



PRIMARY RESEARCH

Vehicle types classification using deep neural network techniques

Jian-Da Wu^{1*}, Yi-Cheng Luo², Hsien-Yu Lin³^{1,2,3} Institute of Vehicle Engineering, National Changhua University of Education, Changhua, Taiwan

Index Terms

Vehicle Classification System
Deep Neural Network
Convolutional
Generalized Regression
Image Processing

Abstract— Traffic flow is one of the most important information elements in intelligent traffic transportation engineering. This study developed a vehicle type classification system using a neural network technique. The architecture of this study is divided into two parts, vehicle pictures are collected first, and then divided into motorcycles, sedans, recreation vehicles, buses and trucks to build a contrast database. The image processing techniques included median filtering and edge detection used to de-noise to improve recognition efficiency. The second stage is processing the previous data stage into the system identification database. All data created by the database were then input into the classifier for calculation. The classification recognition rate was finally obtained. This study uses the Generalized Regression Shallow Learning Neural Network (GRNN), Deep Neural Network (DNN) and Convolutional Neural Network (CNN) classification algorithms. The results show that vehicle classification using the convolutional neural network is better than that obtained with the deep neural network using the restricted Boltzmann machine. Both types of neural networks produced much higher classification than the generalized neural network. The deep learning technique was shown better than the shallow learning approach in this study.

Received: 10 July 2017**Accepted:** 12 September 2017**Published:** 4 December 2017

© 2017 The Author(s). Published by TAF Publishing.

I. INTRODUCTION

Technology development is changing with each passing day. When city transportation system hardware reaches a certain level, or the government lacks the funds to provide large transport improvement construction, the transportation system goes into crisis and if not addressed, eventually influences transport. Intelligent transportation technology can improve transportation capacity. The intelligent transportation system uses information, communications, electronics, control and management to add value to existing transportation facilities. One of many intelligent transportation systems is the detection, monitoring, control, transmission and display of roadway traffic conditions. This system improves traffic flow characteristics using real-time traffic flow calculation. The traffic flow characteristics are the indispensable information in social transport planning and city construction. The influences of traffic lights and signs on traffic flow characteristics are indispensable data. The traffic flow of passenger cars equivalent is one of

the ways flow on the road is calculated. The passenger car equivalent is based on the existing road layout and traffic composition. Each individual vehicle has a relatively small car impact on the traffic flow. The passenger car equivalent formula has different values in different places because each city's speed limit and traffic composition is different. The passenger car equivalent presents different qualities under different circumstances [1, 2].

This study uses ministry of transportation compiled specifications for the passenger car equivalent formula. The weight coefficient for a motorcycle is set at 0.3, for a small car is 1 and for a large car is 2. The equivalent number of passenger cars will be set according to region-wide traffic flow. Neural network technology is proposed for conducting vehicle type classification pictures. In order to understand neural network technology performance in vehicle classification, deep learning and shallow learning algorithms are used and compared in this study.

* Corresponding author: Jian-Da Wu

† Email: jdwu@cc.ncue.edu.tw

Artificial intelligence has been used in many applications such as information and electronic engineering. The deep model was proposed by Hinton in 2006 with multilayer hidden layer learning [3]. This method uses the Smolensky research based on the restricted Boltzmann machine proposed in 1986 [4]. This model guarantees that the data logarithm increases as the number of layers increases. When enough layers are studied, this deep structure becomes a generation model that can be reconstructed using top-down sampling, reconstructing the whole data set. Hinton [5] believed this model can be a dimensional data structure with effectively extracted features. In 2012, Dean and Ng [6, 7] lead the Google team to use the deep neural network in successful applications. In 2011, feed forward neural network deep learning uses convolutional layers and pooling layers with a pure classification layer. The training process also eliminated the need to introduce unsupervised pre-training [8, 9, 10, 11, 12]. This method won various types of pattern recognition contests in 2011 [13], including the 2011 traffic mark recognition contest [14] and other games. Neural network technology is proposed in this study for use in vehicle classification for traffic flow information acquisition and computation in an intelligent transportation system [15, 16, 17, 18, 19].

II. PRINCIPLES OF VEHICLE IMAGE PROCESSING AND NEURAL NETWORK

Neural network technology is used in vehicle classification to train and classify vehicle pictures. Both the median filter and edge detection are used in image processing to improve performance. These two methods will delete some unnecessary image information, reduce the operating time and improve the recognition rate. This study can be divided into three stages: image capture, image processing and classifier identification. These stages will be discussed and analyzed separately. The research methods are described in the following sections.

