Vehicle Re-Identification with the Space-Time Prior

Chih-Wei Wu, Chih-Ting Liu, Cheng-En Chiang, Wei-Chih Tu, Shao-Yi Chien
NTU IoX Center, National Taiwan University
{cwwu, jackieliu, cejiang, wctu, sychien}@media.ee.ntu.edu.tw

Abstract

Vehicle re-identification (Re-ID) is fundamentally challenging due to the difficulties in data labeling, visual domain mismatch between datasets and diverse appearance of the same vehicle. We propose the adaptive feature learning technique based on the space-time prior to address these issues. The idea is demonstrated effectively in both the human Re-ID and the vehicle Re-ID tasks. We train a vehicle feature extractor in a multi-task learning manner on three existing vehicle datasets and fine-tune the feature extractor with the adaptive feature learning technique on the target domain. We then develop a vehicle Re-ID system based on the learned vehicle feature extractor. Finally, our meticulous system design leads to the second place in the 2018 NVIDIA AI City Challenge Track 3.

1. Introduction

Vehicle re-identification (Re-ID) aims at tracking and identifying moving vehicles in multiple videos captured at multiple locations. The vehicle Re-ID task is essential to the envisioned IoT society to make our world safer and smarter.

The vehicle Re-ID problem is fundamentally challenging due to the following difficulties. First, collecting vehicle Re-ID datasets is difficult. It is infeasible to ask human labors to do vehicle Re-ID in the traffic videos. For instance, the traffic is very busy in the rush hours. The video quality and the viewing angle is also limited, useful information like license plates can not be reliably extracted from videos. With only raw videos, the human labors do not even know which specific cars to track. Second, it is hard to collect data for all kinds of environments and car models. Existing vehicle datasets [9, 10, 17, 15] were collected in particular urban areas. Usually the cityscape changes from city to city and the car models may also differ from time to time, making the existing vehicle datasets hard to adapt to new testing environments. Third, one vehicle may look very different under varying conditions such as object scale, camera viewing angle, vehicle pose, or environment light. It is challenging to associate such varying observations as the same vehicle. As such, it is not easy to model the vehicle Re-ID task as an end-to-end learning problem with these challenges.

In this paper, we present a vehicle Re-ID system, which is built upon a convolutional neural network (CNN) based vehicle feature extractor. To address the issues mentioned above, we also propose the Adaptive Feature Learning (AFL) technique to automatically harvest positive and negative training samples from unlabeled testing videos. We use the automatically harvested samples to fine-tune the feature extractor so that the deep network can adapt to the visual domain of the testing videos. This is made possible based on the fact that one vehicle can not appear at multiple locations at the same time and one vehicle moves continuously along the time. We call this the space-time prior and illustrate the idea in Figure 1. To verify the effectiveness of the AFL technique, we carry out experiments on the closely related human Re-ID task. Experimental results show that the AFL technique is able to improve the performance on existing human Re-ID datasets. We also report the results on the 2018 NVIDIA AI City Challenge [1] Track 3, which is the vehicle Re-ID problem given traffic videos recorded in the Bay Area.

We make the following contributions in this work:

- We propose the adaptive feature learning technique to alleviate the requirements of labeled videos in the testing environment. We also verify the effectiveness of the AFL technique on the human Re-ID datasets.
- We develop a vehicle Re-ID system. It takes raw traffic videos as input and performs vehicle detection, tracking and re-identification using the CNN features fine-tuned by the AFL technique.
- Our system design leads to the second place in the 2018 NVIDIA AI City Challenge Track 3.

2. Adaptive Feature Learning

The vehicle Re-ID problem is fundamentally challenging. First, one vehicle may appear drastically different at multiple time steps due to changes in scale, moving direction, occlusion, or environment light. Second, it is hard to

