重庆大学
CHONGQING UNIVERSITY

Style Uncertainty Based Self-Paced Meta Learning for Generalizable Person Re-Identiffcation

Style Uncertainty Based Self-Paced Meta Learning
for Generalizable Person Re-Identification

Lei Zhang, *Senior Member, IEEE*, Zhipu Liu, Wensheng Zhang, and David Zhang, *Life Fellow, IEEE*

组员：游励东，周旺旺，刘泽新

指导老师：张磊

时间：2024.4.10

目录

重庆大学
CHONGQING UNIVERSITY

I. 引言
II、背景及相关工作

汇报人：游励东

时间：2024.04.10

SuA-SpML目的：将训练域更好地泛化到未知目标域

- 在训练过程中用高斯噪声扰动实例风格，使训练数据的风格随机化，以增加训练域的多样性
- 通过模拟人类的学习机制，逐步提高模型对未知目标域的泛化能力

人员重新鉴定（Re-ID）的目的是在不相连的摄像头下匹配行人

比如：一个摄像头的图像泛化到其他摄像头中仍然能有精确的识别效果

[18]S. Zhao, M. Gong, T. Liu, H. Fu, and D. Tao, “Domain generalization via entropy regularization,” in Proc. Adv. Neural Inf. Process. Syst., vol. 33,2020, pp. 16096–16107.

基于风格不确定性增强的自定进度Meta学习（SuA-SpML）

本文提出了SuA-SpML方法来DG - ReID的领域泛化问题。首先提出了风格不确定性增强(SuA)，通过在训练过程中使用高斯噪声扰动实例风格来实现源域的多样化，紧接着提出了一种新的自定步元学习方法，帮助模型从简单到困难学习，最后提出了一种距离图对齐损失，用于对齐源域之间的特征关系分布来促进网络探索与域无关的表示。

文章贡献

- 全面回顾了传统方法的不足
- 全面提出了基于**SuA**的SpML方法
- 全面总结了基于**SuA**的SpML方法的学习机制
- 通过四个实验，详细讨论SuA-SpML方法的应用前景

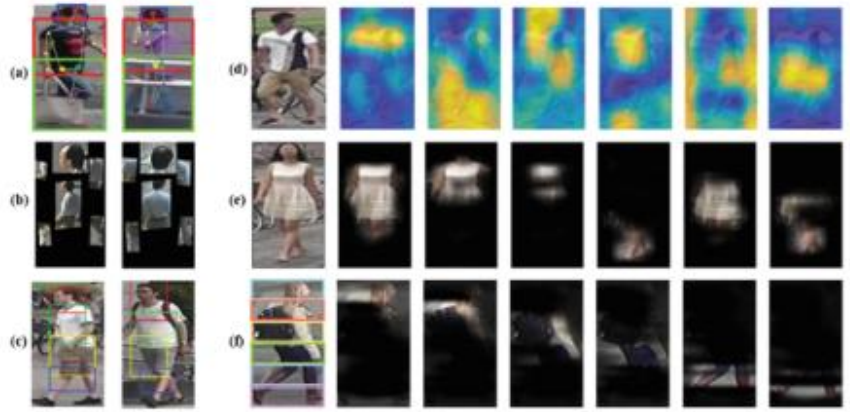
[1] Z. Zheng, X. Yang, Z. Yu, L. Zheng, Y. Yang, and J. Kautz, “Joint discriminative and generative learning for person re-identification,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2019, pp. 2138–2147.

传统方法：空间对齐模型（FGSAM）

- 抑制不准确的检测和遮挡所造成的冗余背景，提出了一个局部增强对齐模式，以解决不同的局部之间的失调问题。

传统方法：基于零件的卷积基线（PCB）的网络

- 是定位好每个零件都要，而不是使用外部资源，采用部分级功能提供细粒度的行人图像描述的信息。

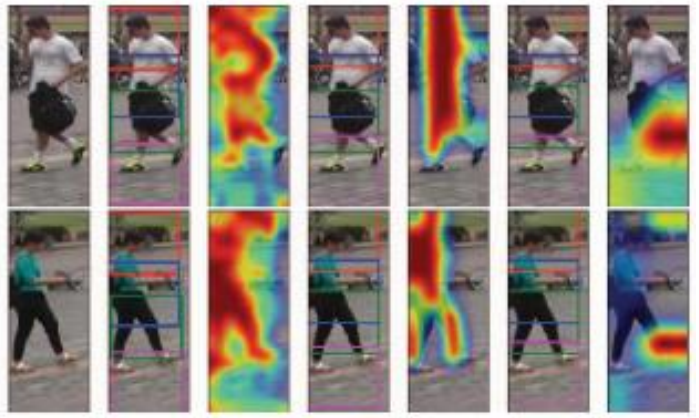


[2] Q. Zhou et al., “Fine-grained spatial alignment model for person reidentiffcation with focal triplet loss,” IEEE Trans. Image Process., vol. 29, pp. 7578–7589, 2020.
[3] Y. Sun, L. Zheng, Y. Yang, Q. Tian, and S. Wang, “Beyond part models: Person retrieval with reffned part pooling (and a strong convolutional baseline),” in Proc. ECCV, vol. 2018, pp. 480–496.

重庆大学 CHONGQING UNIVERSITY

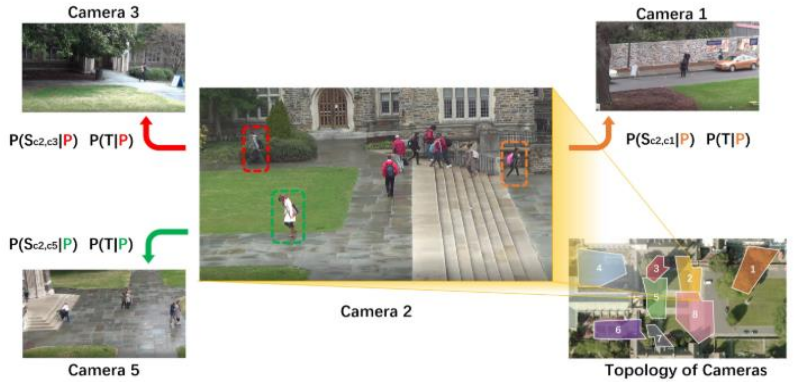
传统方法：协调注意力CNN（HA-CNN）模型

- 通过最大化受re-id区分学习约束的不同视觉注意水平的补充信息，优化不受控制（未对齐）图像中的人物re-id



传统方法：实例级和时空非纠缠Re-ID模型（InSTD）

- 提出个性化信息的框架中，如明确判断移动方向，以进一步缩小搜索空间



[4] W. Li, X. Zhu, and S. Gong, "Harmonious attention network for person re-identiffcation," in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., Jun. 2018, pp. 2285–2294.
[5] M. Ren, L. He, X. Liao, W. Liu, Y. Wang, and T. Tan, "Learning instance-level spatial–temporal patterns for person re-identiffcation," in Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV), Oct. 2021, pp. 14930–14939

