Classification of truck body types using a deep transfer learning approach

1st Reza Vatani Nezafat
Department of Civil & Environmental Engineering
Old Dominion University
Norfolk, Virginia, USA
rvata001@odu.edu

2nd Behrouz Salahshour
Department of Civil & Environmental Engineering
Old Dominion University
Norfolk, Virginia, USA
bsala001@odu.edu

3rd Mecit Cetin
Department of Civil & Environmental Engineering
Old Dominion University
Norfolk, Virginia, USA
MCetin@ odu.edu

Abstract—**Classification of vehicles is one of the most important tasks in intelligent transportation systems (ITS). While there are various types of sensors for measuring vehicle characteristics, this paper is focused on an image-based vehicle classification system. Most traditional approaches for image-based vehicle classification are computationally extensive and typically require a large amount of data for model training. This paper investigates whether it is possible to transfer the learning of a highly accurate pre-trained model for classifying truck images based on body type. Results show that using a pre-trained model to extract low-level features of images increases the accuracy of the model significantly, even with a relatively small size of training data. Furthermore, a convolutional neural network (CNN) is shown to outperform other types of models to classify trucks based on the extracted features.**

Keywords—Vehicle classification, Convolutional neural network, Resnet152, Support vector machines, Transfer learning

I. INTRODUCTION

Most traffic management methods are highly dependent on the accuracy of the sensing mechanism and estimation of traffic parameters. Monitoring traffic flow is necessary for managing the performance of traffic operations. In particular, classifying vehicles into distinct categories (e.g., in the USA FHWA's 13-class scheme) is essential for freight planning, highway design and maintenance, traffic operations, and system management. There are already a number of vehicle detection technologies, such as magnetic loop detectors [1], acoustic sensors [2], lasers [3], radar [4], and image/video [5] in place to detect vehicle axles and other physical characteristics needed for a classification algorithm. Vision-based systems are one of the least disruptive methods for monitoring traffic and have relatively a low cost of maintenance in comparison to other technologies.

The focus of this paper is to investigate how deep neural networks can be designed and employed to classify vehicles based on image data. Rather than solving the typical classification problem of categorizing vehicles (e.g., into cars, small trucks, large trucks, etc.), this study is on detecting truck body types. More specifically, the main objective is to distinguish between two types of trailers/trucks: a truck carrying an intermodal container versus a dry or enclosed van. In most urban areas in the USA, especially those with intermodal ports, these two body types constitute a large percentage of all FHWA Class 9 trucks. Compared to other body types, such as a tank, dump trailer, and auto transporter, these selected two body types are more challenging to classify due to the higher similarity in their shapes and sizes. Being able to classify truck body types is important for freight planning, and commodity flow modeling since body configuration can be linked to many concepts of traffic management such as the types of commodity hauled or stochastic capacity estimation [6, 7]. While there is very limited literature on truck body classification, researchers have investigated the general vehicle classification problem based on image data using various techniques. Support vector machines (SVM) are commonly used in many studies for image classification. Some studies have used high dimensional histograms, to train a SVM [8]. More recently, researchers use histogram of oriented gradient to train a nonlinear SVM with a Gaussian kernel function for classification of vehicle images [9]. In their model, the vehicle images are classified into four categories, motorcycle, car, lorry, and background. Another approach to classification is general active-learning framing which has achieved high accuracy (90%), high recall, and good localization. Their model was applied to static images and roadway video data captured under different traffic conditions (traffic variety, road illumination, weather conditions) [10]. Another popular way to do this classification task is to use statistical models such as the hybrid dynamic Bayesian network [11] which is able to obtain high classification accuracy using low-level features (height, width, and angle). The model classifies an image into one of four classes considered: sedan, pick-up truck, SUV/minivan, and unknown.

