Robust Vehicle Detection by Using Deep Learning Feature and Support Vector Machine

Vinh Dinh Nguyen, Thanh Hoang Tran, Doan Thai Dang, and Narayan C. Debnath

Department of Software Engineering School of Computing and Information Technology,
Eastern International University, Vietnam

Outline

- Introduction
- Proposed Method
- Experimental Results
- Conclusion and Future Work

Introduction

- Vehicle detection and classification is an essential tasks in building an autonomous driving car. However, the performance of the existing vehicle detection system is still not good enough due to unstable features from the original input images.
- This research introduces a new approach to increase the input feature more stable by considering the benefit of the deep learning approach to learn and extract robust features. The feature generated by the proposed deep learning model is then input to support vector machine to accurately detect and classify objects.
- To evaluate the precision, recall, and f1 score of the proposed method and compare methods, we conducted comprehensive experiments on various datasets, such as KITTI, CCD, and HCI. For the KITTI dataset, the recall of the proposed method, the LS-SVM, the Linear SVM, the SVM-HOG, and the SVM are 71.4%, 64.3%, 53.8%, 67.8.0%, and 57.1%, respectively. The f1 score of the proposed method, the LS-SVM, the Linear SVM, the SVM-HOG, and the SVM are 80.0%, 75.0%, 63.6%, 77.5%, and 69.6%, respectively.

- According to the information of WHO, approximately 1.3 million people die each year as a result of road traffic crashes [1]. Therefore, it is necessary to need to find a way to reduce vehicle accidents to save human life. Recently, research in vehicle detection and classification has been attracted by many research institutes and universities. Many methods and approaches have been proposed to increase the performance of vehicle detection and classification systems.

- The existing vehicle detection and classification systems, such as SVM [2], SVM-HOG [3], Linear SVM [4], and LS-SVM [5], are mostly trained and tested on the raw input images. If the input features are affected by noise, their performances also are decreased. Therefore, somehow, we need to find an approach to solve these limitations. This research studies a new approach by using the benefits of the deep learning-based convolutional neural network to generate robust features from the original raw images.

- **Research in obstacle detection and classification algorithm has been extracted by many researchers. Without loss of generality, researchers have classified the existing obstacle detection and classification algorithm into two main classes: one stage-based method and two stage-based methods.**

- **In two stage-based methods, obstacle detection and classification algorithms [6][7] often start with generating many proposal candidates on the input image. After that, the proposed candidates are then input into the detection and classification stage to accurately detect the object in the input image. In one stage-based method [8], obstacle detection and classification algorithms do not need to go through the first stage of the generation proposal candidate, the input images are passed directly to the detection and classification steps.**

- **Researchers found that the two-stage-based method achieves more accuracy than the one stage-based method because of the bene- fits of generating proposal candidates on the input images. However, the processing time of two-stage-based methods is much higher than one-stage-based methods.**

- **In general, we realized that both one-stage and two stages-based methods often use the raw input data, directly, to detect and classify objects. Therefore, somehow, the performance of existing obstacle detection and classification might decrease if the input im- age was affected by noise. This research studies a new approach by using the benefits of the deep learning-based convolutional neural network to generate robust features from the original raw images.**