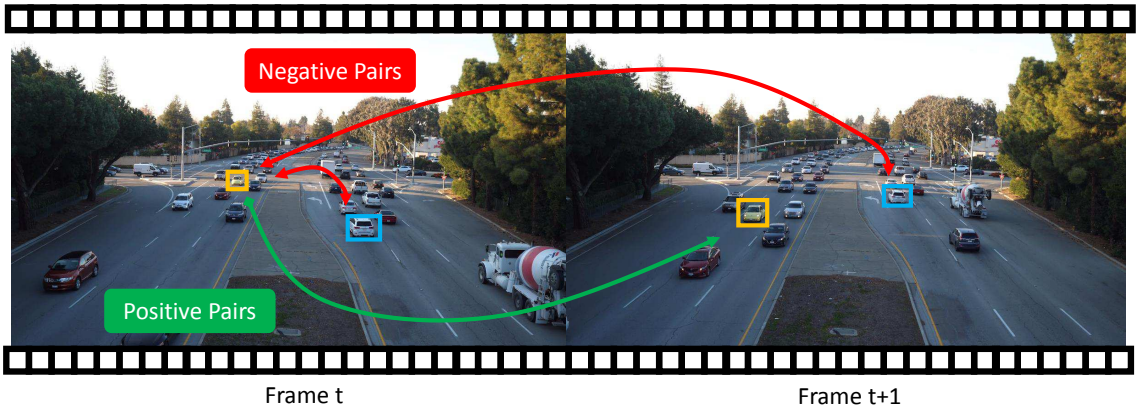


Figure 1: **The space-time prior in the traffic videos.** One vehicle can not appear at multiple locations at the same time, and it should move continuously along the time. We exploit this nature in traffic videos to harvest virtually infinite training samples and use the prior to enhance the results in the Re-ID task. Similar idea has been explored in the multi-face tracking [18].

label ground-truth data for traffic videos. There may be several cars with the same car model being captured in the same location. Other useful information like the license plates is not always available in the videos. Third, the visual domain of one dataset can be very different from another as the cityscape may change from city to city and the popular car models are varying from time to time. As a result, the features learned from one dataset may not be directly able to applied to another dataset. For instance, the AIC dataset provided by the 2018 NVIDIA AI City Challenge [1] only contains raw videos without any ground-truth labels. Figure 1 shows a sample high-way scene in the dataset. As we can see, severe scale change and environment light (*i.e.* shadow, sun light) pose more difficulties in the vehicle Re-ID task. Directly using features learned from existing vehicle datasets is not effective in such different visual domain.

To address these challenges, we introduce the adaptive feature learning technique to adapt the vehicle feature extractor pre-trained on existing datasets to the target domain (*i.e.* testing videos). In particular, we explore the space-time prior to harvest extra training samples from the target domain. As illustrated in Figure 1, we can sample any vehicle detection pairs from the same video frame as the negative samples. This is based on the fact that one vehicle can not appear at multiple locations at the same time, so the detection in the same time step must have different vehicle identities. On the other hand, one vehicle should move continuously along the time, so we can take the same vehicle object at different time steps as positive samples. The data mining scheme is fully unsupervised and can provide virtually infinite training data in the target domain. We use the harvested data in the target domain to fine-tune a pre-trained CNN-based vehicle feature extractor. In this way, the feature extractor is adapted to the visual domain of the testing videos and that is why we call it the adaptive feature learning technique.

3. Proposed Vehicle Re-ID System

The goal of the vehicle Re-ID system is to track and identify vehicles in multiple videos recorded at different locations or different time. In this paper, we design a three-stage pipeline to tackle this challenging problem. The first stage is the vehicle proposal. We leverage an off-the-shelf detector to locate as many vehicles as possible at this stage. In the second stage, we perform single-camera tracking to link vehicles across different time steps in a single video. Then we use a CNN-based feature extractor to extract features from the tracklets. Specifically, the network is pre-trained with existing vehicle datasets and fine-tuned using the AFL technique described in Sec. 2 on the testing AIC dataset [1]. In the last stage, we perform multi-camera matching to find corresponding tracklets across videos. Tracklets with similar CNN features are associated with the same identity. Figure 2 shows an overview of our vehicle Re-ID system. We describe more details regarding to each stage in the following subsections.

3.1. Vehicle Proposal

The vehicle proposal stage aims at locating the bounding boxes of all vehicles in a given video. To locate as many vehicles as possible, we take advantage of the state-of-the-art object detector, the Detectron [5, 4], which is publicly available for the research purpose. The Detectron has a good generalization ability to apply to different visual domains. Moreover, it also has favorable performance in de-

