


# Graph@FIT Submission to the NVIDIA AI City Challenge 2018

Jakub Sochor    Jakub Špaňhel    Roman Juránek    Petr Dobeš    Adam Herout  
 Graph@FIT, Brno University of Technology, Czech Republic  
 {isochor, ispanhel, ijuranek, idobes, herout}@fit.vutbr.cz  
<https://medusa.fit.vutbr.cz/traffic/>


## Abstract



In our submission to the NVIDIA AI City Challenge, we address **speed measurement of vehicles and vehicle re-identification**. For both these tasks, we use a calibration method based on extracted vanishing points. We detect and track vehicles by a CNN-based detector and we construct 3D bounding boxes for all vehicles. For the speed measurement task, we estimate the speed from the movement of the bounding box in the 3D space using the calibration. Our approach to vehicle re-identification is based on extraction of visual features from “unpacked” images of the vehicles. The features are aggregated in temporal domain to obtain a single feature descriptor for the whole track. Furthermore, we utilize a validation network to improve the re-identification accuracy.

## 1. Introduction

In this submission, we address the tasks of vehicle speed measurement and re-identification of the NVIDIA AI City Challenge (i.e. *Task1* and *Task3*). We used our previously published approach to camera calibration [6, 31]. Also, for the re-identification task, we use the approach proposed in our paper [34], which is currently under review. In the paper, we show that using “unpacked” versions of vehicles [30, 33] improves the re-identification performance. Furthermore, in the paper, we propose a method for feature aggregation in temporal domain LTFD and Weighted Euclidean distance which we also use in our approach to this re-identification challenge task.



The vehicles are detected by a CNN-based detector, tracked in time and a 3D bounding box is constructed for every detected vehicle. These 3D bounding boxes are used for different purposes in the tasks. In the re-identification task, we use the 3D bounding box to produce “unpacked” version of the vehicle and normalize the input image. In the speed measurement task, the 3D bounding box is used to compute the center of the vehicle’s base, as it is a point which lies in the road plane; therefore, it can be used for

speed measurement.

To put our approach to a larger context, we include a brief overview of the state of the art in camera calibration for speed measurement and vehicle re-identification. After that, we describe the used methods for both speed measurement and re-identification in detail.

## 2. Related Work

### 2.1. Camera Calibration for Vehicle Speed Measurement

Camera calibration (obtaining intrinsic and extrinsic parameters of the surveillance camera) is critical for the accuracy of vehicle speed measurement by a single monocular camera, as it directly influences the speed measurement accuracy. There is a very recent comprehensive review of the traffic surveillance calibration methods [32], so for detailed information we refer to this review and we include only a brief description of the methods.

Several methods [11, 2, 9] are based on the detection of vanishing points as an intersection of road markings (lane dividing lines). Other methods [6, 7, 27, 3] use vehicle motion to calibrate the camera. There is also a set of methods which use some form of manually measured dimensions on the road plane [24, 25, 29, 21, 22, 4, 17].

Finally, in our previous paper [31], we proposed an algorithm for camera calibration by detecting vanishing points using edgelets and scene scale estimation based on alignment of 3D models to recognized vehicles.

### 2.2. Vehicle Re-Identification

There are mainly two types of methods – methods based on automatic license plate recognition [5, 15, 38], which are not anonymous and require zoomed-in cameras. The other type of methods is based on vehicles’ visual appearance [1, 8, 42, 19] or on a combination of both approaches [20].

Formerly, different types of *hand-crafted* features were used. For example, authors used PCA-SIFT [1], HOG descriptors and color histograms [42], SIFT-BOW and Color Names model [19], or just information about date, time,

