Vehicle Re-Identification: an Efficient Baseline Using Triplet Embedding

Ratnesh Kumar Edwin Weill Farzin Aghdasi Parthasarathy Sriram
NVIDIA
{ ratneshk,eweill } @nvidia.com

Abstract

In this paper we tackle the problem of vehicle re-identification in a camera network utilizing triplet embeddings. Re-identification is the problem of matching appearances of objects across different cameras. With the proliferation of surveillance cameras enabling smart and safer cities, there is an ever-increasing need to re-identify vehicles across cameras. Typical challenges arising in smart city scenarios include variations of viewpoints, illumination and self occlusions. Most successful approaches for re-identification involve (deep) learning an embedding space such that the vehicles of same identities are projected closer to one another, compared to the vehicles representing different identities. Popular loss functions for learning an embedding (space) include contrastive *or* triplet *loss. In this paper we provide an extensive evaluation of these losses applied to vehicle re-identification and demonstrate that using the best practices for learning embeddings outperform most of the previous approaches proposed in the vehicle re-identification literature. Compared to most existing state-of-the-art approaches, our approach is simpler and more straightforward for training utilizing only identity-level annotations, along with one of the smallest published embedding dimensions for efficient inference. Furthermore in this work we introduce a formal evaluation of a triplet sampling variant (*batch sample*) into the re-identification literature.*

1. Introduction

Matching appearances of objects across multiple cameras is an important problem for many computer vision applications, *e.g.* object retrieval and object identification. This problem of object re-identification is closely related to object recognition and fine grained classification. In the realm of video understanding, most higher level algorithms such as event recognition and anomaly detection rely upon *Multiple Camera Multiple Object Tracking* (MC-MOT). An important component for a MC-MOT is an *object verification* (*i.e.* re-identification) module for expressing *confidence* to associate objects across multiple videos [35]. Re-identification approaches can also be used in a single camera setup, wherein the task would be to determine if the same object has re-appeared in the scene [20, 47, 42].

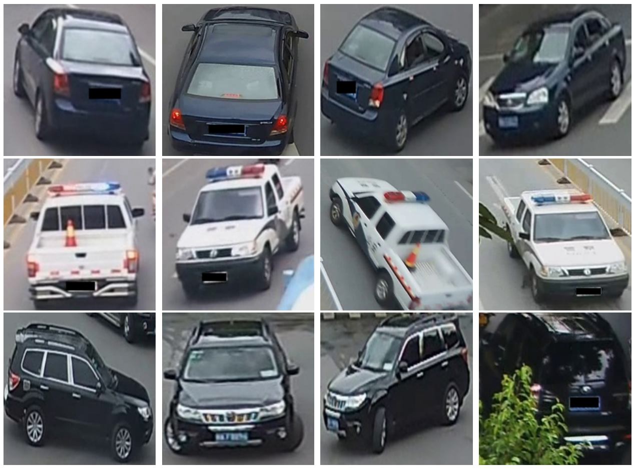

Figure 1. Each row is a separate identity (samples taken from **VeRi** dataset [25]). Despite large intra-class variations for views, vehicle-model could be discerned from most views.

The task of vehicle re-identification is to identify the same vehicle across a camera network. With the deployment of camera sensors for traffic management and smart cities, there is an imminent need to perform vehicle search from video databases [33]. Previous works [41, 16] have shown that automatic recognition of license plates as a global unique identifier have given state-of-the-art identification performance. However in general traffic scenes at streets, license plates are practically invisible in many views to recognize due to their top view installations. Therefore, a vision-based re-identification has a great practical value in real world scenarios. Re-identification of objects is challenging due to significant appearance & viewpoint shifts, lighting variations and varied object poses. Figure 1 shows some typical challenging intra-class variations.

Compared to person and face re-identification, vehicle re-identification is a relatively under-studied problem. A few of the unique characteristics pertaining to the problem of vehicle re-identification which make it a difficult task are:

- Multiple views of the same vehicle are visually diverse and semantically (i.e. color and model) correlated, meaning that the same identity must be deduced no matter which viewpoint of the vehicle is given.

- In real world scenarios, a re-identification system is expected to extract subtle physical cues such as the presence of dust, written marks, or dents on vehicle surfaces, to be able to distinguish between vehicles which are the same color and model.

- The vehicle labels are less fine-grained than person (or face)-identity labels. Given that there are a finite number of vehicle colors and models, the diversity in a given dataset is less than that of a person or face re-identification dataset.

In order to match appearances of objects, firstly we need to obtain an embedding for the objects, also denoted as a feature vector or signature. A match is then performed by using a suitable distance metric expressing the closeness of two objects in an embedding space. A good embedding should be invariant to illumination, scale and viewpoint changes. Prior to the advancements in deep learning, most embedding learning approaches focus on handcrafting using mixture of multiple feature extractors and/or learning suitable ranking functions to minimize distance across objects of similar identities. Some of the notable approaches are [43, 3, 29, 28, 6, 22, 50].

In this paper we focus on the embedding part of the re-identification process and make the following **contributions**:

- Utilizing the recent advances in *sampling* informative data points for learning embedding for the person re-identification task[11], we extensively evaluate their application to the vehicle re-identification problem, and demonstrate state-of-the-art performance across diverse datasets on various performance metrics.

- We introduce a formal evaluation of a triplet sampling variant, *batch sample*, into the re-identification literature.

The rest of the paper is organized as: in the following section we provide an overview of related works and the subsequent section will elaborate on triplet and contrastive losses, including popular sampling techniques to optimize these losses. Section 4 details on datasets and hyperparameters used for various experimental settings. Results and discussions are presented in section 5.

2. Related Works

In recent years with the evolution of end-to-end learning using *Convolutional Neural Networks* (CNN), significant improvements have been made in feature representations using large amounts of training data. These approaches outperform all previous baselines using handcrafted features. A CNN learns hierarchical image features by stacking convolutional layers with downsampling layers. The outputs from one convolutional layer is fed to a non-linearity layer before being fed to the subsequent convolutional layer.

