Determining Hybrid Re-id Features of Vehicles in Videos for Transport Analysis

Dávid Papp and Regő Borsodi

Abstract—**The research topic presented in this paper belongs to computer vision problems in the transport application area, where the statistical data of the results give the input for the transport analysis. Although object tracking in a controlled environment could be performed with good results in general, accurate and detailed annotation of vehicles is a common problem in traffic analysis. Such annotation includes static and dynamic attributes of numerous vehicles. Most recent object trackers employ CNNs to compute the so-called re-identification features of the bounding boxes. In this paper we introduce hybrid re- identification features, which combine latent, static, and dynamic attributes to improve tracking. Furthermore, we propose a lightweight solution that could be integrated in a real-time multi- camera tracking system.**

Index Terms—**transport analysis, deep learning, feature extraction, re-identification, multiple object tracking, multi-target multi-camera tracking**

I. Introduction

The subset of Intelligent Transport Systems allows cooperation [15] among the vehicles and infrastructure, which is called Co-operative Transport System (CTS). CTS systems are designed for cooperative sensing and predicting flow, infrastructure and environmental conditions surrounding traffic, with a goal of improving the safety and efficiency of road transport operations [28]. The efficiency depends on the individual vehicles as well, for example their route planning, as an optimization problem. The uncertainty influences the route; however, a sophisticated model with an appropriate algorithm can handle this uncertainty to find the best route [31]. Finding a good solution for route planning in a transport network is a general problem with arbitrary network type, like a network of buses, a network of tram rails, or any other type of a transport network [30].

Video-based vehicle behavior analysis is done by following and annotating the vehicles across multiple cameras. This requires accurate multi-target multi-camera tracking (MTMC) that must be built upon information coming from single cameras. The detection and tracking of multiple vehicles on a single-camera is frequently referred to as MTSC (multi-target single-camera tracking) or MOT (multiple object tracking). These methods first run an object detector network to detect all object instances, whose bounding boxes are then matched with the trajectories based on previous frames. A critical part of fusing MOTs into MTMC is matching the individual aims to retrieve images from a *gallery* that contain the object of the same identity as a provided *query* image. Recent solutions for MOT extract feature vectors (so called re-id features) using special CNNs (for example ResNet-IBN variants [23]) and rank gallery images based on their cosine similarity to the query [9], [21], [45]. To improve single-camera tracking, some MOT methods employ the re-id features to help the association between bounding boxes.

Static and dynamic attributes (such as axle number, differentiating signs or velocity) of the vehicles could aid MTMC trajectory matching. Determining these attributes require frame by frame analysis. Passing vehicles usually appear in several, most frequently (but not necessarily) neighboring, frames. Thus, to determine dynamic features, it is required to correctly identify their trajectories including all bounding boxes of the object during their progress in front of the camera. For calculating static features this is not necessary in general, but it could enhance accuracy by using an ensemble decision. The same reasoning holds for a system of multiple cameras, where vehicle re-identification and tracking is preferable.

In this paper we introduce hybrid re-id features, which combines latent features, static and dynamic attributes of the vehicle, and ordinary re-id features. We examine different scenarios to calculate the hybrid re-id feature, from most accurate to most lightweight, that could even be used in a real-time MTMC system.

II. Related works

A. Transport Analysis

In transport networks different situations can be analyzed, one of which is equilibrium at the case of uncertainty situations, where the uncertainty comes from lack of information. The uncertainty can be represented by Dempster-Shafer theory, an interval-based solution has been developed for handling this situation [29]. In transport analysis different influencing factors of the traffic congestion can be investigated on the roads using uncertain probabilities described by probability intervals [32].

Vehicle behavior analysis consists of some parts, like car-following, lane change maneuvers, velocities of the cars, etc. As the fundamental control strategy of intelligent vehicles, car-following control directly affects vehicle performance. In practical driving, drivers usually predict the behavior of vehicles in the adjacent lane before modulating the driving strategy of the host vehicle [6]. Prediction of lane change maneuvers intended by the driver is solved by an artificial neural network with fusing features modeling the environmental situation [14]. The characterization of vehicles'

Department of Telecommunications and Media Informatics, Budapest University of Technology and Economics (BME), Budapest, Hungary (e-mail: pappd@tmit.bme.hu, rego.borsodi@edu.bme.hu)