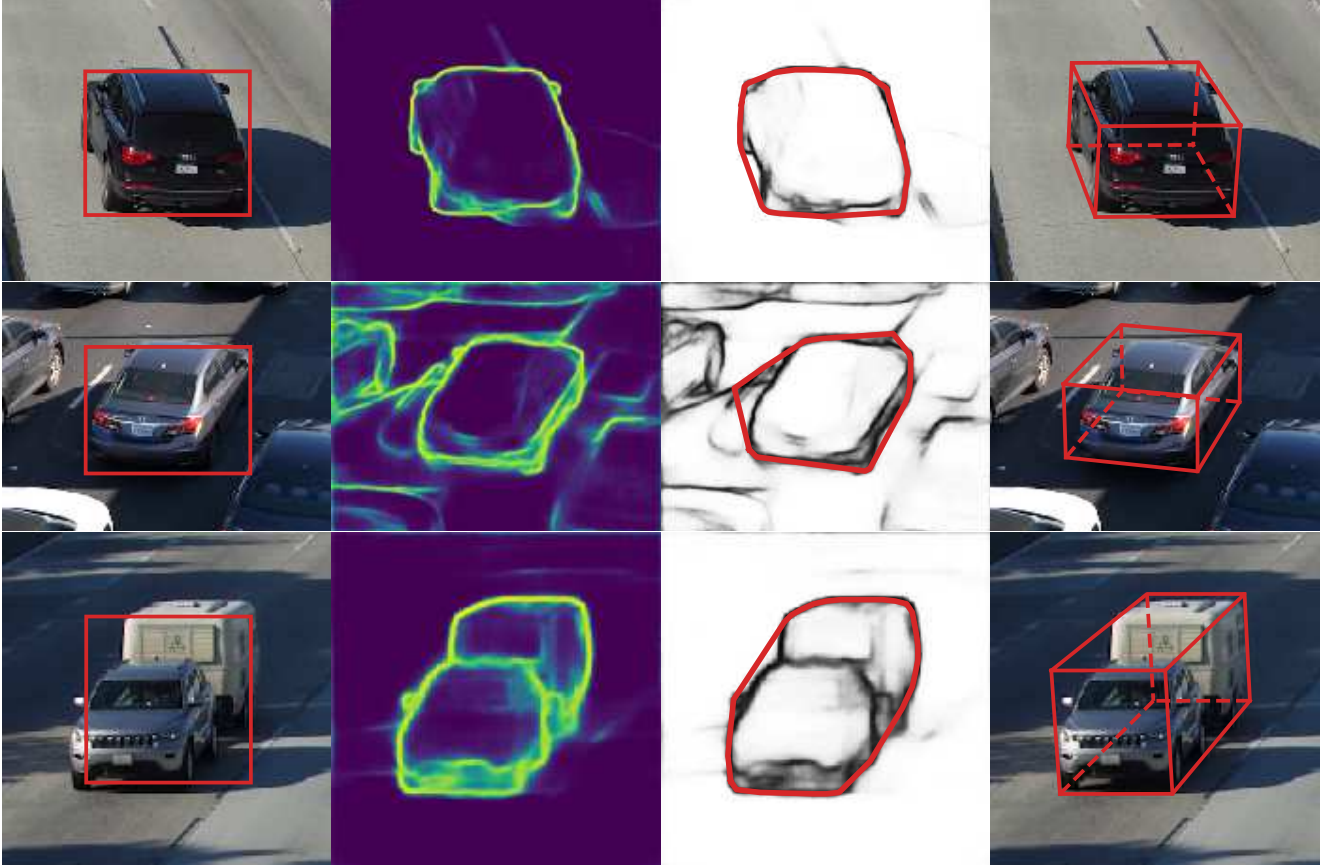


Figure 1. **Left:** detected vehicles using Faster-RCNN detector [26], **center left:** contour probability map estimated by general object contour detector [40], **center right:** estimated contour from the contour probability map [33], **right:** constructed 3D bounding box [6, 33].

color, speed and vehicles’ dimensions [8]. Recently, *deep* features learned by CNNs [18, 28, 37, 39, 43] are being used for this task. Liu et al. [20] combine the hand-crafted and deep features.

Improvements were also made by exploiting *spatio-temporal* [20, 37] or *visual-spatio-temporal* [28] properties. Some of them benefit from Siamese CNNs for license plate verification [20] or vehicle image similarities [28]. Moreover, introduction of triplet loss [43] or Coupled Cluster Loss (CCL) [18] led to accuracy improvements and faster convergence. Recently, Yan et al. [39] propose to use Generalized Pairwise Ranking or Multi-Grain based List Ranking for retrieval of similar vehicles, which performs even better than CCL.

Also, in our paper [34], which is currently under review, we propose a method for feature aggregation in temporal domain of multiple observations of the vehicle in one track.

### 2.3. Vehicle Re-Identification Datasets

Datasets of vehicles are available [16, 41, 33], which are created for fine-grained recognition with annotations on several attributes such as type, make and color. However,

the identities of the vehicles in the datasets are not known; thus, the datasets are not directly applicable for vehicle re-identification, especially for evaluation.

When it comes to genuine vehicle re-identification, Liu et al. [20] constructed a rather small VeRi-776 dataset containing 50,000 images of 776 vehicles. Liu et al. [18] collected VehicleID dataset containing 26,267 vehicles in 220k images taken from a frontal/rear viewpoint above road. Recently, Yan et al. [39] published two datasets VD1 and VD2 for vehicle re-identification and fine-grained classification with over 220k of vehicles in total, with make, model, and year annotation. However, both datasets are limited to frontal viewpoints only.