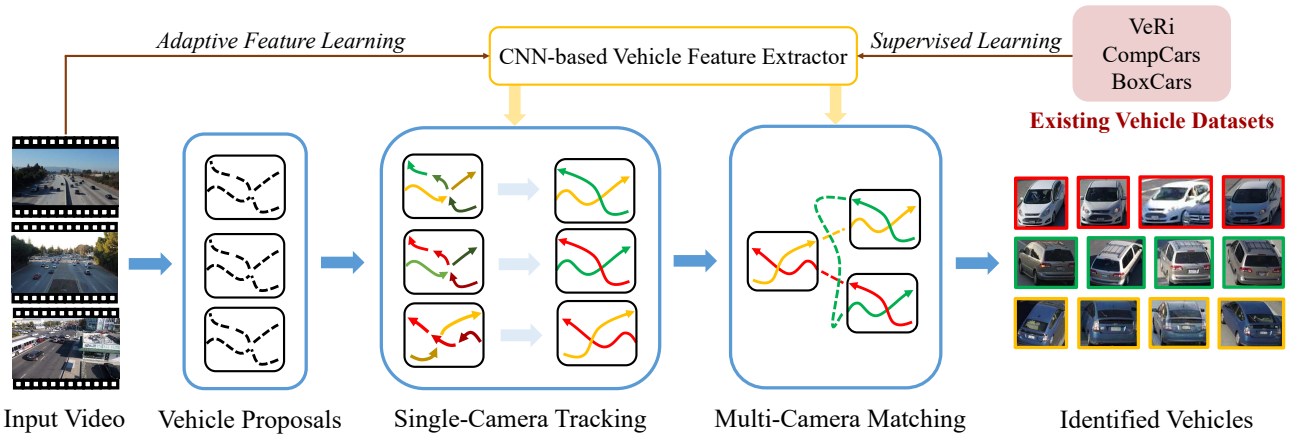


Figure 2: **An illustration of the proposed vehicle Re-ID pipeline.** The proposed system contains three stages. Given input videos, Vehicle Proposals propose vehicle detection bounding boxes. Next, the Single Camera Tracking stage links the detection with high overlaps into a tracklet in each video sequence. Meanwhile, the feature extracted from trained CNN is used to combine small tracklets into large tracklets. The last Multi-Camera Matching stage groups the tracklets across all sequences by their CNN features. Our vehicle Re-ID system can be easily applied to any other visual domain thanks to the core Adaptive Feature Learning (AFL) technique.

tecting small objects, which is a crucial part for the real-world urban scenes. We find it desirable to our need for the vehicle Re-ID task on the AIC dataset. The Detectron can be configured to meet different requirements. In particular, we configure the network backbone of the Detectron as the ResNet-101 [6] with the feature pyramid network structure [8].

We perform object detection using the Detectron on the AIC dataset and we notice it sometimes still results in unsatisfactory detection, such as bounding boxes with unreasonable aspect ratios, or huge bounding boxes with no meaningful objects inside. To address these issues, we empirically remove bounding boxes whose aspect ratio is larger than 5 or smaller than 1/5. We also filter bounding boxes with area bigger than 0.04 times of a full frame.

To understand the performance of our detection algorithm, we use the UA-DETRAC [16], a real-world multi-object detection and tracking dataset, as a performance indicator. Results show that the Detectron followed by simple outlier removal reaches 79.19% in terms of overall AP, achieving the second place on the leaderboard, and is only 0.66% away from the top ranked submission. As such, our detector provides abundant and accurate vehicle proposals for stages afterward in the Re-ID system.

3.2. Single-Camera Tracking

Owing to the large number of detected bounding boxes in all testing videos, it is impractical to extract features for every single detection and use these features to match different vehicle objects across time steps. Therefore, we first exploit a light-weight tracking algorithm, the IoU tracker [3], to reduce the processing elements from detection to tracklets. The IoU tracker grows the tracklets in a greedy manner. For the first video frame, each detected vehicle represents an independent tracklet. For every next frame, it computes the IoU (Intersection over Union) score of a new detection and the last detection of some tracklet in previous frame. If the IoU score of the two detection is greater than some threshold percentage, the new detection will be accepted to the tracklet. Otherwise, a new tracklet will be spawned with the new detection. After all frames are processed, it checks the length of tracklets and drop those whose number of detection is less than a given threshold. Thanks to the robustness of the Detectron-based detector, we can easily obtain high-quality tracklets even with the simple IoU tracker.

Though we can obtain satisfactory tracklets using the simple IoU tracker, we notice that such simple tracking strategy can not always associate one vehicle to the same ID. For some cases, the trajectory of one vehicle breaks into multiple tracklets. Therefore, we further compare the appearance features extracted from the tracklets and try to connect tracklets as they do not violate the space-time prior. Specifically, we sample some detection from these short tracklets and extract features for each detection based on a CNN feature extractor (Sec. 3.3). We find all physically plausible (*i.e.* spatially and temporally continuous) tracklet pairs and connect them if their appearance features are similar in terms of the Euclidean distance.

