BoxCars: 3D Boxes as CNN Input for Improved Fine-Grained Vehicle Recognition

Jakub Sochor[*], Adam Herout, Jiří Havel
Graph@FIT, Brno University of Technology
Brno, Czech Republic
{isochor,herout,ihavel}@fit.vutbr.cz

Abstract

We are dealing with the problem of fine-grained vehicle make & model recognition and verification. Our contribution is showing that extracting additional data from the video stream – besides the vehicle image itself – and feeding it into the deep convolutional neural network boosts the recognition performance considerably. This additional information includes: 3D vehicle bounding box used for "unpacking" the vehicle image, its rasterized low-resolution shape, and information about the 3D vehicle orientation. Experiments show that adding such information decreases classification error by 26 % (the accuracy is improved from 0.772 to 0.832) and boosts verification average precision by 208 % (0.378 to 0.785) compared to baseline pure CNN without any input modifications. Also, the pure baseline CNN outperforms the recent state of the art solution by 0.081. We provide an annotated set "BoxCars" of surveillance vehicle images augmented by various automatically extracted auxiliary information. Our approach and the dataset can considerably improve the performance of traffic surveillance systems.

1. Introduction

We are developing a system for traffic surveillance from roadside cameras. It is meant to be fully automatic (not requiring manual per-camera configuration) and tolerant to sub-optimal camera placement (the cameras will not be placed above the lanes, but on the road side, wherever it is naturally possible).

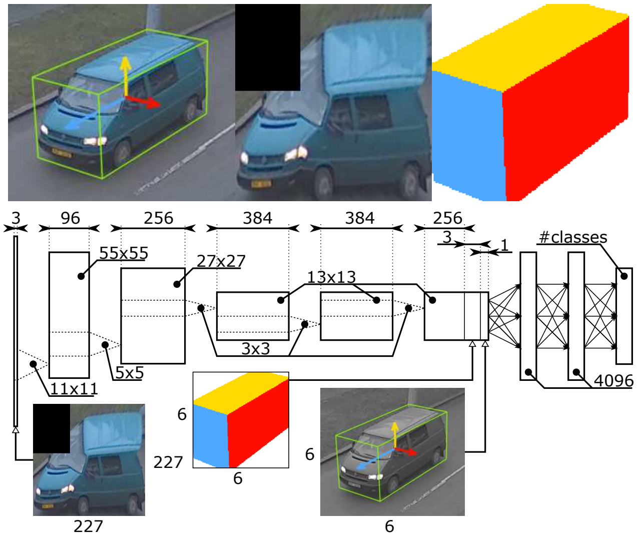

Figure 1. We take advantage of the surveillance camera being fixed, use its automatically obtained calibration to *unpack* the input image so that it is better aligned, and we add additional inputs to the CNN. These modified inputs boost the performance of vehicle recognition and especially vehicle make & model verification.

One important component of such a system is recognition of vehicle make & model – as accurate as possible. This fine-grained recognition serves multiple purposes. Besides obvious collection of statistics and demographic information and verification of license plate authenticity, recognition of dominant and characteristic types can establish a highly accurate scale calibration of the camera, much more precise than a statistic of undifferentiated cars [8]. The system should also be able to adapt to new models of cars on its own. It should therefore not only recognize the pre-trained set of models, but also *verify* whether two given vehicle samples are of the same make & model or not – without previously seeing these particular vehicle types.

Fine-grained vehicle recognition has been receiving increased research attention recently. Many works and datasets focus on recognition of "web images" shot from a limited set of viewpoints, typically from eye-level views [43, 16, 19, 13, 25, 36]. Some works also deal with data of surveillance nature [43, 22, 14].

Our work goes beyond a recent work by Yang et al. [43]. They collected presumably the first dataset of sufficient proportion for training convolutional neural networks (the

[*] Jakub Sochor is a Brno Ph.D. Talent Scholarship Holder — Funded by the Brno City Municipality. This work was supported by TACR project "RODOS", TE01020155.

surveillance-nature portion of their dataset contains around 50k samples). They also propose a CNN architecture for fine-grained vehicle recognition and publish benchmarking results.

Since we aim at a fixed-camera surveillance system, we take advantage of fully automatic camera calibration including scale [8] and we use the automatically extracted information for improving the recognition system (Fig. 1). The automatically calibrated camera allows us to extract a 3D bounding box of the passing vehicle. The system then "unpacks" the bounding box to get a better aligned image representation. The shape and location of the bounding box is also input to the CNN and helps it to reference the relevant information. Lastly, the view direction extracted for each vehicle sample is also encoded and input to the fully connected CNN layers, further boosting the performance. The whole algorithm is designed to work with low-resolution vehicle images taken from arbitrary viewpoints of surveillance nature (frontal, sideways, varying elevation, etc.).

We collected a dataset *BoxCars* from a network of surveillance cameras and we make it publicly available for training and benchmarking. The cameras are automatically calibrated and the vehicle samples are automatically augmented by the 3D bounding box information. This information is easily obtainable in real time and it can be a part of any surveillance system.

The experiments show that the proposed enhanced information boosts the average precision of vehicle recognition considerably (0.772 to 0.832 for medium difficulty, 0.733 to 0.804 for hard cases). The same modification helps even much more for the *vehicle type verification* task: given observations of two vehicles, tell if they are of the same type (in the fine-grained sense, i.e. including make, model, year). The particular vehicle types have not been necessarily seen by the classifier during training. The improvement in this task was from 0.378 to 0.785 for medium difficulty samples and 0.353 to 0.710 for difficult cases. This verification task is important for growing the set of vehicles recognizable by the system in an unsupervised manner – without collecting and annotating the samples in advance.

The **contributions of this paper** are the following: **i)** We show that additional information easily obtainable in real time for static surveillance cameras can boost the CNN verification performance greatly (by 208 %), **ii)** The vehicle fine-grained classification error was decreased by 26 %, **iii)** We collected a dataset of vehicle samples accompanied with the 3D bounding boxes (*BoxCars*, 21,250 samples, 63,750 images, 27 different makes, 148 make & model + submodel + model year classes).

