Multi-stream Deep Networks for Vehicle Make and Model Recognition

Mohamed Dhia Elhak Besbes[1], Yousri Kessentini[2] and Hedi Tabia[1]
[1]*IBISC, Univ. Evry, Université Paris-Saclay, 91025, Evry, France*
[2]*Digital Research Center of Sfax, Sfax, Tunisia*

Keywords: Fine-grained Vehicle Recognition, Convolutional Neural Network, Multi-stream Fusion.

Abstract: Vehicle recognition generally aims to classify vehicles based on make, model and year of manufacture. It is a particularly hard problem due to the large number of classes and small inter-class variations. To handle this problem recent state of the art methods use Convolutional Neural Network (CNN). These methods have however several limitations since they extract unstructured vehicle features used for the recognition task. In this paper, we propose more structured feature extraction method by leveraging robust multi-stream deep networks architecture. We employ a novel dynamic combination technique to aggregate different vehicle part features with the entire image. This allows combining global representation with local features. Our system which has been evaluated on publicly available datasets is able to learn highly discriminant representation and achieves state-of-the-art result.

1 INTRODUCTION

Vehicle Make and Model Recognition (VMMR) is both coarse and fine-grained classification problem. On one hand, vehicles can have unconstrained poses when taken under multiple view points. Classification under such condition can be seen as coarse grained problem. On the other hand, the unique hierarchical structure starting from mark, model to year of manufacture makes vehicle categories very similar with a subtle inter-class variation.

In the literature, only few papers have addressed the above-mentioned problems. Most of the earlier vehicle identification research focuses on license plate recognition (Li et al., 2017; Cheang et al., 2017; Li and Shen, 2016; Masood et al., 2017; Du et al., 2013; Gou et al., 2016; Hsu et al., 2013) and vehicle make recognition (Gao and Lee, 2015; Khan et al., 2010; Wei Wu et al., 2001; Lai et al., 2001; Xiaoxu Ma and Grimson, 2005; Yishu Peng et al., 2014; Psyllos et al., 2010).

Early works on vehicle model recognition focused on low level features representation: (Psyllos et al., 2009) uses Scale Invariant Feature Transform (SIFT (Lowe, 2004; Lowe, 1999; Lowe, 2001)) to describe make-model instances. This method is computationally expensive. To overcome this issue Speeded Up Robust Features (SURF (Bay et al., 2008)) and Histogram Oriented Gradients (HOG) have been used by (Hsieh et al., 2014a) for more robustness and speed. The SURF method which uses the Hessian matrix approximation to detect key points gives more robust results while being faster than the SIFT based methods (Psyllos et al., 2009). Several variations of SURF descriptor have also been used. These variations include (1) Features from Accelerated Segment Test(FAST) which is a key-points detection method designed for real-time applications, (2) Binary Robust Independent Elementary Features (BRIEF) and (3) Oriented FAST which uses FAST detector for key-points detection and BRIEF as descriptor.

The majority of these methods only rely on low level features without any structured information. This makes them very sensitive to different types of noise, especially occlusions and the presence of several vehicles in a single image.

Unlike the conventional feature extraction algorithms (e.g. SIFT, HOG), Convolutional Neural Networks (CNN) uses several hidden layers to hierarchically learn high level representation of the image. Convolving filters (or kernels) on the image allows the network to extract more relevant features. Activation functions and pooling layers allow the network to be more robust to scaling, translation and rotation variations. Moreover, high level feature representations are less sensitive to noise. Due to this fact CNNs became very popular tools extensively used by the computer vision community (Sam et al., 2017; Luvizon et al., 2018; Paumard et al., 2018). In particularly in vehicle model and make recognition, CNN based

Besbes, M., Kessentini, Y. and Tabia, H.
Multi-stream Deep Networks for Vehicle Make and Model Recognition.
DOI: 10.5220/0008975404130419
In *Proceedings of the 15th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISIGRAPP 2020) - Volume 5: VISAPP*, pages 413-419
ISBN: 978-989-758-402-2; ISSN: 2184-4321

approaches are achieving impressive results. Works such as proposed in (He et al., 2015) recognize vehicle make and model from surveillance camera by first, detecting frontal view components such as the grilles, the plate or the lights. Then, specialized CNNs classify each part of the vehicle, before a global car classification. While this method achieves high performance, it is limited to frontal view points only.

