Generalizable Person Re-identification by Domain-Invariant Mapping Network

Jifei Song[1] Yongxin Yang[2] Yi-Zhe Song[3] Tao Xiang[3,4] Timothy M. Hospedales[2,3]
[1]Queen Mary University of London [2]The University of Edinburgh
[3]SketchX, CVSSP, University of Surrey [4]Samsung AI Centre, Cambridge
j.song@qmul.ac.uk, {y.song, t.xiang}@surrey.ac.uk, {yongxin.yang, t.hospedales}@ed.ac.uk

Abstract

We aim to learn a domain generalizable person re-identification (ReID) model. When such a model is trained on a set of source domains (ReID datasets collected from different camera networks), it can be directly applied to any new unseen dataset for effective ReID without any model updating. Despite its practical value in real-world deployments, generalizable ReID has seldom been studied. In this work, a novel deep ReID model termed Domain-Invariant Mapping Network (DIMN) is proposed. DIMN is designed to learn a mapping between a person image and its identity classifier, i.e., it produces a classifier using a single shot. To make the model domain-invariant, we follow a meta-learning pipeline and sample a subset of source domain training tasks during each training episode. However, the model is significantly different from conventional meta-learning methods in that: (1) no model updating is required for the target domain, (2) different training tasks share a memory bank for maintaining both scalability and discrimination ability, and (3) it can be used to match an arbitrary number of identities in a target domain. Extensive experiments on a newly proposed large-scale ReID domain generalization benchmark show that our DIMN significantly outperforms alternative domain generalization or meta-learning methods.

1. Introduction

The problem of person re-identification (ReID) has been studied intensively. A ReID model is used to match people across non-overlapping camera views. Given image pairs of the same person across views, most recent ReID models apply deep convolutional neural networks (CNNs) to learn a feature embedding space where people can be matched based on feature distances [4, 6, 17, 19, 29, 42, 46, 51, 54, 59]. These models are trained and tested on the same dataset. In practice, however, if we consider each dataset as a domain, there are often domaiDIMNn gaps, because different datasets are often collected in very different visual scenes (*e.g.*, indoors/outdoors, shopping malls, traffic junctions and airports). Deep ReID models that are directly applied to new dataset/domain without model updating are known to suffer from considerable performance degradation [9, 30, 50, 52, 61], thus suggesting model overfitting and poor domain generalization.

In this paper, we aim to learn domain generalizable ReID models. Such a model is trained on a set of source domains/datasets, and should generalize to any new unseen dataset for effective ReID without any model updating. Such a model thus needs to solve a domain generalization problem with different class (person identity) label spaces for different datasets/domains. A domain generalizable ReID model has great value for real-world large-scale deployment. Specifically, when a customer purchases a ReID system for a specific camera network, the system is expected to work out-of-the-box, without the need to go through the tedious process of data collection, annotation and model updating/fine-tuning.

Surprisingly, there is very little prior study of this topic. Existing ReID works occasionally evaluate their models' cross-dataset generalization, but no specific design is made to make the models more generalizable. Recently, unsupervised domain adaptation (UDA) methods for ReID [9, 30, 50, 52, 61] have been studied to adapt a ReID model from source to target domain. However, UDA models update using unlabeled target domain data, so data collection and model update are still required. Beyond ReID, the problem of domain generalization (DG) has been investigated in deep learning, with some recent few-shot meta-learning approaches also adapted for DG. However, existing DG methods [20, 34, 23, 39] assume that the source and target domain have the same label space; whilst existing meta-learning models [49, 49, 10, 35, 40] assume a fixed number of classes for target domains and are trained specifically for that number using source data. They thus have limited efficacy for ReID, where target domains have a different and variable number of identities.

Our solution to generalizable ReID is based on a novel Domain-Invariant Mapping Network(DIMN). DIMN is de-

signed to learn a mapping between a person image and its identity classifier weight vector, *i.e*., it produces a classifier using a single shot. Once learned, for a target domain, each gallery image will be fed into the network to generate the weight vector of a specific linear classifier for the corresponding identity. A probe image will then be matched using the classifier by computing a simple dot product between the weight vector and a deep feature vector extracted from the probe. To make the model domain-invariant, we follow a meta-learning pipeline and sample a subset of source domain training tasks (identities) during each training episode. However, the model is significantly different from conventional meta-learning methods in that: (1) No model updating is required for the target domain. (2) Different training tasks share a memory bank which is updated with a running average strategy. This memory bank ensures that the model training is scalable to a large number of identities in the source domain, and importantly the learned DIMN becomes more discriminative. (3) Once trained, the model can be used to match an arbitrary number of identities in a target domain.

Our contributions are as follows: (i) For the first time, the domain generalization problem in person ReID is explicitly highlighted and also tackled by designing a ReID model that is tailor-made for coping with unknown target domains. (ii) A novel Domain-Invariant Mapping Network(DIMN) is proposed whose generalizability comes from its ability to map an image directly into an identity classifier. An effective meta-learning based training strategy is also formulated with a new memory bank module introduced for scalability and discriminativity. (iii) A large-scale ReID domain generalization benchmark is defined, using five existing ReID datasets as source domains and four others as target domains. Extensive experiments validate the generalizability of our DIMN and suggest that it is superior to the state-of-the-art domain generalization and meta-learning alternatives.

2. Related Work

Person Re-Identification Recent person ReID models are dominated by deep feature learning approaches [4, 6, 17, 19, 29, 42, 46, 51, 54, 59]. Since different datasets contain different person identities captured by cameras of different viewing conditions, it has been noted that these state-of-the-art ReID models often overfit to training datasets and generalize poorly when applied directly to a new dataset with fine-tuning [9, 30, 50, 52, 61]. This had led to a new research direction on unsupervised ReID based on unsupervised domain adaptation (UDA). A UDA model assumes that there exists an unlabelled training set from the target domain. Recent UDA based ReID models mainly exploited GAN [11] based image-synthesis [9, 61] or domain alignment [50, 30]. However, GAN based model training is unstable; domain alignment methods often rely on attribute annotation thus having limited applicability. In contrast, our DIMN is a domain generalization model that does not require any data from the target domain for updating. It is thus much more generally applicable.

