

Traffic Density Identification based on Neural Network and Histogram

Luong Anh Tuan Nguyen
Faculty of Information Technology
HCMC University of Transport

No.2, D3 Street, Ward 25, Binh Thanh District, Ho Chi Minh City, Vietnam

Thi-Ngoc-Thanh Nguyen
Faculty of Information Technology
HCMC University of Transport

No.2, D3 Street, Ward 25, Binh Thanh District, Ho Chi Minh City, Vietnam

ABSTRACT

The traffic density identification will support the traffic problems such as intelligent traffic signal control, traffic planning, etc. This paper proposes a traffic density identification method based on histogram and neural network. The system model was designed and evaluated with the traffic image datasets of Ho Chi Minh city. The best identifying result can obtain 96%.

Keywords

Traffic Image, Histogram, Neural Network, Traffic Density.

1. INTRODUCTION

There are several methods of identifying traffic density [1], [2], although the technique is effective but calculations is so complicated.

A traffic image with very crowded traffic density is easy to distinguish between the traffic objects and the background of road. The traffic objects will appear most of image. Besides, the traffic objects in image have the contrast with the background of road. This paper proposes a novel technique to identify the traffic density based on neural network and histogram. This paper was developed from our previous work [3].

The remainder of the paper is organized as follows. The theoretical background is discussed in Section 2. Section 3 presents the design of system model. In section 4, the numerical results of experiment are illustrated. Finally, Section 5 concludes this paper and figures out the future works.

2. THEORETICAL BACKGROUND

2.1 Monochrome Image

A monochrome image [4], [5], [6] is a 2-D array. Each element of array is a pixel. Each pixel occupies one byte of memory. The total number of bytes in the image is $M \times N$ bytes (M : rows, N : columns). The following equation illustrates the pixel value of monochrome image.

$$f(x, y) = \begin{matrix} f(0,0) & \dots & f(0, N-1) \\ f(1,0) & \dots & f(1, N-1) \\ \vdots & & \vdots \\ f(M-1,0) & \dots & f(M-1, N-1) \end{matrix}$$

Where, $f(x,y) \in Z$ (Z belongs to the positive integers), and $0 \leq f(x,y) \leq 255$.

2.2 Color Image

Color image is presented by a triple RGB (Red, Green, Blue). The value of color channels ranges from 0 to 255. Set of 3 color channels will make 2^{24} colors ($256 * 256 * 256$ colors).

2.3 Grayscale Image

Grayscale images [6], [7] are the color images using RGB color system in which the Red, Green, Blue have the same light intensity. So, the grayscale image just need to use one light intensity to show each pixel. The gray level of grayscale image ranges from 0 to 255.

2.4 Histogram

Histogram [6], [8] is the chart that shows the frequency of occurrence of each gray level in an image.

- The horizontal axis X is the value of gray level.
- Y-axis is the number of pixels with gray levels corresponding to the image (frequency of occurrence).

Figure 1 illustrates the histogram of an image.

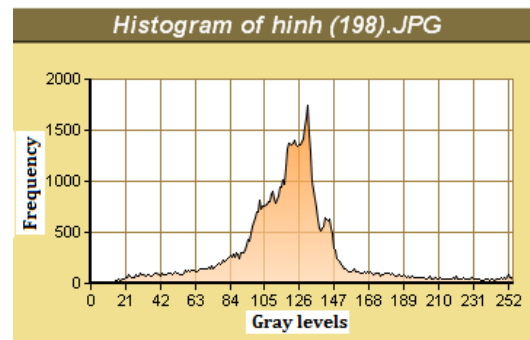


Fig 1: Histogram of image

3. SYSTEM DESIGN

3.1 System Model

The system model including 6 stages is presented in Figure 2.

- The first stage: Color image obtained from the camera.
- The second stage: Converting color image into grayscale image.
- The third stage: Calculating histogram of grayscale image.
- The fourth stage: Calculating Medium and Maximum of histogram.
- The fifth stage: Neural Network.
- The sixth stage: Identifying the traffic density based on the neural network.

