Espinosa, J.E., Velastin, S.A. y Branch, J.W. (2017). Vehicle Detection Using Alex Net and Faster R-CNN Deep Learning Models: A Comparative Study. In *Advances in Visual Informatics. Lecture Notes in Computer Science*, 10645, pp. 3-15.

DOI: https://doi.org/10.1007/978-3-319-70010-6_1

Vehicle Detection Using Alex Net and Faster R-CNN Deep Learning Models: A Comparative Study

Jorge E. Espinosa[1], Sergio A. Velastin[2,3(✉)], and John W. Branch[4]

[1] Facultad de Ingenierías,
Politécnico Colombiano Jaime Isaza Cadavid – Medellín, Medellín, Colombia
jeespinosa@elpoli.edu.co
[2] University Carlos III - Madrid, Madrid, Spain
sergio.velastin@ieee.org
[3] Queen Mary University of London, London, UK
[4] Facultad de Minas, Universidad Nacional de Colombia – Sede Medellín, Medellín, Colombia
jwbranch@unal.edu.co

Abstract. This paper presents a comparative study of two deep learning models used here for vehicle detection. Alex Net and Faster R-CNN are compared with the analysis of an urban video sequence. Several tests were carried to evaluate the quality of detections, failure rates and times employed to complete the detection task. The results allow to obtain important conclusions regarding the architectures and strategies used for implementing such network for the task of video detection, encouraging future research in this topic.

Keywords: Convolutional Neural Network · Feature extraction · Vehicle classification

1 Introduction

Currently traffic management is supported by urban traffic analysis. Traditionally, vehicle counting and road density evaluations are done with inductive loop sensors and increasingly with video information. Nevertheless, video detection faces different challenges due to changes in illumination conditions and high vehicle densities with frequent occlusions. Most video detection systems are based on appearance features or motion features. Appearance features [1–4] such as shape, color, edge maps and texture, are used to detect vehicles even in stationary positions. Other works are based on HOG (Histogram of Oriented Gradients) and some variations of it [5, 6]. Motion features are obtained based on the dynamics of the traffic movement. Such methods are generally based on background subtraction [7], use of frame difference [8], Kalman filter [9], optical flow [10, 11], etc. For a detailed survey of traditional vehicle detection methods please see [12].

On the other hand, *deep learning theory* (DL) applied to image processing is the current dominant computer vision theory especially in tasks such as image recognition.

Since 2010 the annual image recognition challenge known as the ImageNet Large-Scale Visual Recognition Competition (ILSVRC) [13] is being dominated by this approach.

For vehicle detection, several works using Deep Learning in vehicle detection are reported in the literature. Earlier approaches relied on 2D Deep Belief Networks (2D-DBN) [14], learning features by means of this architecture and using a pre-training sparse filtering process [15] or Hybrid architectures (HDNN) which overcome the issue of the single scale extraction features of traditional DNNs [16]. Color as a discriminative feature is used in [17] and [18]. There are also pre-training schemes [19] that obtain competitive results even with low resolution images and implementable in real time as in [20]. More recently, detection and classification of multiples classes is performed using integrated models as Fast R-CNN and Faster R-CNN [21–26]. Reports exist of methods able to recognize vehicle make and models (MMR) [27, 28], re-identification architectures for security urban surveillance [29–31], strategies using DBN [14, 32–34] that work with relatively few labelled data and models that are able to classify even the pose or orientation of the vehicle [23, 35, 36]. Generally most of the detection and classifications models are implemented using different CNN architectures such as CaffeNet [37, 38], GoogLeNet [39], and VGGNet [26] used in [27]. AlexNet [40] is used by Su et al.[18] in conjunction with GoogleNet [39] and NIN (Network in Network) [41].

Nevertheless, as far as we know, there are no comparative studies of DL strategies used for vehicle classification, nor on the use of CNNs already trained for feature extraction to perform vehicle discrimination in video sequences.

This work compares the results of a CNN used for feature extraction and a CNN integrated model network, both used for the task of classifying vehicles in video sequences. The paper is organized as follows: Sect. 2 gives a brief explanation of the architecture of the convolutional neural networks, explaining the advantage of the use of an already-trained network for feature extraction and the benefits of the integrated CNN model. Section 3 shows the classification approaches, describing the character-istics of the models built for the video detection task. Section 4 shows the results of the two models, comparing and explaining the results. Section 5 presents the conclusions and proposes some future work.

2 CNN Architectures Used

In this section, we describe the principal characteristics of the CNN AlexNet and the Faster R-CNN networks used in this comparative study.

