This CVPR 2020 paper is the Open Access version, provided by the Computer Vision Foundation.
Except for this watermark, it is identical to the accepted version;
the final published version of the proceedings is available on IEEE Xplore.

Unity Style Transfer for Person Re-Identification

Chong Liu [1,2] Xiaojun Chang [3] Yi-Dong Shen [1*]

[1] State Key Laboratory of Computer Science, Institute of Software, Chinese Academy of Sciences, China
[2] University of Chinese Academy of Sciences, Beijing 100049, China
[3] Monash University, Melbourne, Australia

liuchong@ios.ac.cn, cxj273@gmail.com, ydshen@ios.ac.cn

Abstract

Style variation has been a major challenge for person re-identification, which aims to match the same pedestrians across different cameras. Existing works attempted to address this problem with camera-invariant descriptor subspace learning. However, there will be more image artifacts when the difference between the images taken by different cameras is larger. To solve this problem, we propose a UnityStyle adaption method, which can smooth the style disparities within the same camera and across different cameras. Specifically, we firstly create UnityGAN to learn the style changes between cameras, producing shape-stable style-unity images for each camera, which is called UnityStyle images. Meanwhile, we use UnityStyle images to eliminate style differences between different images, which makes a better match between query and gallery. Then, we apply the proposed method to Re-ID models, expecting to obtain more style-robust depth features for querying. We conduct extensive experiments on widely used benchmark datasets to evaluate the performance of the proposed framework, the results of which confirm the superiority of the proposed model.

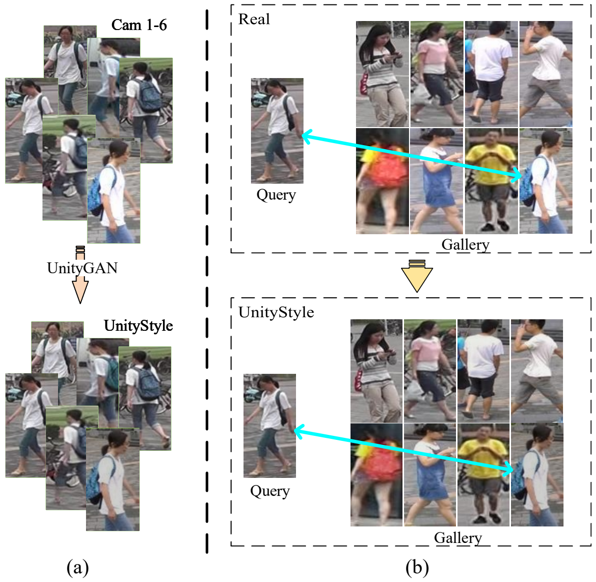


Figure 1. **(a)** Different cameras often have significant style differences, and UnityGAN is used to convert image styles to each other to achieve a unified style. **(b)** By using UnityStyle, the query collection is similar to the gallery collection image style, making it easier to find the right match.

1. Introduction

Person re-identification (re-ID) [37] is a multi-camera query task. Given the image of one or a group of target people, the same person is identified from the database of multi-camera. There are multiple cameras in the re-id task, and the images style taken by each camera are often different for the same person due to various factors (environment, light, *etc.*) [22, 24]. Besides, even the same camera will take different styles of images due to the different time (morning, noon, afternoon, *etc.*). Therefore, the style of the image changes, which has a considerable impact on the final task results.

There are some methods available to address different camera style problems. One solution is to obtain stable feature representations between different cameras. There are KISSME[16], DNS [34] and so on in the traditional method. IDE [37], PCB [32] solve problems through deep representation learning. Another way is to use the GANs [44] to learn the style disparities between different cameras and to enhance the data through the style transfer method, resulting in CamStyle [43].

Compared with the previous methods, this paper proposes an alternative solution to smooth the style disparities across different cameras and within the same camera. We start from CamStyle and find that although this method can

[*] Corresponding author

learn and transform styles between different cameras, there are still some problems. **1)** There will be image artifacts in the transfer sample generated by CycleGAN [44], especially for the shaped part, which produces a considerable number of error images (Fig. 2). **2)** The generated enhanced images introduce some noise to the system, and the Label Smooth Regularization (LSR) is required to adjust the network performance. **3)** The generated enhanced images can only be used as a data enhancement method to extend the training set, which is not effective. **4)** The number of models that need to be trained is C_C^2 **(where C is the number of cameras)**, which means that as the number of cameras increases, the number of models that need to be trained will become larger and larger, which is not applicable in scenarios where computing resources are insufficient.

To access the above problems, we build a UnityStyle adaption method, which can smooth the style disparities within the same camera and across different cameras. We overcome the problem of CycleGAN's easy deformation with UnityGAN and generate style-unity images with higher quality (Fig. 2). Further, we rely on the style data of each camera learned by UnityGAN to get a UnityStyle image suitable for all camera styles, which makes the generated enhanced images more efficient. Finally, we combine real images and UnityStyle images as the new data augmentation training set.

The UnityStyle adaption method has the following advantages. **First**, as a data enhancement scheme, the generated enhanced sample can be treated the same as original images. Thus, LSR is no longer needed for UnityStyle images. **Secondly**, by adapting to different camera styles, it is also robust to style changes within the same camera. **Thirdly**, it does not require additional information, and all enhancements are derived from the general information of the re-ID task. **Finally**, it only needs to train C UnityGAN models, which only require a small number of computing resources.

To summarize, this paper has the following contributions:

- The UnityStyle method for Re-ID is proposed to generate shape-stable style-changing enhanced images, which can be treated equally as the real images. LSR is no longer needed for the style transferred images.

- The UnityStyle does not require training a large number of models. It only needs to train C UnityGAN models, which use a small number of computing resources.

- We propose a novel data enhancement scheme. We use UnityStyle to eliminate style differences between different images, which makes a better match between the images of query and gallery, and the generated enhanced images more efficient.

