Automatic Vehicle type Classification with Convolutional Neural Networks

Conference Paper · June 2018

DOI: 10.1109/IWSSIP.2018.8439406

Gustavo H. G. Matsushita
Universidade Federal do Paraná
4 PUBLICATIONS **8** CITATIONS

Automatic vehicle type classification with convolutional neural networks

Max N. Roecker*, Yandre M. G. Costa*, João L. R. Almeida*, Gustavo H. G. Matsushita†
*Departament of Informatics – State University of Maringá, Maringá, Paraná, Brazil
†Departament of Informatics – Federal University of Paraná, Curitiba, Paraná, Brazil
{*max.roecker, jlramalheira, gustavomatsushita*}*@gmail.com, yandre@din.uem.br*

***Abstract*—This paper proposes a convolutional neural network model for classification of vehicle types with low-resolution images from a frontal perspective. This characteristic can be useful to the development of systems with limited resources, like embedded systems. We trained the model as a multinomial logistic regression where cross-entropy of the ground truth labels and the model's prediction estimates the error. To prevent overfitting, we performed data augmentation in the training dataset and regularized the model using the dropout method. Experimental results in a subset of the BIT-Vehicle Dataset with samples uniformly distributed between classes shows that the model achieves an accuracy of 93.90%. We conclude that the model is discriminative and capable of generalizing the patterns of the vehicle type classification task.**

Keywords*—vehicle classification, convolutional neural networks, machine learning, computer vision, pattern recognition.

I. Introduction

Object classification is a large field of research in computer vision and machine learning that aims to classify objects present in images into meaningful categories [1]. In the context of intelligent traffic systems, object classification of vehicles assumes a fundamental role and have a wide range of employment such as traffic surveillance, route optimization, and anomaly detection.

Humans can classify vehicles in images using key aspects such as trademarks, forms, and ornaments. For computer systems, however, the vehicle type classification in images can be a challenging task because image inputs have high-dimensional features [1] and vehicles have a wide variation in form, size, and color. Also, the acquisition of images in the traffic is subject to environmental conditions such as lighting, noising, partial occlusion, and weather.

In the last years, several works proposed methods for vehicle type classification in digital images. Dong et al. [2] group these approaches in two categories: model-based methods and appearance-based methods. Model-based methods address the problem using dimensional attributes of the vehicle, such as length, area, and height, to create a model and classify it [3] [4]. On the other hand, appearance-based methods address the problem by extracting visual features from the vehicle, such as edges, filters and visual descriptors [5] [6].

Every day, traffic surveillance cameras acquire a large number of vehicles' frontal view images. In these circumstances, the use of an appearance-based method is a better alternative, since the model-based method may perform poorly due to the lack of variance of perspectives in viewpoints. However, most of the appearance-based methods use multiple handcrafted features, which cannot efficiently describe the complexity of the patterns for vehicle type classification in images.

In the last years, methods inspired on the biological behavior of the of the mammal's visual cortex were proposed, including the NeoCognitron [7], the HMAX (Hierarchical Model and X) [8] and the CNN (Convolutional Neural Networks) [9]. Convolutional Neural Networks, a specialized kind of neural network for processing data that have spatial interactions, have gained prominence recently due to its high capacity to generalize patterns in images. This characteristic is useful for the vehicle type classification task as it minimizes the problems previously mentioned.

In this paper, we proposed a simple model to classify vehicle images into six types: truck, bus, sedan, microbus, minivan, and SUV; using a low-resolution input and tiny convolutional filters. As a result, using supervised training with high-intensity data augmentation, we achieved state-of-art accuracy on the BIT-Vehicle Dataset [2].

This paper is organized as follows. Section II presentes some related works with its highligths and results. In Section III, we describe the model including the architecture and configurations. We present the methodology of the training and evaluation in Section IV and in Section V the results are showed and compared with others models that were evaluated in the original or modified version of the BIT-Vehicle Dataset. Section VI concludes the paper.