2. Related Work

When it comes to fine-grained vehicle classification, many approaches are limited to frontal or rear viewpoint and they are based on detection of the license plate for ROI extraction [32, 7, 31, 27, 44, 2]. Authors of these papers are using different schemes for extracting the feature vectors and for the classification itself. Stark et al. [36] use fine-grained categorization of cars by DPM in order to obtain metric information and get a rough estimate of depth information for single images (containing cars in usable poses). Another approach proposed by Prokaj and Medioni [33] is based on pose estimation and it is able to handle any viewpoint. The authors suggest to use 3D models of vehicles, fit them to the recognized pose, project them to 2D and use SIFT-like features for the comparison of the vehicles. Krause et al. [19] used 3D CAD models to train geometry classifiers and improve results of 3D versions of Spatial Pyramid and BubbleBank [6] by 3D patch sampling and rectification. Lin et al. [25] proposed to use 3D Active Shape Model fitting to obtain positions of landmarks and achieved much better results than other methods on their own dataset FG3DCar. Authors of [17] propose to learn discriminative parts of vehicles with CNN and use the parts for fine-grained classification. Gu et al. [13] used pose estimation and active shape matching to deal with pose variation and normalization. Hsiao et al. [15] use 3D chamfer matching of backprojected curves on an automatically generated visual hull of the vehicle. However, the authors assume to have shots of vehicles against clean background and that the shots are taken under regular intervals.

Very recent work by He et al. [14] focuses on surveillance images; however, the authors assume to have high-resolution frontal image of the vehicle to correctly detect license plate and other artificial anchors. Liao et al. [22] used Strongly Supervised DPM (SSDPM) to categorize frontal images of vehicles and classification based on discriminative power of different parts of SSDPM. Hsieh et al. [16] proposed a new symmetrical SURF keypoint detector to detect and segment frontal vehicle images into several grids for fine-grained classification. Very recent work by Yang et al. [43] proposed to use Convolutional Neural Networks for fine-grained classification, regression of parameters etc. Krause et al. [18] proposed to use co-segmentation and automatic part localization in combination with R-CNN to overcome missing parts annotations.

Recently, Deep Convolutional Neural Networks (CNN) consistently succeed in hard image recognition tasks such as the ImageNet [34] contest. After the network by Krizhevsky et al. [20], deeper and more complex CNNs such as the GoogLeNet by Szegedy et al. [38] seem to be consistently winning the contest. Authors also used input normalization to improve performance of CNN [39] and adding additional training data to CNN [20]. Parts of the CNN can be viewed as feature extractors and independently reused. These trained feature extractors outperform the hand-crafted features [3, 39]. Recently, a relatively large number of authors

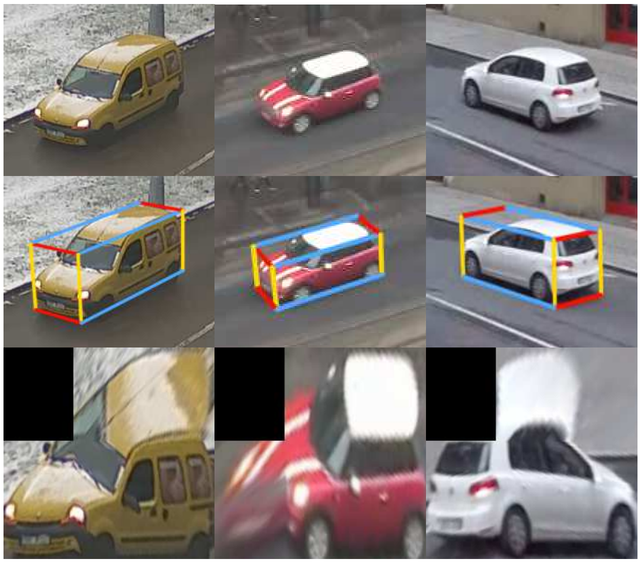

Figure 2. Examples of vehicles (top), their 3D bounding boxes (middle) and unpacked version of the vehicles (bottom).

proposed to use Deep Neural Networks for fine-grained classification in general [39, 41, 43, 23, 42, 9, 24].

To sum up, in most cases, the existing approaches either use 2D frontal images, or 3D CAD models to allow viewpoint invariance. We propose to extract and use 3D information based on video data from the surveillance camera at general viewpoints. This information is fed to a CNN as additional input, leading to better car classification and especially type verification.

3. Fine-Grained Vehicle Classification and Verification Methodology

In agreement with the recent progress in the Convolutional Neural Networks [39, 20, 5], we propose to use CNN for both classification and verification. The classification task will be done directly by the net and for the verification task, we use features (activations) extracted from the last-but-one layer and cosine distance. We enhance the input of the net by several techniques using automatically extracted 3D bounding boxes [8]. We focus on vehicle images obtained from surveillance cameras where the automatic extraction of 3D bounding boxes is possible cheaply in real time. We used BVLC Reference CaffeNet [20] pretrained on ImageNet [34] and then fine-tuned on our dataset as a baseline from which we improve.

3.1. Unpacking the Vehicles' Images

We based our work on 3D bounding boxes [8] (Fig. 2) which can be automatically obtained for each vehicle seen by a surveillance camera (see our original paper [8] for further details). These boxes allow us to identify side, roof, and front/rear side of vehicles in addition to other information about the vehicles. We use these localized segments to normalize the image of observed vehicles.

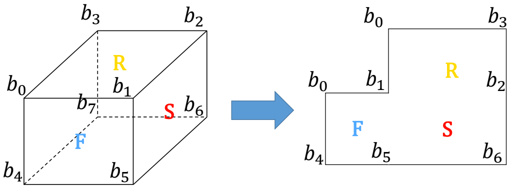

Figure 3. Unpacking the input vehicle image based on its bouding box. Points b_i are vertices of the 3D bounding box [8].