研究SuA-SpML的动机

DG ReID的挑战性

↓

训练域很难泛化到未知目标域

背景

↓

数据增强已被证明有利于更好地利用源数据来提高模型的泛化

因此

传统方法：生成像素级图像

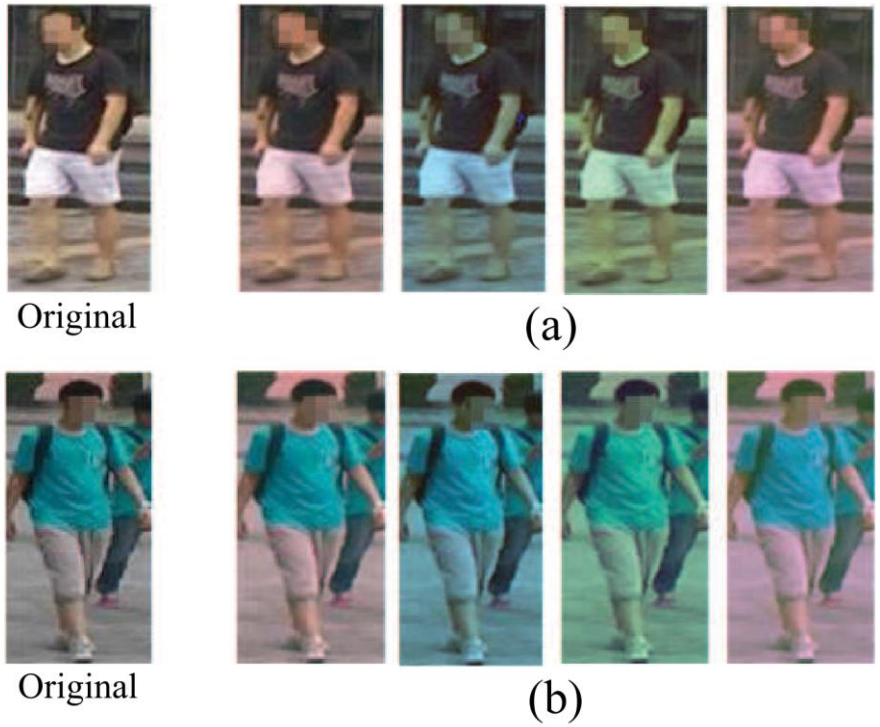
- 需要额外设计和训练一个生成网络(非常复杂)
- 提供的增强数据的多样性有限

>

提出SuA-SpML

风格不确定性增强（SuA）的主要思想

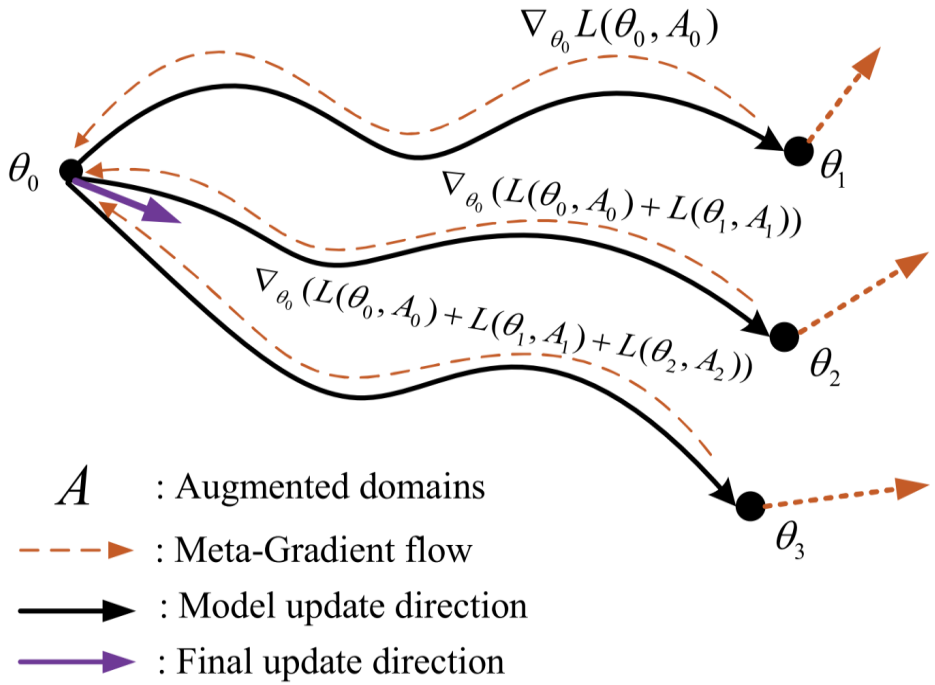
- 在训练过程中用高斯噪声扰动实例风格，使训练数据的风格随机化，以增加训练域的多样性