A. Deep Neural Network

The deep DNN used in this study is pre-trained by the Deep Belief Network (DBN) to provide good initial parameters for deep neural networks [20, 21]. Three or more Restricted Boltzmann Machine (RBM) layers compose our hidden layer, as shown in Figure 1. The completed deep neural network architecture is shown in Figure 2.

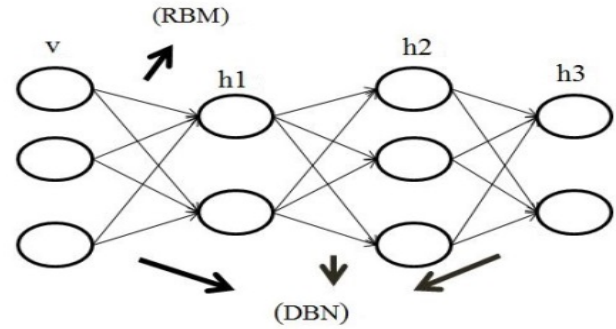


Fig. 1 . Structure of DBN-RBM

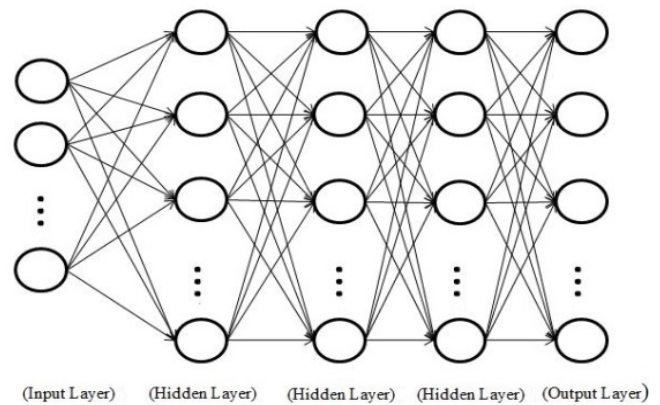


Fig. 2 . Structure of DNN

Figure 1 shows the RBM will limit these networks to a visual layer and a hidden layer. V represents the visual layer and $h1$ to $h3$ represent the hidden layer. These layers are connected, but the layers of neurons are not connected to each other. These hidden layers are trained to capture the relevant feature data from the visual layer. In RBM pre-training the joint probabilities of a visual node and a hidden node are shown in Equation 1.

$$p(v, h; \theta) = \frac{\exp(-E(v, h, \theta))}{\sum_v \sum_h \exp(-E(v, h, \theta))} \quad (1)$$

The joint probability of each visible node, as shown in Equation 2.

$$p(v, h; \theta) = \frac{\sum_h \exp(-E(v, h, \theta))}{\sum_v \sum_h \exp(-E(v, h, \theta))} \quad (2)$$

where θ is the RBM parameter and v is one of the visible nodes. h is one of the hidden nodes and E is an energy function. There are two training algorithms in RBM. The first algorithm converts the input real random variable into a binary random variable, as shown in Equation 3.

$$\begin{aligned}
 E(v, h; \theta) &= - \sum_{i=1}^i \sum_{j=1}^j w_{ji} v_i h_j - \sum_{i=1}^i n_i v_i - \sum_{j=1}^j m_j h_j \\
 p(h_j = 1 | v; \theta) &= \sigma(\sum_{(i=1)}^i w_{ij} v_i + m_j) \\
 p(v_i | h; \theta) &= \sigma(\sum_{(j=1)}^j w_{ij} h_j + n_i)
 \end{aligned}
 \tag{3}$$

The second algorithm converts the binary random variable calculated above into another binary variable, as

shown in Equation), so that RBM can retrieve the feature.

$$\begin{aligned}
 E(v, h; \theta) &= - \sum_{i=1}^i \sum_{j=1}^j w_{ji} v_i h_j + \frac{1}{2} \sum_{i=1}^i (v_i - n_j)^2 - \sum_{j=1}^j m_j h_j \\
 p(h_j = 1 | v; \theta) &= \sigma(\sum_{(i=1)}^i w_{ij} v_i + m_j) \\
 p(v_i | h; \theta) &= N(\sum_{(j=1)}^j w_i h_j + n_i, 1)
 \end{aligned}
 \tag{4}$$

where w_{ij} is the weight value of i visual nodes and j hidden nodes, m and n are the offset vectors of the visible and hidden layers, and σ is an activation function defined as Equation 5.

