Resource Aware Person Re-identification across Multiple Resolution

一. 论文概述

现行状况下，大多的方法都是利用一刀切式的高维特征对行人进行重识别。而这种方法一方面难以处理拍摄情况比较糟糕的图片，一方面对于比较理想的图片花费太多的计算资源。针对于这样的情况，作者在文中提出了一种采用深度监督训练的多重卷积层模型，它能够充分地融合嵌入层中提取出来的特征。

同时，作者也在文中提出了两种在计算资源有限的情况下完成行人重识别任务的方法。

二. 作者方法

对一个人的辨别既可以通过他的身形、服饰的颜色等整体特征，也可以通过他身上的细节部分，比如印花的形状、饰品等特征来确定他的身份。考虑到这样的实际情况，作者认为高层级特征对识别固然重要，但也不能因此就直接丢弃掉一些有用低级的特征。故，作者通过对不同层级特征的融合来使最终用于辨别的特征可以同时具有较高的分辨率和高度抽象的语义细节，以此来提升模型识别的准确率（较高的分辨率易于捕捉形状、颜色等一类的整体特征，高度抽象的语义细节则更多包含印花、饰品等一类的局部特征）。

基于上面的想法，作者提出了 DaRe 模型（Deep Anytime Re-ID）。与以往的模型对比，它将 a）每层的信息融合在一起，同时 b）引入中间损失函数来监督对不同层特征提取的训练，其具体结构如图 2.1 所示。

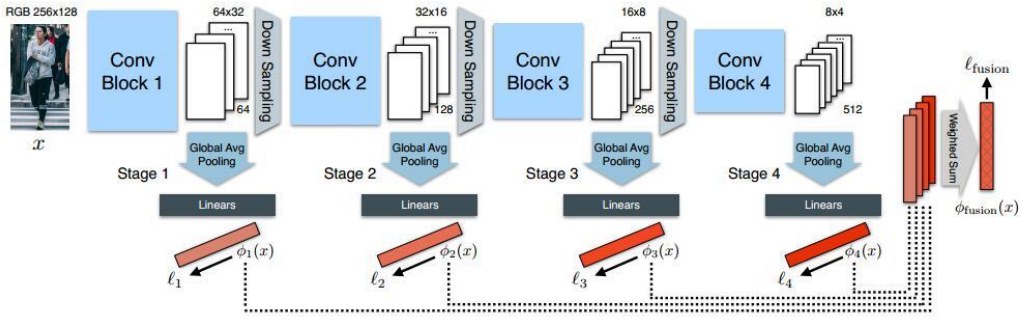

图 2.1　DaRe 模型结构

·网络结构

网络由 4 个阶段组成，基于 ResNet-50。每个阶段都包括一个对特征进行操作的卷积层，图片在经过它们之后分辨率都将下降一半。在一阶段结束时，特征都将经过下采样操作之后再送到下一阶段进行训练。

作者将每个阶段的特征单独提取出来经过平均池化和两层全连接之后形成一组新的嵌入特征。这里全连接层的作用仅是将所有的嵌入特征都统一到一个维度上。

记$\phi_s(x)$为图片x在阶段s的嵌入特征，那么最终融合后的嵌入特征就可以表示为：

$$\phi_{\text{fusion}}(x) = \sum_{s=1}^{4} w_s \phi_s(x)$$

其中，w_s 表示学习的权重。

·损失函数

作者在文中采用的损失函数如下形式：

$$\ell_{\text{all}} = \sum_{s=1}^{4} \ell_s + \ell_{\text{fusion}}$$

其中，ℓ_s 表示一阶段的损失函数，ℓ_{fusion} 表示最后融合嵌入特征时的损失函数。

更具体地来讲，作者用三胞胎损失函数来表示每一个损失函数，其表达形式如下：

$$\ell = \sum_{p=1}^{P}\sum_{k=1}^{K} \ln\Big(1+\exp\Big(\overbrace{\max_{a=1,\dots,K} D\left(\phi(x_p^k),\phi(x_p^a)\right)}^{\text{furthest positive}} - \underbrace{\min_{\substack{q=1,\dots,P\\ b=1,\dots,K\\ q\neq p}} D\left(\phi(x_p^k),\phi(x_q^b)\right)}_{\text{nearest negative}}\Big)\Big),$$

