Multi-View Vehicle Re-Identification using 
Temporal Attention Model and Metadata Re-ranking

Tsung-Wei Huang¹, Jiarui Cai¹, Hao Yang², Hung-Min Hsu¹,³, and Jenq-Neng Hwang¹

¹ Department of Electrical and Computer Engineering
² Department of Civil and Environmental Engineering
University of Washington
³ Research Center for Information Technology Innovation, Academia Sinica, Taiwan
{twhuang, jrcai, haoya, hmhsu, hwang}@uw.edu

Abstract

Object re-identification (ReID) is an arduous task which requires matching an object across different non-overlapping camera views. Recently, many researchers are working on person ReID by taking advantages of appearance, human pose, temporal constraints, etc. However, vehicle ReID is even more challenging because vehicles have fewer discriminant features than human due to viewpoint orientation, changes in lighting condition and inter-class similarity. In this paper, we propose a viewpoint-aware temporal attention model for vehicle ReID utilizing deep learning features extracted from consecutive frames with vehicle orientation and metadata attributes (i.e., type, brand, color) being taken into consideration. In addition, re-ranking with soft decision boundary is applied as post-processing for result refinement. The proposed method is evaluated on CVPR AI City Challenge 2019 dataset, achieving mAP of 79.17% with the second place ranking in the competition.

1. Introduction

Object re-identification (ReID) is a challenging task in computer vision community and gains a lot of attention in various applications, such as pedestrian retrieval and public safety monitoring [4, 38, 9, 33, 34]. Generally speaking, ReID could be considered as a retrieval problem, i.e., given a probe object, either an image or a video clip, we need to search in the gallery for the same object that appears in multiple cameras. However, this retrieval is difficult because 1) Different viewpoints of an object are distinct in appearance and shape. 2) Intra-class variability due to background clutters, resolution, illumination, and object size across cameras. 3) Inter-class similarity of objects. Recently, numerous methods have been proposed to solve person ReID [37, 39, 15, 25] and vehicle ReID problems [18, 16, 31]. In a ReID task, the correspondence of probe and gallery candidate is determined based on a measurement of similarity distance. To overcome these difficulties, the state-of-the-art methods apply metric learning on global features and the training is end-to-end. People also explore tricks such as fine-grain features [19], synthesized information [39], temporal constraints [7] and re-ranking [40], etc.

As for vehicle ReID, though the problem has been studied by the research community for long, most existing methods take advantages of web images [5], which are less distorted and in high resolution. In terms of traffic cameras, studies on license plate [18] and images with fixed vehicle orientation [32] have shown exceptional performance. However, in real-world applications, the orientations and lighting could be varied while license plates are usually occluded, for example, in the top-view or side-view. Therefore, we focus on constructing reliable and discriminant features for individual vehicles.

Our proposed method is focusing on image-to-video vehicle ReID. As shown in 1, for each clip, frame-based features, including CNN features for appearance and vehicle structure features, are aggregated through a temporal attention model. The ReID network is trained end-to-end using a metric learning method with batch sample triplet loss and cross entropy loss. Finally, the metadata classification feature is used for soft-thresholding and re-ranking to refine the results.

In summary, our contributions are threefold: 1) We propose a novel method for vehicle ReID which incorporates vehicle appearance, orientation, attributes, and temporal information. 2) We propose a vehicle similarity measurement algorithm based on feature fusion and metadata re-ranking.
Figure 1. Overview of the proposed vehicle ReID method. During training, for each video clip, we first extract frame-level features, including appearance and structure features. Then, a temporal attention model (TA) is trained to obtain the aggregated clip-level features (purple). These clip-level features are fused with structure features and used for training the ReID model by cross-entropy and batch sampling triplet losses. Finally, combining the similarity of learning-based features with metadata re-ranking, we refine the final ReID results.

3) Our vehicle ReID method achieves mAP of 79.17% and rank-1 accuracy 82.51% in the CVPR 2019 AI City Challenge [1].

2. Related Works

2.1. Vehicle Feature Extraction

The feature extraction methods generally fall into two primary categories, one is traditional keypoint-descriptor based methods, like SURF [3], ORB [24], and the other is the deep learning feature extractors, like CNN [30, 21, 36]. Comparing with traditional handcrafted features, CNN feature extractors usually perform better because they can extract more discriminant features robustly by supervised learning while training the neural networks. More specifically, the feature extraction stage is jointly optimized with the following classification/regression stage in the deep learning framework. However, most of the popular CNN feature extractors are trained on different classes of objects which may have apparent inter-class features. However, for this task, we need to distinguish the differences within one single vehicle class. To deal with this problem, we need to retrain the CNN model, and more importantly, create more discriminant features for different types of vehicles.