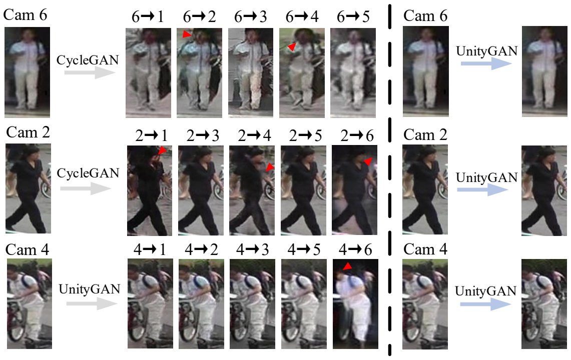


Figure 2. Examples of generated by CycleGAN and UnityGAN in Market-1501. UnityGAN can produce better style-transferred images. As can be seen from the comparison in the figure, UnityGAN solves the problem that CycleGAN generates error images, and no longer has the problem of marking.

- Our experiments show that UnityStyle can be easily applied to a wide range of models and can significantly improve experimental performance.

2. Related Work

2.1. Deep learning person re-identification

In the field of person re-ID, many deep learning methods [37, 35, 18, 9, 19, 2, 26, 20, 4, 3, 23] have been proposed. [37] uses the ImageNet [17] pre-trained model, and trains the model as image classification in re-ID, which is called the ID-discriminative embedding (IDE). Problems with overfitting and insufficient sample types have emerged as the CNN models become sufficiently complex. There are some data augmentation methods been proposed to solve this problem. [40] used DC-GAN [28] to improve the discriminative ability of learned CNN embeddings. [6, 27, 39] focus on the person's pose, and improve the final performance by generating different pose images. More related to this work, [43] proposed CamStyle data augmentation approach which transfers images from one camera to the style of another camera.

2.2. Generative Adversarial Networks

Generative adversarial networks (GANs)[8] have achieved remarkable success since its introduction. Recently, GANs are used in image translation [13, 15, 44], style transfer [5, 14] and image editing [21]. In [5], a style transfer model was proposed by extracting and reconstructing the substance and style of the image. [13] showed through Pix2Pix that GANs can learn style mapping between different domains. Recently, the Pix2Pix-like framework have evolved to apply to unsupervised pairs [15, 44]. [7] proposed a new method built upon the DiscoGAN [15] and CycleGAN [44] architectures, which

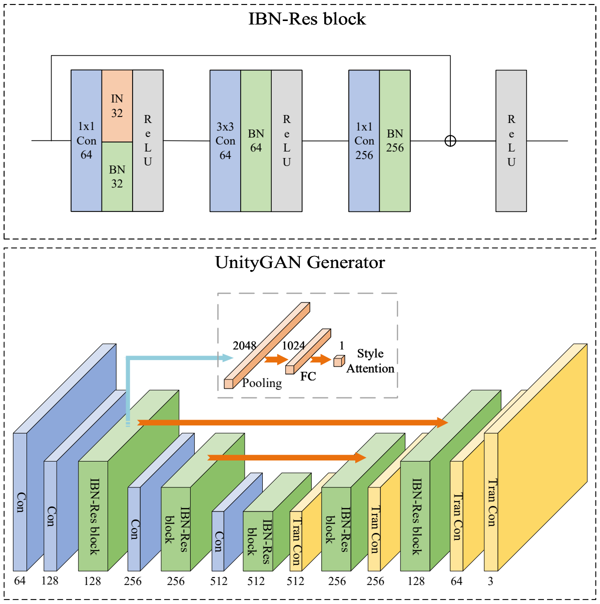


Figure 3. Generators from different unsupervised image translation models. **Blue box** is convolution layer. **Green box** is residual block. **Yellow box** is transposed convolution layer. **Orange box** is Style Attention module.

overcome the limitations of shape changes through more effective learning.

2.3. IBN-Net

According to [25], Batch Normalization (BN) [12] improves the sensitivity of features to image content, and Instance Normalization (IN) [11] improves robustness to style class changes. Therefore, IN achieves better results than BN in the field of style migration. Shallow features are related to style, and deep features are related to high-level features (content features such as face and gesture). This paper raises two criteria:

- IN is only added to the shallow network and is not added to the deep network. Because the feature of IN extraction reduces the difference between images, it cannot be placed in the deep layer to influence the classification effect.
- The shallow BN layer should also be retained to ensure that content-related information can be smoothly passed into the deep layer.

3. The Proposed Method

In this section, we will explain in detail the method we proposed. First, we create UnityGAN and make it suitable for re-ID task. Then, we propose how to generate UnityStyle images. Finally, our proposed method is shown to be applied to multiple models and explain the detailed operation of the pipeline.

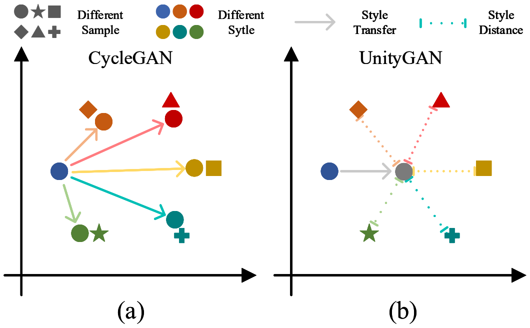


Figure 4. The position of the sample in the style space. **(a)** Each sample requires multiple style transfer to generate different style samples. **(b)** Each sample only needs one style transfer to generate a uniform style sample.

3.1. UnityGAN

In this section, we use UnityGAN for Style Transfer and train the Style Transfer model. UnityGAN absorbs the advantages of DiscoGAN and CycleGAN and improves them. DiscoGAN uses the standard architecture, and its narrow bottleneck layer may prevent the output image from retaining visual details in the input image. CycleGAN introduces residual blocks to increase the capacity of DiscoGAN, but the use of residual blocks on a single-scale layer does not preserve information on multiple scales. UnityGAN combines two networks (Fig. 3, bottom), and introduces residual blocks and skip connections in the multi-scale layer, which can retain multi-scale information at the same time, making the transformation more precise and accurate. Through the transfer of multi-scale information, UnityGAN can generate structurally stable images and avoid generating images of the wrong structure (Fig. 2, right). Unlike CycleGAN, UnityGAN tries to generate a picture that blends all styles without having to learn a transfer for each style (Fig. 4).

