Vehicular Detection and Classification for Intelligent Transportation System: A Deep Learning Approach Using Faster R-CNN Model

Maria Crystal E. Orozco, Corazon B. Rebong
Technological Institute of the Philippines, Philippines.
Email: orozco.mariacrystal@gmail.com; corazonbrebong@gmail.com

Abstract - Intelligent Transportation System (ITS) is one of the attributes that describe smart cities. One of its functions is detection and classification of vehicles that pass through roadways. With this information, traffic management sectors can plan and implement road rules for the betterment of the traffic flow. Vision-based approaches and other methods, however, work only in ideal environment which make researchers find new ways on how limitations like occlusions, nighttime and camera angle can be solved. This paper demonstrates using a deep learning method to accurately detect and classify vehicles on urban roadways in a certain city. Additionally, a vehicle classifier was built and tested using a machine learning framework known as TensorFlow. Faster R-CNN model, with captured CCTV-video as dataset, was used to train the vehicle classifier. The performance of the newly-trained classifier has been evaluated using different classification metrics. Results show that using the proposed method, 93% accuracy and 78% F1-score in detecting and classifying vehicles were achieved based on labeled data. However, researchers also took note of the detection errors that showed during testing. Configurations in some steps has been provided to minimize such misclassifications. It was also recommended that the method be integrated as vital part of Intelligent Transportation Systems (ITS) in terms of vehicle detection and classification for future smart cities.

Keywords - vehicle detection and classification, deep learning, Faster R-CNN, TensorFlow, ITS

I. INTRODUCTION

Traffic monitoring has evolved with the use of technologies such as CCTV camera systems that help traffic enforcers monitor and control highways and major roads in different cities every day. [1] This leads to the development of Intelligent Transportation System (ITS) that includes smart monitoring of traffic flow through vehicle detection and classification.

Currently, machine learning applications, such as Support Vector Machines or SVM and Artificial Neural Network or ANN have been implemented for vehicle classification, however, low vehicle type recognition rates and camera perspectives are some of the limitations. [2] Meanwhile, the development of (Convolutional Neural Network) CNN has a fairly high vehicle type recognition rate of 95.7% and is able to cope with spatial invariance and changing camera angles. [3] [4] [5]

Transforming standard CCTV systems into intelligent and effective surveillance tool is the idea of the study. It consists of different steps which includes vehicle detection and classification to aid in decision support of City Traffic Management Office.

This paper presents an application of deep learning method employing a region-based convolutional neural network model (R-CNN) in vehicle detection and classification as an element of intelligent transportation system in a certain city of the Philippines. The objectives are to (1) detect vehicles from CCTV-captured videos by training and building vehicle classifier (2) classify the detected vehicles to which class it belongs; and (3) evaluate the performance of the vehicle detector / classifier by using classification metrics.

II. LITERATURE REVIEW

A. Deep Learning

Deep learning is a subset of machine learning that teaches computers to learn by example. It has network architectures that has many layers which help train models by using sets of labelled data. In deep learning, datasets of images, sound and text are fed into the computer to perform classification tasks through supervised training. Deep learning models can achieve state-of-the-art accuracy. [6] Figure 1 illustrates the relationship between artificial intelligence, machine learning, and deep learning. [7]
A standout amongst the most prevalent types of profound neural networks is known as convolutional neural networks (CNN or ConvNet). A CNN convolves learned features with input data, and utilizes 2D convolutional layers, making this architecture appropriate in handling 2D information, like pictures. CNNs take out the requirement for manual feature extraction. Training a collection of images and learning the relevant features works simultaneously. This automated feature extraction makes deep learning models highly accurate for computer vision tasks such as object classification. [6]

One type of CNN is R-CNN which uses a selective search algorithm to extract regions from image called region proposals. Eventually, R-CNN has evolved until the launch of Faster R-CNN which proves to be used on real-time object detection. It eliminates the time-consuming selective search algorithm and uses a separate network to predict the region proposals. The predicted region proposals are then reshaped using a RoI pooling layer which is then used to classify the image within the proposed region and predict the offset values for the bounding boxes. [8].

B. Tensorflow Framework

Machine learning is a multifaceted discipline. But applying machine learning models is far less overwhelming and challenging than it used to be, because of machine learning frameworks such as Google’s TensorFlow that ease the process of acquiring data, training models, serving predictions, and refining future results. [9] It is an open source software library for high performance numerical computation. [10].

C. Related Studies

There are a number of studies related to vehicle detection and classification that have utilized different methods based from different purposes.

Most researches have utilized image processing and vision-based vehicle detection and tracking. [11] [12] [13] These techniques work well in ideal environment conditions such as daylight video captures, top angle view of CCTVs, and noiseless background. On the other hand, the detection accuracy and vehicle classification may be affected by object occlusions, nighttime captures and busy or noisy background.