Domain Generalization Our model tackles the Domain generalization (DG) [20, 34] problem. DICA [34] proposed to learn the domain-invariant features via a kernel-based optimization. Recently, Motiian *et al*. extended a supervised domain adaptation network to DG by explicitly imposing a semantic alignment loss on every unpaired data [33]. The idea of adversarial training for unseen domain data synthesis is exploited in CrossGrad [39], where pseudo training instances are generated by perturbations in the direction of the gradient of the domain classifier and category classifier respectively. As an early attempt to apply meta-learning techniques to DG, MLDG [23] proposed to align meta-train and meta-test gradients, using the same training schedule, *i.e*., task (re)sampling, as the meta-learning model MAML [10]. Though both are derived from MAML, Reptile [35] does not consider the expensive second-order gradients in episodic training. Note that our DG ReID problem is more challenging than the category-level recognition problems considered in existing DG studies. This is because the target classes/identities are different to the source ones, which means we have to deal with domain gap and disjoint label space simultaneously. We show that our DIMN is much more effective than a number of state-of-the-art DG baselines including [39, 23, 35] (see Sec. 4.2), due to its unique end-to-end image-to-classifier learning.

Meta-learning Learning to learn or meta-learning [45] is topical in the machine learning community, and one of its well-received applications is few-shot learning (FSL). FSL aims to recognize novel visual categories from limited labelled examples, where conventional fine-tuning is unlikely to work due to over-fitting. Matching network [49] used an attention mechanism to learn a more generalizable embedding space from labelled images, and can be viewed as a weighted nearest neighbour classifier while predicting unseen classes' images. Prototypical networks [41] proposed to learn a prototype for each class, where the classification is based on computing the distances to those prototypes. Instead of using the prototype to generate the linear classifier, PPA [40] learns to derive the classifier parameters from the averaged supporting activations. Apart from metric learning solutions [41, 49], another promising approach is learning to optimize. E.g., [37] reformulated stochastic gradient descent (SGD) optimizer in an LSTM-based meta-learner, by replacing the fixed updating rule (SGD) with the data-driven one (trainable LSTM). Model-agnostic meta-learning (MAML) [10] aims to learn a good initialization, where the model can be adapted to a new task quickly, *e.g*., through one or few SGD updating steps. Many meta-

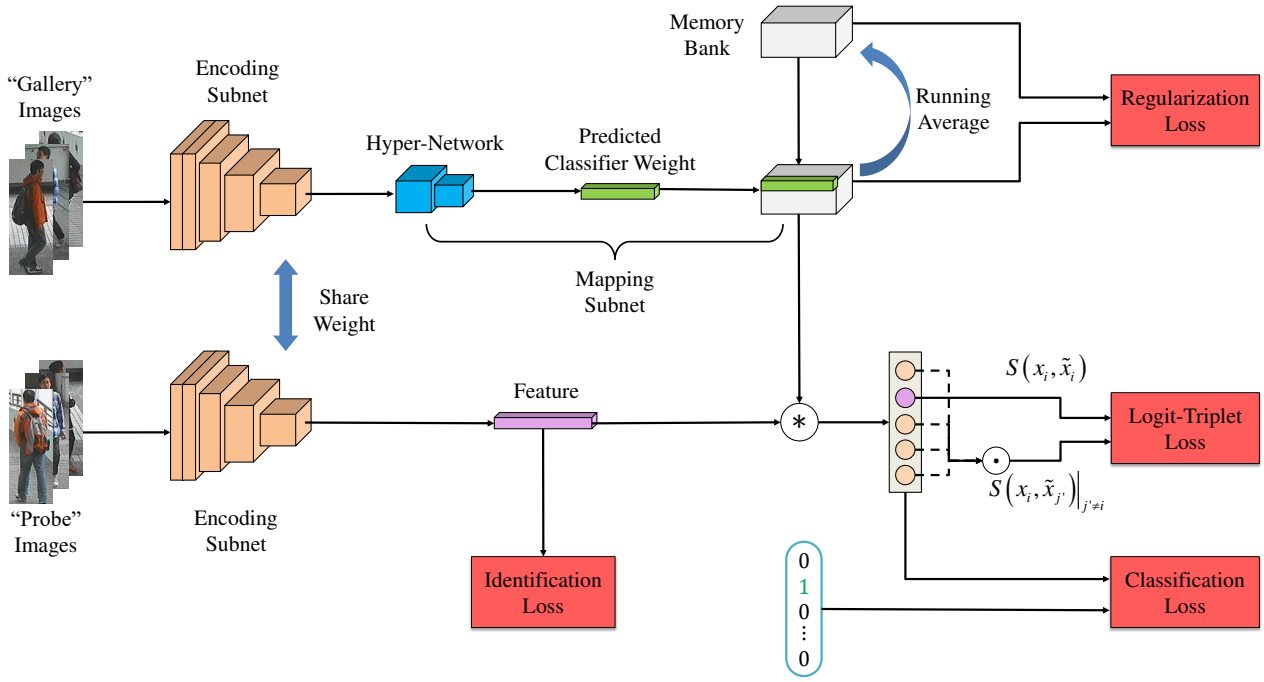

Figure 1. The proposed Domain-Invariant Mapping Network.

learning methods fix the number of classes during training and testing (typically 5-20), so they will have difficulties in scaling to ReID datasets with variable and much larger class numbers. Our DIMN is most closely related to the PPA model [40] in that both aim to predict classifier from a single image. However, our model learns an image-to-classifier mapping whilst PPA focuses on the much simpler task of feature-to-classifier mapping. Our experiments show that DIMN beats PPA by a large margin when applied to ReID (see Sec. 4.2).

3. Methodology

Overview We study a generalized person re-identification problem, where in the training stage, we have the access to M datasets (domains), $\mathcal{D}_1$, $\mathcal{D}_2$, ... and $\mathcal{D}_M$, and each domain has its own label space (person identities). The trained model will be deployed directly to a new domain/dataset, and is expected to work without any further model update. To this end, we propose a Domain-Invariant Mapping Network(DIMN), illustrated in Fig. 1. The training images are organized into gallery and probe sets to simulate the testing scenario where a probe image is compared against a gallery set for matching. The proposed network consists of three modules: (1) Two weight-tied base networks, the encoding subnets, which serve as feature extractors for gallery and probe images respectively. (2) A hyper-network [15], namely mapping subnet, which takes the gallery image embedding as input and tunes it into the classifier's weight vector that represents the identity of the gallery image identity. (3) A memory bank that stores all classifiers in training domains. We will detail the design of each module in the following sections.

Encoding Subnet For the encoding subnet, we use MobilenetV2 [38] – a lightweight CNN with competitive performance compared to heavier alternatives such as ResNet [16] and InceptionV3 [44]. We found it to be both more efficient and more effective for our large-scale DG ReID benchmark.

