International Journal of Engineering & Technology, 7 (4.44) (2018) 177-180

International Journal of Engineering & Technology

Website: www.sciencepubco.com/index.php/IJET

Research paper

Rear-Approaching Vehicle Detection using Frame Similarity base on Faster R-CNN

Yeunghak Lee[1]*, Israfil Ansari[2], Jaechang Shim[3]

[1]Andong National University
[2] Andong National University
[3] Andong National University
**Corresponding author E-mail: yhsjh.yi@gmail.com*

Abstract

In this paper, we propose a new algorithm to detect rear-approaching vehicle using frame structure similarity based on deep learning algorithm for use in agricultural machinery systems. The commonly used deep learning models well detect various types of vehicles and detect the shapes of vehicles from various camera angles. However, since the vehicle detection system for agricultural machinery needs to detect only a vehicle approaching from the rear, when a general deep learning model is used, a false positive is generated by a vehicle running on the opposite side (passing vehicle). In this paper, first, we use Faster R-CNN model that shows excellent accuracy rate in deep learning for vehicle detection. Second, we proposed an algorithm that uses the structural similarity and the root mean square comparison method for the region of interest(vehicles area) which is detected by Faster R-CNN between the coming vehicle and the passing vehicle. Experimental results show that the proposed method has a detection rate of 98.2% and reduced the false positive values, which is superior to general deep learning method.

Keywords: *faster r-cnn; vehicle detection; structural similarity index; deep learning; agricultural machine*.

1. Introduction

Many agricultural machinery systems are improving in performance due to the development of the industry. And the dependence of agricultural machinery on cultivating crops is increasing. The characteristics of farm machinery currently in use are that they move slower than cars. Since the use of agricultural machinery is used not only in rural areas but also in urban suburbs, many crashes and serious injuries occur every year. This paper proposes a new rear approaching vehicle detection algorithm for agricultural machine systems to prevent accidents in agricultural machinery.

Vehicle detection is used in many places such as public safety, public security, surveillance, intelligent traffic volume control, and autonomous driving. There are two kinds of vehicle detection: front view and rear view. The front view is the detection of a vehicle running in front of a traveling vehicle. The rear view is the detection of a vehicle approaching from behind (or approaching with a fixed camera). The rear approaching vehicle detection method is divided into vision based and audio (or sound) based. In an audio-based study, Chen [1] detected a rear-access car using the natural frequency analysis of the car. Ananthanarayanan [2] extracted various types of features using sound (conversation, music, wind, automobile, etc.) as input data and developed a rear approaching vehicle detection system by analyzing the characteristics. The advantage of these systems is that they are low cost to manufacture and have low computational complexity. However, due to the use of sound (frequency), areas with noises can be highly susceptible to noise, and if the intensity of the frequency is weak, it may be exposed to danger because the rear approaching vehicle id detected very closely.

Vision based vehicle detection research has focused on front-running vehicle detection for use in autonomous vehicles. There is not much research on vehicle detection (frontal) approaching a camera installed behind a vehicle. In general, vehicle detection algorithms to obtain object features has been used a scale invariant feature transform (SIFT), histogram of oriented gradients (HOG), speed up robust features (SURF), Haar cascade, and Bigausian edge detection (BED). Adaboost, Support Vector Machine (SVM), and K-Nearest Neighbor (KNN) has been used to classify vehicles and other objects. These features lack the generalization ability to detect different objects. In addition, feature extraction is influenced by complicated and various illumination changes, camera view angle, and image complexity, and so on. These bad feature extractions can reduce the object detection rate and be difficult to apply in real time because of system degradation. Vision based rear approaching vehicle detection research has been focused on vehicle detection in blind spots of running vehicles. Chen [3] has used the one - dimensional distance information and two intersection times to detect the blind area approaching vehicle. The paper [4] detected vehicles in the blind spot using cascade classification through Adaboost learning based on the modified census transformation feature vectors. In the above studies, vehicles detected as traveling and approaching vehicles with similar speed and at a short distance, so it is somewhat distant from this research.