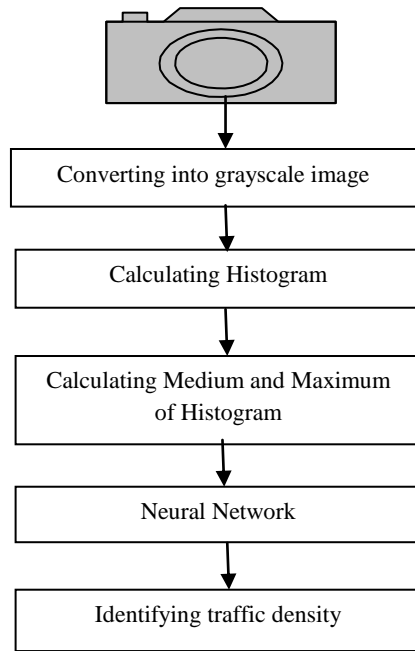


Fig 2: System Model

3.2 Convert into Grayscale Image

There are some techniques to convert color image into grayscale image such as lightness with the equation $(\max(R, G, B) + \min(R, G, B))/2$, luminosity with the equation $0.21R + 0.72G + 0.07B$. In this paper, the average method is chosen, so the value of pixel in grayscale image is calculated by (1).

$$V = (R + G + B) / 3 \quad (1)$$

For example, one pixel of color image has $R=140$, $G=200$, $B=98$, when converting into grayscale image, the value of pixel in grayscale image is $(140+200+98)/3=146$.

3.3 Calculating Histogram

Calculating histogram of image is performed as follows:

- Building the pixel matrix of the grayscale image.
- From the pixel matrix of the grayscale image, building the frequency of pixel.

Figure 3 presents the procedure of calculating the histogram of the grayscale image. Figure 4, 5 illustrate color image and grayscale image.

3.4 Calculating Medium and Maximum of Histogram

First, arranging the frequency of pixels (F_i) in ascending direction. The maximum value of the histogram is the last value in the frequency table. The medium value of the histogram is calculated as follows:

- If n is odd then $Med = F_{n/2}$
- If n is even then $Med = (F_{n/2} + F_{n/2+1}) / 2$

Where, n is the total number of elements in the frequency table.

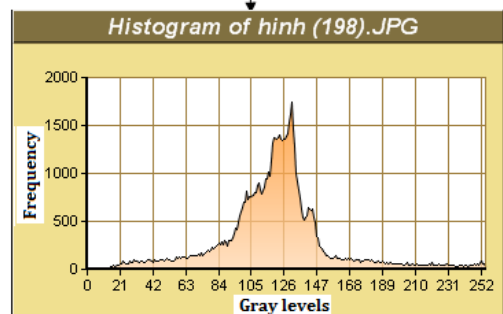
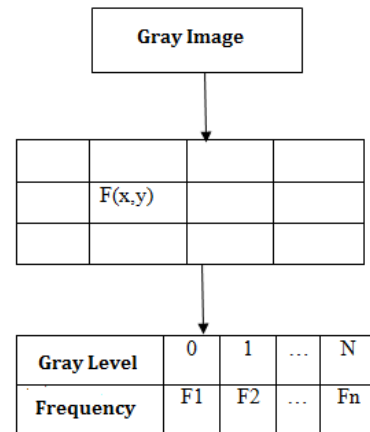


Fig 3: The procedure of calculating the histogram of image



Fig 4: color image



Fig 5: grayscale image

3.5 Neural Network

3.5.1 The Model

The model of neural network was designed as in Fig 6. The model was designed with three layers as follows:

- The first layer, called the input layer, contains two nodes that are Medium and Maximum of histogram.
- The second layer, called the hidden layer, contains 3 nodes.
- The third layer, called the output layer, contains one node.

