Omni-Scale Feature Learning for Person Re-Identification

Kaiyang Zhou[1*] Yongxin Yang[1] Andrea Cavallaro[2] Tao Xiang[1,3]

[1]University of Surrey [2]Queen Mary University of London
[3]Samsung AI Center, Cambridge

{k.zhou, yongxin.yang, t.xiang}@surrey.ac.uk a.cavallaro@qmul.ac.uk

Abstract

As an instance-level recognition problem, person re-identification (ReID) relies on discriminative features, which not only capture different spatial scales but also encapsulate an arbitrary combination of multiple scales. We call these features of both homogeneous and heterogeneous scales omni-scale features. In this paper, a novel deep CNN is designed, termed Omni-Scale Network (OSNet), for omni-scale feature learning in ReID. This is achieved by designing a residual block composed of multiple convolutional feature streams, each detecting features at a certain scale. Importantly, a novel unified aggregation gate is introduced to dynamically fuse multi-scale features with input-dependent channel-wise weights. To efficiently learn spatial-channel correlations and avoid overfitting, the building block uses both pointwise and depthwise convolutions. By stacking such blocks layer-by-layer, our OSNet is extremely lightweight and can be trained from scratch on existing ReID benchmarks. Despite its small model size, our OSNet achieves state-of-the-art performance on six person-ReID datasets.

1. Introduction

Person re-identification (ReID), a fundamental task in distributed multi-camera surveillance, aims to match people appearing in different non-overlapping camera views. As an instance-level recognition problem, person ReID faces two major challenges as illustrated in Fig. 1. First, the intra-class (instance/identity) variations are typically big due to the changes of camera viewing conditions. For instance, both people in Figs. 1(a) and (b) carry a backpack; the view change across cameras (frontal to back) brings large appearance changes in the backpack area, making matching the same person difficult. Second, there are also small inter-class variations – people in public space often wear similar clothes; from a distance as typically in surveillance videos, they can look incredibly similar (see the impostors for all four people in Fig. 1).

Figure 1. Person ReID is a hard problem, as exemplified by the four triplets of images above. Each sub-figure shows, from left to right, the query image, a true match and an impostor/false match.

To overcome these two challenges, key to ReID is to learn discriminative features. We argue that such features need to be of *omni-scales*, defined as a combination of variable homogeneous scales and heterogeneous scales, each of which is composed of a mixture of multiple scales. The need for omni-scale features is evident from Fig. 1. To match people and distinguish them from impostors, features corresponding both small local regions (e.g. shoes, glasses), and global whole body regions are important. For example, 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) would narrow down the search to the true match (middle) and an impostor (right). Now the local-scale features come into play – the shoe region gives away the fact that the person on the right is an impostor (trainers vs. sandals). However, for more challenging cases, even features of variable homogeneous scales would not be enough. More complicated and richer features that span multiple scales are required. For instance, to eliminate the impostor in Fig. 1(d) (right), one needs 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. Similarly, the white T-shirt is everywhere in summer (e.g. Fig. 1(a)). It is the unique combination, captured by

*Work done as an intern at Samsung AI Center, Cambridge.

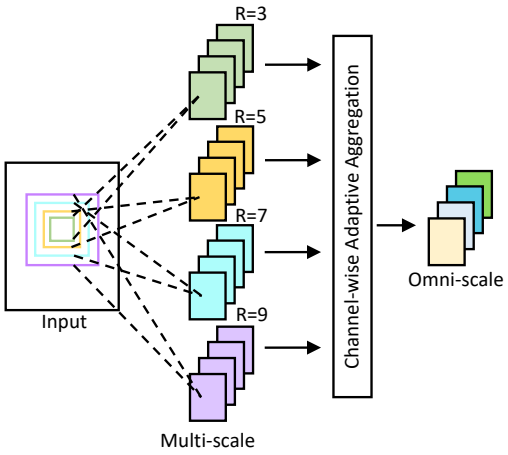

Figure 2. A schematic of the proposed building block for OSNet. R: Receptive field size.

heterogeneous features spanning both small (logo size) and medium (upper body size) scales, that makes the features most effective.

Nevertheless, none of the existing ReID models addresses omni-scale feature learning. In recent years, deep convolutional neural networks (CNNs) have been widely used in person ReID to learn discriminative features [1, 2, 3, 4, 5, 6]. However, most of the CNNs adopted, such as ResNet [7], were originally designed for object category-level recognition tasks that are fundamentally different from the instance-level recognition task in ReID. For the latter, omni-scale features are more important, as explained earlier. A few attempts at learning multi-scale features also exist [8, 1]. Yet, none has the ability to learn features of both homogeneous and heterogeneous scales.

