# Learning Generalisable Omni-Scale Representations for Person Re-Identification

Kaiyang Zhou, Yongxin Yang, Andrea Cavallaro, and Tao Xiang

**Abstract**—An effective person re-identification (re-ID) model should learn feature representations that are both discriminative, for distinguishing similar-looking people, and generalisable, for deployment across datasets without any adaptation. In this paper, we develop novel CNN architectures to address both challenges. First, we present a re-ID CNN termed omni-scale network (OSNet) to learn features that not only capture different spatial scales but also encapsulate a synergistic combination of multiple scales, namely omni-scale features. The basic building block consists of multiple convolutional streams, each detecting features at a certain scale. For omni-scale feature learning, a unified aggregation gate is introduced to dynamically fuse multi-scale features with channel-wise weights. OSNet is lightweight as its building blocks comprise factorised convolutions. Second, to improve generalisable feature learning, we introduce instance normalisation (IN) layers into OSNet to cope with cross-dataset discrepancies. Further, to determine the optimal placements of these IN layers in the architecture, we formulate an efficient differentiable architecture search algorithm. Extensive experiments show that, in the conventional same-dataset setting, OSNet achieves state-of-the-art performance, despite being much smaller than existing re-ID models. In the more challenging yet practical cross-dataset setting, OSNet beats most recent unsupervised domain adaptation methods without using any target data. Our code and models are released at `https://github.com/KaiyangZhou/deep-person-reid`.

**Index Terms**—Person Re-Identification; Omni-Scale Learning; Lightweight Network; Cross-Domain Re-ID; Neural Architecture Search



## 1 Introduction

Person re-identification (re-ID), as a fine-grained instance recognition problem, aims to match people across non-overlapping camera views. With the development of deep learning technology, recent research in person re-ID has shifted from tedious feature engineering [1], [2] to end-to-end feature representation learning with deep neural networks [3], [4], [5], [6], especially convolutional neural networks (CNNs).

Though the re-ID performance has been improved significantly thanks to end-to-end representation learning with CNNs, two problems remain unsolved. They hinder large-scale deployment of re-ID models in real-world applications. The first problem is *discriminative* feature learning. As an instance recognition task, re-identifying people under disjoint camera views needs to overcome both intra-class variations and inter-class ambiguity. For instance, in Fig. 1(a) the view change from front to back across cameras brings large appearance changes in the backpack region, making person matching a challenging task. Moreover, from a distance as typical in video surveillance scenes, people can look incredibly similar, as exemplified by the false matches in Fig. 1. This requires the re-ID features to capture fine-grained details for distinguishing people of similar appearances (e.g., the sun glasses in Fig. 1(d)).


Fig. 1. Example images from four person re-ID datasets showing that discriminative and generalisable features are essential for re-ID. Each sub-figure contains, from left to right, a query image, a true match, and a false match (distractor).

The second problem is *generalisable* feature learning. Due to intrinsic domain gaps between re-ID datasets caused by differences in, for example, lighting conditions, background and viewpoint (see Fig. 1), directly applying a re-ID model trained on a source dataset to an unseen target dataset will typically lead to large performance drops [7], [8], [9], [10]. This suggests that the learned re-ID features severely overfit the source domain data and hence are not domain-generalisable. A domain-generalisable re-ID model has great values for real-world large-scale deployment. This is because such a model can work in any unseen scenarios, without the need to go through the tedious processes of data collection, annotation, and model updating/fine-tuning.

In this paper, we address both problems by designing novel CNN architectures. First, we argue that discriminative re-ID features need to be of *omni-scale*, defined as the combination of variable homogeneous scales and heteroge-

- *K. Zhou is with Nanyang Technological University, Singapore. E-mail: {kaiyang.zhou}@ntu.edu.sg*
- *Y. Yang and T. Xiang are with the University of Surrey, Guildford, GU2 7XH, UK. E-mail: {k.zhou, y.yang, t.xiang}@surrey.ac.uk*
- *A. Cavallaro is with Queen Mary University of London, London, E1 4NS, UK. E-mail: a.cavallaro@qmul.ac.uk*

neous scales, each of which is composed of a mixture of multiple scales. The need for omni-scale features is evident from Fig. 1. Specifically, to match people and distinguish them from distractors that cause false matches, the features corresponding to small local regions (e.g., shoes and glasses) and global whole body regions are equally important. For instance, given the query image in Fig. 1(a, left), looking at the global-scale features (e.g., young man, a white T-shirt + grey shorts combo) could narrow down the search to the true match (middle) and a distractor (right). Now the local-scale features come into play—the shoe region explains away the fact that the person on the right is a distractor (trainers vs. sandals). However, for more challenging cases, even the features of variable homogeneous scales are not enough; more complicated and richer features that span multiple scales are required. For instance, to eliminate the distractor in Fig. 1(b, right), one needs the features that represent a white T-shirt with a specific logo in the front. Note that the logo is not distinctive on its own—without the white T-shirt as context, it can be confused with many other patterns. Moreover, the white T-shirt is likely everywhere in summer, e.g., Fig. 1(a). It is however the unique combination, captured by heterogeneous features spanning both small (logo size) and medium (upper body size) scales, that makes the features most effective.

We therefore propose *omni-scale network* (OSNet), a novel CNN architecture designed specifically for omni-scale feature learning. The underpinning building block of OSNet consists of multiple convolutional streams with different receptive field sizes[1] (see Fig. 2). The feature scale that each stream focuses on is determined by *exponent*, a new dimension factor that linearly increases across streams to ensure that various scales can be captured in each individual block. Critically, the resulting multi-scale feature maps are dynamically fused by channel-wise weights generated by a unified aggregation gate (AG). The AG is a mini-network sharing parameters across all streams with a number of desirable properties for effective model training. Since the AG are trainable, the generated channel-wise weights are input-dependent, realising a dynamic scale fusion. This novel AG design is crucial for learning omni-scale feature representations: conditioning on a specific input image, the gate can focus on a single scale by assigning a dominant weight to a particular stream or scale; alternatively, it can select and mix jointly to produce features with heterogeneous scales.

Another key characteristic of OSNet is *lightweight*. A lightweight re-ID model has a couple of benefits: (1) Re-ID datasets are often of moderate size due to difficulties in collecting cross-camera matched person images. A lightweight network with a small number of parameters is thus less prone to overfitting; (2) In large-scale surveillance applications (e.g., city-wide surveillance with thousands of cameras), the most practical way for re-ID is to perform feature extraction at the camera end and send the extracted features to the central server rather than the raw videos. For on-device processing, small re-ID networks are clearly preferred. To this end, in our building block we factorise standard convolutions with pointwise and depthwise convolutions [11], [12], making OSNet not only discriminative in feature learning but also efficient in implementation and deployment.

1. In this paper, 'scale' and 'receptive field' are used interchangeably.

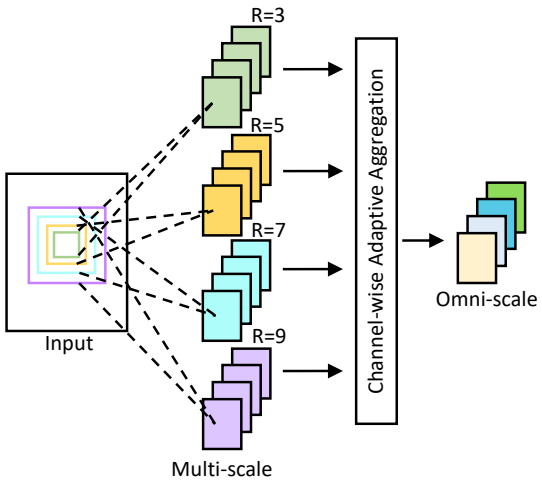


Fig. 2. A schematic of OSNet building block. R: Receptive field size.

To address the second problem caused by domain gaps across different re-ID datasets, we notice that these gaps are typically reflected by different image styles, such as brightness, colour temperatures and view angles (see Fig. 1). These style variations are caused by differences in both lighting condition and camera characteristics/setup in different camera networks. Existing works address this problem using unsupervised domain adaptation (UDA) methods [7], [8], [9], [10]. These require unlabelled target domain data to be available for model adaptation. In contrast, we treat this as a *more general* domain generalisation (DG) problem [13] *without using any target domain data*. By eliminating the tedious processes of data collection and model updating given a new target domain, our approach enables a re-ID model trained using source datasets to be applied out-of-the-box for any *unseen* target dataset.

Concretely, our solution to domain-generalisable feature learning is to introduce instance normalisation (IN) [14] to our OSNet architecture. Unlike batch normalisation (BN) [15] based on mini-batch level statistics, IN calibrates a sample using inner statistics, thus eliminating instance-specific contrast and style that are largely affected by domain-specific environments [16], [17], [18]. In this way, IN can naturally address the fundamental style discrepancy problem in cross-domain person re-ID. However, IN has never been exploited for solving such a cross-domain issue in re-ID. It has been noticed that where and how many IN layers to have in a CNN are critical for DG [19], [20], but there is no clear guidance on how to design the network architecture. We therefore propose to *learn* the optimal model configuration directly from data via differentiable architecture search. More specifically, we design a novel search space, which contains candidate building blocks with different IN configurations. As the discrete selection variables disable search differentiation, we further leverage the Gumbel-Softmax [21], [22] to create continuous representations for candidate selection, allowing end-to-end optimisation via gradient descent.

The contributions can be summarised as follows. **(1)** We introduce, for the first time, the concept of omni-scale

feature learning for discriminative person re-ID. This leads to OSNet, a novel CNN architecture capable of simultaneously learning *homogeneous*- and *heterogeneous*-scale features. By using factorised convolutions, OSNet is lightweight with only 2.2 million parameters—more than one order of magnitude smaller than the common ResNet50-based re-ID models. **(2)** To improve domain generalisation cross datasets we incorporate instance normalisation (IN) into the OSNet design through differentiable architecture search, which we call OSNet-AIN. To our knowledge, this is the first work that explores both IN and neural architecture search for cross-domain re-ID. **(3)** We evaluate OSNet by conducting extensive experiments on seven person re-ID datasets. In the same-domain setting, OSNet achieves state-of-the-art performance, outperforming many far larger re-ID models, often by a clear margin. Importantly, in the cross-domain setting, OSNet-AIN exhibits a remarkable generalisation ability: it beats most recent unsupervised domain adaptation (UDA) methods on unseen target domains while maintaining strong source domain performance, *requiring neither the target domain data nor per-domain training*. Our code and models are publicly available to facilitate future research in re-ID.[2]

## 2 Related Work

**CNN Architectures for Person Re-ID** Most existing deep re-ID methods [5], [6], [23], [24], [25] adopt CNN architectures that are originally designed for generic object classification problems, especially those ImageNet-winning models [26], [27], [28]. These architectures are intrinsically limited for instance recognition in re-ID. Modifications are thus made to tackle problems specific to re-ID, such as misalignment [29], [30] and pose variations [25], [31]. As persons usually stand upright, [23], [30], [32] partition feature maps horizontally and inject parallel supervision signals to each stripe in order to enhance the learning of part-level features. Attention mechanisms are designed in [5], [24], [33] to focus feature learning on the foreground image regions. In [34], [35], [36], [37], [38], [39], body part specific CNNs are learned by means of off-the-shelf pose detectors. In [40], [41], [42], CNNs are branched to learn representations from global and local image regions. Since low-level visual cues such as colour are relevant for re-ID, [6], [43], [44], [45] combine multi-level features extracted at different CNN layers. However, none of the existing re-ID networks can learn multi-scale features *explicitly* at each CNN layer as in our OSNet. Unlike OSNet, they typically rely on external pose models and/or hand-pick some layers for multi-scale learning. Moreover, the ability to learn heterogeneous-scale features representing the mixture of different scales is also missing.