II. Related Works

Convolutional networks [9], also known as Convolutional Neural Networks (CNN), are a specialized kind of neural network for processing data that have spatial interactions. The "convolutional" name part indicates that the network employs a convolution operation in place of general matrix multiplication in at least one of its steps [10].

In recent years, the convolutional networks have been used to address the vehicle type classification task, but with distinct datasets and approaches. Dong et al. [2] propose a vehicle type classification method using a semi-supervised convolutional network from vehicle frontal-view image of the BIT-Vehicle Dataset, also introduced in [2]. The architecture of the model consists of two convolutional stages, and each stage contains a convolution, a non-linearity absolute rectification, a local contrast normalization, and average pooling. The input of the first stage is the image, and the output of the first stage is the input of the second stage. The fully connected stage takes

978-1-5386-6979-2/18/$31.00 ©2018 IEEE

as input the fusion of the outputs of the first and the second stages. In the end, the model outputs the probability of each of the six vehicle types: Bus, Microbus, Sedan, SUV, and Truck. To achieve an accuracy of 88.11%, Dong et al. [2] also employ a Laplacian Filter to obtain the initial value for the kernels of the network with large amounts of unlabeled data.

Selbes and Sert [11] address the vehicle type classification task using a multimodal method from videos of traffic scenarios, extracting both image's and audio's features and fusing it to feed a Support Vector Machine (SVM) multiclassifier. To extract the image-based features, the authors use the trained versions of the renowned convolutional network's architectures AlexNet [12] and GoogleNet [13]. Mel-frequency Cepstral Coefficients (MFCCs) are used to extract audio-based features from the video. The SVM then classify each video snippet as an armored vehicle, a construction vehicle, a crane vehicle, an emergency vehicle, a military vehicle, a motorcycle, and a rescue vehicle. This multimodal method achieves 72.1% accuracy.

Kim and Lim [14] propose a new scheme of vehicle type classification for multi-view images of surveillance cameras using convolutional networks with data augmentation, bootstrap aggregating (bagging) and, a post-processing voting between the models of the bagging method. The model consisted of seven independently trained convolutional networks with the same characteristics that output a prediction by voting. Inspired by the works of Simonyan and Zisserman [15], the authors modeled all the convolutional networks very deep with fifteen convolutional layers. The model was evaluated in a subset of the ImageNet Dataset with eleven classes: articulated truck, background (negative examples), bicycle, bus, car, motorcycle, non-motorized vehicle, pedestrian, pickup truck, single-unit truck and work van; achieving an accuracy of 97.84%.

III. Model description

Motivated by the works of Cireşan et al. [16], Krizhevsky et al. [12] and Simonyan and Zisserman [15], we designed all the stages of the model with the same principles to simplify the model and minimize setup of hyperparameters. In this section, firstly we present the specific configurations and then detail the layout of the architecture of the model (Subsection III-A). The design choices are discussed and compared with other models in Subsection III-B.

A. Configurations

The model receives as input an RGB image with 32 pixels in width and 32 pixels in height, i.e., the input is a tensor with the shape of $32 \times 32 \times 3$. The input pass through a stack of convolutional layers, where a variable number of filters with a size of 3×3, the smallest size to capture a notion of direction. We fixed the convolution stride at one point and preserved the spatial padding, i.e., the output of the convolution has the same size as the input by adding a zero-valued border in the input.

The output of the convolution is set up with the leaky rectifier activation function (LReLU) instead of the traditional rectifier (ReLU). Maas et al. [17] demonstrated that in some cases the ReLU activation could "kill" some neurons when all of its weights become zero and cannot activate. A LReLU activation $\iota : \mathbb{R} \to \mathbb{R}$ is defined as $\iota(x) = \max(x, \alpha x)$ where $\alpha \in \{x \in \mathbb{R} \mid 0 < x < 1\}$. In this model, we fixed $\alpha = 0.01$ as an hyperparameter. A spatial pooling finalizes each stack of convolutional layers. The spatial pooling performs a maximum-value subsampling over a 2×2 pixel window with a stride of 2.