As shown in Fig. 1, the two Siamese encoding subnets in DIMN are used in the gallery and probe branches respectively. To generate the inputs for both branches, we follow a specific mini-batch sampling procedure. Assuming we have C unique identities in total in the aggregated M training domains, we sample C_b ($C_b \ll C$) identities randomly for each mini-batch. For each identity l_i, we further sample two images, of which we assign one as gallery $\tilde{x}_i$ and the other as probe x_i. Therefore, we have $2C_b$ image/label pairs in a mini-batch, as illustrated in Fig. 2.

Assuming the encoding subnet produces a D-dimensional feature vector, the first training objective for DIMN is an identification loss for the total C identities, but over one mini-batch of C_b identities, denoted as $\mathcal{L}_{\text{id}}$,

$$\mathcal{L}_{\text{id}} = \sum_{i=1}^{C_b} \text{Cross\_Entropy}(l_i, \text{Softmax}(f_\theta(g_\phi(x_i)))) \quad (1)$$

where x_i is the input image and l_i is the one-hot encod-

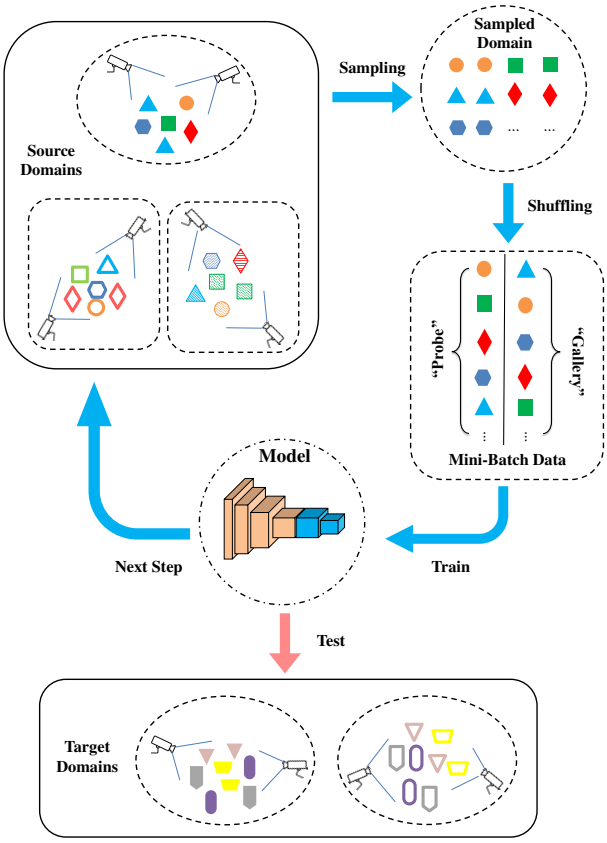

Figure 2. An illustration of the mini-batch sampling strategy.

ing of its label (a C-dimensional unit vector). $g_\phi(\cdot)$ is the encoding subnet parameterized by ϕ. $f_\theta(\cdot)$ is the classifier parameterized by θ where $\theta \in \mathbb{R}^{D\times C}$.

Mapping Subnet The deep feature vector, extracted from each gallery image using the encoding subnet, is then fed into a mapping subnet to compute a classifier weight vector for the corresponding identity. Formally, given an instance of the jth class from the gallery branch, denoted as $\tilde{x}_j$. Instead of learning the jth classifier weight vector $\theta_{\cdot,j}$ as part of the model parameters, as in a conventional classification CNN, we *generate* it as a layer of the network using $\tilde{x}_j$ as input. We thus have:

$$\hat{\theta}_{\cdot,j} = h_\omega(g_\phi(\tilde{x}_j)), \tag{2}$$

where the mapping subnet $h_\omega(\cdot)$ can be understood as a hyper-network [15] since it generates the parameters for another neural network (the probe branch). Here we simply apply a multi-layer perceptron (MLP) as the basic architecture of our mapping subnet. Note that we omit the bias term in the weight generation for simplicity.

Given a gallery image $\tilde{x}_j$, and a probe image x_i, the mapping subnet generates an identity classifier weight vector $\hat{\theta}_{\cdot,j}$ based on the gallery image, $\tilde{x}_j$. We then take the dot product of the generated classifier weight vector $\hat{\theta}_{\cdot,j}$ and the probe image feature $g_\phi(x_i)$, to produce a logit vector p whose elements corresponding the identity of $\tilde{x}_j$: $p_j = h_\omega(g_\phi(\tilde{x}_j)) \cdot g_\phi(x_i)$. Passing the vector p into a softmax layer then gives us the predicted probability of how likely the input identity x_i in the probe branch is matched with the identity $\tilde{x}_j$ in the gallery branch. The ground truth label y for the matching network will be 1 if x_j matches with $\tilde{x}_j$, and 0 otherwise. y can then be used for computing a classification loss.

Note that the logit vector p is a C-dimensional vector which can be of very high-dimensionality with a large number of identities in the source domains. If we follow the standard meta-learning practice and reduce the dimensionality to the much smaller number C_b, the model training becomes tractable. However, we then lose the discriminative power: the mapping network is trained to perform a much easier task of classifying C_b people rather than C. To have the better of both worlds, we introduce memory bank to keep both scalability and discriminativity.

Memory Bank The memory bank is realized by a weight matrix $W \in \mathbb{R}^{D\times C}$. In one mini-batch, we feed C_b samples (one sample in each of C_b classes), denoted as $[\tilde{x}_1, \tilde{x}_2, \ldots, \tilde{x}_{C_b}]$, to the gallery branch, after the encoding subnet g_ϕ and the mapping subnet h_ω, we have C_b predicted weight vectors, $\{\hat{\theta}_{\cdot,j}, j = [1, 2, \ldots, C_b]\}$, stacked as a $D \times C_b$ matrix $\hat{\theta}$.

Since we know the identity of each of those C_b samples, we can locate its corresponding position in W and replace that column with the predicted weight vector. First, we make a full copy of W as $\hat{W} \leftarrow W$, and then carry out the replacement by,

$$\hat{W}_{\cdot,L(j)} \leftarrow \hat{\theta}_{\cdot,j} \quad \forall j \in [1, 2, \ldots, C_b], \tag{3}$$

where $L(j)$ is the function that retrieves the index of the jth sample's class label in the full label space. After the replacing operations, we define a new classification loss involving $\hat{W}$ for the matching network,

$$\mathcal{L}_{\text{mat}} = \sum_{i=1}^{C_b} \text{Cross\_Entropy}(y_i, \text{Softmax}(\hat{W}^T g_\phi(x_i))) \tag{4}$$

where $[x_1, x_2, \ldots, x_{C_b}]$ are probe branch inputs. Since a part of $\hat{W}$ (*i.e.*, those C_b columns that have been replaced) is parameterized by ϕ and ω, minimizing Eq. 4 have an impact on these trainable variables.