**Multi-Scale and Multi-Stream CNNs** As far as we know, the concept of omni-scale deep feature learning has never been introduced before. Nonetheless, the importance of multi-scale feature learning has been recognised recently, and the multi-stream building block design has also been adopted in re-ID [46]. Compared to a small number of re-ID networks that have multi-stream building blocks [6], [47], OSNet is significantly different. Specifically, the layer design in [6] is based on ResNeXt [48], where each stream learns features at the same scale, while the streams in each OSNet block cover different scales. The network in [47] is built on Inception [27], [49], where the multi-streams were originally designed for low computational cost with a hand-crafted mixture of convolutional and pooling layers. In contrast, our building block uses a scale-controlling factor to diversify the spatial scales. Moreover, [47] fuses multi-stream features with learnable but fixed-once-learned stream-wise weights only at the final block. Whereas we fuse multi-scale features within each building block using dynamic (input-dependent) channel-wise weights to learn combinations of multi-scale patterns. Therefore, only our OSNet is capable of learning omni-scale features with each feature channel potentially capturing more discriminative features of either a single scale or a weighted mixture of multiple scales. Our experiments (in Sec. 4.1) show that OSNet significantly outperforms the models in [6], [46], [47].

**Lightweight Network Design** With embedded AI becoming topical, lightweight CNN design has attracted increasing attention. SqueezeNet [50] compresses feature dimensions using $1\times1$ convolutions. IGCNet [51], ResNeXt [48] and CondenseNet [52] leverage group convolutions. Xception [53] and the MobileNet series [11], [12] are based on depth-wise separable convolutions. Dense $1 \times 1$ convolutions are grouped with channel shuffling in ShuffleNet [54]. In terms of lightweight design, our OSNet is similar to MobileNet—both use factorised convolutions—but with a modification in the ordering that empirically works better for omni-scale feature learning (see Sec. 3.1 for the details).

**Domain Generalisation** Cross-dataset generalisation has been studied in re-ID [55], but no specific designs have ever been introduced to make re-ID models more *intrinsically* generalisable. Recently, unsupervised domain adaptation (UDA) methods [7], [8], [9], [10], [56] have been extensively studied to adapt a re-ID model from source to target domain. However, UDA methods have to use unlabelled target domain data, so data collection and (per-domain) model update are still required. In contrast, without these steps, our OSNet-AIN is much more efficient in practice. Beyond re-ID, the problem of domain generalisation (DG) has been investigated in deep learning [57], [58], [59], [60], [61], [62] (see [13] for a comprehensive survey in this topic). However, most existing DG methods [58], [59], [63] assume that the source and target domains have the same label space, which apparently conflicts with the disjoint label space case in re-ID. Some recent few-shot meta-learning approaches are also re-purposed for DG [64]. However, they assume a fixed number of classes for the target domain and are trained specifically for that number using source data. Therefore, they cannot be directly applied to re-ID, where the target domain has a different and variable number of identity classes.

Our DG-oriented re-ID solution is based on instance normalisation (IN) layers [14]. IN's ability to eliminate instance-specific style discrepancy has been investigated for the style transfer task [16], [17], [18]. Recently, several works have attempted to integrate CNNs with IN layers to improve model generalisation. [20] tackle multi-domain learning by fusing

2. https://github.com/KaiyangZhou/deep-person-reid.

BN and IN with a convex weight. In [19], an architecture called IBN-Net is engineered by inserting IN to shallow CNN layers for cross-domain semantic segmentation. The empirical study in [19] suggests that appearance variations mainly lie in shallow CNN layers, and therefore, inserting IN to shallow layers should be more effective. However, there is no clear definition of what 'shallow' layers are in deep neural networks. Moreover, person re-ID is an instance recognition problem, for which the empirical rule derived from the semantic segmentation task might not work. In this paper, instead of hand-picking layers for inserting IN, we propose to use neural architecture search to optimally explore the capability of IN for improving DG.

**Neural Architecture Search** Neural architecture search (NAS) aims to automate the process of network architecture engineering. Early NAS methods are typically based on either reinforcement learning (RL) [65], [66] or evolutionary algorithm (EA) [67], [68], where hundreds and thousands of models need to be trained from scratch and evaluated on a separate validation set to provide the supervision signal (reward for RL and fitness score for EA). This is computationally extremely expensive, requiring hundreds or even thousands of GPU days to complete the search. The follow-up research is mainly focused on accelerating the search by, for example, weight sharing [69], [70]. Recently, there has been a growing interest in modelling NAS with directed acyclic graph (DAG) and using continuous representations for end-to-end optimisation. DARTS [71] uses softmax to relax the discrete one-hot actions over search space. Similarly, SNAS [72] and GDAS [73] utilise a discrete gradient estimator [21], [22] to overcome the non-differentiable nature in categorical variables. To address the high GPU memory problem in differentiable NAS, ProxylessNAS [74] forces gradients to propagate through only one of the candidate paths. Different from these NAS methods, we do not search architecture from scratch. Instead, we base the architecture on OSNet and leverage NAS to find the best way to combine OSNet with IN.

An **earlier and preliminary version** of this work was published in ICCV'19 [75]. Compared with [75], which only focuses on discriminative feature learning for same-domain re-ID, this paper brings up the problem of generalisable feature learning for cross-domain re-ID, which has been largely overlooked in existing re-ID research. To that end, this work proposes to combine OSNet with IN by automatically searching for the best model configuration directly from data. Extensive experiments in the cross-domain re-ID setting, together with a comprehensive comparison with state-of-the-art cross-domain re-ID methods, demonstrate that our OSNet achieves strong performance on unseen target datasets even without 1) using any target domain data and 2) undesirable per-domain model adaptation steps.

## 3 Omni-Scale Network for Person Re-ID

In this section, we detail the design of our omni-scale network (OSNet), which is aimed at learning omni-scale feature representations for person re-ID. We first discuss depthwise separable convolutions, which are used to make OSNet lightweight. Then, we introduce our novel omni-scale residual block for learning discriminative re-ID features. Finally, to enhance generalisation in unseen datasets, we extend OSNet by adding instance normalisation (IN) layers and further present a differentiable architecture search mechanism to automatically infer the optimal IN configuration.

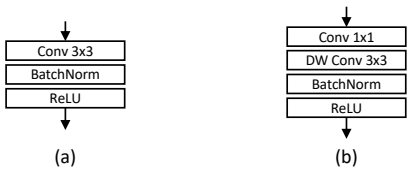


Fig. 3. (a) Standard and (b) Lite $3\times3$ convolution. DW: Depth-Wise.

### 3.1 Depthwise Separable Convolutions

For lightweight network design, we adopt the depthwise separable convolution [11], [53]. The basic idea is to divide a convolution layer, $\mathrm{ReLU}(\boldsymbol{w} * \boldsymbol{x})$, with kernel $\boldsymbol{w} \in \mathbb{R}^{k\times k\times c\times c'}$, into two separate layers, $\mathrm{ReLU}((\boldsymbol{v} \circ \boldsymbol{u}) * \boldsymbol{x})$, with depthwise kernel $\boldsymbol{u} \in \mathbb{R}^{k\times k\times 1\times c'}$ and pointwise kernel $\boldsymbol{v} \in \mathbb{R}^{1\times 1\times c\times c'}$, where $*$ denotes convolution, $k$ the kernel size, $c$ the input channel width and $c'$ the output channel width. Given an input tensor $\boldsymbol{x} \in \mathbb{R}^{h\times w\times c}$ of height $h$ and width $w$, the computational cost is reduced from $h \cdot w \cdot k^2 \cdot c \cdot c'$ to $h \cdot w \cdot (k^2 + c) \cdot c'$, and the parameter size from $k^2 \cdot c \cdot c'$ to $(k^2 + c) \cdot c'$. In our implementation, we find that $\mathrm{ReLU}((\boldsymbol{u} \circ \boldsymbol{v}) * \boldsymbol{x})$ (pointwise $\rightarrow$ depthwise instead of depthwise $\rightarrow$ pointwise) turns out to be more effective for omni-scale feature learning.[3] We call such layer *Lite* $3\times3$ hereafter. The design is depicted in Fig. 3.

### 3.2 Omni-Scale Residual Block

The building block in OSNet is based on the residual bottleneck [26], but equipped with the Lite $3\times3$ layer (Fig. 4). Given an input $\boldsymbol{x}$, a residual bottleneck aims to learn a residual $\tilde{\boldsymbol{x}}$ via a mapping function $F$, i.e.

$$\boldsymbol{y} = \boldsymbol{x} + \tilde{\boldsymbol{x}}, \quad \text{with} \quad \tilde{\boldsymbol{x}} = F(\boldsymbol{x}), \tag{1}$$

where $F$ denotes a Lite $3\times3$ convolution layer that learns *single-scale* features (i.e. receptive field = $3\times3$). Note that here the $1\times1$ convolution layers are ignored in notation as they are used to manipulate feature channels and do not contribute to the aggregation of spatial information [26], [48].

**Multi-Scale Feature Learning** To achieve multi-scale feature learning, we extend the residual function $F$ by introducing a new dimension, *exponent* $t$, to represent the feature scale. For $F^t$, with $t > 1$, we stack $t$ Lite 3×3 layers, resulting in a receptive field of size $(2t + 1) \times (2t + 1)$. Then, the residual to be learned, $\tilde{\boldsymbol{x}}$, is the sum of incremental scales of representations up to $T$:

$$\tilde{\boldsymbol{x}} = \sum_{t=1}^{T} F^t(\boldsymbol{x}), \qquad T \geqslant 1. \tag{2}$$

When $T = 1$, Eq. (2) reduces to Eq. (1), i.e. the baseline single-scale bottleneck as shown in Fig. 4(a). Considering the computational cost, we use $T = 4$ in this paper where the largest receptive field is $9\times9$. This is depicted in Fig. 4(b).

3. The subtle difference between these two orders is when the channel width is increased: pointwise $\rightarrow$ depthwise increases the channel width before spatial aggregation.

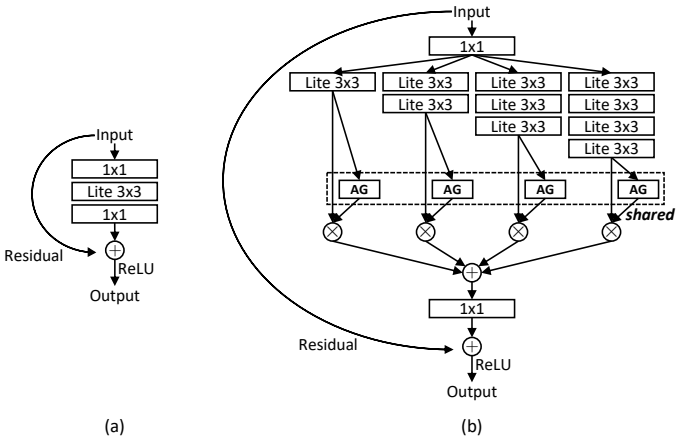


Fig. 4. (a) Baseline bottleneck. (b) OSNet bottleneck. AG: Aggregation Gate. The first/last $1\times1$ layers used to reduce/restore feature dimension.