Thus, we include a unique feature for the vehicles – vehicle keypoints, and visibility into our feature set. Vehicle keypoints feature is a kind of object shape prior which has been widely used for object and scene reconstruction [42, 28, 22, 2]. Among these works, Ansari et al. [2] train a keypoint localization network, which is based on a stacked-hourglass architecture [23] using a synthesized dataset built from 3D CAD models of vehicles. When combined with visibility information, this keypoints feature can give us a useful structure feature for each vehicle to improve our ReID performance potentially.

Besides the above vehicle features, to deal with inter-class similarity problem, we also consider vehicle attributes. Vehicle classification, especially on brand or model, is a fine-grained classification task. Several benchmark datasets are focusing on car model classification [32, 13], however, the samples are web images that are of relatively high resolution and better viewpoints comparing to traffic cameras. The other problem is that the majority of vehicle types in these datasets are sedans and SUVs, and limited in other common categories such as pickup trucks, trucks, and bus. On the other hand, in the traffic cameras, studies are concentrated on type, and color features [12] due to the labeling difficulties of attributes. Here, we propose a CNN classifier to identify the vehicle type, brand and color using progressive training combining both web image dataset and the AI City Track2 traffic camera image dataset.

2.2. Video-based ReID

The typical ReID tasks can be divided into image-based and video-based. Image-based ReID uses images as the content in probe and gallery, while video-based ReID uses videos. However, the principal solutions are similar. The state-of-the-art methods apply metric learning with different loss functions, such as hard triplet loss [11], cross entropy loss [26], center loss [29], and their combination to train classifiers [41]. For video-based ReID, we need to consider the features from a clip, i.e., a small set of consecutive images [35, 17, 7]. Temporal information is important for ReID in a video clip, as evidenced by the temporal attention modeling proposed in [7] to give different attention scores to different frames.

2.3. Re-Ranking

The purpose of re-ranking is to improve the ranking results from a previous ReID outcome. Usually, it is used as a post-processing step in ReID tasks, which can be implemented without any additional training. Many previous works on re-ranking have been proposed for retrieval and person ReID tasks [34, 40, 8]. Zhong et al. [40] propose a re-ranking method with $k$-reciprocal encoding. They first increase the $k$-reciprocal nearest neighbors into a more ro-
bust set by adding selected positive samples which are more similar to the previous candidates. Then, a Jaccard distance is re-calculated for the new candidates set. It is a simple but efficient idea for re-ranking in ReID problems. However, in our scenarios, we have more information that we can benefit from like vehicle metadata; therefore, this re-ranking method is further improved in this paper.

3. Our Proposed Approach

In this section, we introduce our proposed method for vehicle ReID based on video clips. Firstly, a feature extraction method precisely for vehicles is described in both frame-level and clip-level features. Then, a metadata classifier is designed for classifying different types, colors, and brands. After that, the network is trained using metrics learning with hard triplet loss and cross-entropy loss. Finally, a novel re-ranking method with soft decision boundary involving metadata information is introduced for refining the ReID rankings.

3.1. Feature Extraction

Frame-level Feature Extraction. To reduce noise, the training images are fed into a preprocessing flow including size check, image quality filtering and keypoint detection. The prepossessed frame features are then extracted from a ResNet50 [12] network that is pre-trained on ImageNet. The 2048-dim fully-connected layer before classification layer is used to represent the appearance of the vehicle. In addition, we use the keypoint localization method described in [2] to obtain 36 points on the vehicle with semantic meaning as structure features to infer the viewpoint. An example of extracted vehicle keypoints is shown in Figure 2.

Vehicle Orientation Feature Descriptor. To describe the vehicle orientation from the keypoints’ layout on the 2D image, we leverage the 3D relationships between each keypoint on the 3D vehicle model. For each surface formed by the keypoints on the 3D vehicle model (Figure 2), we define its surface normal as the normal vector pointing outward of the vehicle body. Using the right-hand rule, we can calculate the signed-area of the projection of the surface on the 2D image. If the signed-area is positive, we know the surface is facing toward the camera and vice versa. The example of positive and negative signed-areas are shown in Figure 3. By concatenating the signed-areas of the projection of all the surfaces and performing L2-normalization, we get the 18-dimensional vehicle orientation feature descriptor \( f_o \). To demonstrate the descriptor, we use t-SNE [20] to convert the descriptor into 2D space (Figure 4). We can see that the feature descriptor well captures the vehicle orientation and forms clusters of the front view, left/right side view, back view, ..., etc. Note that we only consider the areas and ignore the in-plane rotation of the image because the vehicles are usually standing upright on the ground.