- 左图，通过在SuA中使用不同的高斯噪声参数设置，我们在图2中展示了(a)(b)两个增强的示例

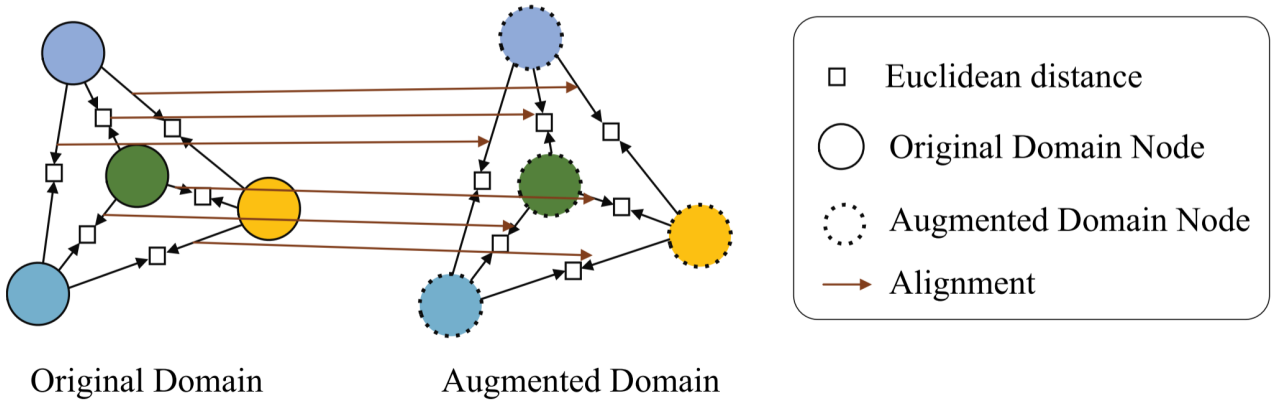
自步Meta学习(SpML)的主要思想

- 将传统的单阶段元学习扩展到多阶段的训练过程。逐步增加方差，来获得更好的泛化适应。
- 其合理性在于通过模拟人类的学习机制，逐步提高模型对未知目标域的泛化能力



距离图对齐损失的主要思想

- 对齐域之间的特征关系分布，以促进网络探索图像的域不变表示



- 左图，将源域和未知目标的特征一一相连后，再进行映射，对齐

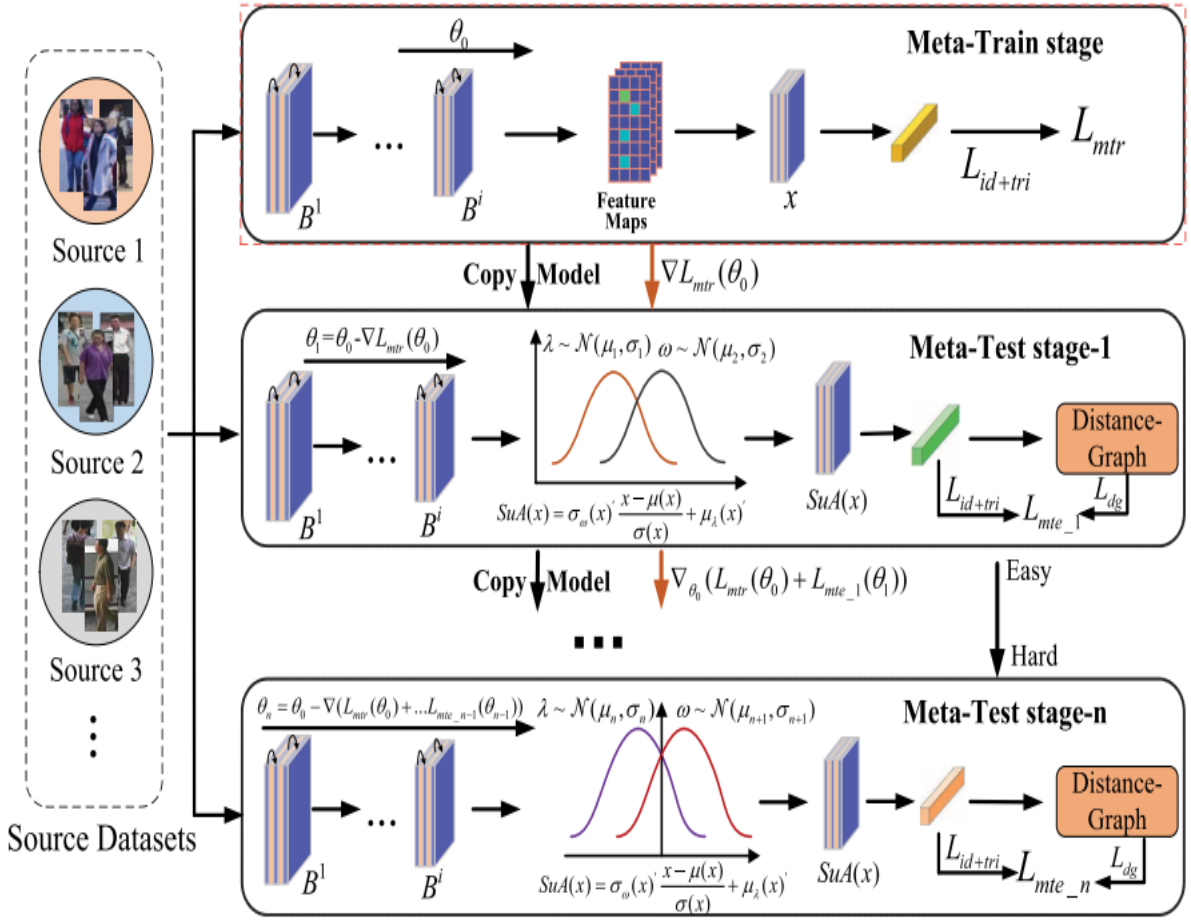
重庆大学
CHONGQING UNIVERSITY

III. 本文方法

汇报人：周旺旺

时间：2024.04.10

SUA-SPML框架



左图展示了一个元学习模型的训练和测试流程。

在元训练阶段，通过最小化训练损失 来更新模型参数。

$$\theta_i = \theta_0 - \nabla L_{mtr}(\theta_0)$$

在元测试阶段进行特征标准化。

$$SuA(x) = \sigma(x)\frac{x-\mu(x)}{\sigma(x)} + \mu(x)$$

在元测试阶段，用于优化模型的损失函数。

$$L_{id+tri} + L_{mte}$$

SuA算法

Algorithm 1 PyTorch-Like Pseudo-Code for SuA

```
# x: input features of shape (B, C, H, W)
# eps: a small value added before square root for numerical stability
# mode: control the difficulty of the augmentation style
if not in training mode:
    return x
B = x.size(0) # batch size
mu = x.mean(dim=[2, 3], keepdim=True) # compute instance mean
var = x.var(dim=[2, 3], keepdim=True) # compute instance variance
sig = (var + eps).sqrt() # compute instance standard deviation
mu, sig = mu.detach(), sig.detach() # block gradients
x_normed = (x - mu) / sig # normalize input
if mode == 0: # data without augmentation
    mu_per = 0
    sig_per = 0
elif mode == 1: # augment data with easy style
    mu_per = torch.normal(0, 0.1, mu.size(), requires_grad=False)
    sig_per = torch.normal(0, 0.1, sig.size(), requires_grad=False)
elif mode == 2: # augment data with harder style
    mu_per = torch.normal(0, 0.15, mu.size(), requires_grad=False)
    sig_per = torch.normal(0, 0.15, sig.size(), requires_grad=False)
mu_SuA = mu + mu_per # generate SuA mean
sig_SuA = sig + sig_per # generate SuA standard deviation
return x_normed * sig_SuA + mu_SuA # denormalize input using
the SuA statistics
```

该算法计算实例均值和标准差，根据不同模式加入噪声，生成增强后的均值和标准差，再对归一化输入反归一化，实现数据增强。

1.计算实例均值和标准差：

$$\mu = \frac{1}{HW}\sum_{h=1}^{H}\sum_{w=1}^{W} x_{hw}$$

$$\sigma = \sqrt{\frac{1}{HW}\sum_{h=1}^{H}\sum_{w=1}^{W}(x_{hw}-\mu)^2 + \epsilon}$$

2.归一化输入：

$$x_{\text{normed}} = \frac{x-\mu}{\sigma}$$

SuA算法

Algorithm 1 PyTorch-Like Pseudo-Code for SuA

```
# x: input features of shape (B, C, H, W)
# eps: a small value added before square root for numerical stability
# mode: control the difficulty of the augmentation style
if not in training mode:
    return x
B = x.size(0) # batch size
mu = x.mean(dim=[2, 3], keepdim=True) # compute instance mean
var = x.var(dim=[2, 3], keepdim=True) # compute instance variance
sig = (var + eps).sqrt() # compute instance standard deviation
mu, sig = mu.detach(), sig.detach() # block gradients
x_normed = (x - mu) / sig # normalize input
if mode == 0: # data without augmentation
    mu_per = 0
    sig_per = 0
elif mode == 1: # augment data with easy style
    mu_per = torch.normal(0, 0.1, mu.size(), requires_grad=False)
    sig_per = torch.normal(0, 0.1, sig.size(), requires_grad=False)
elif mode == 2: # augment data with harder style
    mu_per = torch.normal(0, 0.15, mu.size(), requires_grad=False)
    sig_per = torch.normal(0, 0.15, sig.size(), requires_grad=False)
mu_SuA = mu + mu_per # generate SuA mean
sig_SuA = sig + sig_per # generate SuA standard deviation
return x_normed * sig_SuA + mu_SuA # denormalize input using
the SuA statistics
```