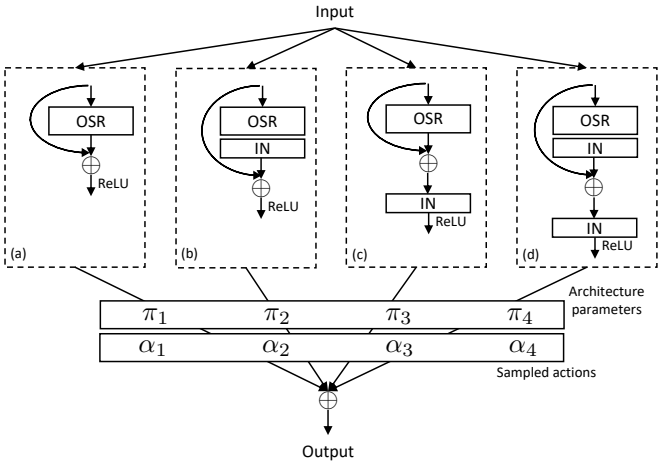


Fig. 5. Our architecture search space consists of four different omni-scale residual (OSR) blocks each with a learnable parameter $\pi$. During a forward pass, the selected candidate is determined by sampling discrete actions (one-hot) from a categorical distribution parameterised by the architecture parameters. To make the computational graph differentiable, we relax the discrete variables to continuous representations using the Gumbel-Softmax [21], [22]. IN: Instance Normalisation [14].

**Dynamic and Unified Aggregation Gate** So far, each individual stream gives us features of only one specific scale, i.e. *scale homogeneous*. To learn effective omni-scale features, we propose to combine the outputs of different streams in a *dynamic* way, i.e. different weights are assigned to different scales according to the input image, rather than being fixed and identical for all the data after training. More specifically, the dynamic scale fusion is achieved by a novel aggregation gate (AG), which is essentially a *learnable neural network*.

Let $\boldsymbol{x}^t$ denote $F^t(\boldsymbol{x})$, the omni-scale residual $\tilde{\boldsymbol{x}}$ is then formulated by

$$\tilde{\boldsymbol{x}} = \sum_{t=1}^{T} G(\boldsymbol{x}^t) \odot \boldsymbol{x}^t, \quad \text{with} \quad \boldsymbol{x}^t \triangleq F^t(\boldsymbol{x}), \tag{3}$$

where $G(\boldsymbol{x}^t)$ is a data-conditioned vector with length spanning the entire channel dimension of input $\boldsymbol{x}^t$, and $\odot$ denotes the Hadamard product. $G$ is implemented as a mini-network composed of a non-parametric global average pooling layer [76] and a multi-layer perceptron (MLP) with one ReLU-activated hidden layer, followed by the sigmoid activation. To reduce parameter overhead, we follow [77], [78] to reduce the MLP's hidden dimension with a reduction ratio, which is set to 16.

In our design, the AG is *shared* across all the feature streams in the same omni-scale residual block (dashed box in Fig. 4(b)). In spirit, this is similar to the parameter sharing of convolution filters in CNNs, resulting in a number of advantages. First, the number of parameters is independent of the number of streams $T$, thus the model becomes more scalable. Second, unifying AG has a nice property when performing gradient backpropagation. Concretely, suppose the network is supervised by a differentiable loss function $\mathcal{L}$ and the gradient $\frac{\partial \mathcal{L}}{\partial \tilde{\boldsymbol{x}}}$ can be computed. The gradient w.r.t $G$, based on Eq. (3), is

$$\frac{\partial \mathcal{L}}{\partial G} = \frac{\partial \mathcal{L}}{\partial \tilde{\boldsymbol{x}}} \frac{\partial \tilde{\boldsymbol{x}}}{\partial G} = \frac{\partial \mathcal{L}}{\partial \tilde{\boldsymbol{x}}} (\sum_{t=1}^{T} \boldsymbol{x}^t). \tag{4}$$

It is clear that the second term in Eq. (4) indicates that supervision signals from all streams are gathered together to guide the learning of $G$. This desirable property disappears when each stream has its own gate.

We further stress two design considerations. First, in contrast to using a single-scalar gate function that provides a coarse scale fusion, we use *channel-wise vector* gating, i.e. AG's output $G(\boldsymbol{x}^t)$ is a vector rather a scalar for the $t$-th stream. This design results in a more fine-grained fusion that tunes each feature channel. Second, the weights are *dynamically* computed by conditioning on the input data. This is crucial for re-ID as the training and test data describe disjoint identity populations; input adaptive feature-scale fusion is hence more effective and scalable.

### 3.3 Inserting Instance Normalisation Layers

Different from batch normalisation (BN) [15], which normalises each sample using statistics computed over a mini-batch, IN performs normalisation on each sample using its own mean and standard deviation [16]. As such, IN allows the instance-specific style information to be effectively removed. Therefore inserting IN layers into a re-ID CNN has the potential of eliminating image style differences caused by distinct environments, lighting conditions, camera setups, etc. in each dataset. However, it is unclear how to integrate a re-ID CNN with IN to maximise the gain, e.g., which layers to insert? Inside or outside a residual block?

**Architecture Search Space** We propose to learn the optimal way of integrating OSNet with IN by neural architecture search (NAS). To that end, we define a novel search space $\Omega$ consisting of candidate omni-scale (OS) blocks in different IN-incorporating designs. Specifically, besides the standard OS block (Fig. 4(b)), we design three other variants (see Fig. 5 for an illustration of our search space). Following [79], we keep the residual learning module unchanged, i.e. only adding IN after the residual. For clarity, we refer to Fig. 5(b-d) as OS+$\text{IN}_{\text{in}}$ block, OS+$\text{IN}_{\text{out}}$ block, and OS+$\text{IN}_{\text{in}-\text{out}}$ block, respectively.

**Formulation** In our NAS formulation, the output $\boldsymbol{y}$ of each OS layer is obtained as a weighted sum of operations

in $\Omega$,

$$\boldsymbol{y} = \sum_{\omega \in \Omega} \alpha_\omega \omega(\boldsymbol{x}), \tag{5}$$

where $\boldsymbol{\alpha} = [\alpha_\omega]_{\omega \in \Omega}$ is a $|\Omega|$-dimensional one-hot vector, with the activated element "1" corresponding to the IN design selection.

The objective is to minimise the following expectation through jointly optimising the model architecture $\boldsymbol{\alpha}$ and the parameters $\boldsymbol{\theta}$ as,

$$\mathbb{E}[\mathcal{L}(\boldsymbol{x}, \boldsymbol{\theta}, \boldsymbol{\alpha})]. \tag{6}$$

For an OSNet with $m$ blocks, the search space contains a total of $4^m$ different architecture design choices. A key challenge lies in the optimisation of the discrete selection that is non-differentiable due to its discontinuous nature. This disables the adoption of strong gradient-based architecture search optimisation.

**Relaxation and Reparameterisation Trick** To solve the non-differentiable problem, we develop a continuous relaxation and reparameterisation strategy. This is achieved by first treating $\boldsymbol{\alpha}$ as a continuous $\ell_1$-normalised random variable sampled from a probability distribution $P_{\boldsymbol{\pi}}$ parameterised by $\boldsymbol{\pi}$ (i.e. the target architecture parameters) as

$$P(\alpha_\omega = 1) = \frac{\exp(\pi_\omega)}{\sum_{\omega' \in \Omega} \exp(\pi_{\omega'})}, \tag{7}$$

and then reparameterising this sampling process by the Gumbel-Softmax [21], [22] defined as

$$\alpha_\omega = f_{\boldsymbol{\pi}}(z_\omega) = \frac{\exp\left((\log \pi_\omega + z_\omega)/\lambda\right)}{\sum_{\omega' \in \Omega} \exp\left((\log \pi_{\omega'} + z_{\omega'})/\lambda\right)}, \tag{8}$$

where $\lambda$ is the softmax temperature and $z_\omega \sim \mathrm{Gumbel}(0, 1)$ a Gumbel distribution. Concretely, $z_\omega$ is obtained by the following transformation of the uniform distribution: $z_\omega = -\log(-\log(u_\omega))$, where $u_\omega \sim \mathrm{Uniform}(0, 1)$.

The objective function is reformulated as:

$$\mathbb{E}_{\boldsymbol{z} \sim P(\boldsymbol{z})}[\mathcal{L}(\boldsymbol{x}, \boldsymbol{\theta}, f_{\boldsymbol{\pi}}(\boldsymbol{z}))], \tag{9}$$

which is fully differentiable w.r.t. both $\boldsymbol{\theta}$ and $\boldsymbol{\pi}$. The gradients can be approximated by Monte Carlo sampling [80],

$$\nabla_{\boldsymbol{\theta}} \mathbb{E}_{\boldsymbol{z} \sim P(\boldsymbol{z})}[\mathcal{L}(\boldsymbol{x}, \boldsymbol{\theta}, f_{\boldsymbol{\pi}}(\boldsymbol{z}))] \tag{10}$$

$$= \mathbb{E}_{\boldsymbol{z} \sim P(\boldsymbol{z})}[\nabla_{\boldsymbol{\theta}} \mathcal{L}(\boldsymbol{x}, \boldsymbol{\theta}, f_{\boldsymbol{\pi}}(\boldsymbol{z}))] \tag{11}$$

$$\simeq \frac{1}{S} \sum_{s=1}^{S} \nabla_{\boldsymbol{\theta}} \mathcal{L}(\boldsymbol{x}, \boldsymbol{\theta}, f_{\boldsymbol{\pi}}(\boldsymbol{z}^s))], \tag{12}$$

where $S$ denotes the number of sampling steps, and similarly,

$$\nabla_{\boldsymbol{\pi}} \mathbb{E}_{\boldsymbol{z} \sim P(\boldsymbol{z})}[\mathcal{L}(\boldsymbol{x}, \boldsymbol{\theta}, f_{\boldsymbol{\pi}}(\boldsymbol{z}))] \tag{13}$$

$$= \mathbb{E}_{\boldsymbol{z} \sim P(\boldsymbol{z})}[\nabla_{\boldsymbol{\pi}} \mathcal{L}(\boldsymbol{x}, \boldsymbol{\theta}, f_{\boldsymbol{\pi}}(\boldsymbol{z}))] \tag{14}$$

$$\simeq \frac{1}{S} \sum_{s=1}^{S} \nabla_{\boldsymbol{\pi}} \mathcal{L}(\boldsymbol{x}, \boldsymbol{\theta}, f_{\boldsymbol{\pi}}(\boldsymbol{z}^s))]. \tag{15}$$

In doing so, we transfer the dependency on $\boldsymbol{\pi}$ from $P$ to $f$. Importantly, as proved in [22], when $\lambda \to 0$, the relaxed softmax formulation (Eq. (8)) approaches the discrete $\mathrm{argmax}$ computation, i.e. an unbiased gradient estimator.

**Search Outcome** At the end of search, we derive a compact network architecture by selecting for each layer the OS block with the largest $\pi$, i.e. $\omega^* = \arg\max_{\omega \in \Omega} \pi_\omega$.