Viewpoint-Aware Temporal Attention Model. After we extract the frame-level features, we combine them into clip-level features using a temporal attention modeling (TA) [7]. The structure of the temporal attention modeling is shown in Figure 5. The spatial convolutional network is a 2D convolution operation and temporal convolutional network is a 1D convolution operation. We train these two networks to get more reliable attention scores for the frames in video clips. After the weighted average, we can get the clip-level features \( f_c \).

3.2. Loss Function

Inspired from metric learning for face verification problems, FaceNet [6] proposes triplet loss to force the data points from the intra-class to be closer to each other than a data point from any inter-class. Triplet loss is used to train a transformation to project an input image to an embedding feature space so that the Euclidean distance of the embedding features are optimized. Assume there is an anchor feature \( a \), the same vehicle \( y_a \) should be projected to a positive feature \( x_p \), which is closer to anchors position,
instead of a negative feature $x_n$, belonging to another class $y_b$, by at least a margin $m$. Based on [14], we compared the performance of different batch-based sampling approaches for training triplet embedding, we adopt batch sample (BS) [14] instead of batch hard (BH) [6] in the triplet generation. In terms of BS, it uses the multinomial distribution of anchor-to-sample distances to sample data for training. The idea of BS is to filter sampling outliers during the training. The BS triplet loss in a mini-batch $X$ is defined as,

$$
L_{BSTri}(\theta; X) = \sum_{\text{all batches}} \sum_{a \in B} l_{\text{triplet}}(a),
$$

where

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

with $w_p$ and $w_n$ are the weighting of positive and negative samples, respectively, $D_{ap}$ and $D_{an}$ are the distances between the anchor sample to the positive sample and negative sample, respectively, and $m$ is the defined margin.

Based on BS strategy, the weighting of positive and negative samples are defined as follows,

$$
w_p = P(x_p = \text{multinomial}_{x \in P(a)} \{ D_{ax} \}),
$$

$$
w_n = P(x_n = \text{multinomial}_{x \in N(a)} \{ D_{ax} \}),
$$

where $x_p$ and $x_n$ are positive and negative samples, respectively.

Moreover, we also include cross-entropy ($Xent$) loss...
in the training as follows,

\[ \mathcal{L}_{Xent} = - \sum_{i=1}^{P} \log (\text{prob}(i))q(i), \] \hspace{1cm} (4)

where \( q(i) \) is the one-hot ground truth label, \( \text{prob}(i) \) is the probability of the probe vehicle belongs to vehicle \( i \).

The overall loss function is a weighted combination of BS triplet loss and cross-entropy loss,

\[ \mathcal{L}_{Total} = \lambda_1 \mathcal{L}_{BS} + \lambda_2 \mathcal{L}_{Xent}. \] \hspace{1cm} (5)

### 3.3. Re-Ranking Strategy

**Metadata classification.** Metadata classification is a typical multi-label classification task. Here, vehicle type, brand, and color are considered as vehicle metadata attributes. We adopted the main structure of a 29-layer light CNN framework proposed by Wu et al. [30]. Small kernel sizes of convolution layers, network-in-network layers and residual blocks have been implemented to reduce the parameter space and improve performance. The max-feature-map (MFM) operation is an extension of maxout activation, which combines two feature maps and outputs element-wise maximum one. We include CompCar [32] which combines two feature maps and outputs element-wise maximum one. We include CompCar [32] dataset and 8 cameras (1 hour each) of self-record traffic videos as part of our training data.

The orientation and visibility of a vehicle are estimated using the vehicle keypoints. The four wheels are marked as \( P_{\text{front},\text{left}}, P_{\text{front},\text{right}}, P_{\text{back},\text{left}} \) and \( P_{\text{back},\text{right}} \), the driving direction of a vehicle is described as a vector pointing from the center of back axle to the center of front axle, i.e., \( \vec{r} = ((P_{\text{back},\text{left}} + P_{\text{back},\text{right}})/2, (P_{\text{front},\text{left}} + P_{\text{front},\text{right}})/2) \). The orientation of a vehicle could be simply modeled by the angle of \( \vec{r} \) from horizontal. The 2D space are split into eight zones that are \([350^\circ, 10^\circ)\), \([10^\circ, 80^\circ)\), \([80^\circ, 100^\circ)\), ..., \([280^\circ, 350^\circ)\), as in Figure 6. In each zone, the visible surfaces of a vehicle are known and it is straightforward to localize the areas semantically, shown in Figure 7.