TABLE I. Architecture of model.

#	Layer type	Input units	Parameters units	Stride (x, y)
1	Convolutional	$32 \times 32 \times 3$	$3 \times 3 \times 32$	$(1, 1)$
2	Convolutional	$32 \times 32 \times 32$	$3 \times 3 \times 32$	$(1, 1)$
3	Pooling	$32 \times 32 \times 32$	2×2	$(2, 2)$
4	Convolutional	$16 \times 16 \times 32$	$3 \times 3 \times 64$	$(1, 1)$
5	Convolutional	$16 \times 16 \times 64$	$3 \times 3 \times 64$	$(1, 1)$
6	Pooling	$16 \times 16 \times 64$	2×2	$(2, 2)$
7	Fully-connected	4096	512	–
8	Fully-connected	512	512	–
9	Fully-connected	512	6	–
10	Softmax	6	6	–

A stage of three fully-connected layers follows the end of the convolutional stage. The fully connected layers have the structure similar to multi-layer perceptron (MLP) that receives as input the result of the convolutional stage. The first two layers have 512 units and the third one, as it performs the classification, has 6 units. All the fully-connected layers are also set up with a LReLU with $\alpha = 0.01$.

The last stage of the model takes the fully-connected stage's output applies a normalized exponential function (softmax) and "squashes" a vector of arbitrary real values into probabilities that add up to one.

Table I describes the model architecture evaluated in this work. The column "Input units" displays the number of units that the layer receives and the "Parameters units" presents the number of parameters (weights) in the layer. Both columns are in the "width × height × depth" format at convolutional layers and in the "width × height" format at pooling layers. The "Stride" column shows the stride of the convolution and pooling operations in the (x, y) axis.

B. Discussion

The model presented in this paper is distinct from the model used by Dong et al. [2] to classify the BIT-Vehicle Dataset. Firstly, we employed small kernel sizes throughout the whole model rather than using large filters (e.g., 9×9 with $(1, 1)$ stride as used by Dont et al. [2]). Our model is also serial, i.e., each the layer only takes as input the output of the previous layer. There is no fusion nor parallelization of convolutional feature maps.

A stack of two convolutional layers (without spatial pooling) has an effect of a 5×5 filter, as can be seen in the model. It is preferable to use a stack of two 3×3 convolutional layers instead of a 5×5 because it can be incorporated two non-linear rectifications instead of one, which makes the decision function more discriminative [16] [15]. The model also does not employ a Local Response Normalization (LNR) [12] since its use does not improve the performance and it leads to increased memory consumption and computation time [15].

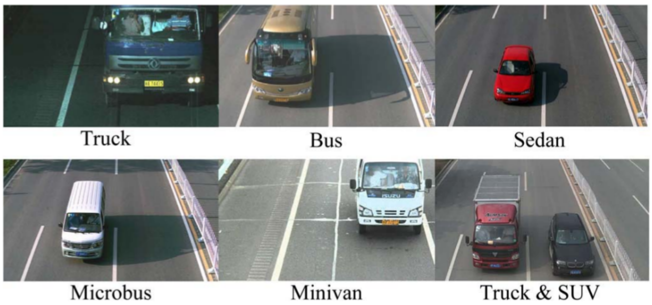

Fig. 1. Some samples of the BIT-Vehicle dataset. All vehicles in the dataset fall into one of six types: Bus, Microbus, Minivan, Sedan, SUV, and Truck. Source: [2]

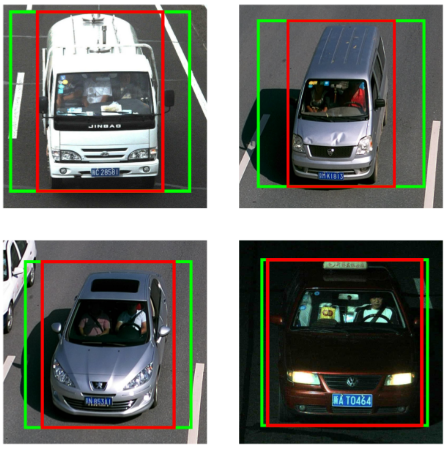

Fig. 2. The image crop approaches in the BIT-Vehicle Dataset. The red square represents the dataset annotation of the vehicle detection. The green squared is the actually sample used as input in the model.