3.数据增强：

mode==0：

$$\mu_{\text{per}} = 0$$

$$\sigma_{\text{per}} = 0$$

mode==1：

$$\mu_{\text{per}} \sim \mathcal{N}(0, 0.1)$$

$$\sigma_{\text{per}} \sim \mathcal{N}(0, 0.1)$$

mode==2：

$$\mu_{\text{per}} \sim \mathcal{N}(0, 0.15)$$

$$\sigma_{\text{per}} \sim \mathcal{N}(0, 0.15)$$

4.生成新的均值和标准差：

$$\mu_{\text{SuA}} = \mu + \mu_{\text{per}}$$

$$\sigma_{\text{SuA}} = \sigma + \sigma_{\text{per}}$$

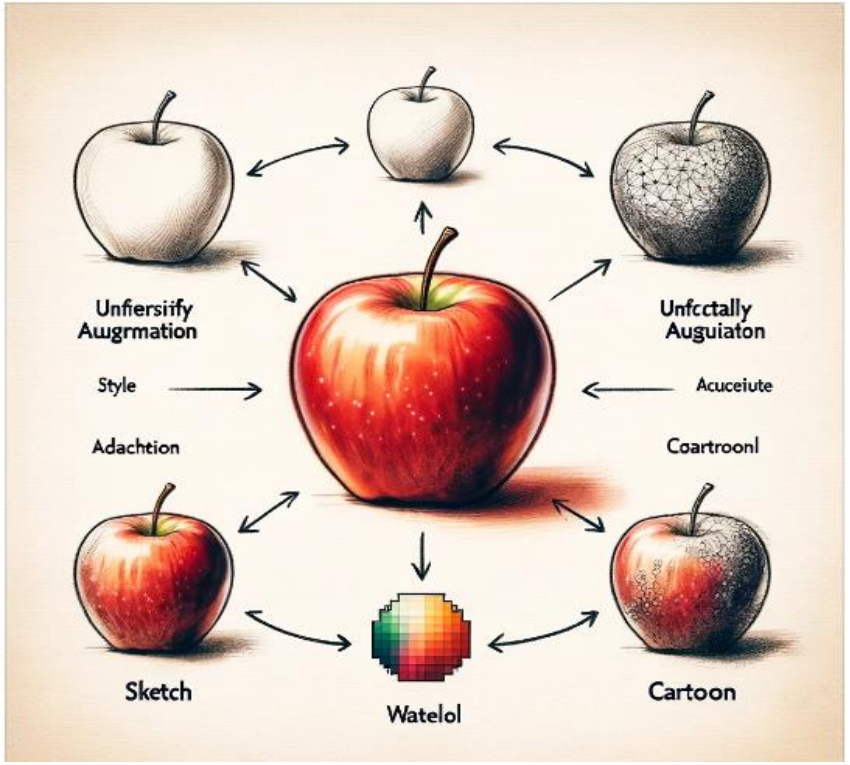
5.反归一化输入：

$$x_{\text{SuA}} = x_{\text{normed}} \cdot \sigma_{\text{SuA}} + \mu_{\text{SuA}}$$

该算法在训练过程中应用风格不确定性增强（SuA），生成不同风格的数据变体，以增强模型的稳定性和适应性。

风格不确定性增强

风格不确定性增强即通过对输入数据的均值和标准差进行随机扰动，生成不同风格的变体，以增加数据多样性和提升模型鲁棒性。左图展示了原始苹果通过素描、像素化、卡通等风格变换，右图展示了原始橙子的多种风格变体，均体现了这种增强方法的应用。



实例归一化

实例归一化的步骤：首先计算每个通道的均值和标准差，然后使用这些统计量对输入特征进行归一化，并通过可训练的参数进行调整，完成整个归一化过程。

1.计算每个通道的均值：

$$\mu(x)_c = \frac{1}{HW}\sum_{h=1}^{H}\sum_{w=1}^{W} x_{c,h,w}$$

2.计算每个通道的标准差：

$$\sigma(x)_c = \sqrt{\frac{1}{HW}\sum_{h=1}^{H}\sum_{w=1}^{W}(x_{c,h,w} - \mu(x)_c)^2}$$

3.使用均值和标准差对输入归一化，并应用可训练参数：

$$IN(x) = \gamma\frac{x - \mu(x)}{\sigma(x)} + \beta$$

最后，通过归一化步骤调整输入特征，应用可训练的参数 γ和 β，完成归一化过程。

风格不确定性

风格不确定性的步骤：这些公式通过随机扰动均值和标准差引入风格不确定性。先从正态分布采样扰动参数，再调整输入数据的均值和标准差，最终构建增强数据，增加数据多样性，提升模型鲁棒性和泛化能力。

1.均值和标准差的扰动：

$$\mu_\lambda(x)' = \mu(x)_c + \lambda_c \sigma(x)_c$$

$$\sigma_\omega(x)' = \sigma(x)_c + \omega_c$$

2.正态分布中采样：

$$\lambda \sim \mathcal{N}(\mu_1, \sigma_1)$$

$$\omega \sim \mathcal{N}(\mu_2, \sigma_2)$$

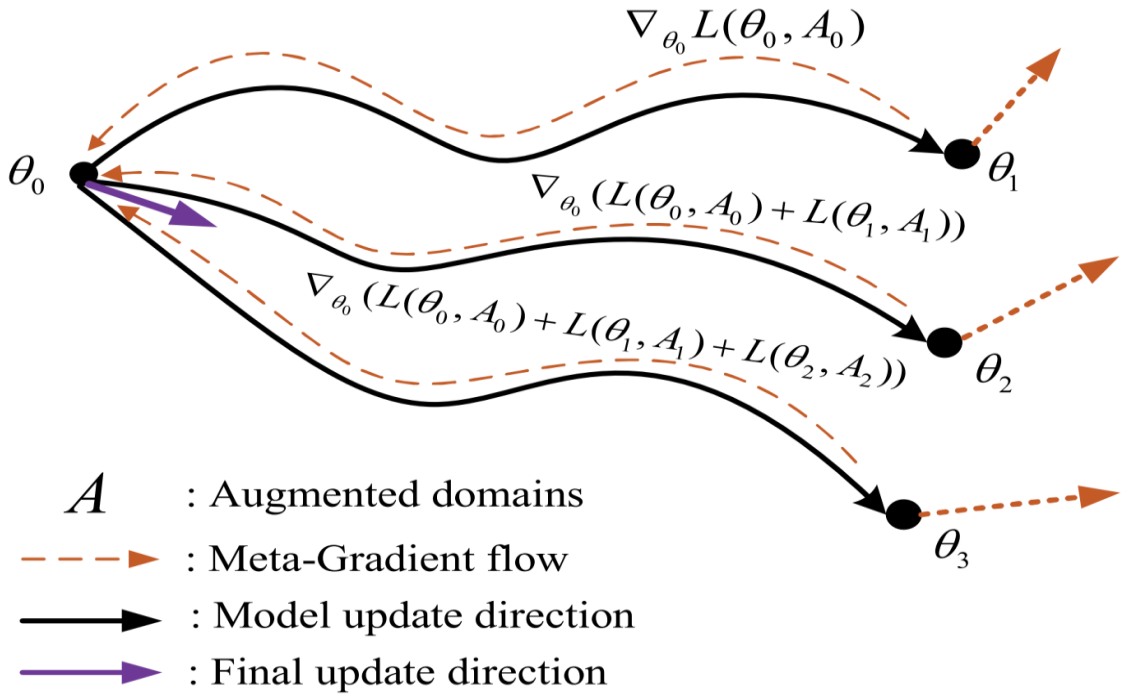
3.构建风格不确定性增强：

$$SuA(x) = \sigma_\omega(x)\frac{x - \mu(x)}{\sigma(x)} + \mu_\lambda(x)'$$

最后，使用扰动后的均值和标准差对输入数据进行重新构建，实现风格不确定性增强。

自定进度元学习过程概述

下图展示了参数 θ的逐步优化过程，通过在不同增强域 A 上进行元梯度流（虚线）和模型更新方向（实线）的迭代优化。起始于 $\theta 0$，首先通过 $\nabla\theta 0 L(\theta 0, A0)$ 更新到 $\theta 1$，然后通过 $\nabla\theta 0(L(\theta 0, A0)+L(\theta 1, A1))$更新到 $\theta 2$，依次类推，最终达到最优参数 $\theta 3$。紫色箭头表示最终的更新方向。



