Automatic vehicle classification using fast neural network and classical neural network for traffic monitoring

Mahammad Abdul HANNAN*, Chew Teik GEE, Mohammad Saleh JAVADI
Department of Electrical, Electronic, and Systems Engineering, Faculty of Engineering and Built Environment, Universiti Kebangsaan Malaysia, Bangi Selangor, Malaysia

Received: 10.11.2012 ● Accepted/Published Online: 05.03.2013 ● Printed: 31.12.2015

Abstract: This paper introduces an automatic vehicle classification for traffic monitoring using image processing. In this technique the fast neural network (FNN) as a primary classifier and then the classical neural network (CNN) as a final classifier are applied to achieve high classification performance. The FNN gains a useful correlation between the input and the weighted neurons using a multilayer perceptron to provide detection with a high level of accuracy. The Fourier transform is used to speed up the procedure. In the CNN, a lighting normalization method is employed to reduce the effect of variations in illumination. The combination of the FNN and CNN is used to verify and classify the vehicle regions. False detection is added to the training procedure using a bootstrap algorithm to get nonvehicle images. Experimental results demonstrate that the proposed system performs accurately with a low false positive rate in both simple and complex scenarios in detecting vehicles in comparison with previous vehicle classification systems.

Key words: Automatic vehicle classification, classical neural network, fast Fourier transform, fast neural network

1. Introduction

Today the need to understand the number and type of vehicles on roadways is becoming more important in order to record vehicular traffic data [1,2]. Automatic vehicle classification techniques have been considered widely since many real-world applications can be accomplished based on these techniques, such as motorway surveillance systems, toll collection, fare collection, breakdown roadside services, etc.

Various methods have been proposed for vehicle classification, some of which were successful in constrained scenarios, but to accomplish this task completely there are still some challenges along the way, such as changes in illumination, image scale, image quality, size, and color [3,4]. In this study, vehicle classification techniques based on computational intelligence methods such as genetic algorithms and neural networks have been investigated. Neural networks are able to learn and adapt complex nonlinear patterns and complex problems can be optimized using genetic algorithms. Various surveillance systems have employed these techniques.

The paper is organized as follows: an overview of the methods for vehicle classification is provided in Section 2, and then Section 3 emphasizes the proposed method. The result is presented in Section 4, followed by the conclusion in Section 5.

2. Related works

Vehicle classification is an important research area in the automobile industry. To provide safety and comfort to road users, it is necessary to increase the quality of the traffic surveillance system [5,6]. A reliable vehicle
classification in the traffic surveillance system can be regarded as an International Space Station in automobile applications [7].

There are several imaging challenges to robust vehicle detection and classification, such as lighting variations, vehicle variations, image apertures, corresponding regions, and finding a narrow band of electromagnetic spectrum [8]. Different lighting conditions will create great intensity variations in the images that may cause loss of information such as blurred images, loss of color, and so on [9,10]. An intuitive vehicle variation problem is that the vehicle can be of different sizes, colors, and heights [11,12]. The aperture problem implies the ambiguity of one-dimensional motion of a simple striped pattern viewed through an aperture that is indistinguishable in different physical motions [13].

Over the past 20 years, numerous approaches to image processing have been introduced for object and vehicle detection based on the image shape, color, depth, texture information, or fusion of these data in still and moving images [14,15]. Fuzzy neural network color image vehicular detection is an approach introduced for vehicle detection by adjusting background differencing methods [16]. The theory of the video detector is to photograph the traffic flow scenes through field cameras and then use them in the image processing techniques and vehicle detection algorithm to collect the parameters of the traffic. However, the system is costly due to the installation of the camera on-site and the connection between the cameras to the workstation as shown in Figure 1.

Another system based on range imagery is also proposed for vehicle classification. This method is suitable for real-time applications using range sensors and image processing techniques. Range sensors are not sensitive to change in illumination and also they are more robust to environmental conditions such as rain and fog in comparison to conventional video sensors [17].

Shaoqing et al. [18] proposed a real-time approach for vehicle classification on multilane roads with high traffic flow. This method is able to categorize vehicles as cars, buses, and trucks. They used three cameras mounted with the angle of 60° facing down towards the road to monitor two lanes. Two of the cameras focus on each lane to capture the license plate of vehicles, and the third one snaps a wide view of the two lanes and the vehicle features are obtained from it.

