# Dual Embedding Expansion for Vehicle Re-identification

*Clint Sebastian[1,2], *Raffaele Imbriaco[1], Egor Bondarev[1], Peter H.N. de With[1,2]
[1]VCA Group, Eindhoven University of Technology [2]Cyclomedia B.V
* *equal contribution*
{c.sebastian, r.imbriaco}@tue.nl

## Abstract

*Vehicle re-identification plays a crucial role in the management of transportation infrastructure and traffic flow. However, this is a challenging task due to the large view-point variations in appearance, environmental and instance-related factors. Modern systems deploy CNNs to produce unique representations from the images of each vehicle instance. Most work focuses on leveraging new losses and network architectures to improve the descriptiveness of these representations. In contrast, our work concentrates on re-ranking and embedding expansion techniques. We propose an efficient approach for combining the outputs of multiple models at various scales while exploiting tracklet and neighbor information, called dual embedding expansion (DEx). Additionally, a comparative study of several common image retrieval techniques is presented in the context of vehicle re-ID. Our system yields competitive performance in the 2020 NVIDIA AI City Challenge with promising results. We demonstrate that DEx when combined with other re-ranking techniques, can produce an even larger gain without any additional attribute labels or manual supervision.*

Figure 1: Qualitative comparison of top-$k$ ranks on the retrieved results (row 1, 3) and after applying the re-ranking schemes (row 2, 4).

## 1. Introduction

Large-scale traffic video analysis can enable efficient management of transportation infrastructure and traffic flow. With the availability of a large number of sensors, an Intelligent Transportation System (ITS) can be developed to facilitate AI-powered smart cities. It is beneficial to periodically recognize a vehicle across different locations and cameras to estimate the traffic flow. This concept is known as vehicle re-identification (re-ID), where the objective is to match a specific vehicle irrespective of location, time or camera view. In essence, vehicle re-ID is a constrained image retrieval task. Given a query image, we rank all the database images based on their similarity to the query. Image retrieval systems perform two fundamental processes: retrieval, and re-ranking. First, a feature extractor produces a compact representation of the image to facilitate retrieval. Then a similarity score is computed for each representation. Second, a re-ranking technique is applied to indicate the relevance of the retrieved results.

Similar to other re-ID tasks such as person re-ID, vehicle re-ID also suffers from occlusion and low-quality images. However, vehicle re-ID is more challenging, since vehicles have low intra-class variations. For example, the same model may have only slight appearance changes (rims, lights) depending on their make. More drastic appearance variations such as glare can occur due to weather conditions, surroundings or camera position. Furthermore, a human body can be separated into a few semantic parts, while the outcome of separation of a vehicle into parts depends on the point of view. The front is symmetric with eas-

ily identifiable elements, such as the headlights, grille, and bumper, whereas the side is asymmetric and contains other parts like rims and doors. The combination of these factors makes the vehicle re-ID a more challenging task. For vehicle re-ID, license plates can be exploited to improve performance [36]. However, they are not always available at both sides or not visible, due to poor lighting conditions and occlusions. Apart from these issues, the usage could raise privacy concerns [38]. Besides license plates, other attributes can be generated to improve performance such as color, brand and vehicle type [33, 5]. In this research, we focus on improving performance by producing a descriptive representation using nearest neighbors. Therefore, we propose a novel dual embedding expansion method offering promising results on the 2020 NVIDIA AI City Challenge. Our contributions are summarized as follows.

- We propose an efficient embedding expansion strategy across CNN models and image scales that improves performance. The improved performance comes without any additional overhead during online retrieval.
- We present an effective way to jointly use both tracklets and $k$-nearest neighbors ($k$-NN) of a query to enrich the embedding representation.
- We provide a comparative study of popular re-ranking techniques from landmark image retrieval on the CityFlow vehicle re-identification dataset.

## 2. Related Work

**Feature extraction.** In recent years, vehicle re-ID has received large interest from the research community due to advances in deep learning. This has resulted in the publication of larger and more challenging datasets, such as CityFlow [35], VeRi-776 [22], VehicleID [20], and VeRi-Wild [24]. Conventional feature extraction for image retrieval has been performed using hand-engineered descriptors e.g. SIFT [25], SURF [3] or HoG [6]. However, modern approaches deploy Convolutional Neural Networks (CNNs) as feature extractors. A major common focus has been on improving the descriptive capabilities of CNNs. Liu *et al.* [20] propose to progressively refine the retrieval results using a combination of hand-engineered features (color, texture) together with global descriptors. Similarly, in [34], features are aggregated with Bag-of-Words [30] and are fused with CNN descriptors.

Popular choices for CNN-based architectures for re-ID are generally part-based. Part-based models [23, 4, 2, 5] split the output of the network into several regions that learn part-specific features. These features are then merged into a single representation that is used as a final embedding for retrieval. Attentive models in [33, 13, 17, 16, 37, 46] train specialized modules to detect salient regions and improve retrieval performance. Attention modules are adapted for saliency in the spatial, channel, or temporal domain. Typically, CNNs are trained with a classification loss. However, metric learning methods such as triplet, contrastive, or center losses are almost ubiquitous in re-ID literature [23, 42]. These are employed to improve performance by minimizing the distance between embeddings directly. Other angular losses, such as CosFace [40], ArcFace [7], and SphereFace [21], are also applied for re-ID tasks.