Recent advancements in computational power and graphical processing units have increased the performance of machine learning methods significantly. Computers have achieved superior performance for tasks such as image retrieval [12], object detection, and tracking [13, 14]. Convolutional Neural Network (CNN) classification is one of the most widely used machine-learning methods in current research. The advantage of CNN-based image classification is the sophisticated network structure that gives it the ability to perform feature extraction and selection automatically from large-scale training data. High classification performance is obtained by learned feature representation in CNN models. It describes the property of different categories much better than other approaches. The CNN needs a tremendous amount of data to optimize the numerous parameters of its network structure [15]. Recently, many researchers in the field of transportation engineering have begun using CNN classification. For instance, Kafai et al. [16] have applied a deep CNN to automatically detect cracks on hot-mix asphalt and portland cement concrete using surfaced pavement images. In another study, to extract variable-scale features for vehicle detection, a hybrid deep CNN is developed and applied to satellite images [17]. They have divided the maps of the last convolutional layer and the max-pooling layer of a deep CNN into multiple blocks of variable receptive field sizes or max-pooling field sizes. It was shown that the proposed model is able to extract variable-scale features and outperform simple deep CNN.

Because of the limitations in computational resources and the size of training data, the training process of large CNNs is time-consuming and easily results in overfitting. Since the CNN model requires huge amounts of data for training, some studies have tried a pre-training and fine-tuning approach to overcome this problem [18]. A GoogleNet model [19] was pre-trained on the ILSVRC-2012 dataset to obtain the initial model. Then the initial model was fine-tuned on their vehicle dataset containing 13,700 images extracted from surveillance cameras. They have reached 98.26% accuracy for classification. Although this approach lessens the overfitting problem, the pre-training is still computationally intensive. Usually, low-level features are very similar to various images, so it is intuitive to keep low-level features learned from one dataset and transfer it for classification of other datasets. Transfer learning saves computational power by partially using the feature descriptor parts of an already existing trained model such as AlexNet [20] but replacing the classifier part with the new task-specific variables. Many researchers [21-24] have used intermediate activation functions learned with pre-trained models on large datasets to improve the accuracy and proficiency of new models with limited training data.

There are several CNN architectures trained on ImageNet [15] that can be used as pre-trained CNNs including CaffeNet [25], GoogleNet [19], VGGNet [26], and AlexNet [20]. One of the common problems that arise in these models is the degradation problem, i.e., both training and testing accuracy begin to unexpectedly degrade as the network depth increases. Recent studies have implemented residual learning to overcome this problem [27, 28].

II. Methodology

The transfer learning method used in this article incorporate a pre-trained ResNet [29] model as a feature descriptor and then feed those features as an input for a simple supervised classifier.

A. *Pre-trained ResNet*

ResNet proposes that residual learning blocks be added for solving the degradation problem caused by multiple nonlinear layers. The degradation problem makes it difficult to achieve an identity mapping for a layer, even if it is the optimum. By using residual learning, if the optimal solution for a specific case is closer to an identity mapping (i.e., the output is a slightly altered version of the input), the solvers can reach it by simply driving the weights of the multiple nonlinear layers toward zero. This way, the solver should converge easier by retaining the input rather than learning the function like a new one.

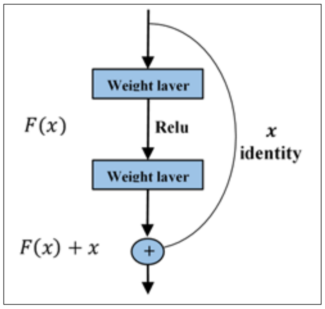

Fig. 1. Identity block

The mathematical formulation of the added residual learning units can be expressed as:

$$y = F(x, \{W_i\}) + x$$

Where x and y are, respectively, the input and output vectors of the layers considered; and the function $F(x, \{W_i\})$ represents the residual mapping to be learned. The architecture of this building block is represented in Fig. 1. The added shortcut solves the degradation problem without introducing extra parameters or computation complexity.

The ResNet architecture used in this article has 151 convolutional layers and a final dense layer with a Softmax activation function. The structure of the model, along with its respective hidden units, is presented in Fig. 2. As it can be seen, a residual learning block is defined for every few stacked layers (yellow boxes). Building blocks are shown in white boxes with the numbers of residual blocks stacked written on the right (i.e., $\times 3$). Down-sampling is done by blocks conv3_1, conv4_1, and conv5_1 with a stride of 2.

B. *Supervised classification*

The supervised classification is one of the most popular subfields of machine learning concept. It is most often used for quantitative analysis to find regions that can be associated with the classes of interest on the spectral domain of the application [30]. There are many different supervised classification algorithms. In this paper, three of popular algorithms have been utilized as a classifier on extracted features of the pre-trained ResNet model.