[4] proposed one of the first approaches to learn visual relationships using CNN. *Siamese* CNN [4] computes an embedding space such that similar examples have similar embeddings and vice versa. [5] uses *contrastive* loss on Siamese CNN to learn embedding for face verification. One of the recent prominent works using CNNs for learning face embedding [36] uses *triplet loss* to train a CNN for learning face embeddings for identification. While triplet loss considers three samples *jointly* for computing a loss measure, contrastive loss requires only two samples. Contrastive loss is computationally more efficient than triplet, however, several approaches [30, 9, 35, 2, 11, 13] have reported state-of-the-art performances using triplet loss. This superiority of triplet loss is attributed to the additional context using three samples. Section 3 in this paper elaborates on these losses.

Another method for obtaining an embedding for an object is utilizing a traditional softmax layer [48, 18], wherein a fully-connected (embedding) layer is added prior to the softmax-loss layer. Each identity is considered as a separate category and the number of categories is equal to the number of identities in the training set. Once the network is trained using classification loss (*e.g.* cross-entropy), the classification layer is stripped off and an embedding is obtained form the new final layer of the network. [18] proposed a similar approach to learning vehicle embedding based on training a network for vehicle-model classification task. Since the network is not directly trained on embedding or metric learning loss, usually the performance of such a network is poor when compared to networks incorporating embedding loss. Cross entropy loss ensures separability of features but the features may not be discriminative enough for separating unseen identities. Furthermore learning becomes computationally prohibitive when considering datasets of *e.g.* 10^6 identities. Some recent works [47, 34, 37] unify classification loss with metric learning.

Vehicle Classification: Fine grained vehicle classification is a closely related problem to vehicle re-identification. Notable works for vehicle classification are [21, 8, 15, 23, 40, 27]. The general task is to predict vehicle *model*, *e.g.* BMW-i3-2016, Toyota-Camry-1996. Vehicle re-identification is a relatively finer grained problem than vehicle-model classification: a re-identification approach should be able to extract visual differences between two vehicles belonging to the same model category. The visual differences could include subtle cosmetic and color differences making this problem more difficult. Furthermore a re-identification method is expected to work without any *a priori* knowledge of all possible vehicle models in the city or a geographical entity.

Vehicle Re-identification: Some notable approaches prior to deep learning are [26, 50]. Popular deep learning approaches for vehicle re-identification are [44, 24, 49, 2, 25, 9, 38, 45, 52, 18, 53]. [25] proposed fusion of handcrafted features *e.g.* color, texture along with high level attribute feature obtained using CNN. [44] proposed a progressive refinement approach to searching query vehicles. A list of candidates is obtained for a query using embeddings from a siamese-CNN trained using contrastive loss. This list is then pruned using a siamese network to match license plates. In order to get reliable query for visually similar vehicles, authors factor in the usage of spatio-temporal distance comparison in addition to visual embedding distances.

[9] presents a structured deep learning loss comprising a classification loss term (based on vehicle model) as well as coarse and fine grained ranking terms. [24] proposed a modification of triplet loss by replacing anchor samples with corresponding class center in order to suppress effects of using poor anchors. Furthermore the deep model is trained for both vehicle model classification and identity labels in a multi-level process. [49] focuses on the relationship between different vehicle images as multiple grains by using diverse vehicle attributes. The authors proposed ranking methods incorporated into multi-grain classification.

In a recent work [2], the authors propose to include group-based sub-clustering in a triplet loss framework. This helps in explicitly dealing with intra-class variations of vehicle identification problem. During training an online grouping method is used to cluster samples within each identity into disparate clusters. The authors demonstrate state-of-the-art results in different datasets.

[52] proposes to use a view-point synthesis approach to predict embedding for unknown views given a true view image. These synthetic embeddings for unknown views are generated using bi-directional LSTM [12]. The complete network is trained using a combination of contrastive, reconstruction and generative adversarial loss [7]. Similar to the objective of [52] for inferring a global feature vector using view-synthesis, authors in [53] propose a *viewpoint attentive multi-view* framework. Utilizing attentive [32] and adversarial loss, authors transform a single view feature into a global multi-view feature representation.

[45] develops a framework utilizing keypoint annotations on vehicles to learn viewpoint invariant features from a CNN. To further enhance the retrieval of matching vehicles the authors use probabilistic spatio-temporal regularization using random variables representing camera transition probabilities. The authors demonstrate superior results by adding this regularization during retrieval procedure. [38] formulate these *camera transition* probabilities by generating proposals of path (trajectories) and employing a LSTM and Siamese CNN to obtain a robust re-identification performance.

3. Loss functions for embedding

For a reliable re-identification of objects, the following are some desired characteristics of an embedding function:

- An embedding should be invariant to viewpoints, illumination and shape changes to the object.
- For a practical application deployment, computation of embedding and ranking should be efficient.

Consider a dataset $\mathcal{X} = \{(x_i, y_i)\}_{i=1}^{N}$ of N training images $x_i \in \mathbb{R}^D$ and their corresponding class labels $y_i \in \{1 \cdots C\}$. Re-identification approaches aim to learn an embedding $f(x;\theta) : \mathbb{R}^D \rightarrow \mathbb{R}^F$ to map images in $\mathbb{R}^D$ onto a feature (embedding) space in $\mathbb{R}^F$ such that images of similar identity are metrically close in this feature space. θ corresponds to the parameters of the learning function.

$$\theta^* = \arg\min_{\theta} \ \mathcal{L}(f(\theta, \mathcal{X})) \tag{1}$$

Let $D(x_i, x_j) : \mathbb{R}^{\mathbb{F}} \times \mathbb{R}^{\mathbb{F}} \rightarrow \mathbb{R}$ be a metric measuring distance of images x_i and x_j in embedding space. For simplicity we drop the input labels and denote $D(x_i, x_j)$ as D_{ij}. $y_{ij} = 1$ is both samples i and j belong to the same class and $y_{ij} = 0$ indicates samples of different classes.