We call the replacements $\hat{W}$ as *online* memory, and the original memory bank W as *target* memory. In each iteration, we update the target memory by running average [14],

$$W \leftarrow (1 - \alpha)W + \alpha\hat{W} \tag{5}$$

It can be seen that, only for those identities/classes that appear in the mini-batch, their memory will be updated.

We find that two design choices further help stabilize the training: (i) We do column-wise ℓ_2 normalization on W, *i.e.*, the generated classifiers are projected onto a unit hypersphere. (ii) We use W as a prior for the predicted classifiers, *i.e.*, the predicted classifier from an instance should not be far away from its cached version in the memory bank, as the latter should be more reliable. To realize (ii), we simply add a regularization term,

$$\mathcal{L}_{\text{reg}} = \sum_{j=1}^{C_b} \|W_{\cdot,L(j)} - \hat{\theta}_{\cdot,j}\|_2^2 \tag{6}$$

The regularization term, $\mathcal{L}_{\text{reg}}$, can also help the subnet generate a more stable prediction for each class. In addition, $\mathcal{L}_{\text{reg}}$ implicitly lets both the online memory and target memory converge to the same vectors.

Training Objective We further introduce a specific triplet loss built on our matching network, named as logit-triplet loss. As a by-product of building the mapping subset, for every instance in the probe branch, x_i, we can find its only positive pair $\tilde{x}_i$ in the gallery and compute the logit: $p = h_\omega(g_\phi(\tilde{x}_i)) \cdot g_\phi(x_i)$, meanwhile, we can also find negative pairs by computing: $n = h_\omega(g_\phi(\tilde{x}_j)) \cdot g_\phi(x_i)|_{j \neq i}$ among all the gallery identities. To further increase the negative pairs, we also include the cached 'identities' in the memory bank, thus negative pairs can be rewritten as $n = \hat{W}_{\cdot,j'} \cdot g_\phi(x_i)|_{j' \neq i},\ j' = [1, 2, ..., C]$. Both p and n will be further normalized to produce valid probabilities as the result of applying softmax function in Eq. 4. Denote the normalized p and n as $S(x_i, \tilde{x}_i)$ and $S(x_i, \tilde{x}_{j'})|_{j' \neq i}$, respectively, which also means the similarity score or matching probability between the probe and gallery pairs. We can then adopt the following logit-triplet loss with the hard mining [17],

$$\mathcal{L}_{\text{tri}} = \sum_{i=1}^{C_b} \max\left(0, \Delta + \max_{j' \neq i} S(x_i, \tilde{x}_{j'}) - S(x_i, \tilde{x}_i)\right) \tag{7}$$

The model is trained in an end-to-end fashion and the full training objective $\mathcal{L}_{\text{full}}$ is a weighted sum of Eq. 1, Eq. 4, Eq. 6, and Eq. 7.

$$\mathcal{L}_{\text{full}} = \mathcal{L}_{\text{id}} + \lambda_1 \mathcal{L}_{\text{mat}} + \lambda_2 \mathcal{L}_{\text{reg}} + \lambda_3 \mathcal{L}_{\text{tri}} \tag{8}$$

The training pipeline is summarized in Alg. 1.

Model Testing Trained in a meta-learning pipeline by sampling domains in each episode, both the encoding subnet ($g_\phi(\cdot)$) and mapping subnet ($h_\omega(\cdot)$) in our DIMN are supposed to be domain invariant. During the testing stage, given a query image x_i, and a gallery image $\tilde{x}_j$, we directly take the logits (or probability after the softmax layer) $h_\omega(g_\phi(\tilde{x}_j)) \cdot g_\phi(x_i)$ as the ranking score. It is importantly to point out that: (1) Although it looks like a one-shot learning method, DIMN is a DG method as the model itself (*i.e.*, $g_\phi(\cdot)$ and $h_\omega(\cdot)$) is fixed once trained on source domain. (2) Conventional deep ReID models only have a encoding network $g_\phi(\cdot)$, and uses the Euclidean distance between $g_\phi(x_i)$ and $g_\phi(\tilde{x}_j)$ as the ranking score. Comparing to them, our DIMN has very similar inference cost during the testing stage.

Algorithm 1 Training Domain-Invariant Mapping Network

Input: $\mathcal{D}_1$, $\mathcal{D}_2$, ... and $\mathcal{D}_M$;

for $t = 1$ to Max_Iter **do**
  Sample a domain $\mathcal{D}_l \in \{\mathcal{D}_1, \mathcal{D}_2, ..., \mathcal{D}_M\}$
  Sample $\{(x_1, \tilde{x}_1, y_1), \ldots, (x_{C_b}, \tilde{x}_{C_b}, y_{C_b})\} \in \mathcal{D}_l$
  $\hat{\theta} \leftarrow h_\omega(g_\phi(\tilde{x}))$
  Construct online memory $\hat{W}$ using Eq. 3
  Calculate losses: $\mathcal{L}_{\text{id}}$, $\mathcal{L}_{\text{mat}}$, $\mathcal{L}_{\text{reg}}$, and $\mathcal{L}_{\text{tri}}$
  Optimize $\mathcal{L}_{\text{full}}$ via the optimizer
  Update target memory using Eq. 5
end for

Domain	Dataset	# Train IDs		# Train images	
Source	CUHK02	1,816		7,264	
	CUHK03	1,467		14,097	
	Duke	1,812		36,411	
	Market1501	1,501		29,419	
	PersonSearch	11,934		34,574	
Domain	Dataset	# Test IDs		# Test images	
		#Pr. IDs	#Ga. IDs	#Pr. imgs	#Ga. imgs
Target	VIPeR	316	316	316	316
	PRID	100	649	100	649
	GRID	125	900	125	1,025
	i-LIDS	60	60	60	60

Table 1. Dataset statistics. "Pr.": Probe; "Ga.":Gallery.

4. Experiments

4.1. Dataset and Settings

A Large-Scale ReID benchmark We introduce a large-scale ReID benchmark to evaluate the domain generalization ability of a ReID model. We aim to simulate a real-world scenario where a ReID model is likely to be trained with all the public ReID datasets, in the hope that it can generalize well to an unseen domain. To this end, we deliberately use existing large-scale ReID datasets to form the source domains, and the smaller ones as target domains. More specifically, the source datasets include CUHK02 [26], CUHK03 [27], Market-1501 [57], DukeMTMC-ReID [60], and CUHK-SYSU PersonSearch [53]. All the images in these datasets, regardless of their original train/test splits, are used for model training. The test datasets/domains are VIPeR [13], PRID [18], GRID [31], and i-LIDS [58]. We evaluate the model performance on the standard testing split only, so our results are

directly comparable to prior reported results. The dataset details are listed in Table 1. Note that the total number of training identities is $C = 18,530$ with $121,765$ training images, much bigger than the size of each dataset alone.