1. the K-nearest neighbor (KNN) is a non-parametric approach used for supervised classification [31]. When there is a little knowledge about the distribution and space of the dataset, KNN would be one of the popular choices for the classification task. The model finds K-nearest neighbors based on the Euclidean distance between a test sample and labeled training samples. The majority class label of its k nearest training samples would be assigned to the test sample. To avoid ties, K is usually chosen to be odd.

2. The SVM is considered one of the fundamental supervised machine learning approaches that have been used in many studies [32]. The model solves an optimization problem on training data to find the optimal line to separate two classes. By choosing desired kernel function, the classifier line can be

considered linear or nonlinear depending on the complexity of the model.

3. The MLP model is a supervised learning algorithm which consists of an input layer, an output layer, and a multi-layer hidden layer. MLP, unlike most regression algorithms, makes no prior assumption about the distribution of data. Several nonlinear functions are used to train and generalize the unseen input data to predict the objective variable. Moreover, MLP can be used to generate multiple outputs, whereas support vector machines allow only one output [33].

4. *Proposed Model*

The ResNet model pre-trained on the ILSVRC-2015 dataset is used here to determine the image features. Since the model is pre-trained, extractions of the already learned features will save a great amount of computational power. However, features in a CNN grow in complexity as we step deeper into the network. Therefore, the next task is to identify the optimal point at which the ResNet structure should be cut, in order to get the right level of feature complexity for our task. We have tested four different positions in this article for a simple KNN classifier to be inserted in the structure, as shown in Fig. 2. The features are separately and independently extracted from the ResNet model in each of these cases and used as the feature descriptors for the respective KNN classifier. The implemented KNN classifier search for five nearest neighbors for the classification.

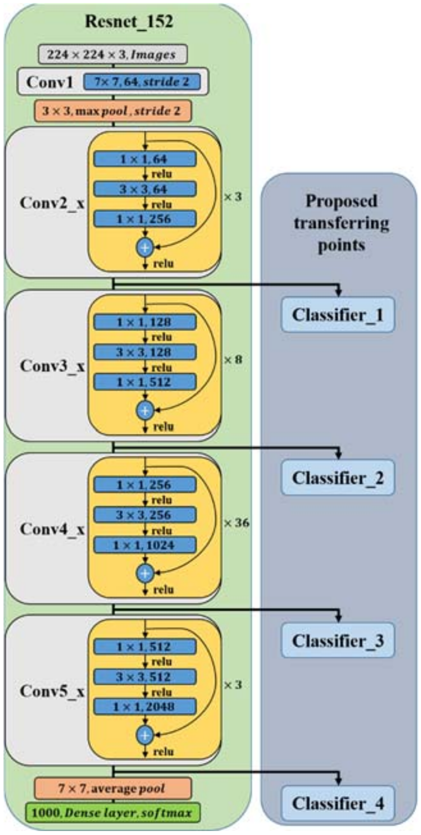

Fig. 2. The architecture of the ResNet-152 with the proposed positions for the classifier