Recently, we collected dataset **CarsReId30k** [34], which contains  $\sim 30k$  of vehicle tracks from various viewpoints with precise ground truth identity acquired from a zoomed-in camera by license plate recognition.

### 3. Used Approach

In our submission to the NVIDIA AI City Challenge 2018, we focused on vehicle speed measurement (Track1) and vehicle re-identification (Track3). In the following text,

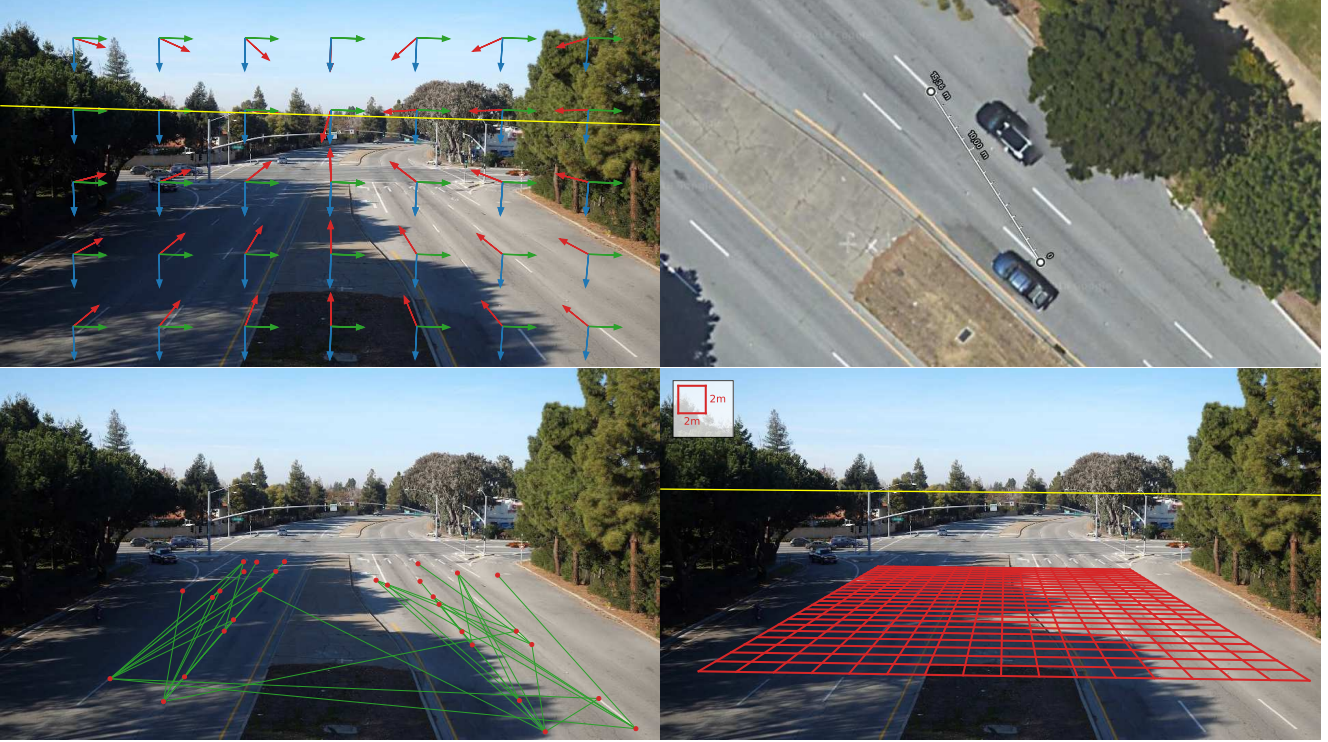


Figure 2. **Top left:** visualization of estimated 3 vanishing points (arrows with different color coding) and horizon (yellow line), **top right:** an example of used measurement on the road plane from Google Earth, **bottom left:** all measurements on the road plane, **bottom right:** final regular orthogonal grid with 2m sides.

our approach to both these tasks is described. However, we describe common processing steps of our approach to both these problems first.

### 3.1. Vehicle Detection and Data Preprocessing

As a first step of processing of the video, it is necessary to **detect vehicles** in all frames. We used Faster-RCNN [26] with ResNet101 [10] backbone. The detector was trained on UA-DETRAC [23] and COD20K [12] datasets. The detections were merged to tracks using Kalman filter [13]. Examples of the detections can be found in Figure 1 (left).

Furthermore, in every video, we detected **vanishing points** using our recent algorithm [31]. Visualization of the detected vanishing points can be found in Figure 2 (top left).