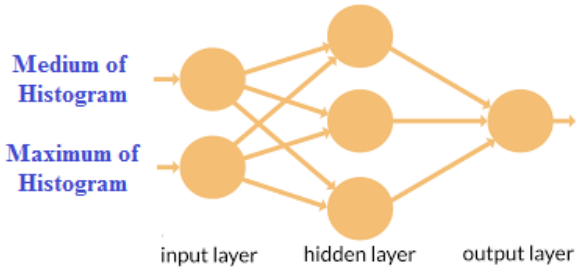


Fig 6: Neural Network Model

3.5.2 The Training Algorithm

Before using the model to identify the traffic density, the neural network need to pass the training phase to learn. The flow of the training algorithm is shown in Fig. 7, representing a backpropagation learning procedure [9].

Entering pairs of input and output values from the training data set, this procedure iteratively propogates values from the input layer to output layer and updates the weights from the input layer to the output layer until the error of the output node is lower than the threshold.

The algorithm performs two phases as follows:

- The "propagation" phase calculates the input value, the output value of each node in the hidden layer and the output layer.
 - The input value of the hidden nodes is calculated by equation (2).

$$I_i = \sum_{j=1}^3 w_{ji} * V_j ; j = 1,2 \quad (2)$$

Where, I_i , V_j and w_{ji} are the i^{th} input value of the hidden layer, the j^{th} value of the input layer and the weight from the j^{th} node of the previous layer to the i^{th} node of the next layer respectively.

- The output value of the hidden nodes is calculated by equation (3).

$$O_i = \frac{1}{1 + e^{-I_i}} ; i = 1..3 \quad (3)$$

Where, I_i and O_i are the i^{th} input value and the i^{th} output value in the hidden layer respectively.

- The input value of the output node is calculated by equation (4).

$$I = \sum_{i=1}^3 w_i * O_i \quad (4)$$

Where, O_i and W_i are the i^{th} value of the hidden layer and the weigh from the i^{th} node of the hidden layer to the output node respectively.

- The output value of the output node is calculated by equation (5).

$$O = \frac{1}{1 + e^{-I}} \quad (5)$$

Where, O and I the input value and the output value in the output layer respectively.

- The "weight update" phase calculates the error of the nodes in the hidden layer and the output layer, then updates the weights.

- The error of the output node is calculated by equation (6).

$$Err = O * (1 - O) * (T - O) \quad (6)$$

Where, T and O are the real value of sample in training dataset, the output value of output node respectively.

- The error of the i^{th} node in the hidden layer is calculated by equation (7)

$$Err_i = O_i * (1 - O_i) * \overset{\circ}{a} Err * w_i ; i = 1..3 \quad (7)$$

Where O_i , W_i and Err are the output value of the i^{th} hidden node, the weight of the connection from the i^{th} hidden node to the output node and the error of the output node respectively.

- The weights from the hidden layer to the output layer are updated by equation (8).

$$\begin{cases} Dw_i = R * Err * O_i \\ w_i = w_i + Dw_i ; i = 1..3 \end{cases} \quad (8)$$

Where, O_i , Err and R are the i^{th} output value of the hidden layer, the error of the output node and the learning rate respectively.

- The weights from the input layer to the hidden layer are updated by equation (9).

$$\begin{cases} Dw_{ji} = R * Err_i * V_j \\ w_{ji} = w_{ji} + Dw_{ji} ; i = 1..3, j = 1..2 \end{cases} \quad (9)$$

Where, V_j , Err_i and R are the j^{th} input value, the error of the i^{th} hidden node and the learning rate respectively.

- The system error based on RMSE [10] is calculated by equation (10).

$$Err_s = \sqrt{\frac{\sum_{i=1}^N (Err)^2}{N}} \quad (10)$$

Where, N and Err are the number of samples in the training data set and the error of the output node respectively.

