Progressive Data Mining and Adaptive Weighted Multi-Model Ensemble for Vehicle Re-Identification

Yongli Sun* , Wenpeng Li*, Hua Wei*, Longtao Zhang*, Jiahao Tian, Guangze Sun, Gang Wang,
Junliang Cao, Zhifeng Zhao, Junfeng Ding
Nanjing Fiberhome Tiandi CO., LTD
Nanjing, Jiangsu, China
sunyongli@fiberhome.com liwp@fiberhome.com

Abstract

In this paper, we introduce our solution to the vehicle re-identification (vehicle ReID) track2 in AI City Challenge 2021. As the key point of intelligent Traffic System, vehicle ReID has been a challenging task due to the higher intra-class and inter-class errors which are owing to variable vehicle orientation, camera and lighting. To reduce this error, at first, we innovatively propose a progressive data mining method to obtain more valid data from testing set. Then, we use the image to the mean of each tracklet method in the matching stage which can ensure the precision of image matching by reducing the error with the information of tracklets. Besides, we propose an adaptive weighted ensemble method which effectively improve the model capability. Finally, our method achieves 0.6533 in the mAP score which yields 4th place in the competition.

1. Introduction

Vehicle ReID has become an important research hotspot and it has been widely used in the intelligent traffic system and smart city, such as vehicle monitoring and so on.

Vehicle ReID aims to identify the target vehicle in many images or videos across cameras. The research of vehicle ReID mainly includes sensor-based method, manual selection features-based method and deep learning-based method, where the sensor-based method has high requirements for hardware devices which is not suitable for daily use, the manual selection features-based method is complex and has poor performance. By comparison, deep learning-based method can automatically learn the high level features and effectively represent the input image. So, the first two methods are gradually abandoned by researchers. The deep learning-based method has attracted more and more attention.

With the enrichment of neural network theory, many effective networks [10, 12] has been proposed which can learn the discriminant features, the loss functions [13] that can effectively constrain sample learning process have also been proposed and many tricks [20] which are beneficial to the process of model training and testing have been put forward.

In this paper, our method focuses on obtaining the robust feature and reducing the biases of variable camera, vehicle orientation and lighting. In summary, our contributions are:

- We innovatively propose a progressive data mining method for vehicle ReID which is beneficial to obtain more valid training data and improve the robustness of model.
- We replace the method of image to image with the image to the mean of tracklet, which effectively reduces the influence of noise images in the matching process.
- We propose an adaptive weighted model ensemble method which improves the matching accuracy by combining the advantages of different models.

2. Related Works

In recent years, deep learning-based vehicle ReID methods has achieved great progress such as CNN-based features. These approaches outperform all the previous baselines using handcrafted features.

Like other branches of computer vision, large-scale and high-quality datasets is essential for deep learning based vehicle ReID. Liu et al. [1] proposed the VeRi776 dataset, which is the first large-scale benchmark for vehicle ReID. Liu et al. [2] introduced VehicleID dataset, which includes multiple images of the same vehicle from various cameras.

Despite there is different details between person ReID and vehicle ReID, it still provides a lot of insights. Sun et al. [3] introduces SVDNet in person ReID, improve the model accuracy with batch normalization neck. Luo et al. [4] introduces a strong baseline exploits the transformer for both person ReID and vehicle ReID, which achieves comparable performance with CNN-based methods.

On the other hand, some methods focus on exploiting data processing methods to make further improvement on the ReID results. Zheng et al. [5] explored the effectiveness of the synthetic data in vehicle ReID by using data generation methods. Yao et al. [6] augment datasets with different orientation by apply a graphic engine. Zhuge et al.

* Equal Contribution.