Meanwhile, deep learning approaches that are heavily based on neural networks have been of special interest during the past years. Experimental results have shown that deep detectors are among the best performing modern models. [14][15]

Additionally, most of the previous works succeeded deal with highways or freeways rather than urban environments. The main technical challenge from the application perspective lies in the camera view and operating condition, which impose many additional limitations. [16]

III. METHODOLOGY

In this research, a deep learning method was used to automatically detect and classify vehicles in video captured by CCTVs in four (4) designated roadways in a certain city. The summary of steps are as follows:

1. Capture CCTV traffic videos
2. Convert videos to images using FreeStudio’s Video to JPG converter tool
3. Label vehicle objects in the images using labelImg software
4. Configure training by modifying Tensorflow’s object detection API’s python scripts
5. Run training using the newly-created Tensorflow virtual environment and Tensorboard
6. Test the vehicle detector/classifier by running the object detection API’s python script
7. Evaluate the vehicle detector/classifier using accuracy, precision, recall and F1-score performance metrics

A. Step One: Capture videos

Videos captured by the CCTV were collected from the command center. It consisted of traffic scenes from four (4) designated areas in the city namely: (1) Almazora; (2) Mamplasan; (3) Olivarez; and (4) Platero. The properties of the video files are presented in Table 1. Videos 1 and 2 have the largest sizes because of their long durations of nearly 10 minutes each, the video quality of 1280 x 960 resolution and frame rate of 15 frames per second. Videos 3 and 4 has duration of almost 5 minutes, video resolution of 1280 x 720 and 10 frames per seconds. In contrast, the testing videos 5, 6 and 7 has exactly 3-minutes duration, 1280 x 720 video resolution and 10 fps. Also, day and night video captures per location were collected so as to test vehicle detection based on illumination.

B. Step Two: Convert videos to images

Using a video converter tool, FreeStudio’s Video to JPG converter, the input videos 1, 2, 3 and 4 were converted to 50 frames each. A total of 200 image files were used to train the classifier while videos 5, 6 and 7 were used as test videos after the vehicle classifier has been built. The image files’ sizes are less than 3KB and resolutions of not more than 1706 x 960.

<table>
<thead>
<tr>
<th>Video No</th>
<th>Video Location</th>
<th>Duration (S)</th>
<th>Size (Kb)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Almazora Day</td>
<td>498</td>
<td>91815</td>
</tr>
<tr>
<td>2</td>
<td>Almazora Night</td>
<td>545</td>
<td>70112</td>
</tr>
<tr>
<td>3</td>
<td>Olivarez Day</td>
<td>258</td>
<td>35122</td>
</tr>
<tr>
<td>4</td>
<td>Olivarez Night</td>
<td>228</td>
<td>30830</td>
</tr>
<tr>
<td>5</td>
<td>Mamplasan Day</td>
<td>180</td>
<td>34904</td>
</tr>
<tr>
<td>6</td>
<td>Mamplasan Night</td>
<td>180</td>
<td>15546</td>
</tr>
<tr>
<td>7</td>
<td>Platero</td>
<td>180</td>
<td>36551</td>
</tr>
</tbody>
</table>

TABLE I. ATTRIBUTES OF VIDEO DATA
C. Step Three: Label vehicle objects in images

Each image file contained different classes of vehicles. Figure 2 shows a sample image of video 1. Notice that a single frame contains a number of vehicle objects. Also notice that this is an example of a crowded traffic scene where other objects such as people, road signs, and pavements are present. The researchers have identified eight (8) vehicle classes namely: bus, bicycle, car, jeep, motorcycle, truck, and van that pass on the roadways. The 200 images have been divided into two (2) with 80% of them was put in a train folder, and remaining 20% in the test folder.

![Figure 2. A sample frame in Video 1](image)

It took roughly four (4) hours to finish the labelling process. This is a meticulous step but is required in training the model and in creating an object detection model eventually. Figure 3 shows the labelling process using labelImg software. The red box represents the currently selected vehicle for labelling. The vehicles bounded by green boxes are already labelled objects. The Box Labels on the right indicates the classes of vehicles labelled which include six (6) car classes, four (4) jeep classes, and one (1) labelled as tricycle.

![Figure 3. Labelling of jeep vehicle](image)

The labelling process created xml files that contain the coordinates of the bounding boxes of the labelled objects in each image file. A python script is run to convert these xml files to csv files namely: (1) train labels.csv and (2) test labels.csv as shown in Figure 4. The blue-bordered area of the figure indicates that the image file Almazora.jpg has thirteen (13) vehicle objects labelled as evident in the number of its instances in the list. The figure also presents the dimension of the image as well as the classes of labelled objects and more importantly, the bounding box coordinates.