Proposed Method

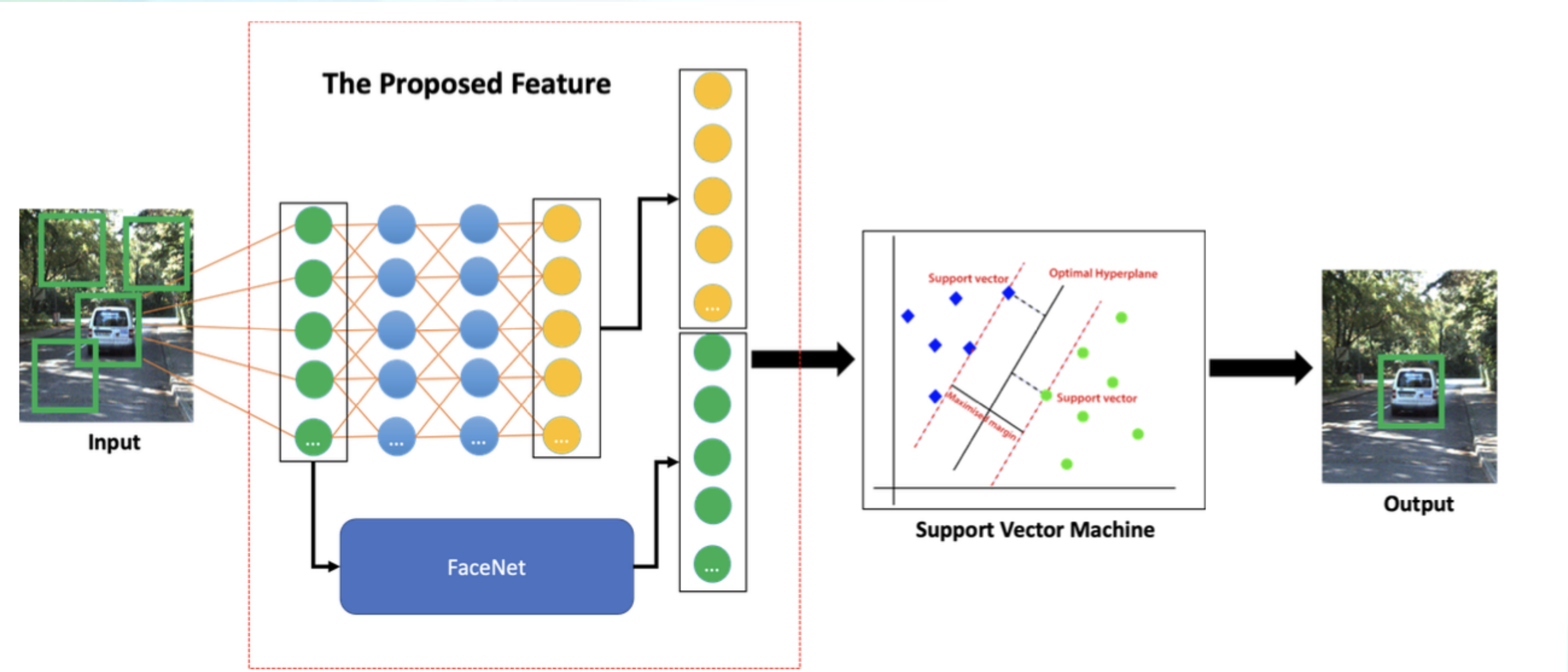

Figure 1 Flow chart of the proposed system for vehicle detection

```
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 dense_36 (Dense)            (None, 1024)              16778240

 dropout_32 (Dropout)        (None, 1024)              0

 dense_37 (Dense)            (None, 512)               524800

 dropout_33 (Dropout)        (None, 512)               0

 dense_38 (Dense)            (None, 256)               131328

 dropout_34 (Dropout)        (None, 256)               0

 dense_39 (Dense)            (None, 128)               32896

 dropout_35 (Dropout)        (None, 128)               0

 dense_40 (Dense)            (None, 2)                 258

=================================================================
Total params: 17,467,522
Trainable params: 17,467,522
Non-trainable params: 0

_________________________________________________________________
```

Figure 2 Configuration of the proposed deep learning model for feature extraction

Experimental Results

4.1 Datasets, Evaluation Methods and System Configuration

There are various datasets were used to evaluate the performance of existing object detection systems. This research aims to conduct experiments on three datasets, the KITTI dataset [9], the HCI Dataset [12], and the CCD dataset [7].

To analyze the results of the proposed method and related research, we conduct experiments and compare them to various object detection methods, such as SVM [2], SVM-HOG [3], Linear SVM [4], and LS-SVM [5].

To measure the accuracy of the object detection method, precision ∂ and recall metric β , and F1 score are calculated as follows:

$$\partial = \frac{\delta}{\sigma + \delta}$$

$$\beta = \frac{\sigma}{\nu + \sigma}$$

$$F1 = 2\frac{\partial * \beta}{\partial + \beta}$$

Where δ is the number of true positive detection results. σ is the number of false positive detection results. ν is a number of false negative detection results.

