BoxCars: Improving Fine-Grained Recognition of Vehicles using 3D Bounding Boxes in Traffic Surveillance

Jakub Sochor, Jakub Špaňhel, Adam Herout

***Abstract*—In this paper, we focus on fine-grained recognition of vehicles mainly in traffic surveillance applications. We propose an approach orthogonal to recent advancement in fine-grained recognition (automatic part discovery, bilinear pooling). Also, in contrast to other methods focused on fine-grained recognition of vehicles, we do not limit ourselves to frontal/rear viewpoint but allow the vehicles to be seen from any viewpoint. Our approach is based on 3D bounding boxes built around the vehicles. The bounding box can be automatically constructed from traffic surveillance data. For scenarios where it is not possible to use the precise construction, we propose a method for estimation of the 3D bounding box. The 3D bounding box is used to normalize the image viewpoint by "unpacking" the image into plane. We also propose to randomly alter the color of the image and add a rectangle with random noise to random position in the image during training Convolutional Neural Networks. We have collected a large fine-grained vehicle dataset BoxCars116k, with 116k images of vehicles from various viewpoints taken by numerous surveillance cameras. We performed a number of experiments which show that our proposed method significantly improves CNN classification accuracy (the accuracy is increased by up to 12 percent points and the error is reduced by up to 50 % compared to CNNs without the proposed modifications). We also show that our method outperforms state-of-the-art methods for fine-grained recognition.**

I. Introduction

Fine-grained recognition of vehicles is interesting both from the application point of view (surveillance, data retrieval, etc.) and from the point of view of research of general fine-grained recognition applicable in other fields. For example, Gebru et al. [1] proposed estimation of demographic statistics based on fine-grained recognition of vehicles. In this article, we are presenting methodology which considerably increases the performance of multiple state-of-the-art CNN architectures in the task of fine-grained vehicle recognition. We target the traffic surveillance context, namely images of vehicles taken from an **arbitrary viewpoint** – we do not limit ourselves to frontal/rear viewpoints. As the images are obtained from surveillance cameras, they have challenging properties – they are often small and taken from very general viewpoints (high elevation). Also, we construct the training and testing sets from images from different cameras as it is common for surveillance applications that it is not known a priori under which viewpoint the camera will be observing the road.

Authors are with Brno University of Technology, Faculty of Information Technology, Centre of Excellence IT4Innovations , Czech Republic {isochor,ispanhel,herout}@fit.vutbr.cz

Jakub Sochor is a Brno Ph.D. Talent Scholarship Holder — Funded by the Brno City Municipality.

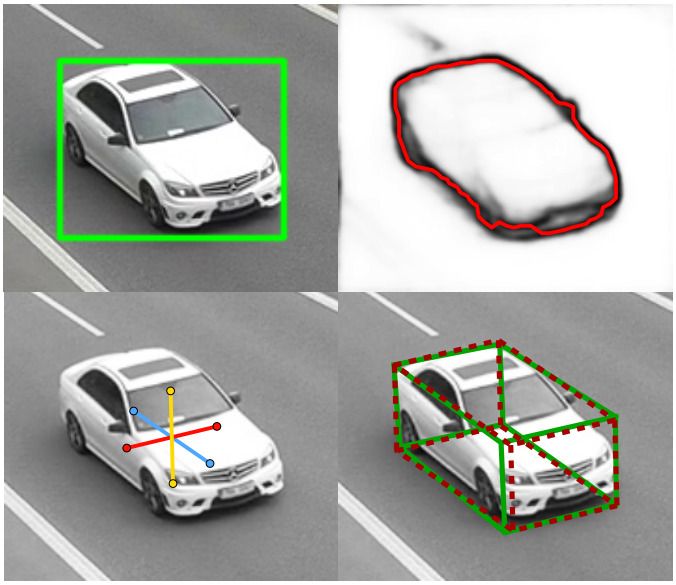

Fig. 1. Example of automatically obtained 3D bounding box used for fine-grained vehicle classification. **Top left:** vehicle with 2D bounding box annotation, **top right:** estimated contour, **bottom left:** estimated directions to vanishing points, **bottom right:** 3D bounding box automatically obtained from surveillance video (green) and our estimated 3D bounding box (red).

Methods focused on fine-grained recognition of vehicles usually have some limitations – they can be limited to frontal/rear viewpoint or use 3D CAD models of all the vehicles. Both these limitations are rather impractical for large scale deployment. There are also methods for fine-grained recognition in general which were applied on vehicles. The methods recently follow several main directions – automatic discovery of parts [2], [3], bilinear pooling [4], [5], or exploiting structure of fine-grained labels [6], [7]. Our method is not limited to any particular viewpoint and it does not require 3D models of vehicles at all.

We propose an orthogonal approach to these methods and use CNNs with modified input to achieve better image normalization and data augmentation (therefore, our approach can be combined with other methods). We use 3D bounding boxes around vehicles to normalize vehicle image, see Figure 3 for examples. This work is based on our previous conference paper [8]; it pushes the performance further and mainly we propose a new method how to build the 3D bounding box without any prior knowledge, see Figure 1. Our input modifications are able to significantly increase the classification accuracy (up

to **12 percent points**, classification error is reduced by up to **50 %**).

The key contributions of the paper are:

- Complex and thorough evaluation of our previous method [8].
- Our novel data augmentation techniques further improve the results of the fine-grained recognition of vehicles relative both to our previous method and other state-of-the-art methods (Section III-C).
- We remove the requirement of the previous method [8] to know the 3D bounding box by estimating the bounding box both at training and test time (Section III-D).
- We collected more samples to the BoxCars dataset, increasing the dataset size almost twice, see Section IV.

We make the collected dataset and source codes for the proposed algorithm publicly available[1] for future reference and comparison.

II. Related Work

To provide context to the proposed method, we present a summary of existing fine-grained recognition methods (both general and focused on vehicles).

A. General Fine-Grained Object Recognition

We divide the fine-grained recognition methods from recent literature into several categories as they usually share some common traits. Methods exploiting annotated model parts (e.g [9], [10]) are not discussed in detail as it is not common in fine-grained datasets of vehicles to have the parts annotated.

1) Automatic Part Discovery: Parts of classified objects may be discriminatory and provide lots of information for the fine-grained classification task. However, it is not practical to assume that the location of such parts is known a priori as it requires significantly more annotation work. Therefore, several papers [2], [3], [11]–[15] have dealt with this problem and proposed methods how to automatically (during both training and test time) discover and localize such parts. The methods differ mainly in the way which is used for the discovery of discriminative parts. The features extracted from the parts are usually classified by SVMs.

2) Methods using Bilinear Pooling: Lin et al. [4] use only convolutional layers from the net for extraction of features which are classified by bilinear classifier [16]. Gao et al. [5] followed the path of bilinear pooling and proposed a method for Compact Bilinear Pooling getting the same accuracy as the full bilinear pooling with a significantly lower number of features.

3) Other Methods: Xie et al. [6] proposed to use hyper-class for data augmentation and regularization of fine-grained deep learning. Zhou et al. [7] use CNN with Bipartite Graph Labeling to achieve better accuracy by exploiting the fine-grained annotations and coarse body type (e.g. Sedan, SUV). Lin et al. [17] use three neural networks for simultaneous localization, alignment and classification of images. Each of these three networks does one of the three tasks and they are connected into one bigger network. Yao et al. [13] proposed an approach which is using responses to random templates obtained from images and classify merged representations of the response maps by SVM. Zhang et al. [18] use pose normalization kernels and their responses warped into a feature vector. Chai et al. [19] propose to use segmentation for fine-grained recognition to obtain foreground parts of image. A similar approach was also proposed by Li et al. [20]; however, the authors use a segmentation algorithm which is optimized and fine-tuned for the purpose of fine-grained recognition. Finally, Gavves et al. [21] propose to use object proposals to obtain the foreground mask and unsupervised alignment to improve fine-grained classification accuracy.

B. Fine-Grained Recognition of Vehicles

The goal of fine-grained recognition of vehicles is to identify the exact type of the vehicle, that is its make, model, submodel, and model year. The recognition system focused only on vehicles (in relation to general fine-grained classification of birds, dogs, etc.) can benefit from that the vehicles are rigid, have some distinguishable landmarks (e.g. license plates), and rigorous models (e.g. 3D CAD models) can be available.

1) Methods Limited to Frontal/Rear Images of Vehicles: There is a multitude of papers [22]–[29] using a common approach: they detect the license plate (as a common landmark) on the vehicle and extract features from the area around the license plate as the front/rear parts of vehicles are usually discriminative.

There are also papers [30]–[35] directly extracting features from frontal images of vehicles by different methods and optionally exploiting standard structure of parts on the frontal mask of car (e.g. headlights).

2) Methods based on 3D CAD Models: There were several approaches how to deal with viewpoint variance using synthetic 3D models of vehicles. Lin et al. [36] propose to jointly optimize 3D model fitting and fine-grained classification, Hsiao et al. [37] use detected contour and align the 3D model using 3D chamfer matching. Krause et al. [38] propose to use synthetic data to train geometry and viewpoint classifiers for 3D model and 2D image alignment. Prokaj et al. [39] propose to detect SIFT features on the vehicle image and on every 3D model seen from a set of discretized viewpoints.