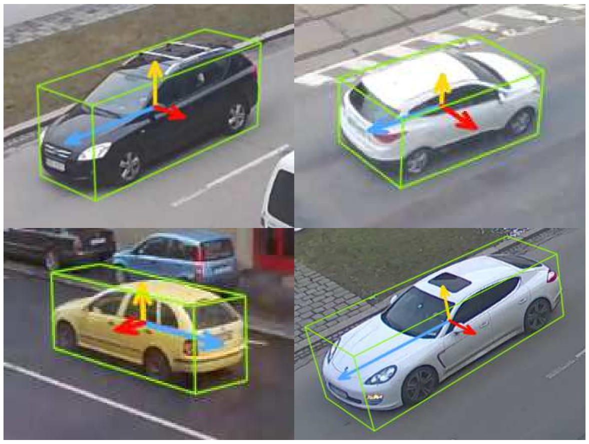

Figure 4. Examples of vectors encoding the viewpoint.

The normalization is done by unpacking the image into a plane. The plane contains rectified versions of the front/rear (**F**), side (**S**), and roof (**R**). These parts are adjacent to each other (Fig. 3) and they are organized into the final matrix **U**:

$$\mathbf{U} = \begin{pmatrix} \mathbf{0} & \mathbf{R} \\ \mathbf{F} & \mathbf{S} \end{pmatrix} \qquad (1)$$

The unpacking itself is done by obtaining homography between points b_i (Fig. 3) and perspective warping parts of the original image. The left top submatrix is filled with zeros. This unpacked version of the vehicle can be used instead of the original image to feed the net. The unpacking is beneficial as it localizes parts of the vehicles, normalizes their position in the image and all that without the necessity to use DPM or other algorithms for part localization.

3.2. Viewpoint Encoding

We also found out that it improves the results when the net is aware of the viewpoint of the vehicles. The viewpoint is extracted from the orientation of the 3D bounding box – Fig. 4. We encode the viewpoint as three 2D vectors v_i, where $i \in \{f, s, r\}$ (*front/rear*, *side*, *roof*) and pass them to the net. Vectors v_i are connecting the center of the bounding box with the centers of the box's faces. Therefore, it can be computed as $v_i = \overrightarrow{C_bC_i}$. Point C_b is the center of the bounding box and it can be obtained as the intersection of

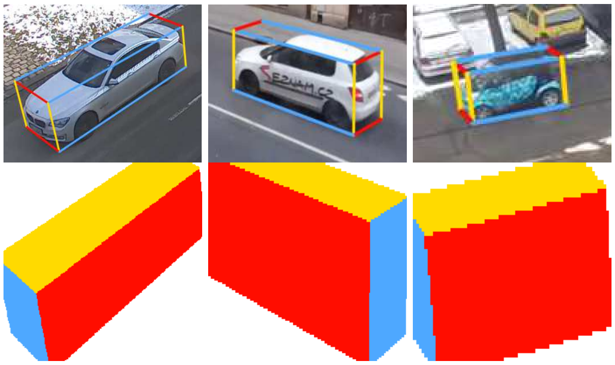

Figure 5. Examples of rasterized bounding boxes for CNN (colors are R,G,B in the actual computation, but are changed here for aesthetic reasons).

diagonals $\overleftrightarrow{b_2b_4}$ and $\overleftrightarrow{b_5b_3}$. Points C_i for $i \in \{f, s, r\}$ denote the centers of each face, again computed as intersections of face diagonals. The vectors are normalized to have unit size; storing them with a different normalization (e.g. the front one normalized, the other in the proper ratio) did not improve the results.

3.3. Rasterized Bounding Boxes

Another way of encoding the viewpoint and also the relative dimensions of vehicles is to rasterize the 3D bounding box and use it as an additional input to the net. The rasterization is done separately for all sides, each filled by one color. The final rasterized bounding box is then a three-channel image containing each visible face rasterized in a different channel. Formally, point (x, y) of the rasterized bounding box $\mathbf{T}$ is obtained as

$$\mathbf{T}_{x,y} = \begin{cases} (1,0,0) & (x,y) \in \text{▱} b_0 b_1 b_4 b_5 \\ (0,1,0) & (x,y) \in \text{▱} b_1 b_2 b_5 b_6 \\ (0,0,1) & (x,y) \in \text{▱} b_0 b_1 b_2 b_3 \\ (0,0,0) & \text{otherwise} \end{cases} \qquad (2)$$

where $\text{▱} b_0 b_1 b_4 b_5$ denotes the quadrilateral defined by points b_0, b_1, b_4 and b_5 in Figure 3.

Finally, the 3D rasterized bounding box is cropped by the 2D bounding box of the vehicle. For an example, see Figure 5, showing rasterized bounding boxes for different vehicles taken from different viewpoints.

3.4. Final CNN Using Images + Auxiliary Input

All this information is finally passed to the CNN (Fig. 1). The unpacked version of vehicles is used directly as the image input instead of the original image. The rasterized bounding box and encoded viewpoints are added to the net after the convolutional layers. We experimented with changing the layer where the information is added but different positions did not improve the results further and the mentioned setting is easiest with regard to pre-training the network.

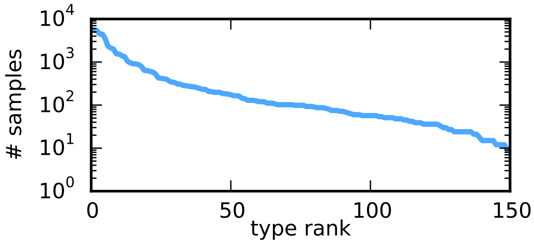

Figure 6. Distribution of samples in our *BoxCars* dataset across vehicle types. The distribution corresponds to real-life occurence of the models.