TABLE 1
Omni-scale network architecture for person re-ID. Input image size is $256 \times 128$. AIN: Automatic search + Instance Normalisation.

| stage | output | OSNet | OSNet-AIN |
|---|---|---|---|
| conv1 | 128×64, 64 | 7×7 conv, stride 2 | |
| pool | 64×32, 64 | 3×3 max pool, stride 2 | |
| conv2 | 64×32, 256 | OS block | OS+$\mathrm{IN}_{\mathrm{in}}$ block |
| | 64×32, 256 | OS block | OS+$\mathrm{IN}_{\mathrm{in}}$ block |
| transition | 64×32, 256 | 1×1 conv | |
| | 32×16, 256 | 2×2 average pool, stride 2 | |
| conv3 | 32×16, 384 | OS block | OS block |
| | 32×16, 384 | OS block | OS+$\mathrm{IN}_{\mathrm{in}}$ block |
| transition | 32×16, 384 | 1×1 conv | |
| | 16×8, 384 | 2×2 average pool, stride 2 | |
| conv4 | 16×8, 512 | OS block | OS+$\mathrm{IN}_{\mathrm{in}}$ block |
| | 16×8, 512 | OS block | OS block |
| conv5 | 16×8, 512 | 1×1 conv | |
| gap | 1×1, 512 | global average pool | |
| fc | 1×1, 512 | fc | |
| # params | | 2.2M | 2.2M |
| Mult-Adds | | 978.9M | 978.9M |

### 3.4 Network Architecture

**OSNet** is constructed by stacking the proposed lightweight bottleneck (OS block) layer-by-layer. The detailed network architecture is shown in Table 1. For comparison, the same network architecture with normal convolutions has 6.9 million parameters and 3,384.9 million mult-add operations. This is 3× larger than OSNet with the Lite $3 \times 3$ convolution layer design. The OSNet architecture in Table 1 can be easily scaled up or down in practice, to balance model size, computational cost and performance. To this end, we use a width multiplier[4] and an image resolution multiplier, following [11], [12], [54].

**OSNet-AIN** denotes the network architecture with automatically searched IN layers. In the experiment, we run the searching algorithm four times with different random seeds and select the one with the best cross-domain performance as our final model, which is a commonly adopted protocol in the NAS literature [71], [72], [74]. The found best architecture model is shown in the OSNet-AIN column in Table 1. The detailed experimental setup for NAS will be covered in Sec. 4.2. As IN only introduces a small number of parameters, the complexity between OSNet-AIN and OSNet is similar.

**Relation to Prior Architectures** In terms of the multi-stream design, OSNet is related to Inception [27] and ResNeXt [48], but has several crucial differences. 1) First, the multi-stream design in OSNet strictly follows the scale-incremental principle dictated by the exponent variable (see Eq. (2)). Such a design is more effective for covering a wide range of scales. In contrast, Inception was originally

4. Width multiplier with magnitude smaller than 1 works on all layers in OSNet except the last fc layer whose feature dimension is fixed to 512.

TABLE 2
Statistics of person re-ID datasets.

| Dataset | # IDs | # images | # cameras |
|---|---|---|---|
| Market1501 [81] | 1,501 | 32,668 | 6 |
| CUHK03 [3] | 1,467 | 28,192 | 2 |
| Duke [82], [83] | 1,812 | 36,411 | 8 |
| MSMT17 [84] | 4,101 | 126,411 | 15 |
| VIPeR [85] | 632 | 1,264 | 2 |
| GRID [86] | 251 | 1,275 | 6 |
| CUHK01 [87] | 971 | 3,882 | 2 |

designed to have a low computational cost by sharing computations with multiple streams. Therefore, its structure, which includes mixed operations of convolution and pooling, was hand-crafted. ResNeXt has multiple equal-scale streams, thus learning features at the same scale. 2) Second, Inception/ResNeXt aggregates features by concatenation/addition while OSNet uses the unified AG (Eq. (3)) to facilitate the learning of heterogeneous-scale features. Critically, this means that the fusion in OSNet is dynamic and adaptive to each individual input image, which is more effective in dealing with disjoint label space in person re-ID. 3) Third, OSNet uses factorised convolutions and thus the building block and subsequently the whole network is lightweight. Therefore, the OSNet architecture is fundamentally different from Inception and ResNeXt in nature and design. While the AG borrows the design from SENet [78], they differ conceptually with separate purposes. SENet aims to re-calibrate feature channels by re-scaling the activation values for a single stream. Whereas, OSNet aims to selectively fuse multiple feature streams of different receptive field sizes for learning omni-scale features.

## 4 Experiments

### 4.1 Same-Domain Person Re-Identification

We first evaluate OSNet in the conventional person re-ID setting where the model is trained and tested on the same dataset (domain).

**Datasets and Settings** Seven popular re-ID benchmarks are used, including Market1501 [81], CUHK03 [3], DukeMTMC-reID (Duke) [82], [83], MSMT17 [84],[5] VIPeR [85], GRID [86] and CUHK01 [87]. The overall dataset statistics are detailed in Table 2. The first four are typically considered as *big* re-ID datasets—even though their sizes are fairly moderate (around 30k training images for the largest dataset MSMT17). The rest three datasets are generally too *small* to train deep models without proper pre-training [34], [40]. For CUHK03, we use the 767/700 split [88] with the detected images. For VIPeR, GRID and CUHK01 (485/486 split [1]), we follow [29], [34], [40], [42], [44] to perform pre-training on large re-ID datasets and then fine-tune the model on the target dataset, where the results are averaged over 10 random splits. For evaluation metrics, we use cumulative matching characteristics (CMC) rank accuracy and mean average precision (mAP). The performance is reported in percentage.

**Implementation Details** A classification layer (linear fc + softmax) is mounted on the top of OSNet. The training follows the standard classification paradigm where each person identity is regarded as a unique class. Similar to [5], [6], the cross-entropy loss with label smoothing [49] is used for supervision. For fair comparison against existing methods, we implement two versions of OSNet. One is trained from scratch while the other is fine-tuned from ImageNet pre-trained weights. Person matching is based on the cosine distance using 512-D feature vectors extracted from the last fc layer. The batch size and the weight decay are set to 64 and 5e-4 respectively. Images are resized to $256 \times 128$.

For training from scratch, SGD is used to optimise the network for 350 epochs. The learning rate starts from 0.065 and is decayed by 0.1 at the 150-th, the 225-th, and the 300-th epoch, respectively. Data augmentation includes random flip, random crop and random patch.[6] For fine-tuning, we train the network with AMSGrad [102] and the initial learning rate of 0.0015 for 250 epochs. The learning rate is decayed using the cosine annealing strategy [103] (without restart). During the first 10 epochs, the ImageNet pre-trained base network is frozen. Only the randomly initialised classifier is open for training [104]. Data augmentation includes random flip and random erasing [105].

**Results on Big Re-ID Datasets** From Table 3, we have the following observations. **(1)** OSNet achieves the best overall performance, outperforming most recently published methods by a clear margin. It is evident that the performance on re-ID benchmarks, especially Market1501 and Duke, has been saturated lately. Therefore, the improvements obtained by OSNet are significant. Crucially, the improvements are achieved with *much smaller model size*—most top-performing re-ID models are based on the ResNet50 backbone, which has more than 23.5 million parameters (except extra customised modules), whereas our OSNet has only 2.2 million parameters. Notably, OSNet is around $6\times$ smaller than the automatically searched model, Auto-ReID [90], but obtains better performance on three out of four datasets. These results verify the effectiveness of omni-scale feature learning for re-ID, achieved by an extremely compact network. As OSNet is orthogonal to some methods such as the image generation-based DGNet [101], they can be combined to potentially boost the re-ID performance in practice. **(2)** OSNet yields strong performance with or without ImageNet pre-training. Among the very few existing lightweight re-ID models that can be trained from scratch (Auto-ReID, HAN and BraidNet in the top group), OSNet exhibits more significant advantages. For instance, in terms of mAP, on Market1501, OSNet beats Auto-ReID, HAN and BraidNet by 6.4%, 5.3% and 11.5%, respectively. Compared with MobileNetV2, which is a general-purpose lightweight CNN, OSNet achieves a large margin consistently at a similar model size. Overall, these results demonstrate the versatility of OSNet: it enables effective feature tuning from generic object categorisation tasks, and offers robustness against model overfitting when trained from scratch on datasets of moderate size. **(3)** Compared with re-ID models [5], [6], [29], [35], [37], [46] also based on multi-scale/multi-stream architectures, namely Inception or ResNeXt, OSNet

5. Throughout this paper, we use the v1 version for MSMT17.

6. RandomPatch works by (1) constructing a patch pool that stores randomly extracted image patches and (2) pasting a random patch selected from the patch pool onto an input image at random position.

TABLE 3
Results on big re-ID datasets. It is noteworthy that OSNet surpasses most published methods by a clear margin on all datasets with only 2.2 million parameters, far less than the best-performing ResNet-based methods. -: not reported. †: reproduced by us.

| Method | Venue | Backbone | Params (M) | Market1501 R1 | Market1501 mAP | CUHK03 R1 | CUHK03 mAP | Duke R1 | Duke mAP | MSMT17 R1 | MSMT17 mAP |
|---|---|---|---|---|---|---|---|---|---|---|---|
| *Trained from scratch* | | | | | | | | | | | |
| MobileNetV2† [12] | CVPR'18 | MobileNetV2 | 2.2 | 87.0 | 69.5 | 46.5 | 46.0 | 75.2 | 55.8 | 50.9 | 27.0 |
| BraidNet [89] | CVPR'18 | BraidNet | - | 83.7 | 69.5 | - | - | 76.4 | 59.5 | - | - |
| HAN [5] | CVPR'18 | Inception | 4.5 | 91.2 | 75.7 | 41.7 | 38.6 | 80.5 | 63.8 | - | - |
| Auto-ReID [90] | ICCV'19 | Auto | 13.1 | 90.7 | 74.6 | - | - | - | - | - | - |
| OSNet (*ours*) | This work | OSNet | 2.2 | **93.6** | **81.0** | **57.1** | **54.2** | **84.7** | **68.6** | **71.0** | **43.3** |
| *Pre-trained on ImageNet* | | | | | | | | | | | |
| SVDNet [91] | ICCV'17 | ResNet | >23.5 | 82.3 | 62.1 | 41.5 | 37.3 | 76.7 | 56.8 | - | - |
| PDC [35] | ICCV'17 | Inception | >6.8 | 84.1 | 63.4 | - | - | - | - | 58.0 | 29.7 |
| HAP2S [92] | ECCV'18 | ResNet | >23.5 | 84.6 | 69.4 | - | - | 75.9 | 60.6 | - | - |
| DPFL [46] | ICCVW'17 | Inception | >6.8 | 88.6 | 72.6 | 40.7 | 37.0 | 79.2 | 60.6 | - | - |
| DaRe [45] | CVPR'18 | DenseNet | >23.5 | 89.0 | 76.0 | 63.3 | 59.0 | 80.2 | 64.5 | - | - |
| PNGAN [93] | ECCV'18 | ResNet | >23.5 | 89.4 | 72.6 | - | - | 73.6 | 53.2 | - | - |
| GLAD [29] | ACM MM'17 | Inception | >6.8 | 89.9 | 73.9 | - | - | - | - | 61.4 | 34.0 |
| KPM [31] | CVPR'18 | ResNet | >23.5 | 90.1 | 75.3 | - | - | 80.3 | 63.2 | - | - |
| MLFN [6] | CVPR'18 | ResNeXt | 32.5 | 90.0 | 74.3 | 52.8 | 47.8 | 81.0 | 62.8 | - | - |
| FDGAN [94] | NeurIPS'18 | ResNet | >23.5 | 90.5 | 77.7 | - | - | 80.0 | 64.5 | - | - |
| DuATM [24] | CVPR'18 | DenseNet | >7.0 | 91.4 | 76.6 | - | - | 81.8 | 64.6 | - | - |
| Bilinear [37] | ECCV'18 | Inception | >6.8 | 91.7 | 79.6 | - | - | 84.4 | 69.3 | - | - |
| G2G [95] | CVPR'18 | ResNet | >23.5 | 92.7 | 82.5 | - | - | 80.7 | 66.4 | - | - |
| DeepCRF [96] | CVPR'18 | ResNet | 26.1 | 93.5 | 81.6 | - | - | 84.9 | 69.5 | - | - |
| PCB [23] | ECCV'18 | ResNet | 27.2 | 93.8 | 81.6 | 63.7 | 57.5 | 83.3 | 69.2 | 68.2 | 40.4 |
| SGGNN [97] | ECCV'18 | ResNet | >23.5 | 92.3 | 82.8 | - | - | 81.1 | 68.2 | - | - |
| Mancs [98] | ECCV'18 | ResNet | >25.1 | 93.1 | 82.3 | 65.5 | 60.5 | 84.9 | 71.8 | - | - |
| AANet [99] | CVPR'19 | ResNet | >23.5 | 93.9 | 83.4 | - | - | 87.7 | 74.3 | - | - |
| CAMA [100] | CVPR'19 | ResNet | >23.5 | 94.7 | 84.5 | 66.6 | 64.2 | 85.8 | 72.9 | - | - |
| IANet [25] | CVPR'19 | ResNet | >23.5 | 94.4 | 83.1 | - | - | 87.1 | 73.4 | 75.5 | 46.8 |
| DGNet [101] | CVPR'19 | ResNet | >23.5 | **94.8** | 86.0 | 65.6 | 61.1 | 86.6 | 74.8 | 77.2 | 52.3 |
| Auto-ReID [90] | ICCV'19 | Auto | 13.1 | 94.5 | 85.1 | **73.3** | **69.3** | - | - | 78.2 | 52.5 |
| OSNet (*ours*) | This work | OSNet | 2.2 | **94.8** | **86.7** | 72.3 | 67.8 | **88.7** | **76.6** | **79.1** | **55.1** |

