

RAM: A REGION-AWARE DEEP MODEL FOR VEHICLE RE-IDENTIFICATION

Xiaobin Liu¹, Shiliang Zhang¹, Qingming Huang², Wen Gao¹

¹School of Electronics Engineering and Computer Science, Peking University, China

²School of Computer and Control Engineering, University of Chinese Academy of Sciences, China

{xbliu.vmc, slzhang.jdl, wgao}@pku.edu.cn, qmhuang@ucas.ac.cn

ABSTRACT

Previous works on vehicle Re-ID mainly focus on extracting global features and learning distance metrics. Because some vehicles commonly share same model and maker, it is hard to distinguish them based on their global appearances. Compared with the global appearance, local regions such as decorations and inspection stickers attached to the windshield, may be more distinctive for vehicle Re-ID. To embed the detailed visual cues in those local regions, we propose a Region-Aware deep Model (RAM). Specifically, in addition to extracting global features, RAM also extracts features from a series of local regions. As each local region conveys more distinctive visual cues, RAM encourages the deep model to learn discriminative features. We also introduce a novel learning algorithm to jointly use vehicle IDs, types/models, and colors to train the RAM. This strategy fuses more cues for training and results in more discriminative global and regional features. We evaluate our methods on two large-scale vehicle Re-ID datasets, *i.e.*, *VeRi* and *VehicleID*. Experimental results show our methods achieve promising performance in comparison with recent works.

Index Terms— Vehicle Re-ID, Deep Convolutional Neural Network (DCNN), Region-Aware Deep Model

1. INTRODUCTION

Vehicle Re-Identification (Re-ID) targets to identify the reappearing vehicles taken by a camera network. It is potential to address the challenging issues like intelligent surveillance video analysis and processing. It is also important for promising applications on intelligent transportation and smart city, such as finding and tracking specific vehicles. Several related tasks on vehicle identification have been extensively studied, such as vehicle attribute prediction [1] and fine-grained vehicle classification [2, 1, 3]. Those tasks mainly focus on identifying the fine-grained categories of vehicles, such as the specific maker and model. Differently, vehicle Re-ID requires the model to distinguish different vehicle instances. As different vehicles of same maker and model may be similar with each other in global appearance, vehicle Re-ID is more challenging and is far from being solved.



Fig. 1: Examples of different vehicles with similar global appearance. Each column shows two different vehicles. The differences on local regions are highlighted with red circles. It can be observed that, the differences between similar vehicles mostly lie on some local regions.

Most of existing works focus on designing or learning a robust visual representation for vehicle Re-ID [4, 3]. Some additional clues may also be helpful for this task, *e.g.*, the spatio-temporal information [4, 5, 6]. The quality of surveillance videos is commonly affected by many factors, such as illumination, weather, viewpoint, and occlusion, *etc.* Therefore, hand-crafted features may be unstable in surveillance scenario. Recently, Deep Convolutional Neural Network (DCNN) makes breakthrough in many applications including person Re-ID [7, 8, 9], fine-grained retrieval [10, 11], and face recognition [12, 13]. In those applications, deep features have shown substantial advantages over hand-crafted features in handling noisy visual data. Recently, some datasets have been released to facilitate the research on vehicle Re-ID, such as *VeRi* [4, 14] and *VehicleID* [3]. Those large-scale datasets make it possible to train and design DCNN for this task. Liu *et al.* [3] propose a two-branch DCNN structure trained with vehicle IDs and models categories, respectively. They also propose a new distance metric learning method called coupled cluster loss to improve the traditional triplet loss. Liu *et al.* [14] investigate different features and show improvements by fusing hand-crafted features and deep features. Wang *et al.* [6] first predict 20 keypoints and locate 4 regions based on those keypoints. They then fuse local features ex-