SpML的训练过程

Algorithm 2 Training Procedure of SpML

Input: n augmented domains $X = \{x, \hat{x}^1, ..., \hat{x}^n\}$.
Init: The initial model with parameters θ_0;
Inner loop learning rate α;
Outer loop learning rate η;

for iter in SpML **do**
  Divide learning process into multiple stages;
  Meta-Train Stage:
  Take the original domain data x ;
  Compute meta-train loss $L_{mtr}(\theta_0)$;
  Optimize the first stage model by SGD and inner loop learning rate α;
  $\theta_1 \leftarrow SGD(\nabla L_{mtr}(\theta_0), x, \alpha)$;
  Meta-Test Stage1:
  Take the augmented domain data $\hat{x}^1$;
  Compute the meta-test stage1 loss $L_{mte\_1}(\theta_1)$;
  Optimize the second stage model by the combination of the first two losses:
  $\theta_2 \leftarrow SGD(\nabla L_{mtr}(\theta_0) + \nabla L_{mte\_1}(\theta_1), \hat{x}^1, \alpha)$;
  Meta-Test Stage2:
  Take the augmented domain data $\hat{x}^2$;
  Compute the meta-test stage1 loss $L_{mte\_1}(\theta_1)$ and optimize the third stage model:
  $\theta_3 \leftarrow SGD(\nabla L_{mtr}(\theta_0) + \nabla L_{mte\_1}(\theta_1) + \nabla L_{mte\_2}(\theta_2), \hat{x}^2, \alpha)$
  Repeat this operation until training all augmented data
End;
Optimize the original model:
$\theta \leftarrow Adam((\nabla L_{mtr}(\theta_0) + \nabla L_{mte\_1}(\theta_1) + ...), \eta)$

该算法通过逐步累加训练和测试损失，使用多阶段优化来逐渐调整模型参数，从而增强模型的泛化能力和稳定性。

1.元训练阶段：

$$\theta_1 = \theta_0 - \alpha \nabla L_{mtr}(\theta_0)$$

2.元测试阶段1：

$$\theta_2 = \theta_1 - \alpha \nabla (L_{mtr}(\theta_0) + L_{mte\_1}(\theta_1))$$

3.元测试阶段2：

$$\theta_3 = \theta_2 - \alpha \nabla (L_{mtr}(\theta_0) + L_{mte\_1}(\theta_1) + L_{mte\_2}(\theta_2))$$

SpML的训练过程

Algorithm 2 Training Procedure of SpML

Input: n augmented domains $X = \{x, \hat{x}^1, ..., \hat{x}^n\}$.
Init: The initial model with parameters θ_0;
Inner loop learning rate α;
Outer loop learning rate η;

for iter in SpML **do**
  Divide learning process into multiple stages;
  Meta-Train Stage:
  Take the original domain data x ;
  Compute meta-train loss $L_{mtr}(\theta_0)$;
  Optimize the first stage model by SGD and inner loop learning rate α;
  $\theta_1 \leftarrow SGD(\nabla L_{mtr}(\theta_0), x, \alpha)$;
  Meta-Test Stage1:
  Take the augmented domain data $\hat{x}^1$;
  Compute the meta-test stage1 loss $L_{mte\_1}(\theta_1)$;
  Optimize the second stage model by the combination of the first two losses:
  $\theta_2 \leftarrow SGD(\nabla L_{mtr}(\theta_0) + \nabla L_{mte\_1}(\theta_1), \hat{x}^1, \alpha)$;
  Meta-Test Stage2:
  Take the augmented domain data $\hat{x}^2$;
  Compute the meta-test stage1 loss $L_{mte\_1}(\theta_1)$ and optimize the third stage model:
  $\theta_3 \leftarrow SGD(\nabla L_{mtr}(\theta_0) + \nabla L_{mte\_1}(\theta_1) + \nabla L_{mte\_2}(\theta_2), \hat{x}^2, \alpha)$
  Repeat this operation until training all augmented data
End;
Optimize the original model:
$\theta \leftarrow Adam((\nabla L_{mtr}(\theta_0) + \nabla L_{mte\_1}(\theta_1) + ...), \eta)$

4.一般化公式：

$$\theta_{i+1} = \theta_i - \alpha \nabla \left(L_{mtr}(\theta_0) + \sum_{j=1}^{i} L_{mte\_j}(\theta_j) \right)$$

5.最终优化：

$$\theta = \theta - \eta \nabla \left(L_{mtr}(\theta_0) + \sum_{i=1}^{n} L_{mte\_i}(\theta_i) \right)$$

这些公式展示了逐步累加损失，通过梯度下降优化参数，以提高模型泛化能力，最终使用Adam优化器进行整体优化。

距离图对齐

这些公式通过计算向量间的欧氏距离和其预测值的差异，形成距离图对齐损失，以优化模型使其预测的距离图与真实距离图一致。

1.计算两个向量 之间的欧氏距离：

$$m_{ij} = d(V_i, V_j)$$

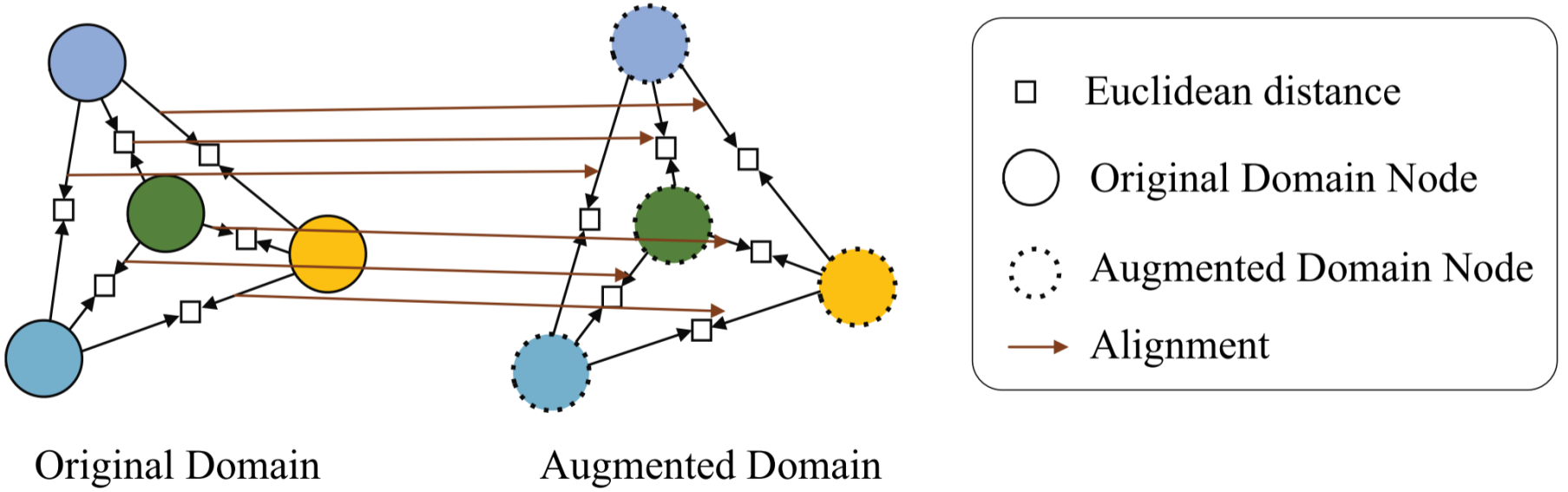
2.通过计算真实距离图 与预测距离图之间的差异，得到距离图对齐损失：

$$L_{dg} = \frac{1}{B \times B} \sum_{i=1}^{B} \sum_{j=1}^{B} \| m_{ij} - \hat{m}_{ij} \|_2$$