is clearly better. As discussed in Sec. 3, this is attributed to the unique ability of OSNet to learn heterogeneous-scale features by combining multiple homogeneous-scale features with the dynamic and unified AG.

**Results on Small Re-ID Datasets** Small re-ID datasets are more challenging for deep re-ID models because they have much less training images and classes than the big datasets. Table 4 compares OSNet with six state-of-the-art deep re-ID methods. On VIPeR, we observe that OSNet outperforms all alternatives by a significant margin (more than 11%). GRID is even more challenging than VIPeR because it has only 250 training images of 125 identities. Moreover, it was captured by real (operational) analogue CCTV cameras installed in busy public spaces, presenting more observation noise. OSNet remains the best on GRID, marginally above JLML [40], which is the current state-of-the-art. On CUHK01, which has around 1,900 training images, OSNet significantly outperforms Spindle and JLML by 6.7% and 16.8%, respectively. Overall, the performance of OSNet on these small datasets is excellent, indicating its promising advantage in real-world applications *without* large-scale training data.

**Ablation Study** Table 5 evaluates our architectural design choices for omni-scale feature learning. The primary model is model 1. $T$ denotes the stream cardinality in Eq. (2). The results are summarised as follows. **(1)** *vs. standard convolutions:* Overall, factorising convolutions does not significantly harm the performance while has a positive effect on large datasets like MSMT17 (model 2 vs. 1). This means our design helps maintain the representational power while shrinking the model size by more than 3×. **(2)** *vs. ResNeXt-*

TABLE 4
Comparison with deep methods on small re-ID datasets at rank-1.

| Method | Backbone | VIPeR | GRID | CUHK01 |
|---|---|---|---|---|
| MuDeep [47] | Inception | 43.0 | - | - |
| DeepAlign [42] | Inception | 48.7 | - | - |
| JLML [40] | ResNet | 50.2 | 37.5 | 69.8 |
| Spindle [34] | Inception | 53.8 | - | 79.9 |
| GLAD [29] | Inception | 54.8 | - | - |
| HydraPlus-Net [44] | Inception | 56.6 | - | - |
| OSNet (*ours*) | OSNet | **68.0** | **38.2** | **86.6** |

TABLE 5
Ablation study on omni-scale residual learning.

| Model | Architecture | CUHK03 R1 | CUHK03 mAP | MSMT17 R1 | MSMT17 mAP |
|---|---|---|---|---|---|
| 1 | $T = 4$ + unified AG (**primary model**) | 57.1 | 54.2 | 71.0 | 43.3 |
| 2 | $T = 4$ w/ full conv + unified AG | 59.1 | 56.2 | 70.5 | 43.2 |
| 3 | $T = 4$ (same depth) + unified AG | 54.4 | 52.8 | 69.1 | 40.4 |
| 4 | $T = 4$ + concatenation | 52.2 | 50.5 | 66.4 | 37.7 |
| 5 | $T = 4$ + addition | 52.5 | 51.1 | 64.5 | 36.3 |
| 6 | $T = 4$ + separate AGs | 53.6 | 51.0 | 68.3 | 39.4 |
| 7 | $T = 4$ + unified AG (stream-wise) | 54.4 | 51.9 | 67.9 | 40.4 |
| 8 | $T = 4$ + learned-and-fixed gates | 52.9 | 50.8 | 64.5 | 36.2 |
| 9 | $T = 1$ | 43.3 | 42.1 | 52.9 | 26.8 |
| 10 | $T = 2$ + unified AG | 52.2 | 50.0 | 65.6 | 37.0 |
| 11 | $T = 3$ + unified AG | 54.0 | 51.8 | 67.7 | 39.6 |

*like design:* OSNet is transformed into a ResNeXt-like architecture by making all streams homogeneous in depth while preserving the unified AG, which refers to model 3. We observe that this variant is clearly outperformed by the primary model, which further validates the necessity of the omni-scale design. **(3)** *Multi-scale fusion strategy:* We change the way how features of different scales are aggregated. The baselines are concatenation (model 4) and addition (model 5). The primary model is clearly better than the two baselines. Nevertheless, models 4 and 5 are still much better than the single-scale architecture (model 9). **(4)** *Unified AG vs. separate AGs:* When separate AGs are learned for each feature stream, the model size is increased and the nice property in gradient computation (Eq. (4)) vanishes. Empirically, unifying AG improves the performance (model 1 vs. 6), despite having less parameters. **(5)** *Channel-wise gates vs. stream-wise gates:* By turning the channel-wise gates into stream-wise gates (model 7), the performance deteriorates. As feature channels represent numerous visual concepts and encapsulate sophisticated correlations [106], it is advantageous to use channel-specific gates. **(6)** *Dynamic gates vs. static gates:* In model 8, feature streams are fused by static (learned-and-then-fixed) channel-wise gates to mimic the design in [47]. As a result, the performance drops noticeably compared with that of dynamic gating (model 1). Therefore, adapting the scale fusion for individual input images is essential. **(7)** *Evaluation on stream cardinality:* The results are substantially boosted from $T = 1$ (model 9) to $T = 2$ (model 10), and gradually progress to $T = 4$ (model 1).

**Model Shrinking Hyperparameters** We can trade-off between model size, computations and performance by adjusting the width multiplier $\beta$ and the image resolution multiplier $\gamma$. Table 6 shows that by keeping one multiplier fixed and shrinking the other, the R1 drops off *smoothly*. It is worth noting that 92.2% R1 accuracy is obtained by a much shrunken version of OSNet with *merely 0.2M parameters and 82.3M mult-adds* ($\beta = 0.25$). Compared with the results in Table 3, we can see that the shrunken OSNet is still very competitive against the latest models (most being $100\times$ bigger in size). This indicates that OSNet is a superior fit for efficient deployment in resource-constrained devices such as surveillance cameras with AI processors.

**Visualisation of Unified Aggregation Gate** As the gating vectors produced by the AG inherently encode the way how the omni-scale feature streams are aggregated, we can understand what the AG sub-network has learned by visualising images of similar gating vectors. To this end, we 1) concatenate the gating vectors of four streams in the last bottleneck as the representations of test images, 2) perform $k$-means clustering, and 3) select top-15 images closest to the cluster centres. Fig. 6 shows four example clusters where images within the same cluster exhibit similar patterns, i.e. combinations of global-scale and local-scale appearance.

**Visualisation of Learned Features** To understand how our designs help OSNet learn discriminative features, we visualise the activations of the last convolutional feature maps to investigate where the network focuses on to extract features. Following [107], the activation maps are computed as the sum of absolute-valued feature maps along the channel dimension followed by a spatial $\ell_2$ normalisation. Fig. 7 compares the activation maps of OSNet and the single-scale baseline (model 9 in Table 5). It is clear that OSNet can capture the local discriminative patterns of Person A (e.g., the clothing logo) which distinguish Person A from Person B. In contrast, the single-scale model over-concentrates on the face region, which is unreliable for re-ID due to low resolution of surveillance images. This qualitative result shows that our multi-scale design and unified aggregation gate enable OSNet to identify subtle differences between visually similar persons—a vital ability for accurate re-ID.

TABLE 6
Results of varying width multiplier $\beta$ and resolution multiplier $\gamma$ for OSNet. For input size, $\gamma = 0.75$: $192 \times 96$; $\gamma = 0.5$: $128 \times 64$; $\gamma = 0.25$: $64 \times 32$.

| $\beta$ | # params | $\gamma$ | Mult-Adds | Market1501 R1 | Market1501 mAP |
|---|---|---|---|---|---|
| 1.0 | 2.2M | 1.0 | 978.9M | 94.8 | 86.7 |
| 0.75 | 1.3M | 1.0 | 571.8M | 94.5 | 84.1 |
| 0.5 | 0.6M | 1.0 | 272.9M | 93.4 | 82.6 |
| 0.25 | 0.2M | 1.0 | 82.3M | 92.2 | 77.8 |
| 1.0 | 2.2M | 0.75 | 550.7M | 94.4 | 83.7 |
| 1.0 | 2.2M | 0.5 | 244.9M | 92.0 | 80.3 |
| 1.0 | 2.2M | 0.25 | 61.5M | 86.9 | 67.3 |
| 0.75 | 1.3M | 0.75 | 321.7M | 94.3 | 82.4 |
| 0.75 | 1.3M | 0.5 | 143.1M | 92.9 | 79.5 |
| 0.75 | 1.3M | 0.25 | 35.9M | 85.4 | 65.5 |
| 0.5 | 0.6M | 0.75 | 153.6M | 92.9 | 80.8 |
| 0.5 | 0.6M | 0.5 | 68.3M | 91.7 | 78.5 |
| 0.5 | 0.6M | 0.25 | 17.2M | 85.4 | 66.0 |
| 0.25 | 0.2M | 0.75 | 46.3M | 91.6 | 76.1 |
| 0.25 | 0.2M | 0.5 | 20.6M | 88.7 | 71.8 |
| 0.25 | 0.2M | 0.25 | 5.2M | 79.1 | 56.0 |

TABLE 7
Ablation study for instance normalisation and architecture search.

| Method | Market1501→Duke R1 | R5 | R10 | mAP | Duke→Market1501 R1 | R5 | R10 | mAP |
|---|---|---|---|---|---|---|---|---|
| IBN-Net [19] | 43.7 | 59.1 | 65.2 | 24.3 | 50.7 | 69.1 | 76.3 | 23.5 |
| OSNet | 44.7 | 59.6 | 65.4 | 25.9 | 52.2 | 67.5 | 74.7 | 24.0 |
| OSNet-IBN | 47.9 | 62.7 | 68.2 | 27.6 | 57.8 | 74.0 | 79.5 | 27.4 |
| OSNet-AIN | **52.4** | **66.1** | **71.2** | **30.5** | **61.0** | **77.0** | **82.5** | **30.6** |

## 4.2 Cross-Domain Person Re-Identification

In this section, we evaluate the domain-generalisable OSNet with IN, i.e. OSNet-AIN, in the cross-dataset re-ID setting. In particular, we aim to assess the generalisation performance of OSNet-AIN by first training the model on a source dataset and then *directly* testing its performance on an unseen target dataset *without the need for per-domain model adaptation*. This differs significantly from the current state-of-the-art unsupervised domain adaptation (UDA) methods [7], [10], [112], [113], which require per-domain adaptation on the target domain data (hence more computationally expensive and less scalable).