3) Other Methods: Gu et al. [40] propose to extract center of vehicle and roughly estimate the viewpoint from the bounding box aspect ratio. Then, they use different Active Shape Models for alignment of data taken from different viewpoints and use segmentation for background removal.

Stark et al. [41] propose to use an extension of Deformable Parts Model (DPM) [42] to be able to handle multi-class recognition. The model is represented by latent linear multi-class SVM with HOG [43] features. The authors show that the system outperforms different methods based on Locally-constrained Linear Coding [44] and HOG. The recognized vehicles are used for eye-level camera calibration.

Liu et al. [45] use deep relative distance trained on vehicle re-identification task and propose to train the neural net with

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

Coupled Clusters Loss instead of triplet loss. Boonsim et al. [46] propose a method for fine-grained recognition of vehicles at night. The authors use relative position and shape of features visible at night (e.g. lights, license plates) to identify the make&model of a vehicle, which is visible from the rear side.

Fang et al. [47] propose to use an approach based on detected parts. The parts are obtained in an unsupervised manner as high activations in a mean response across channels of the last convolutional layer of used CNN. [48] introduce spatially weighted pooling of convolutional features in CNNs to extract important features from the image.

4) Summary of Existing Methods: Existing methods for fine-grained classification of vehicles usually have significant limitations. They are either limited to frontal/rear viewpoints [22]–[35] or they require some knowledge about 3D models of the vehicles [36]–[39] which can be impractical when new models of vehicles emerge.

Our proposed method does not have such limitations. The method works with arbitrary viewpoints and we require only 3D bounding boxes of vehicles. The 3D bounding boxes can be either automatically constructed from traffic video surveillance data [49], [50] or we propose a method how to estimate the 3D bounding boxes both at training and test time from single images (see Section III-D).

C. Datasets for Fine-Grained Recognition of Vehicles

There is a large number of datasets of vehicles (e.g [51], [52]) which are usable mainly for vehicle detection, pose estimation, and other tasks. However, these datasets do not contain annotations of the precise vehicles' make & model.

When it comes to the fine-grained recognition datasets, there are some [33], [36], [38], [41] which are relatively small in number of samples or classes. Therefore, they are impractical for training of CNN and deployment of real world traffic surveillance applications.

Yang et al. [53] published a large dataset *CompCars*. 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 with 50k frontal images of vehicles taken from surveillance cameras.

Liu et al. [54] published dataset *VeRi-776* for the vehicle re-identification task. The dataset contains over 50k images of 776 vehicles captured by 20 cameras covering an 1.0 km^2 area in 24 hours. Each vehicle is captured by $2 \sim 18$ cameras under different viewpoints, illuminations, resolutions and occlusions. The dataset also provides various attributes, such as bounding boxes, vehicle types, and colors.

III. Proposed Methodology for Fine-Grained Recognition of Vehicles

In agreement with the recent progress in the Convolutional Neural Networks [55]–[57], we use CNN for both classification and verification (determining whether a pair of vehicles has the same type). However, we propose to use several data normalization and augmentation techniques to significantly boost the classification performance (up to 50 % error reduction compared to base net). We utilize information about 3D bounding boxes obtained from traffic surveillance camera [49]. Finally, to increase the applicability of our method to scenarios where the 3D bounding box is not known, we propose an algorithm for bounding box estimation both at training and test time.

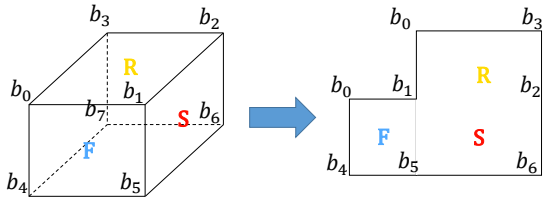

Fig. 2. 3D bounding box and its unpacked version.

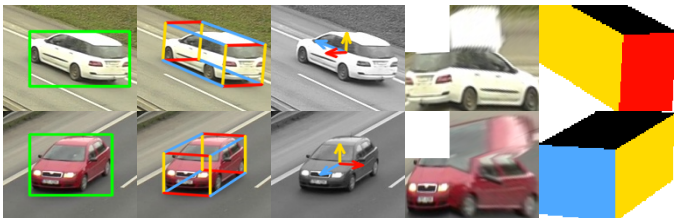

Fig. 3. Examples of data normalization and auxiliary data feeded to nets. **Left to right:** vehicle with 2D bounding box, computed 3D bounding box, vectors encoding viewpoints on the vehicle (**View**), unpacked image of the vehicle (**Unpack**), and rasterized 3D bounding box feeded to the net (**Rast**).

A. Image Normalization by Unpacking the 3D Bounding Box

We based our work on 3D bounding boxes proposed by [49] (Fig. 3) which can be automatically obtained for each vehicle seen by a surveillance camera (see the original paper by [49] for further details). These boxes allow us to identify side, roof, and front (or rear) side of vehicles in addition to other information about the vehicles. We use these localized segments to normalize the image of the observed vehicles (considerably boosting the recognition performance).

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. 2) and they are organized into the final matrix $\mathbf{U}$:

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

The unpacking itself is done by obtaining homography between points b_i (Fig. 2) and perspective warping parts of the original image. The left top submatrix is filled with zeros. This unpacked version of the vehicle is 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. In the further text, we will refer to this normalization method as **Unpack**.

B. Extended Input to the Neural Nets

It it possible to infer additional information about the vehicle from the 3D bounding box and we found out that these

Fig. 4. Examples of proposed data augmentation techniques. Left most image contains the original cropped image of the vehicle and other images contains augmented versions of the image (**Top – Color**, **Bottom – ImageDrop**).

data slightly improve the classification and verification performance. One piece of this auxiliary information is the encoded viewpoint (direction from which the vehicle is observed). We also add rasterized 3D bounding box as an additional input to the CNNs. Compared to our previously proposed auxiliary data fed to the net [8], we handle frontal and rear vehicle side differently.

View. The viewpoint is extracted from the orientation of the 3D bounding box – Fig. 3. 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 $v_i = \overrightarrow{C_c C_i}$. Point C_c is the center of the bounding box and it can be obtained as the intersection of diagonals $\overleftrightarrow{b_2 b_4}$ and $\overleftrightarrow{b_5 b_3}$. Points C_i for $i \in \{f, s, r\}$ denote the centers of each face, again computed as intersections of face diagonals. In contrast to our previous approach [8], which did not take the direction of the vehicle into account; instead, we encode information about the vehicle direction ($d = 1$ for vehicles going to camera, $d = 0$ for vehicles going from the camera), to determine which side of the bounding box is the frontal one. 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.

Rast. 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 four-channel image containing each visible face rasterized in a different channel. Formally, point p of the rasterized bounding box $\mathbf{T}$ is obtained as

$$\mathbf{T}_p = \begin{cases} (1,0,0,0) & p \in \measuredangle b_0 b_1 b_4 b_5 \text{ and d = 1} \\ (0,1,0,0) & p \in \measuredangle b_0 b_1 b_4 b_5 \text{ and d = 0} \\ (0,0,1,0) & p \in \measuredangle b_1 b_2 b_5 b_6 \\ (0,0,0,1) & p \in \measuredangle b_0 b_1 b_2 b_3 \\ (0,0,0,0) & \text{otherwise} \end{cases} \quad (2)$$

where $\measuredangle b_0 b_1 b_4 b_5$ denotes the quadrilateral defined by points b_0, b_1, b_4 and b_5 in Figure 2.

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

C. Additional Training Data Augmentation

To increase the diversity of the training data, we propose additional data augmentation techniques. The first one (denoted as **Color**) deals with the fact that for fine-grained recognition of vehicles (and some other objects), the color is irrelevant. The other method (**ImageDrop**) deals with some potentially missing parts on the vehicle. Examples of the data augmentation are shown in Figure 4. Both these augmentation techniques are done only with predefined probability during training, otherwise they are not modified. During testing, we do not modify the images at all.

The results presented in Section V-E show that both these modifications improve the classification accuracy both in combination with other presented techniques or by themselves.

Color. To increase training samples color variability, we propose to randomly alternate the color of the image. The alternation is done in HSV color space by adding the same random values to each pixel in the image (each HSV channel is processed separately).

ImageDrop. Inspired by Zeiler et al. [58] who evaluated the influence of covering a part of the input image on the probability of the ground truth class, we take this step further and in order to deal with missing parts on the vehicles, we take a random rectangle in the image and fill it with random noise, effectively dropping any information contained in that part of image.

D. Estimation of 3D Bounding Box from a Single Image

As the results (Section V) show, the most important part of the proposed algorithm is **Unpack** followed by **Color** and **ImageDrop**. However, the 3D bounding box is required for the unpacking of the vehicles and we acknowledge that there may be scenarios when such information is not available. For these cases, we propose a method how to estimate the 3D bounding box for both training and test time with only limited information available.