最后，优化模型使其预测的距离图与真实距离图一致。

距离图对齐损失过程

这张图展示了原始域和增强域节点的对齐过程。原始域节点用实线圆圈表示，增强域节点用虚线圆圈表示，方块表示欧氏距离。箭头表示对齐过程，通过对齐两个域中的节点，确保特征保持一致性和相似性，以优化模型性能，增强特征的鲁棒性和泛化能力。



目标函数

这些公式通过结合识别损失、三元组损失和距离图对齐损失，形成一个综合损失函数 L，以优化模型在分类和检索任务中的整体性能。

1.识别损失函数 Lid用于优化分类任务。

$$L_{id} = -\log \frac{\exp(\mathcal{M}[i] f(x_i)/\tau)}{\sum_{k=1}^{nT} \exp(\mathcal{M}[k] f(x_i)/\tau)}$$

2.三元组损失函数 $Ltri$ 确保相同身份样本距离更近，不同身份样本距离更远。

$$L_{tri} = \sum_{i=1}^{P} \sum_{j=1}^{\Omega} \left[d_{pos}^{i,j} - d_{neg}^{i,j} + \delta \right]_+$$

3.距离图对齐损失 Ldg 通过计算预测距离图与真实距离图的差异来优化模型。

$$L_{dg} = \frac{1}{B \times B} \sum_{i=1}^{B} \sum_{j=1}^{B} \| m_{ij} - \hat{m}_{ij} \|_2$$

4.这些损失函数结合形成组合损失 L，综合优化模型分类和检索性能。

$$L = L_{id} + L_{tri} + L_{dg}$$

重庆大学
CHONGQING UNIVERSITY

IV. 实验
V.结论

汇报人：刘泽新

时间：2024.04.10

行人重识别数据集

1.Market-1501

包括由6个摄像头(其中5个高清摄像头和1个低清摄像头)拍摄到的1501个行人、32668个检测到的行人矩形框。

2.DukeMTMC-reID

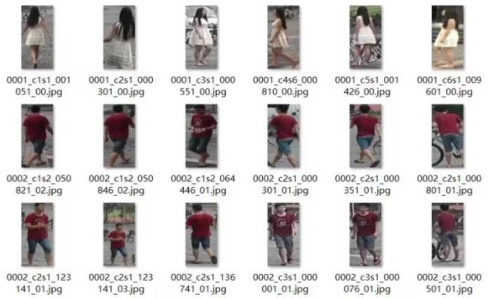
包括36411张图像。一共有1404个人出现在大于两个摄像头下，有408个人只出现在一个摄像头下。

3. CUHK 03

包括由6台摄像机拍摄的14097张图片，1467个行人的图像组成。

4. MSMT 17

包括由15个不同相机拍摄的4101个行人，126441张图像组成。



评估方法

- **Protocol-1**：仅使用源域的训练集进行训练。
- **Protocol-2**：使用源域的训练集和测试集来训练模型。
- 通过Top-k的平均精度值(mAP)和累积匹配特征(CMC)曲线来评估性能。

- **主干网络**：ResNet-50
- **输入图像**：256 × 128
- **batch size**：64

参数设置

SuA对域增强

a) 高斯噪声均值和方差：0、0.1

b) 在SpML下一阶段的方差增加 0.05

Adam优化器

- 权重衰减：0.0005
- 内部循环α和外部循环η学习率：3.5×10^{-5}
- 前10个epoch线性增加到3.5×10^{-4}
- 20、30、40 epoch α和η分别衰减0.1

与最新方法进行比较

使用leave-one-domain-out协议将四个数据集划分为训练/测试域

M:Market-1501 D:DukeMTMC-ReID C:CUHK 03 MS:MSMT 17

Method	Backbone	MS+D+C→M		MS+M+C→D		M+D+C→MS		MS+M+D→C		Average	
		mAP	Rank-1	mAP	Rank-1	mAP	Rank-1	mAP	Rank-1	mAP	Rank-1
QAConv [22]	ResNet50	39.5	68.6	43.4	64.9	10.0	29.9	19.2	22.9	28.0	46.6
CBN [75]	ResNet50	47.3	74.7	50.1	70.0	15.4	37.0	25.7	25.2	34.6	51.7
SNR [20]	SNR	48.5	75.2	48.3	66.7	13.8	35.1	29.0	29.1	34.9	51.5
DAML [30]	ResNet50	49.3	75.5	47.6	66.5	12.6	32.2	29.3	29.8	34.7	51.0
CamStyle [76]	ResNet50	48.7	75.1	47.1	66.0	11.7	31.1	28.6	29.3	34.0	50.4
M^3L [39]	ResNet50	51.1	76.5	48.2	67.1	13.1	32.0	30.9	31.9	35.8	51.9
M^3L [39]	ResNet50-IBN	52.5	78.3	48.8	67.2	15.4	37.1	31.4	31.6	37.0	53.6
OSNet [77]	OSNet	44.2	72.5	47.0	65.2	12.6	33.2	23.3	23.9	31.8	48.7
OSNet-IBN [77]	OSNet-IBN	44.9	73.0	45.7	64.6	16.2	39.8	25.4	25.7	33.0	50.8
OSNet-AIN [77]	OSNet-AIN	45.8	73.3	47.2	65.6	16.2	40.2	27.1	27.4	34.1	51.6
MetaBIN [23]	ResNet50-IBN	52.7	78.8	50.4	68.5	14.8	36.1	29.3	29.4	36.8	53.2
SuA-SpML (ours)	ResNet50	**56.6**	**81.0**	**51.2**	**70.8**	**18.2**	**42.8**	**32.0**	**33.1**	**39.5**	**56.9**
SuA-SpML (ours)	ResNet50-IBN	**59.1**	**83.2**	**51.9**	**71.6**	**20.4**	**47.1**	**33.3**	**34.4**	**41.2**	**59.1**

与最新方法进行比较

测试域	主干网络	mAP	Rank-1
MS+D+C→M	ResNet50	56.6	81.0
	ResNet50-IBN	59.1	83.2
MS+M+C→D	ResNet50	51.2	70.8
	ResNet50-IBN	51.9	71.6
M+D+C→MS	ResNet50	18.2	42.8
	ResNet50-IBN	20.4	47.1
MS+M+D→C	ResNet50	32.0	33.1
	ResNet50-IBN	33.3	34.4