For each detected vehicle, we used general object contour detector [40] to estimate the **contours** [33] of the vehicles (Figure 1 – center left and center right) and then constructed **3D bounding boxes** of the vehicles [6] (Figure 1 right). The 3D bounding box is used for two main purposes. First, it is possible to use it to normalize the image for vehicle fine-grained recognition [30, 33] and re-identification [34]. Second, it is possible to use it for estimation of a point on the road plane which can be used for speed measurement of the vehicles [6, 31].

### 3.2. Vehicle Speed Measurement

As we obtained the camera calibration in the form of 3 vanishing points, we are able to estimate intrinsic camera parameters (i.e. focal length) and rotation of the road plane with respect to the camera. However, one unknown remains and that is the scale of the scene (i.e. distance of the road plane from camera). Therefore, it is necessary to estimate this last parameter.

To **estimate the scale**, we used measurements on the road plane. We used Google Earth to measure the real world distance of two points (Figure 2 – top right). However, as we assume that the measurements will be imprecise we used a large number of measurements ( $\sim 40$ ) on the road plane to reduce the error.

The set of such measurements  $\mathcal{M}$  was divided into two groups. The first group  $\mathcal{M}_u$  contains measurements in the direction to the first vanishing point  $u$  (represented by red arrows in Fig. 2, top left). The other group  $\mathcal{M}_v$  is computed as  $\mathcal{M}_v = \mathcal{M} \setminus \mathcal{M}_u$ .

As the second vanishing point  $v$  (green arrows in Fig. 2 – top left) is sometimes detected imprecisely (see horizon line in Figure 2 – top left), we further refine its position. To





Figure 3. Examples of regular orthogonal grids and horizon lines for every location (from top left: Loc1 – Loc4). The size of the grid cells is  $2 \times 2$  meters.

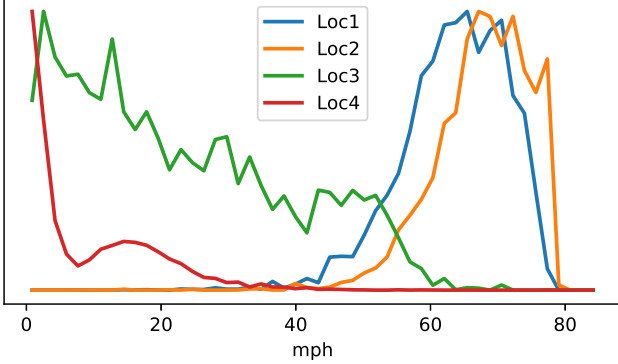


Figure 4. Histogram of all measured speeds of observed vehicles.

achieve that, we optimize the following term

$$\mathbf{v}^* = \arg \min_{\mathbf{v}} \sum_{\mathbf{p}_1, \mathbf{p}_2, m \in \mathcal{M}_{\mathbf{u}}} |m - d_{\mathbf{u}, \mathbf{v}, \mathcal{M}_{\mathbf{v}}}(\mathbf{p}_1, \mathbf{p}_2)|, \quad (1)$$

where  $d_{\mathbf{u}, \mathbf{v}, \mathcal{M}_{\mathbf{v}}}(\mathbf{p}_1, \mathbf{p}_2)$  represents the distance of two image points  $\mathbf{p}_1$  and  $\mathbf{p}_2$  in meters. It should be noted that the distance depends on the vanishing points and on the scene scale (which is computed using measurements  $\mathcal{M}_{\mathbf{v}}$ ); therefore, we include all these variables in the lower index of the function. The final scene scale  $\lambda$  is computed using all measurements  $\mathcal{M}$ . For further details about the optimization and scene scale computation see our paper [31]. **Final**

**calibrations** represented by the regular orthogonal grid can be found in Figure 3.

Using this calibration, it is possible to measure distance of two image points in the road plane in meters. Furthermore, with known video framerate, it is possible to measure elapsed time and thus compute the speed of the vehicles. To make the speed measurement stable, we measure the speed between detections 10 video frames apart.

As we need to measure the speed of a point in the road plane, we construct the center of the base of the 3D bounding box as the intersection of its diagonals. In order to further eliminate noise in the speed measurement, we represent short tracks by the median value. Longer tracks are filtered to suppress high frequencies in the speed signal. All measured speeds in all videos are shown in Figure 4.

### 3.3. Acquisition of Training Data for Vehicle Re-Identification

As the vehicle re-identification task contains vehicles observed from various viewpoints in video, it is necessary to acquire a similar dataset. We used our dataset CarsReId30k [34] which contains 8,343 unique vehicles, 29,676 observed tracks, and 92,846 positive pairs.

The dataset was collected using 8 cameras recording at the same time. Four cameras always observed the same location from different viewpoints (left, center, right), and one



Figure 5. Video screenshots from one recording session with a vehicle with the same identity. Image source: [34].