As proposed by [49], the vehicle's contour and the vanishing points are required for the bounding box construction. Therefore, it is necessary to estimate the contour and the vanishing points for the vehicle. For estimating the vehicle contour, we use Fully Convolutional Encoder-Decoder network designed by Yang et al. [59] for general object contour detection and masks with probabilities of vehicles contours for each image pixel. To obtain the final contour, we search for global maxima along line segments from 2D bounding box centers to edge points of the 2D bounding box. For examples, see Figure 5.

We found out that the exact position of the vanishing point is not required for the 3D bounding box construction, but the directions to the vanishing points are much more important. Therefore, we use regression to obtain the directions towards the vanishing points and then assume that the vanishing points are in infinity.

Following the work by Rothe et al. [60], we formulated the regression of the direction towards the vanishing points as a classification task into bins corresponding to angles and we use ResNet50 [61] with three classification outputs. We found this approach as more robust than a direct regression.

Fig. 5. Estimation of 3D bounding box. **Left to right:** image with vehicle 2D bounding box, output of contour object detector [59], our constructed contour, estimated directions towards vanishing points, ground truth (**green**) and estimated (**red**) 3D bounding box.

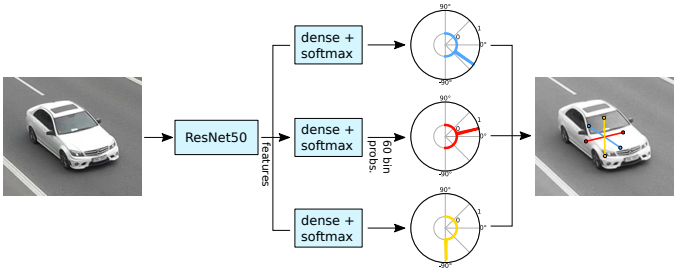

Fig. 6. Used CNN for estimation of directions towards vanishing points. The vehicle image is fed to ResNet50 with 3 separate outputs which predict probabilities for directions of vanishing points as probabilities in a quantized angle space (60 bins from $-90°$ to $90°$).

We added three separate fully connected layers with softmax activation (one for each vanishing point) after the last average pooling in the ResNet50 (see Figure 6). Each of these layers generates probabilities for each vanishing point belonging to specific direction bin (represented as angles). We quantized the angle space by bins of $3°$ from $-90°$ to $90°$ (60 bins per vanishing point in total).

As the training data for the regression we used BoxCars116k dataset (Section IV) with the test samples omitted. To construct the lines on which the vanishing points are, we use the center of the 2D bounding box.

With all these estimated information it is then possible to construct the 3D bounding box in both training and test time. It is important to note that using this 3D bounding box estimation, it is possible to use this method outside the scope of traffic surveillance. It is only necessary to train the regressor of vanishing points directions. For training of such regressor, it is possible to use either the directions themselves or viewpoints on the vehicle and focal lengths of the images.

Using this estimated bounding box, it is possible to unpack the vehicle image in test time without any additional information required and enabling the usage of the method when the traffic surveillance data are not available. The results in Section V-C show that using this estimated 3D bounding boxes, our method still significantly outperforms other convolutional neural networks without input modification.

IV. BoxCars116k Dataset

We collected and annotated a new dataset *BoxCars116k*. 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. The cameras were placed on various locations around Brno, Czech Republic and recorded the passing traffic from an arbitrary (reasonable) surveillance viewpoint. Each correctly detected vehicle (by Faster-RCNN [62] trained on COD20k dataset [63]) is captured in multiple images, as it is passing by the camera; therefore, we have more visual information about each vehicle.

A. Dataset Acquisition

The dataset is formed by two parts. The first part consists of data from *BoxCars21k* dataset [8] which were cleaned up and some imprecise annotations were corrected (e.g. missing model years for some uncommon vehicle types).

We also collected other data from videos relevant to our previous work [49], [50], [64]. We detected all vehicles, tracked them and for each track collected images of the respective vehicle. We downsampled the framerate to ~ 12.5 FPS to avoid collection of multiple almost identical images of the same vehicle.

The new dataset was annotated by multiple human annotators with interest in vehicles and sufficient knowledge about vehicle types and models. The annotators were assigned to clean up the processed data from invalid detections and assign exact vehicle type (make, model, submodel, year) for each obtained track. While preparing the dataset for annotation, 3D bounding boxes were constructed for each detected vehicle using the method proposed by [49]. Invalid detections were then distinguished by the annotators based on these constructed 3D bounding boxes. In the cases when all 3D bounding boxes were not constructed precisely, the whole track was invalidated.

Vehicle type annotation reliability is guaranteed by providing multiple annotations for each valid track (~ 4 annotations per vehicle). The annotation of vehicle type is considered as correct in the case of at least three identical annotations. Uncertain cases were authoritatively annotated by the authors.

The tracks in *BoxCars21k* dataset consist of exactly 3 images per track. In the new part of the dataset, we collect an arbitrary number of images per track (usually more then 3).

B. Dataset Statistics

The dataset contains 27 496 vehicles (116 286 images) of 45 different makes with 693 fine-grained classes (make & model & submodel & model year) collected from 137 different cameras with a large variation in the viewpoints. Detailed statistics about the dataset can be found in in Figure 8 and the supplementary material. The distribution of types in the dataset is shown in Figure 8 (**top right**) and samples from the dataset are in Figure 7. The dataset also includes information about the 3D bounding box [49] for each vehicle and an image with a foreground mask extracted by background subtraction [65], [66]. The dataset is made publicly available[2] for future reference and evaluation.

Compared to "web-based" datasets, the new *BoxCars116k* dataset contains images of vehicles relevant to traffic surveillance which have specific viewpoints (high elevation), usually

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

Fig. 7. Collate of random samples from the BoxCars116k dataset.

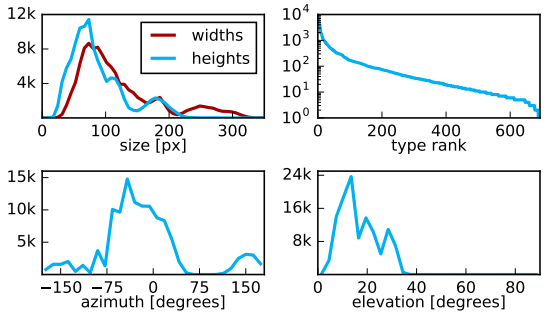

Fig. 8. BoxCars116k dataset statistics – **top left:** 2D bounding box dimensions, **top right:** number of fine-grained types samples, **bottom left:** azimuth distribution (0° denotes frontal viewpoint), **bottom right:** elevation distribution.

small images etc. Compared to other fine-grained surveillance datasets, our dataset provides data with a high variation in viewpoints, see Figure 8 and 3D plots in the supplementary material.

C. Training & Test Splits

Our task is to provide a dataset for fine-grained recognition in traffic surveillance without any viewpoint constraint. Therefore, we construct the splits for training and evaluation in a way which reflects the fact that it is usually not known a priori from which viewpoints the vehicles will be seen by the surveillance camera.

Thus, for the construction of the splits, we randomly selected cameras and used all tracks from these cameras for training and vehicles from the rest of the cameras for testing. This way, we are testing the classification algorithms on images of vehicles from previously unseen cameras (viewpoints). This training & test splits selection process causes that some of the vehicles from the testing set may be taken under slightly different viewpoint the are present in the training set.

We constructed two splits. In the first one (**hard**), we are interested in recognizing the precise type including model year. In the other one (**medium**), we omit the difference in model years and all vehicles of the same subtype (and potentially different model years) are present in the same class. We selected only types which have at least 15 tracks in the training set and at least one track in the testing set. The hard split contains 107 fine-grained classes with 11 653 tracks (51 691 images) for training and 11 125 tracks (39 149 images) for testing. Detailed split statistics can be found in the supplementary material.

V. Experiments

We thoroughly evaluated our proposed algorithm on the BoxCars116k dataset. First, we evaluated how these methods improve for different nets, compared them to the state of the art, and analyzed how using approximate 3D bounding boxes influence the achieved accuracy. Then, we searched for the main source of improvements, analyzed improvements of different modifications separately, and we also evaluated the usability of features from the trained nets for the task of vehicle type identity verification.

To show that our modifications improve the accuracy independently on the used nets, we use several of them:

- **AlexNet** [56]
- **VGG16**, **VGG19** [67]
- **ResNet50**, **ResNet101**, **ResNet152** [61]
- CNNs with Compact Bilinear Pooling layer [5] in combination with VGG nets denoted as **VGG16+CBL** and **VGG19+CBL**.

As there are several options how to use the proposed modifications of input data and add additional auxiliary data, we define several labels which we will use:

- **ALL** – All five proposed modifications (Unpack, Color, ImageDrop, View, Rast).
- **IMAGE** – Modifications working only on the image level (Unpack, Color, ImageDrop).
- **CVPR16** – Modifications as proposed in our previous CVPR paper [8] (Unpack, View, Rast – however, for the View and Rast modifications differ from those ones used in this paper as the original modifications do not distinguish between the frontal and the rear side of vehicles).