本文方法在主干网络为ResNet50和ResNet50-IBN的测试结果均优于其他方法，实现了最佳性能

本文的方法具有更好的泛化能力！

四个测试域下ResNet50主干网络平均mAP和Rank-1比M^3L高出3.7%和5.0%

ResNet50-IBN主干网络平均mAP和Rank-1比M^3L高出4.2%和5.5%

Protocol-2方案进行比较

Method	MS+D+C→M		MS+M+C→D	
	mAP	Rank-1	mAP	Rank-1
RaMoE [21]	56.5	82.0	56.9	73.6
SuA-SpML(ours)	**60.8**	**84.8**	53.9	72.7
SuA-SpML(ours, IBN)	**62.4**	**86.6**	54.8	73.4
Method	**M+D+C→MS**		**MS+M+D→C**	
	mAP	Rank-1	mAP	Rank-1
RaMoE [21]	13.5	34.1	35.5	36.6
SuA-SpML(ours)	**19.6**	**44.2**	34.8	35.7
SuA-SpML(ours, IBN)	**21.7**	**48.6**	**36.0**	**36.8**

- Market 1501高出5.9% mAP和4.6% rank-1
- MSMT 17高出8.2% mAP和14.5% rank-1
- DukeMTMC-ReID数据集上性能有所降低
- CUHK 03数据集上性能相当

虽然RaMoE在DukeMTMC-ReID和CUHK 03数据集上实现了与本文的方法性能相当，但本文的方法在Market 1501和MSMT 17数据集上的性能大大优于它，这表明本文的方法具有更好的泛化能力

单源实验

一个域进行训练，另一个域上进行测试

1. Market→Duke
2. Duke → Market
3. Market→MSMT
4. Duke → MSMT

Method	Backbone	Market→Duke		Duke→Market	
		mAP	Rank-1	mAP	Rank-1
ResNet-IBN [82]	ResNet50	24.3	43.7	23.5	50.7
OSNet[77]	OSNet	25.9	44.7	24.0	52.2
OSNet-IBN [77]	OSNet-IBN	27.6	47.9	27.4	57.8
OSNet-AIN [77]	OSNet-AIN	30.5	52.4	30.6	61.0
$QAConv_{50}$ [22]	ResNet50	28.7	48.8	27.2	58.6
SuA-SpML (ours)	ResNet50	**33.4**	**54.4**	**35.2**	**64.1**
SuA-SpML (ours)	ResNet50-IBN	**34.8**	**55.5**	**36.3**	**65.8**
Method	**Backbone**	Market→MSMT		Duke→MSMT	
		mAP	Rank-1	mAP	Rank-1
OSNet-AIN [77]	OSNet-AIN	8.2	23.5	10.2	30.3
$QAConv_{50}$ [22]	ResNet50	7.0	22.6	8.9	29.0
SuA-SpML (ours)	ResNet50	**9.1**	**26.0**	**12.0**	**33.8**
SuA-SpML (ours)	ResNet50-IBN	**11.0**	**30.1**	**13.6**	**37.8**

在MSMT上，本文的方法只使用Duke或Market进行训练，可以达到与使用多源数据集训练的其他方法相当的性能，表明此方法具有良好的应用能力。

消融实验

基于两个主干：ResNet 50和ResNet 50-IBN进行消融研究，以分析所提出方法中每个组件的有效性。

Method	MS+D+C→M		MS+M+C→D	
	mAP	Rank-1	mAP	Rank-1
ResNet50	44.2	72.8	46.2	65.5
+SuA	50.3	76.1	48.5	67.4
+SpML	49.2	75.4	48.2	67.1
+SuA+SpML	52.6	78.0	49.8	68.5
+SuA+L_{dg}	54.5	79.2	50.2	69.4
SuA-SpML(w/o ms)	55.4	79.9	50.5	69.8
SuA-SpML	56.6	81.0	51.2	70.8
ResNet50-IBN	47.6	74.0	47.5	66.4
+SuA	54.2	78.8	49.4	68.5
+SpML	53.1	77.4	48.8	68.1
+SuA+SpML	56.5	80.2	50.8	70.1
+SuA+L_{dg}	57.2	81.1	50.8	70.3
SuA-SpML(w/o ms)	57.9	82.0	51.2	70.7
SuA-SpML	59.1	83.2	51.9	71.6

1.风格增强SuA训练的模型mAP和rank-1均有显著改进，可以有效地提高模型的泛化能力

2.自定进度元学习 SpML的训练策略能够帮助模型学习不可见域的内容，进一步提高模型的泛化能力

3.距离图对齐损失L_{dg}可以进一步促进基于具有相同语义内容的增强域的域不变表示的学习

四、实验

为了进一步探索距离图对齐损失在方法中是如何工作的，进行了两个比较实验：1.有无对齐损失的域间图距离的比较 2.对齐策略的比较。

在没有对齐损失的情况下，原始数据和增强数据之间存在显著差异，并且这种差异随着SuA中扰动方差的增加而增加。

Target	Graph Dis(w/o L_{dg})		Graph Dis(with L_{dg})	
	Var(0.1)	Var(0.15)	Var(0.1)	Var(0.15)
Market1501	0.73	1.38	0.0104	0.0154
DukeMTMC	0.78	1.41	0.0105	0.0163
MSMT17	0.63	1.24	0.0094	0.0136
CUHK03	0.82	1.75	0.0106	0.0178

通过将原始数据与增强数据(O→A)，并将增强数据与原始数据对齐(A→O)

O → A比A→ O表现更好，但两者都劣于本文的方法，这是由于存在域偏置，其不能通过单向对齐来减轻。

Target	Original→Augmented		Augmented→Original	
	mAP	Rank-1	mAP	Rank-1
Market1501	45.6	74.1	38.9	66.8
DukeMTMC	38.2	57.3	27.3	45.4
MSMT17	12.4	32.5	7.3	20.6
CUHK03	24.5	24.9	15.1	14.5

四、实验

组件的效率分析

Method	Train Time (s)	Test Time(s)	MS+D+C→M	
			mAP	Rank-1
ResNet50	0.92	0.007	44.2	72.8
+SuA	1.84	0.007	50.3	76.1
+SuA+L_{dg}	2.34	0.007	54.5	79.2
SuA-SpML(w/o ms)	5.92	0.007	55.4	79.9
SuA-SpML	9.83	0.007	56.6	81.0

- 我们的SuA和L_{dg}实现了10.3%的mAP和6.4%的rank-1的显着改进。
- 在添加自定步Meta学习SpML后，尽管模型每次迭代需要的时间略多,但性能进一步提高到56.6% mAP和81.0% rank-1。
- DG ReID由于其实际应用而更加关注效率测试，可认为本文的模型没有额外的计算开销。

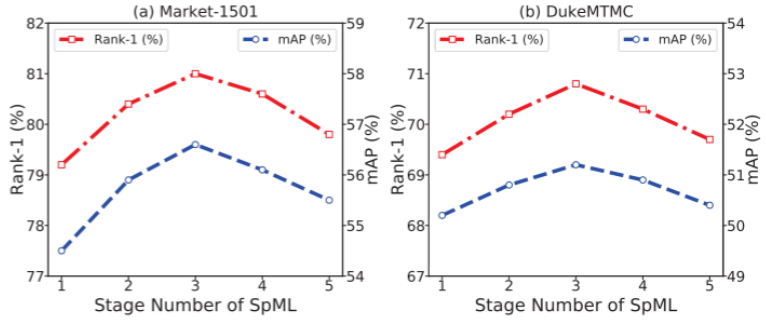
分析部署SuA位置

- 将SuA应用于较低级别层实现了比在较高级别层中更好的性能。
- 将SuA应用于res1和res2两者实现了最佳性能。
- 当将SuA应用于res4时，性能降低。