Other approaches enrich the labels by adding additional information. Yan *et al.* in [45] add brand, make and color to dataset instances and train a network with these extra attributes. This strategy is also adopted by in [33]. Fine-grained labeling [9, 47] with individual semantic parts such as 'logo' and 'windshield' are labeled and detected to improve the re-identification performance. In [41], semantically relevant key points are annotated, instead of fine-grained attribute labels. A key disadvantage of these approaches is that this requires additional labeling. When additional data from a different domain is available, few approaches also apply domain adaptation strategies that are beneficial to improve performance[19].

**Re-ranking.** After obtaining the initial set of ranks from retrieval, a re-ranking step is usually utilized to refine the rankings. In vehicle re-ID [13, 29, 15, 5, 4, 48, 19], a common re-ranking technique is *k*-reciprocal [49]. It checks the *k* nearest-neighbors of the query and their reciprocals in the database. The query representation is expanded by selecting top matches of the retrieved sets using the Jaccard distance.

A modification to $k$-reciprocal is presented in [12]. They employ the Mahalanobis distance when computing the similarity of the query and database images. Furthermore, they propose to re-organize the ranked list according to the vehicle tracklets. A similar technique is explored in [2], with re-ranking by direct computation of the query similarity against the averaged tracklet representation. Huang *et al.* [13] presents a novel re-ranking technique based on instance metadata. This technique trains a network to learn specific attributes like vehicle type, brand, color and uses the predictions to remove irrelevant matches. Afterwards, $k$-reciprocal is employed to improve the ranking list. Similarly, several other metadata-based constraints, such as speed, time, and location are exploited to improve vehicle re-ID performance [5, 33].

Most of the previously discussed work uses the $k$-reciprocal encoding or metadata constraints for re-ranking. However, techniques common in image retrieval literature, such as query expansion and database augmentation, are largely absent. In [27], re-ranking is performed using spatial verification by handcrafted local descriptors and are aggregated using Bag-of-Words [30]. Consistency across matches is enforced using visual words, and the local

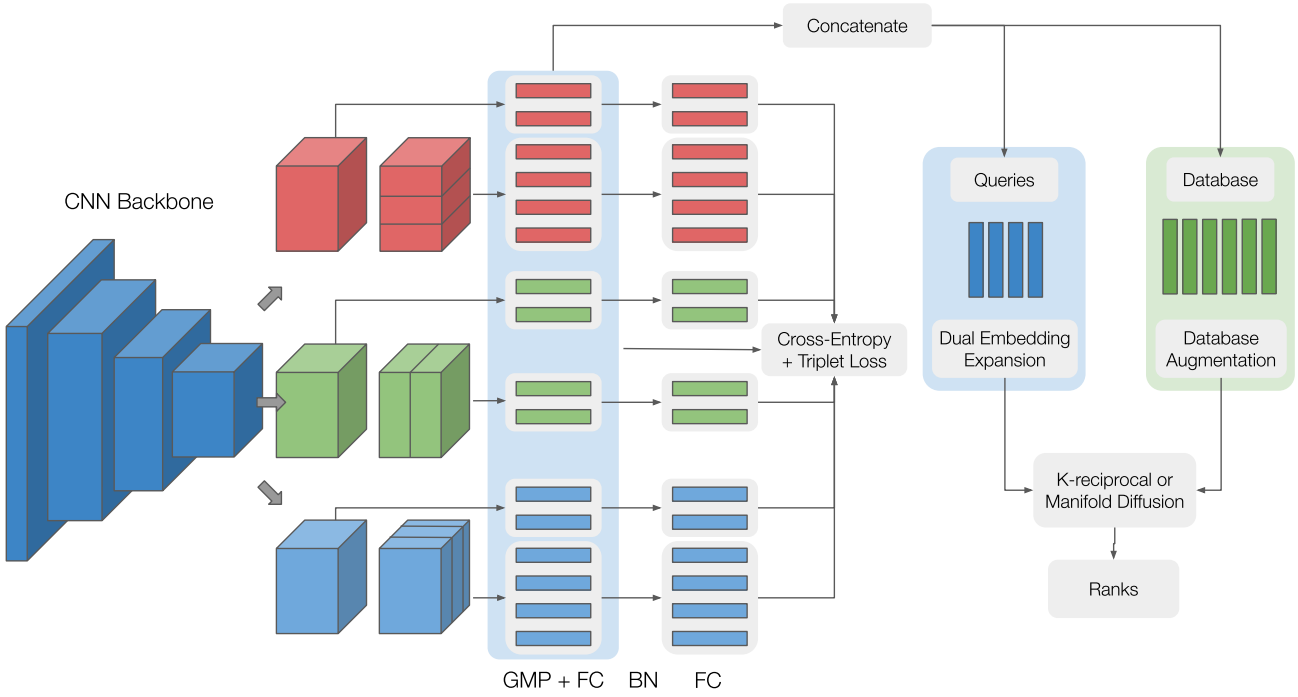


Figure 2: Overview of the feature extraction network and proposed re-ranking strategy. The features are extracted using a CNN backbone with a part-based model, similar to the Partition and Reunion Network [4]. Each block represents partitions across height, channel, and width. All the outputs after global max-pooling (GMP) and fully connected layers are fed to triplet loss, whereas additional classification heads are applied with cross-entropy loss. Note that for simplicity, both losses are shown in a single block in the figure. The final output representations are extracted for both query and database/gallery images and then our dual embedding expansion is applied, followed by $k$-reciprocal re-ranking.

descriptors provide geometric constraints. Alternatively, in [48], the average query expansion is applied to enhance query representation, and database features are clustered in groups to the closest clustered centroid. In our work, we attempt to bridge the gap between vehicle re-ID and image retrieval, by studying the impact of popular re-ranking methods from image retrieval and proposing a novel dual embedding expansion technique that is suited for vehicle re-ID.