3.1. Contrastive Loss

Contrastive loss (2) was employed in [5] for the face verification problem, wherein the objective is to verify if two presented faces belong to the same identity. This discriminative loss directly optimizes (1) by encouraging all similar class distances to approach 0 while keeping all dis-similar class distances to be above a pre-defined threshold α.

$$l_{contrastive}(i,j) = y_{ij} D_{ij}^2 + (1 - y_{ij})[\alpha - D_{ij}^2]_+ \tag{2}$$

Notice that the choice of α is same for all dissimilar classes. This implies that for dissimilar identities, visually diverse classes are embedded in the same feature space as the visually similar ones. This assumption is stricter when compared to triplet loss, and restricts the structure of the embedding manifold thereby impairing discriminative learning. The training complexity is $O(N^2)$ for a dataset of N samples.

3.2. Triplet Loss

Inspired from the seminal work on metric learning for nearest neighbor classification by [46], *facenet* [36] proposed a modification suited for retrieval tasks *i.e.* equation (3), termed: triplet loss. Triplet loss forces the data points from the same class to be closer to each other than a data point form any other class. Notice that contrary to contrastive loss in (2), triplet loss adds context to the loss function by considering both a positive and negative pair distances from the same point.

$$l_{triplet}(a, p, n) = [D_{ap} - D_{an} + \alpha]_{+} \tag{3}$$

Training complexity of triplet loss is $O(N^3)$ which is computationally prohibitive. High computational complexity of triplet and contrastive losses have motivated a host of sampling approaches for an efficient optimization.

Dataset Sampling

As triplet and contrastive losses are computationally prohibitive for practical datasets, most proposed approaches resort to sampling *effective* data points for computing losses. This is important as computing loss over trivial data points could only impair convergence of the algorithm. In the context of vehicles, it will be more informative for a loss function to sample from different views (*e.g.* side or front-view) for the same identity, than considering samples of similar views repeatedly.

A popular sampling approach to find informative samples is *hard data mining*, and is employed in many computer vision applications *e.g.* object detection. Hard data mining is a bootstrapping technique which is used in iterative training of a model, wherein at every iteration the current model is applied on a validation set to mine hard data on which this model is performing poorly. Only these hard data are then presented to the optimizer which increases the ability of the model to learn effectively and converge faster to an optimum. On the flip side, if a model is only presented with hard data, which could comprise outliers, its ability to discriminate outliers *w.r.t.* normal data would suffer.

In order to deal with the outliers during hard data sampling, facenet [36] proposed *semihard* sampling which mines moderate triplets that are neither too hard nor too trivial for getting meaningful gradients during training. The generation of semihard samples is performed offline and on CPU which severely impedes convergence. [11] proposed a very efficient and effective approach to mine samples directly on GPU. The authors construct a data batch by randomly sampling P identities from $\mathcal{X}$ and then randomly sampling K images for each identity, thus resulting in a batch size of PK images. In a batch size of PK images, the authors [11] proposed two sampling techniques, namely **batch hard** (BH) (also in [31]) and **batch all** (BA). Another sampling technique **batch sample** (BS) is actively discussed in the implementation webpage of [11], however to the best of our knowledge we could not find a formalized study and evaluation for this sampling technique.

[35] unifies different batch sampling techniques in [11] under one expression. Let a be an anchor sample and $N(a)$ and $P(a)$ represent a subset of negative and positive samples for the corresponding anchor a. The triplet loss can then be written as:

$$l_{triplet}(a) = [\alpha + \sum_{p \in P(a)} w_p D_{ap} - \sum_{n \in N(a)} w_n D_{an}]_{+} \tag{4}$$

With respect to an anchor sample a: w_p represents the weight (importance) of positive sample p and similarly w_n signifies the importance of the negative sample n.

The total loss in an epoch is then obtained by:

$$\mathcal{L}(\theta; \mathcal{X}) = \sum_{all\ batches} \sum_{a \in B} l_{triplet}(a)$$

Table 1 summarizes different ways of sampling positives and negatives. We formalize BS method in this regime. BH is hard data mining in the batch, using only the hardest positive and negative samples for every anchor. BA is a straightforward sampling which gives uniform weights to all samples. Uniform weight distribution can ignore the contribution of important tough samples as these samples are typically outnumbered by the trivial easy samples. In order to mitigate this issue with BA, [35] employs a weighting scheme **batch weighted** (BW), wherein a sample is weighted based on its distance from the corresponding anchor, thereby giving more importance to the informative harder samples than trivial samples.

BS uses the distribution of anchor-to-sample distances to mine a positive and negative data for an anchor. This technique thereby avoids sampling outliers when compared with BH, and also hopes to find out the most relevant sample as the sampling is done using distances-to-anchor distribution.

In the following sections, we evaluate the embedding losses, along with the sampling variants presented in Table 1.

4. Experiments

For our evaluation purposes we use three popular publicly available datasets: VeRi, VehicleID and PKU-VD.

VeRi: This dataset is proposed by [25] and is one of the main datasets used in vehicle re-identification literature for comparative study. This dataset encompasses 40,000 bounding box annotations of 776 cars (identities) across 20 cameras in traffic surveillance scenes. Each vehicle is captured in 2-18 cameras in various viewpoints and varying illuminations. Notably the viewpoints are not restricted to only front/rear but also side views, thereby making it one of the challenging datasets. The annotations include make and model of vehicles, color and inter-camera relations and trajectory information.