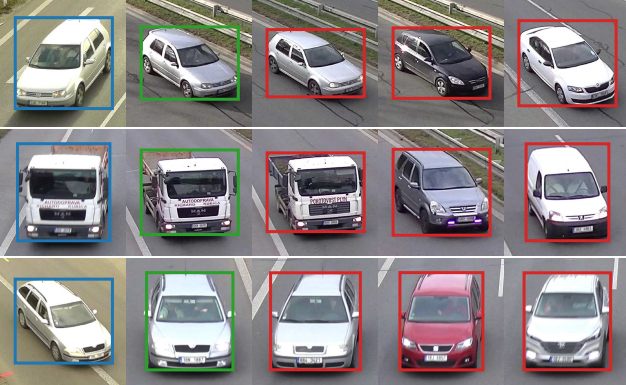


Figure 6. Examples of queries, positive, and negative samples. The negatives are sorted by difficulty from left to right (hard to easy) based on distances obtained from our re-identification feature vectors. It should be noted that the hardest negative sample has usually subtle differences (e.g. missing a small spoiler in the first row). Image source: [34].

camera was zoomed-in and it was used for license plate detection and recognition by our recent method [35]. The videos at one location were approximately synchronized and the recognized license plates were assigned to the detected vehicles from other cameras, producing the identities for all the vehicles. See Figure 5 for examples of videos from one recording session.

The **CarsReId30k dataset** [34] contains vehicles from different viewpoints but only from the frontal sides of vehicles. Therefore, we expanded the dataset and added videos from viewpoints which observe the rear side of the vehicles from different viewpoints in the same manner. The final dataset contains 17,681 unique vehicles, 73,976 observed tracks, and 292,226 positive pairs. For examples of positive and negative pairs, see Figure 6.

### 3.4. Approach to Vehicle Re-Identification

Following the methodology from our previous paper [34], we first **fine-tuned the CNN** on vehicle *identification*

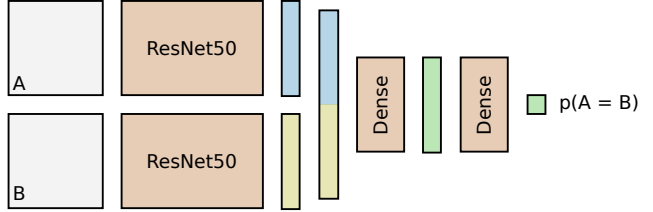


Figure 7. Validation network used for refinement of the vehicle re-identification results. The network takes two images of vehicles as inputs and produces the probability that the identity is the same for both of them.

task. We used InceptionResNetV2 [36] with “unpacked” version of vehicle images [33] and the input image size was  $331 \times 331$ . The fine-tuning was done with Adam [14] optimizer and learning rate  $1e-4$ .

Afterwards, we cached the identification features and trained **LFTD network** [34] to aggregate the features in temporal domain as there are multiple observations for the vehicle as they pass in front of the cameras. The LFTD network contains one fully connected layer with 1,024 output features and tanh non-linearity. Furthermore, the network contains feature weighting mechanism which weights different elements of the feature vectors by different weights. The network is trained as a Siamese network.

Furthermore, we used **Weighted Euclidean distance** which is expressed as

$$d_{WE}(\mathbf{u}, \mathbf{v}) = \sqrt{\sum_{i=1}^D w_i (u_i - v_i)^2}, \quad (2)$$

where  $\mathbf{u}, \mathbf{v}$  are feature vectors and  $\mathbf{w} = [w_1, w_2, \dots, w_D]$  are learned weights.

We kept queries from two recording sessions (nine recording sessions for training) for validation. The Hit@1 with identification features and average temporal pooling was 69 %. It increased to 78 % with the LFTD network and standard Euclidean distance. Finally, using the weighted