**Architecture Search** We first discuss the experimental details regarding how OSNet-AIN is searched. For the dataset to perform NAS, we choose MSMT17, which contains the largest camera network (15 cameras) and has diverse image qualities/styles (collected in four days of different weather conditions within a month). Once the network architecture

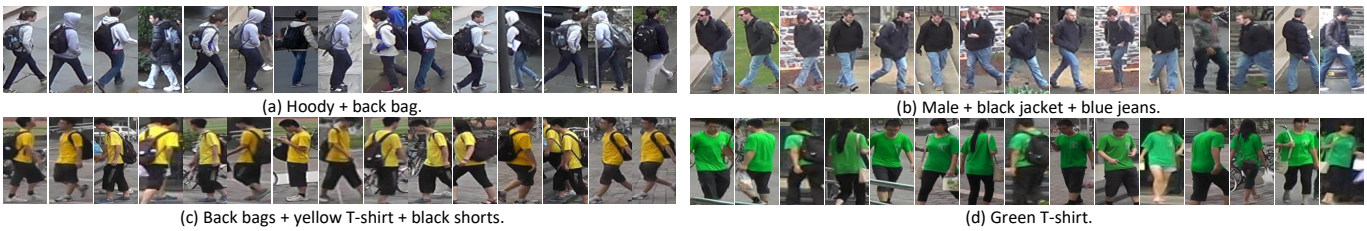


Fig. 6. Image clusters of similar gating vectors. The visualisation shows that the proposed unified aggregation gate is capable of learning the combination of homogeneous and heterogeneous scales conditioned on the input data.

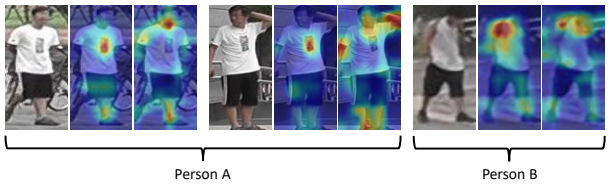


Fig. 7. Visual attention insight. Each triplet contains, from left to right, the original image, activation map of OSNet, and the single-scale baseline. OSNet can detect subtle differences between visually similar persons.

TABLE 8
Performance of OSNet-AIN in the same-domain re-ID setting.

| Method | Market1501 | | Duke | |
|---|---|---|---|---|
| | R1 | mAP | R1 | mAP |
| OSNet | 94.8 | 86.7 | 88.2 | 76.7 |
| OSNet-AIN | 94.2 | 84.4 | 87.9 | 74.2 |

is found, we directly transfer it to other re-ID datasets without re-searching. The over-parameterised network (Fig. 5) is trained from scratch using SGD, batch size of 512, initial learning rate of 0.1 and weight decay of 5e-4 for 120 epochs on 8 Tesla V100 32GB GPUs. The learning rate is annealed down to zero using the cosine annealing trick [103] without restart. The softmax temperature $\lambda$ (Eq. (8)) starts from 10 and decreases by 0.5 every 20 epochs (the minimum is fixed to 1.[7]) Though a larger Monte Carlo sampling number $S$ in Eqs. (12) & (15) is theoretically better for convergence, we empirically found that setting $S = 1$ worked well and greatly shortened the training time. The objective function is the cross-entropy loss with label smoothing. Random flip and colour jittering are used for data augmentation.

**Datasets and Settings** Following the recent UDA re-ID works [7], [9], [10], [112], [113], we experiment with Market1501→Duke, Duke→Market1501, MSMT17→Market1501/Duke, and Market1501/Duke→MSMT17. When the source dataset is MSMT17, we use all 126,441 images of 4,101 identities for model training, following [112], [113]. OSNet-AIN is first pre-trained on ImageNet and then fine-tuned on a source dataset for re-ID on a target dataset. The training pipeline and details for this cross-domain setting follow those used in the same-domain setting, except that the maximum epoch is 100 for Market1501/Duke and 50 for MSMT17 and the data augmentation includes random flip and colour jittering.

**Effect of Instance Normalisation** Table 7 shows that OSNet-AIN significantly improves upon OSNet by 7.7% R1 and 4.6% mAP on Market1501→Duke and 8.8% R1 and 6.6% mAP on Duke→Market1501. This justifies the effectiveness of IN for cross-domain re-ID. Comparing OSNet with IBN-Net, we observe that our basic omni-scale backbone is already stronger than the IN-equipped ResNet50 model. This suggests that our omni-scale network design is not only effective for learning discriminative re-ID features for source datasets, but also helps the learning of generalisable person features for unseen target datasets. This is because omni-scale features capture both global- and local-scale patterns, intuitively a domain-agnostic capability.

Table 8 further tests the model performance effect of IN in the same-domain re-ID setting by comparing OSNet-AIN with OSNet. We observe that IN slightly decreases the performance. This is not surprising: during feature learning IN progressively removes dataset-specific features that are detrimental to cross-domain re-ID but potentially beneficial to same-domain recognition. This is thus a price one has to pay to make the model more generalisable to unseen domains. Note that the performance of OSNet-AIN in the same-domain setting is still very competitive when compared with the state-of-the-art alternatives in Table 3.

**Search vs. Engineering** To justify the contribution of our architecture search algorithm, we hand-engineer an OSNet+IN model by mimicking the design rule in IBN-Net. Specifically, we add IN only to the lowest layers in OSNet (conv1 & conv2 as shown in Table 1), which we call *OSNet-IBN*. Table 7 shows that OSNet-AIN significantly outperforms OSNet-IBN by 4.5% R1 and 2.9% mAP on Market1501→Duke and 3.2% R1 and 3.2% mAP on Duke→Market1501. This strongly demonstrates the superiority of our architecture search algorithm over handcrafted architecture design. Given that the search space for the entire network contains $4^6 = 4096$ different configurations, it is much more efficient to learn the network configuration rather than exhaustively trying all possible choices.

**Comparative Results** Table 9 compares OSNet-AIN with current state-of-the-art cross-domain re-ID methods based on UDA, which obtain unfair gains by using the target data. It is clear that OSNet-AIN achieves promising results on the target datasets despite only using source data—it outperforms most UDA methods in terms of R1. When the source dataset is large such as MSMT17, OSNet-AIN substantially improves the cross-domain performance (R1) from 52.4% to 71.1% on Duke and 61.0% to 70.1% on Market1501, which are close to the performance of the latest UDA methods. It is noted that most top-performing UDA methods (ECN, HHL, CamStyle and ATNet in the small source case) rely on image-to-image translation models such as CycleGAN [114]

7. Setting $\lambda < 1$ makes the training unstable.

TABLE 9
Comparison with current state-of-the-art unsupervised domain adaptation methods in the cross-domain re-ID setting. OSNet-AIN achieves highly comparable performance despite *only using the source training data without per-domain model adaptation*. $U$: Unlabelled.

| Method | Venue | Source | Target: Duke | | | | Source | Target: Market1501 | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | R1 | R5 | R10 | mAP | | R1 | R5 | R10 | mAP |
| MMFA [108] | BMVC'18 | Market1501 + Duke ($U$) | 45.3 | 59.8 | 66.3 | 24.7 | Duke + Market1501 ($U$) | 56.7 | 75.0 | 81.8 | 27.4 |
| SPGAN [109] | CVPR'18 | Market1501 + Duke ($U$) | 46.4 | 62.3 | 68.0 | 26.2 | Duke + Market1501 ($U$) | 57.7 | 75.8 | 82.4 | 26.7 |
| TJ-AIDL [110] | CVPR'18 | Market1501 + Duke ($U$) | 44.3 | 59.6 | 65.0 | 23.0 | Duke + Market1501 ($U$) | 58.2 | 74.8 | 81.1 | 26.5 |
| ATNet [10] | CVPR'19 | Market1501 + Duke ($U$) | 45.1 | 59.5 | 64.2 | 24.9 | Duke + Market1501 ($U$) | 55.7 | 73.2 | 79.4 | 25.6 |
| CamStyle [9] | TIP'19 | Market1501 + Duke ($U$) | 48.4 | 62.5 | 68.9 | 25.1 | Duke + Market1501 ($U$) | 58.8 | 78.2 | 84.3 | 27.4 |
| HHL [8] | ECCV'18 | Market1501 + Duke ($U$) | 46.9 | 61.0 | 66.7 | 27.2 | Duke + Market1501 ($U$) | 62.2 | 78.8 | 84.0 | 31.4 |
| ECN [7] | CVPR'19 | Market1501 + Duke ($U$) | 63.3 | 75.8 | 80.4 | 40.4 | Duke + Market1501 ($U$) | 75.1 | 87.6 | 91.6 | 43.0 |
| SSG [111] | ICCV'19 | Market1501 + Duke ($U$) | 73.0 | 80.6 | 83.2 | 53.4 | Duke + Market1501 ($U$) | 80.0 | 90.0 | 92.4 | 58.3 |
| OSNet-AIN (*ours*) | This work | Market1501 | 52.4 | 66.1 | 71.2 | 30.5 | Duke | 61.0 | 77.0 | 82.5 | 30.6 |
| MAR [112] | CVPR'19 | MSMT17+Duke ($U$) | 67.1 | 79.8 | - | 48.0 | MSMT17+Market1501 ($U$) | 67.7 | 81.9 | - | 40.0 |
| PAUL [113] | CVPR'19 | MSMT17+Duke ($U$) | 72.0 | 82.7 | 86.0 | 53.2 | MSMT17+Market1501 ($U$) | 68.5 | 82.4 | 87.4 | 40.1 |
| OSNet-AIN (*ours*) | This work | MSMT17 | 71.1 | 83.3 | 86.4 | 52.7 | MSMT17 | 70.1 | 84.1 | 88.6 | 43.3 |

TABLE 10
Cross-domain results on the more challenging MSMT17 dataset.

| Method | Source | Target: MSMT17 | | | |
|---|---|---|---|---|---|
| | | R1 | R5 | R10 | mAP |
| ECN [7] | Market1501 + MSMT17 ($U$) | 25.3 | 36.3 | 42.1 | 8.5 |
| SSG [111] | Market1501 + MSMT17 ($U$) | 31.6 | - | 49.6 | 13.2 |
| OSNet-AIN (*ours*) | Market1501 | 23.5 | 34.5 | 40.2 | 8.2 |
| ECN [7] | Duke + MSMT17 ($U$) | 30.2 | 41.5 | 46.8 | 10.2 |
| SSG [111] | Duke + MSMT17 ($U$) | 32.2 | - | 51.2 | 13.3 |
| OSNet-AIN (*ours*) | Duke | 30.3 | 42.2 | 47.9 | 10.2 |

to synthesise target-style images and have complex adaptation procedures for each target domain. These thus severely hinder their deployment in real-world applications where out-of-the-box solution is desired. In contrast, OSNet-AIN enables adaptation-free, plug-and-play deployment once trained on a source dataset. Notably, in the more challenging scenario where the source dataset is small but the target dataset is large, i.e. Market1501→MSMT17, as shown in Table 10, OSNet-AIN can achieve performance on par with ECN—the latter benefits from more unlabelled target data from MSMT17.