Deep learning technology is becoming more and more popular in the field of artificial intelligence. Especially, CNN is widely used in various fields such as image recognition, speech recognition, pedestrian detection, face recognition, etc. Unlike general feature extraction systems, CNN uses raw images as inputs and extracts features through a large amount of training with high flexibility and generalization capabilities. It shows considerably higher object classification accuracy rate than traditional feature extraction method. CNN can be applied to object detection using region based R-CNN [5] model, which was greatly improved object detection

accuracy compared to existing feature based detector. The R-CNN performs Region of Interesting (RoI) cropping on the region obtained from the region proposal that there is an object. Then it uses a selective search method to find the bounding box of the most appropriate location.

The final step in R-CNN is to classify the images using a support vector machine. The classified object uses a linear regression model to find the exact bounding box position. However, the R-CNN has a drawback that it becomes computationally expensive. In other words, the problem with R-CNN is that it has to perform the CNN, classify the object using SVM, and run the linear regression for every bounding box to train. Moreover, it cannot be implemented real time as it takes around 47 seconds for each test image. Fast-RCNN [6] solved the problems of R-CNN. Fast-RCNN region extraction is performed by using RoI pooling method in which bounding box information is maintained while passing through CNN and the corresponding region is extracted from the final CNN characteristic map. However, Fast-RCNN has a bottleneck at the region proposal stage that creates a bounding box. Faster-RCNN [7] solved the problem of Fast-RCN by placing the Region Proposal Network phase in CNN. Faster-RCNN calculate the bounding box coordinates and bounding box values using the anchor boxes of various ratios / sizes from the convolutional feature map based on scaled sliding window.

R-CNN based vehicle detection research is as follows. Fan et al. [8] demonstrated the superiority of vehicle detection using the Faster R-CNN model through a comprehensive analysis of the basic structure of the model and various experiments. Hsu et al. [9] used the algorithm that it was removed unnecessary parts from detected object area in order to speed up the training process.

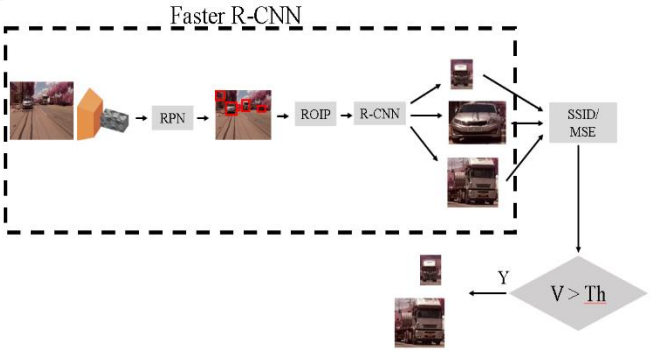


Fig. 1. The architecture of proposed algorithm.

In the study of the previously mentioned deep learning methods, the target object distance was very near to the running vehicle. It detected the various types of vehicles and vehicle shapes viewed from various camera angles. However, in this study, additional algorithms are needed because only the vehicles approaching from the rear should be detected while driving the farm machinery. In order to detect only the vehicles with rear approaching, we reduced the ratio of false positives (cars passing behind the cultivator) by using frame structural similarity index and mean squared errors. The overview of proposed algorithm is explained in Figure 1.

2. Faster R-CNN

It is more difficult to distinguish what objects are in the image than to classify them. Basically, R-CNN goes through several stages. First, R-CNN creates a region proposal or bounding box that identifies the region in which the object is possible. Bounding boxes are found using selective search algorithms. It is a way to combine adjacent pixels with similar colour, brightness, pattern, etc. The second is to force the size to be unified to use the extracted bounding box as input to CNN. The third is to classify the selected region using SVM. Finally, the precise coordinate setting of the bounding box of the classified object uses a linear regression model.

