Theory and Practice of Science and Technology
2025, VOL. 6, NO. 1, 126-132
DOI: 10.47297/taposatWSP2633-456924.20250601

Exploring Camera Style Adaptation for Person Re-identification Using MobileNetV3 Networks

Weiyi Wang

Physics and Electronic Information Department, YanTai University, Yantai, Shandong, 264005, China

ABSTRACT
In response to the problem that the large number of ResNet50 parameters in the re-ID program for camera style adaptation based on GANs is not suitable for environments with low power consumption and small hardware size, I modified the backbone of the target program using MobileNetV3 , and finally achieved a significant reduction in computing speed and number of parameters at the cost of a small loss of accuracy, and in the process I found that the pre-trained network provided by Torchvision has some problems in the actual replacement of the backbone.

KEYWORDS
MobileNetV3; GANs; Re-ID; Pretrained model; Torchvision

1 Introduction

As a cross-camera data retrieval task, person re-ID systems are often applied to cross-surveillance camera and cross-UAV camera surveillance of targets. This means that the system often works on small embedded devices, which have high requirements for low power consumption and small hardware size. MobileNet, as a convolutional network proposed by Microsoft, creatively uses depth-separable convolution to dramatically reduce the number of parameters of the network and increase the computational speed, so that it can work on a large number of embedded devices and hardware environments with only CPU. Therefore, in this paper I tackle this challenge by introducing MobileNetV3 to modify the camera style adaptation based on person re-identification program. The experimental results demonstrate that the use of MobileNetV3 can effectively reduce the number of parameters and operations of the model and increase the speed of operations while slightly reducing the accuracy.

2 Related work

2.1 GAN

Generative Adversarial Networks (GANs) were proposed in 2014 by Ian Goodfellow et al. In the paper[1] Ian Goodfellow et al. authors describe in detail the principles of GANs, its advantages, and its application to image generation. In 2015, Emily Denton et al.[2] proposed the Laplace Pyramid, which evolved from a single convolutional neural network to a "per-layer pyramid" (i.e., a layer of convolutional neural networks with different specific resolution) has a layer of convolutional neural networks, each of which generates a sharper image than the previous layer and uses the output as input for the next layer. In 2016, Reed et al.[3] published the paper *Generative Adversarial Text to Image Synthesis*, which describes how GANs can be used to perform text to Image Transformation. In the same year, the Cortex research team developed a new loss function for GANs to recover texture and small-grain details for images that have been drastically downsampled, which is the super-resolution technique. In the development history of GANs, there are also a large number of different architectures, such as DCGAN[4], StyleGAN[5], BigGAN[6], StackGAN[7], Pix2pix[8], CycleGAN[9], and so on. For the CycleGAN used in this paper, which is an architecture suitable for style migration, is an unsupervised image transformation technique proposed by Jun-Yan Zhu et al. in 2017.

2.2 MobileNet

The most prominent contribution of MobileNet[10], proposed by AG Howard et al. in 2017, is the proposed depth-separable convolution . It focuses on lightweight platforms like embedded and mobile devices, sacrificing a bit of accuracy in exchange for drastically reducing the number of parameters and operations of the model. MobileNetV1 uses depth-separable convolution and 1x1 convolution, which drastically improves the speed with a small amount of reduction in accuracy. MobileNetV2[11] uses linear bottlenecks and inverted residuals on top of v1. Further, the MnasNet network builds on the MobileNetV2 network by introducing a lightweight attention module based on squeezing and excitation in the bottleneck structure. MobileNetV3[12] further optimizes the MnasNet network. It redesigned the time-consuming layers to use h-wish instead of relu6 as the activation function, while it used the netadapt algorithm to obtain

the optimal number of filters in the extension layer and output channels in the bottleneck layer.

2.3 Re-ID

Person re-identification was first introduced in 1997 by Huang and Russell et al.[13] In 2005, Wojciech Zajdel, Zoran Zivkovic, and Ben J. A. Kr¨ose first introduced the term "Person Re-identification" in the context of multicameral visual tracking research[14]. The term "person re-identification" was first introduced in the study of multimetric visual tracking. In 2006, Gheissari[15] proposed image-based re-identification, a work that marked the separation of person re-identification from binocular vision tracking as a separate computer vision task. In 2014, Yi[16]and Li et al[17]. first generalized neural networks to the RE-ID task. In April 2018, Zheng Zhedong et al[18]. proposed camera style re-adaptation-based RE-ID, whose core idea is to process the dataset into different camera style samples by generative adversarial networks, and the samples can continue to follow the original labels. Although annotating large-scale datasets with different styles is very effective in improving the robustness of the trained model, preprocessing the annotated datasets can save a lot of labor costs.