A. Improvements for Different CNNs

The first experiment which we have done is evaluation how our modifications improve classification accuracy for different CNNs.

All the nets were fine-tuned from models pre-trained on ImageNet [51] for approximately 15 epochs which was sufficient for the nets to converge. We used the same batch size (except for ResNet151, where we had to use smaller batch size because of GPU memory limitations), the same initial learning rate and learning rate decay and the same hyperparameters for every net (initial learning rate $2.5 \cdot 10^{-3}$, weight decay $5 \cdot 10^{-4}$, quadratic learning rate decay, loss is averaged over 100 iterations). We also used standard data augmentation techniques as horizontal flip and randomly moving bounding box [67]. As ResNets do not use fully connected layers, we only use **IMAGE** modifications for them.

TABLE I
SUMMARY STATISTICS OF IMPROVEMENTS BY OUR PROPOSED MODIFICATIONS FOR DIFFERENT CNNS. THE IMPROVEMENTS OVER BASELINE CNNS ARE REPORTED AS SINGLE SAMPLE ACCURACY/TRACK ACCURACY IN PERCENT POINTS. WE ALSO PRESENT CLASSIFICATION ERROR REDUCTION IN THE SAME FORMAT. THE RAW NUMBERS CAN BE FOUND IN THE SUPPLEMENTARY MATERIAL.

	modif.	improvement [pp]		error reduction [%]	
		mean	best	mean	best
medium	ALL	7.49/6.29	11.84/10.99	26.83/34.50	36.71/50.32
	IMAGE	7.19/6.15	12.09/11.63	27.38/36.21	35.23/49.55
	CVPR16	2.99/3.18	5.22/5.65	10.86/17.71	19.76/32.25
hard	ALL	7.00/5.83	11.14/10.85	25.59/33.52	33.40/48.76
	IMAGE	6.74/5.81	11.02/10.53	26.12/35.95	33.04/47.33
	CVPR16	2.12/2.44	3.56/3.92	7.93/14.57	12.68/24.10

For each net and each modification we evaluate the accuracy improvement of the modification in percents points and we also evaluate the classification error reduction.

The summary results for both medium and hard splits are shown in Table I and the raw results are in the supplementary material. As we have correspondences between the samples in the dataset and we know which samples are from the same track, we are able to use mean probability across track samples and merge the classification for the whole track. Therefore, we always report the results in the form *single sample accuracy/whole track accuracy*. As expected, the results for whole tracks are much better than for single samples.

There are several things which should be noted about the results. The most important one is that our modifications significantly improve classification accuracy (up to **+12 percent points**) and reduce classification error (up to **50 % error reduction**). Another important fact is that our new modifications push the accuracy much further compared to the original method [8].

The table also shows that the difference between **ALL** modifications and **IMAGE** modifications is negligible and therefore it is reasonable to only use the **IMAGE** modifications. This also results into CNNs which use just the **Unpack** modification during test time as the other image modifications (Color, ImageDrop) are used only during fine-tuning of CNNs.

Also, the evaluation shows that the results are almost identical for the hard and medium split; therefore, we will further only report results on the hard split, as it is the main goal to distinguish also the model years. The names for the splits were chosen to be consistent with the original version of dataset [8] and the small difference between medium and hard split accuracies is caused mainly by the size of the new dataset.

B. Comparison with the State of the Art

In order to examine the performance of our method, we also evaluated other state-of-the-art methods for fine-grained recognition. We used 3 different algorithms for general fine-grained recognition with published code. We always first used the code to reproduce the results in respective papers to ensure that we are using the published work correctly. All of the methods use CNNs and the used net influences the accuracy, therefore the results should be compared with respective base CNNs.

TABLE II
COMPARISON OF DIFFERENT VEHICLE FINE-GRAINED RECOGNITION METHODS. ACCURACY IS REPORTED AS SINGLE IMAGE ACCURACY/WHOLE TRACK ACCURACY. PROCESSING SPEED WAS MEASURED ON A MACHINE WITH GTX1080 AND CUDNN. * FPS REPORTED BY AUTHORS.

method	accuracy [%]	speed [FPS]
AlexNet [56]	66.65/77.75	963
VGG16 [67]	77.26/86.71	173
VGG19 [67]	76.74/86.06	146
Resnet50 [61]	75.48/84.61	155
Resnet101 [61]	76.46/85.31	95
Resnet152 [61]	77.68/86.20	66
BCNN (VGG-M) [4]	64.83/72.22	87*
BCNN (VGG16) [4]	69.64/78.56	10*
CBL (VGG16) [5]	70.38/80.11	165
CBL (VGG19) [5]	70.69/80.26	141
PCM (AlexNet) [3]	63.24/73.94	15
PCM (VGG19) [3]	75.99/85.24	4
AlexNet + ALL (ours)	77.79/88.60	580
VGG16 + ALL (ours)	**84.13/92.27**	154
VGG19 + ALL (ours)	**84.12**/92.00	133
VGG16+CBL + ALL (ours)	75.06/83.42	146
VGG19+CBL + ALL (ours)	75.62/83.76	126
Resnet50 + IMAGE (ours)	82.27/90.79	151
Resnet101 + IMAGE (ours)	83.41/91.59	93
Resnet152 + IMAGE (ours)	83.74/91.71	65

It was impossible to evaluate methods focused only on fine-grained recognition of vehicles as they are usually limited to frontal/rear viewpoint or require 3D models of vehicles for all the types. In the following text we define labels for each evaluated state-of-the-art method and describe details for the method separately.

BCNN. Lin et al. [4] proposed to use **B**ilinear **CNN**. We used VGG-M and VGG16 networks in a symmetric setup (details in the original paper), and trained the nets for 30 epochs (the nets converged around the 20^{th} epoch). We also used image flipping to augment the training set.

CBL. We modified compatible nets with **C**ompact **Bi**Linear Pooling proposed by [5] which followed the work of [4] and reduced the number of output features of the bilinear layers. We used the Caffe implementation of the layer provided by the authors and used 8 192 features. We trained the net using the same hyper-parameters, protocol, and data augmentation as described in Section V-A.

PCM. Simon et al. [3] propose **P**art **C**onstellation **M**odels and use neural activations (see the paper for the details) to get the parts of the model. We used AlexNet (BVLC Caffe reference version) and VGG19 as base nets for the method. We used the same hyper-parameters as the authors with the exception of fine-tuning number of iterations which was increased, and the C parameter of used linear SVM was cross-validated on the training data.

The results of all the comparisons can be found in Table II. As the table shows, our method significantly outperforms both standard CNNs [56], [61], [67] and methods for fine-grained recognition [3]–[5]. The results for fine-grained recognition methods should be compared with the same used base network

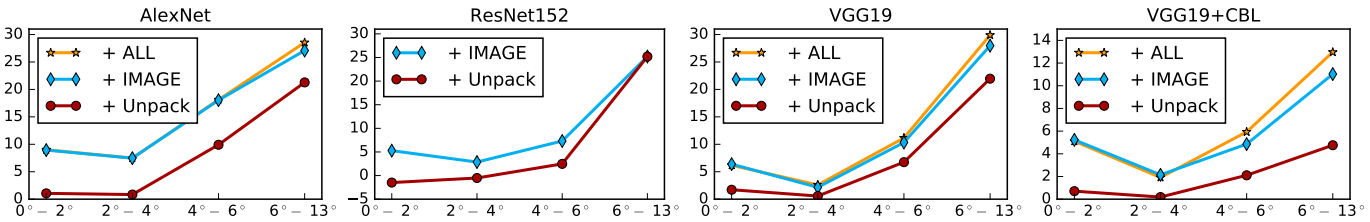

Fig. 9. Correlation of improvement relative to CNNs without modification with respect to train-test viewpoint difference. The x-axis contains bins viewpoint difference bins (in degrees), and the y-axis denotes improvement compared to base net in percent points, see Section V-D for details. The graphs show that with increasing viewpoint difference, the accuracy improvement of our method increases. Only one representative of each CNN family (AlexNet, VGG, ResNet, VGG+CBL) is displayed – results for all CNNs are in the supplementary material.

TABLE III
COMPARISON OF CLASSIFICATION ACCURACY (PERCENT) ON THE HARD SPLIT WITH STANDARD NETS WITHOUT ANY MODIFICATIONS, IMAGE MODIFICATIONS USING 3D BOUNDING BOX FROM SURVEILLANCE DATA, AND IMAGE MODIFICATIONS USING ESTIMATED 3D BB (SECTION III-D).

net	no modification	GT 3D BB	estimated 3D BB
AlexNet	66.65/77.75	77.67/88.28	74.81/87.30
VGG16	77.26/86.71	83.79/92.23	80.60/90.59
VGG19	76.74/86.06	83.91/92.17	81.43/91.57
VGG16+CBL	70.38/80.11	75.04/83.16	72.83/82.92
VGG19+CBL	70.69/80.26	75.47/83.56	73.09/83.09
ResNet50	75.48/84.61	82.27/90.79	79.60/90.40
ResNet101	76.46/85.31	83.41/91.59	80.20/90.42
ResNet152	77.68/86.20	83.74/91.71	80.87/90.93

as for different networks, they provide different results. Our best accuracy (84 %) is better by a large margin compared to all other variants (both standard CNN and fine-grained methods).