$$\sigma = \tanh(z) = \frac{e^z - e^{-z}}{e^z + e^{-z}} \tag{5}$$

All features are accumulated, and then the original market data are reverse transmitted to complete the DBN training. RBM can also be used as a discriminant model, that is, we can use RBM to classify, as long as the conditional probability $p(y | x)$ is calculated as shown in Equation 6. The x is a training sample, and y is a category label, RBM can be classified as shown in Figure 4.

$$\begin{aligned}
 \frac{\partial \log P(y_j | x_i)}{\partial \theta} &= \sum_j \text{sigm}(o_{yj}(x_i)) \frac{\partial o_{yj}(x_i)}{\partial \theta} - \\
 &\sum_{j, y^*} \text{sigm}(o_{y^*i}(x_i)) P(y^* | x_i) \frac{\partial o_{y^*j}(x_i)}{\partial \theta}
 \end{aligned}
 \tag{6}$$

where $o_{yj}(x) = c_j + \sum_k w_{jk} x_k + U_{jy}$. This can be used to calculate the gradient effectively and then use it in random gradient calculation optimization.

B. Convolutional Neural Network

The CNN is a feed forward neural network. Its artificial neurons can respond to a part of the neuron in the coverage area and have excellent performance for image

lution layers and a top-connected full connection layer, as well as a convolutional layer and a pooling layer. This structure makes the CNN extract the image and the characteristic speech information using the two-dimensional input data structure. Compared with other deep learning structures, the convolutional neural network can give the better image and speech recognition results. CNN can also be trained using the back propagation algorithm. Compared with the deep neural network, the feed forward neural network and the CNN require fewer parameters, making the CNN an attractive deep learning structure.

The purpose of the convolution layer is to extract the different input characteristics. The first convolution layer may only extract some low-level features such as edges, lines and corners. More network layers can extract more complex features from the low-level features. Before the convolution operation, we first need to determine the size of the filter, for the upper local experience layer of a feature x_i and we use the convolution kernel k_{ij} as a weight to get $x_i * k_{ij}$ and then sum it plus the offsets, as shown in Equation 7.

$$x_j^l = f(\sum_{i=M_j} x_i^{l-1} * k_{ij}^l + B^l) \tag{7}$$

where M_j is the local feeling corresponding to neuron j , k_{ij}^l is the weight of the first layer of neuron i to j input, and B^l is the only bias of the first layer. The sampling layer, also known as the pool layer, is intended to reduce the net-

work training parameters and the model over-fitting.

Therefore, the pooling layer is usually added after the convolution layer to maintain the original message in the picture. From the information multiplied by the training parameters, coupled with the training bias, the results obtained by the activation function can be obtained by the current neuron output. The activate function usually uses the sigmoid function or *tanh* function, and the first output formula layer, as shown in Equation 8.

$$x_j^l = f(\beta^l \text{down}(x_i^{l-1}) + B^l) \tag{8}$$

where β is the training parameter for the first layer, and *down* (*x*) represents the down-sampling of *x*, and the activation function can use the sigmod function, as shown in Equation 9, or the *tanh* function as shown in Equation 10.

$$f(x) = \frac{1}{1 + e^{-x}} \tag{9}$$

$$f(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \tag{10}$$

After forwarding propagation computation, the convolution neural network will carry out back-propagation. This is aimed at correcting the parameters and reducing the error rate, error output layer, wherein assuming that the output layer *M* nodes, then the output layer node *K* the error term, as shown in Equation 11. The error from the whole connection layer, a first layer before the layer is assumed. There are nodes *L* and next layer has *M* nodes. Error term for node *j* of the first layer as shown in Equation 12.

$$\delta_k = (d_k - y_k)y_k(1 - y_k) \tag{11}$$

$$\delta_j = h_j \sum_{k=1}^M \delta_k W_{jk} \tag{12}$$

where d_k is the target output of node *K*, y_k is the predicted output of node *K* and h_j is the output of node *j*, w_{jk} is the weight of the first layer to the next layer. After calculating the weight value adjustment, the k^{th} input weight vector change in the n^{th} iteration node *j* is as shown in Equation 13. The change in the threshold value is shown in Equation 14.

$$\Delta w_{jk}(n) = \frac{\eta}{1 + N} (\Delta w_{jk}(n - 1) + 1) \delta_k h_j \tag{13}$$

$$\Delta B_k(n) = \frac{\eta}{1 + N} (\Delta B_k(n - 1) + 1) \delta_k \tag{14}$$

where η is the learning rate, *N* is the current number

of input variables, such as after the update is complete as shown in Equation 15. The threshold is shown in Equation 16.