2.1 AlexNet

AlexNet is considered the pioneer work of CNN networks, even after the work of Yann LeCun [42]. The AlexNet model was introduced in the paper “ImageNet Classification with Deep Convolutional Networks”, were the authors created a “large, deep convolutional neural network”, used to win the 2012 ILSVRC (ImageNet Large-Scale Visual Recognition Challenge) [43]. The network was trained on ImageNet data, with over 15 million annotated images from a total of over 22,000 categories.

The architecture of "AlexNet" has 23 layers, integrating 5 convolution layers, 5 ReLu layers (Rectified units), 2 layers for normalization, 3 pooling layers, 3 fully connected layers, one probabilistic layer with softmax units and finally a classification layer ending in 1000 neurons for 1000 categories (Fig. 1).

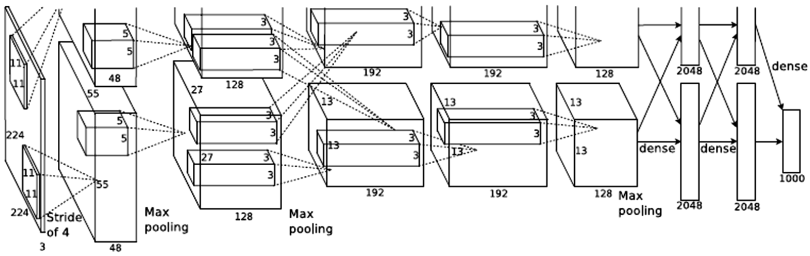

Fig. 1. Alex Net architecture. [40]

The main characteristics of the network includes the use of ReLU for the nonlinearity functions that decrease the training time as ReLUs are faster than the conventional *tanh* function used in MLP. For training proposes, the authors used techniques as data augmentation consisting in horizontal reflections, image translations or even patch extractions. Dropout layers were also included to overcome the problems of vanishing gradient and overfitting. The model was trained using batch stochastic gradient descent, using specific values for momentum and weight decay. It took nearly six days for training using two GTX 580 GPUs.

2.2 Faster R-CNN

The problem of object detection involves the detection of different classes of objects in each image. Traditionally, the strategy used was to deploy a two-class classifier (object vs non-object) in conjunction with a sliding window search. The amount of all windows scales and ratios returned could be huge, and then it was necessary to implement methods like non-maximal suppression for reducing the redundant candidates. This was the origin of object proposal algorithms, with strategies to limit search using calibration information [44], Branch & Bound [45] or grouping adjacent pixels merging them to find a blob region as in Selective Search [46]. These methods also included pre defining windows based on objects candidates as in Spatial Pyramid Pooling [47], or Edge boxes [48]. A valuable comparison of such methods was done by Hosang et al. [49]. The prefiltering strategy has been used with positive results combining it with CNN networks for classification as in R-CNN [50] but employing too much time in the training process. To improve the training process time, Fast R-CNN [21] has been proposed that swaps the extracting strategy of detecting regions and running CNN. A high-resolution image is fed to the CNN network, the network produces a high resolution convolutional feature map. The region proposal produces regions over the feature map (conv5). The convolutional features of these regions are

then fed into fully connected layers, with a linear classifier and a bounding box linear regression module to define regions. This model continues slowly at test time. Faster R-CNN addressed this issue by combining features of a fully convolutional network to perform both region proposals and object detection. Since region proposals depended on features of the image that were already calculated with the forward pass of the CNN (first step of classification), the model reuses the same CNN results for region proposals instead of running a separate selective search algorithm. The region proposal network (RPN) shares convolutional layers with the object detection network, then only one CNN needs to be trained and region proposals is calculated almost for free. Then, additional convolutional layers are used to regress region bounds with scores for object proposal at each location. The RPN works by moving a sliding window over the CNN feature map and at each window, generating k potential bounding boxes and scores associated for how good each of those boxes is expected to be. This k represents the common aspect ratios that candidates to objects could fit, caller anchor boxes. For each anchor box, the RPN output a bounding box and score per position in the image. This model improves significantly the speed and the object detection results.

Besides achieving the highest accuracy on both PASCAL VOC 2007 and 2012, Faster R-CNN was the basis of more than 125 proposed entries in ImageNet detection and localization at ILSVRC 2016 [51] and in the COCO challenge 2015 was the foundation of the winners in several categories [25]. Figure 2 shows the network structure of the Faster R-CNN framework. Both the region proposal network and the object classifier share fully convolutional layers. These layers are trained jointly. The region proposal network behaves as an attention director, determining the optimal bounding boxes across a wide range of scales and using nine candidate aspect ratios to be evaluated for object classi-fication. In other words, the RPN tells the unified network where to look.