## 3. Methods

The proposed approach primarily consists of two parts, a CNN-based feature extractor for retrieval and post-processing using re-ranking techniques.

### 3.1. Feature extractor

Image embeddings are generated using a part-based model similar to the Partition and Reunion Network [4] (PRN). Our model is composed of a convolutional backbone and three branches. Each branch encodes global information and partitions the height, width, and channel to obtain local information. Each branch does not share weights with the others, and partitions are mutually exclusive. However, there are minor differences with the PRN. We generate three horizontal or vertical partitions across the spatial dimensions and two partitions across the channel dimensions. The output of each branch is pooled using global max pooling, followed by a fully connected layer to reduce dimensionality to 256. This is followed by a BNNeck [26] and a classification head for each branch. We use a part-based network, due to its good performance for re-ID tasks. The number of partitions are reduced to avoid the design of excessively large descriptors. During test time, the outputs prior to the last convolutional layers are concatenated to obtain a single representation that encodes both global and local features. The resulting descriptor dimensions are $256 \times 17$. For the final submission, we use the ResNet50 [10], ResNet50 IBN-a [28], SE-ResNeXt50 [44] and EfficientNet [32] architectures as the CNN backbones. The backbone selection and part-based model are motivated by their high performance

in other retrieval, re-ID and classification tasks [12, 4, 39].

**Loss functions** For training, we use a combination of the cross-entropy loss with label smoothing regularization and the triplet loss with batch-hard negative mining. Cross entropy with label smoothing is given as

$$\mathcal{L}_{id} = \sum_{i}^{N} t_i \cdot \log(y_i), \tag{1}$$

where $y_i$ is the output for identity $i$ and label smoothing is applied to produce $t_i$

$$t_i = \begin{cases} 1 - \frac{N-1}{N} \cdot \epsilon, \text{if } \; i = j \\ \frac{\epsilon}{N}, \text{otherwise,} \end{cases} \tag{2}$$

where $\epsilon$ is the regularization term (set to 0.1) and $N$, the number of classes. The triplet loss is given as

$$\mathcal{L}_{triplet} = \sum_{a,p,n} \max[D_{a,p} - D_{a,n} + m, 0], \tag{3}$$

where $D_{a,p}$, $D_{a,n}$ are the distance between anchor ($a$) with positive ($p$) and negative ($n$), respectively. The triplet margin is $m$, the positive samples share the same class as the anchor and negatives are elements from different classes. A key difference compared to previous approaches is that we apply triplet loss to each of the outputs of a fully connected layer, instead of a few selected outputs. We observe faster convergence and slightly improved performance with this approach. An overview of the system is shown in Figure 2.

### 3.2. Re-ranking techniques

For re-ranking, a few techniques are considered. After inference, each embedding from different networks has the same size of 256 $\times$ 17 dimensions. For similarity between the representations, cosine similarity is applied. We have selected the re-ranking techniques by their prevalence in image retrieval and re-ID literature.

**Dual embedding expansion.** The dual embedding expansion (DEx) comprises of two parts. The first part combines representations across models and image scales in an efficient manner. Typically, most ensemble strategies in re-ID or retrieval concatenate the representations, which results in very large embeddings. Although this improves performance, it is computationally expensive during the similarity computation. Similar to different models capturing distinct features of the same image, image scales also capture intricate details of the same image. Given an image *I* and a scale $s \in \mathcal{S}$, we pass each image *I* at scale $s$ to the model $m \in \mathcal{M}$. Therefore, we propose to expand the embedding to produce features $f_{ms}$ by computing

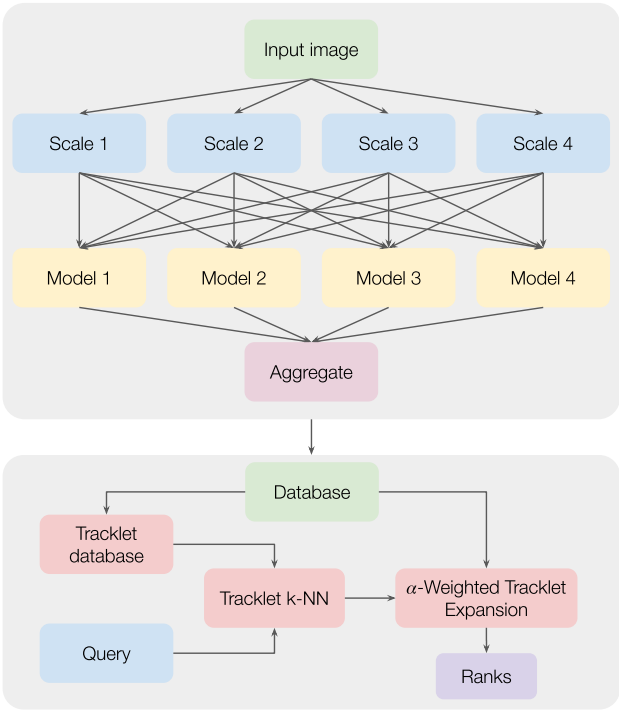


Figure 3: Dual embedding expansion: The input image is processed at multiple scales and fed to all the models in a fully connected manner. The output representations are aggregated to produce an enriched embedding and are fed to tracklet-based query expansion to enhance the representation.

$$f_{ms} = \sum_{m_i \in \mathcal{M}} \sum_{s_j \in \mathcal{S}} m_i(s_j(I)). \tag{4}$$