$$w_{jk}(n + 1) = w_{jk}(n) + \Delta w_{jk}(n) \tag{15}$$

$$B_k(n + 1) = B_k(n) + \Delta B_k(n) \tag{16}$$

The above formula is repeated until the error function decreases to the minimum value and no longer decreases. The error function is shown in Equation 17.

$$E = \frac{1}{2} \sum_{k=1}^M (d_k - y_k)^2 \tag{17}$$

III. DATA PROCESSING OF VEHICLES CLASSIFICATION

Five different vehicle model images were used to classify vehicles. These five vehicle pictures were the sedan, recreation vehicle, bus, truck and motorcycle. Two hundred data points were used for each model. These five models presented different physical structures as the condition for recognition.

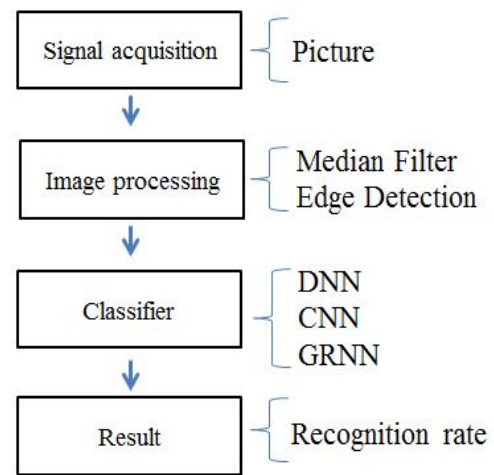


Fig. 3 . Vehicle classification flow chart

A 1000 data point database was established. This database was processed through median filtering, noise reduction and edge detection over again into the classifier.

The study flow chart is shown in Figure 3. Vehicle images were processed through image processing and the classifier for classification. The final recognition rate in this study is the classification result.

A. Image Processing

Gray scale was used after the median filter and edge detection processing to improve the recognition rate. After the pictures were gray scale and median filtered noise reduction was performed followed by edge binarization so that lines can be more clearly displayed. Figure 4 shows five different image gray scale, median filtered and edge detection vehicle pictures.

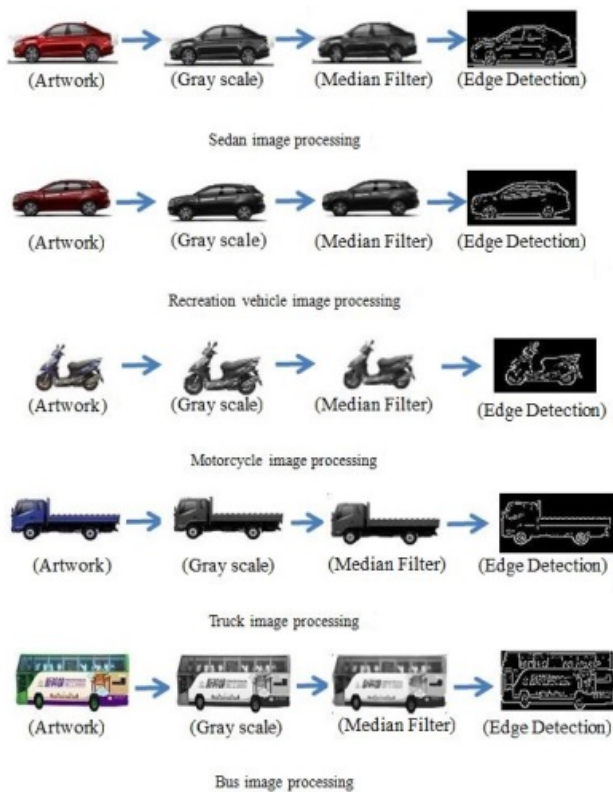


Fig. 4 . Image processing for different vehicle types

B. System Training and Test Conditions

The data were trained and tested using GRNN, DNN and CNN. The test procedure included internal, half-external and external methods of training and testing. The results were then compared. The total amount of information is 1000 pieces of data. Internal refers to all information used in training and all information used in testing. Half-

external refers to half of all information used for training and all data used for testing. Externally refers to all information half of the time used for training, with the remaining information used for testing.