Further, we created an IBN-Res block (Fig. 3, upper) based on [25] discussion, which can increase the robustness of style changes while maintaining structural information. Add the IBN-Res block to the UnityGAN model to adapt the model to the style changes and ensure that the model generates a uniform style of fake images.

Given image domains X and Y, let $G : X \rightarrow Y$ and $F : Y \rightarrow X$. D_X and D_Y denote discriminators for G and F respectively. To preserve the feature information of the image while changing the style, we add the identity mapping loss[44] to the formula. The identity mapping loss can be expressed as:

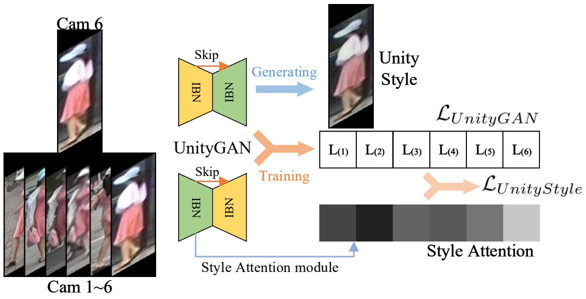


Figure 5. The main process of UnityStyle. In training, UnityGAN uses all camera style pictures for training. In generating, UnityGAN generates UnityStyle for each image.

$$\begin{aligned}\mathcal{L}_{ID} =& \mathbb{E}_{x\sim l_x}(\|F(x)-x\|_1)\\ &+\mathbb{E}_{y\sim l_y}(\|G(y)-y\|_1),\end{aligned} \tag{1}$$

Therefore, UnityGAN's loss function comprises four types of loss normalized terms: the standard GANs loss, the feature matching loss, the identity mapping loss and the cyclic reconstruction loss.

Our total objective function is:

$$\begin{aligned}\mathcal{L}_{UnityGAN}(x,y) = &\lambda_{GAN}SLN(\mathcal{L}_{GAN})\\ &+\lambda_{FM}SLN(\mathcal{L}_{FM})\\ &+\lambda_{ID}SLN(\mathcal{L}_{ID})\\ &+\lambda_{CYC}SLN(\lambda_{SS}\mathcal{L}_{SS}+\lambda_{L1}\mathcal{L}_{L1}),\end{aligned} \tag{2}$$

Where, $\mathcal{L}_{FM} = \mathcal{L}_{FM_X}(G, D_X) + \mathcal{L}_{FM_Y}(F, D_Y)$, $\mathcal{L}_{GAN} = \mathcal{L}_{GAN_X}(F, D_X, Y, X) + \mathcal{L}_{GAN_Y}(G, D_Y, X, Y)$, SLN is scheduled loss normalization [7]. With $\lambda_{GAN} + \lambda_{FM} + \lambda_{ID} + \lambda_{CYC} = 1$, $\lambda_{SS} + \lambda_{L1} = 1$, and all coefficients ≥ 0.

In the training phase, we take each camera and all cameras as a group from the training set to train a UnityGAN model (Fig. 5). Same as [44] during the training phase, we resize all images to 256 × 256. UnityGAN can generate stable structure pictures and reduce the number of training models. However, images generated by UnityGAN is unstable in style (Fig. 8). To solve this problem, we propose the UnityStyle loss function (Eq. 4) in the next section to ensure that UnityGAN generates stable style images.

3.2. UnityStyle

In this section, we propose the concept of UnityStyle. UnityStyle images are generated from UnityGAN, which can smooth the style disparities within the same camera and across different cameras. Then we use UnityStyle images for model training and prediction to improve the performance of the model.

To ensure that UnityGAN can generate UnityStyle images, we add the Style Attention module to the UnityGAN Generator (Fig. 3). Low-level image features get style-related attention features through this module. We define the Style Attention of the input image x,

$$\mathcal{A}(x) = Sigmoid(A_{style}(G_1(x))), \tag{3}$$

where, A_{style} is Style Attention Module, G_1 is the first IBN-Res block output of UnityGAN Generator.

Further, we get the final UnityStyle loss function,

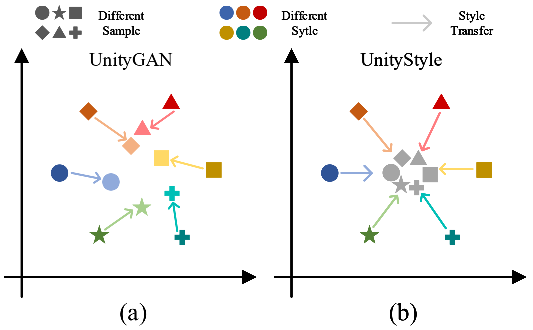
$$\mathcal{L}_{UnityStyle} = \sum_{c=1}^{L}(\mathcal{A}(y_i^{(c)})\mathcal{L}_{UnityGAN}(x_i, y_i^{(c)})), \tag{4}$$

where, c is camera number, C is the number of cameras, $\mathcal{A}(y_i^{(c)})$ is the ith camera's style attention. Under the definition of Eq. 4, the model will generate a style -stable picture, and the style of this picture is between all camera styles (Fig. 6).

From Fig. 8, we can see that UnityGAN with the above module can produce style-stable pictures, compared to UnityGAN without Attention. The image generated by UnityGAN under the definition of Eq. 3 is a uniform style image, and the different style features of multiple cameras are smoothed out. The six images under the six cameras, whose image styles produced by UnityStyle Transfers are no different. Intuitively, our method has better performance, and we will further compare it in the experiment.