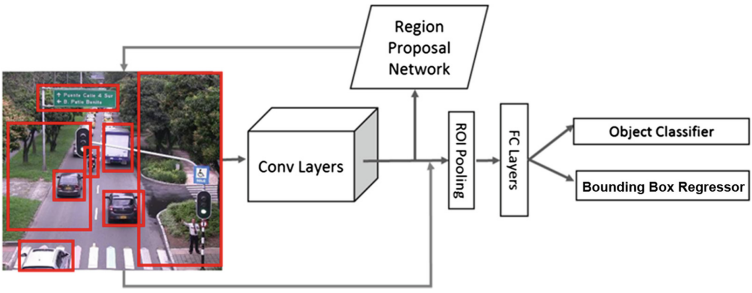

Fig. 2. Faster R-CNN network structure. Modified from [24]

3 Detection and Classification Approaches

An initial ROI (region of interest) step is implemented, allowing the user to select the precise area of analysis. This step optimizes performance and speeds up the processes of detection and classification, by reducing the area of analysis in the video sequences. Two models, AlexNet and Faster R-CNN were used for vehicle detection and classi-fication in this research.

3.1 AlexNet Model

In the AlexNet model, object detection is performed based on background subtraction, using a Gaussian Mixture Model according to Zivkovic [52, 53]. Once the objects are detected, these are classified as vehicles on three categories: motorcycles, cars or buses. Other possible objects detected are classified as part of the urban environment ("urbTree"), any possible remaining detection is assigned to the "unknown" class. The classifier is constructed using a multiclass linear SVM trained with features obtained from a pre-trained CNN. Those features have been extracted for a set of 80 images per category, including the "urbTree" category created from the urban traffic environment. This approach to image category classification is based on the work published by Matlab in "Image category classification using deep learning" [54] and extended in [55] for Motorcycles classification. Here, we extend the categories to include the bus set. The classifier trained using CNN features provides close to 100% accuracy, which is higher than the accuracy achieved using methods such as Bag of Features and SURF. As in [55] the set of images categories has been created corresponding to images related to motorcycles, cars, buses and urban environment related objects ("urbTree"). Those images were obtained from different angles and perspectives in urban traffic in Medellin City (Colombia). Examples of each category are shown in Fig. 3.

Fig. 3. Examples of cars, motorbikes and buses.

Following the strategy described in [55] by using the selected images, the pre-trained CNN "AlexNet" network is used for feature extraction, this technique is detailed described by Razavian et al. in [56]. For this work, the AlexNet network is only used to classify four categories. This pre-trained network was used to learn motorcycles, cars and buses features obtained from the extended dataset, with 80 images per category and 80 examples of the class "urbTree" created from the urban environment. In the end, the total number of examples is only 320.

Features are extracted from the training set, propagating images through the network up to a specific fully connected layer (fc7), extracting activations responses to create a training set of features, which is used later for classification.

For classification, as in [55], a multiclass SVM classifier is trained using the image features obtained from the CNN. Since the length of the feature vector is 4096, a fast stochastic gradient descent solver is used as training algorithm. In this case the clas-sifier is trained with only 96 examples (24 per category). The validation set, which

corresponds to the remaining 224 examples (56 by category) is then classified. The classifier accuracy is evaluated now with the features obtained on this set. Figure 4 shows the results in a confusion matrix. The classifier mismatches three bus images classifying those as cars; one car image is classified as bus and another as a motorcycle. The mean accuracy obtained is 0.978.

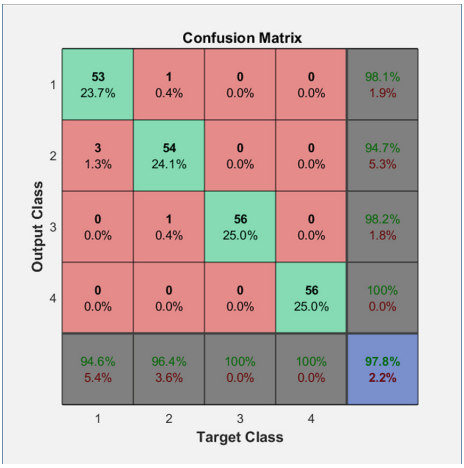

Fig. 4. Confusion matrix of the experiments (Class 1: Buses 2: Cars 3: Motorcycles 4: urbTree)

3.2 Faster R-CNN Model

The Faster R-CNN [25] model used in this research, is the one available for download at https://github.com/ShaoqingRen/faster_rcnn. The model is an object detection framework based on deep convolutional neural networks, which includes two networks: A Region Proposal Network (RPN) and an Object Detection Network. Both networks are trained for sharing convolutional layers to obtain real time results.