In order to provide approximate information about the processing efficiency, we measured how many images of vehicles are different methods and networks able to process per second (referenced as FPS). The measurement was done with GTX1080 and CUDNN whenever possible. In the case of BCNN we report the numbers reported by the authors, as we were forced to save some intermediate data to disk because we were not able to fit all the data to memory (∼200 GB). The results are also shown in Table II; they show that our input modification decreased the processing speed; however, the speed penalty is small and the method is still well usable for real-time processing.

C. Influence of Using Estimated 3D Bounding Boxes instead of the Surveillance Ones

We also evaluated how the results will be influenced when instead of using the 3D bounding boxes obtained from the surveillance data (long-time observation of video [49], [50]), the estimated 3D bounding boxes (Section III-D) would be used.

The classification results are shown in Table III; they show that the proposed modifications still significantly improve the accuracy even if only the estimated 3D bounding box – the less accurate one – is used. This result is fairly important, as it enables to transfer this method to different (non-surveillance) scenarios. The only additional data which is then required is a reliable training set of directions towards the vanishing points (or viewpoints and focal length) from the vehicles (or other rigid objects).

Fig. 10. Each pair has test sample on the left side, and sample from the training set with the lowest angle between its viewpoint and the test sample viewpoint on the right side.

D. Impact of Training/Testing Viewpoint Difference

We were also interested what is the main reason that the classification accuracy improved. We have analyzed several possibilities and found out that the most important aspect is viewpoint difference.

For every training and testing sample we computed the viewpoint (unit 3D vector from vehicles' 3D bounding boxes centers) and for each testing sample we found one training sample with the lowest viewpoint difference (see Figure 10). Then, we divided the testing samples into several bins based on the difference angle. For each of these bins we computed the accuracy for the standard nets without any modifications and nets with the proposed modifications. Finally, we obtained the improvement in percent points for each modification and each bin, by comparing the net's performance on the data in the bin with and without the modification harnessed. The results are displayed in Figure 9.

There are several facts which should be noted. The first and the most important is that the **Unpack** modification alone improves the accuracy significantly for larger viewpoint differences (the accuracy is improved by more then 20 percent points for the last bin). The other important fact which should be noted is that the other modifications (mainly **Color** and **ImageDrop**) push the accuracy furthermore independently on the training-testing viewpoint difference.

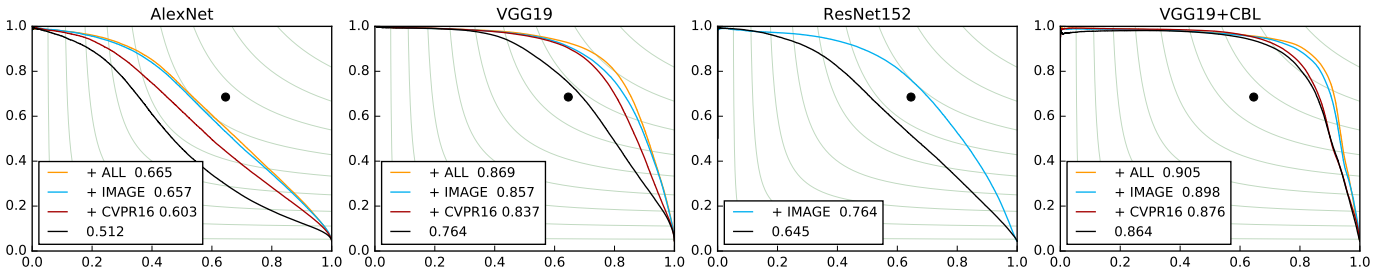

Fig. 11. Precision-Recall curves for verification of fine-grained types. Black dots represent the human performance [8]. Only one representative of each CNN family (AlexNet, VGG, ResNet, VGG+CBL) is displayed – results for all CNNs are in the supplementary material.

TABLE IV
SUMMARY OF IMPROVEMENTS FOR DIFFERENT NETS AND MODIFICATIONS COMPUTED AS $[\textit{base net} + \textit{modification}] - [\textit{base net}]$. THE RAW DATA CAN BE FOUND IN THE SUPPLEMENTARY MATERIAL.

	mean	**best**
Unpack	+2.11/+2.55	+3.47/+4.37
View	−0.32/−0.35	+0.19/+0.31
Rast	−0.03/−0.04	+0.30/+0.72
Color	+3.17/+2.03	+4.80/+3.60
ImageDrop	+0.70/+0.20	+1.53/+0.96

TABLE V
SUMMARY OF IMPROVEMENTS FOR DIFFERENT NETS AND MODIFICATIONS COMPUTED AS $[\textit{base net} + \textit{all}] - [\textit{base net} + \textit{all} - \textit{modification}]$. THE RAW DATA CAN BE FOUND IN THE SUPPLEMENTARY MATERIAL.

	mean	**best**
Unpack	+3.41/+3.48	+6.93/+7.60
View	−0.14/−0.15	+0.36/+0.18
Rast	−0.03/−0.08	+0.30/+0.20
Color	+3.42/+2.43	+6.34/+6.18
ImageDrop	+1.32/+0.77	+4.24/+3.54

E. Impact of Individual Modifications

We were also curious how different modifications by themselves help to improve the accuracy. We conducted two types of experiments, which focus on different aspects of the modifications. The evaluation is not done on ResNets, as we only use **IMAGE** level modifications with ResNets; thus, we can not evaluate Rast and View modifications with ResNets.

The first experiment is focused on the influence of each modification by itself. Therefore, we compute the accuracy improvement (in accuracy percent points) for the modifications as $[\textit{base net} + \textit{modification}] - [\textit{base net}]$, where $[\ldots]$ stands for the accuracy of the classifier described by its contents. The results are shown in Table IV. As it can be seen in the table, the most contributing modifications are **Color**, **Unpack**, and **ImageDrop**.

The second experiment evaluates how a given modification contributed to the accuracy improvement when all of the modifications are used. Thus, the improvement is computed as $[\textit{base net} + \textit{all}] - [\textit{base net} + \textit{all} - \textit{modification}]$. See Table V for the results, which confirm the previous findings and **Color**, **Unpack**, and **ImageDrop** are again the most positive modifications.

F. Vehicle Type Verification

Lastly, we evaluated the quality of features extracted from the last layer of the convolutional nets for the verification task. Under the term *verification*, we understand the task to determine whether a pair of vehicle tracks share the same fine-grained type or not. In agreement with previous works in the field [55], we use cosine distance between the features for the verification.

We collected 5 millions of random pairs of vehicle tracks from test part of *BoxCars116k* splits and evaluate the verification on these pairs. As we used tracks which can have a different number of vehicle images, we use 9 random pairs of images for each pair of tracks and used median distance between these image pairs as the distance between the whole tracks.

Precision-Recall curves and Average Precisions are shown in Figure 11. As the results show, our modifications significantly improve the average precision for each CNN in the given task. Also, as the figure shows, the method outperforms human performance (black dots in Figure 11), as reported in the previous paper [8].

VI. CONCLUSION

This article presents and sums up multiple algorithmic modifications suitable for CNN-based fine-grained recognition of vehicles. Some of the modifications were originally proposed in a conference paper [8], some are results of the ongoing research. We also propose a method for obtaining the 3D bounding boxes necessary for the image unpacking (which has the largest impact on the performance improvement) without observing a surveillance video, but only working with the individual input image. This considerably increases the application potential of the proposed methodology (and the performance for such estimated 3D bboxes is only somewhat lower than when the "proper" bounding boxes are used). We focused on thorough evaluation of the methods: we couple them with multiple state-of-the-art CNN architectures [61], [67], we measure the contribution/influence of individual modifications.

Our method significantly improves the classification accuracy (up to **+12 percent points**) and reduces the classification error (up to **50 % error reduction**) compared to the base CNNs. Also, our method outperforms other state-of-the-art methods [3]–[5] by **9 percent points** in single image accuracy and by **7 percent points** in whole track accuracy.

We collected, processed, and annotated a dataset *BoxCars116k* targeted to fine-grained recognition of vehicles in the surveillance domain. Contrary to majority of existing vehicle recognition datasets, the viewpoints are greatly varying and they correspond to surveillance scenarios; the existing datasets are mostly collected from web images and the vehicles are typically captured from eye-level positions. This dataset is made publicly available for future research and evaluation.

ACKNOWLEDGMENT

This work was supported by The Ministry of Education, Youth and Sports of the Czech Republic from the National Programme of Sustainability (NPU II); project IT4Innovations excellence in science – LQ1602.

REFERENCES

[1] T. Gebru, J. Krause, Y. Wang, D. Chen, J. Deng, E. L. Aiden, and L. Fei-Fei, "Using deep learning and google street view to estimate the demographic makeup of the us," 2017.