As the auxiliary input is added after the convolutional layers, it needs to be passed in 6×6 matrices. The rasterized bounding box is rescaled (Lanczos interpolation) to 6×6 and added to the net. The encoded viewpoints are added to the net in 6×6 one-channel matrix with zeros everywhere except for the first row which contains normalized vectors encoding the viewpoint. The first row $\mathbf{t}$ of this matrix contains all three 2D vectors: $\mathbf{t} = (v^x_f, v^y_f, v^x_r, v^y_r, v^x_s, v^y_s)$.

For better understanding of the text, we define labels for the nets with different input modifications. The original CNN processing cropped images of vehicles without any modifications is referenced as **baseline**. Network denoted as **Rast** contains the rasterized bounding boxes, **View** net is augmented by the encoded viewpoints, and in **Unp** version of the net, the original image is replaced by the unpacked image of vehicles. All these input modifications can be combined, yielding **RastView**, **RastUnp** and **RastViewUnp** nets.

4. BoxCars: New Dataset for Surveillance Vehicle Verification

There is a large number of datasets of vehicles [34, 1, 30, 10, 40, 4, 28, 21, 12, 35, 11, 29, 26] which are usable mainly for vehicle detection, pose estimation, and other tasks. However, these datasets do not contain annotation of the precise vehicles' make & model.

When it comes to the fine-grained datasets, a few of them exist and all are quite recent. Lin et al. [25] published FG3DCar dataset (300 images, 30 classes), Stark et al. [36] made another dataset containing 1,904 vehicles from 14 classes. Krause et al. [19] published two datasets; one of them, called *Car Types*, contains 16k of images and 196 classes. The other one, *BMW 10*, is made of 10 models of BMW vehicles and 500 images. Finally, Liao et al. [22] created a dataset of 1,482 vehicles from 8 classes. All these datasets are relatively small for training the CNN for real-world surveillance tasks.

Yang et al. [43] published a large dataset *CompCars* this year (2015). The dataset consists of a web-nature part, made of 136k of vehicles from 1,600 classes taken from different viewpoints. Then, it also contains a surveillance-nature part

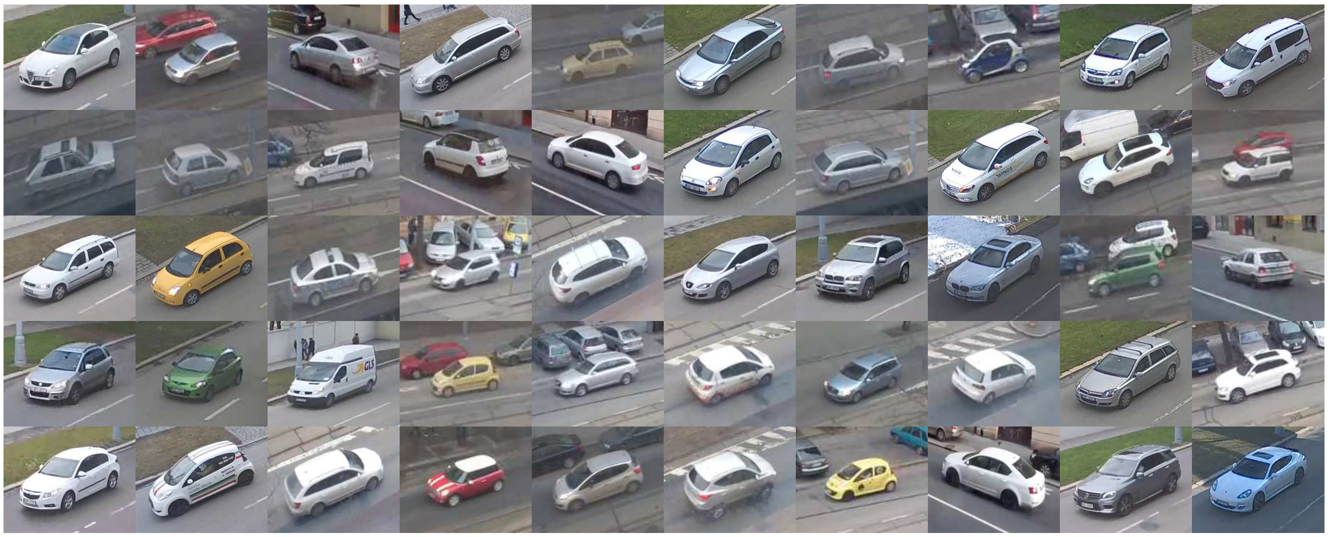

Figure 7. A sample of the novel *BoxCars* dataset. In total, it captures 21,250 vehicles in 63,750 images, from 27 different makes (148 fine-grained classes).

with 50k frontal images of vehicles taken from surveillance cameras.

We collected and annotated a new dataset *BoxCars*. The dataset is focused on images taken from surveillance cameras as it is meant to be useful for traffic surveillance applications. We do not restrict that the vehicles are taken from the frontal side (Fig. 7). We used surveillance cameras mounted near streets and tracked the passing vehicles. Each correctly detected vehicle is captured in 3 images, as it is passing by the camera; therefore, we have more visual information about each vehicle. The dataset contains 21,250 vehicles (63,750 images) of 27 different makes. The vehicles are divided into classes: there are 102 make & model classes, 126 make & model + submodel classes, and 148 make & model + submodel + model year classes. The distribution of types in the dataset is shown in Figure 6 and samples from the dataset are in Figure 7. The data include information about the 3D bounding box [8] for each vehicle and an image with a foreground mask extracted by background subtraction [37, 45]. The dataset is made publicly available[1] for future reference and evaluation.

Our proposed dataset is difficult in comparison with other existing datasets in size of the images (thousands of pixels, min/mean/max): *CompCars* – 107/503/1114, *Cars-196* – 4/479/42120, *BoxCars* – 8/39/253. Also, the samples in our dataset are compressed by realistic h264 codec settings, and unlike most existing surveillance datasets, our viewpoints are diverse and not just frontal/rear.