Faster R-CNN replaces region proposal generation part with computationally expensive with a new application of region proposal networks (RPN) which is integrated into the model. This method is a new application of RPN network to detect objects. The role of RPN is to output the square of object proposals and the score of objects from the input image. It is a fully connected network and is designed to share a convolutional layer with Faster R-CNN. The trained RPN improves the quality of region proposal and object detection accuracy. The Faster R-CNN uses an external slow selective search (calculated by the CPU), while the Faster R-CNN shows the speed improvement using the internal fast RPN (calculated by the GPU). The RPN is located after the last convolutional layer. After that, like Faster R-CNN, ROI pooling and classifier bounding box regress are located as shown in Figure 2.

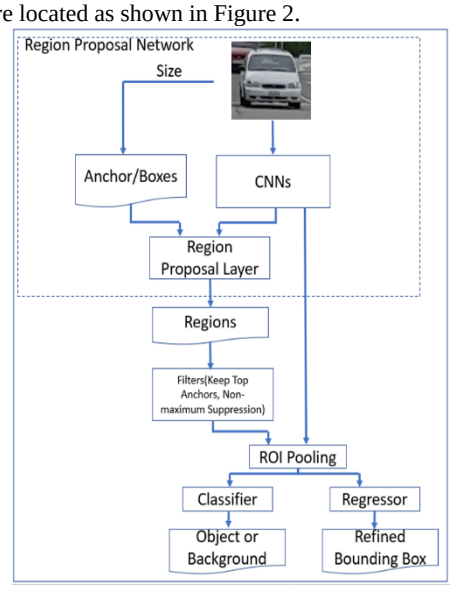


Figure 2. The Architecture of Faster R-CNN

RPN maps the input feature map to features of 256 or 512 size by applying the sliding window with a 3x3 convolution. This output is used to input to the box classifier layer and the box regress layer. Box regression uses a pre-defined reference box named anchor as an initial value, and is a box used as a candidate for bounding box at each position of the sliding window. This extracts the feature by applying anchor boxes of pre-defined various ratio / size using the center position of moved the sliding window with same size. It is used nine anchor boxes (three sizes and three ratios) are used. The anchor box is used as a candidate for the bounding box at each position of the sliding window.

The labelled data is processed using the Faster R-CNN model described above, and the .meta files are generated as learning results. The final inference graph file (".pb" file) is created using ".meta" files obtained previous stage. Finally, the object detection result (bounding boxes and object scores) using the ".pb" file is displayed on the monitor for the input image.

3. Structural Similarity (SSIM) Index

The SSIM index [10] is a method to measure the similarity to the original image with respect to the distortion caused by the compression and the transformation, and it makes a more accurate comparison than the Mean Square Error (MSE) and Peak Signal Noise Ratio (PSNR) methods. This is the evaluation of the test image X for the reference image Y to quantify the visual similarity. A value closer to 1.0 means that the test image is similar from the reference image. A value closer to 0.0 means that the test image is different from the reference image. The SSIM formula is defined as follows.

$$l(x,y) = (2\mu_x\mu_y + C1)/(\mu_x^2 + \mu_y^2 + C1) \quad (1)$$

$$c(x,y) = (2\sigma_x\sigma_y + C2)/(\sigma_x^2 + \sigma_y^2 + C2) \quad (2)$$

$$r(x,y) = (\sigma_{xy} + C3)/(\sigma_x\sigma_y + C3) \quad (3)$$

where, μ_x and μ_y are the average pixel values, σ_x and σ_y are the standard deviation at patches, and σ_{xy} is the covariance value of x and y. *C1*, *C2*, and *C3* are constant values to avoid instabilities when $\mu_x^2 + \mu_y^2$ and $\sigma_x^2 + \sigma_y^2$ or $\sigma_x\sigma_y$ is close to zero. *l(x, y)* is the relation of luminance difference, *c(x, y)* is the contrast difference, and *r(x, y)* is the structure variations between x and y. the general form of SSIM index can be expressed as (4).

$$\text{SSIM}(x,y) = [l(X,Y)]^{\alpha}[c(X,Y)]^{\beta}[r(X,Y)]^{\gamma} \quad (4)$$

where α, β, γ define the relative importance of each component, and the values are used as 1.0 in the experiment.