Neural networks, expectation-maximization, 3D vision technology, and adaptive fusion are other common types of algorithms that have been proposed for vehicle detection and classification. These algorithms and their applications are discussed in Table 1.
Table 1. Types of algorithm and approaches for vehicle detection and classification.

<table>
<thead>
<tr>
<th>Types of algorithm</th>
<th>Details of algorithm approach</th>
<th>Proposed by</th>
</tr>
</thead>
<tbody>
<tr>
<td>Neural network</td>
<td>Image decomposition, convolution neural architecture, polynomial neural network with MLP, and fuzzy neural network are used for object/vehicle detection</td>
<td>[16,19]</td>
</tr>
<tr>
<td>Expectation-maximization algorithm</td>
<td>Expectation-maximization algorithm for vehicle detection and classification</td>
<td>[20]</td>
</tr>
<tr>
<td>3D vision technology</td>
<td>3D-vision technology for occupant detection and classification</td>
<td>[21,22]</td>
</tr>
<tr>
<td>Adaptive fusion algorithm</td>
<td>Moving object is detected by using adaptive fusion algorithm</td>
<td>[23]</td>
</tr>
</tbody>
</table>

3. Methodology

In this section the vehicle detection and classification system structure are explained. The system employed a combination of two different neural networks architectures, the fast neural network (FNN) and classical neural network (CNN), to perform vehicle or nonvehicle classification [15]. The FNN extracts any positive detection and also the false detections. The FNN output is then sent to the CNN to verify the detected regions, as shown in Figure 2.

As explained earlier, the variation of lighting conditions in the FNN stage can cause false detection. To solve this problem, a linear function is used to adjust the intensity values using lighting normalization or histogram equalization. According to the convolution theorem, the result of $x$ with $y$ convolution and the result of $X$ and $Y$ Fourier transformation in the frequency domain are the same. The inverse Fourier transform can transform the point by point multiple of $X$ and $Y$ in the frequency domain into the spatial domain. As the cross-correlation is the product of the frequency domain, it speeds up the detection process.

Upon detection, the neural network is fed with a sliding $m \times n$ subimage $I$ extracted from the $S \times T$ tested image. The multilayer perceptron (MLP) structure for vehicle and nonvehicle detection is shown in Figure 3.

Assume that $w_i$ is the vector that represents weights between the subimage input and the hidden layers. This vector with $mn$ size can be shown as an $m \times n$ matrix, and $h_i$ is the hidden neurons’ output computed in a 2D space.

$$h_i = g \left[ \sum_{j=1}^{m} \sum_{k=1}^{n} w_i(x, y)I(x, y) + b_i \right]$$  \hspace{1cm} (1)

In Eq. (1) the output of each hidden neuron of a particular subimage $I$ is represented. It can be calculated for the whole image $z$ as follows.

$$h_i(u, v) = g \left[ \sum_{j=-w/2}^{w/2} \sum_{k=-\pi/2}^{\pi/2} w_i(x, y)z(v + x, u + y) + b_i \right]$$  \hspace{1cm} (2)

A cross-correlation procedure is presented in Eq. (2). The cross-correlation of any two functions $f(x, y)$ and $d(x, y)$ can be calculated by the following.
Figure 2. The neural network algorithm used for occupancy detection.

Figure 3. The MLP structure for vehicle and nonvehicle detection.
Thus, we can rewrite Eq. (2) as follows.

\[ h_i(u, v) = g \left[ w_i \otimes z + b_i \right] \] (4)

In terms of the Fourier transform, the aforementioned cross-correlation is described as follows.

\[ z \otimes w_i = F^{-1} \left( F(z) \cdot F(w_i) \right) \] (5)

In comparison with conventional neural networks, this enhanced cross-correlation achieved a speed-up ratio. The final output can also be described by replacing Eq. (2) with a MLP or a 3-layer feed-forward neural network as follows.

\[ O(u, v) = g \left( W^O_j \sum f \left( W^H_j \sum f \left( z \otimes W^I_i + b^I_i \right) + b^H_j \right) + b^O \right) \] (6)

The final output as a scalar matrix has dimensions of \((S - m + 1) \times (T - n + 1)\), where the input image size is \(S \times T\) and the sliding subimage size is \(m \times n\). Based on a given input, the CNN performs a cross-correlation operation in order to detect and localize the vehicle or verify the nonvehicle region. The CNN detection output is a matrix with values of +1 and −1, representing vehicle and nonvehicle, respectively.