Experimental Results

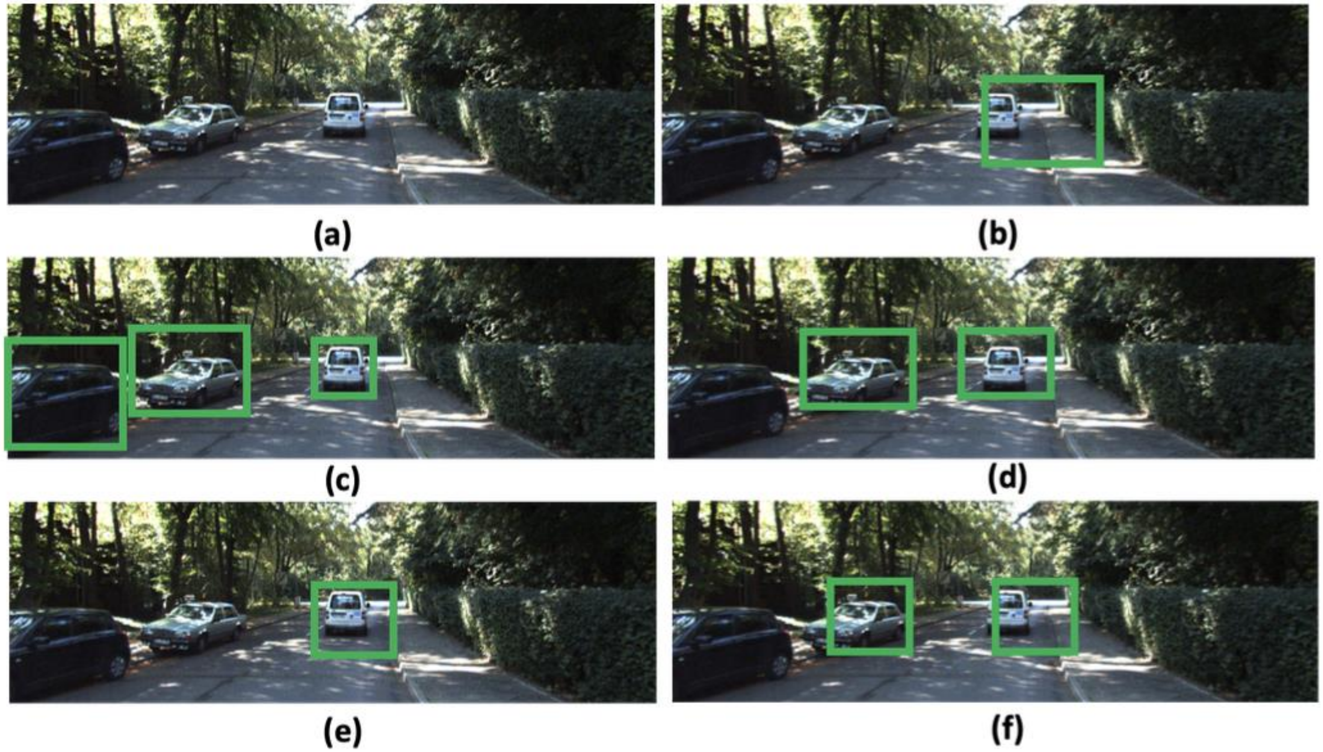

Figure 4 Exprimental results of the proposed system and compared method on the KITTI dataset. (b), (c), (d), (e), and (f) are results of the SVM, the proposed method, Linear SVM, LS-SVM, and SVM-HOG on the input image (a).

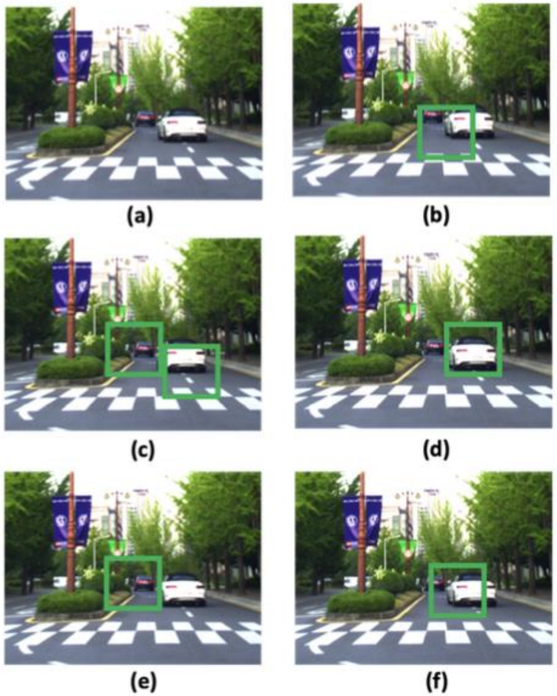

Figure 5 Exprimental results of the proposed system and compared method on the CCD dataset. (b), (c), (d), (e), and (f) are results of the SVM, the proposed method, Linear SVM, LS-SVM, and SVM-HOG on the input image (a).