[2] 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.

[3] M. Simon and E. Rodner, "Neural activation constellations: Unsupervised part model discovery with convolutional networks," in *International Conference on Computer Vision (ICCV)*, 2015.

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

[5] Y. Gao, O. Beijbom, N. Zhang, and T. Darrell, "Compact bilinear pooling," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[6] 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.

[7] F. Zhou and Y. Lin, "Fine-grained image classification by exploring bipartite-graph labels," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[8] J. Sochor, A. Herout, and J. Havel, "Boxcars: 3d boxes as cnn input for improved fine-grained vehicle recognition," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[9] S. Huang, Z. Xu, D. Tao, and Y. Zhang, "Part-stacked cnn for fine-grained visual categorization," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[10] L. Zhang, Y. Yang, M. Wang, R. Hong, L. Nie, and X. Li, "Detecting densely distributed graph patterns for fine-grained image categorization," *IEEE Transactions on Image Processing*, vol. 25, no. 2, pp. 553–565, Feb 2016.

[11] S. Yang, L. Bo, J. Wang, and L. G. Shapiro, "Unsupervised template learning for fine-grained object recognition," in *Advances in Neural Information Processing Systems 25*, F. Pereira, C. Burges, L. Bottou, and K. Weinberger, Eds. Curran Associates, Inc., 2012, pp. 3122–3130. [Online]. Available: http://papers.nips.cc/paper/4714-unsupervised-template-learning-for-fine-grained-object-recognition.pdf

[12] K. Duan, D. Parikh, D. Crandall, and K. Grauman, "Discovering localized attributes for fine-grained recognition," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.

[13] B. Yao, "A codebook-free and annotation-free approach for fine-grained image categorization," in *Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, ser. CVPR '12. Washington, DC, USA: IEEE Computer Society, 2012, pp. 3466–3473. [Online]. Available: http://dl.acm.org/citation.cfm?id=2354409.2355035

[14] 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*, Aug 2014, pp. 26–33.

[15] X. Zhang, H. Xiong, W. Zhou, W. Lin, and Q. Tian, "Picking deep filter responses for fine-grained image recognition," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[16] H. Pirsiavash, D. Ramanan, and C. C. Fowlkes, "Bilinear classifiers for visual recognition," in *Advances in Neural Information Processing Systems 22*, Y. Bengio, D. Schuurmans, J. Lafferty, C. Williams, and A. Culotta, Eds. Curran Associates, Inc., 2009, pp. 1482–1490. [Online]. Available: http://papers.nips.cc/paper/3789-bilinear-classifiers-for-visual-recognition.pdf

[17] 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.

[18] N. Zhang, R. Farrell, and T. Darrell, "Pose pooling kernels for sub-category recognition," in *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, June 2012, pp. 3665–3672.

[19] Y. Chai, E. Rahtu, V. Lempitsky, L. Van Gool, and A. Zisserman, "Tricos: A tri-level class-discriminative co-segmentation method for image classification," in *European Conference on Computer Vision*, 2012.

[20] L. Li, Y. Guo, L. Xie, X. Kong, and Q. Tian, "Fine-Grained Visual Categorization with Fine-Tuned Segmentation," *IEEE International Conference on Image Processing*, 2015.

[21] E. Gavves, B. Fernando, C. Snoek, A. Smeulders, and T. Tuytelaars, "Local alignments for fine-grained categorization," *International Journal of Computer Vision*, vol. 111, no. 2, pp. 191–212, 2015. [Online]. Available: http://dx.doi.org/10.1007/s11263-014-0741-5

[22] V. Petrovic and T. F. Cootes, "Analysis of features for rigid structure vehicle type recognition," in *BMVC*, 2004, pp. 587–596.

[23] L. Dlagnekov and S. Belongie, "Recognizing cars," UCSD CSE Tech Report CS2005-0833, Tech. Rep., 2005.

[24] X. Clady, P. Negri, M. Milgram, and R. Poulenard, "Multi-class vehicle type recognition system," in *Proceedings of the 3rd IAPR Workshop on Artificial Neural Networks in Pattern Recognition*, ser. ANNPR '08. Berlin, Heidelberg: Springer-Verlag, 2008, pp. 228–239. [Online]. Available: http://dx.doi.org/10.1007/978-3-540-69939-2_22

[25] G. Pearce and N. Pears, "Automatic make and model recognition from frontal images of cars," in *IEEE AVSS*, 2011, pp. 373–378.

[26] A. Psyllos, C. Anagnostopoulos, and E. Kayafas, "Vehicle model recognition from frontal view image measurements," *Computer Standards & Interfaces*, vol. 33, no. 2, pp. 142 – 151, 2011, {XVI} {IMEKO} {TC4} Symposium and {XIII} International Workshop on {ADC} Modelling and Testing. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0920548910000838

[27] S. Lee, J. Gwak, and M. Jeon, "Vehicle model recognition in video," *International Journal of Signal Processing, Image Processing and Pattern Recognition*, vol. 6, no. 2, p. 175, 2013.

[28] B. Zhang, "Reliable classification of vehicle types based on cascade classifier ensembles," *IEEE Transactions on Intelligent Transportation Systems*, vol. 14, no. 1, pp. 322–332, March 2013.

[29] D. F. Llorca, D. Colás, I. G. Daza, I. Parra, and M. A. Sotelo, "Vehicle model recognition using geometry and appearance of car emblems from rear view images," in *17th International IEEE Conference on Intelligent Transportation Systems (ITSC)*, Oct 2014, pp. 3094–3099.

[30] B. Zhang, "Classification and identification of vehicle type and make by cortex-like image descriptor HMAX," *IJCVR*, vol. 4, pp. 195–211, 2014.

[31] 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*, vol. 15, no. 1, pp. 6–20, Feb 2014.

[32] C. Hu, X. Bai, L. Qi, X. Wang, G. Xue, and L. Mei, "Learning discriminative pattern for real-time car brand recognition," *Intelligent Transportation Systems, IEEE Transactions on*, vol. 16, no. 6, pp. 3170–3181, Dec 2015.

[33] 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.

[34] R. Baran, A. Glowacz, and A. Matiolanski, "The efficient real- and non-real-time make and model recognition of cars," *Multimedia Tools and Applications*, vol. 74, no. 12, pp. 4269–4288, 2015. [Online]. Available: http://dx.doi.org/10.1007/s11042-013-1545-2

[35] 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*, vol. PP, no. 99, pp. 1–11, 2015.

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

[37] 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.

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

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

[40] H.-Z. Gu and S.-Y. Lee, "Car model recognition by utilizing symmetric property to overcome severe pose variation," *Machine Vision and Applications*, vol. 24, no. 2, pp. 255–274, 2013. [Online]. Available: http://dx.doi.org/10.1007/s00138-012-0414-8

[41] 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.

[42] P. Felzenszwalb, R. Girshick, D. McAllester, and D. Ramanan, "Object detection with discriminatively trained part-based models," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 9, pp. 1627–1645, 2010. [Online]. Available: http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=5255236

[43] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, vol. 1. IEEE, 2005, pp. 886–893.

[44] J. Wang, J. Yang, K. Yu, F. Lv, T. Huang, and Y. Gong, "Locality-constrained linear coding for image classification," in *CVPR*, June 2010, pp. 3360–3367.

[45] H. Liu, Y. Tian, Y. Yang, L. Pang, and T. Huang, "Deep relative distance learning: Tell the difference between similar vehicles," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[46] N. Boonsim and S. Prakoonwit, "Car make and model recognition under limited lighting conditions at night," *Pattern Analysis and Applications*, pp. 1–13, 2016. [Online]. Available: http://dx.doi.org/10.1007/s10044-016-0559-6

[47] J. Fang, Y. Zhou, Y. Yu, and S. Du, "Fine-grained vehicle model recognition using a coarse-to-fine convolutional neural network architecture," *IEEE Transactions on Intelligent Transportation Systems*, vol. PP, no. 99, pp. 1–11, 2016.

[48] Q. Hu, H. Wang, T. Li, and C. Shen, "Deep cnns with spatially weighted pooling for fine-grained car recognition," *IEEE Transactions on Intelligent Transportation Systems*, vol. PP, no. 99, pp. 1–10, 2017.

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

[50] M. Dubská, A. Herout, R. Juránek, and J. Sochor, "Fully automatic roadside camera calibration for traffic surveillance," *Intelligent Transportation Systems, IEEE Transactions on*, vol. 16, no. 3, pp. 1162–1171, June 2015.

[51] 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.

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

[53] 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.

[54] X. Liu, W. Liu, H. Ma, and H. Fu, "Large-scale vehicle re-identification in urban surveillance videos," in *Multimedia and Expo (ICME), 2016 IEEE International Conference on*. IEEE, 2016, pp. 1–6.

[55] Y. Taigman, M. Yang, M. Ranzato, and L. Wolf, "DeepFace: Closing the gap to human-level performance in face verification," in *CVPR*, 2014, pp. 1701–1708. [Online]. Available: http://dx.doi.org/10.1109/CVPR.2014.220

[56] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems 25*, F. Pereira, C. Burges, L. Bottou, and K. Weinberger, Eds. Curran Associates, Inc., 2012, pp. 1097–1105. [Online]. Available: http://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf

[57] 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.