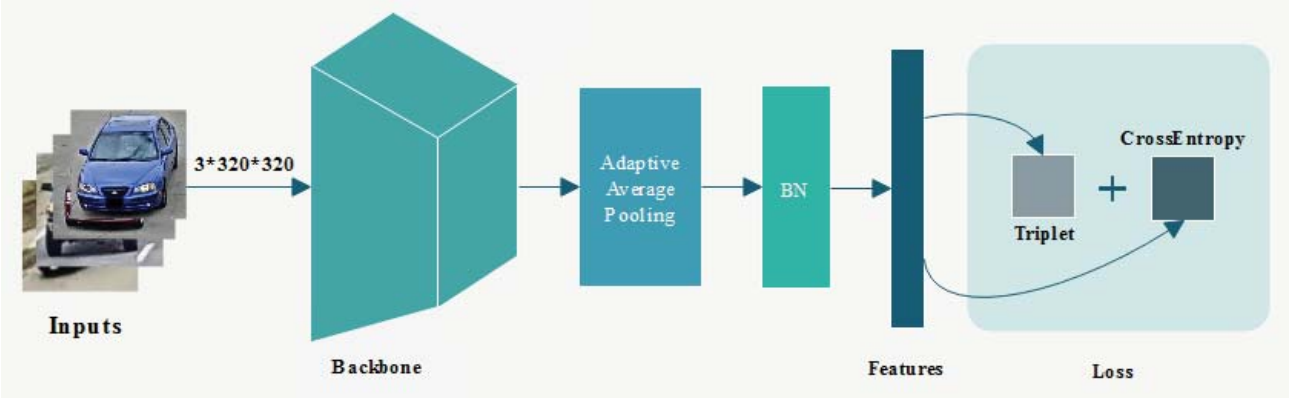

Figure 1: The structure of our model.

[7] introduces random shrink and background substitution augments to improve the robustness of model when the distribution between training set and test set is largely different.

3. Methods

3.1. Model

Data augmentation. Data augmentation is acting an extremely important role to improve the models accuracy and avoid overfitting. One of the important data augmentation works in our work is random shrink [7] where a random scaling factor is generated, then we downsample large images in the training set to improve the performance. We also increase the robustness of features by using random erasing [8] and color jitter [23].

Backbone. The appropriate baseline model is crucial to the stage of post-processing. Some SOTA networks pretrained on ImageNet [9] containing ResNeSt101 [10], ResNeXt101 [11] and ResNet101_ibn_a [12] are used as the backbone module in our model. The structure of our model is shown in Figure 1.

Adaptive average pooling. Adaptive average pooling is used in our model to get the global feature after the backbone module.

Loss. As the widely-adapted loss functions, the Cross-entropy loss and the Triplet loss [13] are adopted as the optimization objective in our model.

The cross-entropy loss could be formulated as:

$$L_{ce} = -\sum_{i=1}^{N} p_i \log\left(p_i\right) \tag{1}$$

Where p_i is the ground truth label of input image. p_i is the predicted probability.

Triplet loss is the soft-margin version as follow:

$$L_{tri} = \log\left[1 + e^{\left(\|f_a - f_p\|_2^2 - \|f_a - f_n\|_2^2 + m\right)}\right] \tag{2}$$

where m is the margin. Hard example mining is used for triplet loss.

Optimizer. In order to obtain a better performance, ADAM_GCC [14] has been taken as the optimizer in our work, which not only directly operates the gradients vectors to have zero mean, but improves the Lipschitzness of the loss function to make the training process more efficient and stable.

3.2. Progressive Data Mining

In AI City Challenge 2021 of track2, the testing data is allowed to be used for unsupervised learning. We all know that the size and correctness of dataset is necessary to the deep learning. We argue that the mining data with certain accuracy is beneficial to improve the robustness of the model. In order to obtain more valid identities, we propose the progressive data mining (PDM) method. In the method of progressive data mining, there are two assumptions, one is that the same vehicle has the greater chance to appear under the close cameras. And the other one is that, with the increase of correct mining samples, the ability of model will be gradually improved.

In query and gallery set, there is the phenomenon of the same vehicle is cut to several sub-clusters. So, we firstly cluster the query images and gallery track-lets respectively.

Query cluster. To reduce the cluster error, we do the query cluster process on a finer dimension instead of the original query set which is denoted as $Q = \{q_1, q_2, ..., q_m\}$. At first, we extract the red, green and blue images with the HSV rules [22] from the original query set, in this way, the query set is cut to four blocks including red, green, blue and other blocks, here the query set is denoted as $Q = \{block_{red}, block_{green}, block_{blue}, block_{others}\}$. And then the red, green and blue blocks are clustered respectively, the blocks of red, green and blue have been subdivided into several subblocks. Finally, the query set is clustered on the