4. Experimental Results

This paper proposed a new rear-approach vehicle detection algorithm using the deep learning and frame similarities for vehicle detection systems to protect cultivator drivers. In this study, an experiment was carried out with an ordinary user computer environment consisting of an Intel Core i7-7700 (3.5 GHz), memory 16G, Geforce TITAN-X and OpenCV and Python 3.5 program. The database for this study used our own videos that was recorded directly from the street (vehicles visible to over 100 meters). The used recording device is RaspberryPi and NoIR camera. The software used was basically Python, Tensorflow and Opencv. The used total images for the raining were 3330 images obtained from different road places.

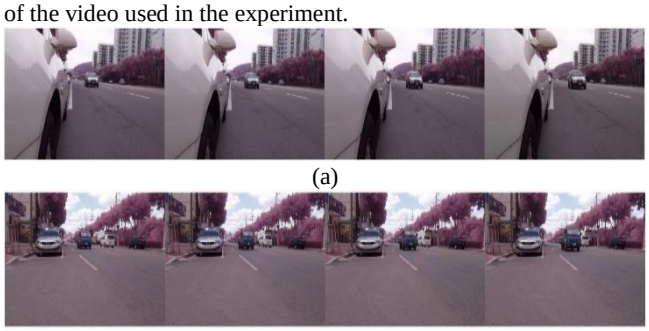
In order to implement the proposed algorithm, the following process has been performed. The first is labeling dataset. The labeling of the vehicle in the image was carried out by using the LabelImg program. The results are shown in Figure 3. This paper has used a variety of vehicles such as passenger cars, SUVs, vans, buses (medium and large), and trucks (small, medium and large) for training process. The labeling result is stored in the .xml file with the four-point coordinates of each rectangle along with the image name.

Fig. 3. Labeling dataset

Second, it is data training. Labeled .xml files must be converted to the tensor flow training data format. In the training process, the input image is saved as a JPEG or PNG file. Since the meta data and labels for these images are stored in separate files, the code becomes complicated because it is necessary to read the image file separately from the meta data or label file when reading the training data. Also, if the image is read in the JPG or PNG format and decoded every time, the performance is degraded. In addition, there is a lot of performance degradation in the data reading part in the learning stage. We used the TFRecord file format to prevent the above performance degradation and to make development easier. The TFRecord file format stores the height and width of the image, the file name, the encoding format, the image binary, the position of the rectangle indicating the position of the object in the image, and the label value. Through this process, the entire learning data is stored in a tensor flow format divided into 70% training and 30% validation. The training uses 70% training data and 30% validation data converted into TFRecord, and this paper selected the Faster R-CNN ResNet (Deep Residual Network) as the basic model. Because it is only necessary to classify the vehicle, the category was set to 1 and the learning steps were 20,000.

The third is the export of the trained model. The learning process stores a check pointer indicating the learning result for each predetermined pointer. The checkpoint file that is a Tensorflow model file format has meta information about the model and can be retrained. However, because the meta files have lot of unnecessary information, it has to be improved for using the actual model. Finally, a ".pb" file is created that combines the model and weight values, except for meta data.

In this paper, experimental data used the video record file recorded in a place without a median strip. Figure 3 shows a frame example of the video used in the experiment.