Evaluation Protocols We follow the standard evaluation protocols on the testing datasets. For VIPeR, results on 5 random train/test splits, plus their swapped version are averaged, as mentioned in [12]. On PRID, in each trial, 100 randomly selected identities in view A are used as the probe set, while the corresponding 100 identities along with 549 unique identities in view B are used as the gallery. The average result of 10 trials is reported. On GRID, we follow the standard testing split recommended in [31]. On i-LIDS, we first split the dataset into training and test sets. In the testing split with r identities, we randomly select one image per identity for r identities as probe images, and randomly one of the remaining images from the corresponding identities to form the gallery set. Similar to the previous works, we do half split on the iLIDS dataset, yielding a testing split with $r = 60$ in our experiment, and we repeat the testing procedure 10 times as well.

Implementation Details We use MobileNetV2 [38] as the encoding subnet, with width multiplier of 1.4. The output feature dimension is thus $1,792$. Our mapping subnet is composed of a single fully-connected (FC) layer. The output size is set to the same as the input size, as the dimension of the classifier weights should be the same as the feature dimension. The running average parameter $\alpha = 0.5$ (Eq. 5). The logit-triplet loss margin (Eq. 7) is set to $\Delta = 0.8$. The weights for the classification loss and logit-triplet loss are set as equal, *i.e.*, $\lambda_1 = \lambda_3 = 1$, with regularization loss weight $\lambda_2 = 0.01$ (Eq. 8). We implement our model in Tensorflow [1] and train it with a single Titan X GPU. The model is trained for a fixed $180,000$ iterations with batch size 64, which means in each iteration, we sample 32 person identities (C_b); each comes with 2 images, of which 32 images are used as "probe" while the remaining 32 images are used as "gallery". Exponential decay learning rate scheduling is used with initial rate 0.00035 and ending with 0.0001. Adam optimizer [21] is used for all experiments.

Evaluation Metrics Two commonly used evaluation metrics are used. The first is cumulative matching characteristics (CMC). We report the CMC at rank-k, where $k = 1, 5, 10$, representing the ranking accuracy of the target identities in the top k results. The second metric is the mean average precision (mAP), which reflects the overall ranking quality rather than looking at top k positions only.

4.2. Comparisons against state-of-the-art

Baselines We compare with a variety of baselines, including the domain aggregation baseline, meta-learning baselines PPA [40] and Reptile [35], and two domain generalization methods, MLDG [23] and CrossGrad [39]. In the domain aggregation baseline, we assume there exists a universal model θ_* which is effective for all domains. There are two versions: Agg_MobileV2 uses the same MobileNetV2 backbone for fair comparison to our DIMN; Agg_PCB uses the PCB model in [43], which has a ResNet50 backbone and achieved the best single dataset performance on Market-1501 and DukeMTMC-ReID so far; Agg_Align uses the AlignReID model in [56] also based on ResNet50, suggesting to enhance the discriminitive ability by take advantage of the global branch and local branch. This domain aggregation baseline has been proven to be a very strong baseline in DG [22], especially given our big source domain size. Two meta-learning methods, PPA [40] and Reptile [35], are effective for alleviating over-fitting in few-shot learning-to-learn, and both can be adapted for the DG ReID problem here (unlike most others that require model updating). Existing Domain generalization baselines are the most relevant competitors, and we include two state-of-the-art methods in the comparison, namely MLDG [23] and CrossGrad [39]. We use "M" to denote the methods coming from the meta-

VIPeR Dataset	Type	rank-1	rank-5	rank-10	mAP
Agg_MobileV2	D	42.88%	61.33%	68.86%	51.91%
Agg_PCB [43]	D	38.10%	53.20%	59.30%	45.38%
Agg_Align [56]	D	42.78%	63.67%	73.58%	52.94%
PPA [40]	M	45.06%	65.09%	72.66%	54.46%
Reptile [35]	M	22.06%	39.43%	49.21%	31.33%
MLDG [23]	D	23.51%	43.80%	52.47%	33.52%
CrossGrad [39]	D	20.89%	39.05%	49.72%	30.40%
Ours	D	**51.23%**	**70.19%**	**75.98%**	**60.12%**
PRID Dataset	Type	rank-1	rank-5	rank-10	mAP
Agg_MobileV2	D	38.90%	63.50%	75.00%	50.98%
Agg_PCB [43]	D	21.50%	42.60%	49.70%	32.04%
Agg_Align [56]	D	17.20%	33.40%	39.60%	25.50%
PPA [40]	M	31.90%	61.10%	70.50%	45.26%
Reptile [35]	M	17.90%	33.80%	44.10%	26.90%
MLDG [23]	D	24.00%	48.00%	53.60%	35.36%
CrossGrad [39]	D	18.80%	35.30%	46.00%	28.18%
Ours	D	**39.20%**	**67.00%**	**76.70%**	**51.95%**
GRID Dataset	Type	rank-1	rank-5	rank-10	mAP
Agg_MobileV2	D	29.68%	51.12%	60.24%	39.79%
Agg_PCB [43]	D	**36.00%**	**53.68%**	63.28%	**44.66%**
Agg_Align [56]	D	15.92%	33.52%	41.44%	24.67%
PPA [40]	M	26.88%	50.48%	61.52%	37.98%
Reptile [35]	M	16.24%	29.44%	38.40%	23.02%
MLDG [23]	D	15.76%	31.12%	39.76%	23.57%
CrossGrad [39]	D	8.96%	22.08%	30.08%	16.00%
Ours	D	29.28%	53.28%	**65.84%**	41.09%
iLIDS Dataset	Type	rank-1	rank-5	rank-10	mAP
Agg_MobileV2	D	69.17%	84.17%	88.83%	75.95%
Agg_PCB [43]	D	66.67%	81.67%	86.83%	73.92%
Agg_Align [56]	D	63.83%	89.17%	**95.50%**	74.69%
PPA [40]	M	64.50%	83.75%	88.00%	72.73%
Reptile [35]	M	56.00%	80.67%	89.83%	67.11%
MLDG [23]	D	53.83%	78.67%	88.00%	65.18%
CrossGrad [39]	D	49.67%	74.17%	83.83%	61.29%
Ours	D	**70.17%**	**89.67%**	94.50%	**78.39%**

Table 2. Comparison against state-of-the-art methods.

learning community, while "D" indicates that the method is of a domain-generalization type.