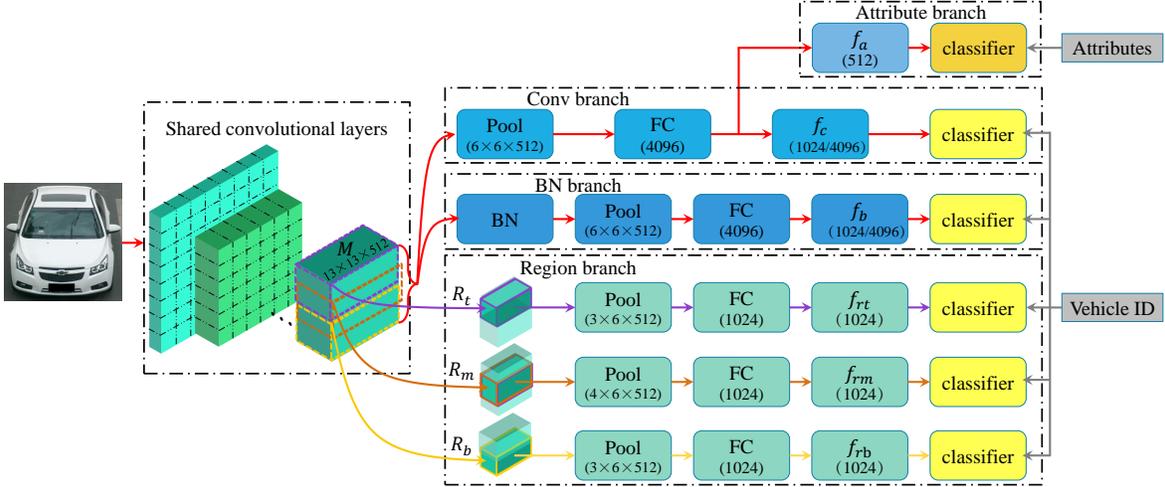


Fig. 2: Structure of the proposed Region-Aware deep Model (RAM), which is composed of several shared convolutional layers and four branches, *i.e.*, Conv branch, BN branch, Regional branch, and Attribute Branch. Each generated feature is trained with an individual classifier. Features are finally concatenated for vehicle instance identification.

tracted on those regions and global features for vehicle Re-ID. This work shares certain similarity with ours. However, our method avoids extra annotations and predictions of key points. Moreover, we achieve better performance using a more concise network structure and less training data.

Although previous works have achieved significant progress, vehicle Re-ID still can be improved from many aspects. As most of previous deep learning based works use the global vehicle image as input, their learned descriptions tend to depict the global appearance and may lose discriminative power to local details. As shown in Fig. 1, different vehicles sharing the same model and maker is similar in global appearance, making it hard to distinguish them. Compared with the global appearance, local regions could be more discriminative. How to effectively extract and embed regional cues for vehicle Re-ID has not been extensively studied. Moreover, fine-grained categorization and vehicle attributes could be helpful in distinguishing vehicles. Although those tasks have been extensively studied in recent years [2, 1], most of vehicle Re-ID works have not considered to leverage vehicle categories or attributes to help vehicle Re-ID.

To utilize region and attribute cues for vehicle Re-ID, we propose a Region-Aware deep Model (RAM) illustrated in Fig. 2. RAM is composed of four branches sharing several convolutional layers. “Conv branch” learns global features from the whole input image. “BN branch” is modified on “Conv branch” with embedding a Batch Normalization (BN) [15] layer to generate complementary global features. “Region branch” learns regional features from three overlapping local regions. “Attribute branch” uses color and model cues to jointly train the model and learn semantic attributes. Features in the four branches are finally concatenated for vehicle Re-ID. As each local region corresponds to a part of the

vehicle, RAM encourages the deep model to learn discriminative features for different local regions. RAM also fuses more cues during the training stage and results in more discriminative global and regional features.

We evaluate our methods on two large-scale vehicle Re-ID datasets. Experimental results show that our methods achieve promising performance in comparison with recent works. The contributions of this work can be summarized into two aspects, *i.e.*,

- We propose a Region-Aware deep Model to jointly learn deep features from both the global appearance and local regions. The learned features are more discriminative to detailed local cues on vehicles than previous global ones.
- Color and model cues are additionally used to jointly train the deep model. The final concatenated feature achieves promising performance in comparison with recent ones.