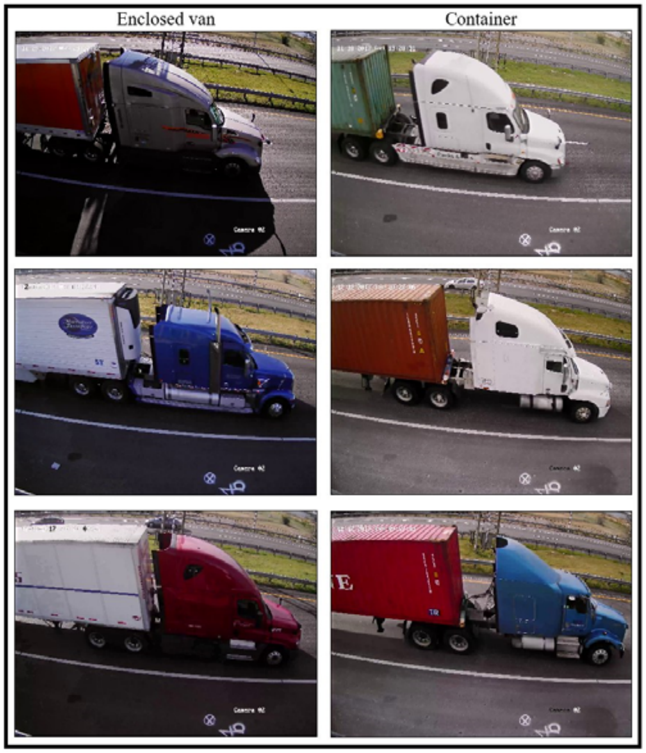

Fig. 3. Dataset sample for truck classification

5. *Dataset and computational configuration*

The data used in this article is collected by a roadside surveillance camera installed at the westbound direction of I-64 near the Hampton Roads Bridge-Tunnel (HRBT). The tested highway has two lanes at this location. There are many categories of truck body configurations that one can consider [6]. In this paper, two of the most challenging categories have been selected for classification, i.e., trailers with containers and enclosed vans. As it can be seen in Fig. 3, the two examined body types are very similar in structure. The total number of truck images used is 1,200 out of which 530 are trailers with containers, and the rest are enclosed vans. The images have been resized to 224 × 224 × 3 to be consistent with ResNet input. 80% of the data has been used for training, and the rest is set aside to be used as the test data. All images are from the same angle, and there are no multiple vehicles in the same image. Ground truth labeling of these images was done manually. All computations in this article were conducted with Tensorflow platform on Windows 7 OS with Intel Xeon E5-2630 2.40 GHz and an NVIDIA Quadro K4200 GPU with 4 GB memory.

TABLE 1. Accuracy for proposed positions for the classifier

Models	Accuracy %
Classifier_1	68.2
Classifier_2	81.7
Classifier_3	84.7
Classifier_4	72.3

III. Results and discussion

Four different placements of KNN classifier, as shown in Fig. 2, have been tested to identify the optimal point at which the ResNet structure should be cut. The accuracy results representing the percentage of correct predictions for each model on the test data are presented in TABLE 1. The Classifier_1 is the worst performing model in predicting the image labels. This is because the features are primary and basic at this level of the network and the Classifier_1 fails to correctly identify the correct truck type based on these features. Examples of possible features at this level will be a color change, the shape of lines, edges, etc. It is evident that it is impossible to identify between a container and an enclosed van using these simplistic features. Features grow in complexity as we go deeper in the network and Classifier_2 will get more complex features from the pre-trained CNN compared to Classifier_1. By the same logic, Classifier_3 and Classifier_4 should be more accurate than their proceeding peers. However, the performance of the last proposed placement for Classifier_4 is lower than Classifier_3. This happens because the features beyond Classifier_3 are becoming adversely complicated for the classifier to distinguish between these two vehicle classes. In other words, there exists an optimal point where the best-suited features for detecting these type of vehicle classes can be accessed. In this case, the first 141 layers of ResNet_152 had the best performance for the feature extraction task.

Three classical approaches of classification were examined to find the best classifier for extracted features. Features extracted from the first 141 layers of ResNet_152 were used to develop MLP, SVM, and KNN models. The implemented MLP has two fully connected layers with 1024 hidden units, the learning rate of 0.001 and 200 epochs of training. The implemented SVM has a radial basis kernel function with the gamma parameter equals to 0.2, and the KNN find 5-nearest neighbors. The accuracy of these models in comparison to the MLP is presented in TABLE 2. The MLP model outperforms both SVM and KNN results.

The convergence of model accuracy for MLP on both training and test data is presented in Fig. 4. An epoch is when all the training samples are used once to update the weights by the optimization algorithm that iteratively improves the model variables (e.g., weights). The accuracy of test data follows approximately the same trend as the accuracy of training data, and after around 100 epochs the model becomes steady. The confusion matrix for the test data is shown in TABLE 3. It is evident that misclassification is a little skewed towards containers.