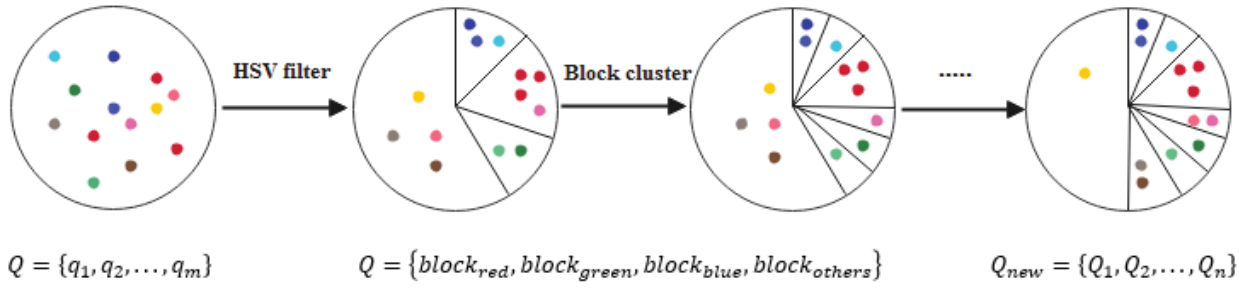

Figure 2: The procedure of query cluster

subblock dimension and we can obtain the final clustering result of query set. Here, the query set is denoted as $Q_{new} = \{Q_1, Q_2, ..., Q_n\}$,where Q_n is the $n-th$ cluster . The process of query cluster is shown as Figure 2.

Gallery cluster. The gallery set consists of multi-tracklets from different vehicles. In order to avoid the phenomenon that same vehicle identity is cut and maintain the integrity of same vehicle. Each tracklet is regard as a block which is different from the operation on the query set. For gallery set, we represent each block by the mean feature of the whole images in each block. Then the cluster is applied on those mean features. In this way, different tracklets of the same vehicle can be merged to a certain extent and which facilities to mine the new vehicle data.

Matching. After the query cluster and gallery cluster, the phenomenon of the same vehicle is cut in the query and gallery set has been solved to some extent and then we begin to mine the new identity from the gallery set. The steps of this process are as follows:

1) Extract the features of query and gallery images with the model trained from the merged dataset Q_{merged} . In different iteration periods, the composition of merged dataset is different.

$$Q_{merged} = \begin{cases} Q_{original} \cup Q_{simulation-X}, iter = 1 \\ Q_{original} \cup Q_{simulation-X} \cup Q_{new}, iter > 1 \end{cases} \quad (3)$$

Where $Q_{original}$ represents the original training dataset, $Q_{simulation-X}$ is the partial simulation dataset picked from the original simulation dataset and Q_{new} is the new mining data which is cumulative after each iteration, because the new identity and the new matched images for the same vehicle will be picked on the basis of last Q_{merged} .

2) Complete the cluster operation of query and gallery respectively as described in Section 3.2.1 and 3.2.2, here, the cluster results of query set is denoted as $Q_{query} = \{q_{c1}, q_{c2}, ..., q_{cs}\}$ and the cluster results of gallery set is denoted as $Q_{gallery} = \{g_{c1}, g_{c2}, ..., g_{ck}\}$.

3) Calculate mean feature $f_Q = \{f_{q_{c1}}, f_{q_{c2}}, ..., f_{q_{cs}}\}$ and $f_G = \{f_{g_{c1}}, f_{g_{c2}}, ..., f_{g_{ck}}\}$ of each cluster for query and gallery.

4) Get the matched distance matrix $Dist(f_Q, f_G)$ based on approximate nearest neighbor (ANN) method [19].

5) The matched threshold zone is defined as $[d_{lower}, d_{upper}]$ which are from the mean and variance of the minimum distance of each row in $Dist(f_Q, f_G)$, for pairs which are the minimum distance from each other, if their distance is in the matched threshold zone. Add those pairs to set P .

6) In order to ensure the accuracy of mining data, based on the first assumption, we filter the matched pairs from different regions with the prior camera information in set P . After several iterations, the camera constraints will be relaxed gradually which is controlled by the proportion of images from close cameras for each matched pair.

7) Repeat the above steps until the merged dataset can't bring higher benefits to the model.

With the progressive data mining, there will be some new vehicle identities. So, the scale of training set is expanded effectively and the model will see more and more vehicle identities, finally, the robustness of the model is improved.