VehicleID: This dataset [24] comprises 221,763 bounding boxes of 26,267 identities, captured across various surveillance cameras in a city. Annotations include 250 vehicle

Sampling variant	Positive weight: w_p	Negative weight: w_n	Comments
Batch all (BA)	1	1	Uniformly weighted
Batch hard (BH)	$[\, x_p == \arg\max_{x \in P(a)} \; D_{ax} \,]$	$[\, x_n == \arg\min_{x \in N(a)} \; D_{ax} \,]$	Hardest sample
Batch sample (BS)	$[\, x_p == \underset{x \in P(a)}{\text{multinomial}} \{D_{ax}\} \,]$	$[\, x_n == \underset{x \in N(a)}{\text{multinomial}} \{D_{ax}\} \,]$	Multinomial sampling
Batch weighted (BW)	$\frac{e^{D_{ap}}}{\sum\limits_{x \in P(a)} e^{D_{ax}}}$	$\frac{e^{-D_{an}}}{\sum\limits_{x \in N(a)} e^{-D_{ax}}}$	Adaptive weights

Table 1. Various ways of mining good samples in a batch, for better optimization of embedding loss.

models and this dataset has an order of magnitude more images than VeRi dataset. However the viewpoints only include front and rear views for vehicles.

PKU-VD: [49] proposed a large dataset for fine grained vehicle analysis including re-identification and classification. To this date this is the largest dataset comprising about *two million* images and their fine grained labels including vehicle model and color. This dataset is split into two sub-datasets, namely **VD1** and **VD2** based on cities from which they were captured. The images in VD1 are captured from high resolution cameras, while images for VD2 are obtained from surveillance cameras. There are about 71k and 36k identities in VD1 and VD2, respectively.

4.1. Training and Hyperparameters

For our experiments, we fix our backbone or meta-architecture to *mobilenet* [14] owing to its better efficiency (parameters, speed) as compared to ResNet-variants [10] and VGG [39]. The imagenet [17] retrieval accuracy for these architectures are in similar ranges.

We use Adam optimizer [19] with default hyperparameters ($\epsilon = 10^{-3}$, $\beta_1 = 0.9$, $\beta_2 = 0.999$). Depending upon if the training is done from scratch or fine-tuned using an imagenet [17] based trained model, we employ different learning rate schedulers. When training from scratch, we use standard learning rate of 0.001. We reduce this rate to 0.0003 when using an imagenet based pre-trained model. For online data augmentation a standard image-flip operation is used. We use Nvidia's Volta GPU for hardware and Tensorflow [1] as the software platform.

We replace the margin α in triplet loss (4) by *softplus* function: $ln(1 + exp(\cdot))$ which avoids the need of tuning this margin [11]. For contrastive loss we follow standard practice of hard margin of 1.0. Using a softplus function produced poorer results for contrastive loss.

For the *batch construction*, unless otherwise specified, we follow the default batch sizes as in [11, 35]. A batch consists of 18 (P) randomly chosen identities, and for each identity, 4 (K) samples are chosen randomly, thereby selecting a total of 72 (PK) images. Samples are chosen such that we iterate over all train set during the course of an epoch. Following the standards in face-verification and person re-identification [35], [36] we set the embedding dimension to *128 units*.

4.2. Evaluation Metrics

We use mean-average-precision (*mAP*) and *top-k* accuracy for evaluating and comparing our presented approaches. In a typical re-identification evaluation setup, we have a query set and a gallery set. For each vehicle in a query set the aim is to retrieve a similar identity from the test set (*i.e.* gallery set). $AP(q)$ for a query image q is defined as:

$$AP(q) = \frac{\sum\limits_k P(k) \times \delta_k}{N_{gt}(q)}$$

where P(k) represents precision at rank k, $N_{gt}(q)$ is the total number of true retrievals for q. δ_k is 1 when the matching of query image q to a test image is correct at rank $<= k$. mAP is then computed as average over all query images:

$$mAP = \frac{\sum\limits_q AP(q)}{Q}$$

where Q is the total number of query images.

5. Results and Discussions

We present our results on the datasets mentioned in the previous section. Different datasets have different ways of constructing test sets which we elaborate in the respective sections. Each model presented below is trained separately on the corresponding dataset using its standard train set.

5.1. VeRi

We follow the standard evaluation protocol by [44]. The total number of query images is 1,678 while the gallery set comprises 11,579 images. For every query image, the gallery set contains images of same query-identity but taken from *different* cameras. This is an important evaluation exclusion as in many cases the same camera samples would contain visually similar samples for the same vehicle.

Table 2 summarizes our results for various sampling configurations, and we can draw following inferences:

Sampling	mAP	top-1	top-2	top-5
Triplet, Not-Normalized				
BH	65.10	87.25	91.54	94.76
BA	66.91	90.11	**93.38**	96.01
BS	**67.55**	**90.23**	92.91	96.42
BW	67.02	89.99	93.15	**96.54**
Triplet, Normalized				
BH	53.72	72.65	80.27	86.83
BA	27.60	42.91	53.16	67.76
BS	33.79	48.75	58.64	73.54
BW	44.29	60.91	69.85	80.63
Contrastive, Normalized				
BH	59.21	80.51	85.52	90.64
BS	52.09	71.51	78.84	86.95
Contrastive, Not-Normalized				
BH	56.84	75.33	82.30	90.29
BS	48.85	65.49	74.55	85.76

Table 2. VeRi accuracy results (%) using triplet and contrastive loss for different batch sampling variants outlined in Table 1.

- Adding a normalized layer performs poorly for the triplet loss. This is also reported by [11] wherein using a normalized layer could result in collapsed embeddings.

- Siamese (contrastive) loss under performs relative to triplet loss. We attribute this to the additional context provided by using both positive and negative samples in the same term for the triplet loss [30].

- For the best performing set, *i.e.* triplet loss with no-normalization layer: all four sampling variants reach about similar accuracy ranges, with BS outperforming others in a close range.

- Figure 2 shows some visual results with embeddings learned from batch-sampling triplet loss. Good top-k retrievals indicate stability of our embeddings across different views and cameras. Notice that query and gallery images are constrained to be from different cameras following the standard evaluation protocol.

Comparison to the state-of-the-art approaches: Table 3 outlines comparisons with the state-of-the-art approaches. Notice that our approach outperforms all the other results for the *mAP* metric. GSTE [2] achieves better top-k accuracy but in terms of mAP our approach performs better indicating robustness at all ranks. Furthermore GSTE [2] has an embedding dimension of 8x more (*i.e.* 1024) than ours, and GSTE includes a complicated training process which requires tuning an additional intra-class clustering parameter.