5. Experimental Results

The evaluation of the improvement caused by our modifications of the CNN input can be only done on our *BoxCars* dataset as other fine-grained datasets listed in Section 4 do not include the information about the bounding boxes. However, to put the performance of the system into context with other published methods, we evaluated the BVLC Reference net [20] on the most recent dataset *CompCars* and the improvement will be measured relatively to this baseline.

	Top-1	Top-5
[43]	0.767	0.917
Ours	**0.848**	**0.954**

Table 1. Comparison of **classification** results on the *CompCars* [43] dataset (accuracy).

	Easy	Medium	Hard
[43]	0.833	0.824	0.761
Ours	**0.850**	**0.827**	**0.768**

Table 2. Comparison of **verification** results on the *CompCars* dataset (accuracy).

5.1. Evaluation on the CompCars Dataset

We trained the baseline net on the *CompCars* dataset [43] and evaluated its accuracy. As Table 1 shows, this net significantly outperforms the CNN used by Yang et al. [43] in their paper in both Top-1 and Top-5 categories. We used the All-view split as the authors achieved the best results if they did not take the viewpoint into account, but instead, they trained a common net on all the viewpoints at once.

We also evaluated the make & model verification accuracy using the activations extracted from the baseline net and cosine distance. The results are shown in Table 2 and our system outperforms the original paper in verification

[1] https://medusa.fit.vutbr.cz/traffic

	# types	# training samples	# test samples
medium	77	40,152	19,590
hard	87	37,689	18,939

Table 3. Training and testing set sizes for the classifications task. Numbers of samples represent the amount of all images used. The number of unique vehicles is one third of these counts.

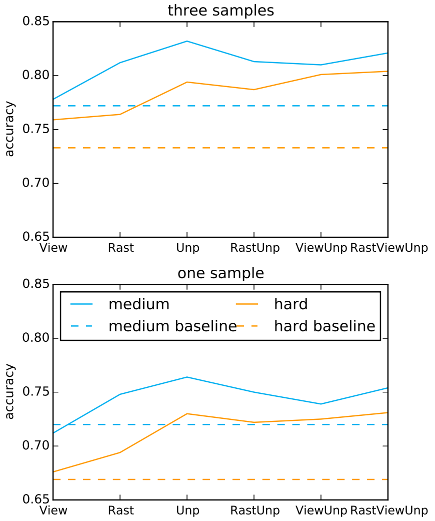

Figure 8. Top-1 accuracies for different input modifications. The dashed line represent the baseline accuracy achieved by the CNN without any input modification. The method identifiers are explained in Sec. 3.4.

as well. It should be noted that the *CompCars* verification dataset has a high random baseline (0.5).

Since the baseline net outperforms the method published by Yang et al. [43], we measure the improvement achieved by our modifications of the CNN input relatively to the performance of this baseline net on our *BoxCars* dataset.

5.2. Classification Results

We defined two ways of splitting the *BoxCars* dataset into the training and testing parts. In both of them, *medium* and *hard*, vehicles taken from 70 % of cameras are included in the training part and the vehicles taken by the rest of the cameras are in the test set. The difference of viewpoints between the training and the test sets is not too large, as it would be if for example the rear views would be in the training set and frontal views in the test set. This kind of splitting is suitable for benchmarking surveillance algorithms because real-life applications would also use cameras placed in roughly predictable viewpoints. The difference between the *medium* and *hard* splittings is that we consider vehicles of the same make+model+submodel but differing in their model year as the same types in the *medium* dataset. In the *hard* dataset, we differentiate also the model year. For stability of the classification, types with too few samples were omitted and the training/testing set sizes can be found in Table 3 (this approach is consistent with [43]).

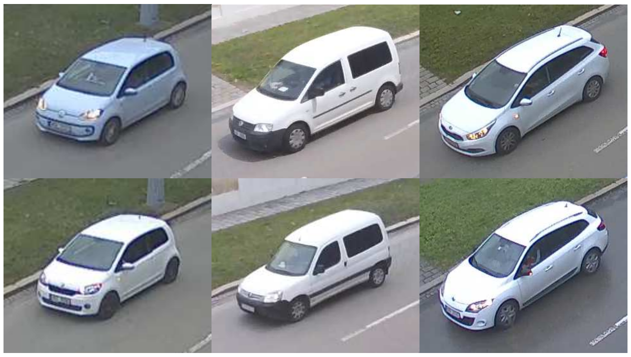

Figure 9. Most misclassified vehicle types for the RastViewUnp version of the net. **left:** Volkswagen Up and Skoda Citigo, **middle:** Volkswagen Caddy and Citroen Berlingo, **right:** Kia Ceed and Renault Megane.

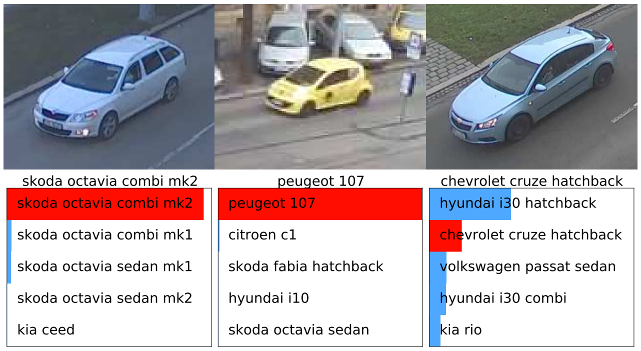

Figure 10. Examples of types probabilities for different vehicles for the RastViewUnp version of the net. Only one sample was used for the classification and the model year was differentied in the first example.

All the CNNs were pre-trained on ImageNet [34] and then fine-tuned on one of the *medium* or *hard* datasets. When the rasterized bounding boxes or encoded viewpoints are introduced to the nets, the weights of fully connected layers are randomly re-initialized and in that case we do not use the pre-trained weights on ImageNet in those layers.