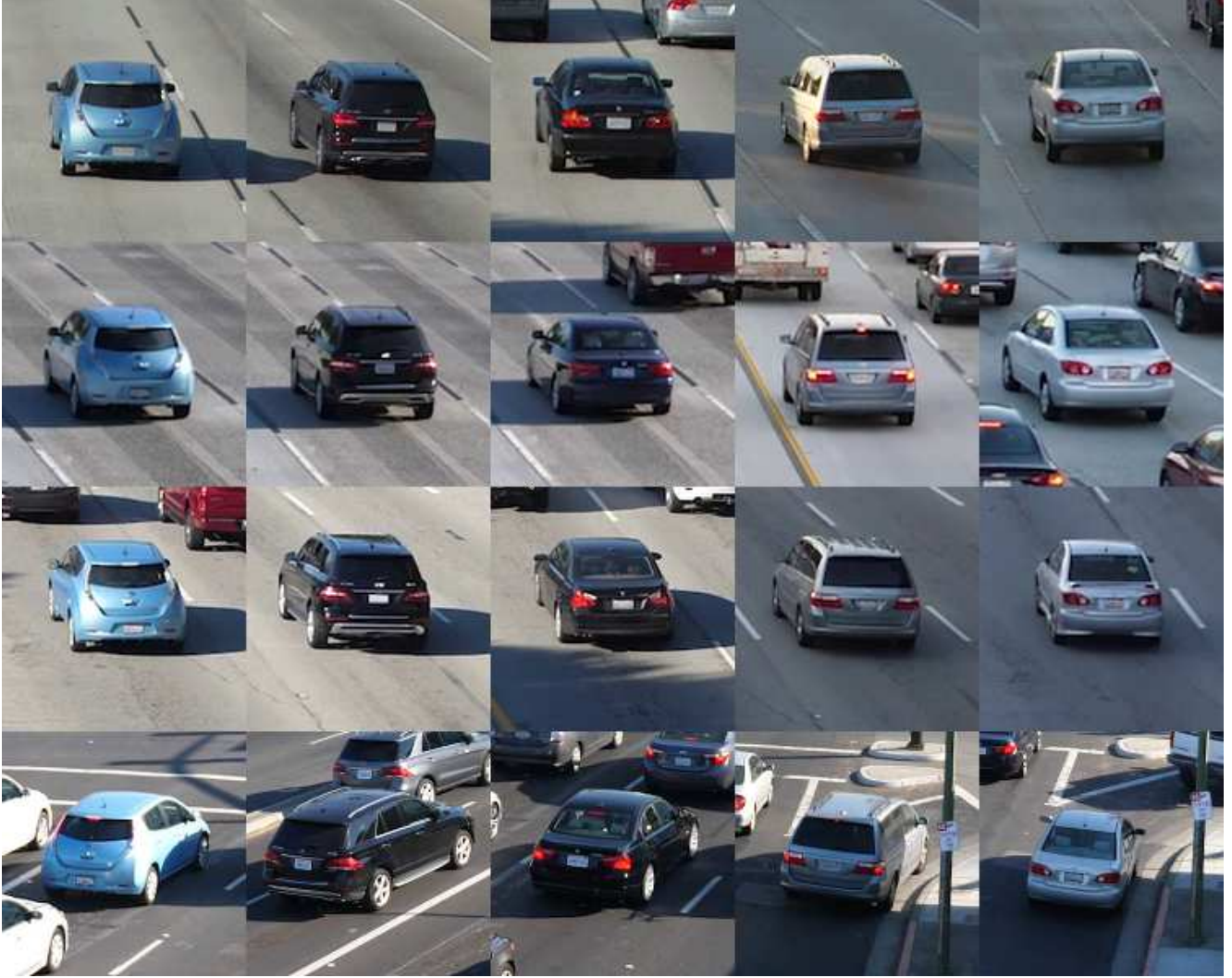


Figure 8. Each column represents a set of vehicles from all locations (from top to bottom: Loc1 to Loc4), which are considered by our re-identification method to share their identity. Only one track per location is shown.

Euclidean distance pushed the performance on validation data to 79 %.

These 1,024-dimensional features and learned weighted Euclidean distance were used for computation of **pairwise distance** between all detected and tracked vehicles at every location. We used these distances to construct quadruplets with one vehicle track from every location. These quadruplets are supposed to represent vehicle tracks with the same identity of the vehicles. As the vehicle needs to be observed at every location, we used the maximal distance of pairs within the quadruplet to measure the total similarity of the quadruplet. Therefore, we are interested only in the quadruplets with low maximal distance of pairs in the quadruplet.

To further improve the vehicle re-identification accuracy, we trained a **validation neural network** which takes two images of vehicles and produces the probability that they

have the same identity. We used the two validation sessions from our dataset for the training and the hardest negative pairs were used as negative samples during the training. Schematic design of the validation network can be found in Figure 7.

We used this validation network with every pair in the quadruplets and used median and minimal probability values as a threshold for the quadruplets and removed the ones with the probabilities below the thresholds.

Finally, we used these sorted and filtered quadruplets as the identities. We took only five best scoring quadruplets to limit false positives. As the vehicles could be observed multiple times at every location, we merged these five quadruplets with any high scoring quadruplet containing at least once the same vehicle. This way, we acquire sets of vehicles with assumed same identity as required by the task

rules. For examples of such sets see Figure 8.