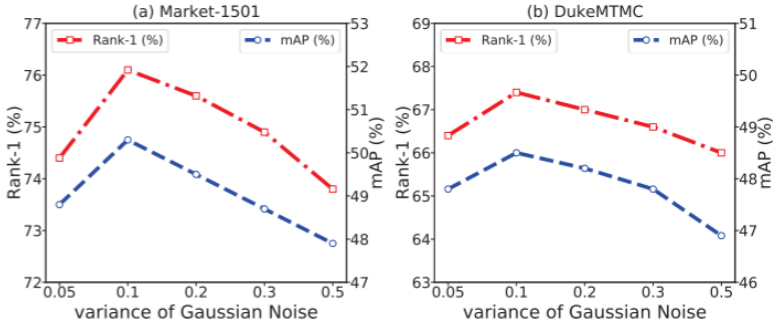
Model	MS+D+C→M		MS+M+C→D	
	mAP	Rank-1	mAP	Rank-1
ResNet50	44.2	72.8	46.2	65.5
SuA($res1$)	47.4	74.3	47.4	66.2
SuA($res2$)	48.2	74.6	47.8	66.5
SuA($res12$)	50.3	76.1	48.5	67.4
SuA($res3$)	44.6	71.6	45.8	65.3
SuA($res13$)	45.6	73.7	46.7	65.6
SuA($res123$)	47.4	74.2	47.4	66.3
SuA($res4$)	38.7	67.3	42.2	61.1
SuA($res14$)	42.4	71.5	45.3	64.4

四、实验

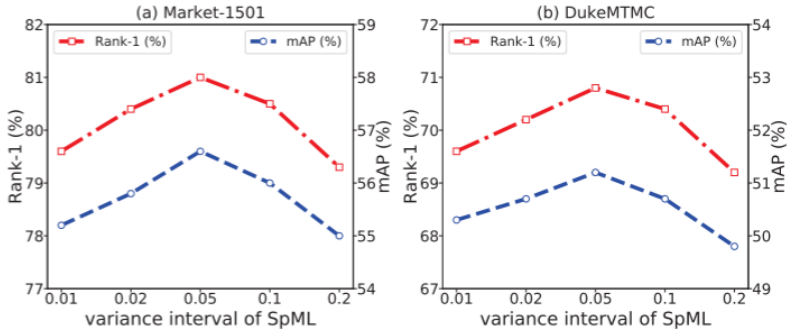
超参数分析



- 在SpML中使用不同学习阶段数n的性能。从这个结果中，我们可以看到，当n = 3时获得最佳性能，并且较大的n不会影响精度。



- SuA中高斯噪声的不同方差σ的性能。结果表明当σ值较小时，模型对源域的扩充不够，而当σ值较大时，噪声较大容易造成负面影响，模型也无法探索域不变表示， σ=0.1最合适。

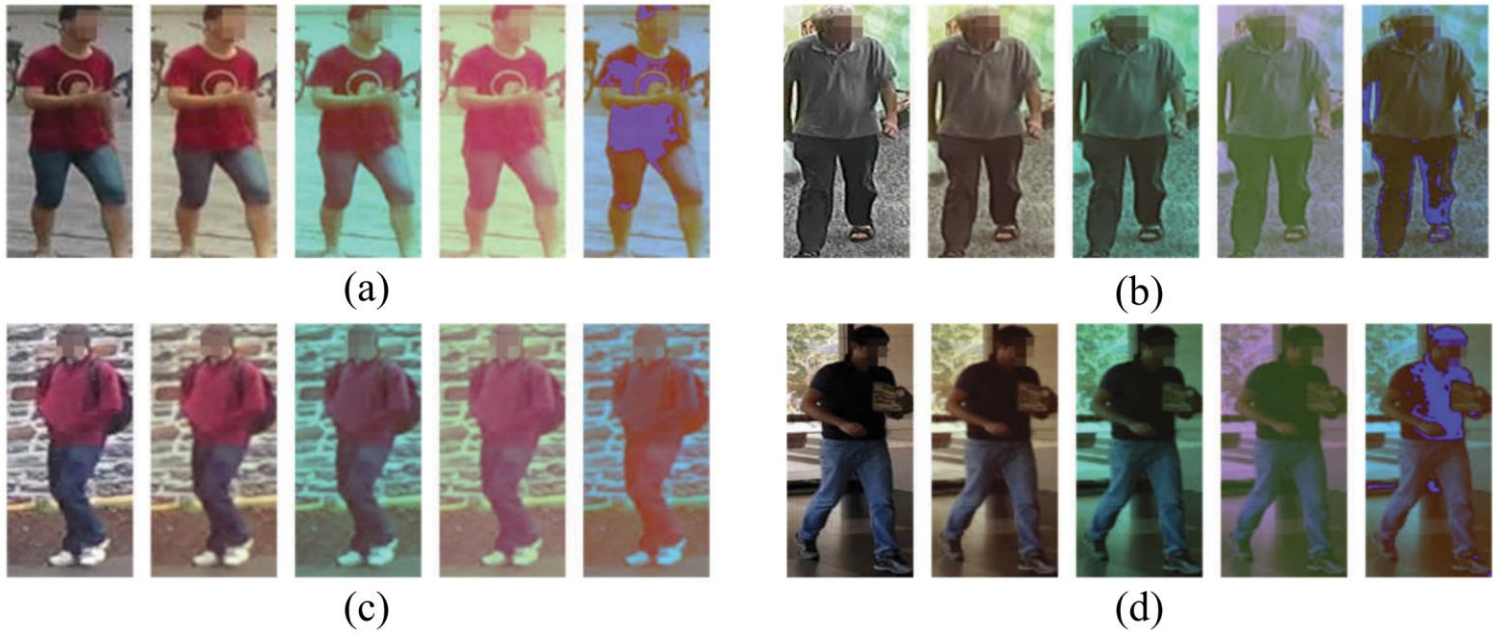


- SpML中不同σ间隔的影响。当σ的区间设置为0.05时获得最佳性能。

SuA可视化

从左到右示出了由SuA中的高斯噪声的方差控制的原始图像和四个方差小到大的图像

- SuA可以有效地使域多样化，同时保留图像的语义内容
- SuA不仅可以将不同身份的图像增强到相似的域，而且可以将相同身份的图像增强到不同的域，这有效地解决了开集DG问题



五、总结

本文提出了一种基于风格不确定性增强的自定步Meta学习（SuA-SpML）以解决DG-ReID的领域泛化问题。

1. 提出了一种新的特征级增强策略，风格不确定性增强(SuA)，通过在训练过程中用高斯噪声扰动实例风格来多样化源域。
2. 提出了一种新的自定步Meta学习方法，帮助模型从易到难地学习，逐步提高模型对未知目标域的泛化能力。
3. 提出了一个距离图对齐损失，对齐源域之间的特征关系分布，以方便网络探索域无关的表示。

谢谢大家，敬请指导！