We evaluated all the net's input modifications and also their combinations. The results are shown in Table 4 and Figure 8. As we have multiple samples for each vehicle, we can use mean probability for each vehicle type and achieve better results, see Table 5. The improvement between one sample and three samples is 0.073 (0.731 to 0.804 in Top-1 accuracy). Also, Table 5 shows that the improvement achieved by the modified CNN in the *medium* dataset is 0.060 (0.772 to 0.832) and 0.071 (0.733 to 0.804) in the *hard* case.

We consider the improvement in classification accuracy as interesting because the task itself is complex and difficult even for a human. Also, the classification error was reduced by 26 % for both splittings. Consider the examples of the most confused types shown in Figure 9, where

	baseline		Unp		Rast		View		RastUnp		ViewUnp		RastViewUnp	
	med	hard	med	hard	med	hard	med	hard	med	hard	med	hard	med	hard
Top-1	0.720	0.669	**0.764**	0.730	0.748	0.694	0.712	0.676	0.750	0.722	0.739	0.725	0.754	**0.731**
Top-5	0.910	0.883	**0.915**	**0.897**	0.903	0.872	0.885	0.865	0.891	0.882	0.894	0.883	0.901	0.890

Table 4. Comparison of classification accuracy results for one sample per vehicle. The method identifiers are explained in Sec. 3.4.

	baseline		Unp		Rast		View		RastUnp		ViewUnp		RastViewUnp	
	med	hard	med	hard	med	hard	med	hard	med	hard	med	hard	med	hard
Top-1	0.772	0.733	**0.832**	0.794	0.812	0.764	0.778	0.759	0.813	0.787	0.810	0.801	0.821	**0.804**
Top-5	0.924	0.903	**0.945**	0.926	0.927	0.906	0.913	0.901	0.928	0.923	0.930	0.919	0.937	**0.929**

Table 5. Comparison of classification accuracy results for three samples per vehicle, the final probability for a class is obtained as mean probability over the samples.

Volkswagen Up and *Skoda Citigo* are manufactured in the same production plant and they differ only in subtle branding parts in the region of the frontal mask. Also, Figure 10 shows examples of probabilities obtained for different vehicles. These graphs indicate that the net is aware of the sample being similar to multiple types and that it can safely distinguish from completely disparate models.

The experimental results indicate that the most important improvement is unpacking the image (Section 3.1), presumably because it leads to better alignment of the vehicle features on the input CNN level. The further input modifications help only in the *hard* splitting, where subtler details make a difference.

5.3. Verification Results

Verification of pairs of vehicle types (for two vehicle samples decide: same types / different types) is as important as classification, especially when it comes to reasoning about unseen and untrained vehicle types. We selected even more difficult splitting for evaluation of the verification performance. Only some cameras are present in the training set (the same ones as in the classification task) and only some vehicle types are present in the training set. The testing is done on pairs of randomly selected 3,000 vehicles mainly taken from cameras which were not present during training (over 80 % of vehicles is from unseen cameras) and the testing set of vehicles also contains types which were not seen during training (over 10 % of samples, approximately 25 % of pairs contain at least one vehicle of an unseen type). Thus, the algorithm is required to verify unseen types of vehicles taken from unseen viewpoints and it has to generalize well.

We have three splittings for the verification task. *Easy* contains pairs of vehicles from the same unseen camera, *medium* contains pairs from different unseen cameras and finally, *hard* contains pairs of vehicles from different unseen cameras and the model year is also taken into account. The training/testing set sizes can be found in Table 6.

	training		testing	
	# types	# samples	# types	# pairs
easy	113	34 929	100	1 394 008
medium	113	34 929	99	1 435 532
hard	126	32 658	113	1 501 156

Table 6. Training and test sets sizes for the verification task. The number of training samples represent the number of images used in training. The number of unique training vehicles is one third of this number.

Again, we used nets pre-trained on ImageNet [34], fine-tuned them during the training and then used the features from the last fully connected layer (fc7) and compared them by cosine distance. Two different training passes were done, one for the *easy* and *medium* splitting (both splittings do not take model year into account) and one for the *hard* one.

We evaluated the algorithm on the three dataset splittings using only one sample for each vehicle and the results are in Figure 11. When the three samples are taken into account by working with the median cosine distance, the results improve as shown in Figure 12. Using the median cosine distance improved the average precision on average by 0.094.

The plots show that our CNN input modifications have a huge impact on the average precision in the verification task. For example, considering the *medium* set and the median cosine distance over the three samples, *RastViewUnp* improved AP of the baseline CNN by 208 %. Figure 13 shows what improved the average precision in verification. The numbers gradually increase as we add more and more modifications of the CNN input. It is rather interesting that both rasterized bounding boxes and orientation encoding help the net, even in combination; we expected that these two would be alternative ways of encoding the same information, but apparently, they encode it slightly differently and both turn out to be helpful.

We also obtained vehicle type verification precision and recall of human subjects for the *BoxCars* dataset. We randomly selected 1,000 pairs of vehicles; one third of the

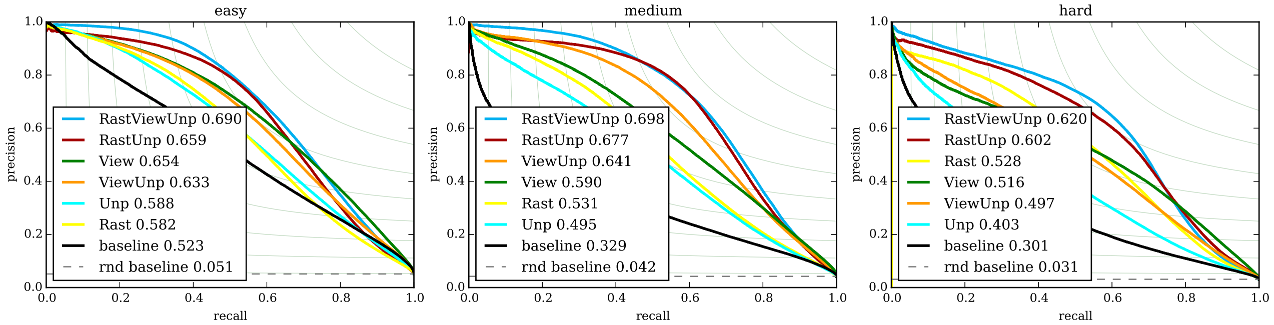

Figure 11. Precision-Recall curves for different verification dataset splittings. Only one sample was used for the verification. Numbers denote Average Precision. The method identifiers are explained in Sec. 3.4; *rnd baseline* denotes random baseline based on the number of positive pairs. Numbers in legend denote Average Precision.