In this paper, we present *OSNet*, a novel CNN architecture designed for learning omni-scale feature representations[1]. The underpinning building block consists of multiple convolutional feature streams with different receptive fields (see Fig. 2). The feature scale that each stream focuses on is determined by *exponent*, a new dimension factor that is linearly increased across streams to ensure that various scales are captured in each block. Critically, the resulting multi-scale feature maps are dynamically fused by channel-wise weights that are generated by a unified aggregation gate (AG). The AG is a sub-network sharing parameters across all streams with a number of desirable properties for effective model training. With the trainable AG, the generated channel-wise weights become input-dependent, hence the dynamic scale fusion. This novel AG design allows the network to learn omni-scale feature representations: depending on the specific input image, the gate could focus on a single scale by assigning a dominant weight to a particular stream or scale; alternatively, it can pick and mix and thus produce heterogeneous scales.

[1]We use scale and receptive field interchangeably.

Apart from enabling omni-scale feature learning, another key design principle adopted in OSNet is to design a *lightweight* network. This brings a couple of benefits: (1) ReID datasets are often of moderate sizes due to the difficulties in collecting across-camera matched person images. A lightweight network with a small number of model parameters is thus less prone to overfitting. (2) In a large-scale surveillance application (e.g. city-wide surveillance using thousands of cameras), the only practical way for ReID is to perform feature extraction at the camera end. Instead of sending the raw videos to a central server, only features need to be sent. For on-device processing, small ReID networks are clearly preferred. To this end, in our building block, we factorise standard convolutions with pointwise and depthwise convolutions [9, 10]. The **contributions** of this work are thus both the concept of omni-scale feature learning and an effective and efficient implementation of it in OSNet[2]. The end result is a lightweight ReID model that is about one order of magnitude smaller than the popular ResNet50-based ones, but performs better: OSNet achieves state-of-the-art performance on six person ReID datasets, beating much larger existing networks, often by a clear margin. We also demonstrate the effectiveness of OSNet on object category recognition tasks, namely CIFAR [11] and ImageNet [12], and multi-label person attribute recognition tasks. The results suggest that omni-scale feature learning is useful beyond instance recognition and can be considered for a broad range of visual recognition tasks.

2. Related Work

Deep ReID Architectures Most existing deep ReID CNNs [13, 14, 15, 16, 17, 18, 19] borrow architectures designed for generic object categorisation problems, such as ImageNet 1K object classification. Recently, some architectural modifications are introduced to reflect the fact that images in ReID datasets contain instances of only one object category (i.e., person) that mostly stand upright. To exploit the upright body pose, [5, 20, 21, 22] add auxiliary supervision signals to features pooled horizontally from the last convolutional feature maps. [4, 23, 2] devise attention mechanisms to focus feature learning on the foreground person regions. In [24, 25, 6, 26, 27], body part-specific CNNs are learned by means of off-the-shelf pose detectors. In [28, 29, 30], CNNs are branched to learn representations of global and local image regions. In [31, 1, 3, 32], multi-level features extracted at different layers are combined. However, *none* of these ReID networks learns multi-scale features explicitly at each layer of the networks as in our OSNet – they typically rely on an external pose model and/or hand-pick specific layers for multi-scale learning. Moreover, heterogeneous-scale features computed from a mixture of different scales are not considered.

[2]Code and models will be released.

Multi-Scale and Multi-Stream Deep Feature Learning 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. Compared to a number of ReID networks with multi-stream building blocks [1, 8], OSNet is significantly different. Specifically the layer design in [1] is based on ResNeXt [33], where each stream learns features at the same scale, while our streams in each block have different scales. Different to [1], the network in [8] is built on Inception [34, 35], where multiple streams were originally designed for low computational cost with handcrafted mixture of convolution and pooling layers. In contrast, our building block strictly follows a scale-incremental pattern to capture a wide range of spatial scales. Moreover, [8] fuses multi-stream features with learnable but fixed-once-learned streamwise weights only after the final block. In contrast, 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 discriminative features of either a single scale or a weighted mixture of multiple scales. Our experiments (see Sec. 4.1) show that OSNet significantly outperforms the models in [1, 8].

Lightweight Network Designs With embedded AI becoming topical, lightweight CNN design has attracted increasing attention. SqueezeNet [36] compresses feature dimensions using 1×1 convolutions. IGCNet [37], ResNeXt [33] and CondenseNet [38] leverage group convolutions. Xception [39] and MobileNet series [9, 10] are based on depthwise separable convolutions. Dense 1×1 convolutions are grouped with channel shuffling in ShuffleNet [40]. In terms of lightweight design, our OSNet is similar to MobileNet by employing factorised convolutions, with some modifications that empirically work better for omni-scale feature learning.