Another part-based vehicle recognition method proposed by (Biglari et al., 2018) attends to find relevant parts for each vehicle class. The method uses one classifier per class and a cascade of classifiers are applied to the input image.

(Hsieh et al., 2014b) uses SIFT-like local descriptors to train weak classifiers over a grid of vehicle parts. (Hu et al., 2017) propose Spacial Weighted Pooling (SWP) instead of the standard pooling in the CNN. SWP layer feds the fully connected layer with robust feature representation by magnifying features corresponding to the discriminant parts of the image. However, it has been shown that the performance of the SWP layers decreases with large variations in the scale and the position of the vehicle. (Ghassemi et al., 2018) propose a deep convolutional architecture built upon multi-scale attention windows. Through those windows the most discriminative parts of the vehicle are aggregated over different scales. The model uses Residual Networks (He et al., 2015) with Spatial transformer networks(STN) (Jaderberg et al., 2015) to improve resilience to affine transformations. However, in an STN with multiple feed-forward alignment modules, the output image of the previous alignment module is directly fed into the next. This is problematic as it can create unwanted boundary effects as the number of geometric prediction layers increase.

In this work, we propose a robust dynamic multi-stream model that is able to extract vehicle feature parts, as well as vehicle global features from the entire image. This allows to jointly detect fine grained features related to each part and the global vehicle representation. We introduce a new network architecture which combines the local and the global representations. This architecture enables dynamic input features to fed into a fully connected layer. Our experiments conducted on a publicly available dataset show the superiority of our method compared to state-of-the-art ones.

2 THE PROPOSED APPROACH

We propose a multi-stream robust architecture to extract and combine both local and global features representations for VMMR. First, given a vehicle image, a pre-trained CNN detector finds vehicle parts. The number of detected parts may vary from one image to another. Moreover, a selection process is applied to only keep relevant parts from all detected ones (see Section 2.2). Then, each part goes, through the multi-stream architecture, into a specialized feature extractor which allows the system to detect subtle inter-class variations. Finally, all extracted features are aggregated using a novel fusion technique described in Section 2.4. An overview of the main steps of our system are depicted in Figure 1.

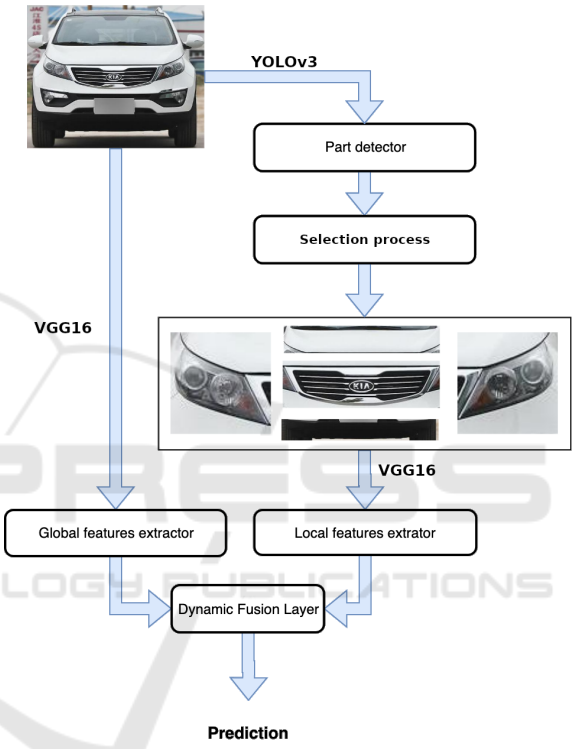

Figure 1: System's main steps. The image is first processed by YOLO to detect the vehicle parts. Global and local features are then extracted from the full vehicle image and the selected parts using VGG. The global and the local representations are then fed to the dynamic fusion layer to perform the final classification.

2.1 Vehicle Part Detection

Our approach starts by extracting a set of parts $\{P_1, P_2, \ldots, P_n\}$ from a given image I. Since vehicle pose may vary across images, we do not assume that all parts appear in I. Conventional detection methods learn classifiers to perform detection. Classifier rules are generally evaluated on a sliding window and a binary output is computed in the corresponding location. More recent deep learning based architectures get around the sliding window techniques and produced higher performance while being faster.