The model can be run based on two referenced CNN networks: ZF Net [57] or VGG16 [26]. In this research, we chose VGG16, which corresponds to the best performance results given the network layer configuration that the literature reports on the use of this architecture.

The VGG16 based model is pretrained with the ImageNet dataset. After downloading, both networks (RPN and ODN) are retrained in the PASCAL VOC 2007 dataset (Fig. 5) [58]. Based on the dataset used for retraining, the ODN is able to classify detections on 20 categories as follows:

- Person: person
- Animal: bird, cat, cow, dog, horse, sheep
- Vehicle: aeroplane, bicycle, boat, bus, car, motorbike, train
- Indoor: bottle, chair, dining table, potted plant, sofa, tv/monitor

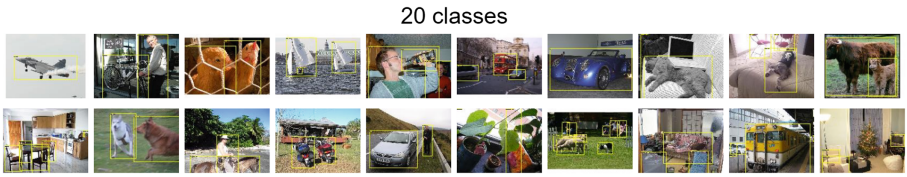

Fig. 5. Examples of each category in PASCAL VOC 2007 dataset [58].

For this experiments all classes different to bus, car or motorbike, are renamed as "unknown" to obtain comparative metrics for the results evaluation. As a preliminary step for video detection and classification, the user defines a ROI within which the detection and classification takes place.

4 Experiments and Results

We selected a video sequence of a real urban environment took in Medellín Colombia, over a secondary street of two lanes. The sequence consists of 1812 RGB frames (640 × 480), during daylight and good weather conditions. The sequence includes 36 different cars to detect (including sedan, VAN or taxis), 7 motorbikes and 1 Bus, which is the class assigned to a detected truck.

An XML-coded ground truth was obtained by means of ViPER [59] tool for annotation, and is used to compare the results of the two models. We use the performance metrics reported in [60]. These metrics have been extended to take into account the multiclass nature of the experiment.

The evaluations for Faster R-CNN are performed based on the NMS parameter threshold, used to reduce the redundancy on proposed regions. This threshold corresponds to the IoU overlap of the proposed regions. Results described in Table 1 and Fig. 6, show that decreasing the IoU threshold criteria increases the correct detection rate in total and for each class analyzed, but at the same time increases the False Alarm Rate. Best results correspond to a NMS threshold of 0.6 with a F1-Score of 0.76.

Table 1. Rates of Faster R-CNN results. NMS (non maximal suppression) – CDR: correct detection rate, Bikes: CDR for motorcycles, Cars: CDR for cars, Bus: CDR for buses or trucks. DFR: Detection Failure Rate. FAR: False Alarm Rate. PR: Precision. RC: Recall. F1:F1-score.

NMS threshold	CDR	CDR bikes	CDR cars	CDR buses	DFR	FAR	PR	RC	F1
0.30	0.75	0.26	0.83	0.33	0.25	0.38	0.62	0.75	0.68
0.40	0.73	0.21	0.81	0.32	0.27	0.31	0.69	0.73	0.71
0.50	0.72	0.18	0.80	0.33	0.28	0.22	0.78	0.72	0.75
0.60	0.70	0.13	0.78	0.29	0.30	0.16	0.84	0.70	**0.76**
0.70	0.65	0.09	0.74	0.23	0.35	0.13	0.87	0.65	0.75
0.80	0.61	0.06	0.70	0.07	0.39	0.10	0.90	0.61	0.73

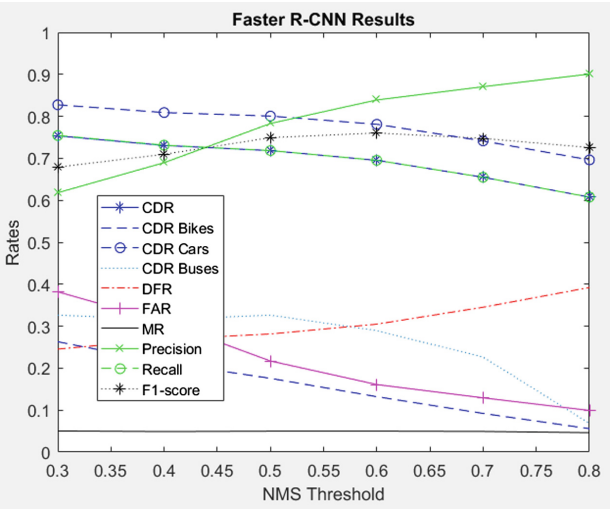

Fig. 6. Faster R-CNN rates results.