To further demonstrate our model's capability, we conduct evaluation on multi-source domain generalisation [115] where a model is trained using multiple source datasets, such as a combination of Market1501, Duke, and CUHK03, and tested on an unseen target dataset, such as MSMT17. Please see Appendix A and Table 11.

## 5 Conclusion

In this paper, we presented OSNet, a lightweight CNN architecture that is capable of learning omni-scale feature representations for person re-ID. Compared with existing re-ID CNNs, OSNet has the unique ability to learn multi-scale features explicitly inside each building block, where the unified aggregation gate dynamically fuses multi-scale features to produce omni-scale features. To improve cross-domain generalisation, we equipped OSNet with instance normalisation via differentiable architecture search, resulting in a domain-adaptive variant called OSNet-AIN. In the same-domain re-ID setting, the results showed that OSNet achieves state-of-the-art performance while being much smaller than ResNet-based opponents. In the cross-domain re-ID setting, OSNet-AIN exhibited a remarkable generalisation ability on unseen target datasets, beating most recent UDA methods without using per-domain model adaptation on target domain data.

## Appendix A
## Multi-Source Domain Generalisation

To further demonstrate the effectiveness of OSNet-AIN, we conduct experiments in the multi-source domain generalisation setting following [115],[8] i.e. a model is trained using multiple source datasets rather than a single dataset. The aim is to show that integrating instance normalisation into OSNet via architecture search is better than hand-engineering.

**Datasets** Following [115], we use the four largest re-ID datasets, namely Market1501, Duke, CUHK03,[9] and MSMT17. In particular, three datasets are used for training and the remaining one is used for testing. During model training, we only use the training split of the source datasets.

**Training Details** All models are trained using the cross-entropy loss with label smoothing. We simply build a single large classification layer for classifying all identities combined from different datasets. We keep the training parameters the same as those used in Sec. 4.2. The details of the training parameters can be found in our Github repository.[10]

**Results** The results are presented in Table 11. Comparing OSNet with OSNet-IBN, we observe that using instance normalisation layers improves the plain OSNet's performance, especially on the most challenging test domain (MSMT17). By automating the architecture engineering process for integrating instance normalisation layers, the performance is further improved (OSNet-AIN vs. OSNet-IBN).

## References

[1] S. Liao, Y. Hu, X. Zhu, and S. Z. Li, "Person re-identification by local maximal occurrence representation and metric learning," in *CVPR*, 2015.

8. https://github.com/HeliosZhao/M3L.
9. The 767/700 split is used for both training and testing.
10. Please see im_osnet_ain_x1_0_softmax_256x128_amsgrad_cosine.yaml. The maximum epoch is set to 50 instead of 100.

TABLE 11
Results in the multi-source domain generalisation setting using MSMT17 (MS), Market1501 (M), Duke (D), and CUHK03 (C).

| Model | MS+D+C→M | | MS+M+C→D | | MS+D+M→C | | D+M+C→MS | |
|---|---|---|---|---|---|---|---|---|
| | mAP | R1 | mAP | R1 | mAP | R1 | mAP | R1 |
| OSNet | 44.2 | 72.5 | 47.0 | 65.2 | 23.3 | 23.9 | 12.6 | 33.2 |
| OSNet-IBN | 44.9 | 73.0 | 45.7 | 64.6 | 25.4 | 25.7 | 16.2 | 39.8 |
| OSNet-AIN | 45.8 | 73.3 | 47.2 | 65.6 | 27.1 | 27.4 | 16.2 | 40.2 |


[2] T. Matsukawa, T. Okabe, E. Suzuki, and Y. Sato, "Hierarchical gaussian descriptor for person re-identification," in *CVPR*, 2016.

[3] W. Li, R. Zhao, T. Xiao, and X. Wang, "Deepreid: Deep filter pairing neural network for person re-identification," in *CVPR*, 2014.

[4] E. Ahmed, M. Jones, and T. K. Marks, "An improved deep learning architecture for person re-identification," in *CVPR*, 2015.

[5] W. Li, X. Zhu, and S. Gong, "Harmonious attention network for person re-identification," in *CVPR*, 2018.

[6] X. Chang, T. M. Hospedales, and T. Xiang, "Multi-level factorisation net for person re-identification," in *CVPR*, 2018.

[7] Z. Zhong, L. Zheng, Z. Luo, S. Li, and Y. Yang, "Invariance matters: Exemplar memory for domain adaptive person re-identification," in *CVPR*, 2019.

[8] Z. Zhong, L. Zheng, S. Li, and Y. Yang, "Generalizing a person retrieval model hetero- and homogeneously," in *ECCV*, 2018.

[9] Z. Zhong, L. Zheng, Z. Zheng, S. Li, and Y. Yang, "Camstyle: A novel data augmentation method for person re-identification," *TIP*, 2019.

[10] J. Liu, Z.-J. Zha, D. Chen, R. Hong, and M. Wang, "Adaptive transfer network for cross-domain person re-identification," in *CVPR*, 2019.

[11] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, "Mobilenets: Efficient convolutional neural networks for mobile vision applications," *arXiv preprint arXiv:1704.04861*, 2017.

[12] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "Mobilenetv2: Inverted residuals and linear bottlenecks," in *CVPR*, 2018.

[13] K. Zhou, Z. Liu, Y. Qiao, T. Xiang, and C. C. Loy, "Domain generalization: A survey," *arXiv preprint arXiv:2103.02503*, 2021.

[14] D. Ulyanov, A. Vedaldi, and V. Lempitsky, "Instance normalization: The missing ingredient for fast stylization," *arXiv preprint arXiv:1607.08022*, 2016.

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

[16] D. Ulyanov, A. Vedaldi, and V. Lempitsky, "Improved texture networks: Maximizing quality and diversity in feed-forward stylization and texture synthesis," in *CVPR*, 2017.

[17] V. Dumoulin, J. Shlens, and M. Kudlur, "A learned representation for artistic style," in *ICLR*, 2017.

[18] X. Huang and S. Belongie, "Arbitrary style transfer in real-time with adaptive instance normalization," in *ICCV*, 2017.

[19] X. Pan, P. Luo, J. Shi, and X. Tang, "Two at once: Enhancing learning and generalization capacities via ibn-net," in *ECCV*, 2018.

[20] H. Nam and H.-E. Kim, "Batch-instance normalization for adaptively style-invariant neural networks," in *NeurIPS*, 2018.

[21] E. Jang, S. Gu, and B. Poole, "Categorical reparameterization with gumbel-softmax," in *ICLR*, 2017.

[22] C. J. Maddison, A. Mnih, and Y. W. Teh, "The concrete distribution: A continuous relaxation of discrete random variables," in *ICLR*, 2017.

[23] Y. Sun, L. Zheng, Y. Yang, Q. Tian, and S. Wang, "Beyond part models: Person retrieval with refined part pooling (and a strong convolutional baseline)," in *ECCV*, 2018.

[24] J. Si, H. Zhang, C.-G. Li, J. Kuen, X. Kong, A. C. Kot, and G. Wang, "Dual attention matching network for context-aware feature sequence based person re-identification," in *CVPR*, 2018.

[25] R. Hou, B. Ma, H. Chang, X. Gu, S. Shan, and X. Chen, "Interaction-and-aggregation network for person re-identification," in *CVPR*, 2019.

[26] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *CVPR*, 2016.

[27] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *CVPR*, 2015.

[28] G. Huang, Z. Liu, K. Q. Weinberger, and L. van der Maaten, "Densely connected convolutional networks," in *CVPR*, 2017.

[29] L. Wei, S. Zhang, H. Yao, W. Gao, and Q. Tian, "Glad: global-local-alignment descriptor for pedestrian retrieval," in *ACM MM*, 2017.

[30] Y. Fu, Y. Wei, Y. Zhou, H. Shi, G. Huang, X. Wang, Z. Yao, and T. Huang, "Horizontal pyramid matching for person re-identification," in *AAAI*, 2019.

[31] Y. Shen, T. Xiao, H. Li, S. Yi, and X. Wang, "End-to-end deep kronecker-product matching for person re-identification," in *CVPR*, 2018.

[32] G. Wang, Y. Yuan, X. Chen, J. Li, and X. Zhou, "Learning discriminative features with multiple granularities for person re-identification," in *ACM MM*, 2018.

[33] C. Song, Y. Huang, W. Ouyang, and L. Wang, "Mask-guided contrastive attention model for person re-identification," in *CVPR*, 2018.

[34] H. Zhao, M. Tian, S. Sun, J. Shao, J. Yan, S. Yi, X. Wang, and X. Tang, "Spindle net: Person re-identification with human body region guided feature decomposition and fusion," in *CVPR*, 2017.

[35] C. Su, J. Li, S. Zhang, J. Xing, W. Gao, and Q. Tian, "Pose-driven deep convolutional model for person re-identification," in *ICCV*, 2017.

[36] J. Xu, R. Zhao, F. Zhu, H. Wang, and W. Ouyang, "Attention-aware compositional network for person re-identification," in *CVPR*, 2018.

[37] Y. Suh, J. Wang, S. Tang, T. Mei, and K. M. Lee, "Part-aligned bilinear representations for person re-identification," in *ECCV*, 2018.

[38] M. Tian, S. Yi, H. Li, S. Li, X. Zhang, J. Shi, J. Yan, and X. Wang, "Eliminating background-bias for robust person re-identification," in *CVPR*, 2018.

[39] Z. Zhang, C. Lan, W. Zeng, and Z. Chen, "Densely semantically aligned person re-identification," in *CVPR*, 2019.

[40] W. Li, X. Zhu, and S. Gong, "Person re-identification by deep joint learning of multi-loss classification," in *IJCAI*, 2017.

[41] D. Li, X. Chen, Z. Zhang, and K. Huang, "Learning deep context-aware features over body and latent parts for person re-identification," in *CVPR*, 2017.

[42] L. Zhao, X. Li, Y. Zhuang, and J. Wang, "Deeply-learned part-aligned representations for person re-identification." in *ICCV*,, 2017.

[43] Q. Yu, X. Chang, Y.-Z. Song, T. Xiang, and T. M. Hospedales, "The devil is in the middle: Exploiting mid-level representations for cross-domain instance matching," *arXiv preprint arXiv:1711.08106*, 2017.

[44] X. Liu, H. Zhao, M. Tian, L. Sheng, J. Shao, S. Yi, J. Yan, and X. Wang, "Hydraplus-net: Attentive deep features for pedestrian analysis," in *ICCV*, 2017.

[45] Y. Wang, L. Wang, Y. You, X. Zou, V. Chen, S. Li, G. Huang, B. Hariharan, and K. Q. Weinberger, "Resource aware person re-identification across multiple resolutions," in *CVPR*, 2018.