The final embedding $f_{ms}$ is the $L_2$ normalized after averaging features from different models at multiple scales. The second part leverages both tracklet information and the $k$-NN to improve the representations. Given a query and the gallery of images, we extract $f_{ms}$ for each, denoted as $Q_{ms}$ and $G_{ms}$, respectively. Since most re-ranking techniques are sensitive to the initial set of rankings, we utilize the tracklet information to construct a new aggregated gallery $T_{ms}$. The new tracklet gallery $T_{ms}$ is constructed by averaging all the embeddings in a given tracklet. We compute the cosine similarity between $Q_{ms}$ and $T_{ms}$ to obtain the top tracklets for each query. The original elements of the tracklets are placed back at the tracklet ranks to obtain a new set of ranks. Following this, we apply $\alpha$ query expansion by utilizing the new ranks generated from the tracklet information. Therefore, for a given track $t$, we compute the newly sorted gallery $g_t$ and the renewed query $\hat{q}_i$ by

$$g_t = k\text{-NN}_t(q_i, g_i), \tag{5}$$

$$\hat{q}_i = q_i + \sum_k g_t \times \cos(q_i, g_t)^\alpha, \tag{6}$$

where $q_i \in Q_{ms}$ and $g_i \in G_{ms}$. The $k$-NN$_t$ and "cos" refer to the $k$-nearest neighbors obtained through the tracklet information and cosine similarity, respectively. The $g_t$ is the sorted gallery based on the tracklets for a query $q_i$. The parameters $k$ and $\alpha$ denote the number of top-$k$ matches and power to weight the distance, respectively. A key advantage of the method is that the new representation retains the same size as the single-model, single-scale representation. Compared to other ensemble strategies such as concatenation, this reduces online computation costs while calculating similarity. The similarity computation cost is the same as a single-model, single-scale representation.

**Database-side feature augmentation.** We also perform database-side feature augmentation (DBA) by constructing a $k$-NN graph of the database to aggregate features. Instead of relying on tracklet information, only nearest neighbors are utilized. This allows to control the number of aggregated gallery images and ensures that all embeddings are expanded using the same amount of data. DBA commonly increases performance in landmark retrieval systems when there are occlusions or visibility constraints [1].

**Re-ranking with $k$-reciprocal encoding.** In $k$-reciprocal encoding, a robust set of matches is constructed given an initial ranking. Given the $k$-NN neighbours of query $q_i$, the $k$-reciprocals of $q_i$ in $\mathcal{R}(q_i, k)$ are defined in [49] as

$$\mathcal{R}(q_i, k) = \{g_i | (g_i \in k\text{-NN}(q_i)) \wedge (q_i \in k\text{-NN}(g_i))\}. \tag{7}$$

In order to include positive images that may have been left out of the top-$k$ ranks, the new set $\mathcal{R}^*(q_i, k)$ is constructed by iteratively adding the half of the $k$-reciprocal images. These images are only added if the following condition holds

$$|\mathcal{R}(q_i, k) \cap \mathcal{R}(g_i, \frac{1}{2}k)| \geq \frac{2}{3}|\mathcal{R}(g_i, \frac{1}{2}k)|, \tag{8}$$

which avoids the addition of too many negative samples to the re-ranked set. Afterwards, the Jaccard distance between the query and gallery reciprocal sets is computed, based on the assumption that images with similar reciprocal sets are closely related. The final representation is a weighted linear combination of the original distance and the Jaccard distance, using a weighting parameter $\lambda$.

**Diffusion.** In contrast to previous methods, diffusion considers the complete dataset manifold and propagates image similarities through the adjacency matrix [8, 14]. In diffusion, random walks through the similarity graph are computed, to spread the query similarity across the full adjacency graph. The results from the iterative computation of these random walks is a ranking matrix that accounts for the intrinsic structure of the dataset manifold. The affinity matrix $A$ is symmetrically normalized by

$$S = D^{-1/2} A D^{-1/2} \tag{9}$$

with $D = \text{diag}(A\,\mathbf{1}_n)$ where $\mathbf{1}_n$ is the unity vector of size $n$. We follow the iterative procedure in [14]. For a given initial $\mathbf{f}^0$ vector, we use an iterative scheme for diffusion given by,

$$\mathbf{f}^t = \alpha\, S\, \mathbf{f}^{t-1} + (1-\alpha)\, \mathbf{y}, \tag{10}$$

where $\mathbf{f}^t \in \mathbb{R}^d$, $S$ is the transition matrix, $\mathbf{y}$ an $\ell^1$ vector. The parameter $\alpha$ regulates the spread of the manifold structure of the affinity matrix to the query points. Parameter $t$ is the iteration count of the solver. Besides $\alpha$ and $t$, diffusion employs two additional parameters. The parameters $k, k_q$, are the local constraints of the affinity and similarity matrices to remove noise. A monomial kernel is used as in [14]

# 4. Experiments

## 4.1. Dataset and metrics

The CityFlow-ReID dataset consists of 56,277 images of 666 different vehicle classes. The dataset is divided into two parts, a training set of 36,935 images of 333 vehicle classes from 1,897 video tracks, and a test set of 18,920 images of 333 vehicle classes from 798 video tracks. The remaining 1,052 images of the test split are used as the query or probe set. For validation, we leverage the training set by holding back a 100 identities as a validation set, while training on the 233 classes. For reporting, the mean Average Precision (mAP) and the Top-1 and Top-5 accuracy of the cumulative match curve (CMC@1, CMC@5) are used.