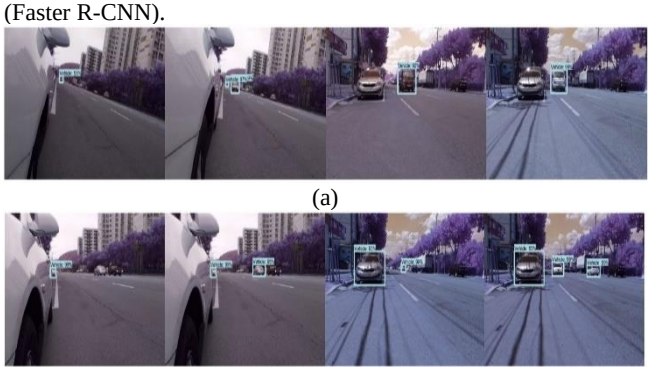


(a)

(b)

Fig. 4. Example of the frames of test videos, (a) video1 and (b) video2.

A video experiment was conducted using the finalized ".pb" file based on the Faster R-CNN model. Figure 5 and Table 1 show the results of a video experiment using general deep learning algorithm (Faster R-CNN).



(a)

(b)

Fig. 5. The frame sequence result of the Faster R-CNN, (a) the result of true positive, (b) the result of false positive.

Table 1. Faster R-CNN results for video images (Frames)

Input Video	Ground Truth	True Positive	False Negative
Video1	4468	4463	65
Video2	50052	45549	4503

From Table 1, precision for Video 1 is 98.5% and precision for Video 2 is 91.0%. The calculation is shown in Equation (5). The Ground Truth represents the number of frames of the entire image in which the vehicle is detected. The True Positive (TP) indicates a case of detecting a rear approaching vehicle, and the False Positive (FP) indicates a case of detecting a vehicle moving away from the camera.

$$\text{Precision} = \text{TP} / (\text{TP} + \text{FP}) \quad (5)$$

General deep learning method has many false positives as well as true positives, as shown in Fig. 4 (b). This is because it takes into account the various angles of the shape when extracting features of objects in the learning process. If the mounted system on the cultivator frequently generates warning lights and alarm sounds, the cultivator driver may be disturbed. Therefore, such a false alarm should be eliminated. The object to be removed is a parked vehicle (similar to rear approaching vehicle) and a vehicle passing through the rear camera (rear-view vehicle). In this paper, we compared

region of interest for the frame images using the SSIM index and the mean square error, and the bounding box area of the deep learning result to remove the false alarm. The parked vehicle presents a high value in frame similarity comparisons, and a passing vehicle will show a very low value of similarity because of smaller vehicle. First, if a vehicle is detected through deep learning, five consecutive frames are stored in the array. The same size area as the detected vehicle area (bounding box area) is stored in f_b, f_c, and f_f from the first, third, and fifth frame, respectively. Then, we calculate the similarity using SSIM index and the mean squared error (MSE) to determine whether the vehicle is finally detected.

$$S_k = \text{ssim}(f_i, f_j) \quad avg_{s1,s2} = (S_1 + S_2)/2 \tag{6}$$

$$M_k = \text{MSE}(f_i, f_j) \tag{7}$$

$$\text{V}_S = \begin{cases} 1, \ if \ avg_{s1,s2} > S_3, \ S_3 < 0.5 \\ 0, \ else \end{cases} \tag{8}$$

$$\text{V}_M = \begin{cases} 1, \ if \ M_{1,2,3} > Th \\ 0, \ else \end{cases} \tag{9}$$

$$V = \begin{cases} 1, \ if \ V_S = 1 \ and \ V_M = 1 \\ 0, \ else \end{cases} \tag{10}$$

where k means 1,2, and 3, i and j has different symbol for the b, c, and f. S_k is the result of structural similarity index value and M_k is root mean square error value for the two frames. If V is 1 in (10), the finally detected vehicle appears on the screen. Table 2 and Figure 6 show the results of applying the proposed algorithm.