C. Deep Neural Network Parameter Settings

Deep learning and shallow learning are different point in that deep learning has many parameters that affect the overall training time and recognition rates. The DNN affects the training time and the recognition rate for the five parameters. The factors are (n_1 , n_2) hidden layers, learning rate, batch number, iteration number and training numbers. The number of hidden neurons is adjusted by giving n_1 and n_2 an initial value. The number of neurons in another hidden layer is adjusted by fixing the number of neurons in one hidden layer until the recognition rate is not increased. Figure 5 shows the relationship between n_1 , n_2 and the recognition rate. The number of neurons in the first hidden layer n_1 is no longer increased at around 900. The number of neurons in the second layer is about 850 when the recognition rate reaches the highest point. This study used $n_1 = 900$ and $n_2 = 850$ hidden neurons so that the identification rate reaches the maximum.

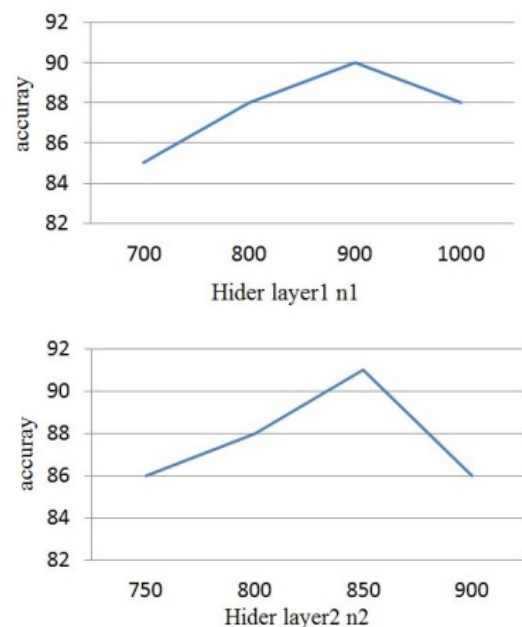


Fig. 5 . Hidden layer parameters of deep neural network

When the weight (iteration) is updated, the gradient algorithm learning rate will be multiplied by a coefficient.

The coefficient is the learning rate. The learning rate adjustment method is accomplished through the study (0.5, 0.05, 0.005) of three values, as shown in Figure 6. The first 100 iterations are studied first, with the first 100 batch number as the initial value. When the learning rate = 0.5, the whole line is turbulent, making the identification rate easy to cross to the minimum, so this will not be used. When the learning rate = 0.005, the whole line has a very stable decline, but the learning rate is too small. This will result in convergence. If the function is too slow the recognition rate will not arrive at the target. A learning rate of 0.05 occurs when the recognition rate can reach a maximum of 92%. When the learning rate is 0.5, 0.005, the recognition rate will reach only 77% and 85%, so this study used the highest learning rate of 0.05.

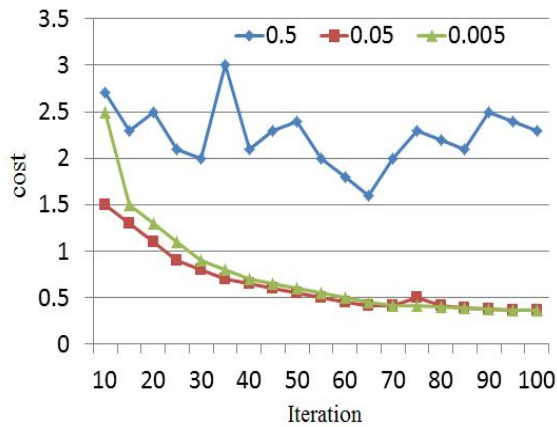


Fig. 6. Learning rate parameters of deep neural network

The batch number is the number of processed samples. If the number of batches is 1, it means that one data point will be calculated. Increasing the number of batches will reduce the recognition rate. These two factors are contradictory to each other. So as the batch number reaches a value, the time will reach the overall optimal. In order to make the recognition rate reach its highest point, the number of batches was set at 10. The number of iterations is maintained from the previous function to find the new value. A repeated feedback process is finally achieved or close to the target and the result. This research verifies that with an increase in the number of iterations, the recognition rate will increase. With an iterations number = 100, the recognition rate no longer increases, while only the training time increases. So the number of iterations selected for the highest recognition rate and shortest training time = 100.

D. Convolutional Neural Network Parameter Setting

CNN and DNN are deep learning models that can be propagated forward and backward. The CNN is not the same as the DNN. The CNN hidden layer is composed of a convolution layer and pool layer, and there is no hidden layer $n1 \ln 2$. These are the two parameters. The CNN has a convolution kernel and the sampling layer needs to be set. In this study these two parameters were not explored, so these two parameters were used as original parameters.