DOI: 10.36244/ICJ.2022.1.3

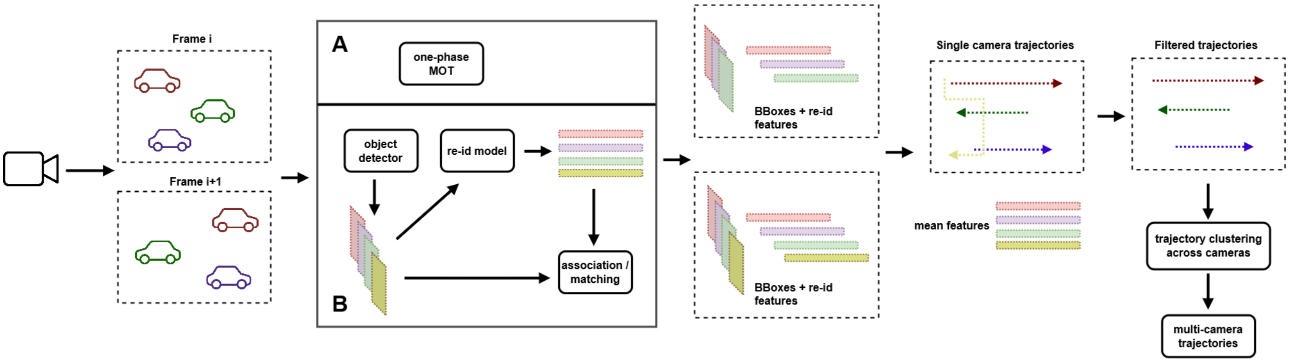

Fig. 1.: Overview of the MTMC tracking process using one-phase single-camera tracking (A) or a two-phase one (B). The yellow trajectory is an erroneous detection; thus, it is filtered out in the single-camera process.

behavior based on their velocities can be modelled by information theory [1]. A vehicle behavior analysis system can be used in traffic jams and under complex weather conditions [26]. To analyze the behavior of vehicles we need determine the static and dynamic features of vehicles in videos, which belongs to the discipline of computer vision.

B. *Computer Vision*

Most solutions for MOT can be categorized as either *one-phase* or *two-phase* approaches. Two-phase methods first run object detection to get the bounding boxes, then extract (re-id) features of the detected objects. For the association step the SORT [2] method uses Kalman filter [10] to predict object locations and computes the overlap with detected objects. The matching is performed with the Hungarian algorithm [13], with the nodes of the graph being the bounding boxes on neighboring frames. The IOU tracker [3], on the other hand, does the matching based entirely on the overlaps of bounding boxes, without the use of the Kalman filter, thus reaching a higher frame rate.

To improve tracking, some two-phase methods, such as DeepSORT [37] - an improved version of SORT, use deep learning to extract re-id features of the detected objects. The re-id features and the IOU (intersection over union) of bounding boxes are used to compute a cost matrix, which is utilized to do the linking task using Kalman filter and the Hungarian algorithm. This approach delivers decent performance in MOTA (multi-object tracking accuracy), however, the two different deep learning models (for object detection and re-id embedding) do not share architecture and, as the networks are run sequentially, the total inference time is the sum of the individual execution times. Moreover, in crowded scenes, the re-id network must be run separately for tens of bounding boxes, further increasing the total running time.

One-phase approaches merge the object detection and re-id embedding phases into a single network, thus, reaching real-t [illegible] and Embedding) [34] tracker uses a FPN (Feature Pyramid Network) [16] built on a Yolov3 [5] backbone. The prediction heads on top of the FPN produce objectness scores, box offset and box size for each anchor and location while also yielding the re-id features. The recently proposed FairMOT [42] tracker eliminates anchors and seeks to strike a balance between accurate detection and re-id features. The prediction head is built on a modified DLA (Deep Layer Aggregation) [41] network on top of a ResNet-34 [7]. The network processes over 25 frames per second on multiple benchmarks [42].

The extraction of re-id features is a crucial part of MOT methods [34], [37], [42]. On the one hand, one-phase trackers such as JDE or FairMOT learn embeddings together with detection by utilizing cross entropy loss or variations of triplet loss [34], [42]. As video datasets with bounding box and identity annotations are scarce, weakly supervised learning was introduced, utilizing images with bounding box annotations, and treating transformed variants of the same objects as the same identity [42]. On the other hand, in a two-phase MOT (scenario B in Figure 1), a separate re-id model is trained for extracting accurate embeddings. Commonly used models for this purpose are IBN-net variants with a ResNet [7] or ResNeXt [38] backbone. *Zhu et al* trained three models for extracting features describing the vehicle, camera, and orientation, then in the final similarity, camera and orientation similarities are subtracted from vehicle similarity to reduce the bias [45]. Given the initial ranking based on similarities, several re-ranking methods have been introduced to improve accuracy, such as the K-reciprocal nearest neighbor method, that favors gallery images having a similar set of k nearest neighbors to the query image [43].