Meanwhile, working with the AlexNet classifier in conjunction with the GMM background subtraction, results are obtained in terms of the parameters of the background subtraction algorithm. First, the history parameter is evaluated against a fixed Mahalanobis distance of 128. History corresponds to the number of frames (LoH) that constitutes the training set for the background model. The best results obtained are for a history of 500 frames (F1 = 0.57). Fixing this number, we then proceed to change the Mahalanobis distance parameter (Tg). This parameter is a threshold for the squared Mahalanobis distance that helps decide when a sample is close to the existing components. A smaller Tg value generates more components. A higher Tg value may result in a small number of components but they can grow too large. The best result is obtained with LoH of 500, and Tg of 20, achieving a CDR of 0.66 with a FAR of 0.32 (Fig. 7 and Tables 2, 3).

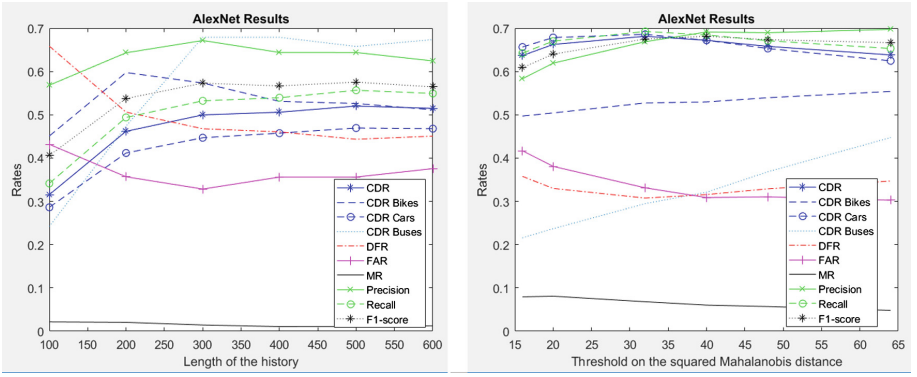

Fig. 7. Alex Net rates results

Table 2. Rates of Faster R-CNN results. NMS (non maximal suppression) – CDR: correct detection rate, Bikes: CDR for motorcycles, Cars: CDR for cars, Bus: CDR for buses or trucks. DFR: Detection Failure Rate. FAR: False Alarm Rate. MR: Merge Rate. PR: Precision. RC: Recall. F1:F1-score.

LoH	CDR	CDR bikes	CDR cars	CDR buses	DFR	FAR	MR	PR	RC	F1
100	0.32	0.45	0.29	0.24	0.66	0.43	0.02	0.57	0.34	0.41
200	0.46	0.60	0.41	0.47	0.51	0.36	0.02	0.64	0.49	0.54
300	0.50	0.57	0.45	0.68	0.47	0.33	0.01	0.67	0.53	0.57
400	0.51	0.53	0.46	0.68	0.46	0.36	0.01	0.64	0.54	0.57
500	0.52	0.53	0.47	0.66	0.44	0.36	0.01	0.64	0.56	**0.58**
600	0.51	0.51	0.47	0.67	0.45	0.38	0.01	0.62	0.55	0.56

Table 3. Rates of Faster R-CNN results. NMS (non maximal suppression) – CDR: correct detection rate, Bikes: CDR for motorcycles, Cars: CDR for cars, Bus: CDR for buses or trucks. DFR: Detection Failure Rate. FAR: False Alarm Rate. MR: Merge Rate. PR: Precision. RC: Recall. F1:F1-score.

Tg	CDR	CDR bikes	CDR cars	CDR buses	DFR	FAR	MR	PR	RC	F1
16	0.64	0.50	0.66	0.22	0.36	0.42	0.08	0.58	0.64	0.61
20	0.66	0.50	0.68	0.24	0.33	0.38	0.08	0.62	0.67	0.64
32	0.68	0.53	0.69	0.29	0.31	0.33	0.07	0.67	0.69	0.67
40	0.67	0.53	0.67	0.32	0.32	0.31	0.06	0.69	0.68	**0.68**
48	0.66	0.54	0.65	0.37	0.33	0.31	0.06	0.69	0.67	0.67
64	0.64	0.55	0.62	0.45	0.35	0.30	0.05	0.70	0.65	0.66

The results obtained show that Faster R-CNN outperforms Alex Net+GMM model, not only in the correct detection rate obtained while producing less false detections, but also in time spending in the analysis. Both model were analyzed on a Windows 10 Machine with a core i7 7th generation, 4.7 GHz, and 32 GB of RAM, using an NVidia Titan X (Pascal) 1531 MHz GPU, achieving close to real time in Faster R-CNN model (40 ms per frame) while AlexNet+GMM took almost 100 ms per frame.

