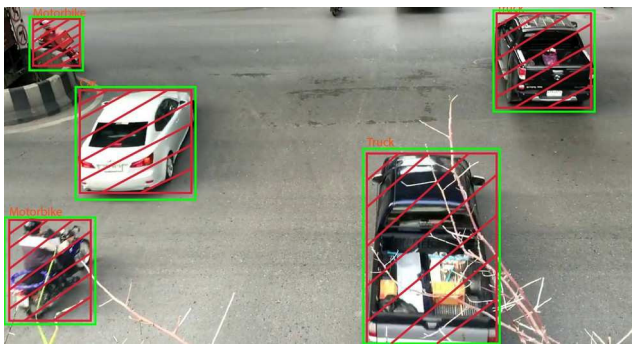

Figure 5: The area of intersection between the prediction result and the ground truth.

The IoU will be calculated as follows:

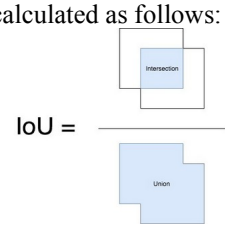

Figure 6: IoU

This ratio is calculated based on the area of intersection between 2 frames with the total area of intersection and non-intersection between them.

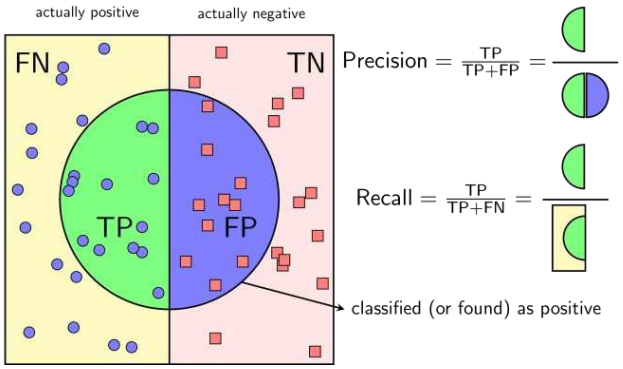

Figure 7: Precision and Recall

For classification problems where the data sets of the classes differ greatly (unbalanced), the Precision-Recall evaluation method will often be used [6].

Then, Precision is defined as the ratio of the number of positive points the model correctly predicts to the total number of points the model predicts is positive. The higher the precision, the higher the number of points the model predicts is positive. Precision = 1, that is, all the points that the model predicts are positive are correct, or there are no points labeled as negative that the model incorrectly predicts as positive.

$$Precision = \frac{TP}{TP+FP} = \frac{TP}{Total\ prediction} \quad (1)$$

Recall is defined as the ratio of the number of positives the model correctly predicted to the total number of points that are actually positive (or the total number of points labeled as positive initially). The higher the recall, the lower the number of missed positives. Recall = 1, that is, all points labeled as positive are recognized by the model [6].

$$Recall = \frac{TP}{TP+FN} = \frac{TP}{ground\ truth} \quad (2)$$

2.2. Training methods

YOLO predicts multiple bounding boxes for each grid cell. To calculate the error for correct predictions, it is necessary to define one of the bounding boxes responsible for the object. For this purpose, choose the one with the highest IoU with a true bounding box. This strategy leads to specialization in bounding box prediction. It is better to predict certain sizes and aspect ratios. YOLO uses the Sum-Squared Error function between the prediction and the desired value to calculate the loss [5].

The YOLO loss function has the form:

$$
\begin{aligned}
L &= \lambda_{coord} \sum_{i=0}^{S^2} \sum_{j=0}^{B} l_{ij}^{obj} [(x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2] \\
&+ \lambda_{coord} \sum_{i=0}^{S^2} \sum_{j=0}^{B} l_{ij}^{obj} [(\sqrt{w_i} - \sqrt{\hat{w}_i})^2 + (\sqrt{h_i} - \sqrt{\hat{h}_i})^2] \\
&+ \sum_{i=0}^{S^2} \sum_{j=0}^{B} l_{ij}^{obj} (C_i - \hat{C}_i)^2 + \lambda_{noobj} \sum_{i=0}^{S^2} \sum_{j=0}^{B} l_{ij}^{noobj} (C_i - \hat{C}_i)^2 \\
&+ \sum_{i=0}^{S^2} l_{ij}^{obj} \sum_{c \in Classes} [p_i(c) - \hat{p}_i(c)]^2
\end{aligned} \tag{3}
$$

The first part of the loss calculation equation deals with the location of the prediction bounding box and the ground truth bounding box based on the coordinates *(xcenter, ycenter)*.

l_{ij}^{obj} is equal to 1 if the object appears inside the predictor bounding box j^{th} in the j^{th} cell, and 0 otherwise. The bounding prediction box will be responsible for predicting an object based on the current highest IoU prediction.