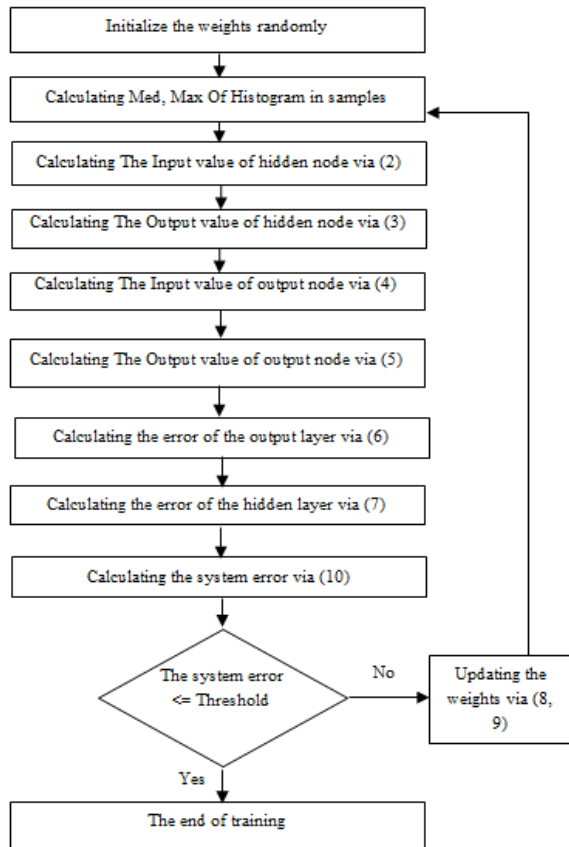


Fig 7: The training algorithm

3.5.3 The Identification Algorithm

The algorithm based on the trained weights of the neural network to identify the traffic density. The flow of the identification algorithm is shown in Fig. 8.

Traffic density based on the output value of the output node as follows:

- The output value of the output node ranges from 0 to 0.25: **Sparse Traffic**
- The output value of the output node ranges from 0.25 to 0.5: **Normal Traffic**
- The output value of the output node ranges from 0.5 to 0.75: **Crowded Traffic**
- The output value of the output node ranges from 0.75 to 1: **Very Crowded Traffic**

4. EXPERIMENT RESULTS

4.1 Datasets

In this paper, 300 images were taken in Ho Chi Minh city, Vietnam on many different streets with different times. These images with the traffic density such as very crowded traffic, crowded traffic, normal traffic and sparse traffic are experimented in proposed technique.

Data set is divided into training data set and testing data set. The training data set contains 200 images. The testing data set contains the remaining 100 images.

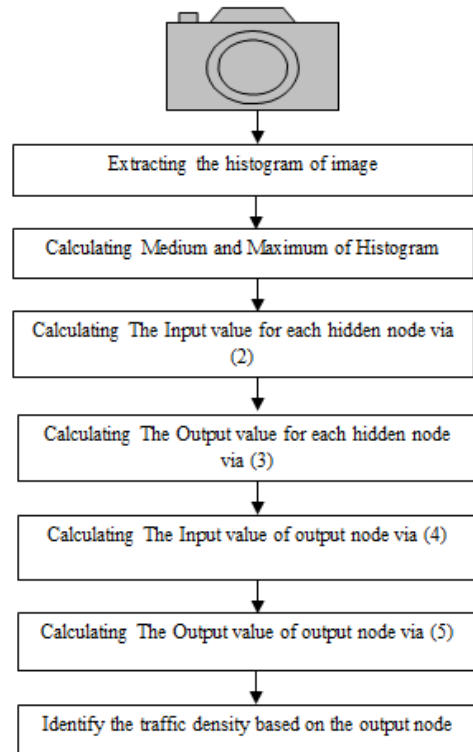


Fig 8: The identification algorithm

4.2 Experiment Procedure and Results

A software is designed by VB.NET programming language to experiment. In the training phase with the training data set, the parameters are set as follows:

- Mean error threshold value: 1×10^{-5}
- Number of Epochs: 1,000
- The weights: initialize weights randomly from -0.5 to 0.5
- Learning Rate: 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.8, 0.9

The experiment result with the testing data set is shown in table 1. The best result obtained 96%. This result is better than the paper [3].