Results We compare the proposed method with five baselines on the four target ReID datasets, VIPeR, PRID, GRID, and i-LIDS. ReID performance is listed in Table 2. The following observations can be made: (1): Overall, our method achieves the best result on all four target datasets among all compared methods. (2) The Aggregation baseline indeed is very strong, given a large and diverse set of source domains. Comparing the two versions with different backbone network, the lighter MobileNetV2 are clearly better over the state-of-the-art ReID models PCB [43] and AlignReID [56], except on GRID. Nevertheless, our DIMN consistently beats this strong baseline with the only exception of GRID. (3) PPA [40] is related to our method in that it also predicts classifier weight vector using a mapping network. However, the mapping network takes as input the feature output of an independently trained encoding network; this two-stage training strategy thus leads to sub-optimal solutions. Table 2 shows that with end-to-end training enabled by the hyper-network and memory bank modules introduced in our model, DIMN outperforms PPA significantly. (4) CrossGrad [39] requires a domain label (in our case, the camera ID) and assumes that the source domains are controlled by a latent domain descriptor that spans the whole spectrum of all possible domains. This assumption is clearly invalid in the ReID case, resulting in very poor performance. (5) Both Reptile [35] and MLDG [23] are variants of a classic meta-learning model MAML [10]. The results show that they all fail completely because they were originally designed for category-level recognition problems and unable to cope with both domain and identity changes.

Comparison against supervised/unsupervised baselines We also compare with the supervised and unsupervised state-of-the-art baselines published in the recent three years. The supervised and unsupervised settings are indicated by "S" and "U" respectively, whilst our domain generalization setting is "D". Note that both supervised and unsupervised settings require the use of the training splits of the target dataset, whilst our model does not, putting it at a big disadvantage. In addition, most compared methods also use a source domain for training and transfer. The results on the four datasets are shown in Tables 3, 4, 5, and 6 respectively. It is clear that our model, *despite not using any target domain data*, outperforms most of the competitors. Even when it is beaten by a supervised baseline, the margin is small. It is also noted that our DIMN significantly outperforms the state-of-the-art unsupervised ReID model [28] which needs additional attribute annotation in the source domain, as well as the unlabelled target domain training split. These results have great relevance when one builds a real-world ReID system: Our model is clearly the first choice because it is almost as good as the best supervised models but can be used out-of-the-box for any unseen domain.

Method	Type	rank-1	rank-5	rank-10	mAP
GatedSia [47]	S	37.80%	66.90%	77.40%	–
DeepRank [5]	S	38.37%	69.22%	81.33%	–
NullReid [55]	S	42.28%	71.46%	82.94%	–
SiaLSTM [48]	S	42.40%	68.70%	79.40%	47.90%
Ensembles [36]	S	45.90%	77.50%	88.90%	–
ImpTrpLoss [8]	S	47.80%	74.40%	84.80%	–
GOG [32]	S	49.70%	**79.70%**	88.70%	–
MTDnet [7]	S	47.47%	73.10%	82.59%	–
OneShot [3]	S	34.30%	–	–	–
SSM [2]	S	**53.73%**	–	**91.49%**	–
SSPR [24]	S	26.50%	50.51%	62.18%	–
JLML [28]	S	50.20%	74.20%	84.30%	–
TJAIDL [50]	U	38.50%	–	–	–
Ours	D	51.23%	70.19%	75.98%	**60.12%**

Table 3. Comparative results against baselines on VIPeR dataset. '-' indicates result not reported.

Method	Type	rank-1	rank-5	rank-10	mAP
NullReid [55]	S	29.80%	52.90%	66.00%	–
Ensembles [36]	S	17.90%	40.00%	50.00%	–
ImpTrpLoss [8]	S	22.00%	–	47.00%	–
MTDnet [7]	S	32.00%	51.00%	62.00%	–
OneShot [3]	S	**41.40%**	–	–	–
TJAIDL [50]	U	34.80%	–	–	–
Ours	D	39.20%	**67.00%**	**76.70%**	**51.95%**

Table 4. Comparative results against baselines on PRID dataset

Method	Type	rank-1	rank-5	rank-10	mAP
GOG [32]	S	24.70%	47.00%	58.40%	–
SSM [2]	S	27.20%	–	61.20%	–
JLML [28]	S	**37.50%**	**61.40%**	**69.40%**	–
Ours	D	29.28%	53.28%	65.84%	**41.09%**

Table 5. Comparative results against baselines on GRID dataset

Method	Type	rank-1	rank-5	rank-10	mAP
Ensembles [36]	S	50.34%	72.00%	82.50%	–
ImpTrpLoss [8]	S	60.40%	82.70	90.70%	–
MTDnet [7]	S	58.38%	80.35%	87.28%	–
OneShot [3]	S	51.20%	–	–	–
DSPSL [25]	S	55.17%	82.00%	90.67%	–
Ours	D	**70.17%**	**89.67%**	**94.50%**	**78.39%**

Table 6. Comparative results against baselines on i-LIDS dataset

4.3. Ablation Study

There are three important components in the proposed DIMN: the memory bank representing the global "identity" information, the running average updating strategy and the specifically designed logit-triplet loss built on the logit vector. To evaluate the contribution of each component, we compare our full model with three stripped-down versions, each of which is obtained by removing one component. Note that by removing the memory bank, we will also lose the running average updating strategy as the latter is built on top of the memory bank mechanism. In addition,

VIPeR Dataset	rank-1	rank-5	rank-10	mAP
w/o Memory bank	45.44%	67.44%	73.73%	55.49%
w/o Running average	50.03%	69.40%	74.40%	59.04%
w/o Logit-triplet	49.53%	68.29%	74.59%	58.29%
Ours-full	**51.23%**	**70.19%**	**75.98%**	**60.12%**
PRID Dataset	rank-1	rank-5	rank-10	mAP
w/o Memory bank	37.10%	58.20%	72.60%	48.27%
w/o Running average	36.50%	58.20%	67.20%	46.70%
w/o Logit-triplet	37.90%	63.60%	72.10%	49.75%
Ours-full	**39.20%**	**67.00%**	**76.70%**	**51.95%**
GRID Dataset	rank-1	rank-5	rank-10	mAP
w/o Memory bank	30.08%	51.68%	60.64%	40.50%
w/o Running average	30.48%	53.20%	**67.12%**	41.49%
w/o Logit-triplet	**32.88%**	53.28%	63.52%	**42.75%**
Ours-full	29.28%	**53.28%**	65.84%	41.09%
iLIDS Dataset	rank-1	rank-5	rank-10	mAP
w/o Memory bank	69.00%	87.17%	94.33%	77.05%
w/o Running average	67.67%	**90.00%**	94.00%	77.15%
w/o Logit-triplet	65.50%	84.50%	92.50%	74.40%
Ours-full	**70.17%**	89.67	**94.50%**	**78.39%**

Table 7. Contributions of different components

without running average updating strategy means that we will use the weight prediction from the hyper-network to directly rewrite the memory. Table 7 shows that each component contributes the ReID performance. Among all the components, the memory bank seems to be the most critical one. Without it, our DIMN is down-graded to a conventional meta-learning method that sacrifices discriminativity in exchange for scalability.