Although using UnityGAN guarantees the stability of the structure, the style and lighting of each image are different (Fig 8 middle line). There are multiple style images in the training set, but current GAN methods cannot converge to a unified style (Different styles of images have different losses). The Style Attention module quantifies the image style and introduces $\mathcal{L}_{UnityStyle}$ for training, ensuring the uniformity of the style of the generated images (Fig 8 last line). In Fig 5, the lighter the Attention square color of the image, the farther the unity-style is away from the corresponding image style (The loss has a lower weight). It is ensured that images with a large difference in unity-style have a small effect on the style of the generated image, and we can generate images with unity-style. In addition, the style information is mainly contained in the shallow layer. This is why we add Style Attention module in the shallow layer.

Past data enhancements are often performed in test data to ensure that the trained model is more robust, but this is only one aspect of the work. In re-ID task, test set consists of two sets of query and gallery. The test finds the image belonging to someone from the gallery images by his given query image. Under the task of multiple cameras, in Fig. 1, the query image and the corresponding gallery image are



(a) (b)

Figure 6. The position of the sample in the style space. **(a)** Sample can not achieve the complete unity of style after the transfer of UnityGAN without the style attention module. **(b)** Sample can be completely unified by the transfer of UnityStyle.

often different styles. Therefore, we can start from the test set and ensure that the training model is more robust while making the multi-camera test set style as close as possible (i.e., close the distribution of the test set, as shown by Fig. 1).

In training, UnityStyle images are used to augment the training set, which makes the model more adaptive. In testing, UnityStyle images are test inputs, which make it easier to match between query and gallery. The test set consists of two sets of query and gallery. The test finds the image belonging to someone from the gallery images by his given query image. Under the task of multiple cameras, as shown in the upper of Fig. 1 (b), the query image and the corresponding gallery image often have different styles. Therefore, we take UnityStyle images as test inputs, which ensures that the training model is more robust and makes the multi-camera test set style as close as possible (i.e. close the distribution of the test set, as shown by the Fig. 1 (b) bottom).

3.3. Deep Re-ID Model

There are already many excellent deep Re-ID models (such as IDE[37], PCB[32] and st-ReID[33]), and our method can be easily applied to these models. Our approach is based on the ID-discriminative embedding (IDE) [37] as an example, and the backbone of the network is ResNet. The basic flow of our proposed pipeline is shown in Fig. 7. In order to resemble a human body, the size of the input image is 256×128. In the training phase, we need to ensure that the final classification layer output is consistent with the number of labels in the training set. As shown in Fig. 7, we replace the last classification layer with two fully connected layers. At the test phase, we use the model's 2048-dimensional feature output for evaluation. Evaluator uses output features to calculate mean average precision(mAP) and top-K represents the proportion of the correct results in the top K retrieval results.

3.4. Pipeline

In this section, we will demonstrate the Pipeline of the model from the stages of training and testing in detail (Fig. 7).

3.4.1 Training

Before training, we used trained UnityGAN Transfers to generate UnityStyle images. We combine real images and UnityStyle images as an enhanced training set. In training, we take the image in the enhanced dataset as input, and all input images sizes are specified as 256×128. We randomly sample N real images and N UnityStyle images. Under the above definition, we get the loss function,

$$\mathcal{L}_{REID} = \frac{1}{N}\sum_{i=1}^{N}(\mathcal{L}_R^i + \mathcal{L}_U^i), \tag{5}$$

where $\mathcal{L}_R^i = \mathcal{L}_{Cross}(x_R^i)$ and $\mathcal{L}_U^i = \mathcal{L}_{Cross}(x_U^i)$. x_R^i is real image sample and x_U^i is UnityStyle image sample. $\mathcal{L}_{Cross}$ is cross-entropy loss function,

$$\mathcal{L}_{Cross}(x) = -\sum_{l=1}^{L}\log(p(l))q(l), \tag{6}$$

where, L is the number of labels, $p(l)$ is the probability that the label of x is predicted to be l, $q(l) = \{1 \text{ if } l = y | 0 \text{ if } l \neq y\}$ is the ground-truth distribution. According to the previous description, $\sum_{l=1}^{L} p(l) = 1$. In $q(l)$, y is the ground-truth label corresponding to the current image.

For x with ground-truth label y, according to Eq. 6 we can get,

$$\mathcal{L}_{Cross}(x) = -\log(p(y)). \tag{7}$$

Further according to Eq. 7 and Eq. 5, we can get,

$$\mathcal{L}_{REID} = -\frac{1}{N}\sum_{i=1}^{N}\log(p_R^i p_U^i), \tag{8}$$

where, p_R^i is the probability that the i-th real image is predicted correctly, p_U^i is the probability that the i-th UnityStyle image is predicted correctly.

As mentioned before, real images and fake images are treated differently in [43], using different loss functions for training. In our version, we overcome the shortcomings of [43], and the resulting images have the same status as real images. Therefore, we no longer use LSR during training.

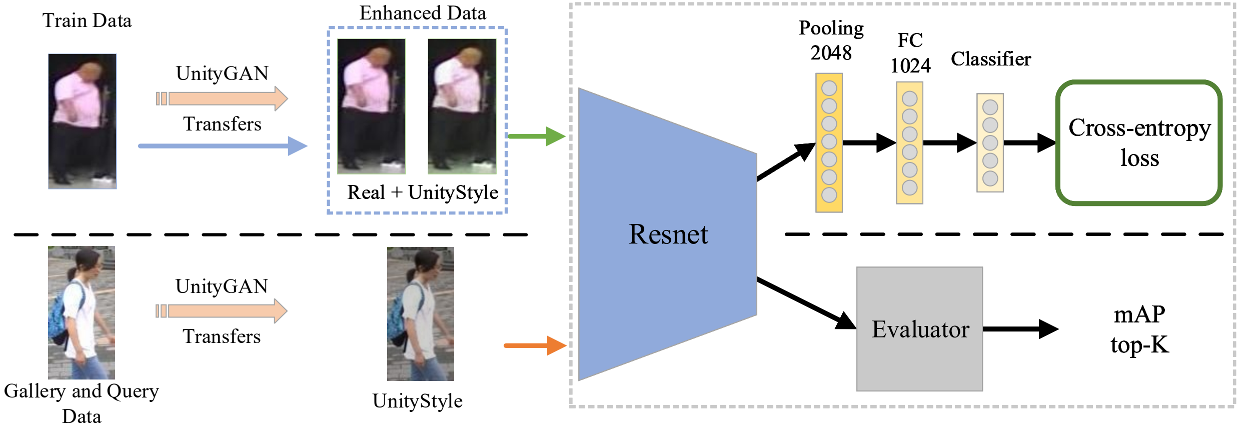


Figure 7. The pipeline of our method (Take IBN as an example, gray boxes can be replaced with other models, such as PCB, st-ReID). In training, the UnityStyle images are generated by training data using the UnityStyle Transfers, and Resnet extracts the features of these two types of data, which will be trained by the classifier. In testing, the UnityStyle images are generated by the gallery and query data through UnityStyle Transfers, and then we use the unified style images to extract features by Resnet, which are evaluated to obtain the results.