Table 1. The Experiment result with the testing data set

Learning Rate (R)	Accuracy Rate	Time
0.1	94%	17ms/ an image
0.2	95%	16ms/ an image
0.3	94%	17ms/ an image
0.4	94%	15ms/ an image
0.5	95%	17ms/ an image
0.6	95%	16ms/ an image
0.7	96%	16ms/ an image
0.8	95%	17ms/ an image
0.9	94%	15ms/ an image

Figure 9, 10, 11 and 12 present some experimental results with the learning rate of 0.7.

5. CONCLUSIONS

In this paper, the traffic density identification technique based on neural network and histogram is proposed. The technique is designed and experimented via VB.NET programming language with 300 images taken in Ho Chi Minh city, Vietnam on many different streets with different times. The best result was obtained with 96% accuracy. In the future, the proposed technique will be improved by upgrading the neural network model and combining with fuzzy theory.

6. REFERENCES

- [1] Ozkurt C, Camci F. Automatic traffic density estimation and vehicle classification for traffic surveillance systems using Neural Networks. *Mathematical and Computational Applications*. 2009; 14(3):187–96.
- [2] C. Stuij, T. A. Runkler, “Classification and Predicts of Road Traffic using Application Specific Fuzzy Clustering”, *Fuzzy Systems, IEEE Transactions*, pp. 297-308, 2002.
- [3] Luong Anh Tuan Nguyen, Thi-Ngoc-Thanh Nguyen. Traffic Image Classification using Horizontal Slice Algorithm. *International Journal of Computer Applications* (ISSN: 0975 – 8887), Volume 148 – No.11, pp. 30-34, August 2016.
- [4] Al Bovik (2000), *Handbook of Image and Video Processing*, Academic Press.
- [5] Rafael C.Gonzalez, Richard E. Woods (1993), *Digital Image Processing*, Addison Wesley Pub.Comp.
- [6] Luong Anh Tuan Nguyen, Huu Khuong Nguyen. Traffic Density Identification Based On Histogram. *Journal of Transportation Science and Technology*, ISSN: 1859-4263, Vol 15-05/2015, pp 23-27.
- [7] Xiangyun Ye, Mohamed Cheriet, Senior Member, Ching Y. Suen (2001), *Stroke-Model-Based Character Extraction from Gray-Level Document Images*, IEEE.
- [8] C. C. Sun. S. J. Ruan, M. C. Shie, T. W. Pai, “Dynamic Contrast Enhancement based on Histogram Specification,” *IEEE Transactions on Consumer Electronics*, 51(4), pp.1300-1305, 2005.
- [9] Jiawei Han, Micheline Kamber (2006), *Data Mining: Concepts and Techniques*, Second Edition, Elsevier Inc. All rights reserved.
- [10] C. Willmott, and K. Matsuura (2005), Advantages of the Mean Absolute Error (MAE) over the Root Mean Square Error (RMSE) in assessing average model performance, *Clim. Res.*, 30, 79–82.