2. PROPOSED METHOD

Our network structure is illustrated in Fig. 2. Given an input vehicle image, a set of features are generated by RAM. Specifically, five shared convolutional layers generate feature maps M . Then, M is fed into four branches to generate different features, *i.e.*, f_c by Conv branch, f_b by BN branch, f_a by Attribute branch, f_{rt} , f_{rm} , and f_{rb} by Regional branch from the top, middle and bottom regions of vehicles, respectively. Conv and BN branches generate global feature f_c and f_b from the whole feature maps, respectively. Specially, BN branch adds a Batch Normalization operation [15] to the Conv branch to learn complementary global features. Region branch first

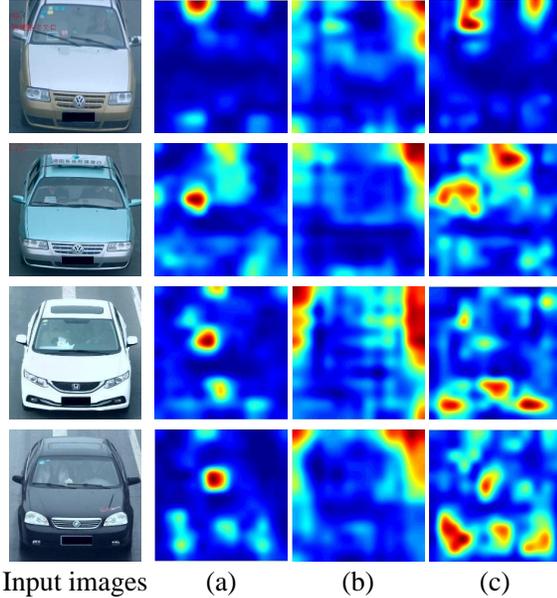


Fig. 3: Responses of feature maps generated by different branches. (a), (b), and (c) show the responses of feature maps generated by Conv branch, BN branch, and Region branch, respectively.

divides feature maps into three overlapped regions denoted as R_t for top, R_m for middle and R_b for bottom, respectively. And then three sets of full connected layers are used to generate regional features f_{rt} , f_{rm} and f_{rb} from corresponding regions. Attribute feature f_a is learned in the Attribute branch. In the following parts, we present the details of the network structure and model training.

2.1. Global Features Extraction

Conv branch first pools the feature maps M into size $6 \times 6 \times 512$ and then uses two Fully Connected (FC) layers to generate feature f_c . f_c is trained with vehicle IDs in a classification task. As discussed in [10], this network structure and training strategy encourage the network to locate and focus on some regions that are discriminative in vehicle classification. In other words, the local regions effective for minimizing the classification loss will be located. The corresponding feature maps M learned by Conv branch would show higher activation values on those regions. As shown in Fig. 3 (a), the highly activated regions cover the distinct regions on the vehicles, thus would be important for vehicle classification.

Besides the regions highlighted on M , other regions may also be useful for vehicle Re-ID. To make the model focus on more and larger contextual regions, we design a BN branch following Yao *et al.* [10], as illustrated in Fig. 2. A BN layer is embedded between M and pooling layer to generate new feature maps M_b . Then two FC layers are used to generate feature f_b . Similarly, a classification task based on vehicle

IDs is finally used to train the BN branch.

As discussed in [10], BN operation tends to depress the highly activated local regions on the feature maps and increase the visibility of the other regions. This enables the BN branch to depict extra contextual cues in addition to the ones captured by the Conv branch. Some examples of M_b after BN operation are shown in the Fig. 3 (b). It's clear that M_b depicts larger contextual regions. Fig. 3 shows that the Conv branch and BN branch depict different cues on the vehicle images, thus may produce complementary global features. We show their complementary in Sec. 3.3.

2.2. Local Features Extraction

As shown in Fig. 1, the differences among similar vehicles may lie on some local regions. We hence design a Region branch to generate regional features.

Firstly, Region branch evenly divides feature maps M into k overlapped local regions from top to bottom, as illustrated in Fig. 2. We experimentally set k as 3, which could be a reasonable trade-off between network complexity and feature performance. These local regions are denoted as: R_t for top, R_m for middle and R_b for bottom, respectively. Each of those regions only corresponds to a part of the whole vehicle, *e.g.*, R_t corresponds to car roof and the top of windshield, and R_b may correspond to head lights. We evenly divide the vehicle image because most of the vehicle images are well-aligned. We use the overlapped regions to enhance the robustness of learned features to possible misalignments or viewpoint variations. A pooling layer is embedded after each region. Then FC layers are applied to generate regional features from each of them, *i.e.* f_{rt} from R_t , f_{rm} from R_m and f_{rb} from R_b . Finally, a classification task with vehicle ID labels is used to supervise each regional feature learning.