## 4. Conclusions

We participated in two tasks of the NVIDIA AI City Challenge 2018: the speed measurement task and vehicle re-identification task. Our solution is using camera calibration by extracting vanishing points and scene scale calibration by measuring distances in orthophoto maps. The camera calibration is used for speed measurement but also for constructing 3D bounding boxes around the vehicles (their contours) which help normalizing and “unpacking” the images for further processing by neural nets. The re-identification task is using these preprocessed images, makes pair-wise comparisons of detected vehicles, constructs quadruplets of vehicles appearing at all the given locations and verifies these sets by another neural net. The neural networks have been trained on our previously collected data sets collected for vehicle surveillance.

## Acknowledgements

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. Also, this work was supported by TACR project “SMARTCarPark”, TH03010529.

## References

- [1] C. Arth, C. Leistner, and H. Bischof. Object reacquisition and tracking in large-scale smart camera networks. In *2007 First ACM/IEEE International Conference on Distributed Smart Cameras*, pages 156–163. IEEE, 2007.
- [2] F. Cathey and D. Dailey. Mathematical theory of image straightening with applications to camera calibration. In *Intelligent Transportation Systems Conference*, 2006.
- [3] D. Dailey, F. Cathey, and S. Pumrin. An algorithm to estimate mean traffic speed using uncalibrated cameras. *IEEE Transactions on Intelligent Transportation Systems*, 1(2):98–107, 2000.
- [4] V.-H. Do, L.-H. Nghiem, N. P. Thi, and N. P. Ngoc. A simple camera calibration method for vehicle velocity estimation. In *Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON), 2015 12th International Conference on*, pages 1–5, June 2015.
- [5] S. Du, M. Ibrahim, M. Shehata, and W. Badawy. Automatic license plate recognition (ALPR): A state-of-the-art review. *IEEE Trans. on Circuits and Systems for Video Technology*, 23(2):311–325, 2013.
- [6] M. Dubská, J. Sochor, and A. Herout. **Automatic camera calibration for traffic understanding**. In *BMVC*, 2014.
- [7] M. Dubská, A. Herout, R. Juranek, and J. Sochor. Fully automatic roadside camera calibration for traffic surveillance. *Intelligent Transportation Systems, IEEE Transactions on*, 16(3):1162–1171, June 2015.
- [8] R. S. Feris, B. Siddiquie, J. Petterson, Y. Zhai, A. Datta, L. M. Brown, and S. Pankanti. Large-scale vehicle detection, indexing, and search in urban surveillance videos. *IEEE Trans. on Multimedia*, 14(1):28–42, 2012.
- [9] L. Grammatikopoulos, G. Karras, and E. Petsa. Automatic estimation of vehicle speed from uncalibrated video sequences. In *Proceedings of International Symposium on Modern Technologies, Education and Professional Practice in Geodesy and Related Fields*, pages 332–338, 2005.
- [10] 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.
- [11] X. C. He and N. H. C. Yung. A novel algorithm for estimating vehicle speed from two consecutive images. In *IEEE Workshop on Applications of Computer Vision, WACV*, 2007.
- [12] R. Juránek, A. Herout, M. Dubská, and P. Zemčík. **Real-time pose estimation piggybacked on object detection**. In *ICCV*, 2015.
- [13] R. E. Kalman. **A new approach to linear filtering and prediction problems**. *Transactions of the ASME—Journal of Basic Engineering*, 82(Series D):35–45, 1960.
- [14] D. Kinga and J. B. Adam. A method for stochastic optimization. In *International Conference on Learning Representations (ICLR)*, 2015.
- [15] K. Kluwak, J. Segen, M. Kulbacki, A. Drabik, and K. Wojciechowski. *ALPR - Extension to Traditional Plate Recognition Methods*, pages 755–764. Springer Berlin Heidelberg, Berlin, Heidelberg, 2016.
- [16] J. Krause, M. Stark, J. Deng, and L. Fei-Fei. **3D object representations for fine-grained categorization**. In *The IEEE International Conference on Computer Vision (ICCV) Workshops*, June 2013.
- [17] J. Lan, J. Li, G. Hu, B. Ran, and L. Wang. Vehicle speed measurement based on gray constraint optical flow algorithm. *Optik - International Journal for Light and Electron Optics*, 125(1):289 – 295, 2014.
- [18] 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.
- [19] X. Liu, W. Liu, H. Ma, and H. Fu. Large-scale vehicle re-identification in urban surveillance videos. In *IEEE International Conference on Multimedia and Expo (ICME)*, pages 1–6. IEEE, 2016.
- [20] X. Liu, W. Liu, T. Mei, and H. Ma. A deep learning-based approach to progressive vehicle re-identification for urban surveillance. In *ECCV*, pages 869–884. Springer, 2016.
- [21] D. Luvizon, B. Nassu, and R. Minetto. Vehicle speed estimation by license plate detection and tracking. In *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on*, pages 6563–6567, May 2014.
- [22] D. C. Luvizon, B. T. Nassu, and R. Minetto. A video-based system for vehicle speed measurement in urban roadways. *IEEE Transactions on Intelligent Transportation Systems*, PP(99):1–12, 2016.
- [23] S. Lyu, M.-C. Chang, D. Du, L. Wen, H. Qi, Y. Li, Y. Wei, L. Ke, T. Hu, M. Del Coco, et al. Ua-detrac 2017: **Report**





of avss2017 & iwt4s challenge on advanced traffic monitoring. In *Advanced Video and Signal Based Surveillance (AVSS), 2017 14th IEEE International Conference on*, pages 1–7. IEEE, 2017.