Following this advance, we choose to use YOLO (Redmon and Farhadi, 2018) to detect each vehicle part. Thanks to the encoded context information, YOLO detector provides robust results in all our experiments.

2.2 Selection Process

Usually different combinations of vehicle parts leads to different performance. We show in the experimental section that the front bumper, for instant, combined with the entire image yields better results than combining the front bumper, front left light, front right light and the entire image (see Table 4). To alleviate this problem, we introduce a selector which filters out the best performing set of parts from all possible combinations.

The selection process is based on a look-up table memorising the combination rates from the best to the worst performance, computed on the validation set.

To select a combination the algorithm iterates on the look-up table until it finds an existing combination. Figure 2 presents the selection process.

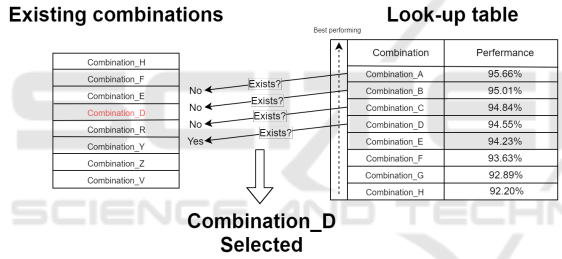

Figure 2: The selection algorithm iterates over the look-up table testing if the current part combination exists. Once a relevant combination is found, the algorithm filters out the chosen parts for the feature extraction module.

2.3 Multi-stream Architecture

Vehicle manufactures copy traits from previous models to produce new ones. This increases the complexity of the VMMR task, since the model recognition relies on the subtle variations between vehicle parts. Figure 4 shows the subtle differences between models. To detect these subtle variations, we employ a multi-stream architecture to apply specialized features extractors for every part and every combination of parts. Figure 4 also shows same models with different variations. In this case a global representation may benefits the recognition task rather than the part information. Our multi-stream architecture successfully combines both a global and local representations. It also provides a flexible system that can use any available stream which feds the input. Figure 3 shows the used multi-stream architecture.

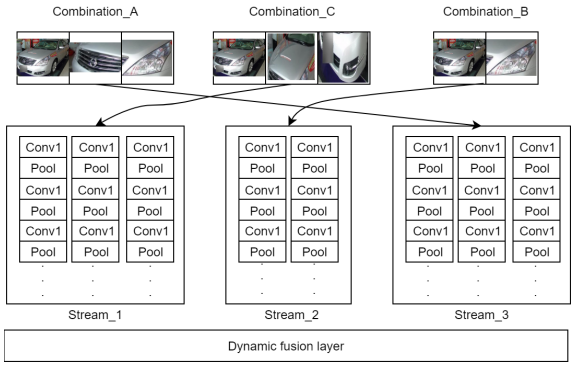

Figure 3: Multi-stream architecture used with different combinations. The figure shows different streams and how the selection process finds a combination which is then fed to the relevant stream.

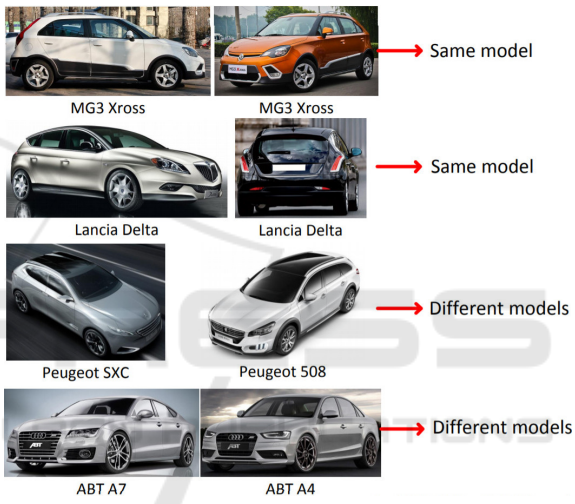

Figure 4: Examples of subtle differences between vehicle models. From a global perspective the vehicles seem similar, yet they belong to different vehicle models. This is due to very subtle variations.

We set streams as different combination of detected parts. Our selection process (section 2.2) ensure that the best performing stream, according to the validation set, is used. The number of combinations is empirically fixed bounded by the available memory.