The two convolution kernel layers use 5 and 6. The sampling layer uses 2*2 sizes for this study. The other parameter adjustment method is the same as the DNN parameter. The CNN learning rate passes first (1, 0.1, 0.01) three kinds of values, as shown in Figure 7. When the learning rate = 1, the clear shock function is found, making the recognition rate easy to cross the minimum, so this is not considered for use. When the learning rate = 0.01, the function is in stable decline, but the learning rate is too small, resulting in the convergence function being too slow. The recognition rate will not arrive at the target. This study used half-external data. At a learning rate of 0.1 the recognition rate can reach 98%. When the learning rate is 1, 0.01, the recognition rate is only 96%, 86%, so this study used a CNN learning rate of 0.1. The batch number parameter adjustment is the same as in DNN adjustment. This study used different batch values to select the best value for the recognition rate. The batch number of 2 was found to be the best value. The recognition rate reached the highest point. The adjustment in the number of iterations is also the same as in the DNN. The number of iterations was selected for the highest recognition rate and shortest iteration time = 20.

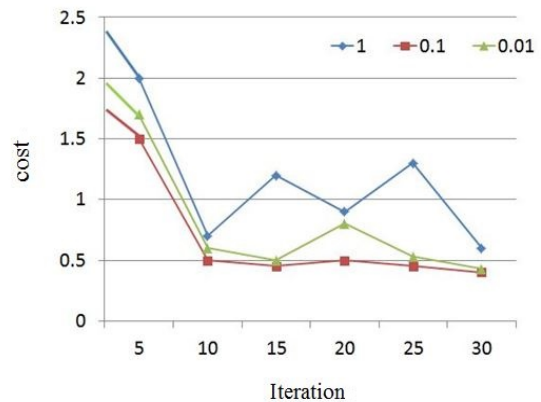


Fig. 7. Learning rate parameters of convolutional neural network

IV. RESULTS OF CLASSIFICATION AND ANALYSES

Five different vehicle pictures were collected for classification including cars, sedans, motorcycles, buses and trucks. Data were trained and tested using the GRNN, DNN and CNN. In order to understand the neural network performance in vehicle classification, this study was divided into two for comparison. Table 1 shows the recognition rates and training time for the CNN, DNN and GRNN classification types without image processing. Three internal classifier identification rates were all 100%, half-external and external CNN and DNN recognition rate reached more than ninety percent. GRNN attained only seventy percent in the half-external and external recognition rate. CNN and DNN achieved higher rates than GRNN in classifier identification. Although CNN and DNN presented good recognition rate

performance, they spent too much time training compared to the GRNN.

In Table 2, CNN, DNN and GRNN are used to classify vehicle types. The identification rate and training time were processed with a median filter and edge detection image processing. The CNN and DNN can effectively shorten the training time with image processing. When the GRNN used median filtering and edge detection, the training time did not change much, but the recognition rate increased by three to four percent. Tables 1 and 2 show the comparisons. CNN and DNN were used as the identification classifier. Regardless if the image was median filtered and edge detected or other image processing, the recognition rate achieved more than 90 percent. The GRNN achieved a recognition rate from 70 percent to 80 percent after median filtering and edge detection.

TABLE 1
COMPARISON OF THE RECOGNITION RATES AND TRAINING TIME IN VEHICLE TYPES CLASSIFICATION USING DIFFERENT NEURAL NETWORKS (WITHOUT IMAGE PROCESSING)

	GRNN		DNN		CNN	
	Recognition Rate (%)	Time (s)	Recognition Rate (%)	Time (s)	Recognition Rate (%)	Time (s)
Internal	100	64	100	2530	100	39328
Half-external	79	31	97	1753	99	23464
External	74	15	94	1541	94.1	17359

TABLE 2
COMPARISON OF THE RECOGNITION RATES AND TRAINING TIME IN VEHICLE TYPES CLASSIFICATION USING DIFFERENT NEURAL NETWORKS (IMAGE PROCESSED)

	GRNN		DNN		CNN	
	Recognition Rate (%)	Time (s)	Recognition Rate (%)	Time (s)	Recognition Rate (%)	Time (s)
Internal	100	59	100	2430	100	37328
Half-external	83	31	98	1553	99.2	21464
External	77	15	94	1441	94.3	15359

CNN uses unsupervised learning, has strong adaptability and is good at data feature capture. The weight sharing structure network makes CNN more like a biological neural network, with better results in image identification. DNN also belongs to unsupervised learning. The network in each layer operates as the unsupervised RBM. Supervised back-propagation is used to make adjustments so that deep learning improves the recognition rate accuracy. The GRNN uses supervised learning. Artificial preparation examples are used to predict the function of any possible input value for the output. Therefore, the original data after median filtering, edge detection and other image processing can re-

move unnecessary structure so that the GRNN can more easily identify the object. Image processing is a necessary step in GRNN. DNN and CNN can find their own features, but the training time is too long. Median filtering and edge detection can effectively reduce the training time. Image processing is also a necessary step in DNN and CNN. The final result shows that the DNN and CNN recognition rates are much higher than that of GRNN, with or without image processing. Deep learning produces a much higher recognition rate than shallow learning. Deep learning can replace the previous shallow learning.