## 4.2. Implementation details

We have trained eight different CNN backbones, ResNet50 (+CBAM [43]), ResNet50-IBN-a (+CBAM), SE-ResNeXt50 and EfficientNet-B1, B2, B3 with the Adam optimizer [18]. In our experiments, we have found that larger EfficientNet and ResNet-based models are lowering the performance. Hence, the experiments are restricted to the smaller variants. The learning rate is set to $2 \times 10^{-4}$ and decays at an exponential rate with exponent 0.01 at each epoch. Triplet and cross-entropy losses are employed and the models are trained for 500 epochs. Cross-entropy and triplet losses are weighted by a factor of 2 and 1, respectively. Label smoothing [31] is utilized for the classification loss with $\epsilon = 0.1$. Image triplets are obtained using a batch-hard mining strategy as in [11]. Per batch, 8 classes are

sampled containing between 4 to 8 images each, depending on the CNN backbone. The images are resized to $288 \times 288$ pixels, and augmentations such as horizontal flipping, contrast, and random erasing [50], are applied.

We deploy both dual embedding expansion and database augmentation and use the closest 20 and 10 neighbors, respectively. The value of $\alpha = 2$. This is followed by $k$-reciprocal encoding and its parameters are set to $\lambda = 0.5$, $k_1 = 60$ and $k_2 = 30$.

### 4.3. Ablation studies

The ablations studies are provided over the baseline network (ResNet50-IBN-a). First, the benefits of adding additional synthetic data during training are explored. A total of 193 synthetic vehicle IDs are randomly chosen and added to the training set. The IDs are added such that a relatively balanced ratio between synthetic and real images is maintained. Second, the impact of different re-ranking schemes are studied which involves query expansion, database side augmentation, diffusion and the $k$-reciprocal encoding.

**Impact of additional synthetic data.** The fourth edition of the AI City Challenge includes, in addition to the train, test and query image sets, a collection of synthetic vehicle images. We study how the addition of synthetic data impacts the performance of our baseline network ResNet50-IBN-a. Three distinct scenarios are considered. First, training with real images only. Second, learning using a balanced number of synthetic and real images. Third, we train by merging both image sets. Note that to ensure a fair comparison, the test set in our split is composed only by real images. The results from these experiments are shown in Table 1.

For the second scenario, there is a data imbalance issue. We try to balance the number of real to synthetic images in a 1:1 ratio. However, in our experiments, this ratio is only an approximate due to the variable number of images per tracklet. In total, we consider 426 unique IDs, out of which 233 are from the real dataset. For the final submission, we consider a total of 526 classes (including 333 real) for training the models. It can be observed that balancing the number of real and synthetic samples provides better performance than using only real data. The balanced inclusion of synthetic images increases the baseline mAP by roughly 4.3%, whereas training with all the synthetic samples only yields a minor gain. However, the CMC metric reduces slightly with the inclusion of synthetic data. We conjecture that the addition of the synthetic images acts as a form of regularization and prevents the network from overfitting. Nevertheless, the addition of too many samples from the synthetic domain may be changing the underlying dataset statistics, thereby reducing the quality of the learned embeddings. Therefore, further experiments are using the 'Balanced' training set.

| Training set | mAP | CMC@1 | CMC@5 |
|---|---|---|---|
| Real only | 68.0 | **89.7** | **92.3** |
| Balanced | **72.4** | 87.9 | 91.4 |
| Full | 68.1 | 87.7 | 91.6 |

Table 1: Performance of ResNet50-IBN-a with different real to synthetic data ratios. Balanced refers to a 1:1 ratio.

**Impact of dual embedding expansion.** The dual embedding expansion has two components, a model and scale component and a tracklet-based query expansion. For a single model, we provide the impact of different scales in Table 2. Complementary features from different scales improve the performance without additional online similarity computation costs. As expected, the scale that is used to train offers the best performance. Any other scale reduces the performance. However, the aggregation of all the scales results in the highest gains.

| Scale | mAP | CMC@1 | CMC@5 |
|---|---|---|---|
| 0.9 | 69.3 | 87.2 | 90.4 |
| 1 | 72.4 | 87.9 | 91.4 |
| 1.1 | 71.6 | 87.7 | 91.5 |
| 1.2 | 70.8 | 88.2 | 91.4 |
| All | **73.8** | **89.0** | **92.0** |

Table 2: Impact of the image scale on re-ID performance.

When combined with $\alpha$-weighted tracklet query expansion, the performance improves further. We compare the proposed DEx with other popular query expansion methods in 3. For fair comparisons, single-scale DEx is also compared. DEx consistently outperforms other query expansion techniques. As shown in Table 3, higher values for $k$ yields higher performance. In the final submission, we have set $k$=20 for DEx to prevent over-tuning on the validation set.

| *k* | AQE | $\alpha$QE | DEx-*ss* | DEx-*ms* |
|---|---|---|---|---|
| 5 | 74.6 | 74.5 | 76.6 | 77.7 |
| 10 | 75.4 | 75.3 | 77.1 | 78.0 |
| 20 | 76.6 | 76.6 | 77.6 | 78.4 |
| 40 | 77.1 | 77.2 | 77.9 | 78.5 |

Table 3: Comparison of retrieval performance (mAP) with different query expansion techniques. DEx-*ss* and DEx-*ms* are dual embedding expansion with single and multi-scale features.