7. APPENDIX

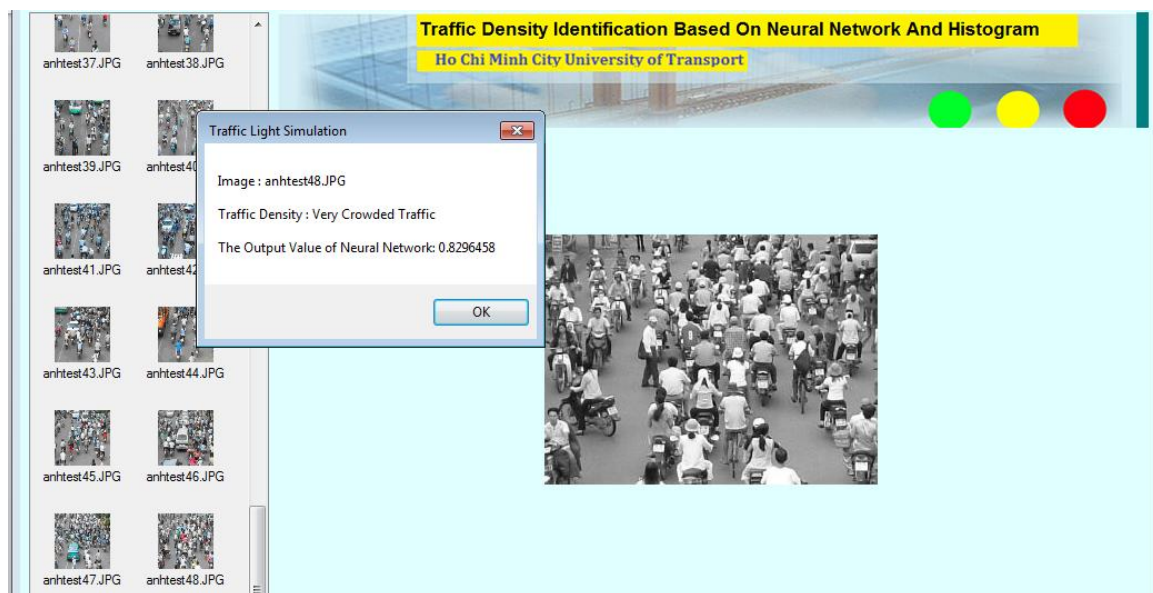


Fig 9: The result of “Very Crowded Traffic” density

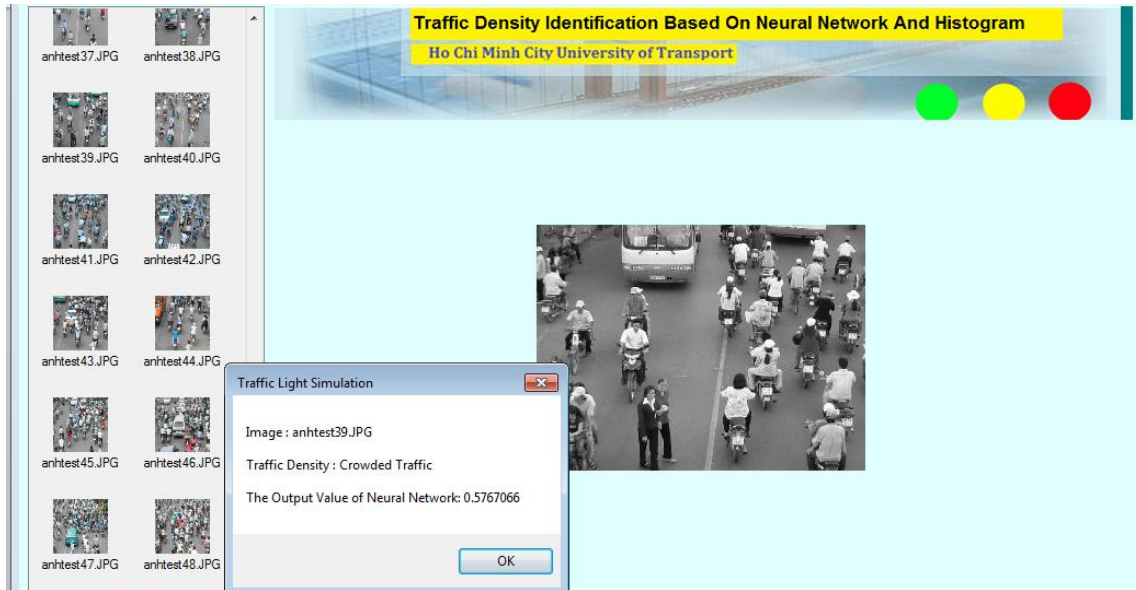


Fig 10: The result of “Crowded Traffic” density