TABLE 2. The accuracy of different classifiers fo extracted features

Models	Accuracy %
MLP	96.5
SVM	88
KNN	84.7

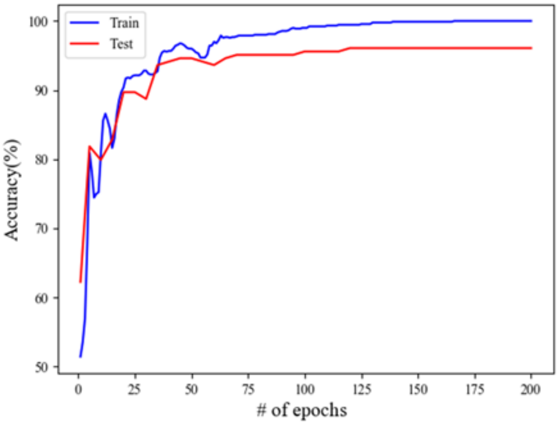

Fig. 4. Convergence of the model accuracy for MLP

TABLE 3. The confusion matrix of MLP model for the test data

		Predicted labels	
		Enclosed van	Container
True labels	Enclosed van	134 (98.5 %)	8 (7.7 %)
	Container	2 (1.5 %)	96 (92.3 %)

IV. Conclusion

In this paper, a transfer learning model is developed for classification of truck body types based on image data. Since the simple features for any type of image dataset are the same, it was possible to transfer features learned by a pre-trained model (ResNet_152) to another classifier and build highly accurate models with small datasets. Two of the most challenging categories of trucks are chosen to train the model. Four different experiments are conducted to find the optimal level of complexity for transferring learned features. It is shown that first 141 layers of the ResNet_152 have the best performance on this dataset. Furthermore, three different classifiers are investigated with transferred features. Results show that MLP outperforms other classifiers with a 96.5% accuracy on the selected test data. A similar strategy can be applied to other categories of trucks, different angles of camera and different weather or visibility conditions. In the future, these complex conditions and additional categories of truck body types will be considered.

Acknowledgment

This research was funded by the Mid-Atlantic Transportation Sustainability University Transportation Center (MATS UTC). The authors also would like to thank Virginia Department of Transportation (VDOT) for assisting in data collection.

REFERENCES

[1] Y. Mita and K. Imazu, "Range-measurement-type optical vehicle detector," 1995.

[2] K. Wang *et al.*, "Vehicle recognition in acoustic sensor networks via sparse representation," in *Multimedia and Expo Workshops (ICMEW), 2014 IEEE International Conference on*, 2014, pp. 1-4: IEEE.

[3] L. Peiyu, T. Dapeng, and L. Boyu, "Embedded flexible assembly system for Car Latch based on laser identification," 2006.

[4] H.-t. Kim and B. Song, "Vehicle recognition based on radar and vision sensor fusion for automatic emergency braking," in *Control, Automation and Systems (ICCAS), 2013 13th International Conference on*, 2013, pp. 1342-1346: IEEE.

[5] Y. Zhou, H. Nejati, T.-T. Do, N.-M. Cheung, and L. Cheah, "Image-based vehicle analysis using deep neural network: A systematic study," in *Digital Signal Processing (DSP), 2016 IEEE International Conference on*, 2016, pp. 276-280: IEEE.

[6] S. V. Hernandez, A. Tok, and S. G. Ritchie, "Integration of Weigh-in-Motion (WIM) and inductive signature data for truck body classification," *Transportation Research Part C: Emerging Technologies,* vol. 68, pp. 1-21, 2016/07/01/ 2016.

[7] S. Sohrabi, A. Ermagun, and R. Ovaici, "Finding Optimum Capacity of Freeways Considering Stochastic Capacity Concept," presented at the 96th Annual Meeting Transportation Research Board, Washington DC, USA, 8-12 January 2017.

[8] Z. Chen, N. Pears, M. Freeman, and J. Austin, "Road vehicle classification using support vector machines," in *Intelligent Computing and Intelligent Systems, 2009. ICIS 2009. IEEE International Conference on*, 2009, vol. 4, pp. 214-218: IEEE.

[9] L. T. Ng, S. A. Suandi, and S. S. Teoh, "Vehicle classification using visual background extractor and multi-class support vector machines," in *The 8th International Conference on Robotic, Vision, Signal Processing & Power Applications*, 2014, pp. 221-227: Springer.

[10] S. Sivaraman and M. M. Trivedi, "A general active-learning framework for on-road vehicle recognition and tracking," *IEEE Transactions on Intelligent Transportation Systems,* vol. 11, no. 2, pp. 267-276, 2010.