III. MTMC VEHICLE TRACKING

A high-level overview of MTMC process is shown in Fig. 1. Video streams are fed into a one-phase (A) or a two-phase (B) tracker, which both provide bounding boxes, re-id features, and class confidence levels. Tracking algorithms (e.g. DeepSORT) generate a trajectory when no more bounding boxes are appended to it for a given interval of frames [37]. Trajectory filtering is a camera-specific step, when stationary, too noisy, or unnecessary trajectories, e.g. those containing pedestrians or off-road vehicles, are discarded. When a single-camera trajectory is finalized, it is matched with trajectories on other cameras to create multi-camera tracklets. This step is usually done by clustering the mean feature vectors of tracklets [17].

Multi-target multi-camera tracking has been mostly studied as an *offline* task. For example, the test dataset on Track 3 of the AI City challenge [22] contains 20 minutes of traffic videos from 6 non-overlapping cameras. Many solutions first ran MOT on all cameras, and when all trajectories were available, multi-camera trajectory clustering was deployed [17], [40]. As the locations of cameras were available, spatial-temporal constraints were considered, which greatly reduced the number of possible trajectory matchings. If such constraints are not available, the inter-camera matching can only be done based on vehicle appearance, which becomes increasingly difficult with the growth of the dataset.

Commonly used MOT systems operate in an online manner [3], [34], [37], [42]. In online MTMC, when a single-camera trajectory is generated, it should be immediately connected to an existing multi-camera tracklet or used to initialize a new one. The exact details of this operation heavily depend on the spatial-temporal constraints based on cameras. State-of-the-art single-camera trackers (e.g. FairMOT) also achieve real-time tracking, reaching 25-30 frames per second on the MOT15, MOT16, and MOT17 benchmarks [42]. However, real-time MTMC would require cameras to be well synchronized, and a new vehicle appearing on a camera to be matched with trajectories (or even newly appearing vehicles) from other cameras immediately as it is detected, which would likely deteriorate MTMC accuracy. However, we still consider the running time of the system, including the extraction of static and dynamic features, as it is preferable to be able to process video streams with at least the same speed as they are generated (even if the tracking and extraction do not run strictly in a real-time manner).

IV. MTMC VEHICLE TRACKING USING HYBRID RE-ID

A. Re-id model

For training re-id models, huge datasets, containing multiple images of the same vehicle identities are available (see Table 1) in contrast to the case of one-phase trackers described previously. The VehicleX [39] rendering engine also comes handy in training state-of-the-art re-id models. Firstly, for training an orientation model in the commonly used VOC approach, a dataset annotated with vehicle orientation labels is generated using VehicleX, as creating a real-world dataset containing such labels would require tremendous amount of work. Moreover, extending the dataset with artificial images from VehicleX can alone improve the quality of the features [22].

TABLE I
VEHICLE RE-IDENTIFICATION DATASETS.

Dataset	#Bboxes	#Identities
VRIC [11], [35]	60K	5622
CityFlow(v2) [33]	313K	880
VeRi-776 [19]	50K	776
VeRi-Wild [20]	416K	40K
VehicleID [18]	221K	26K
VehicleX [39]	∞	1362

B. Static and Dynamic attributes

Fig. 2 shows the set of dynamic and static attributes that are associated with each detected object. Each of these attributes could be calculated from the bounding box directly, or we could exploit the re-id features, as those are already computed and should be good representations of the objects.

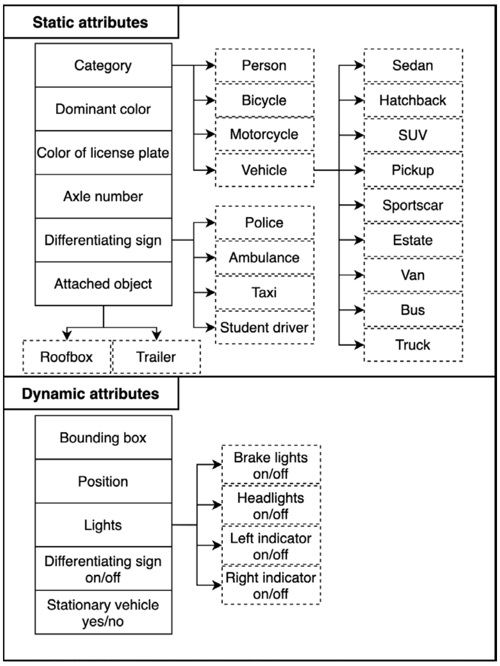

Fig. 2.: Hierarchy of static and dynamic attributes of vehicles; solid line boxes represent the top level