3 Method

This paper is based on the modified re-ID paper and procedure based on camera style re-adaptation proposed by Zheng Zhedong et al. In that paper, the original author uses GANs and resnet5 achieved Rank@1=88.84%, mAP=71.59% results[18]. Further, based on this author's procedure this paper modifies the backbone to MobileNetV3 Large and produces experimental results that ultimately achieve a significant reduction in the number of parameters while losing a small amount of accuracy.

3.1 CycleGANs

A GAN consists of a generator and a discriminator, assuming that a bunch of data samples are now provided with the following distribution:

$$p_{model}$$

The role of the generator is to generate an instance that is very close to the sample distribution, and the generator is represented by the G(x) function for a vector z, the generator can be represented by G(z). In order to ensure that the generator produces a sufficiently close example of the sample distribution, the discriminator D(x) function comes into play. The generator and the discriminator are actually two different neural networks with different back propagation processes and loss functions. In each training session, the generator continuously generates instances similar to the real samples, and the discriminator continuously judges the truth of the real samples and the generated samples.

For CycleGAN, the core idea is to transform images by two generators and two discriminators. One of the generators converts an image of one style into an image of another style, and the other generator performs the opposite operation. The two discriminators are used to evaluate whether the output image of the generator is realistic or not. However, a new problem arises at this time because there is no guarantee that the semantic information of the original image is preserved during the conversion. There are many mapping relations between the two domains, but very few mappings that can preserve the information of x. Moreover, it can easily lead to schema collapse, i.e., the output image is the same no matter what the input is. Therefore, if we can make y mapped back to x again, this will ensure that the semantic information will not be lost in y. So CycleGAN introduces a cyclic consistency loss to ensure that the transformation is bidirectional, i.e., the image consistency is maintained when going from one domain to another and back again. The following is a schematic representation of this bidirectional transformation.

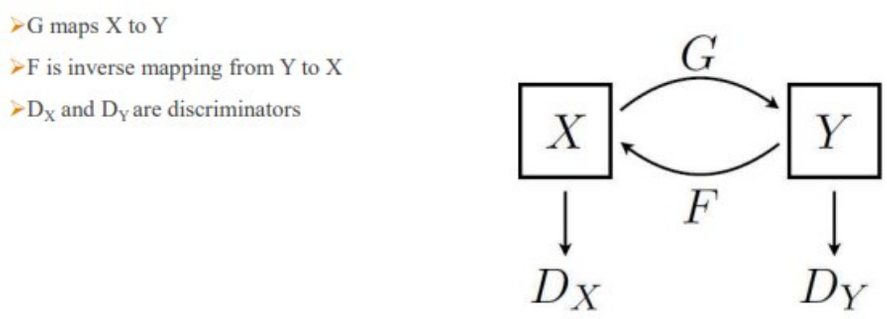

Figure 1 What is a Cyclic Model?[9]

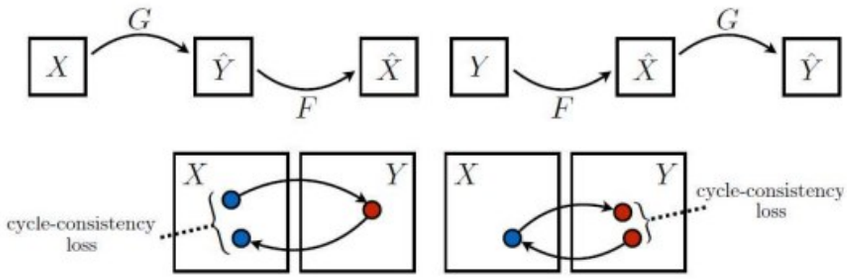

Figure 2 Why do we need Cyclic loss?[9]

CycleGAN is able to learn more robust and accurate image transformations by introducing a loss of circular consistency, Without requiring pairs of training data.This makes CycleGAN excellent in many applications, such as style transfer, season conversion, animal image conversion and so on.