The VeRi dataset includes spatio-temporal (ST) information and [53, 52, 45, 44] utilize ST information in either embedding or in retrieval stages. *Noticeably* without using any ST information, we outperform these approaches using ST. Contrary to us, OIFE [45] requires extra annotations of key-points during training for their orientation invariant embedding learning. Training procedure for VAMI [53] include *generative adversarial network* (GAN) and multi-view attention learning. Path-LSTM [38] employ generation of several path-proposals for their spatio-temporal regularization and requires an additional LSTM to rank these proposals. It is worth noting that our training procedure is more straightforward than most of the approaches presented in Table 3, with an efficient embedding dimension of 128. Table 4 outlines some important differences *w.r.t.* competitive approaches.

Method	mAP	top-1	top-5
BS (Ours)	**67.55**	90.23	96.42
GSTE [2]	59.47	**96.24**	**98.97**
VAMI [53]	50.13	77.03	90.82
VAMI+ST * [53]	61.32	85.92	91.84
OIFE [45]	48.00	89.43	-
OIFE+ST *[45]	51.42	92.35	-
PROVID * [44]	27.77	61.44	78.78
Path-LSTM * [38]	58.27	83.49	90.04

Table 3. **Comparison** of various proposed approaches on VeRi dataset. (*) indicates the usage of spatio-temporal information.

Method	ED	Multi-View	Annotations
Ours	128	No	ID
GSTE [2]	1024	No	ID
VAMI [53]	2048	Yes	ID + Attribute
OIFE [45]	256	No	ID + Keypoints
MGR [49]	1024	No	ID + Attribute
ATT [49]	1024	No	ID + Attribute
C2F [9]	1024	No	ID + Attribute
CLVR [18]	1024	No	Attribute

Table 4. Summary of some important hyperparameters and labeling used during training. **ED** indicates embedding dimension. OIFE merges four datasets to form one large training set. Notice that our ED is the least among other approaches.

Referring to the best results in Table 2, in the subsequent sections we consider only triplet loss without embedding-normalization.

5.2. PKU-VD

PKU-VD is a large dataset combining two sub-datasets, VD1 and VD2. Both of these comprise about 400k training images. The test set of each of the sub-dataset is split into *three* reference sets: small, medium and large. Table 5 shows the number of test images in each sub-dataset. For evaluation, we use the same dataset files for each reference set as provided by the authors [49] of this dataset.

Dataset	Small	Medium	Large
VD1	106,887	604,432	1,097,649
VD2	105,550	457,910	807,260

Table 5. Number of images in each reference test-set.

Compared to VeRi and VehicleID datasets, PKU-VD dataset has an order of magnitude more images, hence a network can be trained from scratch on this dataset. Furthermore with more intra-class samples, one can increase the batch size of triplets. Tables 6, 7 and 8 show results for various configurations. For the BW sampling in Table 6, the numerics following illustrate the P and K values, described previously, which create the batch. Table 7 adds results for the other three sampling variants when training from scratch. Table 8 shows results for the default batch size (18x4).

Using more triplets in the batch improves the accuracy, which is intuitively satisfying. Noticeably using the hardest sample (BH) does not kick-off the training (*c.f.* Table 7). This is expected and also noted in [36], as with BH due to random initialization, the network never learns any understanding to separate hard data from easy samples. One way to deal with this is to start training with a few identities in a multi-class setting in-order to pre-train the network and then proceed with the standard BH procedure. Alternatively one could start from an imagenet trained network (*c.f.* Table 8). The other sampling variants are more robust and hence converges to a better solution than the default PK batch-sized training.

BW sampling with bach size of 18x16 outperforms the precious state-of-the-art by [49]. Multi-grain ranking (MGR) uses permutation probability based ranking method and include vehicle attributes during training process. Noticeably our training procedure is straightforward without using vehicle attributes. Furthermore MGR uses an embedding dimension of 1024 as opposed to 128 for our embedding, thus calling for higher computation cost during inference in [49].

Method	Small	Medium	Large
VD1			
BW (18x16)	**87.48**	**67.28**	**58.77**
MGR [49]	79.10	58.30	51.10
VD2			
BW (18x16)	**84.55**	**69.87**	**63.64**
MGR [49]	74.70	60.60	55.30

Table 6. **mAP** (%) for retrievals on various reference sets. Training is performed from scratch without using pretrained weights.

Dataset, Sampling	Small	Medium	Large
No pretrained weights			
VD1, BA	85.02	62.84	54.68
VD1, BH	0.00	0.00	0.00
VD1, BS	87.24	66.62	58.26
VD2, BA	83.39	68.58	62.34
VD2, BH	0.00	0.00	0.00
VD2, BS	83.30	68.45	62.36

Table 7. **mAP** (%) for retrievals on various reference sets of different sizes. Training is performed without pretrained weights with batch size of 18x16.

Dataset, Sampling	Small	Medium	Large
With pretrained weights			
VD1, BW	**82.66**	60.15	52.10
VD1, BS	81.36	58.91	50.68
VD1, BA	79.46	56.79	49.26
VD1, BH	82.04	**60.40**	**52.17**
VD2, BW	**80.93**	**65.44**	**58.94**
VD2, BS	75.52	58.35	51.71
VD2, BA	70.07	50.56	43.46
VD2, BH	78.95	62.32	55.86

Table 8. **mAP** (%) for retrievals on various reference sets of different sizes. Training is performed using imagenet pretrained weights with default batch size of 18x4.

5.3. VehicleID

VehicleID [24] is a larger dataset than VeRi containing front and rear views for the vehicles. We follow the standard evaluation protocol of [24] and provide results on four reference *query* sets. Reference sets: small, medium. large and X-large contain 800, 1600, 2400 and 13164 identities, respectively. For each reference set, an exemplar for an identity is randomly chosen, and a *gallery* set is constructed. This process is repeated ten times to obtain *averaged* evaluation metrics. For training we use mobilenet network, pretrained using imagenet dataset, without normalization-layer for embedding. Similarly to the PKU-VD dataset training we set the batch size (PK) to 18x16 images. For the sake of completeness we provide the results with default PK batch size of 18x4.