$$\lambda_{coord} \sum_{i=0}^{S^2} \sum_{j=0}^{B} l_{ij}^{obj} [(x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2]. \tag{4}$$

The second part of the loss function, YOLO calculates the error in predicting the width and height. However, the magnitude of the error in the large boxes affects the equation in the small boxes. Since both width and height are normalized between 0 and 1, their square root increases more than the difference for small and large values. From here on, the square root of the bounding box's width and height is used instead of the direct width and height.

$$\sum_{i=0}^{S^2} \sum_{j=0}^{B} l_{ij}^{obj} (C_i - \hat{C}_i)^2 + \lambda_{noobj} \sum_{i=0}^{S^2} \sum_{j=0}^{B} l_{ij}^{noobj} (C_i - \hat{C}_i)^2 \tag{5}$$

The loss of confidence score is calculated in both cases whether the subject is present in the bounding box or not. The loss function only evaluates the reliability of the object if that predictor is responsible for the ground truth box. l_{ij}^{obj} is equal to 1 when there is an object in the cell and 0 otherwise. l_{ij}^{noobj} is the opposite.

$$\lambda_{coord} \sum_{i=0}^{S^2} \sum_{j=0}^{B} l_{ij}^{obj} [(\sqrt{w_i} - \sqrt{\hat{w}_i})^2 + (\sqrt{h_i} - \sqrt{\hat{h}_i})^2] \tag{6}$$

The last part of the loss function is similar to the normal loss function. This term is used because YOLO evaluates the classification error even if no objects appear in cell [7].

2.3. Data preparation and processing

The dataset used during model training and test runs is extracted from car dash cams and fixed cameras at intersections. With the model training dataset, the videos are converted into 1280x720 px images. Includes 500 images, of which 450 images are used for training and 50 images for testing the accuracy of the newly trained model.

The author uses the retraining method from the training model yolov5s.pt and yolov7-tiny.pt

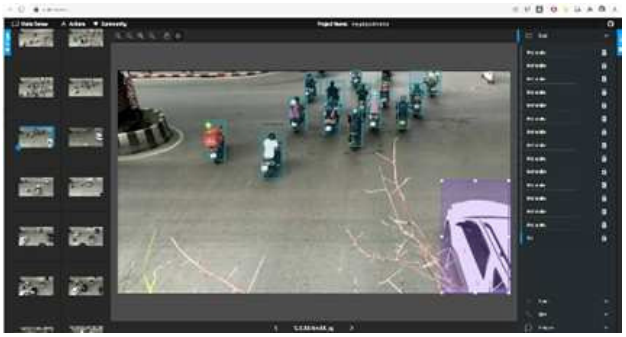

Figure 8: Create a database using makesense.ai.

The model training image dataset is preprocessed with makesense.ai and attached parameters include: [class number] [x coordinate of object center] [y-coordinate of object center] icon] [width] [height]. The data is stored as a .txt file for each image. In which ID for the classes in turn: 0-Car, 1-Motorbike, 2-Truck, 3-Bus.

2.4. Model training

Model training is a program that works continuously, consuming a lot of resources such as RAM, GPU, CPU, so to ensure fast and accurate training based on available hardware platforms, we have using Google's virtual server called Google Colab [8]. Colab offers 3 types of configurations:

Table 1: Technical Specifications of Google Colab

CPU	GPU	TPU
- Intel Xeon Processor with two core @ 2.30 Ghz and 13GB RAM - OS: Ubuntu 18.04.2 LTS - Total size of Disk: 78.0 GB (48.0 GB Used)	- Up to Tesla K80 12 GB of GDDR5 VRAM, Intel Xeon Processor with two core @ 2.30 Ghz and 13GB RAM - OS: Ubuntu 18.04.2 LTS - Total size of Disk: 78.0 GB (48.0 GB Used)	- Cloud TPU with 180 teraflops of computation, Intel Xeon Processor with two core @ 2.30 Ghz and 13GB RAM - OS: Ubuntu 18.04.2 LTS - Total size of Disk: 78.0 GB (48.0 GB Used)

We choose the GPU configuration to train the model as well as run the test of the mobile vehicle detection and recognition program using the YOLO model.

The YOLO project is hosted on Github and to use it we downloaded the project, then ran the train.py file so that we could start training the removable vehicle recognition model.

2.5. Installation and testing

2.5.1. Test environment

The test platform is Ubuntu 18.04.2 LTS operating system along with NVIDIA Jetson TX2 hardware with the following configuration:

Table 2: Technical Specifications of NVIDIA Jetson TX2

GPU:	256-core NVIDIA Pascal™ GPU architecture with 256 NVIDIA CUDA cores
CPU:	Dual-Core NVIDIA Denver 2 64-Bit CPU & Quad-Core ARM® Cortex®-A57 MPCore
Memory :	8GB 128-bit LPDDR4
Storage:	32GB eMMC 5.1

With GPU graphics hardware that supports CUDA cores, NVIDIA Jetson TX2 delivers powerful and specialized performance in handling AI & ML related tasks.