IV. Evaluation Methodology

In this section, we describe some details about the dataset used in the evaluation of the model created (Subsection IV-A), the training procedure and the methodology of the tests (Subsection IV-B) and, some technical details of the implementation (Subsection IV-C).

A. Dataset and Preprocessing

The BIT-Vehicle Dataset [2] is a challenging dataset composed of 9,850 vehicle images in high resolution (1600×1200 px and 1920×1080 px). Figure 1 shows some samples of the dataset. The images are in a wide range of changes in illumination, scale, surface color and position of the vehicles.

Each image may contain more than one vehicle, and so the dataset also contains the annotation of each bounding box of each vehicle in the image. Since the input of the model must receive an image with a 1:1 aspect ratio, each sample was cropped off the image mapped by a square bounding box sized equal to the most significant dimension of the database's bounding box annotation: height or width. The centroid of the annotation box serves as the centroid of the square box. Figure 2 illustrates the differences between the bounding boxes of a vehicle given by the annotation in the dataset (in red) in contrast to the actual crop that will feed the model (in green). This approach has a downside of adding more background information into the sample since the bounding box rarely has the aspect ratio of 1:1. However, it minimize distortions in the input that can lead to a wrong generalization of the model.

All the samples in the dataset are labeled according to one of six classes: Bus, Microbus, Minivan, Sedan, SUV, and Truck. The numbers of samples per class are 558, 883, 476, 5,922, 1,392, and 822, respectively. As can be seen, the distribution probability of the classes is far from uniform, with a coefficient of variation almost equal to 1.27. To meet the uniformity in the dataset, we create a subset of 476 randomly selected samples of each class; totaling 2,856. All the samples selected were resized to a 32×32 RGB image. Then samples were uniformly distributed into three sets: the training, testing, and validation sets containing proportions of 65%, 30%, and 5%, respectively. Right before entering the model, we also applied a normalization of the values of all image channels and transforming it from an interval from $[0, 255]$ to $[0, 1]$. No contrast normalization or mean subtraction was performed.

B. Training and testing

The training procedure follows Krizhevsky et al. [12] and Simonyan and Zisserman [15]. Since all the classes are mutually exclusive, the training consists of optimising a multinomial logistic regression (softmax regression). The error of the model was defined as the cross-entropy of the prediction and the label of the sample. Thus, let $\mathbf{y}$ be the a label of the dataset and $\hat{\mathbf{y}}$ a prediction of the model, the cross-entropy of $\mathbf{y}$ and $\hat{\mathbf{y}}$, denoted as $H(\hat{\mathbf{y}}, \mathbf{y})$, is defined as $H(\hat{\mathbf{y}}, \mathbf{y}) = -\mathbf{y} \cdot \log(\hat{\mathbf{y}})$, where $\cdot$ is a dot product.

The training was regularized by dropout regularization in the first two fully connected layers. According to Srivastava et al. [18], consider a neural network with L hidden layers. Let $l \in \{1, 2, \ldots, L\}$ index the hidden layers of the network. Let $\mathbf{z}^{(l)}$, $\mathbf{y}^{(l)}$, W^l and $\mathbf{b}^{(l)}$ denote the vector of inputs, the vector of outputs, weights and the vector of biases of layer l, respectively; where $\mathbf{y}^{(0)} = \mathbf{x}$. The feed-forward operation of a standard neural network can be described as

$$z_i^{(l+1)} = \mathbf{w}_i^{(l+1)} \mathbf{y}^{(l)} + b_i^{(l+1)},$$

$$y_i^{(l+1)} = f(z_i^{(l+1)})$$

for $l \in \{0, 1, \ldots, L-1\}$ and for any i hidden unit where f is an activation function. With dropout regularization, the feed-forward operation becomes