3.4.2 Testing

For testing, it is divided into query and gallery data sets. Previously we introduce the concept of UnityStyle to ensure that the images of two datasets generate the corresponding UnityStyle images by UnityStyle Transfers before starting the test. We use the generated UnityStyle images as the new input to test. As discussed before, UnityStyle narrows the gap between the style of elements in the test set (Fig. 1), which can further improve query performance.

4. Experiments

4.1. Datasets and Evaluation Setting

We use two common and representative re-ID datasets, Market-1501 [36] and DukeMTMC-reID [40]. Top-1 accuracy and mAP were used to evaluate both datasets.

Market-1501 [36] Contains 32,668 labeled images from 6 camera angles of view, which has 1501 identities and some noise images. The data set is divided into three parts: 751 individuals used for training containing 12936 images, 750 individuals for testing containing 19732images, and 3368 images for the query in total.

DukeMTMC-reID [40] Contains 36,411 labeled images from 8 camera views, which has 1404 identities and some noise images. Similar to Market-1501, the data set is divided into three parts: 702 individuals used for training containing 16522 images, 702 individuals for testing containing 17661 images, and 2228 images for the query in total.

Evaluation metrics [36] The Cumulative Matching Characteristic (CMC) and mAP are used to evaluate the person re-ID task. For each query, its average precision is calculated from the accuracy recall curve. Then the average of the average precision in the query is measured using the mAP. The retrieval accuracy is reflected in the CMC, and the recall rate is reflected in the mAP.

4.2. Experiment Settings

4.2.1 UnityStyle Transfer Model

For the style transfer part, the hyperparameter is introduced in Eq. 2, setting $\lambda_{GAN} = 0.25$, $\lambda_{FM} = 0.1$, $\lambda_{ID} = 0.15$, $\lambda_{CYC} = 0.5$, $\lambda_{SS} = 0.7$, $\lambda_{L1} = 0.3$ for all datasets. Before the start of the follow-up experiment, we need to train a style transfer for each camera, and we need to train multiple style transfers under different camera numbers. So we need to train 6 transfers for Market-1501, and 8 transfers for DukeMTMC-reID. Each style transfer uses 256×256 images for 50 epochs training, making the resulting image to an excellent level.

4.2.2 Deep Re-ID Model

Our approach can be applied to many models, this paper uses three models (IDE[37], PCB[32], st-ReID[33]) to verify the effectiveness of our approach. We trained using the baseline model with an input image size of 256×128 and processed training images using random cropping [17], random horizontal flipping [31] and random erasing [42] during training. We performed 50 epochs training on the model, using 128 real images and 128 UnityStyle images for each batch of each epoch, and using the learning rate = 0.1 for SGD to solve the model.

Measure (%)	top-1	top-5	top-10	mAP
PAN [38]	82.8	-	-	63.4
GAN [40]	83.9	-	-	66.1
TriNet [10]	84.9	94.2	-	69.1
PAN+RE [38]	85.8	93.4	-	76.6
CamStyle [43]	89.5	-	-	71.6
PSE+ECN [29]	90.3	94.5	-	84.0
HA-CNN [19]	91.2	-	-	75.7
IDE [37]	85.7	93.1	95.3	65.9
PCB [32]	91.2	97.0	98.2	75.8
st-ReID [33]	98.0	98.9	99.1	95.5
IDE+UnityStyle	93.2	96.1	96.9	89.3
PCB+UnityStyle	95.8	97.9	98.7	93.6
st-ReID+UnityStyle	**98.5**	**99.0**	**99.1**	**95.8**

Table 1. Evaluation on the Market-1501 dataset.

Measure (%)	top-1	top-5	top-10	mAP
TriNet [10]	72.4	-	-	53.5
CamStyle [43]	78.3	-	-	57.6
PSE+ECN [29]	79.8	89.7	92.2	62.0
HA-CNN [19]	80.5	-	-	63.8
MLFN [1]	81.2	-	-	62.8
DuATM [30]	81.8	90.2	95.4	64.6
IDE [37]	72.3	86.2	89.5	51.8
PCB [32]	83.8	91.7	94.4	69.4
st-ReID [33]	94.5	96.8	97.1	92.7
IDE+UnityStyle	85.9	93.5	94.8	82.3
PCB+UnityStyle	89.3	95.7	96.2	85.7
st-ReID+UnityStyle	**95.1**	**97.0**	**97.3**	**93.6**

Table 2. Evaluation on the DuckMTMC-ReID dataset.