All images are preprocessed with orientation estimation and the visible parts are cropped for data augmentation. We enlarge the input size to be \( 512 \times 512 \), the aspect ratio is kept with zero padding in the boundary. Comparing to the original version in [30], additional one network-in-network layer is added and the size of fully-connected layer is extended to \( 2048 \)-dim rather than \( 256 \)-dim. The details of metadata labeling are described in 4.1. Due to the limited number in labeled traffic data, we train the classifier in a progressive way. The self-recorded data are fed into the model that is pre-trained on CompCar and AI City training set, samples with high confidence are included into the training set. The model is trained and evaluated iteratively until it achieves a decent accuracy on the validation set.

**Metadata Distance.** After we get the initial ReID results, metadata information can also be considered. The intuitive idea is that the samples with different metadata classes would have larger distance. According to this idea, we propose a metadata distance given the metadata probabilities \( p_i \) and \( p_j \) for samples \( i \) and \( j \) from classes \( c_i \) and \( c_j \), separately

\[ s(p_i, p_j) = \text{conf}(p_i) \times \text{conf}(p_j) \times \text{confu}(p_i, p_j) \] \hspace{1cm} (6)

where

\[ \text{conf}(p_i) = \frac{D_{KL}(p_i \parallel p_U)}{\log N_c} \] \hspace{1cm} (7)

and

\[ \text{confu}(p_i, p_j) = -\log N_c \cdot P(c_i = c_j | p_i, p_j) \] \hspace{1cm} (8)

are the classification confidence and confusion distance respectively. The \( D_{KL}(P || Q) \) is the KL Divergence between \( P \) and \( Q \), \( N_c \) is the number of classes and \( p_U \) is the uniform distribution. Therefore, we have \( 0 \leq \text{conf}(P) \leq 1 \). The \( P(c_i = c_j | p_i, p_j) \) can be derived from the confusion matrix so that the lower the \( P(c_i = c_j | p_i, p_j) \), the larger the confusion distance.

By using metadata distance, we can obtain a new distance by combining the initial ReID distance and metadata distance,

\[ d'(p, g_i) = d(p, g_i) + \gamma \cdot \sum_n s_n(p, g_i). \] \hspace{1cm} (9)

where \( p \) is a probe image, \( g_i \) is the \( i \)-th gallery image, \( \gamma \) is a hyperparameter that can be fine-tuned, and \( n \) is metadata category.

**Re-ranking with \( k \)-reciprocal Encoding.** Re-ranking is a post-processing step typically used for ReID tasks. For
Figure 7. Examples of vehicle keypoints detection and visibility estimation. Keypoints are marked in blue dots, occlusion parts are estimated. The red, green, orange and purple outlines indicate the front, back, left side and right side, respectively. The yellow arrow represents the driving direction $\vec{r}$. Cropped views of visible surfaces are shown as well.

In this part, we basically adopt the re-ranking method proposed in [40] and fine-tune the hyperparameters for our case. Firstly, we generate $k$-reciprocal nearest neighbor set for the original $k$-NN set as shown in Figure 8. Then, we re-calculate the distance between probe and gallery by adding Jaccard distance,

$$d_J(p, g_i) = 1 - \frac{|R^*(p, k) \cap R^*(g_i, k)|}{|R^*(p, k) \cup R^*(g_i, k)|}. \quad (10)$$

Thus, the final distance $d^*$ is defined as

$$d^*(p, g_i) = (1 - \lambda)d_J(p, g_i) + \lambda d'(p, g_i). \quad (11)$$

Figure 8. Example of the $k$-reciprocal neighbors expansion process. The positive vehicle $G$ which is similar to $C$ is added into $R^*(Q, 20)$.

4. Experiments

4.1. Dataset

The benchmark dataset [1] is captured by 40 traffic cameras, including the scenario of intersections, street roads and highways. The resolution and lightening condition vary for each camera, and distortion is introduced if captured by fish-eye cameras. A total of 666 vehicles are annotated with distinct vehicle IDs, in which 333 vehicles are used for training and the remaining 333 vehicles are for testing. There are 56277 images in total. 18290 images are in the testing set, 36935 images are in the training set and 1052 images are for query. On average, each vehicle has 81 image signatures from 4.53 camera views. License plates are masked in black for privacy consideration.