2.4 Dynamic Fusion Layer

The multi-stream architecture provides important advantages such as robustness and specialized feature extractors, However, the resulted features will have different shapes depending on the used streams. Moreover, a single static classification layer may not be sufficient for representing all of the variations of the multi-stream architecture. As a solution we introduce a dynamic fusion layer which only considers relevant weights that fit the input and swap the others

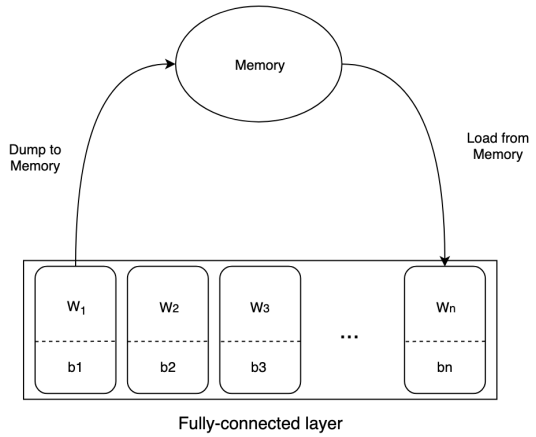

Figure 5: Dynamic fully-connected layer. Depending on the input, weights w and biases b are dynamically swapped. Relevant w and b are loaded from the memory while irrelevant w and b are dumped to memory. A final weight matrix W is the result of concatenating all the weights loaded from memory.

at the run-time. The swapping process is depicted in Figure 5.

Each part has a weight matrix W of shape 4096×431, 431 is the number of classes, and a bias vector b of shape 431. The system stores in memory all of the weights and biases then, at run-time, depending on available parts the system dumps unrelated weights and biases to the memory and load only the weights and biases of present parts.

This technique allows the fully-connected layer to have variable input shapes and to store part-specific features.

3 MODEL ARCHITECTURE

The model architecture for each stream is composed of three main sections. (1) The shared section: convolutional blocks that are common to all parts and the entire image. (2) specialized feature extractors: convolutional blocks and fully-connected layers specialized for each of the vehicle parts and the entire image.(3) The dynamic fusing layer for features aggregation and classification. Figure 6 shows the model architecture of a single stream. The Cross-entropy loss, or log loss $\sum_{c=1}^{M} y_{o_c} \times log(p_o, c)$ were M is the number of classes , y is a binary indicator (0 or 1) if class label c is the correct classification for observation o and p is the predicted probability observation o is of class c. Log loss is used for all of the training sessions. The loss is back-propagated til the fifth convolutional layer leaving the first four layers unchanged.

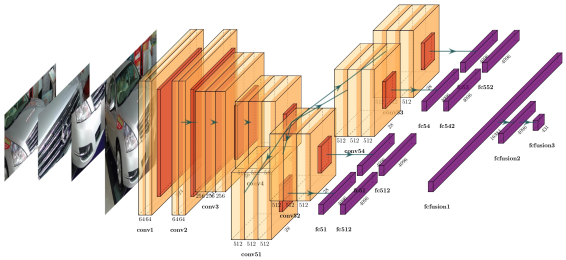

Figure 6: Model architecture. Initially, parts are introduced to the model sequentially from a single input. The batch B is composed of n mini-batches $\{b_1, b_2, \ldots, b_n\}$,$n$ being the number of parts, each mini-batch contains similar parts $\{p_1, p_2, \ldots, p_k\}$ for $k \leq n$ so the batch size is $k \times n$. The images are passed through the first four convolutional block then the batch B is split into n groups $\{\{p_{1_1}, p_{1_2}, \ldots, p_{1_n}\}, \{p_{2_1}, p_{2_2}, \ldots, p_{2_n}\}, \ldots, \{p_{k_1}, p_{k_2}, \ldots, p_{k_n}\}\}$ each image in the group is passed to a part-specific fifth convolutional block. Finally, the image and it parts features are aggregated with the dynamic fully-connected layer.

3.1 Shared Section

The Shared section is the set of convolutional blocs that extracts basic features: lines, edge,... the weights are uploaded from a pre-trained model on a large dataset. The layers on this section are frozen, the weights are not updated by back-propagation.

The shared section of the VGG16 based stream is composed of three convolutional blocs pre-trained on the ImageNet dataset.

3.2 Specialized Feature Extractors

This section contains layers with unfrozen weights to learn local features(parts features) and global features(entire image features).