5 Conclusions and Future Work

This paper has compared the performance of two deep learning models for vehicle detection and classification in urban video sequences. Although the AlexNet model is used for feature extraction in an ad-hoc set of examples oriented to urban scenarios, the pre-trained Faster R-CNN model achieves better results in correct detections according to F1-score measure. It is important to remark that the Faster R-CNN model does not use any dynamic attributes for vehicle detection whereas GMM background subtraction used in AlexNet model. In fact, as the merge rates (MR) result shows, GMM

background subtraction still has issues with stationary vehicles and occluded scenarios. In Faster R-CNN, the RPN component results could be improved providing some urban context information as restriction size of the regions.

For future work, we intend to improve the results of the RPN component of the Faster R-CNN model enriching it with traffic context information, and improve the classification component with feature extraction using a Deep Architecture as AlexNet, ZF or VGG, with a wider set of urban road user classes (e.g. trucks, vans, cyclists, pedestrians).

Acknowledgments. S.A. Velastin is grateful to funding received from the Universidad Carlos III de Madrid, the European Union's Seventh Framework Programme for research, technological development and demonstration under grant agreement no. 600371, el Ministerio de Economía y Competitividad (COFUND2013-51509) and Banco Santander. The authors wish to thank Dr. Fei Yin for the code for metrics employed for evaluations. Finally, we gratefully acknowledge the support of NVIDIA Corporation with the donation of the GPUs used for this research. The data and code used for this work is available upon request from the authors.

References

1. Tsai, L.W., Hsieh, J.W., Fan, K.C.: Vehicle detection using normalized color and edge map. IEEE Trans. Image Process. **16**(3), 850–864 (2007)
2. Ma, X., Grimson, W.E.L.: Edge-based rich representation for vehicle classification. In: 10th IEEE International Conference on Computer Vision (ICCV 2005), vol. 1–2, pp. 1185–1192 (2005)
3. Buch, N., Orwell, J., Velastin, S.A.: 3D extended histogram of oriented gradients (3DHOG) for classification of road users in urban scenes (2009)
4. Feris, R.S., et al.: Large-scale vehicle detection, indexing, and search in urban surveillance videos. IEEE Trans. Multimed. **14**(1), 28–42 (2012)
5. Chen, Z., Ellis, T.: Multi-shape descriptor vehicle classification for urban traffic. In: 2011 International Conference on Digital Image Computing Techniques and Applications (DICTA), pp. 456–461 (2011)
6. Chen, Z., Ellis, T., Velastin, S.A.: Vehicle detection, tracking and classification in urban traffic. In: 2012 15th International IEEE Conference on Intelligent Transportation Systems, pp. 951–956 (2012)
7. Gupte, S., Masoud, O., Martin, R.F., Papanikolopoulos, N.P.: Detection and classification of vehicles. IEEE Trans. Intell. Transp. Syst. **3**(1), 37–47 (2002)
8. Cucchiara, R., Piccardi, M., Mello, P.: Image analysis and rule-based reasoning for a traffic monitoring system. IEEE Trans. Intell. Transp. Syst. **1**(2), 119–130 (2000)
9. Messelodi, S., Modena, C.M., Zanin, M.: A computer vision system for the detection and classification of vehicles at urban road intersections. Pattern Anal. Appl. **8**(1–2), 17–31 (2005)
10. Huang, C.-L., Liao, W.-C.: A vision-based vehicle identification system. In: Proceedings of 17th International Conference on Pattern Recognition, ICPR 2004, vol. 4, pp. 364–367 (2004)
11. Ottlik, A., Nagel, H.-H.: Initialization of model-based vehicle tracking in video sequences of inner-city intersections. Int. J. Comput. Vis. **80**(2), 211–225 (2008)