**Impact of diffusion.** Table. 4 contains the results from our experiments with diffusion. We observe that large values of $k$ or $k_q$ have a detrimental effect on retrieval perfor-

mance. We set $\alpha = 0.95$ and run for up to 25 iterations. Diffusion provides a gain of 8-10 mAP over baselines depending on the choice of $k$ and $k_q$.

| $k$ | $k_q$ | mAP | CMC@1 | CMC@5 |
|---|---|---|---|---|
| 25 | 25 | 80.1 | 90.5 | 92.0 |
| 50 | 25 | 82.2 | **92.2** | 93.6 |
| 100 | 25 | 81.4 | 91.4 | **93.8** |
| 25 | 50 | 80.6 | 90.7 | 91.9 |
| 50 | 50 | **82.6** | 92.1 | 93.7 |
| 100 | 50 | 81.4 | 91.4 | 93.8 |

Table 4: Retrieval performance with diffusion for various parameter settings.

**Impact of database augmentation.** Previous work [4] has also explored a similar technique by encoding the entire database tracklets as the average representation of its images. Whereas this limits the possibility of polluting the database representations with those of negative IDs, it prevents the expansion with images belonging to other tracklets. In our work, we compute the $k$-NN of each database descriptor and aggregate the top-$k$ matches as the new image representation. This provides additional flexibility

| $k$ | mAP | CMC@1 | CMC@5 |
|---|---|---|---|
| 5 | 76.0 | 86.7 | 91.2 |
| 10 | 76.9 | 86.1 | 91.9 |
| 20 | 78.1 | **87.0** | **92.0** |
| 40 | **79.2** | 86.7 | 91.7 |

Table 5: Retrieval performance with DBA for various values of $k$.

when re-ranking, since the value of $k$ can be tuned for the best performance. These results are presented in Table 5. In general, increasing the number of neighbors aggregated into the database representation has a beneficial effect on the mAP metric. However, the CMC metric shows a slight deterioration. This indicates that some erroneous matches are being placed at the top-1 and top-5 ranks with DBA re-ranking, but correct matches are being ranked at adjacent positions. It can be observed that the highest overall mAP is obtained when $k = 40$. Nevertheless, setting $k$ to this value lowers the performance on the private test set. This may occur if the test database has many visually similar instances of different vehicles. In this case, using too large values of $k$ reduces the discriminating power of the descriptors, so that $k = 10$ is adopted in the final submission.

**Impact of re-ranking schemes.** We also present a comparison of the studied re-ranking techniques in Table 6. Among these, the single best performing one is Diffusion, yielding an increase in mAP over the baseline of roughly

| Re-ranking scheme | mAP | CMC@1 | CMC@5 |
|---|---|---|---|
| Baseline | 72.4 | 87.9 | 91.4 |
| +DEx | 78.4 | **93.3** | **94.7** |
| +DBA | 76.9 | 86.1 | 91.9 |
| +$k$-reciprocal | 80.4 | 91.8 | 93.9 |
| + Diffusion | 82.2 | 92.2 | 93.6 |
| +Tracklet | 76.7 | 87.9 | 88.0 |
| +DEx+DBA | 81.8 | 89.6 | 94.1 |
| +DEx+$k$-reciprocal | 82.5 | 94.4 | 95.2 |
| +DEx+Diffusion | **84.3** | **94.4** | **95.4** |
| +DEx+DBA+$k$-reciprocal | 83.7 | 88.6 | 94.0 |
| +DEx+DBA+Diffusion | **85.0** | **92.6** | **94.7** |

Table 6: Retrieval performance of different re-ranking techniques on the validation set.

10 mAP points, outperforming other singleton methods. Nonetheless, DEx obtains CMC scores higher than others by roughly 1.1 and 0.8 points. The addition of tracklet information along with the combination of post-processing steps did not yield gains on the validation set. However, we observe up to 2% improvement on the private test set. Figure 4 shows a few examples of retrieved vehicles in the test set using the baseline network and after re-ranking with our approach. The gain from diffusion similarly translates when combined with embedding enhancing techniques such as DEx and DBA.

### 4.4. Results on 2020 AI City Challenge

The results when using the proposed methods are submitted to the re-ID track of 2020 AI City Challenge. We have obtained an mAP at top-100 ranks (mAP@100) of 51.66 without any additional labeling or supervision. This is a large improvement over the baseline mAP@100 of 26.3 [35]. The results are obtained with the DEx+DBA+$k$-reciprocal combination, followed by pulling in the tracklets. The DEx+DBA+Diffusion yields slightly lower performance with 50.78 mAP. This is possibly due to overfitting of the diffusion parameters on the validation set.

## 5. Conclusions

In this paper, we propose a dual embedding expansion strategy that leverages multiple networks, scales along with weighted tracklet information to improve the re-ID performance. Our embedding expansion yields competitive performance to the de-facto standard in re-ID, outperforming conventional query expansion methods. We have performed ablations to study the impact of synthetic data during training and discovered that a balanced addition of synthetic data is beneficial. Furthermore, a detailed study is presented of several re-ranking techniques in image retrieval for vehicle re-ID. Although $k$-reciprocal is the common choice for

(a) Top-10 retrieval results of the ResNet50-IBN-a network.

(b) Top-10 retrieval results after re-ranking using DEx+DBA+$k$-reciprocal scheme.

Figure 4: Retrieval results of the baseline ResNet50-IBN-a network (top) and re-ranking (bottom). Re-ranking removes possible mismatches (second row) and increases the homogeneity of the top ranks. The query is the leftmost image.

re-ranking, diffusion offers competitive results with similar computational costs. Our evaluation of the different re-ranking techniques shows that $k$-reciprocal or diffusion is the best for re-ID, which can be further enhanced by combining it with other embedding expansion techniques without any additional annotations or manual supervision.