[58] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," in *European conference on computer vision*. Springer, 2014, pp. 818–833.

[59] J. Yang, B. Price, S. Cohen, H. Lee, and M.-H. Yang, "Object contour detection with a fully convolutional encoder-decoder network," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 193–202.

[60] R. Rothe, R. Timofte, and L. Van Gool, "Deep expectation of real and apparent age from a single image without facial landmarks," *International Journal of Computer Vision*, pp. 1–14, 2016. [Online]. Available: http://dx.doi.org/10.1007/s11263-016-0940-3

[61] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[62] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," in *Advances in Neural Information Processing Systems (NIPS)*, 2015.

[63] R. Juránek, A. Herout, M. Dubská, and P. Zemčík, "Real-time pose estimation piggybacked on object detection," in *ICCV*, 2015.

[64] J. Sochor, R. Juránek, J. Špaňhel, L. Maršík, A. Široký, A. Herout, and P. Zemčík, "BrnoCompSpeed: Review of traffic camera calibration and a comprehensive dataset for monocular speed measurement," *Intelligent Transportation Systems (under review), IEEE Transactions on*, 2016.

[65] C. Stauffer and W. E. L. Grimson, "Adaptive background mixture models for real-time tracking," in *CVPR*, vol. 2, 1999, pp. 246–252.

[66] Z. Zivkovic, "Improved adaptive gaussian mixture model for background subtraction," in *ICPR*, 2004, pp. 28–31.

[67] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *CoRR*, vol. abs/1409.1556, 2014.

Jakub Sochor received the M.S. degree from Brno University of Technology (BUT), Brno, Czech Republic. He is currently working toward the Ph.D. degree in the Department of Computer Graphics and Multimedia, Faculty of Information Technology, BUT. His research focuses on computer vision, particularly traffic surveillance – fine-grained recognition of vehicles and automatic speed measurement.

Jakub Špaňhel received his M.S. degree from Faculty of Information Technology, Brno University of Technology (BUT), Czech Republic. He is currently working toward the Ph.D. degree in the Department of Computer Graphics and Multimedia, Faculty of Information Technology, BUT. His research focuses on computer vision, particularly traffic analysis – detection and re-identification of vehicles from surveillance cameras.

Adam Herout received his PhD from Faculty of Information Technology, Brno University of Technology, Czech Republic, where he works as a full professor and leads the Graph@FIT research group. His research interests include fast algorithms and hardware acceleration in computer vision, with his focus on automatic traffic surveillance.

IEEE T-ITS (1524-9050) Supplementary Material

BoxCars: Improving Fine-Grained Recognition of Vehicles using 3D Bounding Boxes in Traffic Surveillance

Jakub Sochor, Jakub Špaňhel, Adam Herout

1 Additional BoxCars116k Dataset Statistics

# tracks	27 496
# samples	116 286
# cameras	137
# make	45
# make & model	341
# make & model & submodel	421
# make & model & submodel & model year	693

	hard	medium
# classes	107	79
# train+val cameras	81	81
# test cameras	56	56
# training tracks	11 653	12 084
# training samples	51 691	54 653
# validation tracks	637	611
# validation samples	2 763	2 802
# test tracks	11 125	11 456
# test samples	39 149	40 842

Table 1: **Left:** Statistics of our new *BoxCars116k* dataset. **Right:** Statistics about splits with different difficulty (*hard* and *medium*).

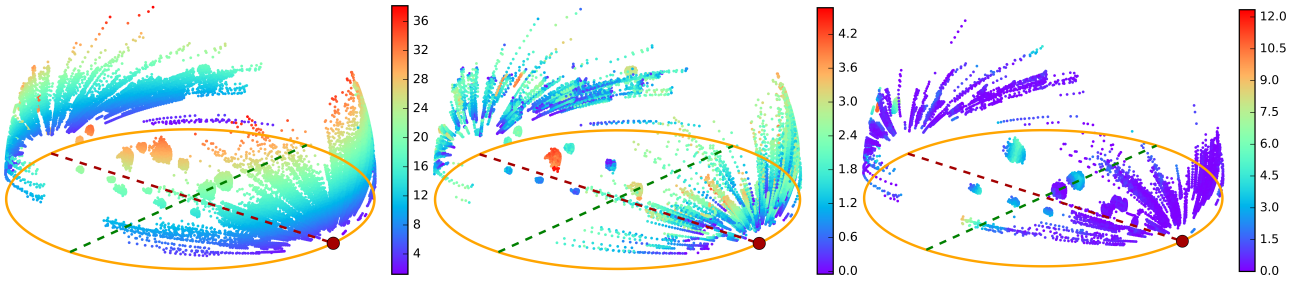

Figure 1: Viewpoints to dataset samples (horizontal flips are not included). Red dot on the unit circle denotes the frontal viewpoint. **Left:** all samples with elevation color coding (in degrees), **center:** training samples for hard split with color coded by 2D BB area (in thousands of pixels), **right:** test samples for hard split color coded by angle to the nearest training viewpoint sample (in degrees).

2 Additional Experimental Data

Due to page limit restrictions, we present some of the raw experimental data and results in this supplementary document.

2.1 Vehicle Types Resisting to Fine-Grained Recognition

net	**accuracy** [%] all types	merged types
AlexNet + ALL	77.79/88.60	79.08/89.70
VGG16 + ALL	84.13/92.27	85.42/**93.28**
VGG16+CBL + ALL	75.06/83.42	76.82/85.07
VGG19 + ALL	84.12/92.00	**85.51**/92.97
VGG19+CBL + ALL	75.62/83.76	78.56/86.62
ResNet50 + IMAGE	82.27/90.79	83.51/91.79
ResNet101 + IMAGE	83.41/91.59	84.65/92.55
ResNet152 + IMAGE	83.74/91.71	85.10/92.84

Table 2: Comparison of accuracy with all types and 8 merged types into supertypes.

Figure 2: Example of vehicle types merged into one supertype. **Left:** Renault Traffic, **right:** Opel Vivaro.

As possible applications of the fine-grained recognition may vary, we merged pairs of fine-grained classes during testing into one supertype. The merge was done for vehicles which are made by the same concern, have the same dimensions and shape, and which are only differentiated by subtle branding details on the mask. This merge can be beneficial if the task is for example determining the dimensions of the vehicle.
We merged 8 pairs of vehicle types (see Figure 2 for an example) affecting 1 034 tracks and 5 567 image samples. We show the results in Table 2; the accuracy improves only slightly – by ~ 1 percent point.

	AlexNet	VGG16+CBL	VGG19+CBL	VGG16	VGG19	**mean**	**best**
Unpack	+3.47/+4.37	+0.69/+1.06	+1.02/+1.31	+2.07/+2.51	+3.29/+3.48	+2.11/+2.55	+3.47/+4.37
View	−0.96/−1.20	−0.19/−0.19	+0.19/+0.31	−0.46/−0.93	−0.19/+0.26	−0.32/−0.35	+0.19/+0.31
Rast	−0.80/−1.18	+0.30/+0.27	+0.28/+0.72	−0.20/−0.08	+0.28/+0.09	−0.03/−0.04	+0.30/+0.72
Color	+4.80/+3.60	+2.08/+0.97	+2.47/+1.65	+2.72/+1.38	+3.79/+2.55	+3.17/+2.03	+4.80/+3.60
ImageDrop	+0.05/−0.47	+0.29/−0.43	+1.53/+0.96	+0.63/+0.07	+1.00/+0.84	+0.70/+0.20	+1.53/+0.96

Table 3: **Raw data for Table IV of the main document.** Improvements for different nets and modifications computed as $[\textit{base net} + \textit{modification}] - [\textit{base net}]$, where $[\ldots]$ stands for the accuracy of the classifier described by its contents.

	AlexNet	VGG16+CBL	VGG19+CBL	VGG16	VGG19	**mean**	**best**
Unpack	+6.93/+7.60	+2.18/+2.22	+2.06/+2.32	+2.82/+2.46	+3.07/+2.82	+3.41/+3.48	+6.93/+7.60
View	+0.09/+0.18	−0.41/−0.19	−0.78/−0.64	+0.36/+0.15	+0.05/−0.27	−0.14/−0.15	+0.36/+0.18
Rast	+0.22/+0.17	+0.11/−0.08	−0.76/−0.58	+0.30/+0.20	−0.01/−0.11	−0.03/−0.08	+0.30/+0.20
Color	+6.34/+6.18	+2.54/+1.28	+2.21/+1.31	+3.08/+1.73	+2.92/+1.67	+3.42/+2.43	+6.34/+6.18
ImageDrop	+1.07/+0.79	+4.24/+3.54	−0.79/−1.21	+0.89/+0.05	+1.19/+0.68	+1.32/+0.77	+4.24/+3.54

Table 4: **Raw data for Table V of the main document.** Improvements for different nets and modifications computed as [*base net + all*] − [*base net + all − modification*], where [...] stands for the accuracy of the classifier described by its contents.