Extracting static and dynamic characteristics of vehicles is generally the output of the annotation process; however, we propose to feed this information back into the MTMC system in order to improve tracking and trajectory matching. Since dynamic features are time dependent, it is required to extract them for all bounding boxes of the object during their progress in front of the camera. Overlapping and well synchronized cameras allow to compensate for false negatives; otherwise, they are replaced with the attributes extracted in neighboring frames. In an ideal situation, static features are also calculated for all bounding boxes, but this is not necessary, because they are constant for the entire trajectory. There are essentially two ways to determine static features:

- Weighted majority vote of frame-by-frame extraction
- Mean of best (high confidence, close to camera) detections

Fig. 3 proposes multiple scenarios for extracting static features. The features can be determined by reusing the re-id features, either by feeding them into a single NN, that has a divided prediction head for each task (C), or by training one classifier for each static feature (E). These classifiers could be NNs, SVMs, GBMs or even random forests. In scenarios D and F, the region of interests (ROIs) from cropped bounding boxes are fed into either a single CNN with a stacked prediction head (D) or into separate CNNs (F). Most likely, scenario F provides the most accurate predictions, however, it requires multiple networks to run for each cropped bounding box on all frames. The process can be optimized by running the models on only

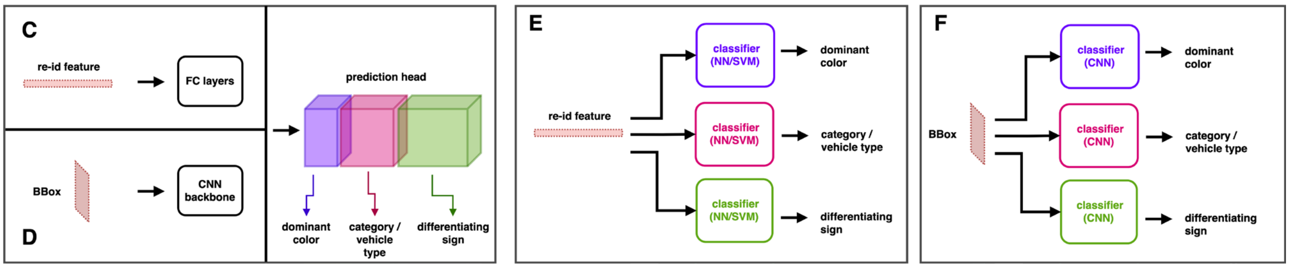

Fig. 3.: Different scenarios for extracting static features: feeding re-id features into a single fully connected network (C), classifying bounding boxes with a single CNN (D), using separate classifiers to extract features from re-id vectors (E), running separate classifiers for cropped bounding boxes (F).

some designated Bboxes, after finalizing a single-camera trajectory. In scenarios C and E, if the static features are determined using the mean re-id features, the inference runs once per trajectory, which is the most lightweight solution.

C. Fusion of features

Hybrid re-id features are created in two phases, first during single-camera tracking, then during multi-camera trajectory matching. In the former case, temporal static features are merged with the re-id features of bounding boxes; while in the latter, the finalized static features are merged with the mean re-id features. In case of well synchronized cameras, dynamic attributes could also be used in the trajectory matching, but we do not consider this situation. Dynamic attributes such as brake light on/off could help filtering out candidate bounding boxes during the association step. In case of architectures D and F (Fig. 3), dropping the prediction layers results in a feature extractor network, whose features then can be merged with the temporal re-id features. The same holds for trajectory matching, as methods C and E deliver the interpretable static attributes (e.g. license plate color, differentiating sign), which then can be used for filtering purposes. Whereas following architectures D or F gives latent attributes.

Fig. 4 shows the multi-target multi-camera vehicle tracking using hybrid re-id features. It is basically the same process as shown in Fig. 1, and therefore we grayed the closely related but less relevant elements (regarding hybrid re-id features), while omitted the non-related ones. The first part of the process is a two-phase single-camera tracking, which is followed by the multi-camera trajectory filtering and matching. As it can be seen in Fig. 4, hybrid re-id features are used in both; highlighted with blue boxes. Furthermore, classification of basic re-id and mean re-id features produces interpretable attributes that are integrated into the filtering approach; highlighted with green boxes. In case of multi-camera trajectory filtering, these attributes are exclusively static attributes. What is more interesting is that the matching step in single-camera tracking could benefit from the dynamic attributes as well (as mentioned above). We call the dynamic and static attributes on the bounding box level together as temporal attributes.