3.3. Vehicle Feature Extractor

In the vehicle Re-ID task, it is desired to have a robust feature extractor invariant to pose, brightness and viewing

angle. The CNN-based feature extractors have achieved state-of-the-art performance in a number of vision tasks, so we also construct our image-based feature extractor using a CNN. Inspired by recent development of the human Re-ID works [19, 7, 21], we choose ResNet-50 [6] architecture as the backbone of our feature extractor. We train the network in a supervised manner on three existing vehicle datasets, VeRi [9, 10], CompCars Surveillance [17] and BoxCars [15]. Table 1 summarizes the labels provided by different datasets.

The VeRi dataset is the only public dataset for the vehicle Re-ID task, as it provides ID labels for cars captured in all given images. We train the feature extractor using triplet loss as well as cross-entropy loss. For triplet loss, we use the batch-hard soft-margin triplet loss [7], which minimizes the maximum loss in a training batch. It also replaces the hinge function in triplet loss with a softplus function. This triplet loss variant has been shown effective on the human Re-ID task. On the other hand, we adopt cross-entropy loss to utilize the ID and color labels provided by VeRi dataset.

In addition to the VeRi dataset, we also train the feature extractor on the CompCars Surveillance and the BoxCars datasets. These two datasets are collected for the car model classification problem. We train the feature extractor to minimize the classification loss regarding to the car model. As we will show in the experimental results, compared to training only on the VeRi dataset using the vehicle ID information, we note that joint training with car model or color classification help the network learn discriminative features and benefit the vehicle Re-ID task.

Finally, the feature extractor is used in the testing videos of the AIC dataset. However, the AIC dataset possesses very different visual domain from these three training datasets, which limits the performance of the pre-trained feature extractor. We apply the AFL technique described in Sec. 2 to address this issue. Extra positive and negative samples are discovered in an unsupervised manner from the testing videos. These extra training samples are then used to fine-tune the pre-trained feature extractor so that the network can adapt to the visual domain of the testing videos. Specifically, we train the extra examples with batch-hard soft-margin triplet loss, which is the same triplet loss we use to pre-train the network on the VeRi dataset. The final feature for a vehicle proposal is then extracted. We average features within a vehicle tracklet to produce a fixed length representation for each tracklet in the single-camera tracking and multi-camera matching stages.

3.4. Multi-Camera Matching

After obtaining single-camera vehicle tracklets in Sec. 3.2, we perform multi-camera matching to group tracklets of the same vehicle identity in different videos together based on the extracted CNN feature. We have tried three kinds of matching schemes, which are K-Means clustering, K-nearest neighbor classification, and image retrieval query matching. All vehicles grouped into the same cluster are associated with the same identity. For the vehicle Re-ID challenge, we further filter the results and keep only those with vehicles passing through all four locations. Finally, we create a rank list of the candidate pool, which is sorted according to the cluster inertia. Now we provide detail of all the matching schemes.

Table 1: **A summary of dataset and label information.** A large scale labeled vehicle dataset is still demanded. Existing datasets only provide limited information.

Dataset	ID	Color	Car Model
VeRi [9, 10]	✓	✓	✗
CompCars Surveillance [17]	✗	✓	✓
BoxCars [15]	✗	✗	✓
AIC [1]	✗	✗	✗

K-Means Clustering. We first try the K-Means algorithm [11] to cluster tracklets into groups. However, it is difficult to directly cluster all tracklets due to the large number of data. Instead of running standard K-Means for all tracklets at the same time, we perform the Mini Batch K-Means algorithm [13] to speed up the clustering process. We simply call this variant K-Means in the following discussion.

Bottom-Up K-Means Clustering. We note that the visual domains can still be very different in the same dataset as these videos were captured from different locations. For example, there exist videos taken on a highway as well as videos taken at an intersection in the AIC dataset. It is still difficult to match features from different locations as the space-time constraint does not provide extra training samples across different locations.

Instead of clustering all tracklet data at the same time, we first run K-Means to cluster the data for each location separately. Then, we treat the cluster centers from each location as new data points and run K-Means again using these new data. This two-step algorithm guarantees that similar tracklets in the same location will be put together in the same cluster. It can also reduce the primitives to work on. We name this variant the Bottom-Up K-Means method.