CycleGAN has its own unique loss function[9]

$$L\left(G,F,D_X,D_Y\right)=\mathcal{L}_{GAN}\left(G,D_Y,X,Y\right)+\mathcal{L}_{GAN}\left(F,D_X,X,Y\right)+\lambda\mathcal{L}_{cyc}\left(G,F\right) \tag{1}$$

3.2 Person re-identification based on style transformation

The core of person re-identification based on style transformation designed by Zhedong et al. Is to design a set of preprocessing procedures for data sets. The flow chart is as follows:

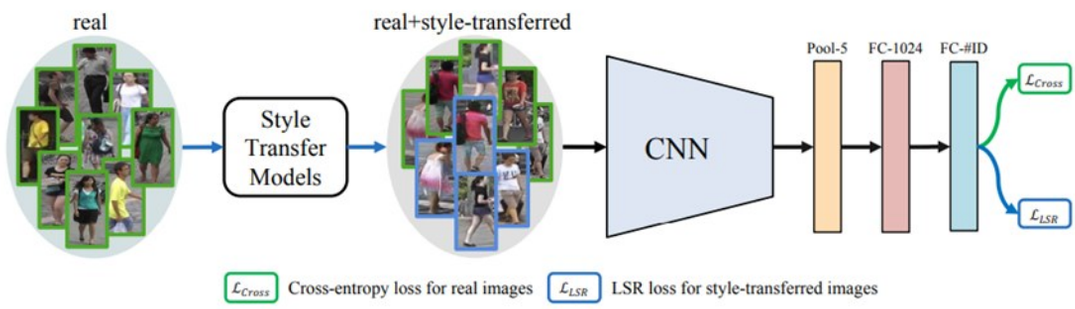

Figure 3 The process of camera style adaptation[18]

As shown in the figure, the real images will be combined at the same time after style processing and used to train the CNN network. Finally, the cross-entropy loss function is used to process the real image, The label smoothing regularization loss function is used to process the image after the style transformation.

The cross-entropy loss function can be written as:

$$\mathcal{L}_{Cross}=-\sum_{c=1}^{C}\log\left(p(c)\right)q(c) \tag{2}$$

Where C is the number of classes and p (C) is the predicted probability that the input belongs to label C.

And the formula of the smooth regular loss function is:

$$\mathrm{L}_{LSR}=-(1-\epsilon)\log p(y)-\frac{\epsilon}{C}\sum_{c=1}^{C}\log p(c) \tag{3}$$

3.3 MobileNet

The core innovation of Mobilenetv1 is the use of depth-separable convolution, which integrates an ordinary convolution into a depth convolution and a point-by-point convolution. The formula for the number of parameters of ordinary convolution is as follows[10]:

$$D_K*D_K*M*N*D_F*D_F \tag{4}$$

The formula for calculating the parameters of the depth-separable convolution is as follows:

$$D_K*D_K*M*D_F*D_F+M*N*D_F*D_F \tag{5}$$

The number of parameters of depth-separable convolution is that of ordinary convolution.

$$\frac{1}{N}+\frac{2}{D_K^2} \tag{6}$$

Unlike previous versions, MobileNetV3 uses a linear activation function called ReLU6, but instead uses a non-linear activation function called Swish. This function is defined as[12]:

$$swish\, x = x\,\sigma(x) \tag{7}$$

However, the cost of using this loss function is to reduce the performance of mobile and embedded devices. So v3 uses a "hard vision", defined by h-swish as follows:

$$h-swish[x] = x\frac{ReLU6(x+3)}{6} \tag{8}$$

This change is almost the same in terms of accuracy, but it is easier to deploy.

In addition, MobileNetV3 has redesigned the latency layer, which has reduced latency by 7ms. The number of operations has been reduced by 30 million times with little loss of accuracy.