12. Tian, B., et al.: Hierarchical and networked vehicle surveillance in its: a survey. IEEE Trans. Intell. Transp. Syst. **16**(2), 557–580 (2015)
13. ImageNet Large Scale Visual Recognition Competition (ILSVRC). http://www.image-net.org/challenges/LSVRC/. Accessed 24 Oct 2016
14. Wang, H., Cai, Y., Chen, L.: A vehicle detection algorithm based on deep belief network. Sci. World J. **2014**, e647380 (2014)
15. Dong, Z., Pei, M., He, Y., Liu, T., Dong, Y., Jia, Y.: Vehicle type classification using unsupervised convolutional neural network. In: 2014 22nd International Conference on Pattern Recognition (ICPR), pp. 172–177 (2014)
16. Chen, X., Xiang, S., Liu, C.L., Pan, C.H.: Vehicle detection in satellite images by hybrid deep convolutional neural networks. IEEE Geosci. Remote Sens. Lett. **11**(10), 1797–1801 (2014)
17. Hu, C., Bai, X., Qi, L., Chen, P., Xue, G., Mei, L.: Vehicle color recognition with spatial pyramid deep learning. IEEE Trans. Intell. Transp. Syst. **16**(5), 2925–2934 (2015)
18. Su, B., Shao, J., Zhou, J., Zhang, X., Mei, L.: Vehicle color recognition in the surveillance with deep convolutional neural networks (2015)
19. Zhang, F., Xu, X., Qiao, Y.: Deep classification of vehicle makers and models: the effectiveness of pre-training and data enhancement. In: 2015 IEEE International Conference on Robotics and Biomimetics (ROBIO), pp. 231–236 (2015)
20. Bautista, C.M., Dy, C.A., Mañalac, M.I., Orbe, R.A., Cordel, M.: Convolutional neural network for vehicle detection in low resolution traffic videos. In: 2016 IEEE Region 10 Symposium (TENSYMP), pp. 277–281 (2016)
21. Girshick, R.: Fast R-CNN. In: Proceedings of IEEE International Conference on Computer Vision, pp. 1440–1448 (2015)
22. Wang, S., Liu, F., Gan, Z., Cui, Z.: Vehicle type classification via adaptive feature clustering for traffic surveillance video. In: 2016 8th International Conference on Wireless Communications Signal Processing (WCSP), pp. 1–5 (2016)
23. Chabot, F., Chaouch, M., Rabarisoa, J., Teulière, C., Chateau, T.: Deep MANTA: a coarse-to-fine many-task network for joint 2D and 3D vehicle analysis from monocular image. arXiv preprint arXiv:170307570 (2017)
24. Fan, Q., Brown, L., Smith, J.: A closer look at faster R-CNN for vehicle detection. In: 2016 IEEE Intelligent Vehicles Symposium (IV), pp. 124–129 (2016)
25. Ren, S., He, K., Girshick, R., Sun, J.: Faster R-CNN: towards real-time object detection with region proposal networks. In: Advances in Neural Information Processing Systems, pp. 91–99 (2015)
26. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:14091556 (2014)
27. Liu, D., Wang, Y.: Monza: image classification of vehicle make and model using convolutional neural networks and transfer learning. http://cs231n.stanford.edu/reports/2015/pdfs/lediurfinal.pdf. Accessed 16 Oct 2017
28. Gao, Y., Lee, H.J.: Local tiled deep networks for recognition of vehicle make and model. Sensors **16**(2), 226 (2016)
29. Liu, X., Liu, W., Mei, T., Ma, H.: A deep learning-based approach to progressive vehicle re-identification for urban surveillance. In: European Conference on Computer Vision, pp. 869–884 (2016)
30. Bromley, J., et al.: Signature verification using a "siamese" time delay neural network. IJPRAI **7**(4), 669–688 (1993)
31. Su, B., Shao, J., Zhou, J., Zhang, X., Mei, L., Hu, C.: The precise vehicle retrieval in traffic surveillance with deep convolutional neural networks. Int. J. Inf. Electron. Eng. **6**(3), 192 (2016)