K-Nearest Neighbor Classification. The Bottom-Up K-Means may still produce unsatisfactory results because clusters from the same location are usually similar than other locations, so it tend to put clusters from the same location together. This does not help identify vehicles passing through all locations.

We describe another alternative based on the K-nearest neighbor classification [2]. Similar to the Bottom-Up K-Means, we first run clustering in a location-wise manner. Then we use the cluster centers in one location to run the K-nearest neighbor classification for tracklets from other locations. We simply term this alternative the K-NN method. The K-NN method is better than Bottom-Up K-Means in the way that the initial cluster centers are guaranteed separated during the clustering. In practice, we choose the cluster centers in the location where the visual domain is the most different from others as the initial centers (Location 4 for the AIC dataset).

Image Retrieval Query Matching. A major drawback of the above mentioned methods is that they all perform hard assignment, which means one tracklet data can only be associated with one identity. This can easily cause irrecoverable error especially for those tracklets situated in the outskirt of its predicted cluster. Therefore, we relax the hard assignment constraint and allow multiple assignments for each tracklet. Inspired by query-gallery image retrieval evaluation in human Re-ID [20], we conduct the query matching algorithm, which we call the Query-Gallery in the following discussion.

We set the data at location 4 as query set, and construct gallery sets for every other locations following the strategy used in the K-NN variant. Furthermore, in consideration of the run time on such large data, we reduce query amount by grouping location 4's tracklets with K-Means preliminarily. For each query, we adopt K-reciprocal nearest neighbor matching [12] to find candidates in every location's gallery respectively. We then perform re-ranking, a common technique used in image retrieval, to improve the outcome of the query results. Specifically, we use a simplified version of Zhong *et al.* [22] to re-rank the candidates. This way, we are able to maximize the probability of retrieving the correct tracklets given a query.

4. Experiments

In this section, we provide experimental results to evaluate the effectiveness of the AFL technique as well as the proposed vehicle Re-ID system.

4.1. Effectiveness of the AFL Technique

As there is only one labeled vehicle Re-ID dataset, it is hard to verify the effectiveness of the AFL technique on the vehicle Re-ID task. As an alternative, we carry out experiments on the closely related human Re-ID task. We compare the performance with and without the AFL technique on the human Re-ID datasets.

Specifically, we use the Market-1501 [19] dataset as our labeled training domain, and use the DukeMTMC-reID [21] dataset, which provides suitable data for our AFL technique, for testing. We start with a ResNet-50 model pretrained on the ImageNet classification. We fine-tune the network on the Market-1501 dataset with the Adam optimizer as the baseline. Then we fine-tune the network again with extra data obtained by the AFL technique on the target DukeMTMC-reID dataset. The learning rate is set to 0.0001 for both fine-tuning stages.

Table 2: **The effectiveness of the AFL technique.** We compare the performance of the feature extractor on the human Re-ID task w/ and w/o the AFL technique on the testing DukeMTMC-reID [21] dataset. Feature extractors are first trained on the Market-1501 [19] dataset.

Method	mAP (%) ↑	Rank-1 (%) ↑
w/o AFL	13.46	25.99
w/ AFL	**14.20**	**28.50**

The results are shown in Table 2. The first row shows the performance of the model trained only on the Market-1501 dataset, while the second row is the model jointly trained on the Market-1501 dataset and extra data from the DukeMTMC dataset. The results show clear advantage of using the AFL technique in terms of the mean average precision (mAP) and the rank-1 accuracy. We believe the same improvement can be applied to the vehicle Re-ID task.

4.2. Vehicle Re-ID

Learning general vehicle features. To learn a general feature for such large data, we first perform supervised learning on three existing vehicle datasets, VeRi [9, 10], CompCars Surveillance [17] and BoxCars [15]. Each of them provides different kinds of vehicle attributes for training as summarized in Table 1. We combine car ID, color and model information to learn a general feature extractor. Aside from labeled data, we further adapt our feature extractor to the testing domain using the AFL technique.

We evaluate the feature extractor on the VeRi dataset. Table 3 shows the performance of our feature extractor compared with previous works on the VeRi dataset. By using the batch-hard soft-margin triplet loss as described in Sec. 2 to train the feature extractor (third row), we are able to outperform the state-of-the-art (first row) with the same network backbone and same training data. In addition, training with the additional CompCars, BoxCars, and the harvested AIC data further boosts the performance, which can directly compete with the state-of-the-art (second row). It is worth noting that the state-of-the-art (second row) utilizes extra temporal information, while ours only utilizes visual information. This indicates that training feature extractor with multiple tasks such as car color classification and car model

Table 3: **Feature extractor performance on the vehicle Re-ID task.** All feature extractors use the ResNet-50 as backbone. All methods are evaluated on the VeRi dataset [9, 10]. * indicates utilizing camera temporal information in addition to visual information. SOTA stands for state-of-the-art.