Tables 9, 10 and 11 show comparative results for mAP and top-k metrics, respectively. Similarly to the PKU-VD results, using a larger batch size increases the retrieval rankings, however the margin of improvement is smaller. This could be due to limited variability in this dataset in terms of viewpoints and number of vechicle-models, owing to which increasing the batch size does not necessarily increase informative statistics.

BS and BW outperform other sampling variants, including all state-of-the-art approaches in the mAP metric. Table 4 and section 5.1 summarizes important differences of

Method	Small	Medium	Large	X-Large
BA	84.65	79.85	75.95	59.74
BS	**86.19**	**81.69**	**78.16**	**62.41**
BW	85.92	81.41	78.13	62.12
BH	85.59	80.76	76.87	60.33
C2F [9]	63.50	60.00	53.00	-
GSTE [2]	75.40	74.30	72.40	-
ATT [49]	62.80	62.30	58.60	-
CCL [24]	54.60	48.10	45.50	-

Table 9. Accuracy results on VehicleID using **mAP** metric (%). Batch size for our experiments is set to 18x16 samples.

Method	Small	Medium	Large	X-Large
BA	81.90	76.57	72.60	54.95
BS	84.17	79.05	75.52	59.10
BW	**84.90**	**80.80**	**77.20**	**60.92**
BH	83.34	78.72	75.02	57.97

Table 10. Accuracy results on VehicleID using **mAP** metric (%). This is with default PK batch size of (18x4).

Method	Small	Medium	Large	X-Large
Top-1				
BA	76.69	71.20	66.71	50.22
BS	**78.80**	73.41	69.33	**53.07**
BW	78.49	73.10	69.41	52.82
BH	77.90	72.14	67.56	50.67
OIFE [45]	-	-	67.00	-
OIFE+ [45]	-	-	68.00	-
VAMI [53]	63.12	52.87	47.34	-
CCL [24]	49.00	42.80	38.20	-
C2F [9]	61.10	56.20	51.40	-
GSTE [2]	75.90	**74.80**	**74.00**	-
CLVR [18]	62.00	56.10	50.60	-
Top-5				
BA	95.26	91.17	87.75	70.48
BS	**96.17**	**92.57**	**89.45**	**73.06**
BW	95.83	92.48	89.36	72.72
BH	95.74	92.03	88.81	71.23
OIFE [45]	-	-	82.90	-
VAMI [53]	83.25	75.12	70.29	-
CCL [24]	73.50	66.80	61.60	-
C2F [9]	81.70	76.20	72.20	-
GSTE [2]	84.20	83.60	82.70	-
CLVR [18]	76.00	71.80	68.00	-

Table 11. Results on VehicleID dataset using **top-k** metric (%). Batch size for our experiments is set to 18x16 samples.

state-of-the-art approaches *w.r.t.* our approach. GSTE [2] achieves better performance in terms of top-1 accuracy, but their accuracy drops for top-5. Lower mAP and top-5 indicates GSTE's sub-par retrieval performances for ranks $k > 1$. OIFE+ [45] achieves close accuracy in top-5 to ours. As opposed to our approach, OIFE+ requires keypoint annotations and a separate metric learning module from [51]. Furthermore OIFE combines VeRi, VehicleID, CompCars [27] and Cars21k [40] into one large train set.

Contrary to our method, other approaches [9, 24, 49], all utilize model annotations (in addition to identity annotations) from the training set for re-identification.

6. Conclusion and Future Work

In this paper we propose a strong baseline for vehicle re-identification using the best practices in learning deep triplet embedding [11]. The core ideas behind this set of best practices lie in constructing a batch to facilitate extracting meaningful statistics in order to guide training and convergence. We also introduced a formal exposition and evaluation of a triplet sampling variant, *batch sample* to the re-identification literature.

We compared our baselines with the state-of-the-art approaches on three datasets and outperform almost all of them in a wide range of evaluation criteria. The sampling variants: *batch sample* and *batch weighted* proved generally more effective and robust than *batch hard* and *batch all*.

We hinged our research on the belief that despite the intra-class variations, the identity of a vehicle is less fine grained than other object re-identification task, *e.g.* person re-identification. Our results demonstrate this by using the recent advances in embedding learning, we can push the frontiers of vehicle re-identification much further without using any spatio-temporal information. On the other hand, two vehicles of exactly the same color and model (with subtle or no discerning marks, *e.g.* last row in Figure 2) would be very difficult to distinguish without any spatio-temporal information. Incorporating spatio-temporal information along with other attributes in an effective manner is an important contribution as future work.