During the training of each branch, FC layers are updated to identify vehicles only with a part of feature maps as input. This procedure enforces the network to extract discriminative details in each region. The regional branch thus has potential to discover more local details than the Conv branch. Some examples of feature maps trained with Region branch are shown in Fig. 3 (c). It's clear that more discriminative local regions can be identified than the feature maps of Conv branch. It is reasonable to infer that, cues on those local regions will be conveyed in the resulting regional features.

2.3. Attribute Features Extraction

Attributes like models, makers, colors, *etc.*, can be regarded as mid-level descriptions to vehicles. Compared with visual features, attributes are more robust to variations of appearance caused by the changes of viewpoints, illuminations, backgrounds, *etc.* Therefore, attribute could be complementary to visual features extracted on global and regional images. We thus use attributes to learn features for vehicle Re-ID.

Attribute prediction can be regarded as an easier identification task than the fine-grained vehicle identification. Therefore, we learn attribute features from the Attribute branch for vehicle Re-ID. As illustrated in Fig. 2, Attribute branch takes the output of the first FC layer in Conv branch as input. Attribute feature f_a is then generated by a FC layer. Finally, attribute feature are learned in an attribute classification task. Compared with directly learning attribute features from input images, this strategy introduces less parameters and makes the training procedure easier.

2.4. Training

Every branch in RAM is trained with an individual classification task with softmax loss. The RAM is optimized in multiple classification tasks, and the overall objective function can be formulated as:

$$L(\Theta) = \ell_{conv} + \lambda_1 \ell_{BN} + \lambda_2 \ell_{re} + \lambda_3 \ell_{att}, \quad (1)$$

where Θ denotes the parameters in the deep model. ℓ_{conv} , ℓ_{BN} , ℓ_{re} , and ℓ_{att} denote the classification loss in Conv, BN, Region and Attribute branch, respectively. λ_1 , λ_2 and λ_3 denote the weights for corresponding loss. Note that, ℓ_{re} is composed of three equally weighted classification losses on different regions.

Training the four branches from scratch could be hard to converge. Instead, we train the model in a step-by-step manner. We first train a model only having the Conv branch. The other branches, *i.e.*, BN, Region and Attribute branches are added orderly. The convolutional layers will be shared by different branches and fine-tuned in multiple classification tasks.

As shown in Fig. 2, the proposed RAM is wide and deep, allowing to utilize multiple supervisions for model training. The resulting features depict vehicles from different aspects. For example, the global and regional features depict the discriminative visual cues. The attribute features depict the attributes and would be more robust to appearance variations and noises. Performance of the learned features will be tested in Sec. 3.3.

3. EXPERIMENTS

3.1. Datasets

We evaluate our method on two large-scale datasets for vehicle Re-ID: *VeRi* [14, 4] and *VehicleID* [3]. We first evaluate contributions of each branch in RAM, then make comparison with state-of-the-arts. Details of the two datasets are given as follows:

VeRi contains over 50,000 images of 776 vehicles captured by 20 surveillance cameras. Images are annotated with IDs, types and colors. There are 9 types, such as sedan, bus and truck, and 10 colors, such as red, black and orange. 576 vehicles (37,778 images) are used for training and others are

used for testing. 1,678 images from the testing set are selected as queries and others are used as gallery images.

VehicleID contains 221,763 images of 26,267 vehicles. All of the images are labelled with IDs. A part of vehicles are labelled with colors and vehicle models. 250 models and 7 colors are annotated as attributes. 13,134 vehicles are selected for training and the others are selected for testing. Three subsets are extracted from the testing set with 800, 1600, and 2400 vehicles, respectively. In each subset, an image of each vehicle is randomly selected as gallery image, resulting in 800, 1600, and 2400 gallery images respectively.

3.2. Implementation Details

Proposed RAM is implemented with caffe [16]. We design the structure of RAM based on VGG_CNN_M [17] and VGG_CNN_M_1024 [17] pre-trained on *ImageNet* [18] for *VeRi* and *VehicleID*, respectively for fair comparisons with previous methods. All of the loss weights are set to 1 in Eq. 1. The base learning rate is 0.001 and decreases by multiplying 0.1 after 10 epochs. The size of mini-batch is set to 64. Images are warped to 224×224 for both training and testing. Feature maps M is of size $13 \times 13 \times 512$. In Region branch, the size of each regions is set as $7 \times 13 \times 512$. The size of overlapped region between neighboring regions is $4 \times 13 \times 512$.