### 3.1. Multiscale detection

The sizes of the vehicle and nonvehicle are usually large in the real environment image. As the training procedure for the neural network was performed by 25 × 25 pixel windows, it would only detect a vehicle and nonvehicle of that size. Thus, the input image was scaled down 1.2 times per each step in the pyramid. The input image will be swept with different resolutions; therefore, vehicle and nonvehicle candidates can be detected by subsampling at several scales before sending to the neural network. During the cross-correlation operation, the subsampling is obtained using the Fourier transform scaling property as follows.

\[ f(\alpha x, \beta y) \Leftrightarrow \frac{1}{|\alpha \beta|} F \left( \frac{u}{\alpha}, \frac{v}{\beta} \right) \] (7)

Both scaling factors are set to 2 (i.e. \(a = b = 2\)) in order to reduce the image size to half of the original size.

### 3.2. Network training principles

The key to success in neural network training lies in the appropriate selection of network properties, as shown in Table 2. The first stage of the training process is to get the image training data for the two different classes. These collected training data are then labeled accordingly to match the training system. There are a number of methods considered and applied in the training of neural networks. In this system, the feed-forward MLP is required to perform this task. Each training image used produces three different spatial invariant images. The subimage histogram equalization is calculated by the following equation.

\[ H^{(a)}_I(z) = \frac{1}{T} \sum_v \delta \left( z - I^{(a)} \right)(v) \] (8)
A bootstrap algorithm is employed to enhance the performance of the neural network by adding nonvehicle images to the database and automatically removing the false detections in order to insert these data into the training set. Network weights are adjusted based on their error rates in the training procedure of the bootstrap algorithm as it is calculated in the following equation.

\[ w_j = c \exp \left[ -y_j \sum_{t=1}^{i} \varphi_t(x_j) \right] \]  

(9)

Training examples are required for vehicle and nonvehicle classification for each set.

### Table 2. Neural network characteristics.

<table>
<thead>
<tr>
<th>Network Properties</th>
<th>FNN</th>
<th>CNN</th>
</tr>
</thead>
<tbody>
<tr>
<td>Trained image size</td>
<td>25 × 25</td>
<td>25 × 25</td>
</tr>
<tr>
<td>Number of input units</td>
<td>30</td>
<td>25</td>
</tr>
<tr>
<td>Function of activation</td>
<td>Sigmoid</td>
<td>Sigmoid</td>
</tr>
<tr>
<td>Number of hidden units</td>
<td>15</td>
<td>10</td>
</tr>
<tr>
<td>Preprocessing type</td>
<td>-</td>
<td>Lighting normalization</td>
</tr>
<tr>
<td>Number of bootstrap iterations</td>
<td>3</td>
<td>3</td>
</tr>
</tbody>
</table>

#### 3.3. Training vehicles and nonvehicles

The vehicle images as the positive training images have been collected from various sources including the Internet and a personal camera. A cropping program is then employed and labeled manually, pointing to the position of the bumper and center of the vehicle in order to align the position and using the same scale. After that, the vehicles are adjusted to a uniform size of all the training set within a 25 × 25 pixel window. A set of 702 vehicle images has been produced from 176 vehicle samples where they have been randomly scaled down 1.2 for each pyramid step, translated to a half pixel, and finally rotated with a random angle of −50 to 50.

A set of 1639 nonvehicle images is also cropped and labeled manually by positioning nonvehicles: for example, a computer mouse, porcelain cup, etc. Next, the images are subjected to the same procedure as in the case of vehicle images that involves alignment, pixel resizing, scaling, translating, and rotating.