$$r_j^{(l)} \sim \text{Bernoulli}(p)$$

$$\tilde{\mathbf{y}}^{(l)} = \mathbf{r}^{(l)} \circ \mathbf{y}^{(l)}$$

$$z_i^{(l+1)} = \mathbf{w}_i^{(l+1)} \tilde{\mathbf{y}}^{(l)} + b_i^{(l+1)}$$

$$y_i^{(l+1)} = f(z_i^{(l+1)})$$

where $\circ$ denotes the Hadamard product of the tensors. For any layer l, $\mathbf{r}^{(l)}$ is a vector of independent Bernoulli random

978-1-5386-6979-2/18/$31.00 ©2018 IEEE

variables each of which has a probability p of being 1. This vector is sampled and multiplied element-wise with the outputs of that layer, $\mathbf{y}^{(l)}$, to create a thinned output $\tilde{\mathbf{y}}^{(l)}$ that the next layer uses as input. This process is applied to each layer. During the training phase, we fixed $p = 0.5$ as a hyperparameter.

The optimization approach consists of mini-batch gradient descent with the Adam Algorithm [19]. The batch size was set to 128 and the exponential decay rates β_1 and β_2 were set to 0.9 and 0.999. The learning rate was set at 10^{-3} and do not decay in the training procedure. It stopped the training process after 64 epochs.

Since the dataset used to evaluate the model was subsampled from the original, to prevent overfitting, we applied a data augmentation in the training set. The data augmentation consisted of adding random variations in each image including translation, rotation, blur, sharpening, brightness, and saturation. Figure 3 shows some examples of samples after the augmentation process. Each one of the samples generates 15 new augmented samples, thus increasing the total number of samples in training set from 1,860 to 29,760.

The initialization of the parameters in a neural network model is important since inaccurate initialization can lead to stall the optimizer due to the instability of the gradient in training nets. For the convolutional layers, we followed the initialization scheme proposed by He et. al [20], where the kernels' weights were initialized with values drawn from a distribution $i_c \sim \mathcal{N}(0, \sigma_c^2)$, a normal distribution with mean equal to zero and variance $\sigma_c^2 = {}^2\!/_n$ where n is the number of the input units. Differently, for all the fully-connected layers, we followed the initialization proposed by Glorot and Bengio [21], where the units' weights were initialized with values sampled from a distribution $i_f \sim \mathcal{N}(0, \sigma_f^2)$, where $\sigma_f^2 = {}^2\!/_m$ and m is the sum of the number of units in the input and output of the layer. All the initial values of bias parameters were zero.

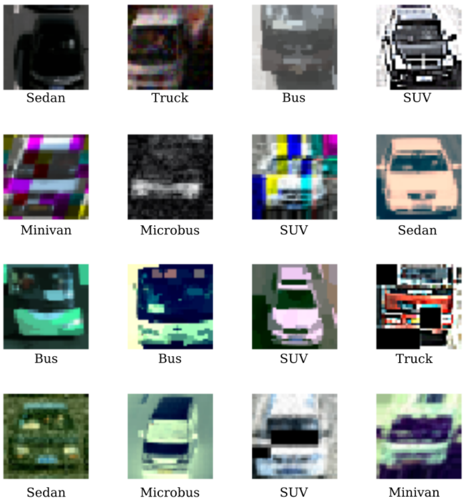

Fig. 3. Some resized samples of the augmented BIT-Vehicle dataset. There are a random number of variations in each image such as translation, rotation, blur, sharpening, brightness, and saturation.

TABLE II. MODEL'S CONFUSION MATRIX ON THE BIT-VEHICLE DATASET

Predicted Actual	Bus	Microbus	Minivan	SUV	Sedan	Truck
Bus	138	0	0	0	0	0
Microbus	0	134	3	0	4	2
Minivan	2	2	135	0	2	4
SUV	0	2	0	135	12	1
Sedan	1	3	1	7	123	0
Truck	1	1	3	0	1	135

TABLE III. PRECISION, RECALL, AND F-MEASURE BY EACH CLASS.