3. Omni-Scale Feature Learning

In this section, we present OSNet, which specialises in learning omni-scale feature representations for the person ReID task. We start with the factorised convolutional layer and then introduce the omni-scale residual block and the unified aggregation gate.

Factorised Convolutions To reduce the number of parameters, we adopt the depthwise separable convolutions [9, 39], which split the standard convolutions into two separate layers: pointwise convolutions and depthwise convolutions. Standard convolutions are parameterised by a 4D tensor $\boldsymbol{w} \in \mathbb{R}^{k \times k \times c \times c'}$, where k is the kernel size, c is the depth of the input channel, and c' is the depth of the output channel. To learn spatial-channel correlations on an input tensor $\boldsymbol{x} \in \mathbb{R}^{h \times w \times c}$, where h is the height and w is the width, the convolution operation can be formulated as $\boldsymbol{x}' = \phi(\boldsymbol{w} * \boldsymbol{x})$, where ϕ is a nonlinear mapping (ReLU) and $*$ denotes convolutions. Biases are omitted for clarity. Fig. 3(a) depicts the practical implementation of a standard 3×3 convolutional layer.

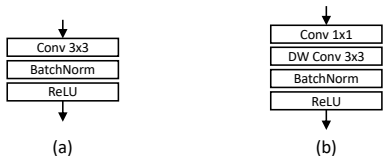

Figure 3. (a) Standard 3×3 convolution. (b) *Lite* 3×3 convolution. DW: Depth-Wise.

Let $\boldsymbol{u} \in \mathbb{R}^{1 \times 1 \times c \times c'}$ be a pointwise convolutional kernel, which densely connects to the channel dimension, and $\boldsymbol{v} \in \mathbb{R}^{k \times k \times 1 \times c'}$ be a depthwise convolutional kernel, which aggregates local information with receptive field k on each feature map. We disentangle the learning of spatial-channel correlations by decomposing $\boldsymbol{w}$ to $\boldsymbol{v} \circ \boldsymbol{u}$, leading to $\boldsymbol{x}' = \phi((\boldsymbol{v} \circ \boldsymbol{u}) * \boldsymbol{x})$, which is illustrated in Fig. 3(b). As a result, 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 number of parameters from $k^2 \cdot c \cdot c'$ to $(k^2 + c) \cdot c'$. As we factorise 3×3 convolutions, we refer such layers to *Lite* 3×3[3].

Omni-Scale Residual Block The building block in our architecture is the residual bottleneck [7], equipped with our Lite 3×3 layer (see Fig. 4(a)). Given an input $\boldsymbol{x}$, this bottleneck aims to learn a residual $\tilde{\boldsymbol{x}}$ with a mapping function F, i.e.

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

where F represents a Lite 3×3 layer that learns single-scale features (scale = 3). Note that here the 1×1 layers are ignored in notation as they are used to manipulate feature dimension and do not contribute to the aggregation of spatial information [7, 33].

To achieve omni-scale representation learning, we extend the residual function F by introducing a new dimension, *exponent* t, which represents the scale of the feature. For F^t, with $t > 1$, we stack t Lite 3×3 layers, and this results 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:

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

When $T = 1$, Eq. 2 reduces to Eq. 1. In this paper, our bottleneck is set with $T = 4$ (i.e. the largest receptive field

[3]Note that our implementation is different from the original depthwise separable convolutions [39], which applies depthwise convolutions before pointwise convolutions. Empirically, we found that our design (pointwise→depthwise) is more effective for omni-scale feature learning, compared to the original version (depthwise→pointwise).

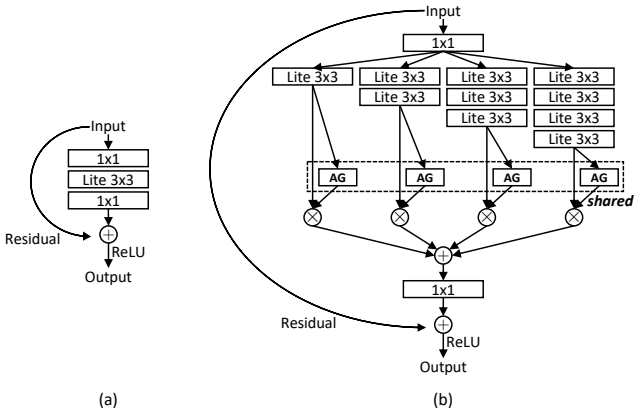

Figure 4. (a) Baseline bottleneck. (b) Proposed bottleneck. AG: Aggregation Gate. The first/last 1×1 layers are used to reduce/restore feature dimension.