![Figure 4 Partial Contents of the .csv file](image)

Using the csv file, the number of labeled objects were summarized using the COUNTA Excel built-in function. Table 2 presents the number of objects per class. It also shows that cars, jeep, and motorcycles are the most classes of vehicles that pass through the city’s roadways. In summary, a total of 552 vehicles for training and 174 for testing, are labelled.

<table>
<thead>
<tr>
<th>Vehicle Class</th>
<th>Number of Labeled Objects for Training</th>
<th>Number of Labeled Objects for Testing</th>
</tr>
</thead>
<tbody>
<tr>
<td>bicycle</td>
<td>24</td>
<td>5</td>
</tr>
<tr>
<td>bus</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>car</td>
<td>140</td>
<td>33</td>
</tr>
<tr>
<td>jeep</td>
<td>129</td>
<td>35</td>
</tr>
<tr>
<td>motorcycle</td>
<td>128</td>
<td>44</td>
</tr>
<tr>
<td>tricycle</td>
<td>84</td>
<td>42</td>
</tr>
<tr>
<td>truck</td>
<td>27</td>
<td>7</td>
</tr>
<tr>
<td>van</td>
<td>18</td>
<td>5</td>
</tr>
<tr>
<td>Total</td>
<td>552</td>
<td>174</td>
</tr>
</tbody>
</table>

D. Step Four: Configure Training

Before training, some configurations have to be set which is as follows: (1) Generate tfrecord files. These files contain train and test records that will be used to train the vehicle detector; (2) Create label map which will define the object classes and corresponding ids; and (3) Define the model to be used which is Faster Region-based.
Convolutional Neural Network and (4) Add file paths to training data, test data and label maps directories.

**E. Step Five: Run the Training**

The researchers utilized the TensorFlow machine learning framework to train the classifier. With the help of a trainable model, Faster R-CNN Inception v2 COCO [17], a vehicle detection classifier was built. Figure 5 shows the Anaconda prompt window during training. Notice the loss value [18] that started at 4.7942. Training the classifier kept going until the loss was consistently 0.05. [19] If the model’s prediction is perfect, the loss is zero; otherwise, the loss is greater. [13]

![Figure 5. The Anaconda Prompt Window](image)

Training was done using a laptop computer with the following specifications:

- 4GB DDR3 L Memory
- Intel Core i7 (8th gen)
- NVIDIA GeForce MX150 GPU
- 2TB HDD
- Windows 10 OS

After training, the inference graph was generated by issuing a python command. It contains the newly-trained vehicle detection classifier that can categorize the detected vehicles into (8) eight classes.

**F. Step Six: Test the Classifier**

The newly-trained vehicle detector and classifier was tested by modifying some variables like the number of classes and video filenames in the python script and running it. Additionally, the threshold has been tuned to 80%. It means that the detection score should be 80% and above for the bounding boxes to be displayed in the image or video. This threshold value also affects the precision and recall metrics. [20].

**G. Step Seven: Evaluate the Classifier**

The detection and classification of vehicles has four possible outcomes: A true positive (TP) is correctly identified predictions. Similarly, a true negative (TN) is correctly rejected predictions. A false positive (FP) is incorrectly identified class while a false negative (FN) is incorrectly rejected class. [21] The performance of the newly-trained classifier was evaluated using the following
metrics: Accuracy is the number of correct predictions over total number of predictions.

\[
\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}
\]

Given a class, Recall metric answers the question as to what proportion of actual positives was identified correctly. On the other hand, Precision answers how likely the classification be correct.

\[
\text{Recall} = \frac{TP}{TP + FN}
\]

\[
\text{Precision} = \frac{TP}{TP + FP}
\]

F1 Score = \(2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}\)

Meanwhile, F1 score is a more appropriate metric for class-imbalanced data set where there is uneven distribution between positive and negative predictions as shown in the confusion matrix (see Figure 8). Class-imbalanced data set means that each class has different number of classifier predictions (e.g. the classifier has predicted 80 jeeps and 2 buses). In a class-balanced dataset, each class should have the same number of predictions. Moreover, TP, TN, FP, and FN were determined using this matrix. As an example, the yellow cell in the bicycle class, is true positive (TP) prediction while blue, red and green are TN, FP, and FN predictions respectively.