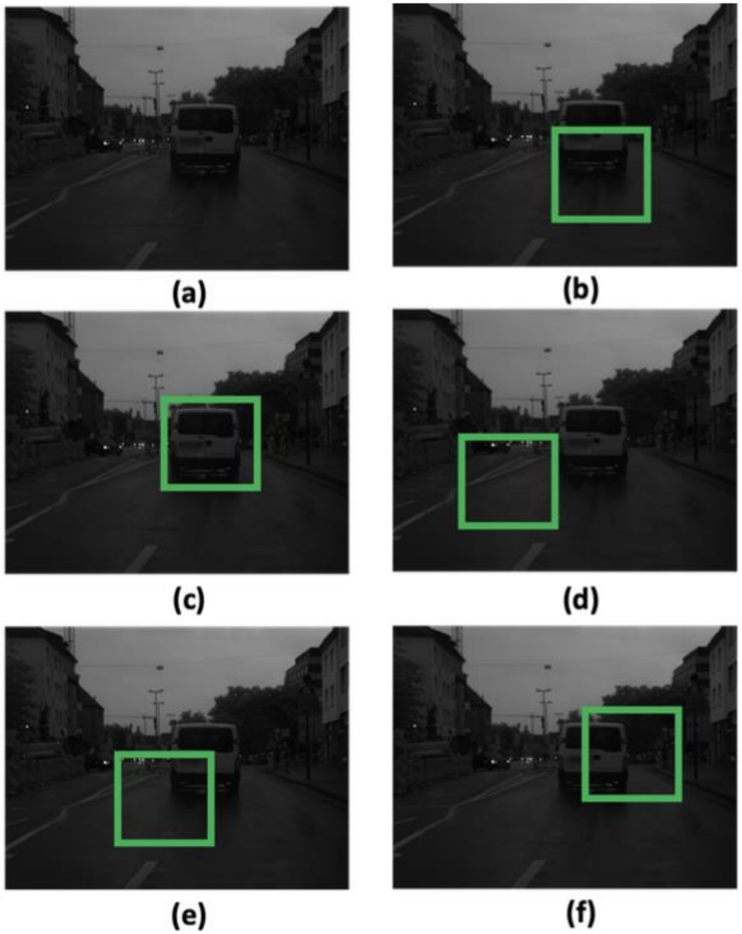

Figure 6 Exprimental results of the proposed system and compared method on the HCI dataset. (b), (c), (d), (e), and (f) are results of the SVM, the proposed method, Linear SVM, LS-SVM, and SVM-HOG on the input image (a).

Table 1. Experimental results using the KITTI

	KITTI Dataset		
	Precision (%)	*Recall (%)*	*F1(%)*
SVM	88.9	57.1	69.6
SVM-HOG	90.0	67.8	77.5
Linear SVM	77.7	53.8	63.6
LS-SVM	90.0	64.3	75.0
The proposed	**90.9**	**71.4**	**80.0**

Table 2. Experimental results using the CCD datasets

	CCD Dataset		
	Precision (%)	*Recall (%)*	*F1(%)*
SVM	72.2	52.0	60.5
SVM-HOG	88.9	53.3	66.7
Linear SVM	79.4	59.2	67.8
LS-SVM	88.2	50.0	63.8
The proposed	**89.6**	**62.10**	**73.4**

Table 3. Experimental results using the HCI datasets

	HCI Dataset		
	Precision (%)	*Recall (%)*	*F1(%)*
SVM	68.7	44.0	53.7
SVM-HOG	87.5	50.0	63.6
Linear SVM	76.7	50.8	61.1
LS-SVM	71.8	44.9	55.2
The proposed	**88.1**	**52.9**	**66.1**

Conclusions

- This research introduces the benefits of the proposed deep learning approach for im- proving the performance of existing vehicle detection systems. By conducting experiments under various datasets, including CCD, KITTI, and HCI, we found that the precision, recall, and f1 scores of the proposed method are better than compared method under various experimental conditions.

- However, there are still several limitations in the proposed method, such as increasing the processing time due to the pre-processing stage to generate robust features of the proposed deep convolutional neural network. The lightweight approach should be considered to reduce the complexity of the pro- posed network in the future.


Thank You!