32. Cai, Y., Sun, X., Wang, H., Chen, L., Jiang, H.: Night-time vehicle detection algorithm based on visual saliency and deep learning. J. Sens. **2016** (2016)
33. Wu, Y.Y., Tsai, C.M.: Pedestrian, bike, motorcycle, and vehicle classification via deep learning: deep belief network and small training set. In: 2016 International Conference on Applied System Innovation (ICASI), pp. 1–4 (2016)
34. Huang, B.-J., Hsieh, J.-W., Tsai, C.-M.: Vehicle detection in Hsuehshan Tunnel using background subtraction and deep belief network. In: Asian Conference on Intelligent Information and Database Systems, pp. 217–226 (2017)
35. Zhou, Y., Liu, L., Shao, L., Mellor, M.: DAVE: a unified framework for fast vehicle detection and annotation. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) ECCV 2016. LNCS, vol. 9906, pp. 278–293. Springer, Cham (2016). doi:10.1007/978-3-319-46475-6_18
36. You, R., Kwon, J.-W.: VoNet: vehicle orientation classification using convolutional neural network. In: Proceedings of 2nd International Conference on Communication and Information Processing, pp. 195–199 (2016)
37. Caffe — Deep Learning Framework. http://caffe.berkeleyvision.org/. Accessed 05 Sept 2016
38. Luo, X., Shen, R., Hu, J., Deng, J., Hu, L., Guan, Q.: A deep convolution neural network model for vehicle recognition and face recognition. Procedia Comput. Sci. **107**, 715–720 (2017)
39. Szegedy, C., et al.: Going deeper with convolutions. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, pp. 1–9 (2015)
40. Krizhevsky, A., Sutskever, I., Hinton, G.E.: ImageNet classification with deep convolutional neural networks. In: Advances in Neural Information Processing Systems, pp. 1097–1105 (2012)
41. Lin, M., Chen, Q., Yan, S.: Network in network. arXiv preprint arXiv:13124400 (2013)
42. Lecun, Y., Bottou, L., Bengio, Y., Haffner, P.: Gradient-based learning applied to document recognition. Proc. IEEE **86**(11), 2278–2324 (1998)
43. ImageNet Large Scale Visual Recognition Competition 2012 (ILSVRC 2012). http://www.image-net.org/challenges/LSVRC/2012/. Accessed 30 Aug 2017
44. Brown, L.M., Fan, Q., Zhai, Y.: Self-calibration from vehicle information. In: 2015 12th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS), pp. 1–6 (2015)
45. Lampert, C.H., Blaschko, M.B., Hofmann, T.: Efficient subwindow search: a branch and bound framework for object localization. IEEE Trans. Pattern Anal. Mach. Intell. **31**(12), 2129–2142 (2009)
46. Uijlings, J.R., Van De Sande, K.E., Gevers, T., Smeulders, A.W.: Selective search for object recognition. Int. J. Comput. Vis. **104**(2), 154–171 (2013)
47. He, K., Zhang, X., Ren, S., Sun, J.: Spatial pyramid pooling in deep convolutional networks for visual recognition. IEEE Trans. Pattern Anal. Mach. Intell. **37**(9), 1904–1916 (2015)
48. Zitnick, C.L., Dollár, P.: Edge boxes: locating object proposals from edges. In: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (eds.) ECCV 2014. LNCS, vol. 8693, pp. 391–405. Springer, Cham (2014). doi:10.1007/978-3-319-10602-1_26
49. Hosang, J., Benenson, R., Dollár, P., Schiele, B.: What makes for effective detection proposals? IEEE Trans. Pattern Anal. Mach. Intell. **38**(4), 814–830 (2016)
50. Girshick, R., Donahue, J., Darrell, T., Malik, J.: Rich feature hierarchies for accurate object detection and semantic segmentation. In: 2014 IEEE Conference on Computer Vision and Pattern Recognition, pp. 580–587 (2014)
51. ILSVRC2016. http://image-net.org/challenges/LSVRC/2016/results. Accessed 30 Aug 2017
52. Zivkovic, Z.: Improved adaptive Gaussian mixture model for background subtraction. In: Proceedings of 17th International Conference on Pattern Recognition, ICPR 2004, vol. 2, pp. 28–31 (2004)

53. Zivkovic, Z., Van Der Heijden, F.: Efficient adaptive density estimation per image pixel for the task of background subtraction. Pattern Recognit. Lett. **27**(7), 773–780 (2006)
54. Image Category Classification Using Deep Learning - MATLAB & Simulink Example. https://www.mathworks.com/help/vision/examples/image-category-classification-using-deep-learni ng.html. Accessed 28 Feb 2017
55. 8th International Conference on Pattern Recognition Systems |Universidad Carlos III de Madrid — Madrid, Spain. http://velastin.dynu.com/icprs17/programme.php. Accessed 30 Aug 2017
56. Razavian, A.S., Azizpour, H., Sullivan, J., Carlsson, S.: CNN features off-the-shelf: an astounding baseline for recognition. In: 2014 IEEE Conference on Computer Vision and Pattern Recognition Workshops, pp. 512–519 (2014)
57. Zeiler, M.D., Fergus, R.: Visualizing and understanding convolutional networks. In: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (eds.) ECCV 2014. LNCS, vol. 8689, pp. 818–833. Springer, Cham (2014). doi:10.1007/978-3-319-10590-1_53
58. The PASCAL Visual Object Classes Challenge 2007 (VOC 2007). http://host.robots.ox.ac.uk/pascal/VOC/voc2007/index.html. Accessed 31 Aug 2017
59. ViPER: The Video Performance Evaluation Resource. http://viper-toolkit.sourceforge.net/. Accessed 31 Aug 2017
60. Yin, F., Makris, D., Velastin, S.A.: Performance evaluation of object tracking algorithms. In: IEEE International Workshop on Performance Evaluation of Tracking and Surveillance, Rio De Janeiro, Brazil, p. 25 (2007)