4.4. Qualitative Result

Some qualitative results are shown in Fig. 3. In this figure, the left column represents the probe images randomly sampled from the four testing datasets, while the remaining person images are the retrieved result using DIMN. From Fig. 3, it is clear to see that our method is able to distinguish the correct match from many impostors with similar appearances.

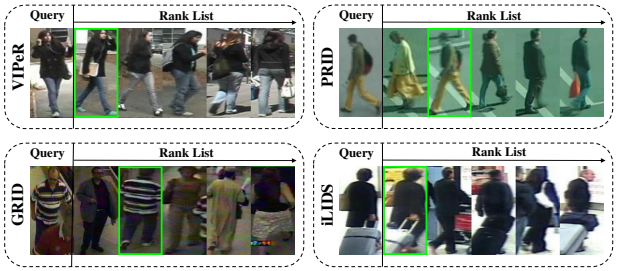

Figure 3. Retrieved result visualization on the four testing datasets.

4.5. Few-shot Learning on MiniImageNet

Although our DIMN is designed specifically for domain-generalizable ReID, its meta-learning pipeline of sampling tasks and performing one-shot classification on each sampled task makes it suitable for generic one-shot recognition tasks. To demonstrate its applicability to other learning-to-learn problems, we repurpose DIMN for the popular 5-way 1-shot/5-shot MiniImageNet benchmark, used by most previous meta-learning works. Originally proposed in [49], MiniImageNet is a subset of the ImageNet ILSVRC-12 dataset. It contains 100 classes split into 64/16/20 for train/validation/test. Each class has 600 examples. We follow the same training split as mentioned in [49]. We adopt the same basenet (SimpleNet) with four convoluational blocks as in [49] and [40]. All compared methods use exactly the same basenet and data split for a fair comparison. Following the standard protocol, We evaluate our method and calculate the average 5-way 1-shot/5-shot accuracy with the 95% confidence interval of 1000 testing rounds. The performance of our methods against other baselines is summarized in Table 8. The results demonstrate that our method is fairly competitive even for category-level recognition.

Mini-ImageNet	1-shot	5-shot
Matching Network [49]	43.56±0.84%	55.31±0.73%
Meta-Learner LSTM [37]	43.44±0.77%	60.00±0.71%
Reptile [35]	47.07±0.26%	62.74±0.37%
MAML [10]	48.70±1.84%	63.11±0.92%
PPA [40]	49.64±0.58%	61.94±0.44%
Ours	**50.74**±0.54%	**63.13**±0.42%

Table 8. Few-shot learning results on MiniImageNet

5. Conclusion

A domain-generalizable person re-identification (ReID) approach was proposed to enable a ReID model to be deployed out-of-the-box for any new camera network domain. Specifically, a novel deep ReID model termed Domain-Invariant Mapping Network (DIMN) was introduced. It has an encoding subnet to extract features from input images and a mapping subnet that predicts a classifier weight vector from a single input image. The two subnets are trained end-to-end by using the mapping subnet as a hyper-network. The training follows a meta-learning pipeline to make the model domain invariant and generalizable to unseen domains. Thanks to a memory bank module, the training is scalable without sacrificing model discriminativity. Extensive experiments on a newly defined large-scale benchmark validated the effectiveness of our DIMN. The experiments also showed that domain generalization in Re-ID is a very hard problem and many existing domain generalization and meta-learning methods failed to beat the strong but naive domain aggregation baseline. However, given our promising results, and the practical value of a domain-agnostic Re-ID system, this is an important avenue for future work.

References

[1] Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, et al. Tensorflow: a system for large-scale machine learning. In *OSDI*, 2016.

[2] Song Bai, Xiang Bai, and Qi Tian. Scalable person re-identification on supervised smoothed manifold. In *CVPR*, 2017.

[3] Slawomir Bak and Peter Carr. One-shot metric learning for person re-identification. In *CVPR*, 2017.

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

[5] Shi-Zhe Chen, Chun-Chao Guo, and Jian-Huang Lai. Deep ranking for person re-identification via joint representation learning. *TIP*, 2016.

[6] Weihua Chen, Xiaotang Chen, Jianguo Zhang, and Kaiqi Huang. Beyond triplet loss: A deep quadruplet network for person re-identification. In *CVPR*, 2017.

[7] Weihua Chen, Xiaotang Chen, Jianguo Zhang, and Kaiqi Huang. A multi-task deep network for person re-identification. In *AAAI*, 2017.

[8] De Cheng, Yihong Gong, Sanping Zhou, Jinjun Wang, and Nanning Zheng. Person re-identification by multi-channel parts-based cnn with improved triplet loss function. In *CVPR*, 2016.

[9] Weijian Deng, Liang Zheng, Guoliang Kang, Yi Yang, Qixiang Ye, and Jianbin Jiao. Image-image domain adaptation with preserved self-similarity and domain-dissimilarity for person reidentification. In *CVPR*, 2018.

[10] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. *arXiv preprint arXiv:1703.03400*, 2017.

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

[12] Douglas Gray, Shane Brennan, and Hai Tao. Evaluating appearance models for recognition, reacquisition, and tracking. In *PETS*, 2007.

[13] Douglas Gray and Hai Tao. Viewpoint invariant pedestrian recognition with an ensemble of localized features. In *ECCV*, 2008.

[14] Shixiang Gu, Timothy P. Lillicrap, Ilya Sutskever, and Sergey Levine. Continuous deep q-learning with model-based acceleration. In *ICML*, 2016.

[15] David Ha, Andrew Dai, and Quoc V Le. Hypernetworks. *arXiv preprint arXiv:1609.09106*, 2016.

[16] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016.

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

[18] Martin Hirzer, Csaba Beleznai, Peter M. Roth, and Horst Bischof. Person re-identification by descriptive and discriminative classification. In *SCIA*, 2011.

[19] Mahdi M. Kalayeh, Emrah Basaran, Muhittin Gökmen, Mustafa E. Kamasak, and Mubarak Shah. Human semantic parsing for person re-identification. In *CVPR*, 2018.