#### 3.4. Vehicle verification

As we discussed earlier, a fast multiscale vehicle detection approach is used in this system. It is based on computing the Fourier transform of the image and using a neural network classifier, and then the image is processed in the Fourier space [15]. A single network in this approach is not reliable since we observed that it suffered from change in illumination, which generated more false detections. Therefore, a classical neural network was employed to select the true vehicle candidates and ignore the false detections. The CNN is trained using the preprocessed dataset with input of 25 × 25 pixel images. The CNN properties are described in Table 2. The step by step procedure for verification is described as follows:

- The resolution of possible candidates for vehicle images in any scaling level is interpolated and subsampled to 25 × 25 pixels.
- A lighting normalization method is applied to reduce the effect of lighting conditions and camera characteristics. First the intensity values of selected regions are subtracted with the best-fit linear function. Then image contrast is enhanced by performing histogram equalization.
The vehicle regions are achieved by sending preprocessed data to the CNN to generate vehicle maps. Those outputs that are below the determined threshold are rejected; otherwise, the vehicle regions are mapped onto the original image.

4. Results and discussion
4.1. Bootstrap process
Two sets of image data were used in the bootstrap testing of this module to classify the vehicle region and the vehicle and nonvehicle classification algorithms’ performances. The dataset, known as the test dataset, consists of 200 test images. Test dataset-1 consists of images with complex backgrounds in variant scenarios changing in scale for the object of interest along with some occlusion and variations in lighting. The second dataset, or test dataset-2, contains 100 test images. All the images contain nonvehicle objects. The module or subsystem undergoes a bootstrapping cycle that ends up with 4500 to 9500 zero samples. Therefore, the true detection performance on test images has been evaluated. The algorithm is tested and optimized for implementation of the proposed approach on an ASUS CPU board, with a 2.5 GHz processor and 2 GB RAM.

4.2. FNN database training process
In this process, the false accepted rate (FAR), true accepted rate (TAR), false rejected rate (FRR), precision, and accuracy were determined. The percentage of FAR is 0.01981%. This shows that during the FNN training process, only 1.39 out of 7044 data images are falsely accepted. For TAR, 7031.45 out of 7044 data images are truly accepted, and 12.54 out of 7044 data images are falsely rejected in FRR. This training process is reliable due to the high precision and accuracy rates, which were 99.9802% and 99.9236%.

The performance of the FNN training process is shown in Figure 4. From the plot, it can be concluded that when the FNN is trained with more images, the sum square error will decrease and hence it will increase the FNN training performance. It did not reach the goal performance \(10^{-6}\) because the gradient of the FNN process had reached the limit, which is \(1.00e^{-6}\).

![Figure 4. The FNN performance.](image)

Figure 5 shows the training state of the FNN training process. The gradient decreases when it is trained with more images. The training process will be stopped when the gradient reaches \(1.00e^{-6}\), the goal of the
training process. The gradient goal can be adjusted to increase the images trained, but in this case, we set the
goal gradient to be $1.00e^{-6}$. In the same plot we can also see that the validation check is 0 during the training
process. This validation check is correct because there was no request for any check on the process, meaning
that the process is correct during the FNN training.

Figure 5. The FNN training state.

Figure 6 shows the regression of the FNN training process. From the plot, we can see that the detection
output was nearly linear with the target output, which was 0.99834. That means that the obtained output from
the training process is nearly the same as the requested output.

Figure 6. The FNN regression.

4.3. CNN database training process

In this process, the percentage of FAR is 0.039968%. This shows that during the CNN database training process,
1.76 out of 7044 data images are falsely accepted. For TAR, 7044 out of 7044 data images are truly accepted,
and 0 out of 7044 data images are falsely rejected in FRR. This CNN database training process produces 99.96%
precision rate and 99.9812% accuracy rate.

Figure 7 presents the performance of the CNN database training process. Due to the gradient limit,
the training process that reaches the gradient limit will stop. To increase the performance, more images are
required to train and hence the gradient limit must be adjusted lower than 1.00e^{-6}.
The training process will stop when the gradient reaches $1.00 \times 10^{-6}$, the goal of the training process. As mentioned above, the goal gradient can be adjusted to increase the images trained, but in this case, the goal gradient was set to $1.00 \times 10^{-6}$. In the same plot we can also see that the validation check is 0 during the training process. This validation check is correct because there is no request for any validation check on the 412 processes, meaning that the process is correct during the CNN database training as shown in Figure 8.

From the plot we can see that the detection output was nearly linear with the target output, which is 0.99962, and it is more accurate compared to the FNN regression, which was 0.99834. That means that the obtained output from the training process is close to the requested output as shown in Figure 9.