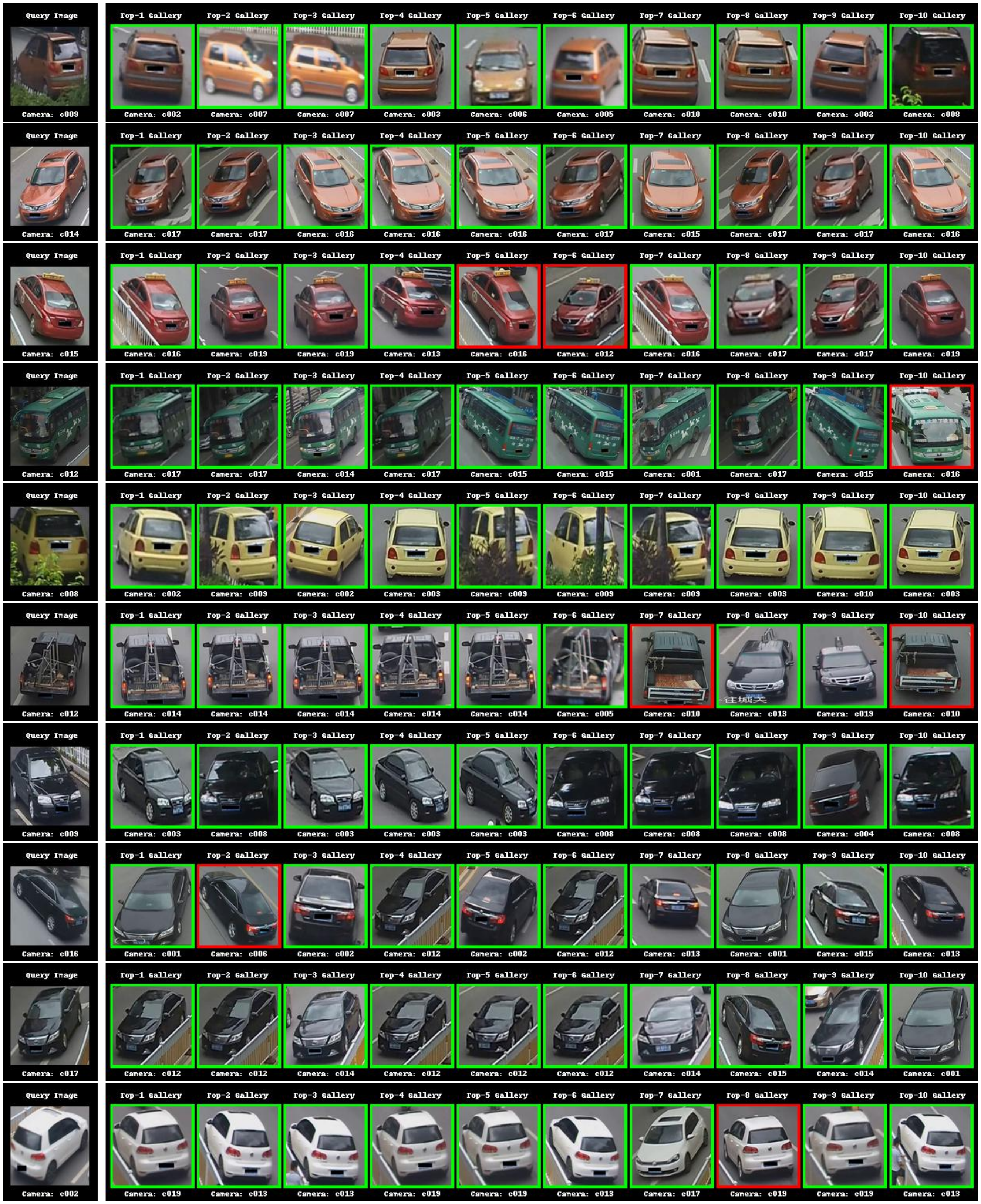

Figure 2. Qualitative results on VeRi dataset using *BS* based triplet embedding. Each row indicates query image and top-10 retrievals for this query image. Red border indicates incorrect retrieval and Green indicates correct retrievals. These demonstrate good embedding quality as the top retrievals include different views and cameras.

References

[1] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. Software available from tensorflow.org.

[2] Y. Bai, Y. Lou, F. Gao, S. Wang, Y. Wu, and L. Duan. Group Sensitive Triplet Embedding for Vehicle Re-identification. *IEEE Transactions on Multimedia*, 2018.

[3] S. Bak, M. S. Biagio, R. Kumar, V. Murino, and F. Bremond. Exploiting Feature Correlations by Brownian Statistics for People Detection and Recognition. *IEEE Transactions on Systems, Man, and Cybernetics*, 2017.

[4] J. Bromley, J. W. Bentz, L. Bottou, I. Guyon, Y. Lecun, C. Moore, E. Säckinger, and R. Shah. Signature Verification Using a Siamese Time Delay Neural Network. *International Journal of Pattern Recognition and Artificial Intelligence*, 1993.

[5] S. Chopra, R. Hadsell, and Y. LeCun. Learning a similiarty metric discriminatively, with application to face verification. In *CVPR*, 2005.

[6] M. Farenzena, L. Bazzani, A. Perina, V. Murino, and M. Cristani. Person Re-Identication by Symmetry-Driven Accumulation of Local Features. In *CVPR*, 2010.

[7] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative Adversarial Networks. In *NIPS*, 2014.

[8] H. Z. Gu and S. Y. Lee. Car model recognition by utilizing symmetric property to overcome severe pose variation. *Machine Vision and Applications*, 2013.

[9] H. Guo, C. Zhao, Z. Liu, J. Wang, and H. Lu. Learning Coarse-to-Fine Structured Feature Embedding for Vehicle Re-Identification. In *AAAI*, 2018.

[10] K. He, X. Zhang, S. Ren, and J. Sun. Deep Residual Learning for Image Recognition. In *CVPR*, 2016.

[11] A. Hermans, L. Beyer, and B. Leibe. In Defense of the Triplet Loss for Person Re-Identification. In *CVPR*, 2017.

[12] S. Hochreiter and J. Schmidhuber. Long Short-Term Memory. *Neural Computation*, 1997.

[13] E. Hoffer and N. Ailon. Deep metric learning using triplet network. In *ICLR Workshops*, 2015.

[14] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam. MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications. In *CVPR*, 2017.

[15] Q. Hu, H. Wang, T. Li, and C. Shen. Deep CNNs with Spatially Weighted Pooling for Fine-Grained Car Recognition. *IEEE Transactions on Intelligent Transportation Systems*, 2017.

[16] V. Jain, Z. Sasindran, A. Rajagopal, S. Biswas, H. S. Bharadwaj, and K. R. Ramakrishnan. Deep automatic license plate recognition system. *ICVGIP*, 2016.

[17] Jia Deng, Wei Dong, R. Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. ImageNet: A large-scale hierarchical image database. In *CVPR*, 2009.

[18] A. Kanaci, X. Zhu, and S. Gong. Vehicle Re-Identification by Fine-Grained Cross-Level Deep Learning. In *BMVC*, 2017.

[19] D. P. Kingma and J. Ba. Adam: A Method for Stochastic Optimization. In *ICLR*, 2015.