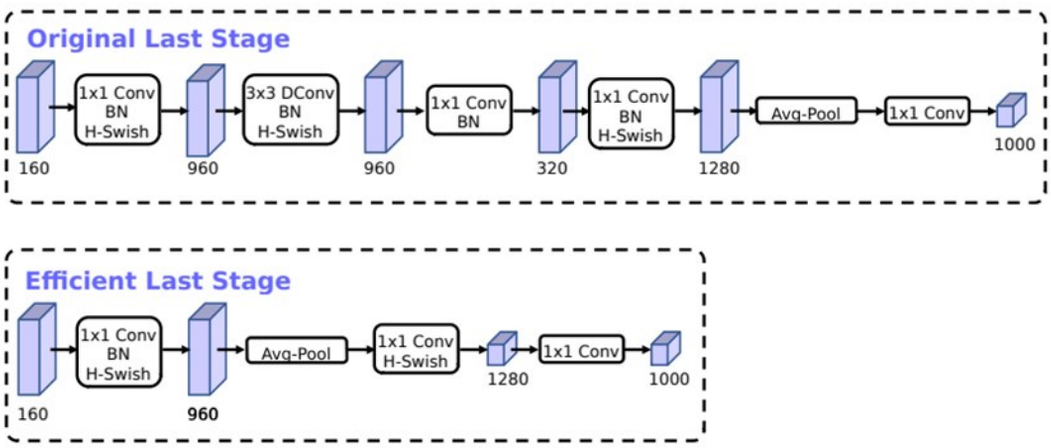

Figure 4 Comparison of MobileNetV3 in final layer modification[12]

In addition, MobileNet provides two adjustable parameters, the width multiplier α and the resolution parameter β. α is the number of channels for uniformly reducing each layer. Letting the number of input channels be M and the number of output channels be N, then after applying the width multiplier α, the number of input channels is αM and the number of output channels is αN. Typically, the values of α are 1, 0.75, 0.5, and 0.25. Common values for β are also 1, 0.75, 0.5, and 0.25. Therefore, the calculation amount of the layer after adding the α and β parameters is

$$D_K{}^*D_K{}^*\alpha M^*\beta D_F{}^*\beta D_F + \alpha M^*\alpha N^*\beta D_F{}^*\beta D_F \tag{9}$$

It is worth noting that the resolution parameter β only affects the amount of computation and does not affect the number of parameters.

The accuracy of the model under the two effects is shown in the chart in the paper:

Table 1 MobileNetWidthMultiplier[10]

Width Multiplier	ImageNet Accuracy	Million Mult-Adds	Million Parameters
1.0 MobileNet-224	70.6%	569	4.2
0.75 MobileNet-224	68.4%	325	2.6
0.5 MobileNet-224	63.7%	149	1.3
0.25 MobileNet-224	50.6%	41	0.5

Table 2 MobileNetResolution[10]

Width Multiplier	ImageNet Accuracy	Million Mult-Adds	Million Parameters
1.0 MobileNet-224	70.6%	569	4.2
1.0 MobileNet-192	69.1%	418	4.2
1.0 MobileNet-160	67.2%	290	4.2
1.0 MobileNet-128	64.4%	186	4.2

4 Experiment

4.1 Dataset

Paper used the Market-1501 data set to evaluate my method because the data set: 1) is a highly recognized data set in

the re-ID field, and has both a training set and a test set; 2) The data set is also one of the data sets used by the source program, and the use of the data set can reduce variables and facilitate the comparison of results.

Market-1501 contains 32668 tag images of 1501 identities captured from 6 cameras. There are two parts: 12936 images of 751 identities for training and 19732 images of 750 identities for testing. In the experiment, all the input images were resized to 256 X 256.

4.2 Experimental Cloud Platform

I used the Autodl computing power cloud platform. The platform offers computing services at different price points from 2080 Ti to A100 SXM4/80GB. In this experiment, I used a 4090 D for training, which effectively saved me a lot of training time. At the same time, the platform supports remote login of different software such as Pycharm and Visual Studio. It also supports Github copy and provides different configuration environments to reduce the work of environment configuration.

4.3 Experimental content

4.3.1 The choice of Epoch

The number of epochs during training is 60, which is the same as the default setting of the source program.λ = 10 in all experiments, batch size = 1, generator learning rate of 0.0002 for the first 30 training cycles, discriminator of 0.0001,And the learning rate is reduced to zero in the rest of the training period. At the same time, in the step of camera perception style transfer, the program will generate L-1 additional fake training images for each training image. And retain that original identity as enhanced train data. This is for the sake of consistency with the variables of the source program.