3.3. Image to The Mean of Tracklet

As we all know, some factors such as camera and illumination have a great influence on the performance of the same vehicle image. Even in a tracklet, there is a big difference for some images. In order to reduce the influence of noise on the matching results, we use the mean feature of the whole images in each tracklet as the track feature and the matching operation is going between the query feature and the track mean features.

3.4. Adaptive Weighted Model Ensemble

For the same input, different model focus on the different content and learn the different level features, and we can get a more complete representation of the same input based on

multiple models. Therefore, we tend to choose several different models to describe the input image. Based on the above facts, it's the key that how to effectively combine multiple models. Our experiment proved that directly combine multi-model is not the best way to integrate the advantages of each model. Instead, adaptive weighted model ensemble can effectively improve the final matching accuracy, where the different weight parameters are assigned to each model by the auto search method.

In the method of adaptive weighted model ensemble, the ensembled model is defined as follows:

$$M_{ensemble} = \sum_{i=1}^{n} \lambda_i M_i \quad (4)$$

Where the parameter n represents the number of used model. The λ_i represents the weight of model M_i.

4. Experiments

4.1. Dataset

Similar to the AI City Challenge in 2020, the Track2 training dataset was composed of both real-world data and synthetic data. The use of synthetic data was encouraged, unsupervised learning strategies on the testing set is also permitted.

Real-world data. The real-world data comes from The CityFlow V2-ReID which is updated from CityFlow dataset [15, 16]. The dataset contains 85058 images come from 46 cameras. It is split into 52717 images for training and 31238 images for testing. An additional 1103 images are used as queries. The single-camera tracklets are provided on both training and testing sets.

Synthetic data. The synthetic data comes from VehicleX dataset [6] which is a large-scale public 3D vehicle dataset, including 192150 images of 1362 vehicles. In addition, some extra labels are also provided for training color and direction classification model.

Validation Data. In order to evaluate our proposed model, we split the training set of CityFlow V2-ReID into training set and validation set, where 42059 come from 352 objects identities for training and 10658 come from other 88 identities for validation with 441 query images and 10217 gallery images. For convenience, we called them as V2-T and V2-V.

4.2. Implementation Details

All input images are resized to 320x320. We use three different backbones: ResNeSt101, ResNet101_ibn_a and ResNeXt101_ibn_a. As for data augmentation, we adopt a series of data augmentation methods in training, including random horizontal flip, padding, random crop, color jitter and random erase. To meet the requirement of triplet loss, we set 8 identities and 8 images of each identity for ResNet101_ibn_a, 8 identities and 6 images of each identity for ResNeSt101 and ResNeXt101_ibn_a due to GPU memory limitation. All models are trained on a single TITAN RTX in total 25 epochs. ADAM_GCC [14] is applied as the optimizer and initial learning rate is set to 3.5e-4. For ResNeSt101, learning rate decay to 3.5e-5 and 3.5e-6 at 10th and 20th epochs. For ResNet101_ibn_a and ResNeXt101_ibn_a, learning rate decay to 3.5e-5 and 3.5e-6 at 8th and 15th epochs.

(a) Real-world data

(b) Synthetic data

Figure 3: Examples of real-world data and synthetic data.

4.3. Analysis of Different Training Datasets

We evaluate different training datasets performance on V2-V in Table 1. S100, S150, S200, S250 and S300 means 100 IDs with 14536 synthetic images, 150 IDs with 21763 synthetic images, 200 IDs with 28986 synthetic images, 250

IDs with 36143 synthetic images and 300 IDs with 43262 synthetic images.

Training Dataset	V2-V			
	mAP	r=1	r=5	r=10
Baseline(V2-T)	87.9	96.4	98.0	98.9
V2-T+S100	88.3	96.4	98.0	98.4
V2-T+S150	88.9	96.4	97.7	98.4
V2-T+S200	89.2	97.1	98.6	99.3
V2-T+S250	88.6	97.3	98.2	98.6
V2-T+S300	88.5	97.3	98.9	99.5

Table 1.The results of different training datasets.