For the VGG16 this section is composed of three convolutional blocs and one fully-connected layer. The same pooling functions are used as the original VGG16 however, we added two batch normalisation layer to speed-up the training process. The learning rate is initialized at 0.001 with a decay factor of 0.1. Although the system is end-to-end trainable we chose to train every feature extractor separately to speed-up the training process. First, the detector crops out the parts from the images. Second, for every feature extractor, we use a batch of the 70 cropped parts for training. Than the error is back-propagated using stochastic gradient descent. Finally, we upload all the feature extractors weights to the system to train the dynamic fusion layer.

3.3 Dynamic Fusion Layer

The dynamic fusion layer is the only section we trained with an end to end fashion. The batch for the entire system is a multiple of 5 depending on the number of streams to be used. Every stream receives a batch of 5 images so the dynamic fusion layer receives a batch of five images. The learning rate for the dynamic layer is initialized at 0.001 with a decay factor of 0.1. Only the dynamic layer weights are updated using stochastic gradient descent.

4 EXPERIMENTAL RESULTS

4.1 CompCars Dataset

The Comprehensive Cars(CompCars) (L. Yang, 2015) is a publicly accessible dataset containing in web-data a total of 136,727 images of the entire car and 27,618 capturing the car parts. The database respects the standard hierarchy with a total of 163 car Marks and 1,716 car models covering most of the commercial car models from 2005 to 2015. In terms of view ports there are five: Front-view, Rear-view, Side-view, Front-Side and Rear-Side. Table 1 shows the total number of images per view-port and the average number of view-port images per model. The

Table 1: Quantity distribution in different view ports.

View-port	No. in total	No.per model
Front	18431	10.9
Rear	13513	8.0
Side	23551	14.0
Front Side	49301	29.2
Rear Side	31150	18.5

dataset is divided into two types (1) The Web dataset: a collection of vehicle images taken from the web from different view port.(2) The nature dataset: a collection of vehicle images taken from surveillance cameras. only the front view port is available. In this work we test our model on the web dataset.
The web dataset contains most of the 1716 car models however the CompCars article (L. Yang, 2015) proposes a train/test split on 431 models. We adopt this split to compare our work to state-of-the-arts results.
Figure 7 shows different examples of the web dataset.

4.2 Results from Individual Parts

Table 2 and 3 shows the results of the VGG16 with Batch normalization on their respective cropped parts.

Figure 7: Web dataset.

The results show that some parts are more descriptive than others. For example, in the front-view-Bumper, the baseline VGG16 achieves 92.60% while the front-right-light baseline VGG16 achieves 61.76%.
We can also see that parts that contains the mark logo or model name like the Trunk and the Grilles achieves best results.

Table 2: Individual front parts of CompCars's web data.

Part	VGG16	Train	Test
Bumper	92.60%	7022	6740
Hood	92.26%	6988	6757
Grilles	93.83%	6622	6385
Left Light	64.72%	5875	5683
Right Light	61.76%	5942	5790

Table 3: Individual rear parts of CompCars's web data.

Part	VGG16	Train	Test
Bumper	89.56%	5603	5312
Trunk	93.90%	5225	5221
Left Light	72.89%	5124	4859
Right Light	82.10%	5587	5300

4.3 Multi-stream Dynamic Fusion

In this section we compare different combinations of parts. Table 4 shows the recognition rate of the best performing combination per view-port. We divide CompCars(L. Yang, 2015) dataset into three sections: (1) Front View, (2) Rear View and (3) Side View. We train and test different combinations on the three sections according to detected parts. The absence of a part in a view-port is considered as falsely recognized. We notice a drop in the performance on the side view-port where no parts are used.

Some parts are more descriptive than others like the Front Bumper for the front view and the Rear Trunk for rear views. However, a combination of the Front Bumper and the other parts decreases the accuracy to 91%.

We present in table 5 the obtained result of the baseline recognition method applying the VGG on the

Table 4: Fusion using different combinations on the CompCars's web data. The test dataset is divided on three sections per view port. we test combinations on their respective section. FB: Front Bumper, FH: Front Hood, FG: Front Grilles, FLL: Front Left Light, FRL: Front Right Light, RB: Rear Bumper, RT: Rear Trunk, RLL: Rear Left Light, RRL: Rear Right Light.