- [24] C. Maduro, K. Batista, P. Peixoto, and J. Batista. Estimation of vehicle velocity and traffic intensity using rectified images. In *Image Processing, 2008. ICIIP 2008. 15th IEEE International Conference on*, pages 777–780, Oct 2008.

- [25] A. Nurhadiyatna, B. Hardjono, A. Wibisono, I. Sina, W. Jatmiko, M. Ma’sum, and P. Mursanto. Improved vehicle speed estimation using gaussian mixture model and hole filling algorithm. In *Advanced Computer Science and Information Systems (ICACSIS), 2013 International Conference on*, pages 451–456, Sept 2013.



- [26] 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.

- [27] T. Schoepflin and D. Dailey. Dynamic camera calibration of roadside traffic management cameras for vehicle speed estimation. *Intelligent Transportation Systems, IEEE Transactions on*, 4(2):90–98, June 2003.

- [28] Y. Shen, T. Xiao, H. Li, S. Yi, and X. Wang. Learning deep neural networks for vehicle re-id with visual-spatio-temporal path proposals. In *The IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.

- [29] I. Sina, A. Wibisono, A. Nurhadiyatna, B. Hardjono, W. Jatmiko, and P. Mursanto. Vehicle counting and speed measurement using headlight detection. In *Advanced Computer Science and Information Systems (ICACSIS), 2013 International Conference on*, pages 149–154, Sept 2013.

- [30] 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.



- [31] J. Sochor, R. Juránek, and A. Herout. **Traffic surveillance camera calibration by 3d model bounding box alignment for accurate vehicle speed measurement**. *Computer Vision and Image Understanding*, 161:87 – 98, 2017.

- [32] J. Sochor, R. Juránek, J. Špaňhel, L. Maršík, A. Šíroký, A. Herout, and P. Zemčík. BrnoCompSpeed: **Review of traffic camera calibration and comprehensive dataset for monocular speed measurement**. arXiv:1702.06441, 2017.

- [33] J. Sochor, J. Špaňhel, and A. Herout. Boxcars: **Improving fine-grained recognition of vehicles using 3-d bounding boxes in traffic surveillance**. *IEEE Transactions on Intelligent Transportation Systems*, PP(99):1–12, 2018.

- [34] J. Sochor, J. Špaňhel, R. Juránek, and A. Herout. **Learning feature aggregation in temporal domain for re-identification**. In *ECCV (under review)*, 2018.

- [35] J. Špaňhel, J. Sochor, R. Juránek, A. Herout, L. Maršík, and P. Zemčík. Holistic recognition of low quality license plates by cnn using track annotated data. In *2017 14th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, pages 1–6. IEEE, Aug 2017.

- [36] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. A. Alemi. Inception-v4, inception-resnet and the impact of residual connections on learning. In *AAAI*, volume 4, page 12, 2017.

- [37] Z. Wang, L. Tang, X. Liu, Z. Yao, S. Yi, J. Shao, J. Yan, S. Wang, H. Li, and X. Wang. Orientation invariant feature embedding and spatial temporal regularization for vehicle re-identification. In *The IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.

- [38] Y. Wen, Y. Lu, J. Yan, Z. Zhou, K. M. von Deneen, and P. Shi. An algorithm for license plate recognition applied to intelligent transportation system. *IEEE Trans. on Intelligent Transportation Systems*, 12(3):830–845, 2011.

- [39] K. Yan, Y. Tian, Y. Wang, W. Zeng, and T. Huang. Exploiting multi-grain ranking constraints for precisely searching visually-similar vehicles. In *The IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.

- [40] 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*, pages 193–202, 2016.

- [41] 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.

- [42] D. Zapletal and A. Herout. **Vehicle re-identification for automatic video traffic surveillance**. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 25–31, 2016.

- [43] Y. Zhang, D. Liu, and Z.-J. Zha. Improving triplet-wise training of convolutional neural network for vehicle re-identification. In *Multimedia and Expo (ICME), 2017 IEEE International Conference on*, pages 1386–1391. IEEE, 2017.