We use the same evaluation criterion with previous literatures to evaluate the performance. On *VeRi*, the performance is evaluated by mean Average Precision (mAP), Top-1 and Top-5 following [14, 5]. On *VehicleID*, Top-1 and Top-5 are used to evaluate the performance following [3, 6].

3.3. Evaluation of RAM

In this section, we show the improvements gained by each proposed branch in RAM. The evaluation is conducted in four steps:

Step-1 first trains the *baseline* model only having the Conv branch.

Step-2 adds the BN branch to the baseline model. Model trained in this step is denoted as *BN*.

Step-3 further adds the Region branch to model *BN*. Model trained in this step is denoted as *BN+R*.

Step-4 adds Attribute branch to model *BN+R*. This final model is denoted as *RAM*.

Each step introduces a new branch for feature learning. Therefore, we test the concatenated feature in each step on *VeRi* and *VehicleID*, respectively.

3.3.1. Performance on VeRi

The experimental results on *VeRi* are summarized in Table 1. Step 1 learns the baseline global feature f_c . Step 2 jointly learns features f_c and f_b . As shown in Table 1, $[f_c; f_b]$ performs better than f_c and f_b . This indicates that f_b extracted by BN branch is an effective complementary to f_c . It is also

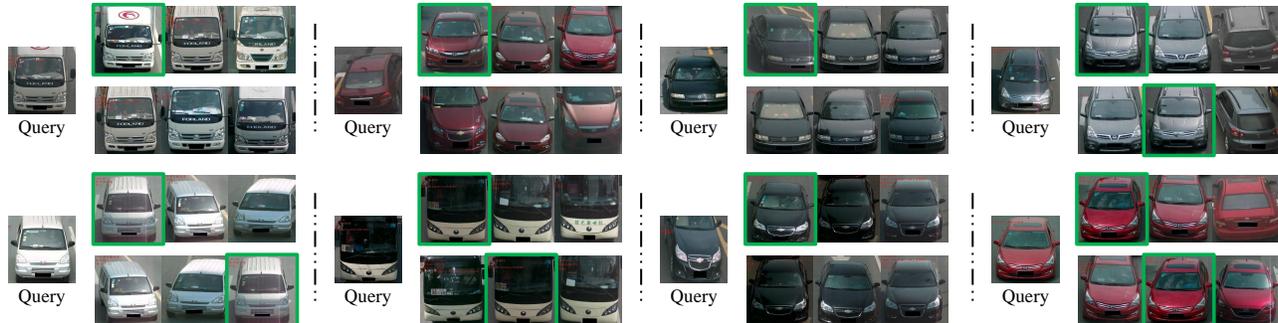


Fig. 4: Examples of returned images on *VehicleID*. In each example, the first and second row show the top 3 images returned by RAM and baseline model, respectively. True positives are annotated by green bounding boxes.

Table 1: Performance of features learned by different models on *VeRi*.

Models	Features	mAP	Top-1	Top-5
<i>baseline</i>	f_c	0.550	0.848	0.931
<i>BN</i>	f_c	0.559	0.854	0.927
	f_b	0.531	0.839	0.930
	$[f_c; f_b]$	0.581	0.871	0.940
<i>BN+R</i>	f_c	0.556	0.852	0.925
	$[f_c; f_b]$	0.598	0.873	0.937
	$[f_c; f_b; f_{rt}]$	0.601	0.883	0.941
	$[f_c; f_b; f_{rm}]$	0.590	0.874	0.934
	$[f_c; f_b; f_{rb}]$	0.593	0.868	0.933
	$[f_c; f_b; f_r]$	0.609	0.887	0.941
<i>RAM</i>	f_c	0.563	0.861	0.925
	$[f_c; f_b]$	0.604	0.869	0.937
	$[f_c; f_b; f_r]$	0.613	0.885	0.942
	$[f_c; f_b; f_r; f_a]$	0.615	0.886	0.940

Table 2: Performance of concatenated features learned by different models on *VehicleID*.