Combination	Compcars
————-Front View————	
full image + FB	96.12%
full image + FH	94.33%
full image + FG	92.74%
full image + FLL	45.36%
full image + FRL	46.85%
All Front parts	91.36%
full image + FB + FH + FG	92.65%
full image + FLL + FRL	90.12%
FB + FH + FG + FLL + FRL	84.12%
————-Rear View————	
full image + RB	91.11%
full image + RT	94.14%
full image + RLL	68.12%
full image + RRL	65.45%
All Rear parts	88.24%
RB + RT + RLL + RRL	65.26%
full image + RB + RT + RLL	87.26%
full image + RB + RT	89.22%
full image + RLL + RRL	62.25%
————Side View————	
full image	91.23%

Table 5: Comparison of the baseline VGG16 with our approach.

Approach	CompCars(web)
Baseline	92.66%
Ours	95.07%

full image (Baseline) and our proposed approach. It is clear that the proposed selective multi-stream combination method improves considerably the performance with a gain of 2.41%.

Table 6 shows recent results on the CompCars dataset were approaches with no deep convolutional networks achieves worst result such as Yang (Yang et al., 2015). BoxCars (Sochor et al., 2016) and Baseline VGG16 (Simonyan and Zisserman, 2014) rely on deep networks for global features representation. However, best results such as SWP-CNN(Hu et al., 2017) and WindowResnet(Ghassemi et al., 2018) used part based approaches. Our approach, combined both global and local representation allowing the system to be robust.

Table 6: Comparison with our approach.

Approach	CompCars(web)
Yang(Yang et al., 2015)	76.7%
BoxCars(Sochor et al., 2016)	84.8%
Ours(VGG16)	95.07%
SWP-CNN(Hu et al., 2017)	97.6%

5 CONCLUSION

In this work, we have proposed a multi-Stream deep networks for Vehicle Make and Model Recognition. The proposed approach combines global representation with local representations using a dynamic fully-connected layer, the multi-stream architecture allows the system to use specialized feature extractors to detect subtle inter-class variations. It also allows the combination of variable number of vehicle part thanks to the dynamic fusion layer. Our experiments shows that our model provides efficient results on the publicly available CompCars dataset.

REFERENCES

Bay, H., Ess, A., Tuytelaars, T., and Van Gool, L. (2008). Speeded-up robust features (surf). *Comput. Vis. Image Underst.*, 110(3):346–359.

Biglari, M., Soleimani, A., and Hassanpour, H. (2018). A cascaded part-based system for fine-grained vehicle classification. *IEEE Transactions on Intelligent Transportation Systems*, 19(1):273–283.

Cheang, T. K., Chong, Y. S., and Tay, Y. H. (2017). Segmentation-free vehicle license plate recognition using convnet-rnn. *CoRR*, abs/1701.06439.

Du, S., Ibrahim, M., Shehata, M., and Badawy, W. (2013). Automatic license plate recognition (alpr): A state-of-the-art review. *IEEE Transactions on Circuits and Systems for Video Technology*, 23(2):311–325.

Gao, Y. and Lee, H. J. (2015). Vehicle make recognition based on convolutional neural network. In *2015 2nd International Conference on Information Science and Security (ICISS)*, pages 1–4.

Ghassemi, S., Fiandrotti, A., Caimotti, E., Francini, G., and Magli, E. (2018). Vehicle joint make and model recognitionwith multiscale attention windows. *Signal Processing: Image Communication*, 72.

Gou, C., Wang, K., Yao, Y., and Li, Z. (2016). Vehicle license plate recognition based on extremal regions and restricted boltzmann machines. *IEEE Transactions on Intelligent Transportation Systems*, 17(4):1096–1107.

He, H., Shao, Z., and Tan, J. (2015). Recognition of car makes and models from a single traffic-camera image. *IEEE Transactions on Intelligent Transportation Systems*, 16(6):3182–3192.

He, K., Zhang, X., Ren, S., and Sun, J. (2015). Deep residual learning for image recognition. *CoRR*, abs/1512.03385.

Hsieh, J., Chen, L., and Chen, D. (2014a). Symmetrical surf and its applications to vehicle detection and vehicle make and model recognition. *IEEE Transactions on Intelligent Transportation Systems*, 15(1):6–20.