**Acknowledgements.** We thank NVIDIA Corp. for their grant of a Titan Xp GPU for research.

## References

[1] Relja Arandjelovic and Andrew Zisserman. Three things everyone should know to improve object retrieval. *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 2911–2918, 2012. 5

[2] Abner Ayala-Acevedo, Akash Devgun, Sadri Zahir, and Sid Askary. Vehicle re-identification: Pushing the limits of re-identification. In *CVPR Workshops*, 2019. 2

[3] Herbert Bay, Tinne Tuytelaars, and Luc Van Gool. Surf: Speeded up robust features. In *ECCV*, 2006. 2

[4] Hao Chen, Benoit Lagadec, and Francois Bremond. Partition and reunion: A two-branch neural network for vehicle re-identification. In *Proc. CVPR Workshops*, pages 184–192, 2019. 2, 3, 4, 7

[5] Yu-Cheng Chen, Longlong Jing, Elahe Vahdani, Ling Zhang, Mingyi He, and Yingli Tian. Multi-camera vehicle tracking and re-identification on ai city challenge 2019. In *CVPR Workshops*, 2019. 2

[6] Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In *2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05)*, volume 1, pages 886–893. IEEE, 2005. 2

[7] Jiankang Deng, Jia Guo, Niannan Xue, and Stefanos Zafeiriou. Arcface: Additive angular margin loss for deep face recognition. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019. 2

[8] Michael Donoser and Horst Bischof. Diffusion processes for retrieval revisited. *2013 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1320–1327, 2013. 5

[9] Bing He, Jia Li, Yifan Zhao, and Yonghong Tian. Part-regularized near-duplicate vehicle re-identification. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019. 2

[10] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 3

[11] Alexander Hermans, Lucas Beyer, and Bastian Leibe. In defense of the triplet loss for person re-identification. *arXiv*

*preprint arXiv:1703.07737*, 2017. 5

[12] Peixiang Huang, Runhui Huang, Jianjie Huang, Rushi Yangchen, Zongyao He, Xiying Li, and Junzhou Chen. Deep feature fusion with multiple granularity for vehicle re-identification. In *Proc. CVPR Workshops*, pages 80–88, 2019. 2, 4

[13] Tsung-Wei Huang, Jiarui Cai, Hao Yang, Hung-Min Hsu, and Jenq-Neng Hwang. Multi-view vehicle re-identification using temporal attention model and metadata re-ranking. In *CVPR Workshops*, 2019. 2

[14] Ahmet Iscen, Giorgos Tolias, Yannis Avrithis, Teddy Furon, and Ondřej Chum. Efficient diffusion on region manifolds: Recovering small objects with compact cnn representations. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 926–935, 2017. 5

[15] Aytaç Kanaci, Minxian Li, Shaogang Gong, and Georgia Rajamanoharan. Multi-task mutual learning for vehicle re-identification. In *CVPR Workshops*, 2019. 2

[16] Pirazh Khorramshahi, Amit Kumar, Neehar Peri, Sai Saketh Rambhatla, Jun-Cheng Chen, and Rama Chellappa. A dual-path model with adaptive attention for vehicle re-identification. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 6132–6141, 2019. 2

[17] Pirazh Khorramshahi, Neehar Peri, Amit Kumar, Anshul B. Shah, and Rama Chellappa. Attention driven vehicle re-identification and unsupervised anomaly detection for traffic understanding. In *CVPR Workshops*, 2019. 2

[18] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 5

[19] Chih-Ting Liu, Man-Yu Lee, Chih-Wei Wu, Bo-Ying Chen, Tsai-Shien Chen, Yao-Ting Hsu, Shao-Yi Chien, and NTU IoX Center. Supervised joint domain learning for vehicle re-identification. In *Proc. CVPR Workshops*, pages 45–52, 2019. 2

[20] Hongye Liu, Yonghong Tian, Yaowei Wang, Lu Pang, and Tiejun Huang. Deep relative distance learning: Tell the difference between similar vehicles. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2167–2175, 2016. 2

[21] Weiyang Liu, Yandong Wen, Zhiding Yu, Ming Li, Bhiksha Raj, and Le Song. Sphereface: Deep hypersphere embedding for face recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 212–220, 2017. 2

[22] Xinchen Liu, Wu Liu, Tao Mei, and Huadong Ma. A deep learning-based approach to progressive vehicle re-identification for urban surveillance. In *European conference on computer vision*, pages 869–884. Springer, 2016. 2

[23] Xiaobin Liu, Shiliang Zhang, Qingming Huang, and Wen Gao. Ram: a region-aware deep model for vehicle re-identification. In *2018 IEEE International Conference on Multimedia and Expo (ICME)*, pages 1–6. IEEE, 2018. 2

[24] Yihang Lou, Yan Bai, Jun Liu, Shiqi Wang, and Ling yu Duan. Veri-wild: A large dataset and a new method for vehicle re-identification in the wild. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3230–3238, 2019. 2

[25] David G Lowe. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2):91–110, 2004. 2

[26] Hao Luo, Youzhi Gu, Xingyu Liao, Shenqi Lai, and Wei Jiang. Bag of tricks and a strong baseline for deep person re-identification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 0–0, 2019. 3