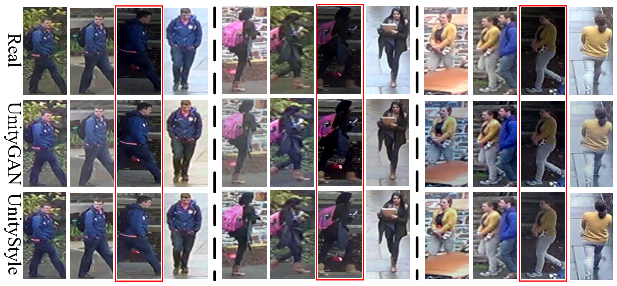
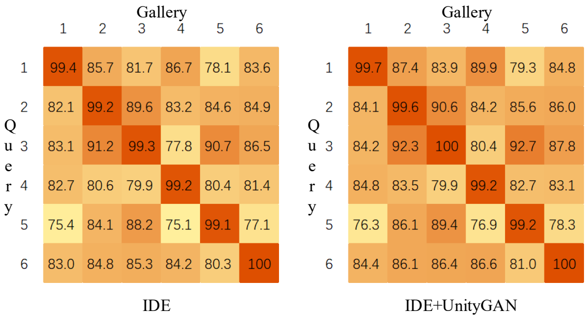


Figure 8. Some samples in DuckMTMC-ReID. Real images, Images generated by UnityGAN without style attention module and UnityStyle images. With the Style Attention module, our model can generate style-stabilized images. We can clearly observe that the dark image (Red box) been enhanced in appearance, and the image generated by UnityStyle has a uniform style and lighting.



IDE

Query \ Gallery	1	2	3	4	5	6
1	99.4	85.7	81.7	86.7	78.1	83.6
2	82.1	99.2	89.6	83.2	84.6	84.9
3	83.1	91.2	99.3	77.8	90.7	86.5
4	82.7	80.6	79.9	99.2	80.4	81.4
5	75.4	84.1	88.2	75.1	99.1	77.1
6	83.0	84.8	85.3	84.2	80.3	100

IDE+UnityGAN

Query \ Gallery	1	2	3	4	5	6
1	99.7	87.4	83.9	89.9	79.3	84.8
2	84.1	99.6	90.6	84.2	85.6	86.0
3	84.2	92.3	100	80.4	92.7	87.8
4	84.8	83.5	79.9	99.2	82.7	83.1
5	76.3	86.1	89.4	76.9	99.2	78.3
6	84.4	86.1	86.4	86.6	81.0	100

Figure 9. Accuracy comparison between **IDE** and **IDE+UnityGAN** with different cameras. Different gallery cameras can be seen with a horizontal orientation and different query cameras with a vertical orientation.

4.3. Evaluation

We compare the proposed method with existing methods on the Market-1501 and DuckMTMC-ReID (Table 1, 2). To verify the broad applicability of UnityStyle, we use IDE, PCB and st-ReID as the baseline. Then we apply our method to the baseline to verify the validity. The results of experiments show that our method achieves the state of the arts on both data sets.

In Market-1501, our method has varying degrees of improvement compared to the baseline. Compared with CamSyle, we also considered similar camera styles and achieved more effective data enhancement. Further we used re-ranking [41] technology to make our final experimental results reach **top-1 = 98.5%** and **mAP = 95.8%** with st-ReID. Our method achieved significant improvements over the underlying method. From the results we can get, our basic method not only improves the accuracy of top-1, but also ensures the accuracy of candidate results, so the final result is significantly improved after the introduction of re-ranking technology.

Same as Market-1501, our method has achieved good performance in DuckMTMC-ReID. Further, we used re-ranking technology to achieve our final experimental results of **top-1 = 95.1%**, **mAP = 93.6%** with st-ReID.

4.4. Experiment Analysis

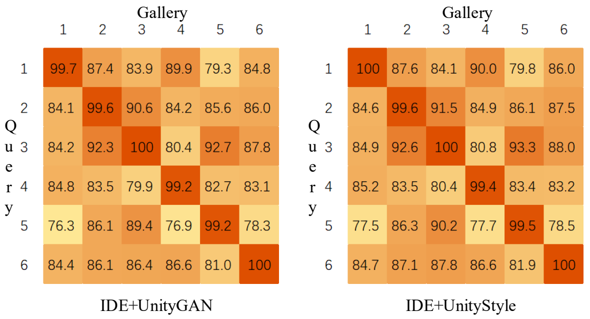
In this section, we will determine the effectiveness of each proposed module, and whether UnityGAN and UnityStyle improve the results from the perspective of datasets.

4.4.1 UnityGAN

We contrast the test performance of IBN and IBN+UnityGAN. To clearly show the test result changes between the same camera and different cameras, we compare the accuracy of each camera's query with the camera's gallery separately. In Fig. 9, UnityGAN has a positive effect on IDE, although UnityGAN generates unstable style images (Fig. 8).

4.4.2 UnityStyle

We propose the Style Attention Module and introduce the concept of UnityStyle, and we hope to use UnityStyle to



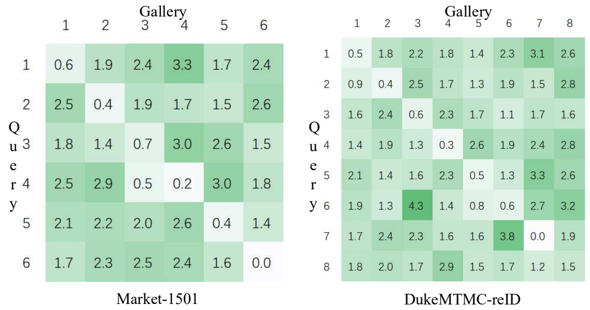
IDE+UnityGAN

Query \ Gallery	1	2	3	4	5	6
1	99.7	87.4	83.9	89.9	79.3	84.8
2	84.1	99.6	90.6	84.2	85.6	86.0
3	84.2	92.3	100	80.4	92.7	87.8
4	84.8	83.5	79.9	99.2	82.7	83.1
5	76.3	86.1	89.4	76.9	99.2	78.3
6	84.4	86.1	86.4	86.6	81.0	100

IDE+UnityStyle

Query \ Gallery	1	2	3	4	5	6
1	100	87.6	84.1	90.0	79.8	86.0
2	84.6	99.6	91.5	84.9	86.1	87.5
3	84.9	92.6	100	80.8	93.3	88.0
4	85.2	83.5	80.4	99.4	83.4	83.2
5	77.5	86.3	90.2	77.7	99.5	78.5
6	84.7	87.1	87.8	86.6	81.9	100

Figure 10. Accuracy comparison between **IDE+UnityGAN** and **IDE+UnityStyle** with different cameras. Different gallery cameras can be seen with a horizontal orientation and different query cameras with a vertical orientation.