4.3.2 Backbone selection and programming

The backbone uses the pre-trained model MobileNet_V3_Large provided by Torchvision. In this article, I use the python language to program the MobileNetV3 network. The MobileNetV3 network structure given in the original paper is shown in the following figure:

$224^2 \times 3$	conv2d	-	16	-	HS	2
$112^2 \times 16$	bneck, 3x3	16	16	-	RE	1
$112^2 \times 16$	bneck, 3x3	64	24	-	RE	2
$56^2 \times 24$	bneck, 3x3	72	24	-	RE	1
$56^2 \times 24$	bneck, 5x5	72	40	✓	RE	2
$28^2 \times 40$	bneck, 5x5	120	40	✓	RE	1
$28^2 \times 40$	bneck, 5x5	120	40	✓	RE	1
$28^2 \times 40$	bneck, 3x3	240	80	-	HS	2
$14^2 \times 80$	bneck, 3x3	200	80	-	HS	1
$14^2 \times 80$	bneck, 3x3	184	80	-	HS	1
$14^2 \times 80$	bneck, 3x3	184	80	-	HS	1
$14^2 \times 80$	bneck, 3x3	480	112	✓	HS	1
$14^2 \times 112$	bneck, 3x3	672	112	✓	HS	1
$14^2 \times 112$	bneck, 5x5	672	160	✓	HS	2
$7^2 \times 160$	bneck, 5x5	960	160	✓	HS	1
$7^2 \times 160$	bneck, 5x5	960	160	✓	HS	1
$7^2 \times 160$	conv2d, 1x1	-	960	-	HS	1
$7^2 \times 960$	pool, 7x7	-	-	-	-	1
$1^2 \times 960$	conv2d 1x1, NBN	-	1280	-	HS	1
$1^2 \times 1280$	conv2d 1x1, NBN	-	k	-	-	1

Figure 5 The layer of MobileNetV3[12]

According to mobile netv3 large. In the code provided in the. py file, Torchvision defines a MobileNetV3 class that provides several properties for external call parameters. In the code, I mainly used the pretrained, width_mult, reduced _ tail, dialated, and num _ classes attributes. It is worth noting that when the pretrained attribute is true, the pre-trained weight file is used. At this point, num _ classes is a fixed value of 1000, which represents the number of classes in ImageNet. If you need to select the number of categories yourself, select False for the pretraining option. For the re-ID

program developed by Zheng Zhedong et al., the number of classes should be set to 751.

4.3.3 The writing of forward transfer function

Torchvision offers the MobileNetV3 family of networks with very limited interfaces to the outside world. Only an interface return self. _ forward_impl (X) is provided that calls an internal forward transfer function. This means that, Different from the pre-training model, ResNet provides a variety of callable functions such as convolution layer, batch normalization layer, activation function layer, classifier layer, etc. Neither MobileNetV3 nor mobilenetv2 allows external changes to the internal hierarchy. For the re-ID program designed by Zheng Zhedong et al., there are 751 classifications, and finally through the classification layer designed by themselves. Output 512 tensors. The input classification number of the pre-training model MobileNetV3 and the output feature number of the final classifier layer use the same external variable and this results in a mismatch between the output feature vector and the required feature vector for some programs that have already been designed. This is a very difficult problem that cannot be solved by modifying or adding or removing the internal hierarchy. In the end, my solution was to rewire the MobileNetV3 network and use the new classification layer provided by the re-ID program.

5 Results

The experimental results are as follows:

Table 3 Results

Model	Rank@1	Rank@5	Rank@10	mAP	time	Training time
ResNet50	0.8916	0.9578	0.9738	0.7234	26.82s	13min30s
MobileNetV3 Large	0.8794	0.9539	0.972	0.7211	25.61s	10min23s

Where time is the result of running the program on the 4090D platform using the Market-1501 test set. Training time is the average time spent on training on the 4090D platform using the Market-1501 training set.

6 Conclusion

The experimental results show that the accuracy of MobileNetV3 large is close to that of resnet50 in the actual training process. It performs well in several core indicators of re-ID, such as Rank @ 1, Rank@5, Rank@10 and mAP, and improves the computing speed. This is due to the superior, low-power platform-oriented architecture of MobileNetV3 .

In addition, for the MobileNet series pre-training model provided by Torchvision, on the one hand, it provides convenience for building backbone quickly. But on the other hand, the conservatism of MobileNet on the external call interface makes further modifications and parameter settings to the network very difficult, the hierarchy cannot be called and modified as freely as ResNet, which is also a pre-trained network. Therefore, for the program design that has been completed, it is very difficult to quickly add a new MobileNet backbone to a pre-trained network, special attention should be paid to the setting of input and output data. Therefore, I believe that in order to make rational use of the pre-training network, it should be built around the network at the beginning of the design. I also called on Torchvision to design more external interfaces to the network, it also calls for more developers to develop one more open and free MobileNet pre-training networks.