Method	mAP (%) ↑	Rank-1 (%) ↑	Rank-5 (%) ↑
SOTA CNN [14]	29.48	41.12	60.31
*SOTA [14]	**58.27**	83.49	90.04
Train on VeRi (Ours)	53.35	82.06	92.31
Train on all (Ours)	57.43	**86.29**	**94.39**

Table 4: **Submission results on the AIC Challenge Track 3.** We report the highest scores achieved by each multi-camera matching algorithm in Sec. 3.4. Our best result, based on the Query-Gallery scheme, ranks the second place on the final leaderboard.

Method	TDR	PR	S3
K-Means	0	0.0006	0.0003
Bottom-Up K-Means	0	0.0015	0.0007
K-NN	0.1429	**0.0020**	0.0725
Query-Gallery	**0.5714**	0.0007	**0.2861**

classification is beneficial to the Re-ID performance.

Submission results on the AIC challenge. Next, we test our vehicle Re-ID system on the AIC dataset. This dataset contains 15 videos captured at four sites in the Bay Area, providing nearly 15 hours of 1080p videos in total. The data amount is so large that all existing labeled training data combined is still less than the unlabeled testing data. The performance is evaluated by the 2018 NVIDIA AI City Challenge Track 3 [1] online submission system. The challenge requires our system to identify vehicles that travel through all four locations.

Owing to the submission rule of the challenge, we can only submit at most 100 identities. Therefore, we pick the first 100 clusters in our rank list as our final submission. We report our four algorithm results of the challenge in Table 4. The performance metrics are listed below:

- **Track detection rate (TDR)**: The ratio of correctly identified ground-truth vehicle tracks and the total number of ground-truth vehicle tracks. A vehicle track is correctly identified if the vehicle has been localized (IoU ≥ 0.5) and associated with the same object ID in at least 30% of the frames containing the ground-truth vehicle in a given video.
- **Localization precision (PR)**: The ratio of correctly localized bounding boxes and the total number of predicted boxes across all videos.
- **S3**: Mean of the TDR and PR scores.