4.4. Test image and simulation results

One set of image data was used in the simulation testing this module to classify vehicle regions and also for the vehicle and nonvehicle classification algorithms’ performances. The dataset known as test images consists of 72 images out of 200 images that contain vehicle regions. The test image consists of a variety of complex background and scale changes for the object of interest along with some occlusion and variations in lighting.

With the use of the combination of the FNN and CNN methods, the detected vehicle percentage obtained was higher compared to the previous classification methods, as shown in Table 3. The percentage of the FNN and CNN combination method is 95.83%, which also means that only 3 out of 72 test images were detected falsely, as shown in Figure 10.
Figure 9. The CNN regression.

Table 3. Comparison between FNN+CNN method to the previous classification methods.

<table>
<thead>
<tr>
<th>Method</th>
<th>Vehicles detected (%)</th>
<th>Vehicles missed (%)</th>
<th>No. falsely detected</th>
</tr>
</thead>
<tbody>
<tr>
<td>FNN+CNN</td>
<td>95.83</td>
<td>4.17</td>
<td>3 out of 72</td>
</tr>
<tr>
<td>Lan and Kuo [16]</td>
<td>90.38</td>
<td>9.62</td>
<td>5 out of 52</td>
</tr>
<tr>
<td>Peng and Harlow [17]</td>
<td>91.00</td>
<td>9.00</td>
<td>90 out of 1000</td>
</tr>
<tr>
<td>Shaoqing et al. [18]</td>
<td>93.13</td>
<td>6.87</td>
<td>46 out of 670</td>
</tr>
</tbody>
</table>

Figure 10. The simulation results for the FNN + CNN processes.

There is a noticeable difference between the previous methods’ performances and the proposed approach. Lan and Kuo [16] introduced a method with 90.38% as the detected percentage, which means that 5 out of 52 test images were detected falsely. Peng and Harlow [17] achieved 91.00% accuracy; 90 out of 1000 test images were detected falsely. Shaoqing et al. [18] presented a technique with the detection rate of 93.13% for vehicle classification (46 out of 670 of test images were detected). The summary of comparisons is described in Table 3. Hence, the combination of the FNN and CNN methods has more credit compared to the previous classification methods.
5. Conclusion
In this paper we have proposed a method based on the combination of the FNN and CNN methods, which, in comparison with the previous classification methods, has been proven to perform well and hence can be implemented in automatic vehicle classification (AVC). Results obtained in this work confirm that the developed methods and algorithms are effective and can provide a comprehensive method for a real-time AVC prototype, which may lead to an overall improvement in vehicles with regards to traffic performance. This proposed method (FNN+CNN) can also be employed in the toll fare collecting system replacement scheme, because the detected percentage of this method was higher than that of the other automatic vehicle classification methods. The system’s performance can be improved by increasing the images for database training. This can be done by adjusting the gradient goal and sum squared error goal.

Nomenclature

\[ g(x) = \frac{1}{1 + e^{-x}} \]  
sigmoid function for neural networks outputs 1 and -1

\[ b_i \text{th hidden neuron bias} \]

\[ h_i \text{th hidden neuron} \]

\[ h_i(u, v) \text{i th hidden unit activity for the whole test image} \]

\[ \bar{F} \text{Fourier transform conjugated complex} \]

\[ O \text{final output} \]

\[ W^I \text{input layers weights} \]

\[ W^H \text{hidden layers weights} \]

\[ W^O \text{output layers weights} \]

\[ f(ax, by) \text{original image} \]

\[ \Phi_{t i o n} \text{boosting classifier in cascade} \]

\[ c \text{initial weight for negative samples} \]

Fourier transform of original image

\[ F(u/a, v/a) \]

scaling factors

\[ a \text{ and } b \]

histogram equalization of subimage

\[ H_i^{(a)}(z) \]

subimage

\[ \alpha \]

normalizing constant

\[ z \]

subimage input vector

\[ \delta I \]

coefficient

\[ I_{(\alpha)}(v) \]

normal subband image through linear convolution

\[ w_j \]

weighting of the sample \( x_j \)

\[ y_j \]

label of the sample \( x_j \)

\[ i \]

index of the current node

References