References

[1] I. J. Goodfellow et al., 'Generative Adversarial Networks', Jun. 10, 2014, arXiv: arXiv:1406.2661. doi: 10.48550/arXiv.1406.2661.

[2] E. Denton, S. Chintala, A. Szlam, and R. Fergus, 'Deep Generative Image Models using a Laplacian Pyramid of Adversarial Networks', Jun. 18, 2015, arXiv: arXiv:1506.05751. doi: 10.48550/arXiv.1506.05751.

[3] S. Reed, Z. Akata, X. Yan, L. Logeswaran, B. Schiele, and H. Lee, 'Generative Adversarial Text to Image Synthesis', Jun. 05, 2016, arXiv: arXiv: 1605.05396. doi: 10.48550/arXiv.1605.05396.

[4] A. Radford, L. Metz, and S. Chintala, 'Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks', Jan. 07, 2016, arXiv: arXiv:1511.06434. doi: 10.48550/arXiv.1511.06434.

[5] T. Karras, S. Laine, and T. Aila, 'A Style-Based Generator Architecture for Generative Adversarial Networks', Mar. 29, 2019, arXiv: arXiv: 1812.04948. doi: 10.48550/arXiv.1812.04948.

[6] A. Brock, J. Donahue, and K. Simonyan, 'Large Scale GAN Training for High Fidelity Natural Image Synthesis', Feb. 25, 2019, arXiv: arXiv: 1809.11096. doi: 10.48550/arXiv.1809.11096.

[7] H. Zhang et al., 'StackGAN: Text to Photo-realistic Image Synthesis with Stacked Generative Adversarial Networks', Aug. 05, 2017, arXiv: arXiv: 1612.03242. doi: 10.48550/arXiv.1612.03242.

[8] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, 'Image-to-Image Translation with Conditional Adversarial Networks', Nov. 26, 2018, arXiv: arXiv: 1611.07004. doi: 10.48550/arXiv.1611.07004.

[9] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, 'Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks', Aug. 24, 2020, arXiv: arXiv:1703.10593. doi: 10.48550/arXiv.1703.10593.

[10] A. G. Howard et al., 'MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications', Apr. 17, 2017, arXiv: arXiv: 1704.04861. doi: 10.48550/arXiv.1704.04861.

[11] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, 'MobileNetV2: Inverted Residuals and Linear Bottlenecks', Mar. 21, 2019, arXiv: arXiv:1801.04381. doi: 10.48550/arXiv.1801.04381.

[12] A. Howard et al., 'Searching for MobileNetV3', Nov. 20, 2019, arXiv: arXiv:1905.02244. doi: 10.48550/arXiv.1905.02244.

[13] T. Huang and S. Russell, 'Object Identification in a Bayesian Context9"'.

[14] W. Zajdel, Z. Zivkovic, and B. J. A. Krose, 'Keeping Track of Humans: Have I Seen This Person Before?', in Proceedings of the 2005 IEEE International Conference on Robotics and Automation, Apr. 2005, pp. 2081–2086. doi: 10.1109/ROBOT.2005.1570420.

[15] N. Gheissari, T. B. Sebastian, and R. Hartley, 'Person Reidentification Using Spatiotemporal Appearance', in 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Volume 2 (CVPR'06), New York, NY, USA: IEEE, 2006, pp. 1528–1535. doi: 10.1109/CVPR.2006.223.

[16] D. Yi, Z. Lei, S. Liao, and S. Z. Li, 'Deep Metric Learning for Person Re-identification', in 2014 22nd International Conference on Pattern Recognition, Aug. 2014, pp. 34–39. doi: 10.1109/ICPR.2014.16.

[17] W. Li, R. Zhao, T. Xiao, and X. Wang, 'DeepReID: Deep Filter Pairing Neural Network for Person Re-identification', in 2014 IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA: IEEE, Jun. 2014, pp. 152–159. doi: 10.1109/CVPR.2014.27.

[18] Z. Zhong, L. Zheng, Z. Zheng, S. Li, and Y. Yang, 'Camera Style Adaptation for Person Re-identification', Apr. 10, 2018, arXiv: arXiv:1711.10295. doi: 10.48550/arXiv.1711.10295.