More identities provide more knowledge, so the result of V2-T+S200/V2-T+S150/V2-T+S100 dataset is better than Baseline(V2-T), mAP is improved from 87.9 to 89.2 with V2-T+S200. However, with the increase of synthetic data in the training set, the performance of the model is getting worse. The reason is that there is still a big gap between real-world domain and synthetic domain. As shown in Figure 3.

4.4. Comparison of Different Pooling Methods

Method	V2-V			
	mAP	r=1	r=5	r=10
Average pooling	89.2	97.1	98.6	99.3
Adaptive Average Pooling	90.3	97.3	98.4	99.1
Adaptive Concat Pooling	76	94.3	95.7	96.4
GeM [17]	87.8	96.8	98.6	98.9

Table 2. The results of different pooling methods.

In our experiments, we evaluate several popular pooling strategies including adaptive average pooling, average pooling, adaptive concat pooling which improve the performance of model in person-ReID task [21] and GeM. The details are in the Table 2 which demonstrates that adaptive average pooling achieves the best performance in vehicle ReID task.

4.5. Comparison of Re-Ranking Strategies

Re-Ranking	V2-V			
	mAP	r=1	r=5	r=10
Baseline(w/o RR)	90.3	97.3	98.4	99.1
Baseline + II RR [18]	95.4	97.7	98.3	98.7
Baseline + ITM RR	97.2	98.6	98.9	98.9

Table 3. The results of different re-ranking strategies.

In Table 3, RR means re-ranking, II RR means image to image method and ITM RR represents image to the mean of tracklet. The results about without RR, II RR and ITM RR strategies on V2-V are shown in Table 3. The baseline without re-ranking achieves 90.3 mAP. The method of image to image brings almost 5.1 gains from baseline. Image to the mean of tracklet re-ranking reaches 97.2 mAP.

4.6. Data-Mining

Method	CityFlow V2-ReID			
	mAP	r=1	r=5	r=10
Baseline(V2-T+S200)	43.9	58.3	58.8	59.2
Baseline(V2-T+S200+ Data-mining)	58.4	71.3	71.3	71.6

Table 4. The results of data-mining.

With the progressive data mining method, the training data is effectively expanded by merging the new mining data. And the ability of model is improved.

4.7. Comparison of Different Model Ensemble Strategies

Ensemble	CityFlow V2-ReID			
	mAP	r=1	r=5	r=1
Baseline(Data-mining + single model)	58.4	71.3	71.3	71.6
Baseline + Directly Ensemble	60.8	69.1	70.1	71.9
Baseline + Adaptive weighted Ensemble	65.6	76.0	76.0	76.3

Table 5. The results of different model ensemble strategies.

We evaluate different model ensemble strategies on the testing set. The single model with best mAP is ResNet101_ibn_a with 58.4. If we directly combine the multi-models with the same weights, the mAP is improved to 60.8. The mAP reaches 65.6 with adaptive weighted model ensemble method. The results in Table 5 confirm the effectiveness of adaptive weighted ensemble strategy.

4.8. Competition Results

Rank	Team ID	Team name	Score
1	47	DMT	0.7445
2	9	NewGeneration	0.7151
3	7	CyberHu	0.6650
4	**35**	**For Azeroth**	**0.6555**
5	125	IDo	0.6373
6	44	KeepMoving	0.6364
7	122	MegVideo	0.6252
8	71	Aiem2021	0.6216
9	61	CybercoreAI	0.6134
10	27	Janus Wars	0.6083

Table 6. Final results of AI City2021 Track2

Table 6 lists the top-10 ranks and results on vehicle Re-ID test of 2021 AI City Challenge Track2. As shown in this Table, our submission ranks the 4th place on final leaderboard.

5. Conclusion

In this paper, we propose a progressive data mining method to enrich the training dataset. We argue that the mining data with certain accuracy rate is beneficial to the model training and which is verified in our experiments. In addition, we replace the image to image method with image to the mean of each tracklet in the matching process, which effectively reduces the influence of noise images. What's more, we introduce an adaptive weighted model ensemble method which combines the advantages of different models and improve the vehicle ReID performance.

References

[1] Xinchen Liu, Wu Liu, Huadong Ma, and *et al.* Large-scale vehicle reidentification in urban surveillance videos. In Multimedia and Expo (ICME), 2016 IEEE International Conference on, page 16, 2016.