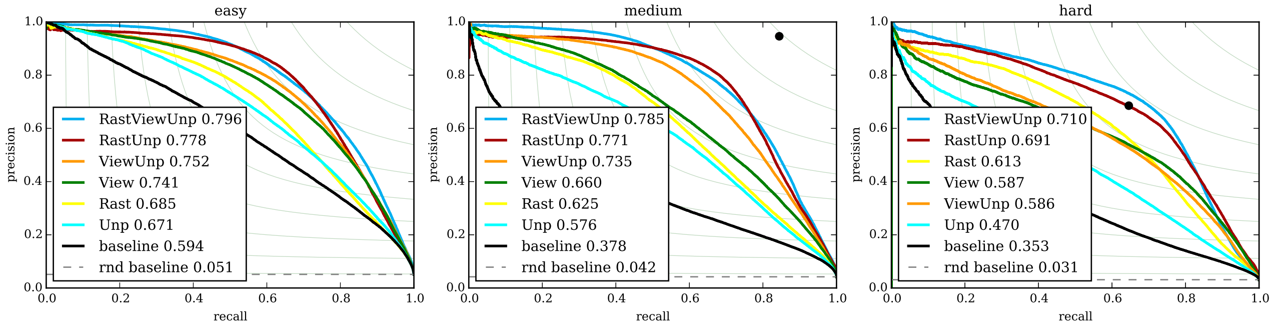

Figure 12. Precision-Recall curves for different verification dataset splittings. Median cosine distance over three vehicle samples was used in this case. **Black dots** represent mean precision and recall obtained by human annotators, see text for details. Numbers in legend denote Average Precision.

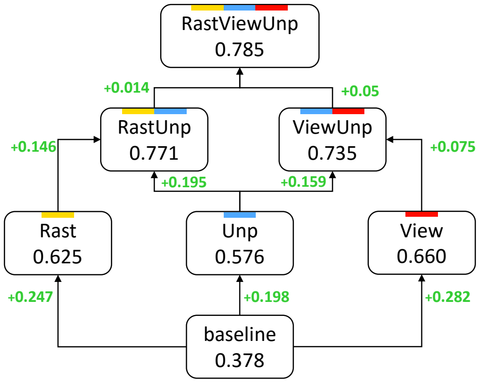

Figure 13. Schematic image of improvements in the verification AP for different CNN input modifications. The image is based on the *medium* splitting with median distance over the samples.

pairs included vehicles of different types, one third of pairs had the same make & model + submodel, but differed in the model year, and the last third contained pairs of vehicles of the same type including model year. Participants were requested to manually indicate one of these three situations for each given pair of vehicles. All three captured images of the vehicles, taken by different cameras, were shown to the participants (that is why the human data is present in *medium* and *hard* cases in Fig. 12 but not in Fig. 11). We received a total of 8,011 inputs (8 per pair) with mean precision 0.946 and mean recall 0.844 for the *medium* case. On the other hand, the results show that the human annotators have problems with correctly distinguishing different model years (the *hard* case), with mean precision 0.685 and mean recall 0.646. These results are shown in Figure 12 as black dots; note that in the *hard* case, the system outperforms the human annotators.

6. Conclusions

Surveillance systems can and should benefit from the camera being fixed. The camera can be fully automatically calibrated and more information can be extracted for the passing vehicles. We show that this information considerably improves the fine-grained recognition by CNN, and tremendously boosts the verification task average precision.

Our dataset *BoxCars* is meant to help experiments in this direction by providing sufficient amount of data enriched by information which can be automatically extracted in real time in actual surveillance systems. We keep collecting samples from new surveillance cameras, so that the size of the dataset will gradually increase in near future.

References

[1] S. Agarwal, A. Awan, , and D. Roth. Learning to detect objects in images via a sparse, part-based representation. *IEEE PAMI*, 26(11):1475–1490, Nov. 2004.

[2] R. Baran, A. Glowacz, and A. Matiolanski. The efficient real- and non-real-time make and model recognition of cars. *Multimedia Tools and Applications*, 74(12):4269–4288, 2015.

[3] T. Bluche, H. Ney, and C. Kermorvant. Feature extraction with convolutional neural networks for handwritten word recognition. In *International Conference on Document Analysis and Recognition (ICDAR)*, pages 285–289, Aug 2013.

[4] C. Caraffi, T. Vojir, J. Trefny, J. Sochman, and J. Matas. A System for Real-time Detection and Tracking of Vehicles from a Single Car-mounted Camera. In *ITS Conference*, pages 975–982, Sep. 2012.

[5] K. Chatfield, K. Simonyan, A. Vedaldi, and A. Zisserman. Return of the devil in the details: Delving deep into convolutional nets. In *British Machine Vision Conference*, 2014.

[6] J. Deng, J. Krause, and L. Fei-Fei. Fine-grained crowdsourcing for fine-grained recognition. In *CVPR*, pages 580–587, June 2013.

[7] L. Dlagnekov and S. Belongie. Recognizing cars. Technical report, UCSD CSE Tech Report CS2005-0833, 2005.

[8] M. Dubská, J. Sochor, and A. Herout. Automatic camera calibration for traffic understanding. In *BMVC*, 2014.

[9] M. C. et al. Lr-cnn for fine-grained classification with varying resolution. In *IEEE International Conference Image Processing (ICIP)*, 2015.