D. Style transfer

Image properties like lighting conditions and color distribution heavily depend on the camera, thus, when the images used for training and testing a re-id network were captured by different cameras, feature vector quality decreases. The domain bias is even more obvious between images generated by VehicleX and real-world images. For the domain adaptation of images SPGAN [4] was used in practice, however SPGAN was designed for images containing people, thus a new network, VTGAN [24] was proposed for vehicles. MixStyle is another domain generalization technique, which does not require to modify training images (in contrast to GAN methods). It was used for training a vehicle re-id baseline by Huyn et al [9]. MixStyle [44] mixes features at the bottom layers of a CNN between instances from different domains, thus improving domain generalization. MixStyle takes an input batch x and shuffles it to create $\hat{x}$. Then standardizes x, and scales back according to the mixed statistics:

$$\mathrm{MixStyle}(x) = \gamma_{mix} \cdot \frac{x - \mu(x)}{\sigma(x)} + \beta_{mix} \tag{1}$$

where

$$\gamma_{mix} = \lambda\sigma(x) + (1 - \lambda)\sigma(\hat{x}) \tag{2}$$

$$\beta_{mix} = \lambda\mu(\mathrm{x}) + (1 - \lambda)\mu(\hat{x}) \tag{3}$$

and λ is a vector, whose elements are sampled from a $\mathrm{Beta}(\alpha, \alpha)$ distribution. If possible, in $\hat{x}$ and x, samples at the same positions are from different domains, thus mixing their feature distributions. Two viable options for re-using multiple public re-id datasets are: inserting MixStyle into our network or train a GAN variant (like VTGAN) for each foreign dataset and transform its images into the style of our domain.

E. Loss function

Choosing appropriate loss functions is critical in training re-id networks. A common technique is to use a weighted sum of two types of losses: *id loss* and *metric loss*. The id loss is measured at the classification layer of the network, while the metric loss is at the feature extraction layer, and its goal is to make features of the same id converge and those from different classes diverge. Triplet loss [25], center loss [36], circle loss [27] and supervised contrastive loss [12] are commonly used as metric losses, while cross entropy is a typical id loss. The weight of id loss and metric loss in the final loss formula can be adapted during the training (in contrast to using constant values), as proposed in [9].

V. TRAJECTORY FILTERING AND MATCHING

The trajectory filtering step (applied to single-camera trajectories) depends on the constellation of cameras. We consider a crossroad with four cameras pointing inwards. Such a scenario is shown in Fig. 5. We define zones as proposed by

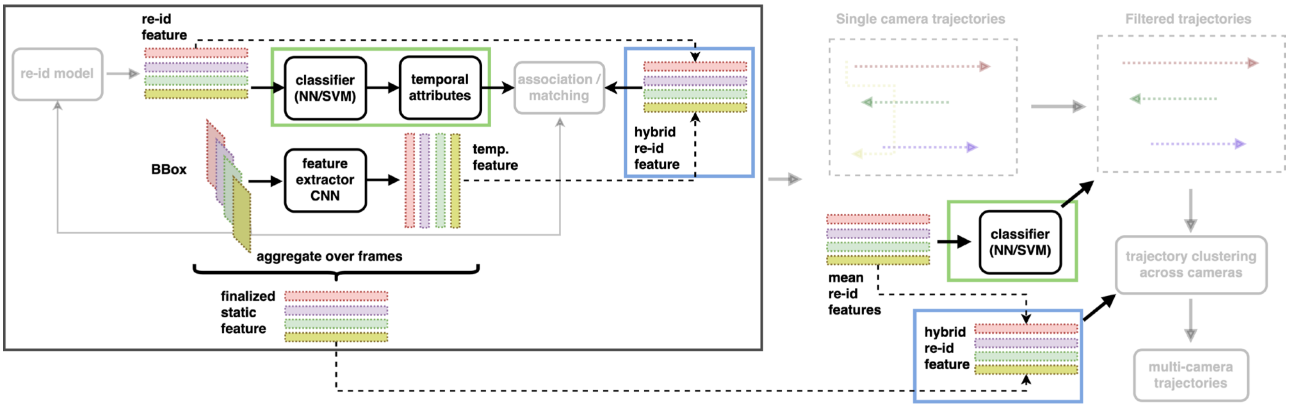

Fig. 4.: Overview of MTMC vehicle tracking using hybrid re-id features. Static attributes and basic re-id features are fused at two parts; highlighted with blue boxes. Classification of basic re-id features gives interpretable attributes for filtering; highlighted with green boxes.