	Bus	Microbus	Minivan	SUV	Sedan	Truck
Precision (%)	97.18	94.37	95.07	95.07	86.62	95.07
Recall (%)	100.00	93.71	93.10	90.00	91.11	95.74
F-Measure (%)	98.57	94.04	94.08	92.47	88.81	95.41

C. Implementation details

The implementation of the model used the open-source software library TensorFlow$^{\text{TM}}$[1], allowing the model to perform computation in heterogeneous environments, such as CPUs or GPUs. The data augmentation was made with the open-source computer vision libraries OpenCV[2] and scikit-image[3]. The model was trained on a system equipped with one NVIDIA GeForce GTX 1050ti with 4 GB, and it took 4–5 hours to converge.

V. RESULTS

In this section, we present the results of vehicle image classification in the test dataset described in Section V-A. We also compare the results with other related works in Section V-B.

A. Results on the BIT-Vehicle Dataset

The results of the model in the testing sets achieved 93.90% accuracy. The confusion matrix is presented in Table II. From the matrix, the SUV and Sedan classes holds most of the misclassifications due to these type of vehicle have considerably similar appearances. This characteristic is also present in the works of Dong et al. [2], but in that case, is the "SUV" class that presents the lower accuracy rates. The model is capable of precisely classify vehicle in images in challenging conditions, and the convolutional network can adjust its parameters to the augmented training set. The results of the Precision, Recall and F-Measure by class can be seen in the Table III.

B. Comparison of the results

When comparing the performance of the proposed model with other models' results evaluated with an original or modified version of the BIT-Vehicle Dataset, summarized in Table IV, our model achieves a better accuracy rate than the methods enumerated.

[1] https://www.tensorflow.org/

[2] https://opencv.org/

[3] http://scikit-image.org/

TABLE IV. Comparison between our model's results and other methods results in the BIT-Vehicle Dataset

Methods	Accuracy (%)	Method
Santos, Souza & Marana [22]	80.62	Boltzmann Machine
Başer & Altun [23]	81.83	Haar Cascade Classifier
Dong *et al.* [2]	88.11	Convolutional Network
Sun *et al.* [24]	90.10	KNNPC + DSRC
Bai, Liu & Yao [25]	91.08	Support Vector Machine
Ours	**93.90**	**Convolutional Network**

We conjecture that the use of a deep convolutional network can optimize its parameters to learn discriminative and reliable features for the vehicle type classification even in low-resolution images. In addition to the depth of the network, the heavy usage of data-augmentation and other regularization techniques prevent the model to overfit and lead it to generalize better, decreasing the error.

VI. Conclusion

In this paper we proposed a model for vehicle type classification from frontal view images by using a convolutional neural network. The convolutional stage of the model takes an low resolution image of the vehicle as input and outputs adjusted features as input for the fully-connected standard network, which outputs a probability of each class of the vehicle.

As demonstrated by the experimental results on the BIT-Vehicle Dataset, the parameters of the model adjusted by the network are discrimative and generalize well even for low-resolution images, showing the efficiency of the proposed model. This characteristic can be useful for its usage, after training, into an embedded intelligent traffic system with low computation power available, such as intelligent traffic lights, traffic signs or road radars; and thus improving the response time and the decisions performed by the system.

References

[1] R. Szeliski, *Computer Vision: Algorithms and Applications*, 1st ed. New York, NY, USA: Springer-Verlag New York, Inc., 2010.

[2] Z. Dong, Y. Wu, M. Pei, and Y. Jia, "Vehicle type classification using a semisupervised convolutional neural network," *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 4, pp. 2247–2256, August 2015.

[3] S. Gupte, O. Masoud, R. F. K. Martin, and N. P. Papanikolopoulos, "Detection and classification of vehicles," *IEEE Transactions on Intelligent Transportation Systems*, vol. 3, no. 1, pp. 37–47, March 2002.