\[
\begin{array}{cccccccc}
\text{Actual Values} & \text{BICYCLE} & \text{BUS} & \text{CAR} & \text{JEEP} & \text{MOTORCYCLE} & \text{TRICYCLE} & \text{TRUCK} & \text{VAN} & \text{TOTAL NUMBER OF PREDICTIONS} \\
\hline
\text{BICYCLE} & 6 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 7 \\
\text{BUS} & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 2 \\
\text{CAR} & 0 & 0 & 64 & 10 & 0 & 2 & 0 & 2 & 78 \\
\text{JEEP} & 0 & 0 & 3 & 68 & 0 & 7 & 0 & 2 & 80 \\
\text{MOTOR CYCLE} & 2 & 0 & 0 & 52 & 1 & 0 & 0 & 0 & 55 \\
\text{TRICYCLE} & 0 & 0 & 3 & 2 & 67 & 0 & 0 & 0 & 72 \\
\text{TRUCK} & 0 & 0 & 2 & 0 & 0 & 5 & 0 & 0 & 8 \\
\text{VAN} & 0 & 0 & 6 & 8 & 0 & 0 & 6 & 0 & 20 \\
\end{array}
\]

Figure 8 Confusion Matrix of Vehicle Classes

IV. RESULTS AND DISCUSSION

The newly-trained vehicle detector and classifier was tested in three CCTV videos: 5-Mamplasan Day, 6-Mamplasan Night and 7-Platero Day.

Occlusion, which is the limitation of previous studies is not a problem anymore. Figure 9 shows detected vehicles even if it is occluded by another detected vehicle. On the left side of the figure, the jeep vehicle is being occluded by the tricycle. More vehicles, such as the cars and jeeps are occluded with each other at the far middle of the figure. Figure 10 shows the 99% correctly classified vehicles bounded by yellow green boxes at night in the Mamplasan video. Notice that even far away vehicles in Figure 11 have been detected using the deep learning method. The tricycle in the far middle of the image has been correctly classified by 98%.
Vehicle detection errors includes (1) vehicles not detected because of small size; (2) object merge wherein two objects are confined within a single bounding box; (3) object splits wherein a single vehicle has two bounding boxes; and (4) misclassification of the detected vehicle, see Figure 12.

Additionally, the performance metrics evaluated the newly-trained vehicle detector and classifier. A total of 322 vehicles have been positively detected by the vehicle classifier from the given dataset. Table 3 presents the four outcomes of detections and its accuracy per class. Interestingly, the classifier correctly predicted that there were six (6) bicycles and 313 non-bicycles out of the 322 vehicles in the video. On the other hand, the classifier has wrong predictions too. Going back to the bicycle class, the classifier has mistakenly identified the two (2) motorcycles as bicycles while one (1) bicycle has been incorrectly predicted as non-bicycle.

Classifying the motorcycle has the highest F1 score (95%) which means that every vehicle detected as motorcycle, 95% (precision) is an actual motorcycle and that every actual motorcycle, 95% (recall) is classified as motorcycle. In contrast, the van class has the lowest F1 score of 40% which means that every 100 vehicles classified as vans, only 60 (precision) are actual vans and that every 100 actual vans in the traffic scene, only 30 are classified correctly. Overall, the F1 score of 78% indicates a good performance of the classifier.

V. CONCLUSION AND RECOMMENDATIONS

This study showed that deep learning methods, can perform high accuracy and F1-score in vehicle detection and classification. Moreover, the vehicle detection accuracy was not greatly affected by vehicle occlusions, nighttime captures, camera angles and noisy background unlike in previous studies where these issues were highly taken into consideration.

In terms of classification, most of the classes had been identified correctly and the likelihood of correct prediction was high which can be seen in their recall and precision scores, 73% and 85% respectively. Meanwhile, there were still misdetections that could be caused by the labelling process or setting of threshold. Based on the F1-score result, the classifier was having a hard time predicting examples of van class because its features have similarity with the car and jeep classes. Sports utility vehicles had been labeled as car and the length of jeep confused the classifier in identifying vans.
The overall accuracy of 96%, which confirmed research studies that uses convolutional neural networks (CNN), and F1 score of 78% were evidences that deep learning methods which uses the Faster R-CNN can level with existing implementations of ANN and SVM in terms of vehicle detection and classification.

**Recommendations:** Based on the conclusions stated, the following recommendations are suggested:

- In machine learning, preparing the train and test data and labelling the objects in images are critical steps because these are input from which the machine will learn. It is recommended to label the objects, specifically those classes which has low F1 score such as van, bus, and truck, in such a way that the classifier can distinguish between their features.

- The threshold can also be changed depending on the problem that will be solved.

- The Tensor-flow object detection API comes with different pre-trained models aside from the Faster R-CNN. Future research works can select other pre-trained models such as R-FCN, and SSD.

- Thus, the methodology is recommended to be integrated as vital part of Intelligent Transportation Systems (ITS) in terms of vehicle detection and classification for future smart cities.

**REFERENCES**