Hsu et al. [8]. If a single-camera trajectory does not start and end in one of the zones or is stationary for a long period (false prediction), then it can be filtered out. When matching trajectories across cameras, only those have to be considered that start and end in the same zone. The constraints, of course, need to be adjusted to the field of view of the cameras, because it is possible that not all cameras have a view of all zones. Another possible constellation is a series of cameras on a highway, with two zones (one direction) or four (two directional) and possible additional ones if the camera has a view on a highway ramp.

The multi-camera trajectory matching step has a strict temporal constraint in the crossroad scenario. If the video streams from cameras are synchronized, or the delays are known, almost exact timestamps are available about vehicles entering and leaving zones, thus the trajectory matching step becomes a simple association step, like the single-camera scenario.

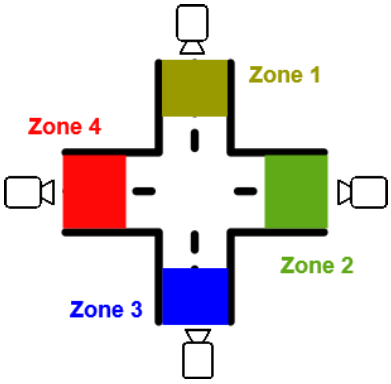

Fig. 5.: Common camera constellation at a crossroad

VI. Conclusion

In this paper we elaborated an approach for multi-target multi-camera tracking using hybrid re-id features. The hybrid re-id features are created from static attributes and (basic) re-id features. However, this requires a two-phase tracking method, which is computationally more expensive than one-phase ones; furthermore, the calculation of static attributes comes with additional computation cost. We propose multiple scenarios to calculate the static attributes, from which the most appropriate one can be selected, based on the requirement of the task, i.e. higher accuracy or higher frame rate.

Our research is currently at the stage of gathering real-world data, which includes multi-camera scenes at crossroads and highways. After the data is collected and cleaned, the proposed methods will be thoroughly tested and evaluated.

Acknowledgment

The research was supported by the Institute of Transport Sciences (KTI) within the Innovative Mobility Program.

The research was supported by the Ministry of Innovation and Technology NRDI Office within the framework of the Autonomous Systems National Laboratory Program.

References

[1] Aquino, A. L., Cavalcante, T. S., Almeida, E. S., Frery, A. C., & Rosso, O. A. (2015). Characterization of vehicle behavior with information theory. The European Physical Journal B, 88(10), 1-12. DOI: 10.1140/epjb/e2015-60384-x

[2] Bewley, A., Ge, Z., Ott, L., Ramos, F., & Upcroft, B. (2016, September). Simple online and realtime tracking. In *2016 IEEE international conference on image processing (ICIP)* (pp. 3464-3468). IEEE. DOI: 10.1109/icip.2016.7533003

[3] Bochinski, E., Eiselein, V., & Sikora, T. (2017, August). High-speed tracking-by-detection without using image information. In *2017 14th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)* (pp. 1-6). IEEE.

[4] Deng, W., Zheng, L., Ye, Q., Kang, G., Yang, Y., & Jiao, J. (2018). Image-image domain adaptation with preserved self-similarity and domain-dissimilarity for person re-identification. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 994-1003). DOI: 10.1109/cvpr.2018.00110

[5] Farhadi, A., & Redmon, J. (2018, April). Yolov3: An incremental improvement. In Computer Vision and Pattern Recognition (pp. 1804-02767). Berlin/Heidelberg, Germany: Springer.

[6] Guo, Y., Sun, Q., Fu, R., & Wang, C. (2019). Improved car-following strategy based on merging behavior prediction of adjacent vehicle from naturalistic driving data. IEEE Access, 7, 44258-44268. DOI: 10.1109/access.2019.2908422

[7] He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 770-778). DOI: 10.1109/cvpr.2016.90

[8] Hsu, H. M., Huang, T. W., Wang, G., Cai, J., Lei, Z., & Hwang, J. N. (2019, June). Multi-Camera Tracking of Vehicles based on Deep Features Re-ID and Trajectory-Based Camera Link Models. In *CVPR Workshops* (pp. 416-424). doi: 10.1109/tip.2021.3078124

[9] Huynh, S. V. (2021). A Strong Baseline for Vehicle Re-Identification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 4147-4154). doi: 10.1109/cvprw53098.2021.00468

[10] Kalman, R. E. (1960). A new approach to linear filtering and prediction problems. doi: 10.1115/1.3662552

[11] Kanacı, A., Zhu, X., & Gong, S. (2018, October). Vehicle re-identification in context. In *German Conference on Pattern Recognition* (pp. 377-390). Springer, Cham. doi: 10.1007/978-3-030-12939-2_26

[12] Khosla, P., Teterwak, P., Wang, C., Sarna, A., Tian, Y., Isola, P., Maschinot, A., Liu, C. and Krishnan, D. (2020). Supervised contrastive learning. arXiv preprint arXiv:2004.11362.