Market-1501

Query \ Gallery	1	2	3	4	5	6
1	0.6	1.9	2.4	3.3	1.7	2.4
2	2.5	0.4	1.9	1.7	1.5	2.6
3	1.8	1.4	0.7	3.0	2.6	1.5
4	2.5	2.9	0.5	0.2	3.0	1.8
5	2.1	2.2	2.0	2.6	0.4	1.4
6	1.7	2.3	2.5	2.4	1.6	0.0

DukeMTMC-reID

Query \ Gallery	1	2	3	4	5	6	7	8
1	0.5	1.8	2.2	1.8	1.4	2.3	3.1	2.6
2	0.9	0.4	2.5	1.7	1.3	1.9	1.5	2.8
3	1.6	2.4	0.6	2.3	1.7	1.1	1.7	1.6
4	1.4	1.9	1.3	0.3	2.6	1.9	2.4	2.8
5	2.1	1.4	1.6	2.3	0.5	1.3	3.3	2.6
6	1.9	1.3	4.3	1.4	0.8	0.6	2.7	3.2
7	1.7	2.4	2.3	1.6	1.6	3.8	0.0	1.9
8	1.8	2.0	1.7	2.9	1.5	1.7	1.2	1.5

Figure 11. The image shows the comparison of results before and after using **UnityStyle** under two data sets.

remove the style differences of different camera images as much as possible to improve the results. In this section, we compare the test results of whether we use the UnityStyle. Fig. 8 visually shows the impact of the Style Attention Module. DuckMTMC-ReID has greater camera views changes than Market-1501, and our method can handle these changes. To clearly show the changes in the test results of the same and different cameras, we compare the accuracy between each camera's query and each camera's gallery separately. As can be seen from Fig. 10, after adding UnityStyle, the accuracy of different cameras is further increased, and the accuracy of the same cameras has also risen even though it has reached a very high level. It shows that the introduction of UnityStyle can improve the adaptability to style changes. Fig. 11 can clearly show the promotion by IDE after using UnityStyle.

4.4.3 Ablation Study

Table 3 shows the effect of using each module separately on the results of the two data sets. The IDE model is the baseline method, the UnityGAN model uses the unstable-style image to enhance the result in train phase, the UnityStyle model uses the UnityStyle image to enhance the result in train and test phase, and the UnityStyle+RE introduces re-ranking technology based on the UnityStyle model. From the results, we can see that the results are constantly improving as the introduction of the model. It's worth noting that we can achieve more significant improvement by using re-ranking technology after introducing UnityStyle, which means that UnityStyle increases the probability that there is a correct answer in the candidate, and re-ranking technology further increases the probability of top-1 the answer is selected. The experimental results prove that the combination of each part can achieve excellent results, and each component of the proposed method is mutually inseparable.

Dataset	Market-1501		DuckMTMC-ReID	
Measure (%)	top-1	mAP	top-1	mAP
IDE	85.7	65.9	72.3	51.8
IDE+RE	87.0	77.5	73.5	69.4
UnityGAN	89.8	73.0	79.1	60.6
UnityGAN+RE	90.9	83.1	82.7	76.5
UnityStyle	91.8	76.5	82.1	65.2
UnityStyle+RE	**93.2**	**89.3**	**85.9**	**82.3**

Table 3. The effect of each module. With the addition of modules, the model's effect is getting better and better. RE: re-ranking.

5. Conclusion

In this paper, we propose UnityStyle method, which can smooth the style disparities within the same camera and across different cameras. We firstly create UnityGAN to learn the style changes between cameras, producing shape-stable style-unity images for each camera. Motivated by the fact that structural information is contained in shallow layers, we add skip connections between multi-depth layers, thus retain more structural information and make the generated image structure more stable. Then, we propose UnityStyle images to eliminate style differences between different images, which makes a better match between query and gallery. It is ensured that images with a large difference in unity-style have a small effect on the style of the generated image, and therefore we can generate images with unity-style.These advantages make the proposed method perform better than existing methods.

Acknowledgments

This work is supported in part by China National 973 program 2014CB340301 and NSFC grant 6197023605, 61973250 and 61702415, Australian Research Council (ARC) Discovery Early Career Researcher Award (DECRA) under grant no. DE190100626, Air Force Research Laboratory and DARPA under agreement number FA8750-19-2-0501.

References

[1] Xiaobin Chang, Timothy M. Hospedales, and Tao Xiang. Multi-level factorisation net for person re-identification. In *CVPR*, 2018.

[2] W. Chen, X. Chen, J. Zhang, and K. Huang. Beyond triplet loss: a deep quadruplet network for person re-identification. In *CVPR*, 2017.

[3] De Cheng, Xiaojun Chang, Li Liu, Alexander G. Hauptmann, Yihong Gong, and Nanning Zheng. Discriminative dictionary learning with ranking metric embedded for person re-identification. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI 2017, Melbourne, Australia, August 19-25, 2017*, pages 964–970, 2017.

[4] De Cheng, Yihong Gong, Xiaojun Chang, Weiwei Shi, Alexander G. Hauptmann, and Nanning Zheng. Deep feature learning via structured graph laplacian embedding for person re-identification. *Pattern Recognit.*, 82:94–104, 2018.

[5] L. A. Gatys, A. S. Ecker, and M. Bethge. Image style transfer using convolutional neural networks. In *CVPR*, 2016.

[6] Yixiao Ge, Zhuowan Li, Haiyu Zhao, Guojun Yin, Shuai Yi, Xiaogang Wang, and hongsheng Li. Fd-gan: Pose-guided feature distilling gan for robust person re-identification. In *NeurIPS*, 2018.

[7] Aaron Gokaslan, Vivek Ramanujan, Daniel Ritchie, Kwang In Kim, and James Tompkin. Improving shape deformation in unsupervised image-to-image translation. In *ECCV*, 2018.

[8] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. In *NeurIPS*, 2014.