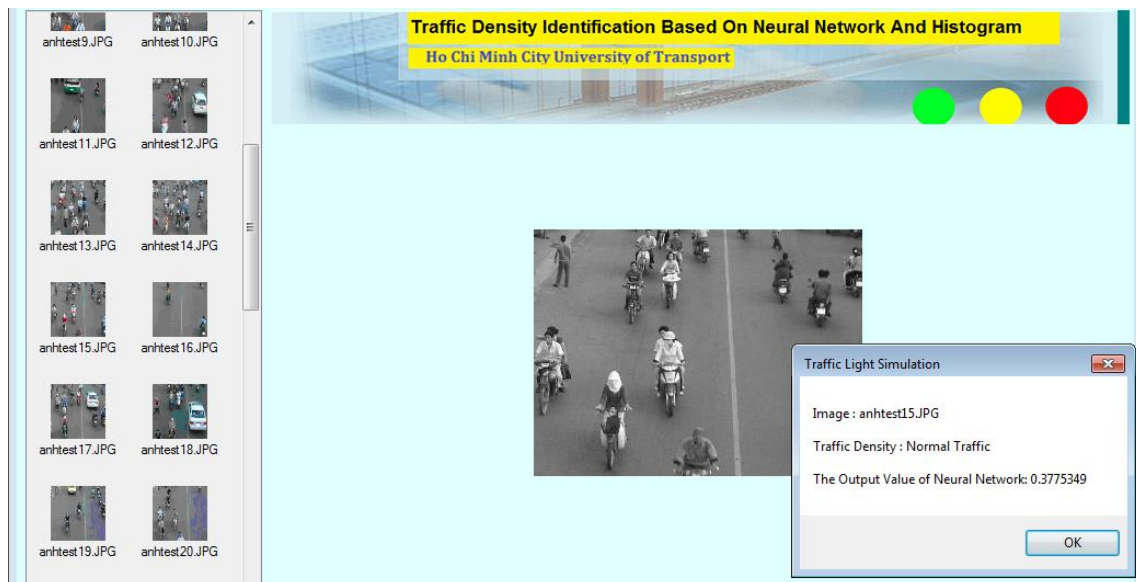


Fig 11: The result of “Normal Traffic” density

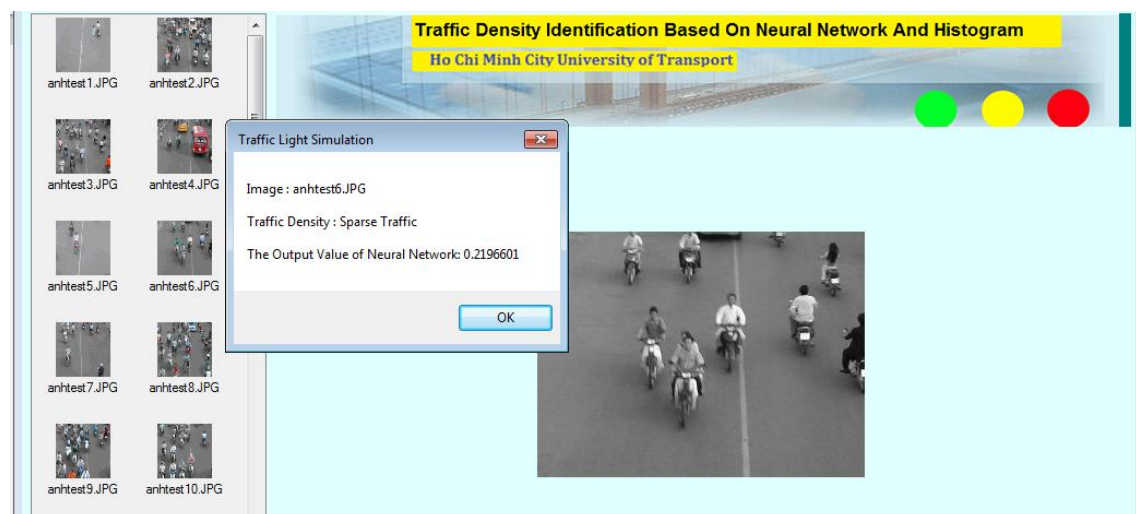


Fig 12: The result of “Sparse Traffic” density