stage	output	OSNet
conv1	128×64, 64 64×32, 64	7×7 conv, stride 2 3×3 max pool, stride 2
conv2	64×32, 256	bottleneck × 2
transition	64×32, 256 32×16, 256	1×1 conv 2×2 average pool, stride 2
conv3	32×16, 384	bottleneck × 2
transition	32×16, 384 16×8, 384	1×1 conv 2×2 average pool, stride 2
conv4	16×8, 512	bottleneck × 2
conv5	16×8, 512	1×1 conv
gap	1×1, 512	global average pool
fc	1×1, 512	fc
# params		2.2M
Mult-Adds		978.9M

Table 1. Architecture of OSNet with input image size 256×128.

is 9×9) as shown in Fig. 4(b). The shortcut connection allows features at smaller scales learned in the current layer to be preserved effectively in the next layers, thus enabling the final features to capture a whole range of spatial scales.

Unified Aggregation Gate So far, each stream can give us features of a specific scale, i.e., they are scale homogeneous. To learn 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 after training. More specifically, the dynamic scale-fusion is achieved by a novel aggregation gate (AG), which is a *learnable neural network*.

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

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

where $G(\boldsymbol{x}^t)$ is a vector with length spanning the entire channel dimension of $\boldsymbol{x}^t$ and $\odot$ denotes the Hadamard product. G is implemented as a mini-network composed of a non-parametric global average pooling layer [41] and a multi-layer perceptron (MLP) with one ReLU-activated hidden layer, followed by the sigmoid activation. To reduce parameter overhead, we follow [42, 43] to reduce the hidden dimension of the MLP with a reduction ratio, which is set to 16.

It is worth pointing out that, in contrast to using a single scalar-output function that provides a coarse scale-fusion, we choose to use channel-wise weights, i.e., the output of the AG network $G(\boldsymbol{x}^t)$ 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. In addition, the weights are dynamically computed by being conditioned on the input data. This is crucial for ReID as the test images contain people of different identities from those in training; thus an adaptive/input-dependent feature-scale fusion strategy is more desirable.

Note that in our architecture, the AG is *shared* for all feature streams in the same omni-scale residual block (dashed box in Fig. 4(b)). This is similar in spirit to the convolution filter parameter sharing in CNNs, resulting in a number of advantages. First, the number of parameters is independent of T (number of streams), thus the model becomes more scalable. Second, unifying AG (sharing the same AG module across streams) has a nice property while performing backpropagation. Concretely, suppose the network is supervised by a loss function $\mathcal{L}$ which is differentiable 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

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

The second term in Eq. 4 indicates that the 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.

Network Architecture OSNet is constructed by simply stacking the proposed lightweight bottleneck layer-by-layer without any effort to customise the blocks at different depths (stages) of the network. The detailed network architecture is shown in Table 1. For comparison, the same network architecture with standard convolutions has 6.9 million parameters and 3,384.9 million mult-add operations, which are $3\times$ larger than our OSNet with the Lite 3×3 convolution layer design. The standard OSNet 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 [9, 10, 40].

Relation to Prior Architectures In terms of multi-stream design, OSNet is related to Inception [34] and

[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.

ResNeXt [33], but has crucial differences in several aspects. First, the multi-stream design in OSNet strictly follows the scale-incremental principle dictated by the exponent (Eq. 2). Specifically, different streams have different receptive fields but are built with the same Lite 3×3 layers (Fig. 4(b)). Such a design is more effective at capturing a wide range of scales. In contrast, Inception was originally designed to have low computational costs by sharing computations with multiple streams. Therefore its structure, which includes mixed operations of convolution and pooling, was handcrafted. ResNeXt has multiple equal-scale streams thus learning representations at the same scale. Second, Inception/ResNeXt aggregates features by concatenation/addition while OSNet uses a unified AG (Eq. 3), which facilitates the learning of combinations of multi-scale features. Critically, it means that the fusion is dynamic and adaptive to each individual input image. Therefore, OSNet's architecture is fundamentally different from that of Inception/ResNeXt in nature. Third, OSNet uses factorised convolutions and thus the building block and subsequently the whole network is lightweight. Compared with SENet [43], OSNet is conceptually different. Concretely, SENet aims to re-calibrate the feature channels by re-scaling the activation values for a single stream, whereas OSNet is designed to selectively fuse multiple feature streams of different receptive field sizes in order to learn omni-scale features.