[13] Kuhn, H. W. (1955). The Hungarian method for the assignment problem. *Naval research logistics quarterly,* 2(1-2), 83-97. doi: 10.1002/nav.3800020109

[14] Leonhardt, V., & Wanielik, G. (2018). Recognition of lane change intentions fusing features of driving situation, driver behavior, and vehicle movement by means of neural networks. In *Advanced Microsystems for Automotive Applications 2017* (pp. 59-69). Springer, Cham. doi: 10.1007/978-3-319-66972-4_6

[15] Ligthart, J. A., Ploeg, J., Semsar-Kazerooni, E., Fusco, M., & Nijmeijer, H. (2018). Safety analysis of a vehicle equipped with Cooperative Adaptive Cruise Control. IFAC-PapersOnLine, 51(9), 367-372. doi: 10.1016/j.ifacol.2018.07.060

[16] Lin, T. Y., Dollár, P., Girshick, R., He, K., Hariharan, B., & Belongie, S. (2017). Feature pyramid networks for object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 2117-2125).

[17] Liu, C., Zhang, Y., Luo, H., Tang, J., Chen, W., Xu, X., Wang, F., Li, H. and Shen, Y.D. (2021). City-scale multi-camera vehicle tracking guided by crossroad zones. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 4129-4137). doi: 10.1109/cvprw53098.2021.00466

[18] Liu, H., Tian, Y., Yang, Y., Pang, L., & Huang, T. (2016). Deep relative distance learning: Tell the difference between similar vehicles. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 2167-2175). doi: 10.1109/cvpr.2016.238

[19] Liu, X., Liu, W., Mei, T., & Ma, H. (2016, October). A deep learning-based approach to progressive vehicle re-identification for urban surveillance. In *European conference on computer vision* (pp. 869-884). Springer, Cham. doi: 10.1007/978-3-319-46475-6_53

[20] Lou, Y., Bai, Y., Liu, J., Wang, S., & Duan, L. (2019). Veri-wild: A large dataset and a new method for vehicle re-identification in the wild. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 3235-3243). doi: 10.1109/cvpr.2019.00335

[21] Luo, H., Chen, W., Xu, X., Gu, J., Zhang, Y., Liu, C., Jiang, Y., He, S., Wang, F. and Li, H. (2021). An empirical study of vehicle re-identification on the AI City Challenge. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 4095-4102). doi: 10.1109/cvprw53098.2021.00462

[22] Naphade, M., Wang, S., Anastasiu, D.C., Tang, Z., Chang, M.C., Yang, X., Yao, Y., Zheng, L., Chakraborty, P., Lopez, C.E. and Sharma, A. (2021). The 5th ai city challenge. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 4263-4273). doi: 10.1109/cvprw53098.2021.00482

[23] Pan, X., Luo, P., Shi, J., & Tang, X. (2018). Two at once: Enhancing learning and generalization capacities via ibn-net. In *Proceedings of the European Conference on Computer Vision (ECCV)* (pp. 464-479). doi: 10.1007/978-3-030-01225-0_29

[24] Peng, J., Wang, H., Zhao, T., & Fu, X. (2019, July). Cross domain knowledge transfer for unsupervised vehicle re-identification. In *2019 IEEE International Conference on Multimedia & Expo Workshops (ICMEW)* (pp. 453-458). IEEE. doi: 10.1109/icmew.2019.00084

[25] Schroff, F., Kalenichenko, D., & Philbin, J. (2015). Facenet: A unified embedding for face recognition and clustering. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 815-823). doi: 10.1109/cvpr.2015.7298682

[26] Song, H. S., Lu, S. N., Ma, X., Yang, Y., Liu, X. Q., & Zhang, P. (2014). Vehicle behavior analysis using target motion trajectories. *IEEE Transactions on Vehicular Technology*, 63(8), 3580-3591. doi: 10.1109/tvt.2014.2307958

[27] Sun, Y., Cheng, C., Zhang, Y., Zhang, C., Zheng, L., Wang, Z., & Wei, Y. (2020). Circle loss: A unified perspective of pair similarity optimization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 6398-6407). doi: 10.1109/cvpr42600.2020.00643

[28] Szűcs, G. (2009). Developing co-operative transport system and route planning. Transport, 24(1), 21-25. doi: 10.3846/1648-4142.2009.24.21-25

[29] Szűcs, G. (2011). Equilibrium estimation based on unreliable information in transport networks by adaptive simulation. International Journal of Advanced Intelligence Paradigms, 3(3-4), 273-285. doi: 10.1504/ijaip.2011.043431