Table 2. Proposed algorithm results for video images (Frames)

Input Video	Ground Truth	True Positive	False Negative
Video1	1779	1765	14
Video2	3960	3853	107

(a)

(b)

Fig. 6. The result of adapted proposal algorithm using video files, (a) good result, (b) bad result

The proposed algorithm removes a parked vehicle and a vehicle passing through the rear camera moving in opposite directions. From Table 2, the accuracy of video 1 is 99.2% and the accuracy of video 2 is 97.3%. Moreover, Figure 5 (a) and (b) show that the proposed algorithm removes many passing vehicle and parked vehicle. Table 2 shows that the Ground Truth is different because all frames containing false positives have been removed through the proposed algorithm. Figure 6 (a) shows the good results for the vehicle detection without a mistake. On the other hand, Figure 6 (b) shows that some of the false positives have been removed, but there are still false positives (red circle). From the experimental results, we was able to learn that the proposed algorithm is significantly reduced the number of false positives, as shown in Table 2.

5. Conclusion

This paper describes a new rear-approaching vehicle detection algorithm that is used in agricultural machine driver protection system. The accident of an agricultural machine and a vehicle collision is easy to become a slander for all the passengers. Vehicle detection of general methods extracts artificial features from images for special purposes, so it is affected by image blindness, camera view angle, weather, and so on. To solve this problem, we are currently studying deep learning methods. Deep learning detects not only the rear approaching vehicle but also the passing vehicle and the stopped vehicle, so there is a lot of unnecessary vehicle detection. In this paper, we remove unnecessary vehicles (vehicles that pass through the camera and parked or stopped vehicles) using the similarity and mean square error method of neighbouring frames to the vehicle image region detected by Faster R-CNN. Experimental results show that the accuracy of the proposed algorithm is 98.2% higher than that of the conventional deep learning method, and that false positive images are significantly reduced. As a future work, it is necessary to compare with other models of deep learning and experiments using correlation of frame motion and frame to reduce false positives.

Acknowledgement

This work was carried out with the support of "Cooperative Research Program for Agriculture Science and Technology Development (Project No. PJ01384901)" Rural Development Administration, Republic of Korea.

References

[1] Chen C, Rear Approaching Vehicle Detection with Microphone, Bachelor's Thesis, Halmstad University, (2013).
[2] Ananthanarayanan VK, Audio Based Detection of Rear Approaching Vehicle on a Bicycle, Graduate School Thesis, Rutgers University, (2012).
[3] Chen CT and Chen YS, Real-time approaching vehicle detection in blind-spot area, *12th Internal IEEE Conference on intelligent Transportation Systems*, 2009.
[4] Kang HW Baek JW, and Jeong YS, Real-Time Side-Rear Vehicle Detection Algorithm for Blind Spot Warning Systems, *KIISE Transactions on Computing Practice*, V.23, No.7, (2018), pp. 408-416.
[5] Dobahue J, Girshick R, Darrell T, and Malik J, Rich feature hierarchies for accurate object detection and semantic segmentation, *IEEE Internal Conference on Computer Vision and Pattern Recognition*, (2014), pp:580-587.
[6] Ross Girshick, Faster-RCNN, *2015 IEEE International Conference on Computer Vision*, (2015), pp:1440-1448.
[7] Ren S, He K, Gisshick R, and Sun J, Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks, *IEEE Transaction on Pattern Analysis and Machine Intelligence*, Vol. 38, No. 6, (2017), pp. 1137-1149.
[8] Quanfu F, Lisa B, and Hohn S, A Closer Look at Faster R-CNN for Vehicle Detection, *2016 Intelligent Vehicle Symposium*, (2016), pp:124-129.
[9] Hsu SC, Huang CL, Chuang CH, Vehicle Detection using simplified Fast R-CNN, *International Workshop on Advanced Image Technology*, (2018).
[10] Kim HS and Park JS, intensity-based efficient Video Quality Assessment for Variable bitrate Streaming, *Korean Institute of Next Generation Computing*, Vol.11, No.5, (2015), pp.63-71.