Models	Top 1			Top 5		
	Small	Medium	Large	Small	Medium	Large
<i>baseline</i>	0.694	0.673	0.632	0.892	0.820	0.795
<i>BN</i>	0.722	0.705	0.666	0.904	0.853	0.832
<i>BN+R</i>	0.747	0.720	0.674	0.908	0.863	0.842
<i>RAM</i>	0.752	0.723	0.677	0.915	0.870	0.845

interesting to observe that f_c learned in the *BN* model performs better than the f_c learned in the baseline model. This implies the advantage of network training in multiple tasks.

With *BN+R* model learned in step 3, we test different combinations of regional features. f_r denotes the combination of three regional features, *i.e.*, $f_r = [f_{rt}; f_{rm}; f_{rb}]$. As shown in Table 1, only fusing single regional feature, *e.g.*, f_{rb} or f_{rm} , might degrade the performance because region segmentation could be sensitive to pose variations and misalignment errors. Jointly fusing all regional features, *i.e.*, $[f_c; f_b; f_r]$, achieves better performance than the other feature

combinations. For example, $[f_c; f_b; f_r]$ achieves the mAP of 0.609, significantly better than the 0.601, 0.590, and 0.593 of $[f_c; f_b; f_{rt}]$, $[f_c; f_b; f_{rm}]$, and $[f_c; f_b; f_{rb}]$, respectively. Note that, the concatenated feature $[f_c; f_b]$ of *BN+R* performs better than that of *BN*. This also indicates the advantage of joint feature learning.

With *RAM* learned in the step 4, we compare different concatenations of the resulting features from each branch. Compared with the $[f_c; f_b]$ generated in previous steps, $[f_c; f_b]$ in *RAM* performs the best. The final feature $[f_c; f_b; f_r; f_a]$ achieves the best performance in mAP among all of the features in Table 1. It exhibits a substantial improvement of 6.5% in mAP over the baseline feature.

3.3.2. Performance on VehicleID

We show the experimental results on *VehicleID* in Table 2. It's clear that, adding more branches constantly improves the performance. Similar to the observations on *VeRi*, the concatenated feature of *RAM* shows the best Rank-1 and Rank-5 accuracies. Some examples of retrieved images by *RAM* and baseline are shown in Fig. 4. In those examples, feature generated by *RAM* substantially outperforms the baseline global feature. It can be inferred that, *RAM* is more effective in distinguishing visually similar vehicles. It could also be observed that, *RAM* is also more robust to viewpoint variations, as shown in the second example of Fig. 4. Therefore, we can conclude that *RAM* extracts discriminative features for vehicle Re-ID.

3.4. Comparison with State-of-the-art Methods

On *VeRi*, we compare *RAM* with several recent methods including FACT [14], FACT+Plate-SNN+STR [4], SCPL [5], OIF [6], and OIF+ST [6]. Performance comparison is summarized in Table 3. On *VehicleID*, the compared works includes VGG+T [3], VGG+CCL [3], Mixed Diff+CCL [3] and OIF [6], as shown in Table 4. In Table 3 and Table 4, we use “*” to denote the model that uses ResNet [19] as base network, which is deeper than our base network VGG_CNN_M.

Table 3: Comparison with recent works on *VeRi*.

Method	mAP	Top-1	Top-5
FACT [14]	0.199	0.597	0.753
FACT+Plate-SNN+STR [4]	0.278	0.614	0.788
SCPL* [5]	0.583	0.835	0.900
OIF+ [6]	0.480	0.659	0.877
OIF+ST+ [6]	0.514	0.683	0.897
RAM	0.615	0.886	0.940

Table 4: Comparison with recent works on *VehicleID*.

Method	Top 1			Top 5		
	Small	Medium	Large	Small	Medium	Large
VGG+T [3]	0.404	0.354	0.319	0.617	0.546	0.503
VGG+CCL [3]	0.436	0.370	0.329	0.642	0.571	0.533
Mixed Diff+CCL [3]	0.490	0.428	0.382	0.735	0.668	0.616
OIF+ [6]	-	-	0.670	-	-	0.829
RAM	0.752	0.723	0.677	0.915	0.870	0.845

We use superscript “+” to denote methods that are trained on a bigger training set, which is merged by the training data of *VeRi*, *VehicleID*, *CompCars* [1] and *BoxCars21K* [20].