[30] Szűcs, G. (2012). Route planning based on uncertain information in transport networks. Transport, 27(1), 79-85. doi: 10.3846/16484142.2012.667835

[31] Szűcs, G. (2015). Decision support for route search and optimum finding in transport networks under uncertainty. Journal of applied research and technology, 13(1), 125-134. doi: 10.1016/s1665-6423(15)30011-0

[32] Szűcs, G., & Sallai, G. (2009). Route planning with uncertain information using Dempster-Shafer theory. In 2*009 International Conference on Management and Service Science* (pp. 1-4). IEEE. doi: 10.1109/icmss.2009.5302815

[33] Tang, Z., Naphade, M., Liu, M. Y., Yang, X., Birchfield, S., Wang, S., ... & Hwang, J. N. (2019). Cityflow: A city-scale benchmark for multi-target multi-camera vehicle tracking and re-identification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 8797-8806). doi: 10.1109/cvpr.2019.00900

[34] Wang, Z., Zheng, L., Liu, Y., Li, Y., & Wang, S. (2020). Towards real-time multi-object tracking. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XI 16* (pp. 107-122). Springer International Publishing.

[35] Wen, L., Du, D., Cai, Z., Lei, Z., Chang, M.C., Qi, H., Lim, J., Yang, M.H. and Lyu, S. (2020). UA-DETRAC: A new benchmark and protocol for multi-object detection and tracking. *Computer Vision and Image Understanding*, 193, 102907. doi: 10.1016/j.cviu.2020.102907

[36] Wen, Y., Zhang, K., Li, Z., & Qiao, Y. (2016, October). A discriminative feature learning approach for deep face recognition. In *European conference on computer vision* (pp. 499-515). Springer, Cham.

[37] Wojke, N., Bewley, A., & Paulus, D. (2017, September). Simple online and realtime tracking with a deep association metric. In *2017 IEEE international conference on image processing (ICIP)* (pp. 3645-3649). IEEE. doi: 10.1109/icip.2017.8296962

[38] Xie, S., Girshick, R., Dollár, P., Tu, Z., & He, K. (2017). Aggregated residual transformations for deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 1492-1500). doi: 10.1109/cvpr.2017.634

[39] Yao, Y., Zheng, L., Yang, X., Naphade, M., & Gedeon, T. (2020). Simulating content consistent vehicle datasets with attribute descent. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part VI 16* (pp. 775-791). Springer International Publishing. doi: 10.1007/978-3-030-58539-6_46

[40] Ye, J., Yang, X., Kang, S., He, Y., Zhang, W., Huang, L., Jiang, M., Zhang, W., Shi, Y., Xia, M. and Tan, X. (2021). A robust MTMC tracking system for AI-City Challenge 2021. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 4044-4053). doi: 10.1109/cvprw53098.2021.00456

[41] Yu, F., Wang, D., Shelhamer, E., & Darrell, T. (2018). Deep layer aggregation. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 2403-2412). DOI: 10.1109/cvpr.2018.00255

[42] Zhang, Y., Wang, C., Wang, X., Zeng, W., & Liu, W. (2021). Fairmot: On the fairness of detection and re-identification in multiple object tracking. *International Journal of Computer Vision*, 1-19. DOI: 10.1007/s11263-021-01513-4

[43] Zhong, Z., Zheng, L., Cao, D., & Li, S. (2017). Re-ranking person re-identification with k-reciprocal encoding. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 1318-1327). DOI: 10.1109/cvpr.2017.389

[44] Zhou, K., Yang, Y., Qiao, Y., & Xiang, T. (2021). Domain generalization with mixstyle. arXiv preprint arXiv:2104.02008.

[45] Zhu, X., Luo, Z., Fu, P., & Ji, X. (2020). VOC-ReID: Vehicle re-identification based on vehicle-orientation-camera. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops* (pp. 602-603). DOI: 10.1109/cvprw50498.2020.00309

Dávid Papp was born in 1990 in Hungary and he has received MSc in Computer Science (at specialization of media informatics) from Budapest University of Technology and Economics (BME) in 2016. He started his PhD work in 2016 in the field of Computer Science at the same university. His research topic includes artificial intelligence, machine learning, computer vision as well as development of algorithms on these fields (e.g. query strategies for classification of visual contents with active learning). He was awarded twice with the scholarship of New National Excellence Program of the Ministry of Human Capacities, in 2018 and 2019.

Regő Borsodi was born in 1997 in Hungary. He is currently a MSc student at Budapest University of Technology and Economics (BME) in Computer Science.