[27] Khac-Tuan Nguyen, Trung-Hieu Hoang, Minh-Triet Tran, Trung-Nghia Le, Ngoc-Minh Bui, Trong-Le Do, Viet-Khoa Vo-Ho, Quoc-An Luong, Mai-Khiem Tran, Thanh-An Nguyen, et al. Vehicle re-identification with learned representation and spatial verification and abnormality detection with multi-adaptive vehicle detectors for traffic video analysis. In *Proc. CVPR Workshops*, 2019. 2

[28] Xingang Pan, Ping Luo, Jianping Shi, and Xiaoou Tang. Two at once: Enhancing learning and generalization capacities via ibn-net. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 464–479, 2018. 3

[29] Adithya Shankar, Akhil Poojary, Varghese Alex Kollerathu, Chandan Yeshwanth, Sheetal Reddy, and Vinay Sudhakaran. Comparative study on various losses for vehicle re-identification. In *CVPR Workshops*, 2019. 2

[30] Josef Sivic and Andrew Zisserman. Video google: A text retrieval approach to object matching in videos. page 1470. IEEE, 2003. 2

[31] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2818–2826, 2016. 5

[32] Mingxing Tan and Quoc V Le. Efficientnet: Rethinking model scaling for convolutional neural networks. *arXiv preprint arXiv:1905.11946*, 2019. 3

[33] Xiao Tan, Zhigang Wang, Minyue Jiang, Xipeng Yang, Jian Wang, Yuan Gao, Xiangbo Su, Xiaoqing Ye, Yuchen Yuan, Dongliang He, Shilei Wen, and Errui Ding. Multi-camera vehicle tracking and re-identification based on visual and spatial-temporal features. In *CVPR Workshops*, 2019. 2

[34] Yi Tang, Di Wu, Zhi Jin, Wenbin Zou, and Xia Li. Multi-modal metric learning for vehicle re-identification in traffic surveillance environment. *2017 IEEE International Conference on Image Processing (ICIP)*, pages 2254–2258, 2017. 2

[35] Zheng Tang, Milind Naphade, Ming-Yu Liu, Xiaodong Yang, Stanley T. Birchfield, Shuo Wang, Ratnesh Kumar, David C. Anastasiu, and Jenq-Neng Hwang. Cityflow: A city-scale benchmark for multi-target multi-camera vehicle tracking and re-identification. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8789–8798, 2019. 2, 7

[36] Zheng Tang, Gaoang Wang, Hao Xiao, Aotian Zheng, and Jenq-Neng Hwang. Single-camera and inter-camera vehicle tracking and 3d speed estimation based on fusion of visual and semantic features. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 108–115, 2018. 2

[37] Shangzhi Teng, Xiaobin Liu, Shiliang Zhang, and Qingming Huang. Scan: Spatial and channel attention network for vehicle re-identification. In *Pacific Rim Conference on Multimedia*, pages 350–361. Springer, 2018. 2

[38] Ries Uittenbogaard, Clint Sebastian, Julien Vijverberg, Bas Boom, Dariu M Gavrila, et al. Privacy protection in street-view panoramas using depth and multi-view imagery. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 10581–10590, 2019. 2

[39] Guanshuo Wang, Yufeng Yuan, Xiong Chen, Jiwei Li, and Xi Zhou. Learning discriminative features with multiple granularities for person re-identification. In *Proceedings of the 26th ACM international conference on Multimedia*, pages 274–282, 2018. 4

[40] Hao Wang, Yitong Wang, Zheng Zhou, Xing Ji, Dihong Gong, Jingchao Zhou, Zhifeng Li, and Wei Liu. Cosface: Large margin cosine loss for deep face recognition. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018. 2

[41] Zhongdao Wang, Luming Tang, Xihui Liu, Zhuliang Yao, Shuai Yi, Jing Shao, Junjie Yan, Shengjin Wang, Hongsheng Li, and Xiaogang Wang. Orientation invariant feature embedding and spatial temporal regularization for vehicle re-identification. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 379–387, 2017. 2

[42] Yandong Wen, Kaipeng Zhang, Zhifeng Li, and Yu Qiao. A discriminative feature learning approach for deep face recognition. In *European conference on computer vision*, pages 499–515. Springer, 2016. 2

[43] Sanghyun Woo, Jongchan Park, Joon-Young Lee, and In So Kweon. Cbam: Convolutional block attention module. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 3–19, 2018. 5

[44] Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1492–1500, 2017. 3

[45] Ke Yan, Yonghong Tian, Yaowei Wang, Wei Zeng, and Tiejun Huang. Exploiting multi-grain ranking constraints for precisely searching visually-similar vehicles. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 562–570, 2017. 2

[46] Xinyu Zhang, Rufeng Zhang, Jiewei Cao, Dong Gong, Mingyu You, and Chunhua Shen. Part-guided attention learning for vehicle re-identification. *arXiv preprint arXiv:1909.06023*, 2019. 2

[47] Yanzhu Zhao, Chunhua Shen, Huibing Wang, and Shengyong Chen. Structural analysis of attributes for vehicle re-identification and retrieval. *IEEE Transactions on Intelligent Transportation Systems*, 2019. 2

[48] Zhedong Zheng, Tao Ruan, Yunchao Wei, and Yezhou Yang. Vehiclenet: Learning robust feature representation for vehicle re-identification. In *CVPR Workshops*, 2019. 2, 3

[49] Zhun Zhong, Liang Zheng, Donglin Cao, and Shaozi Li. Re-ranking person re-identification with k-reciprocal encoding. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3652–3661, 2017. 2, 5

[50] Zhun Zhong, Liang Zheng, Guoliang Kang, Shaozi Li, and Yi Yang. Random erasing data augmentation. *arXiv preprint arXiv:1708.04896*, 2017. 6