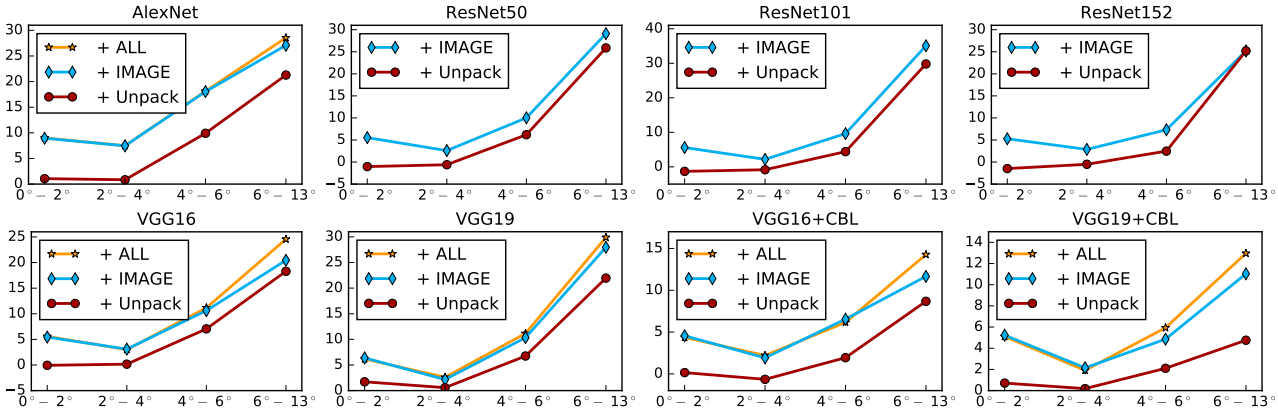

Figure 3: **All results for Figure 9 of the main document.** Correlation of improvement relative to CNNs without modification with respect to train-test viewpoint difference. The x-axis contains bins viewpoint difference bins (in degrees), and the y-axis denotes improvement compared to base net in percent points. The graphs show that with increasing viewpoint difference, the accuracy improvement of our method increases.

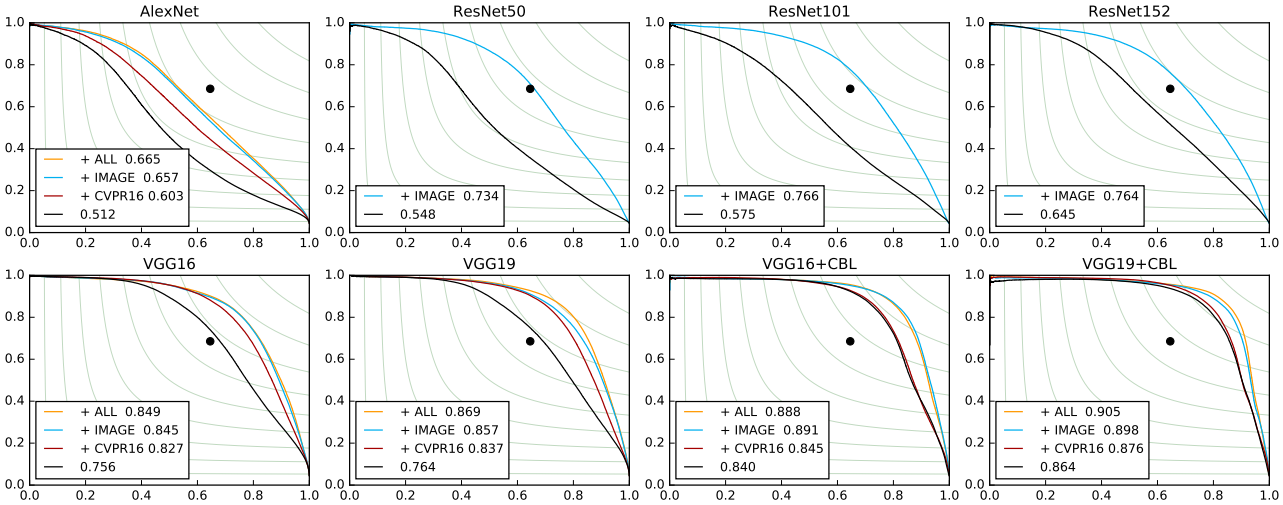

Figure 4: **All results for Figure 11 of the main document.** Precision-Recall curves for verification of fine-grained types. Black dots represent the human performance.

SPLIT: MEDIUM	accuracy [%]	improvement [pp]	error reduction [%]
AlexNet + IMAGE	77.77/88.16	+12.09/+11.64	35.21/49.57
AlexNet + ALL	77.52/87.52	+11.84/+10.99	34.49/46.82
AlexNet + CVPR16	70.90/82.18	+5.23/+5.65	15.22/24.06
AlexNet	65.68/76.53	—	—
VGG16 + ALL	83.89/91.75	+7.93/+6.36	32.99/43.55
VGG16 + IMAGE	83.93/91.69	+7.96/+6.30	33.13/43.13
VGG16 + CVPR16	79.50/88.58	+3.54/+3.19	14.71/21.86
VGG16	75.96/85.39	—	—
VGG16+CBL + IMAGE	75.67/83.49	+4.93/+3.27	16.84/16.55
VGG16+CBL + ALL	75.47/83.23	+4.73/+3.01	16.15/15.23
VGG16+CBL + CVPR16	71.07/81.02	+0.33/+0.80	1.12/4.06
VGG16+CBL	70.74/80.22	—	—
VGG19 + ALL	84.43/92.22	+9.03/+7.88	36.70/50.33
VGG19 + IMAGE	83.98/91.71	+8.58/+7.37	34.88/47.05
VGG19 + CVPR16	80.26/89.39	+4.87/+5.05	19.78/32.27
VGG19	75.40/84.34	—	—
VGG19+CBL + IMAGE	76.88/84.63	+5.34/+3.95	18.75/20.46
VGG19+CBL + ALL	75.47/83.88	+3.92/+3.20	13.79/16.58
VGG19+CBL + CVPR16	72.53/81.90	+0.98/+1.22	3.46/6.32
VGG19+CBL	71.54/80.67	—	—
ResNet50 + IMAGE	82.28/90.63	+7.21/+7.09	28.90/43.08
ResNet50	75.07/83.55	—	—
ResNet101 + IMAGE	83.10/90.80	+6.05/+5.19	26.37/36.08
ResNet101	77.05/85.61	—	—
ResNet152 + IMAGE	83.80/91.38	+5.36/+4.40	24.85/33.78
ResNet152	78.44/86.98	—	—

SPLIT: HARD	accuracy [%]	improvement [pp]	error reduction [%]
AlexNet + ALL	77.79/88.60	+11.15/+10.85	33.42/48.77
AlexNet + IMAGE	77.67/88.28	+11.02/+10.53	33.04/47.31
AlexNet + CVPR16	70.21/81.67	+3.56/+3.92	10.68/17.62
AlexNet	66.65/77.75	—	—
VGG16 + ALL	84.13/92.27	+6.88/+5.56	30.24/41.85
VGG16 + IMAGE	83.79/92.23	+6.53/+5.53	28.71/41.58
VGG16 + CVPR16	79.58/89.27	+2.32/+2.56	10.22/19.27
VGG16	77.26/86.71	—	—
VGG16+CBL + ALL	75.06/83.42	+4.67/+3.31	15.78/16.63
VGG16+CBL + IMAGE	75.04/83.16	+4.66/+3.05	15.73/15.32
VGG16+CBL + CVPR16	70.94/81.08	+0.56/+0.97	1.88/4.88
VGG16+CBL	70.38/80.11	—	—
VGG19 + IMAGE	83.91/92.17	+7.17/+6.11	30.83/43.84
VGG19 + ALL	84.12/92.00	+7.38/+5.94	31.74/42.62
VGG19 + CVPR16	79.69/89.42	+2.95/+3.36	12.69/24.11
VGG19	76.74/86.06	—	—
VGG19+CBL + ALL	75.62/83.76	+4.93/+3.50	16.82/17.71
VGG19+CBL + IMAGE	75.47/83.56	+4.78/+3.30	16.31/16.71
VGG19+CBL + CVPR16	71.92/81.64	+1.23/+1.38	4.20/6.97
VGG19+CBL	70.69/80.26	—	—
ResNet50 + IMAGE	82.27/90.79	+6.79/+6.18	27.69/40.13
ResNet50	75.48/84.61	—	—
ResNet101 + IMAGE	83.41/91.59	+6.95/+6.27	29.52/42.72
ResNet101	76.46/85.31	—	—
ResNet152 + IMAGE	83.74/91.71	+6.06/+5.51	27.16/39.93
ResNet152	77.68/86.20	—	—

Table 5: **Raw data for Table I of the main document.** Improvements of our proposed modifications for different CNNs. The accuracy is reported as single sample accuracy/track accuracy. We also present improvement in percent points and classification error reduction in the same format.