In addition to the original AI C19 ReID dataset, we also spent great effort on labeling all vehicles in the training set with their type, brand, and color. We finalize the labels in the following categories: 1) Type: sedan, suv, minivan, pickup truck, hatchback and truck; 2) Brand: Dodge, Ford, Chevrolet, GMC, Honda, Chrysler, Jeep, Hyundai, Subaru, Toyota, Buick, KIA, Nissan, Volkswagen, Oldsmobile, BMW, Cadillac, Volvo, Pontiac, Mercury, Lexus, Saturn, Benz, Mazda, Scion, Mini, Lincoln, Audi, Mitsubishi and others; 3) Color: black, white, red, grey, silver, gold, blue, green and yellow.

4.2. Vehicle ReID Performance

The performance of ReID is evaluated using mean average precision (mAP), which is the area below the Precision-Recall curve, measuring the mean of the precision of all query samples at different recall values. We also report rank-1, rank-5, rank-10, rank-20 and rank-100 accuracy, meaning the percentage of the queries that the top $\tau$ correspondence in the gallery is true positive.

In the CVPR 2019 AI City Challenge Track2 (vehicle ReID track), our method ranks second place among the total 84 submissions. The performance of top-10 algorithms is shown in Table 1, detailed statistics of our method in Table 2.
and comparisons with baseline method in Table 3. Example Re-ID results of our algorithm are shown in Figure 9.

5. Conclusion

A novel vehicle ReID method based on vehicle clip feature extraction with temporal attention and metadata re-ranking was proposed. We effectively design the representation of clip features in the aspects of appearance, structure, categories and temporal weighted aggregation, so that our algorithm is made comprehensive, robust and efficient.

The proposed method achieves mAP of 79.17% on CVPR AI City Challenge 2019 dataset.

Acknowledgement: The authors would like to thank many people who helped in the improvement of the performance of the proposed system: Yizhou Wang, Haotian Zhang, Ping Zhang, Huihao Chen and Zexin Li. We also thank STAR Lab in the University of Washington for providing traffic video dataset.
Table 1. Competition results of AI City Challenge Re-ID, ours is marked bold.

<table>
<thead>
<tr>
<th>Rank</th>
<th>Team ID</th>
<th>Team Name</th>
<th>mAP Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>59</td>
<td>Zero_One</td>
<td>0.8554</td>
</tr>
<tr>
<td>2</td>
<td>21</td>
<td>UWPL</td>
<td><strong>0.7917</strong></td>
</tr>
<tr>
<td>3</td>
<td>97</td>
<td>ANU</td>
<td>0.7589</td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td>expensiveGPUs</td>
<td>0.7560</td>
</tr>
<tr>
<td>5</td>
<td>12</td>
<td>Traffic Brain</td>
<td>0.7302</td>
</tr>
<tr>
<td>6</td>
<td>53</td>
<td>Desire</td>
<td>0.6793</td>
</tr>
<tr>
<td>7</td>
<td>131</td>
<td>XINGZHI</td>
<td>0.6091</td>
</tr>
<tr>
<td>8</td>
<td>5</td>
<td>UWD_RC</td>
<td>0.6078</td>
</tr>
<tr>
<td>9</td>
<td>78</td>
<td>MVM</td>
<td>0.5862</td>
</tr>
<tr>
<td>10</td>
<td>127</td>
<td>flyZJ</td>
<td>0.5827</td>
</tr>
</tbody>
</table>

Table 2. The mAP, rank-1, rank-5, rank-10, rank-20 and rank-100 performance of our method.

<table>
<thead>
<tr>
<th>Methods</th>
<th>mAP</th>
<th>Rank-1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baselines [27]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Resnet50 + Htri</td>
<td>30.3</td>
<td>44.3</td>
</tr>
<tr>
<td>Resnet50 + Xent</td>
<td>28.6</td>
<td>46.0</td>
</tr>
<tr>
<td>Resnet50 + Htri + Xent</td>
<td>33.0</td>
<td>51.8</td>
</tr>
<tr>
<td>Ours (Resnet50+TA+BStri+Xent+Rerank)</td>
<td><strong>79.2</strong></td>
<td><strong>82.5</strong></td>
</tr>
</tbody>
</table>

Table 3. Results of baseline method and our method. All backbone network is pretrained on ImageNet. The mAP and ranking are in percentage, best is marked in bold.

References