另外，作者还采用了随机擦除和重排序的的方法来提高 DaRe 的准确性。

下面我们讲一下 DaRe 是怎么实现在计算资源优先的情况下完成充实别任务的。

从前面知道每一阶段都会有一个单独的嵌入特征，DaRe 就利用在计算资源限制下能完成阶段的嵌入特征来进行行人重识别。

三. 实验结果

DaRe 和其他优秀方法在 5 组数据集上的比较如表格 3-1 所示：

Method	Dataset									
	Market		MARS		CUHK03(L)		CUHK03(D)		Duke	
	Rank-1	mAP	Rank-1	mAP	Rank-1	mAP	Rank-1	mAP	Rank-1	mAP
CNN+DCGAN(R) [65]	56.2	78.1	-	-	-	-	-	-	67.7	47.1
ST-RNN(C) [68]	-	-	70.6	50.7	-	-	-	-	-	-
MSCAN(C) [27]	80.3	57.5	71.8	56.1	-	-	-	-	-	-
PAN(R) [64]	82.2	63.3	-	-	36.9	35.0	36.3	34.0	71.6	51.5
SVDNet(R) [48]	82.3	62.1	-	-	40.9	37.8	41.5	37.2	76.7	56.8
TriNet(R) [17]	84.9	69.1	79.8	67.7	-	-	-	-	-	-
TriNet(R)+RE* [67]	-	-	-	-	64.3	59.8	61.8	57.6	-	-
SVDNet(R)+RE [67]	87.1	71.3	-	-	-	-	-	-	79.3	62.4
DaRe(R)	86.4	69.3	83.0	69.7	58.1	53.7	55.1	51.3	75.2	57.4
DaRe(R)+RE	88.5	74.2	82.6	71.7	64.5	60.2	61.6	58.1	79.1	63.0
DaRe(De)	86.0	69.9	84.2	72.1	56.4	52.2	54.3	50.1	74.5	56.3
DaRe(De)+RE	**89.0**	**76.0**	**85.5**	**74.0**	**66.1**	**61.6**	**63.3**	**59.0**	**80.2**	**64.5**
IDE(C)+ML+RR [66]	61.8	46.8	67.9	58.0	25.9	27.8	26.4	26.9	-	-
IDE(R)+ML+RR [66]	77.1	63.6	73.9	68.5	38.1	40.3	34.7	37.4	-	-
TriNet(R)+RR [17]	86.7	81.1	81.2	77.4	-	-	-	-	-	-
TriNet(R)+RE+RR* [67]	-	-	-	-	70.9	71.7	68.9	69.36	-	-
SVDNet(R)+RE+RR [67]	89.1	83.9	-	-	-	-	-	-	84.0	78.3
DaRe(R)+RR	88.3	82.0	83.0	79.3	66.0	66.7	62.8	63.6	80.4	74.5
DaRe(R)+RE+RR	90.8	85.9	83.9	80.6	72.9	73.7	69.8	71.2	**84.4**	79.6
DaRe(De)+RR	88.6	82.2	84.8	80.3	63.4	64.1	60.2	61.6	79.7	73.3
DaRe(De)+RE+RR	**90.9**	**86.7**	**85.1**	**81.9**	**73.8**	**74.7**	**70.6**	**71.6**	**84.4**	**80.0**

注：这篇文章中关于第二种在限制资源下完成重识别任务的方法没怎么理解，在这里不做深究，等我在研究研究，论文有点绕口。