[46] Y. Chen, X. Zhu, and S. Gong, "Person re-identification by deep learning multi-scale representations," in *ICCVW*, 2017.

[47] X. Qian, Y. Fu, Y.-G. Jiang, T. Xiang, and X. Xue, "Multi-scale deep learning architectures for person re-identification," in *ICCV*, 2017.

[48] S. Xie, R. Girshick, P. Dollár, Z. Tu, and K. He, "Aggregated residual transformations for deep neural networks," in *CVPR*, 2017.

[49] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *CVPR*, 2016.

[50] F. N. Iandola, S. Han, M. W. Moskewicz, K. Ashraf, W. J. Dally, and K. Keutzer, "Squeezenet: Alexnet-level accuracy with 50x fewer parameters and¡ 0.5 mb model size," *arXiv preprint arXiv:1602.07360*, 2016.

[51] T. Zhang, G.-J. Qi, B. Xiao, and J. Wang, "Interleaved group convolutions," in *ICCV*, 2017.

[52] G. Huang, S. Liu, L. van der Maaten, and K. Q. Weinberger, "Condensenet: An efficient densenet using learned group convolutions," in *CVPR*, 2018.

[53] F. Chollet, "Xception: Deep learning with depthwise separable convolutions," in *CVPR*, 2017.

[54] X. Zhang, X. Zhou, M. Lin, and J. Sun, "Shufflenet: An extremely efficient convolutional neural network for mobile devices," in *CVPR*, 2018.

[55] J. Song, Y. Yang, Y.-Z. Song, T. Xiang, and T. M. Hospedales, "Generalizable person re-identification by domain-invariant mapping network," in *CVPR*, 2019.

[56] K. Zhou, Y. Yang, Y. Qiao, and T. Xiang, "Domain adaptive ensemble learning," *arXiv preprint arXiv:2003.07325*, 2020.

[57] D. Li, Y. Yang, Y.-Z. Song, and T. M. Hospedales, "Deeper, broader and artier domain generalization," in *ICCV*, 2017.

[58] F. M. Carlucci, A. D'Innocente, S. Bucci, B. Caputo, and T. Tommasi, "Domain generalization by solving jigsaw puzzles," in *CVPR*, 2019.

[59] D. Li, J. Zhang, Y. Yang, C. Liu, Y.-Z. Song, and T. M. Hospedales, "Episodic training for domain generalization," in *ICCV*, 2019.

[60] K. Zhou, Y. Yang, T. M. Hospedales, and T. Xiang, "Deep domain-adversarial image generation for domain generalisation." in *AAAI*, 2020.

[61] K. Zhou, Y. Yang, T. Hospedales, and T. Xiang, "Learning to generate novel domains for domain generalization," in *ECCV*, 2020.

[62] K. Zhou, Y. Yang, Y. Qiao, and T. Xiang, "Domain generalization with mixstyle," in *ICLR*, 2021.

[63] Y. Balaji, S. Sankaranarayanan, and R. Chellappa, "Metareg: Towards domain generalization using meta-regularization," in *NeurIPS*, 2018.

[64] O. Vinyals, C. Blundell, T. Lillicrap, D. Wierstra *et al.*, "Matching networks for one shot learning," in *NeurIPS*, 2016.

[65] B. Zoph and Q. V. Le, "Neural architecture search with reinforcement learning," in *ICLR*, 2017.

[66] B. Zoph, V. Vasudevan, J. Shlens, and Q. V. Le, "Learning transferable architectures for scalable image recognition," in *CVPR*, 2018.

[67] E. Real, S. Moore, A. Selle, S. Saxena, Y. L. Suematsu, J. Tan, Q. V. Le, and A. Kurakin, "Large-scale evolution of image classifiers," in *ICML*, 2017.

[68] E. Real, A. Aggarwal, Y. Huang, and Q. V. Le, "Regularized evolution for image classifier architecture search," in *AAAI*, 2019.

[69] G. Bender, P.-J. Kindermans, B. Zoph, V. Vasudevan, and Q. Le, "Understanding and simplifying one-shot architecture search," in *ICML*, 2018.

[70] H. Pham, M. Y. Guan, B. Zoph, Q. V. Le, and J. Dean, "Efficient neural architecture search via parameter sharing," in *ICML*, 2018.

[71] H. Liu, K. Simonyan, and Y. Yang, "Darts: Differentiable architecture search," in *ICLR*, 2019.

[72] S. Xie, H. Zheng, C. Liu, and L. Lin, "Snas: stochastic neural architecture search," in *ICLR*, 2019.

[73] X. Dong and Y. Yang, "Searching for a robust neural architecture in four gpu hours," in *CVPR*, 2019.

[74] H. Cai, L. Zhu, and S. Han, "Proxylessnas: Direct neural architecture search on target task and hardware," in *ICLR*, 2019.

[75] K. Zhou, Y. Yang, A. Cavallaro, and T. Xiang, "Omni-scale feature learning for person re-identification," in *ICCV*, 2019.

[76] M. Lin, Q. Chen, and S. Yan, "Network in network," in *ICLR*, 2014.

[77] S. Woo, J. Park, J.-Y. Lee, and I. So Kweon, "Cbam: Convolutional block attention module," in *ECCV*, 2018.

[78] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *CVPR*, 2018.

[79] K. He, X. Zhang, S. Ren, and J. Sun, "Identity mappings in deep residual networks," in *ECCV*, 2016.

[80] S. Mohamed, M. Rosca, M. Figurnov, and A. Mnih, "Monte carlo gradient estimation in machine learning," *arXiv preprint arXiv:1906.10652*, 2019.

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

[82] E. Ristani, F. Solera, R. Zou, R. Cucchiara, and C. Tomasi, "Performance measures and a data set for multi-target, multi-camera tracking," in *ECCV-W*, 2016.

[83] Z. Zheng, L. Zheng, and Y. Yang, "Unlabeled samples generated by gan improve the person re-identification baseline in vitro," in *ICCV*, 2017.

[84] L. Wei, S. Zhang, W. Gao, and Q. Tian, "Person transfer gan to bridge domain gap for person re-identification," in *CVPR*, 2018.

[85] D. Gray, S. Brennan, and H. Tao, "Evaluating appearance models for recognition, reacquisition, and tracking," in *PETS*, 2007.

[86] C. C. Loy, T. Xiang, and S. Gong, "Multi-camera activity correlation analysis," in *CVPR*, 2009.

[87] W. Li, R. Zhao, and X. Wang, "Human reidentification with transferred metric learning," in *ACCV*, 2012.

[88] Z. Zhong, L. Zheng, D. Cao, and S. Li, "Re-ranking person re-identification with k-reciprocal encoding," in *CVPR*, 2017.

[89] Y. Wang, Z. Chen, F. Wu, and G. Wang, "Person re-identification with cascaded pairwise convolutions," in *CVPR*, 2018.

[90] R. Quan, X. Dong, Y. Wu, L. Zhu, and Y. Yang, "Auto-reid: Searching for a part-aware convnet for person re-identification," in *ICCV*, 2019.

[91] Y. Sun, L. Zheng, W. Deng, and S. Wang, "Svdnet for pedestrian retrieval," in *ICCV*, 2017.

[92] R. Yu, Z. Dou, S. Bai, Z. Zhang, Y. Xu, and X. Bai, "Hard-aware point-to-set deep metric for person re-identification," in *ECCV*, 2018.

[93] X. Qian, Y. Fu, T. Xiang, W. Wang, J. Qiu, Y. Wu, Y.-G. Jiang, and X. Xue, "Pose-normalized image generation for person re-identification," in *ECCV*, 2018.

[94] Y. Ge, Z. Li, H. Zhao, G. Yin, S. Yi, X. Wang, and H. Li, "Fd-gan: Pose-guided feature distilling gan for robust person re-identification," in *NeurIPS*, 2018.

[95] Y. Shen, H. Li, T. Xiao, S. Yi, D. Chen, and X. Wang, "Deep group-shuffling random walk for person re-identification," in *CVPR*, 2018.

[96] D. Chen, D. Xu, H. Li, N. Sebe, and X. Wang, "Group consistent similarity learning via deep crf for person re-identification," in *CVPR*, 2018.

[97] Y. Shen, H. Li, S. Yi, D. Chen, and X. Wang, "Person re-identification with deep similarity-guided graph neural network," in *ECCV*, 2018.

[98] C. Wang, Q. Zhang, C. Huang, W. Liu, and X. Wang, "Mancs: A multi-task attentional network with curriculum sampling for person re-identification," in *ECCV*, 2018.

[99] C.-P. Tay, S. Roy, and K.-H. Yap, "Aanet: Attribute attention network for person re-identifications," in *CVPR*, 2019.

[100] W. Yang, H. Huang, Z. Zhang, X. Chen, K. Huang, and S. Zhang, "Towards rich feature discovery with class activation maps augmentation for person re-identification," in *CVPR*, 2019.

[101] Z. Zheng, X. Yang, Z. Yu, L. Zheng, Y. Yang, and J. Kautz, "Joint discriminative and generative learning for person re-identification," in *CVPR*, 2019.

[102] S. J. Reddi, S. Kale, and S. Kumar, "On the convergence of adam and beyond," in *ICLR*, 2018.

[103] I. Loshchilov and F. Hutter, "Sgdr: Stochastic gradient descent with warm restarts," in *ICLR*, 2017.

[104] M. Geng, Y. Wang, T. Xiang, and Y. Tian, "Deep transfer learning for person re-identification," *arXiv preprint arXiv:1611.05244*, 2016.

[105] Z. Zhong, L. Zheng, G. Kang, S. Li, and Y. Yang, "Random erasing data augmentation," *arXiv preprint arXiv:1708.04896*, 2017.

[106] R. Fong and A. Vedaldi, "Net2vec: Quantifying and explaining how concepts are encoded by filters in deep neural networks," in *CVPR*, 2018.

[107] S. Zagoruyko and N. Komodakis, "Paying more attention to attention: Improving the performance of convolutional neural networks via attention transfer," in *ICLR*, 2017.

[108] S. Lin, H. Li, C.-T. Li, and A. C. Kot, "Multi-task mid-level feature alignment network for unsupervised cross-dataset person re-identification," in *BMVC*, 2018.

[109] W. Deng, L. Zheng, Q. Ye, G. Kang, Y. Yang, and J. Jiao, "Image-image domain adaptation with preserved self-similarity and domain-dissimilarity for person re-identification," in *CVPR*, 2018.

[110] J. Wang, X. Zhu, S. Gong, and W. Li, "Transferable joint attribute-identity deep learning for unsupervised person re-identification," in *CVPR*, 2018.

[111] Y. Fu, Y. Wei, G. Wang, Y. Zhou, H. Shi, and T. S. Huang, "Self-similarity grouping: A simple unsupervised cross domain adaptation approach for person re-identification," in *ICCV*, 2019.
[112] H.-X. Yu, W.-S. Zheng, A. Wu, X. Guo, S. Gong, and J.-H. Lai, "Unsupervised person re-identification by soft multilabel learning," in *CVPR*, 2019.
[113] Q. Yang, H.-X. Yu, A. Wu, and W.-S. Zheng, "Patch-based discriminative feature learning for unsupervised person re-identification," in *CVPR*, 2019.
[114] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networkss," in *ICCV*, 2017.
[115] Y. Zhao, Z. Zhong, F. Yang, Z. Luo, Y. Lin, S. Li, and N. Sebe, "Learning to generalize unseen domains via memory-based multi-source meta-learning for person re-identification," in *CVPR*, 2021.