[9] Lingxiao He, Jian Liang, Haiqing Li, and Zhenan Sun. Deep spatial feature reconstruction for partial person re-identification: Alignment-free approach. In *CVPR*, 2018.

[10] A. Hermans, L. Beyer, and B. Leibe. In defense of the triplet loss for person re-identification. *arXiv preprint arXiv:1703.07737*, 2017.

[11] Xun Huang and Serge Belongie. Arbitrary style transfer in real-time with adaptive instance normalization. In *ICCV*, 2017.

[12] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *ICML*, 2015.

[13] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros. Image-to-image translation with conditional adversarial networks. In *CVPR*, 2017.

[14] J. Johnson, A. Alahi, and L. Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *ECCV*, 2016.

[15] Taeksoo Kim, Moonsu Cha, Hyunsoo Kim, Jung Kwon Lee, and Jiwon Kim. Learning to discover cross-domain relations with generative adversarial networks. In *ICML*, 2017.

[16] M. Kostinger, M. Hirzer, P. Wohlhart, P. M. Roth, and H. Bischof. Large scale metric learning from equivalence constraints. In *CVPR*, 2012.

[17] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *NeurIPS*, 2012.

[18] Wei Li, Xiatian Zhu, and Shaogang Gong. Person re-identification by deep joint learning of multi-loss classification. In *IJCAI*, 2017.

[19] Wei Li, Xiatian Zhu, and Shaogang Gong. Harmonious attention network for person re-identificatio. In *CVPR*, 2018.

[20] Zhihui Li, Wenhe Liu, Xiaojun Chang, Lina Yao, Mahesh Prakash, and Huaxiang Zhang. Domain-aware unsupervised cross-dataset person re-identification. In *ADMA*, 2019.

[21] Xiaodan Liang, Hao Zhang, and Eric P. Xing. Generative semantic manipulation with contrasting gan. In *CVPR*, 2018.

[22] Wenhe Liu, Xiaojun Chang, Ling Chen, Dinh Phung, Xiaoqin Zhang, Yi Yang, and Alexander G. Hauptmann. Pair-based uncertainty and diversity promoting early active learning for person re-identification. *ACM TIST*, 11(2):21:1–21:15, 2020.

[23] Wenhe Liu, Xiaojun Chang, Ling Chen, and Yi Yang. Early active learning with pairwise constraint for person re-identification. In *ECML PKDD*, 2017.

[24] Wenhe Liu, Xiaojun Chang, Ling Chen, and Yi Yang. Semi-supervised bayesian attribute learning for person re-identification. In *AAAI*, 2018.

[25] Xingang Pan, Ping Luo, Jianping Shi, and Xiaoou Tang. Two at once: Enhancing learning and generalization capacities via ibn-net. In *ECCV*, 2018.

[26] X. Qian, Y. Fu, Y.G. Jiang, T. Xiang, and X. Xue. Deep spatial feature reconstruction for partial person re-identification: Alignment-free approach. In *ICCV*, 2017.

[27] Xuelin Qian, Yanwei Fu, Tao Xiang, Wenxuan Wang, Jie Qiu, Yang Wu, Yu-Gang Jiang, and Xiangyang Xue. Pose-normalized image generation for person re-identification. In *ECCV*, 2018.

[28] A. Radford, L. Metz, and S. Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. In *ICLR*, 2016.

[29] M. Saquib Sarfraz, Arne Schumann, Andreas Eberle, and Rainer Stiefelhagen. A pose-sensitive embedding for person re-identification with expanded cross neighborhood re-ranking. In *CVPR*, 2018.

[30] Jianlou Si, Honggang Zhang, Chun-Guang Li, Jason Kuen, Xiangfei Kong, Alex C. Kot, and Gang Wang. Dual attention matching network for context-aware feature sequence based person re-identification. In *CVPR*, 2018.

[31] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In *ICLR*, 2015.

[32] Yifan Sun, Liang Zheng, Yi Yang, Qi Tian, and Shengjin Wang. Beyond part models: Person retrieval with refined part pooling (and a strong convolutional baseline). In *ECCV*, 2018.

[33] Guangcong Wang, Jianhuang Lai, Peigen Huang, and Xiaohua Xie. Spatial-temporal person re-identification. In *AAAI*, 2019.

[34] L. Zhang, T. Xiang, and S. Gong. Learning a discriminative null space for person re-identification. In *CVPR*, 2016.

[35] Ying Zhang, Tao Xiang, Timothy Hospedales, and Huchuan Lu. Deep mutual learning. In *CVPR*, 2018.

[36] L. Zheng, L. Shen, L. Tian, S. Wang, J. Wang, and Q. Tian. Scalable person re-identification: A benchmark. In *ICCV*, 2015.

[37] L. Zheng, Y. Yang, and A. G. Hauptmann. Person reidentification: Past, present and future. *arXiv preprint arXiv:1610.02984*, 2016.

[38] Liang Zheng, Zhedong Zheng, and Yi Yang. Pedestrian alignment network for person re-identification. In *ICCV*, 2017.

[39] Zhedong Zheng, Xiaodong Yang, Zhiding Yu, Liang Zheng, Yi Yang, and Jan Kautz. Joint discriminative and generative learning for person re-identification. In *CVPR*, 2019.

[40] Zhedong Zheng, Liang Zheng, and Yi Yang. Unlabeled samples generated by gan improve the person re-identification baseline in vitro. In *ICCV*, 2017.

[41] Zhun Zhong, Liang Zheng, Donglin Cao, and Shaozi Li. Reranking person re-identification with k-reciprocal encoding. In *CVPR*, 2017.

[42] Zhun Zhong, Liang Zheng, Guoliang Kang, Shaozi Li, and Yi Yang. Random erasing data augmentation. *arXiv preprint arXiv:1708.04896, 2017*, 2017.

[43] Zhun Zhong, Liang Zheng, Zhedong Zheng, Shaozi Li, and Yi Yang. Camera style adaptation for person reidentification. In *CVPR*, 2018.

[44] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A. Efros. Unpaired imageto-image translation using cycleconsistent adversarial networks. In *ICCV*, 2017.