[2] Hongye Liu, Yonghong Tian, Yaowei Wang, and *et al.* Deep relative distance learning: Tell the difference between similar vehicles. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 2167-2175, 2016.

[3] Yifan Sun, Liang Zheng, Weijian Deng, and *et al.* SVDNet for pedestrian retrieval. In Proceedings of the IEEE International Conference on Computer Vision, pages 3800-3808, 2017.

[4] Shuting He, Hao Luo, Pichao Wang, and *et al*. TransReID: Transformer-based Object Re-Identification. arXiv preprint arXiv:2102.04378, 2021.

[5] Zhedong Zheng, Xiaodong Yang, Zhiding Yu, and *et al*. Joint discriminative and generative learning for person re-identification. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 2138-2147, 2019.

[6] Yue Yao, Liang Zheng, Xiaodong Yang, and *et al*. Simulating content consistent vehicle datasets with attribute descent. arXiv:1912.08855, 2019.

[7] Chaoran Zhuge, Yujie Peng, Yadong Li, and *et al*. Attribute-guided Feature Extraction and Augmentation Robust Learning for Vehicle Re-identification. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, pages 618-619, 2020.

[8] Zhun Zhong, Liang Zheng, Guoliang Kang, and *et al*. Random erasing data augmentation. In Proceedings of the AAAI Conference on Artificial Intelligence, pages 13001-13008, 2017.

[9] Jia Deng, Wei Dong, Richard Socher, and *et al*. Imagenet: A large-scale hierarchical image database. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 248-255, 2009.

[10] Hang Zhang, Chongruo Wu, Zhongyue Zhang, and *et al.* ResNeSt: Split-Attention Networks. arXiv: 2004.08955, 2020.

[11] Saining Xie, Ross Girshick, Piotr Dollár, and *et al.* Aggregated Residual Transformations for Deep Neural Networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 1492-1500, 2017.

[12] Jianping Shi, Xingang Pan, Ping Luo, and *et al.* Two at once: Enhancing learning and generalization capacities via ibn-net. In Proceedings of the European Conference on Computer Vision, pages 464-479, 2018.

[13] Alexander Hermans, Lucas Beyer, and Bastian Leibe. In defense of the triplet loss for person re-identification. arXiv preprint arXiv:1703.07737, 2017.

[14] Yong, Hongwei, Jianqiang Huang, Xiansheng Hua, and *et al*. Gradient centralization: A new optimization technique for deep neural networks. In Proceedings of the European Conference on Computer Vision, pages 325-352, 2020.

[15] Milind Naphade, Zheng Tang, Ming-Ching Chang, and *et al*. The 2019 AI City Chanlenge. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, pages 452–460, 2019.

[16] Zheng Tang, Milind Naphade, Ming-Yu Liu, and *et al*. CityFlow: A city-scale benchmark for multi-target multi-camera vehicle tracking and re-identification. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 8797–8806, 2019.

[17] Filip Radenović, Giorgos Tolias, and Ondřej Chum. Fine-tuning cnn image retrieval with no human annotation. IEEE Transaction on Pattern Analysis and Machine Intelligence, vol. 41, no. 7, pages 1655-1668, 2018.

[18] Zhun Zhong, Liang Zheng, Donglin Cao, and *et al.* Reranking person re-identifification with k-reciprocal encoding. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 1318–1327, 2017.

[19] Yongjian Chen, Tao Guan, and Cheng Wang. Approximate nearest neighbor search by residual vector quantization. Sensors, pages 11259-11273, 2010.

[20] Hao Luo. Youzhi Gu, Xingyu Liao, and *et al*. Bag of tricks and a strong baseline for deep person re-identification. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, pages 0, 2019.

[21] Wenpeng Li, Yongli Sun, Jinjun Wang, and *et al.* Collaborative Attention Network for Person Re-identification. 2021 J. Phys.: conf. Ser. 1848, 012074.

[22] Smith A R. Color gamut transform pairs. Computer Graphics Proceedings, Annual Conference Series, ACM SIGGRAPH. New York: ACM Press, 1978:12-19.

[23] Maxim Berman, Herve Jegou, Andrea Vedaldi, Iasonas Kokkinos, and Matthijs Douze. Multigrain: a unified image embedding for classes and instances. arXiv preprint. arXiv:1902.05509, 2019.