TABLE 3
COMPARISON OF THE RECOGNITION RATES AND TRAINING TIME IN VEHICLE TYPES CLASSIFICATION USING DIFFERENT NEURAL NETWORKS (IMAGE PROCESSED)

Image Size	GRNN		DNN		CNN	
	Recognition Rate (%)	Time (s)	Recognition Rate (%)	Time (s)	Recognition Rate (%)	Time (s)
128*64	74.3	1441	94	1441	94.3	15359
64*32	74	4	91	792	93.72	6104
32*16	74	1	88	367	92.53	1976

The DNN, CNN and GRNN picture sizes were changed to see if the recognition rate and training time changed, as shown in Table 3. Changing the picture size did not change the recognition rate for the GRNN. However, the training time was effectively shortened. Changing the picture size of DNN and CNN, produced a slight change in the recognition rate and the training time was also effectively shortened. The DNN and CNN recognition rate decreased probably because the picture size change reduced the picture features, resulting in reducing the deep learning recognition rate. The overall recognition rate was better than that for the GRNN.

V. CONCLUSION

A vehicle type classification system using neural networks is proposed in this study. The study results were obtained in two parts. First, the image performance with image processing technology produced better results than using only original pictures for the deep and shallow learning input. Secondly, changes in the picture size influenced both shallow and deep learning. The results show that vehicle classification using the CNN is better than that from the DNN using the restricted Boltzmann machine. The two types of neural networks produced much higher results than the GRNN. It also shows that the deep learning technique is better than the shallow learning in this study. The results demonstrated that the proposed system is effective for vehicle type classification. Future research will be focused on the development of real time traffic condition classification using an image recorder.

REFERENCES

- [1] M. Adnan, "Passenger car equivalent factors in heterogeneous traffic environment-are we using the right numbers?" *Procedia Engineering, Fourth International Symposium on Infrastructure Engineering in Developing Countries*, vol. 77, pp. 106-113, 2014.
DOI: 10.1016/j.proeng.2014.07.004
- [2] S. Aneeka and Z. W. Zhong, "NO_x and CO₂ emissions from current air traffic in ASEAN region and benefits of free route airspace implementation," *Journal of Applied and Physical Sciences*, vol. 2, no. 2. pp. 32-36, 2016.
DOI: 10.20474/japs-2.2.1
- [3] G. E. Hinton, S. Osindero and Y. W. Teh, "A fast learning algorithm for deep belief nets," *Neural Computation*, vol. 18, no. 7, pp. 1527-1554, 2006.
DOI: 10.1162/neco.2006.18.7.1527
- [4] P. Smolensky, "Information processing in dynamical systems: Foundations of harmony theory," in *Parallel distributed processing: Explorations in the microstructure of cognition*, D. E. Rumelhart, J. L. McClelland and The PDP Research Group, Eds. Cambridge, MA: MIT Press, 1986.
- [5] A. R. Mohamed, G. Dahl and G. Hinton, "Deep belief networks for phone recognition," in *Proceedings of Nips workshop on deep learning for speech recognition and related applications*, Whistler, Canada, 2009.
- [6] G. E. Dahl, D. Yu, L. Deng and A. Acero, "Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 20, no. 1, pp. 30-42, 2012. **DOI:** 10.1109/tasl.2011.2134090
- [7] G. Hinton, L. Deng, D. Yu, D. G. E. Dahl, A. R. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. N. Sainath and B. Kingsbury, "Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups," *IEEE Signal Processing Magazine*, vol. 29, no. 6, pp. 82-97, 2012.
DOI: 10.1109/msp.2012.2205597
- [8] Q. V. Le, "Building high-level features using large scale unsupervised learning," in *IEEE International Conference on Acoustics, Speech and Signal Processing*, Vancouver, Canada, 2013.