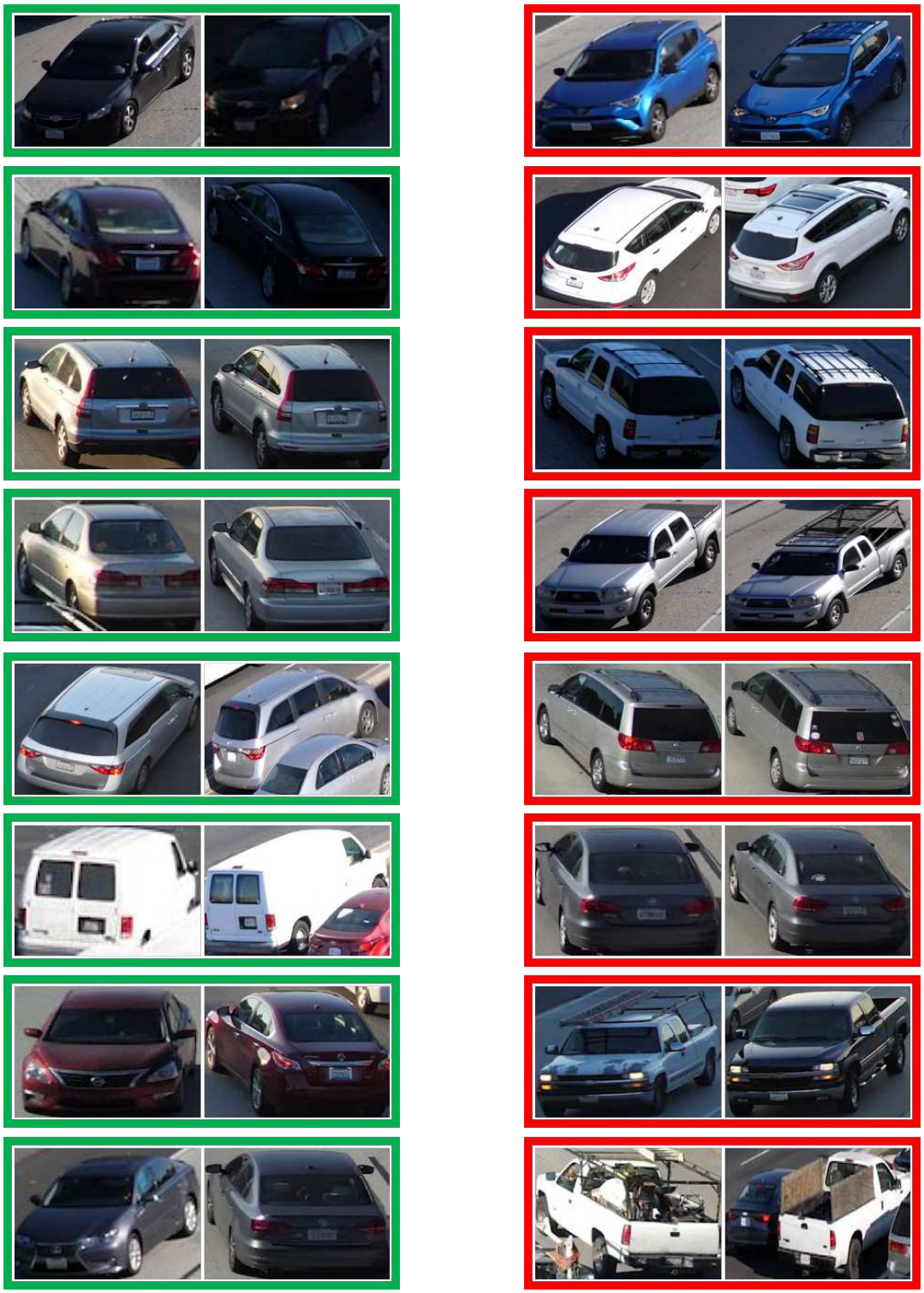
We record the highest S3 scores for the algorithms we described in Sec. 3.4 among the configurations we have tried. First, the K-Means method results in the lowest score. The Bottom-Up K-Means method improves the PR score a bit while it still fails to put corresponding tracklets together in the same cluster. We observe that the K-Means method and its variants tend to cluster tracklets with similar view angle together, which in turn fails to associate tracklets from different cameras together. This drawback is alleviated in the K-NN method by choosing cluster centers from uncommon view angles, guaranteeing these samples to evenly spread across all clusters. This desirable property leads to improvement of the TDR from 0 to 0.1429. Finally, the Query-Gallery scheme ensures to include samples from each camera location by constructing the gallery set for each of them, while maintaining the multiple assignment property to make up the assignment mistakes. In the end, the Query-Gallery method achieves 0.5714 in the TDR score, which achieves the highest S3 score among all methods we have tried. Our final result also achieves the second place in the 2018 NVIDIA AI City Challenge Track 3.

5. Discussion

Limitations of the space-time prior. While the space-time prior allows us to find extra training samples from the target domain, we notice there are still many cases not being discovered in the unsupervised sample mining process.

We manually pick some pairs of detection in our final clustering results. As shown in Figure 3, these sample pairs are organized in the 1×2 boxes. Each box represents a sample pair from one cluster. Figure 3a represents the eight pairs of samples that are successfully re-identified, while Figure 3b shows another eight pairs of samples that are mistakenly identified as the same vehicle. These failure cases are highly similar in appearance. As seen in Figure 3b, our model fails to distinguish pairwise vehicles with or without sunroof, with or without bracket, and with or without stickers. It is hard to discover negative samples with these subtle differences only by the space-time prior. A future direction is to learn the fine-grained feature extractor so that the network looks further into the detail.

Evaluation metrics. We notice that the evaluation protocol is way too biased to the TDR score, which makes the PR score contributes little to the final S3 score. We think this is because the number of ground-truth vehicles are too small, so that once one or two vehicles are correctly iden-



(a) Correct matches

(b) Incorrect matches

Figure 3: **Sample clustering results.** As shown, these sample pairs are organized in the 1×2 boxes. Figure 3a represents the eight pairs of samples that are successfully re-identified, while Figure 3b shows another eight pairs of samples that are mistakenly identified as the same vehicle.

tified, the TDR score is significantly increased. It can be resolved by annotating more ground-truth vehicles or adjusting the weights between the TDR and the PR scores. It would also be more meaningful to see the TDR scores w.r.t. varying detection rate.