[11] M. Kafai and B. Bhanu, "Dynamic Bayesian networks for vehicle classification in video," *IEEE Transactions on Industrial Informatics,* vol. 8, no. 1, pp. 100-109, 2012.

[12] W.-L. Ku, H.-C. Chou, and W.-H. Peng, "Discriminatively-learned global image representation using CNN as a local feature extractor for image retrieval," in *Visual Communications and Image Processing (VCIP), 2015*, 2015, pp. 1-4: IEEE.

[13] R. Girshick, "Fast r-cnn," *arXiv preprint arXiv:1504.08083,* 2015.

[14] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," in *Advances in neural information processing systems*, 2015, pp. 91-99.

[15] O. Russakovsky *et al.*, "Imagenet large scale visual recognition challenge," *International Journal of Computer Vision,* vol. 115, no. 3, pp. 211-252, 2015.

[16] K. Gopalakrishnan, S. K. Khaitan, A. Choudhary, and A. Agrawal, "Deep Convolutional Neural Networks with transfer learning for computer vision-based data-driven pavement distress detection," *Construction and Building Materials,* vol. 157, pp. 322-330, 2017.

[17] X. Chen, S. Xiang, C.-L. Liu, and C.-H. Pan, "Vehicle detection in satellite images by hybrid deep convolutional neural networks," *IEEE Geoscience and remote sensing letters,* vol. 11, no. 10, pp. 1797-1801, 2014.

[18] L. Zhuo, L. Jiang, Z. Zhu, J. Li, J. Zhang, and H. Long, "Vehicle classification for large-scale traffic surveillance videos using Convolutional Neural Networks," *Machine Vision and Applications,* vol. 28, no. 7, pp. 793-802, 2017.

[19] C. Szegedy *et al.*, "Going deeper with convolutions," 2015: Cvpr.

[20] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097-1105.

[21] F. Hu, G.-S. Xia, J. Hu, and L. Zhang, "Transferring Deep Convolutional Neural Networks for the Scene Classification of High-Resolution Remote Sensing Imagery," *Remote Sensing,* vol. 7, no. 11, p. 14680, 2015.

[22] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Transactions on knowledge and data engineering,* vol. 22, no. 10, pp. 1345-1359, 2010.

[23] A. S. Razavian, H. Azizpour, J. Sullivan, and S. Carlsson, "CNN features off-the-shelf: an astounding baseline for recognition," in *Computer Vision and Pattern Recognition Workshops (CVPRW), 2014 IEEE Conference on*, 2014, pp. 512-519: IEEE.

[24] J. Donahue *et al.*, "Decaf: A deep convolutional activation feature for generic visual recognition," in *International conference on machine learning*, 2014, pp. 647-655.

[25] Y. Jia *et al.*, "Caffe: Convolutional architecture for fast feature embedding," in *Proceedings of the*

22nd ACM international conference on Multimedia, 2014, pp. 675-678: ACM.

[26] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556,* 2014.

[27] K. Zhang, W. Zuo, Y. Chen, D. Meng, and L. Zhang, "Beyond a gaussian denoiser: Residual learning of deep cnn for image denoising," *IEEE Transactions on Image Processing,* vol. 26, no. 7, pp. 3142-3155, 2017.

[28] H. Qassim, A. Verma, and D. Feinzimer, "Compressed residual-VGG16 CNN model for big data places image recognition," in *Computing and Communication Workshop and Conference (CCWC), 2018 IEEE 8th Annual*, 2018, pp. 169-175: IEEE.

[29] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770-778.

[30] J. A. Richards, "Supervised classification techniques," in *Remote Sensing Digital Image Analysis*: Springer, 2013, pp. 247-318.

[31] L. E. Peterson, "K-nearest neighbor," *Scholarpedia,* vol. 4, no. 2, p. 1883, 2009.

[32] C. Cortes and V. Vapnik, "Support-vector networks," *Machine learning,* vol. 20, no. 3, pp. 273-297, 1995.

[33] R. C. Deo, M. A. Ghorbani, S. Samadianfard, T. Maraseni, M. Bilgili, and M. Biazar, "Multi-layer perceptron hybrid model integrated with the firefly optimizer algorithm for windspeed prediction of target site using a limited set of neighboring reference station data," *Renewable Energy,* vol. 116, pp. 309-323, 2018/02/01/ 2018.