Hsieh, J., Chen, L., and Chen, D. (2014b). Symmetrical surf and its applications to vehicle detection and vehicle make and model recognition. *IEEE Transactions on Intelligent Transportation Systems*, 15(1):6–20.

Hsu, G., Chen, J., and Chung, Y. (2013). Application-oriented license plate recognition. *IEEE Transactions on Vehicular Technology*, 62(2):552–561.

Hu, Q., Wang, H., Li, T., and Shen, C. (2017). Deep cnns with spatially weighted pooling for fine-grained car recognition. *IEEE Transactions on Intelligent Transportation Systems*, 18(11):3147–3156.

Jaderberg, M., Simonyan, K., Zisserman, A., and Kavukcuoglu, K. (2015). Spatial transformer networks. *CoRR*, abs/1506.02025.

Khan, S. M., Cheng, H., Matthies, D., and Sawhney, H. (2010). 3d model based vehicle classification in aerial imagery. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 1681–1687.

L. Yang, P. Luo, C. C. L. X. T. (2015). A large-scale car dataset for fine-grained categorization and verification. arXiv:1506.08959.

Lai, A. H. S., Fung, G. S. K., and Yung, N. H. C. (2001). Vehicle type classification from visual-based dimension estimation. In *ITSC 2001. 2001 IEEE Intelligent Transportation Systems. Proceedings (Cat. No.01TH8585)*, pages 201–206.

Li, H. and Shen, C. (2016). Reading car license plates using deep convolutional neural networks and lstms. *CoRR*, abs/1601.05610.

Li, H., Wang, P., and Shen, C. (2017). Towards end-to-end car license plates detection and recognition with deep neural networks. *CoRR*, abs/1709.08828.

Lowe, D. G. (1999). Object recognition from local scale-invariant features. In *Proceedings of the Seventh IEEE International Conference on Computer Vision*, volume 2, pages 1150–1157 vol.2.

Lowe, D. G. (2001). Local feature view clustering for 3d object recognition. In *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001*, volume 1, pages I–I.

Lowe, D. G. (2004). Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vision*, 60(2):91–110.

Luvizon, D. C., Picard, D., and Tabia, H. (2018). 2d/3d pose estimation and action recognition using multitask deep learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5137–5146.

Masood, S. Z., Shu, G., Dehghan, A., and Ortiz, E. G. (2017). License plate detection and recognition using deeply learned convolutional neural networks. *CoRR*, abs/1703.07330.

Paumard, M.-M., Picard, D., and Tabia, H. (2018). Image reassembly combining deep learning and shortest path problem. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 153–167.

Psyllos, A., Anagnostopoulos, C.-N., and Kayafas, E. (2009). Sift-based measurements for vehicle model recognition. 1.

Psyllos, A. P., Anagnostopoulos, C. E., and Kayafas, E. (2010). Vehicle logo recognition using a sift-based enhanced matching scheme. *IEEE Transactions on Intelligent Transportation Systems*, 11(2):322–328.

Redmon, J. and Farhadi, A. (2018). Yolov3: An incremental improvement. *arXiv*.

Sam, D. B., Surya, S., and Babu, R. V. (2017). Switching convolutional neural network for crowd counting. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4031–4039. IEEE.

Simonyan, K. and Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556.

Sochor, J., Herout, A., and Havel, J. (2016). Boxcars: 3d boxes as cnn input for improved fine-grained vehicle recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3006–3015.

Wei Wu, Zhang QiSen, and Wang Mingjun (2001). A method of vehicle classification using models and neural networks. In *IEEE VTS 53rd Vehicular Technology Conference, Spring 2001. Proceedings (Cat. No.01CH37202)*, volume 4, pages 3022–3026 vol.4.

Xiaoxu Ma and Grimson, W. E. L. (2005). Edge-based rich representation for vehicle classification. In *Tenth IEEE International Conference on Computer Vision (ICCV'05) Volume 1*, volume 2, pages 1185–1192 Vol. 2.

Yang, L., Luo, P., Loy, C. C., and Tang, X. (2015). A large-scale car dataset for fine-grained categorization and verification. *CoRR*, abs/1506.08959.

Yishu Peng, Yunhui Yan, Wenjie Zhu, and Jiuliang Zhao (2014). Vehicle classification using sparse coding and spatial pyramid matching. In *17th International IEEE Conference on Intelligent Transportation Systems (ITSC)*, pages 259–263.