Datasets. Compared with the human Re-ID task, there is still lack of labelled data for the vehicle Re-ID problem, making it difficult to analyze the source of performance gain from a large Re-ID system. One demanded future work is to collect diverse labeled data or develop unsupervised learning techniques for the vehicle Re-ID task. We also aim to systematically and quantitatively analyze each stage of our vehicle Re-ID system on a large scale labeled dataset in the future.

6. Conclusion

In this paper, we propose a vehicle Re-ID system. To address the lack of labeled training data and visual domain mismatch between datasets, we propose the adaptive feature learning technique based on the space-time prior to harvest virtually infinite training samples from the target videos. We verify the idea on the human Re-ID datasets and use the technique in the vehicle Re-ID system. We observe the same success of the AFL technique on the vehicle Re-ID dataset. Finally, our system achieves the second place in the 2018 NVIDIA AI City Challenge Track 3.

Acknowledgement. This research was supported in part by the Ministry of Science and Technology of Taiwan (MOST 107-2633-E-002-001), National Taiwan University, Intel Corporation, and Delta Electronics.

References

[1] https://www.aicitychallenge.org/, 2018. 1, 2, 4, 6

[2] N. S. Altman. An introduction to kernel and nearest-neighbor nonparametric regression. *The American Statistician*, 46(3):175–185, 1992. 5

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

[4] R. Girshick, I. Radosavovic, G. Gkioxari, P. Dollár, and K. He. Detectron. https://github.com/facebookresearch/detectron, 2018. 2

[5] K. He, G. Gkioxari, P. Dollár, and R. Girshick. Mask r-cnn. In *IEEE International Conference on Computer Vision (ICCV)*, 2017. 2

[6] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 3, 4

[7] A. Hermans, L. Beyer, and B. Leibe. In defense of the triplet loss for person re-identification. *arXiv*, abs/1703.07737, 2017. 4

[8] T.-Y. Lin, P. Dollár, R. Girshick, K. He, and Hariharan. Feature pyramid networks for object detection. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 3

[9] 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)*, 2016. 1, 4, 5, 6

[10] X. Liu, W. Liu, T. Mei, and H. Ma. A deep learning-based approach to progressive vehicle re-identification for urban surveillance. In *European Conference on Computer Vision (ECCV)*, 2016. 1, 4, 5, 6

[11] J. MacQueen et al. Some methods for classification and analysis of multivariate observations. In *Proceedings of 5th Berkeley Symposium on Mathematical Statistics and Probability*, 1967. 4

[12] D. Qin, S. Gammeter, L. Bossard, T. Quack, and L. Van Gool. Hello neighbor: Accurate object retrieval with k-reciprocal nearest neighbors. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2011. 5

[13] D. Sculley. Web-scale k-means clustering. In *ACM International Conference on World Wide Web*, 2010. 4

[14] 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 *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1900–1909, 2017. 6

[15] J. Sochor, J. pahel, 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. 1, 4, 5

[16] L. Wen, D. Du, Z. Cai, Z. Lei, M.-C. Chang, H. Qi, J. Lim, M.-H. Yang, and S. Lyu. Ua-detrac: A new benchmark and protocol for multi-object detection and tracking. *arXiv*, abs/1511.04136, 2015. 3

[17] L. Yang, P. Luo, C. C. Loy, and X. Tang. A large-scale car dataset for fine-grained categorization and verification. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015. 1, 4, 5

[18] S. Zhang, Y. Gong, J.-B. Huang, J. Lim, J. Wang, N. Ahuja, and M.-H. Yang. Tracking persons-of-interest via adaptive discriminative features. In *European Conference on Computer Vision (ECCV)*, 2016. 2

[19] L. Zheng, L. Shen, L. Tian, S. Wang, J. Wang, and Q. Tian. Scalable person re-identification: A benchmark. In *IEEE International Conference on Computer Vision (ICCV)*, 2015. 4, 5

[20] L. Zheng, Y. Yang, and A. G. Hauptmann. Person re-identification: Past, present and future. *arXiv*, 2016. 5

[21] Z. Zheng, L. Zheng, and Y. Yang. Unlabeled samples generated by GAN improve the person re-identification baseline in vitro. *CoRR*, abs/1701.07717, 2017. 4, 5

[22] Z. Zhong, L. Zheng, D. Cao, and S. Li. Re-ranking person re-identification with k-reciprocal encoding. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 5