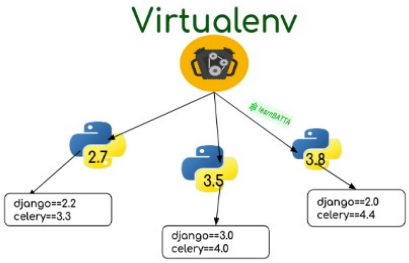

Figure 9: Python Virtual Environment.

First, we set up a virtual environment using Python Virtual Environment [9]. Virtual environments are used to isolate the environments of projects from each other. The virtual environment allows installation and management of installation packages separately and does not conflict with the system-wide installation package manager.

Library packages to install: Python 2.6.9, Pytorch 1.8.0, Torchvision 0.9.0, OpenCv 4.5.4.60, Matplotlib, Pillow, Pyyaml, Tensorboard, Tqdm, Scipy, Pandas, Seaborn, Numpy

2.5.2. Model of a mobile vehicle detection and identification system

The steps of the proposed method are as follows: First the video input data is split into frames and converted to a resolution of 1280x720 px, which is the optimal resolution for speed as well as enough quality to determine object definition. The extracted data will be compared with the pre-trained model and fed into YOLO's object recognition algorithm. Output data includes object coordinates, object ID will be zoned and labeled accordingly. The output of the system is the video displayed in real time along with which will be stored as .mp4.

We run tests on two versions of YOLO v5 and YOLO v7 to compare, evaluate and select the best version for recognizing removable media objects.

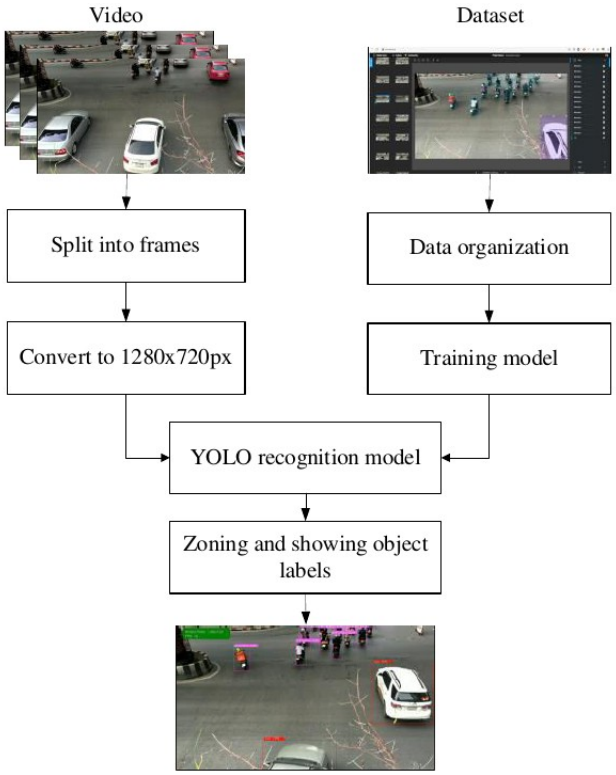

Figure 10: Model of the detection and identification system.

3. Research results and discussion

3.1. Object Recognition from Fixed Camera

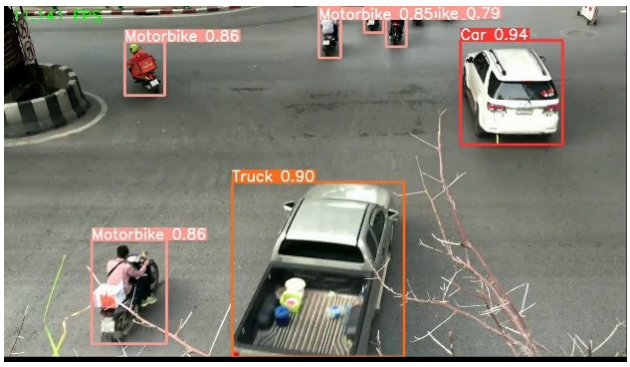

Figure 11: Identification results using fixed camera YOLO v5 model.

Figure 12: Identification results using fixed camera YOLO v7 model.

Experimental results show that, in the environment of the intersection under the bridge, many vehicles passing through both model versions fully recognize the objects. With the YOLO v5 model, the accuracy reaches from 79%, the average frame rate is about 11 FPS at 1280x720 video resolution. With the YOLO v7 model, the accuracy is from 71% but has a higher average frame rate of about 13FPS. The system shows stability and there is no overlapping of the identified frame or subject.