It can be observed that, RAM achieves the best performance on two datasets compared with others. Compared with SCPL [5], which uses a deeper network and extra spatial-temporal cues, RAM outperforms it by 3.2% in mAP on *VeRi*. Compared with OIF [6] that uses more training data and a more complex structure, RAM outperforms it by 13.5% in mAP on *VeRi* and by 0.7% in Top-1 accuracy on *VehicleID*.

4. CONCLUSION

This paper presents a Region-Aware deep Model (RAM) for vehicle Re-ID task. In addition to global features, RAM includes a Region branch to extract regional features from three overlapped local regions. This encourages the deep model to focus on more details in local regions and results in more discriminative features. We also jointly train an Attribute branch to generate attribute features, which are potential to be more robust to viewpoint variations. Experiments on two large-scale vehicle datasets demonstrate that RAM extracts discriminative features and achieves promising performance.

Acknowledgements This work is supported by National Science Foundation of China under Grant No. 61572050, 91538111, 61620106009, 61429201, and the National 1000 Youth Talents Plan, in part to Dr. Qi Tian by ARO grant W911NF-15-1-0290 and Faculty Research Gift Awards by NEC Laboratories of America and Blippar.

5. REFERENCES

- [1] L. Yang, P. Luo, C. Change Loy, and X. Tang, “A large-scale car dataset for fine-grained categorization and verification,” in *CVPR*, 2015.
- [2] J. Krause, M. Stark, J. Deng, and L. Fei-Fei, “3d object representations for fine-grained categorization,” in *ICCV*, 2013.
- [3] H. Liu, Y. Tian, Y. Yang, L. Pang, and T. Huang, “Deep relative distance learning: Tell the difference between similar vehicles,” in *CVPR*, 2016.
- [4] X. Liu, W. Liu, T. Mei, and H. Ma, “A deep learning-based approach to progressive vehicle re-identification for urban surveillance,” in *ECCV*, 2016.
- [5] Y. Shen, T. Xiao, H. Li, S. Yi, and X. Wang, “Learning deep neural networks for vehicle re-id with visual-spatio-temporal path proposals,” in *ICCV*, 2017.
- [6] 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.
- [7] C. Su, J. Li, S. Zhang, J. xing, W. Gao, and Q. Tian, “Pose-driven deep convolutional model for person re-identification,” in *ICCV*, 2017.
- [8] L. Wei, S. Zhang, H. Yao, W. Gao, and Q. Tian, “Glad: Global-local-alignment descriptor for pedestrian retrieval,” in *ACM MM*, 2017.
- [9] L. Wei, S. Zhang, W. Gao, and Q. Tian, “Person transfer gan to bridge domain gap for person re-identification,” *arXiv preprint arXiv:1711.08565*, 2017.
- [10] H. Yao, S. Zhang, Y. Zhang, J. Li, and Q. Tian, “One-shot fine-grained instance retrieval,” *arXiv preprint arXiv:1707.00811*, 2017.
- [11] X. Zhang, F. Zhou, Y. Lin, and S. Zhang, “Embedding label structures for fine-grained feature representation,” in *CVPR*, 2016.
- [12] F. Schroff, D. Kalenichenko, and J. Philbin, “Facenet: A unified embedding for face recognition and clustering,” in *CVPR*, 2015.
- [13] Y. Sun, Y. Chen, X. Wang, and X. Tang, “Deep learning face representation by joint identification-verification,” in *NIPS*, 2014.
- [14] X. Liu, W. Liu, H. Ma, and H. Fu, “Large-scale vehicle re-identification in urban surveillance videos,” in *ICME*, 2016.
- [15] S. Ioffe and C. Szegedy, “Batch normalization: Accelerating deep network training by reducing internal covariate shift,” *arXiv preprint arXiv:1502.03167*, 2015.
- [16] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell, “Caffe: Convolutional architecture for fast feature embedding,” in *ACM MM*, 2014.
- [17] K. Chatfield, K. Simonyan, A. Vedaldi, and A. Zisserman, “Return of the devil in the details: Delving deep into convolutional nets,” *arXiv preprint arXiv:1405.3531*, 2014.
- [18] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, “Imagenet: A large-scale hierarchical image database,” in *CVPR*, 2009.
- [19] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” *arXiv preprint arXiv:1512.03385*, 2015.
- [20] J. Sochor, A. Herout, and J. Havel, “Boxcars: 3d boxes as cnn input for improved fine-grained vehicle recognition,” in *CVPR*, 2016.