[20] R. Kumar, G. Charpiat, and M. Thonnat. Multiple Object Tracking by Efficient Graph Partitioning. In *ACCV'14*.

[21] L. Liao, R. Hu, J. Xiao, Q. Wang, J. Xiao, and J. Chen. Exploiting effects of parts in fine-grained categorization of vehicles. In *ICIP*, 2015.

[22] S. Liao, Y. Hu, X. Zhu, and S. Z. Li. Person re-identification by Local Maximal Occurrence representation and metric learning. In *CVPR*, 2015.

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

[24] H. Liu, Y. Tian, Y. Wang, L. Pang, and T. Huang. Deep Relative Distance Learning: Tell the Difference Between Similar Vehicles. In *CVPR*, 2016.

[25] X. Liu, W. Liu, H. Ma, and H. Fu. Large-scale vehicle re-identification in urban surveillance videos. *ICME*, 2016.

[26] X. Liu, H. Ma, H. Fu, and M. Zhou. Vehicle Retrieval and Trajectory Inference in Urban Traffic Surveillance Scene. In *ICDSC*, 2014.

[27] P. Luo, C. C. Loy, X. Tang, L. Yang, P. Luo, C. C. Loy, and X. Tang. A Large-Scale Car Dataset for Fine-Grained Categorization and Verification. In *CVPR*, 2015.

[28] B. Ma, Y. Su, F. Jurie, B. Ma, Y. Su, and F. Jurie. Local Descriptors Encoded by Fisher Vectors for Person Re-identification. In *ECCV Workshops*, 2012.

[29] B. P. Ma, Y. Su, and F. Jurie. BiCov: a novel image representation for person re-identification and face verification. In *BMVC*, 2012.

[30] R. Manmatha, C. Y. Wu, A. J. Smola, and P. Krahenbuhl. Sampling Matters in Deep Embedding Learning. In *CVPR*, 2017.

[31] A. Mishchuk, D. Mishkin, F. Radenovic, and J. Matas. Working hard to know your neighbor's margins: Local descriptor learning loss. In *NIPS*, 2017.

[32] V. Mnih, N. Heess, A. Graves, and K. Kavukcuoglu. Recurrent Models of Visual Attention. In *NIPS*, 2014.

[33] M. Naphade, M.-C. Chang, A. Sharma, C. Anastasiu, David, V. Jagarlamudi, P. Chakraborty, T. Huang, S. Wang, M. Y. Liu, R. Chellappa, J.-N. Hwang, and S. Lyu. The 2018 NVIDIA AI City Challenge. *CVPR Workshops*, 2018.

[34] O. Rippel, M. Paluri, P. Dollar, and L. Bourdev. Metric Learning with Adaptive Density Discrimination. In *ICLR*, 2016.

[35] E. Ristani and C. Tomasi. Features for Multi-Target Multi-Camera Tracking and Re-Identification. In *CVPR*, 2018.

[36] F. Schroff, D. Kalenichenko, and J. Philbin. FaceNet: A unified embedding for face recognition and clustering. In *CVPR*, 2015.

[37] L. Shen, Z. Lin, and Q. Huang. Relay Backpropagation for Effective Learning of Deep Convolutional Neural Networks. In *ECCV*, 2016.

[38] 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. *ICCV*, 2017.

[39] K. Simonyan and A. Zisserman. Very Deep Convolutional Networks for Large-Scale Image Recognition. In *ICLR*, 2015.

[40] J. Sochor, A. Herout, and J. Havel. BoxCars: 3D Boxes as CNN Input for Improved Fine-Grained Vehicle Recognition. In *CVPR*, 2016.

[41] J. Spanhel, J. Sochor, R. Juranek, A. Herout, L. Marsik, and P. Zemcik. Holistic recognition of low quality license plates by CNN using track annotated data. *AVSS*, 2017.

[42] S. Tang, M. Andriluka, B. Andres, and B. Schiele. Multiple people tracking by lifted multicut and person re-identification. In *CVPR*, 2017.

[43] O. Tuzel, F. Porikli, and P. Meer. Region covariance: A fast descriptor for detection and classification. In *European Conference on Computer Vision*, pages 589–600, 2006.

[44] Y. Wang, L. Xie, S. Qiao, Y. Zhang, W. Zhang, and A. L. Yuille. A Deep Learning-Based Approach to Progressive Vehicle Re-identification for Urban Surveillance. In *ECCV*, 2016.

[45] Z. Wang, L. Tang, X. Liu, Z. Yao, S. Yi, J. Shao, J. Yan, S. Wang, H. Li, and X. Wang. Orientation Invariant Feature Embedding and Spatial Temporal Regularization for Vehicle Re-identification. In *ICCV*, 2017.

[46] K. Q. Weinberger and L. K. Saul. Distance Metric Learning for Large Margin Nearest Neighbor Classification. *The Journal of Machine Learning Research*, 10:207–244, 2009.

[47] N. Wojke and A. Bewley. Deep Cosine Metric Learning for Person Re-identification. In *WACV*, 2018.

[48] T. Xiao, H. Li, W. Ouyang, and X. Wang. Learning Deep Feature Representations with Domain Guided Dropout for Person Re-identification. In *CVPR*, 2016.

[49] K. Yan, Y. Tian, Y. Wang, W. Zeng, and T. Huang. Exploiting Multi-grain Ranking Constraints for Precisely Searching Visually-similar Vehicles. In *ICCV*, 2017.

[50] D. Zapletal, A. Herout, and A. Herout. Vehicle Re-Identification for Automatic Video Traffic Surveillance. In *CVPR Workshops*, 2016.

[51] L. Zhang, T. Xiang, and S. Gong. Learning a Discriminative Null Space for Person Re-identification. In *CVPR*, 2016.

[52] Y. Zhou and L. Shao. Vehicle Re-Identification by Adversarial Bi-Directional LSTM Network. In *WACV*, 2018.

[53] Y. Zhou and L. Shao. Viewpoint-aware Attentive Multi-view Inference for Vehicle Re-identification. In *CVPR*, 2018.