3.2 Object recognition from Dashcam mounted on cars

Figure 13: Identification results using Dashcam YOLO v5 model.

Figure 14: Identification results using Dashcam YOLO v7 model.

Stay in a moving vehicle in a more complex environment with many noisy objects such as trees, buildings, roadside foreign objects. The system gives relatively good results. With the YOLO v5 model, the accuracy is 89% or higher, the average frame rate is about 12 FPS at 1280x720 px resolution. With the YOLO v7 model, the accuracy is 82% or higher, the average frame rate is more than 13 FPS at 1280x720 px resolution. The system shows stability and there is no overlapping of the identified frame or subject.

The recognition system achieves the display accuracy rate of > 51%. Partly due to camera quality, the rest is due to small and blurry objects so there are still unrecognizable vehicles or low % accuracy. During actual driving, objects located on the opposite side of the road and at a distance do not usually cause an accident to the driver so the current results are acceptable.

3.3. Comparison and evaluation

Conduct a detailed test and compare 2 identification models YOLO v5 and YOLO v7 with a training model built specifically for Vietnam's traffic system including: Cars, motorbikes, trucks, buses.

Table 3: Compare recognition models.

Data	Total number of frames	Resolution	Model	Detection Rate (%)	Miss rate (%)	False detection rate (%)	FPS
Fixed camera	513	1280x720	YOLOv5	87.5	12.5	0	11
		1280x720	YOLOv7	87.5	12.5	0	13
Dashcam	1575	1280x720	YOLOv5	66.7	33.3	0	12
		1280x720	YOLOv7	66.7	33.3	0	13

The comparison table shows that, with the recognition model YOLO v5 and YOLO v7, both have a relatively high rate of detecting and correctly identifying objects.

The YOLO v7 model shows an improvement in processing speed compared to the old model YOLO v5 both in terms of fixed and mobile cameras.

4. Conclusion

Preliminary results obtained when using YOLO model in training and object recognition yield relatively good results, showing great potential in building a general intelligent traffic model and assistance system. ADAS advanced driving in particular. Moreover, the successful test on NVIDIA Jetson TX2 device, this opens a new approach in real-time recognition of different objects executed directly devices on camera devices that are available on the mobile system. Currently, the model training data is limited, increasing the number of images will bring better accuracy. In addition, along with the development of hardware technology, the model will be able to improve in processing speed.

Acknowledgment

This work was supported by Ministry of Education and Training, Vietnam, under Grant MOET B2020-SKH-02.

References

[1] A. F. Agarap, Deep Learning using Rectified Linear Units (ReLU), arXiv:1803.08375, 2018.

[2] G. S. W. Luger, Artificial Intelligence: Structures and Strategies for Complex Problem Solving, ISBN 978-0-8053-4780-7, 26 July 2020.

[3] M. Galvani, History and future of driver assistance," IEEE Instrumentation Measurement Magazine, ISSN 1941-0123, 2019.

[4] Ultralytics, "YOLOv5 Documentation," [Trực tuyến]. Available: https://docs.ultralytics.com/.

[5] S. D. R. G. A. F. Joseph Redmon, You Only Look Once: Unified, Real-Time Object Detection, arXiv:1506.02640 [cs.CV], 8 Jun 2015.

[6] M. Schumann, A Book about Colab: (and related activities), ISBN 978-0-89439-085-2, 2015.

[7] GeeksforGeeks, "Python Virtual Environment | Introduction," 2020. [Trực tuyến]. Available: https://www.geeksforgeeks.

[8] C. H. Thuc, "Precision, Recall và F1-score là gì?," 23 02 2020. [Trực tuyến]. Available: https://caihuuthuc.wordpress.com/2020/02/23/precision-recall-va-f1-score-la-gi/.

[9] D. Thuan, Evolution of YOLO Algorithm and YOLOv5: The State-of-the-art Object Detection, Bachelor thesis (3.092Mt), Spring 2021.

[10] H.-S. Vu, J.-X. Guo, K.-H. Chen, S.-J. Hsieh và D.-S. Chen, A real-time moving objects detection and classification approach for static cameras, IEEE International Conference on Consumer Electronics-Taiwan (ICCE-TW), 2016.

[11] V. T. D. T. D. N. Hong Son Vu, MỘt Phương Pháp Phát Hiện Điểm Mù Với Độ Tin Cậy Cao Và Thời Gian Thực Cho Các Hệ Thống Hỗ Trợ Lái Xe Thông Minh, Moet B2020-SKH-02, October 2020.

[12] V. H. Son, A high dynamic range imaging algorithm: implementation and evaluation, Engineering and Technology - Research article, Aug 7, 2019.