[10] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The pascal visual object classes (VOC) challenge. *IJCV*, 88(2):303–338, June 2010.

[11] A. Geiger, P. Lenz, and R. Urtasun. Are we ready for autonomous driving? the KITTI vision benchmark suite. In *CVPR*, 2012.

[12] D. Glasner, M. Galun, S. Alpert, R. Basri, and G. Shakhnarovich. Viewpoint-aware object detection and continuous pose estimation. *Image&Vision Comp.*, 2012.

[13] H.-Z. Gu and S.-Y. Lee. Car model recognition by utilizing symmetric property to overcome severe pose variation. *Machine Vision and Applications*, 24(2):255–274, 2013.

[14] H. He, Z. Shao, and J. Tan. Recognition of car makes and models from a single traffic-camera image. *IEEE Transactions on Intelligent Transportation Systems*, PP(99):1–11, 2015.

[15] E. Hsiao, S. Sinha, K. Ramnath, S. Baker, L. Zitnick, and R. Szeliski. Car make and model recognition using 3D curve alignment. In *IEEE WACV*, March 2014.

[16] J.-W. Hsieh, L.-C. Chen, and D.-Y. Chen. Symmetrical surf and its applications to vehicle detection and vehicle make and model recognition. *Intelligent Transportation Systems, IEEE Transactions on*, 15(1):6–20, Feb 2014.

[17] J. Krause, T. Gebru, J. Deng, L. J. Li, and L. Fei-Fei. Learning features and parts for fine-grained recognition. In *Pattern Recognition (ICPR), 2014 22nd International Conference on*, pages 26–33, Aug 2014.

[18] J. Krause, H. Jin, J. Yang, and L. Fei-Fei. Fine-grained recognition without part annotations. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015.

[19] J. Krause, M. Stark, J. Deng, and L. Fei-Fei. 3D object representations for fine-grained categorization. In *ICCV Workshop 3dRR-13*, 2013.

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

[21] B. Leibe, N. Cornelis, K. Cornelis, and L. Van Gool. Dynamic 3D scene analysis from a moving vehicle. In *CVPR*, pages 1–8, June 2007.

[22] L. Liao, R. Hu, J. Xiao, Q. Wang, J. Xiao, and J. Chen. Exploiting effects of parts in fine-grained categorization of vehicles. In *International Conference on Image Processing (ICIP)*, 2015.

[23] D. Lin, X. Shen, C. Lu, and J. Jia. Deep lac: Deep localization, alignment and classification for fine-grained recognition. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015.

[24] T.-Y. Lin, A. RoyChowdhury, and S. Maji. Bilinear cnn models for fine-grained visual recognition. In *International Conference on Computer Vision (ICCV)*, 2015.

[25] Y.-L. Lin, V. I. Morariu, W. Hsu, and L. S. Davis. Jointly optimizing 3D model fitting and fine-grained classification. In *ECCV*, 2014.

[26] K. Matzen and N. Snavely. NYC3DCars: A dataset of 3D vehicles in geographic context. In *International Conference on Computer Vision (ICCV)*, 2013.

[27] P. Negri, X. Clady, M. Milgram, and R. Poulenard. An oriented-contour point based voting algorithm for vehicle type classification. In *ICPR*, pages 574–577, 2006.

[28] A. Opelt, M. Fussenegger, A. Pinz, and P. Auer. Generic object recognition with boosting. Technical Report TR-EMT-2004-01, EMT, TU Graz, Austria, 2004. Submitted to the IEEE Tr. PAMI.

[29] M. Özuysal, V. Lepetit, and P. Fua. Pose estimation for category specific multiview object localization. In *IEEE CVPR*, pages 778–785, June 2009.

[30] C. Papageorgiou and T. Poggio. A trainable object detection system: Car detection in static images. Technical Report 1673, October 1999. (CBCL Memo 180).

[31] G. Pearce and N. Pears. Automatic make and model recognition from frontal images of cars. In *IEEE AVSS*, pages 373–378, 2011.

[32] V. Petrovic and T. F. Cootes. Analysis of features for rigid structure vehicle type recognition. In *BMVC*, pages 587–596, 2004.

[33] J. Prokaj and G. Medioni. 3-D model based vehicle recognition. In *IEEE WACV*, Dec 2009.

[34] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *IJCV*, 2015.

[35] S. Savarese and L. Fei-Fei. 3D generic object categorization, localization and pose estimation. In *ICCV*. IEEE, 2007.

[36] M. Stark, J. Krause, B. Pepik, D. Meger, J. Little, B. Schiele, and D. Koller. Fine-grained categorization for 3D scene understanding. In *BMVC*, 2012.

[37] C. Stauffer and W. E. L. Grimson. Adaptive background mixture models for real-time tracking. In *CVPR*, volume 2, pages 246–252, 1999.

[38] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. *CoRR*, abs/1409.4842, 2014.

[39] Y. Taigman, M. Yang, M. Ranzato, and L. Wolf. DeepFace: Closing the gap to human-level performance in face verification. In *CVPR*, pages 1701–1708, 2014.

[40] Y. Xiang and S. Savarese. Estimating the aspect layout of object categories. In *CVPR*, pages 3410–3417, 2012.

[41] T. Xiao, Y. Xu, K. Yang, J. Zhang, Y. Peng, and Z. Zhang. The application of two-level attention models in deep convolutional neural network for fine-grained image classification. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015.

[42] S. Xie, T. Yang, X. Wang, and Y. Lin. Hyper-class augmented and regularized deep learning for fine-grained image classification. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015.

[43] L. Yang, P. Luo, C. Change Loy, and X. Tang. A large-scale car dataset for fine-grained categorization and verification. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015.

[44] B. Zhang. Reliable classification of vehicle types based on cascade classifier ensembles. *IEEE Transactions on Intelligent Transportation Systems*, 14(1):322–332, March 2013.

[45] Z. Zivkovic. Improved adaptive gaussian mixture model for background subtraction. In *ICPR*, pages 28–31, 2004.