[20] Aditya Khosla, Tinghui Zhou, Tomasz Malisiewicz, Alexei A Efros, and Antonio Torralba. Undoing the damage of dataset bias. In *ECCV*, 2012.

[21] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[22] Da Li, Yongxin Yang, Yi-Zhe Song, and Timothy M Hospedales. Deeper, broader and artier domain generalization. In *ICCV*, 2017.

[23] Da Li, Yongxin Yang, Yi-Zhe Song, and Timothy M Hospedales. Learning to generalize: Meta-learning for domain generalization. In *AAAI*, 2017.

[24] Jiawei Li, Andy J Ma, and Pong C Yuen. Semi-supervised region metric learning for person re-identification. *IJCV*, 2018.

[25] Kai Li, Zhengming Ding, Sheng Li, and Yun Fu. Discriminative semi-coupled projective dictionary learning for low-resolution person re-identification. In *AAAI*, 2018.

[26] Wei Li and Xiaogang Wang. Locally aligned feature transforms across views. In *CVPR*, 2013.

[27] Wei Li, Rui Zhao, Tong Xiao, and Xiaogang Wang. Deepreid: Deep filter pairing neural network for person re-identification. In *CVPR*, 2014.

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

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

[30] Shan Lin, Haoliang Li, Chang-Tsun Li, and Alex Chichung Kot. Multi-task mid-level feature alignment network for unsupervised cross-dataset person re-identification. In *BMVC*, 2018.

[31] Chen Change Loy, Tao Xiang, and Shaogang Gong. Time-delayed correlation analysis for multi-camera activity understanding. *IJCV*, 2010.

[32] Tetsu Matsukawa, Takahiro Okabe, Einoshin Suzuki, and Yoichi Sato. Hierarchical gaussian descriptor for person re-identification. In *CVPR*, 2016.

[33] Saeid Motiian, Marco Piccirilli, Donald A Adjeroh, and Gianfranco Doretto. Unified deep supervised domain adaptation and generalization. In *ICCV*, 2017.

[34] Krikamol Muandet, David Balduzzi, and Bernhard Schölkopf. Domain generalization via invariant feature representation. In *ICML*, 2013.

[35] Alex Nichol, Joshua Achiam, and John Schulman. On first-order meta-learning algorithms. *CoRR*, 2018.

[36] Sakrapee Paisitkriangkrai, Chunhua Shen, and Anton Van Den Hengel. Learning to rank in person re-identification with metric ensembles. In *CVPR*, 2015.

[37] Sachin Ravi and Hugo Larochelle. Optimization as a model for few-shot learning. In *ICLR*, 2017.

[38] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *CVPR*, 2018.

[39] Shiv Shankar, Vihari Piratla, Soumen Chakrabarti, Siddhartha Chaudhuri, Preethi Jyothi, and Sunita Sarawagi. Generalizing across domains via cross-gradient training. In *ICLR*, 2018.

[40] Qiao Siyuan, Liu Chenxi, Shen Wei, and Yuille Alan. Few-shot image recognition by predicting parameters from activations. In *CVPR*, 2018.

[41] Jake Snell, Kevin Swersky, and Richard Zemel. Prototypical networks for few-shot learning. In *NIPS*, 2017.

[42] Chunfeng Song, Yan Huang, Wanli Ouyang, and Liang Wang. Mask-guided contrastive attention model for person re-identification. In *CVPR*, 2018.

[43] Yifan Sun, Liang Zheng, Yi Yang, Qi Tian, and Shengjin Wang. Beyond part models: Person retrieval with refined part pooling. In *ECCV*, 2018.

[44] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *CVPR*, 2016.

[45] Sebastian Thrun and Lorien Pratt. *Learning to learn*. Springer Science & Business Media, 2012.

[46] Maoqing Tian, Shuai Yi, Hongsheng Li, Shihua Li, Xuesen Zhang, Jianping Shi, Junjie Yan, and Xiaogang Wang. Eliminating background-bias for robust person re-identification. In *CVPR*, 2018.

[47] Rahul Rama Varior, Mrinal Haloi, and Gang Wang. Gated siamese convolutional neural network architecture for human re-identification. In *ECCV*, 2016.

[48] Rahul Rama Varior, Bing Shuai, Jiwen Lu, Dong Xu, and Gang Wang. A siamese long short-term memory architecture for human re-identification. In *ECCV*, 2016.

[49] Oriol Vinyals, Charles Blundell, Tim Lillicrap, Daan Wierstra, et al. Matching networks for one shot learning. In *NIPS*, 2016.

[50] Jingya Wang, Xiatian Zhu, Shaogang Gong, and Wei Li. Transferable joint attribute-identity deep learning for unsupervised person re-identification. In *CVPR*, 2018.

[51] Yan Wang, Lequn Wang, Yurong You, Xu Zou, Vincent Chen, Serena Li, Gao Huang, Bharath Hariharan, and Kilian Q Weinberger. Resource aware person re-identification across multiple resolutions. In *CVPR*, 2018.

[52] Longhui Wei, Shiliang Zhang, Wen Gao, and Qi Tian. Person transfer gan to bridge domain gap for person re-identification. In *CVPR*, 2018.

[53] Tong Xiao, Shuang Li, Bochao Wang, Liang Lin, and Xiaogang Wang. End-to-end deep learning for person search. *arXiv preprint arXiv:1604.01850*, 2016.

[54] Qian Yu, Xiaobin Chang, Yi-Zhe Song, Tao Xiang, and Timothy M Hospedales. The devil is in the middle: Exploiting mid-level representations for cross-domain instance matching. *arXiv preprint arXiv:1711.08106*, 2017.

[55] Li Zhang, Tao Xiang, and Shaogang Gong. Learning a discriminative null space for person re-identification. In *CVPR*, 2016.

[56] Xuan Zhang, Hao Luo, Xing Fan, Weilai Xiang, Yixiao Sun, Qiqi Xiao, Wei Jiang, Chi Zhang, and Jian Sun. Alignedreid: Surpassing human-level performance in person re-identification. In *CVPR*, 2017.

[57] Liang Zheng, Liyue Shen, Lu Tian, Shengjin Wang, Jingdong Wang, and Qi Tian. Scalable person re-identification: A benchmark. In *CVPR*, 2015.

[58] Wei-Shi Zheng, Shaogang Gong, and Tao Xiang. Associating groups of people. In *BMVC*, 2009.

[59] Zhedong Zheng, Liang Zheng, and Yi Yang. A discriminatively learned cnn embedding for person reidentification. *TOMM*, 2017.

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

[61] Zhun Zhong, Liang Zheng, Shaozi Li, and Yi Yang. Generalizing a person retrieval model hetero-and homogeneously. In *ECCV*, 2018.