[4] Z. Zhang, T. Tan, K. Huang, and Y. Wang, "Three-dimensional deformable-model-based localization and recognition of road vehicles," *IEEE Transactions on Image Processing*, vol. 21, no. 1, pp. 1–13, January 2012.

[5] P. Ji, L. Jin, and X. Li, "Vision-based vehicle type classification using partial gabor filter bank," in *IEEE International Conference on Automation and Logistics*, August 2007, pp. 1037–1040.

[6] M. Jiang and H. Li, *Vehicle Classification Based on Hierarchical Support Vector Machine*. Springer International Publishing, 2014, pp. 593–600.

[7] K. Fukushima, "Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position," *Biological Cybernetics*, vol. 36, no. 4, pp. 193–202, April 1980.

[8] T. Serre, L. Wolf, S. Bileschi, M. Riesenhuber, and T. Poggio, "Robust object recognition with cortex-like mechanisms," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 3, pp. 411–426, January 2007.

[9] Y. Lecun, *Generalization and network design strategies*. Zurich, Switzerland: Elsevier, 1989.

[10] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016, http://www.deeplearningbook.org.

[11] B. Selbes and M. Sert, "Multimodal vehicle type classification using convolutional neural network and statistical representations of MFCC," in *14th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, Lecce, Italy, August 2017, pp. 1–6.

[12] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems 25*, F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, Eds. Curran Associates, Inc., 2012, pp. 1097–1105.

[13] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Computer Vision and Pattern Recognition (CVPR)*, Boston, MA, USA, June 2015.

[14] P. K. Kim and K. T. Lim, "Vehicle type classification using bagging and convolutional neural network on multi view surveillance image," in *IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, Honolulu, HI, USA, July 2017, pp. 914–919.

[15] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *International Conference on Learning Representations*, May 2015.

[16] D. C. Cireşan, M. Ueli, J. Masci, L. M. Gambardella, and J. Schmidhuber, "Flexible, high performance convolutional neural networks for image classification," in *Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence - Volume Volume Two*. Barcelona, Catalonia, Spain: AAAI Press, 2011, pp. 1237–1242.

[17] A. L. Maas, A. Y. Hannun, and A. Y. Ng, "Rectifier nonlinearities improve neural network acoustic models," in *ICML Workshop on Deep Learning for Audio, Speech and Language Processing*, 2013.

[18] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *Journal of Machine Learning Research*, vol. 15, pp. 1929–1958, 2014.

[19] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *International Conference on Learning Representations*, May 2015.

[20] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification," in *2015 IEEE International Conference on Computer Vision (ICCV)*, December 2015, pp. 1026–1034.

[21] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, ser. Proceedings of Machine Learning Research, Y. W. Teh and M. Titterington, Eds., vol. 9. Chia Laguna Resort, Sardinia, Italy: PMLR, May 2010, pp. 249–256.

[22] D. F. S. Santos, G. B. de Souz, and A. N. Marana, "A 2d deep boltzmann machine for robust and fast vehicle classification," in *2017 30th SIBGRAPI Conference on Graphics, Patterns and Images (SIBGRAPI)*, Niterói, RJ, Brazil, October 2017, pp. 155–162.

[23] E. Başer and Y. Altun, "Detection and classification of vehicles in traffic by using haar cascade classifier," in *Proceedings of 58th ISERD International Conference*, Prague, Czech Republic, December 2016, pp. 19–22.

[24] W. Sun, X. Zhang, S. Shi, J. He, and Y. Jin, "Vehicle type recognition combining global and local features via two-stage classification," *Mathematical Problems in Engineering*, vol. 2017, November 2017.

[25] S. Bai, Z. Liu, and C. Yao, "Classify vehicles in traffic scene images with deformable part-based models," *Machine Vision and Applications*, pp. 1–11, November 2017.

978-1-5386-6979-2/18/$31.00 ©2018 IEEE