- [9] D. C. Cireşan, U. Meier, J. Masci, L. M. Gambardella and J. Schmidhuber, "Flexible, high performance convolutional neural networks for image classification," in *Proceedings-International Joint Conference on Artificial Intelligence*, San Jose, CA, 2011. DOI: [10.1109/ijcnn.2011.6033458](https://doi.org/10.1109/ijcnn.2011.6033458)
- [10] R. Suryanita, H. Maizir and H. Jingga, "Prediction of structural response based on ground acceleration using artificial neural networks," *International Journal of Technology and Engineering Studies*, vol. 3, no. 2, pp. 74-83, 2017. DOI: [10.20469/ijtes.3.40005-2](https://doi.org/10.20469/ijtes.3.40005-2)
- [11] N. A. Zainuddin, I. Norhuda, I. S. Adeib, A. N. Mustapa and S. H. Sarijo, "Artificial neural network modeling ginger rhizome extracted using rapid expansion Super-Critical Solution (RESS) Method," *Journal of Advances in Technology and Engineering Research*, vol. 1, no. 1, pp. 1-14, 2015. DOI: [/10.20474/jater1.1.1](https://doi.org/10.20474/jater1.1.1)
- [12] N. Z. Tawfeeq Abdulnabi and O. Altun, "Batch size for training convolutional neural networks for sentence classification," *Journal of Advances in Technology and Engineering Studies*, vol. 2, no. 5, pp. 156-163, 2016. DOI: [10.20474/jater-2.5.3](https://doi.org/10.20474/jater-2.5.3)
- [13] H. P. Martinez, Y. Bengio and G. N. Yannakakis, "Learning deep physiological models of affect," *IEEE Computational Intelligence Magazine*, vol. 8, no. 2, pp. 20-33, 2013. DOI: [10.1109/mci.2013.2247823](https://doi.org/10.1109/mci.2013.2247823)
- [14] D. Cireş An, U. Meier, J. Masci and J. Schmidhuber, "Multi-column deep neural network for traffic sign classification," *Neural Networks*, vol. 32, pp. 333-338, 2012. DOI: [10.1016/j.neunet.2012.02.023](https://doi.org/10.1016/j.neunet.2012.02.023)
- [15] M. M. Reza and M. H. U. Ahmed, "A traffic transmission scheme for smart grid communication," *International Journal of Technology and Engineering Studies*, vol. 2, no. 6, pp. 164-171, 2016. DOI: [10.20469/ijtes.2.40001-6](https://doi.org/10.20469/ijtes.2.40001-6)
- [16] U. Iqbal, S. W. Zamir, M. H. Shahid, K. Parwaiz, M. Yasin and M. S. Sarfraz, "Image based vehicle type identification," in *Proceedings of the International Conference on Information and Emerging Technologies*, Karachi, Pakistan, 2010. DOI: [10.1109/iciet.2010.5625675](https://doi.org/10.1109/iciet.2010.5625675)
- [17] Z. Chen, T. Ellis and S. A. Velastin, "Vehicle type categorization: A comparison of classification schemes," in *International IEEE Conference Intelligent Transportation Systems*, Washington, WA: DC, 2011. DOI: [10.1109/itsc.2011.6083075](https://doi.org/10.1109/itsc.2011.6083075)
- [18] Y. Peng, J. S. Jin, S. Luo, M. Xu and Y. Cui, "Vehicle type classification using PCA with self-clustering," in *IEEE International Conference on Multimedia and Expo Workshops*, Melbourne, Australia, 2012. DOI: [10.1109/icmew.2012.73](https://doi.org/10.1109/icmew.2012.73)
- [19] X. B. Zhang and L. Jiang, "Vehicle types recognition based on neural network," in *International Conference on Computational Intelligence and Natural Computing*, Wuhan, China, 2009. DOI: [10.1109/cinc.2009.146](https://doi.org/10.1109/cinc.2009.146)
- [20] B. M. Ozyildirim and M. Avci, "Generalized classifier neural network," *Neural Networks*, vol. 39, pp. 18-26, 2013. DOI: [10.1016/j.neunet.2012.12.001](https://doi.org/10.1016/j.neunet.2012.12.001)
- [21] G. E. Hinton "A practical guide to training restricted boltzmann machines," in *Neural networks: Tricks of the trade*, G. Montavon, Eds. Berlin, Germany, Springer, 2012. DOI: [10.1007/978-3-642-35289-8_32](https://doi.org/10.1007/978-3-642-35289-8_32)
- [22] A. Krizhevsky, I. Sutskever and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Conference of Advances in Neural Information Processing Systems*, Lake Tahoe, CA, 2012.

— This article does not have any appendix. —