4. Experiments

4.1. Evaluation on Person Re-Identification

Datasets and Settings We conduct experiments on six widely used person ReID datasets: Market1501 [44], CUHK03 [13], DukeMTMC-reID (Duke) [45, 46], MSMT17 [47], VIPeR [48] and GRID [49]. Detailed dataset statistics are provided in Table 2. The first four are considered as 'big' datasets even though their sizes (around 30K training images for the largest MSMT17) are fairly moderate; while VIPeR and GRID are generally too small to train without using those big datasets for pretraining. For CUHK03, we use the 767/700 split [50] with the detected images. For VIPeR and GRID, we first train a single OSNet from scratch using training images from Market1501, CUHK03, Duke and MSMT17 (Mix4), and then perform fine-tuning. Following [28], the results on VIPeR and GRID are averaged over 10 random splits. Such a fine-tuning strategy has been commonly adopted by other deep learning approaches [3, 51, 24, 28, 30]. Cumulative matching characteristics (CMC) Rank-1 accuracy and mAP are used as evaluation metrics.

Data Augmentation Images are resized to 256×128. Three data augmentation techniques are used: (1) random 256×128 crops on images rescaled by a factor of 1.25; (2) random horizontal flip; (3) random erasing [60].

Dataset	# IDs (T-Q-G)	# images (T-Q-G)
Market1501	751-750-751	12936-3368-15913
CUHK03	767-700-700	7365-1400-5332
Duke	702-702-1110	16522-2228-17661
MSMT17	1041-3060-3060	30248-11659-82161
VIPeR	316-316-316	632-632-632
GRID	125-125-900	250-125-900

Table 2. Dataset statistics. T: Train. Q: Query. G: Gallery.

Implementation details A classification layer (linear FC + softmax) is mounted on the top of OSNet. Training follows the standard classification paradigm where each person identity is regarded as a unique class. Similar to [2, 1], cross entropy loss with the label smoothing regulariser [35] is used for supervision. For fair comparison against existing models, we implement two versions of OSNet. One is trained from scratch and the other is fine-tuned from ImageNet-pretrained weights. Person matching is based on the ℓ_2 distance of 512-D feature vectors extracted from the last FC layer (see Table 1). Batch size and weight decay are set to 64 and 5e-4 respectively. For training from scratch, the total number of epochs is 350 and the learning rate starts from 0.065 and is decayed by 0.1 at 150, 225 and 300 epochs. For fine-tuning, we first train the randomly initialised classifier (freezing lower layers) for 10 epochs with a learning rate of 0.00065 and then open all layers to continue training for 150 epochs where the learning rate is decayed by 0.1 at 100 epochs.

Results on Big Datasets From Table 3, we have the following observations. (1) OSNet outperforms all compared methods on each of the four datasets. Specifically, on either Rank-1 (R1) accuracy or mAP, the margins obtained by OSNet over the second-best methods are around 2% on Market1501, 7% on CUHK03 and 4% on Duke. These improvements are significant – from Table 3, it is evident that the performance on ReID benchmarks (especially Market1501 and Duke) has been saturated lately. Crucially, these improvements are achieved with *much smaller model size* – most existing state-of-the-art ReID models employ a ResNet50 backbone, which has more than 24 million parameters (considering their customised extra modules), while our OSNet has only 2.2 million parameters. This verifies the effectiveness of omni-scale feature learning for ReID achieved by an extremely compact network. (2) OSNet yields strong performance with or without ImageNet pretraining. Among the very few existing lightweight ReID models that can be trained from scratch (HAN and BraidNet), OSNet exhibits huge advantages. At R1, OSNet beats HAN/BraidNet by 2.4%/9.9% on Market1501 and 4.2%/8.3% on Duke. The margins at mAP are even larger. In addition, general-purpose lightweight CNNs are also compared without ImageNet pretraining. Table 3 shows that OSNet surpasses the popular MobileNetV2 and ShuffleNet by large margins on all datasets. Note that all three networks have similar model sizes. These results thus demon-

Method	Publication	Backbone	Market1501		CUHK03		Duke		MSMT17	
			R1	mAP	R1	mAP	R1	mAP	R1	mAP
ShuffleNet†‡ [40]	CVPR'18	ShuffleNet	84.8	65.0	38.4	37.2	71.6	49.9	41.5	19.9
MobileNetV2†‡ [10]	CVPR'18	MobileNetV2	87.0	69.5	46.5	46.0	75.2	55.8	50.9	27.0
BraidNet† [19]	CVPR'18	BraidNet	83.7	69.5	-	-	76.4	59.5	-	-
HAN† [2]	CVPR'18	Inception	91.2	75.7	41.7	38.6	80.5	63.8	-	-
OSNet† (ours)	-	OSNet	**93.6**	**81.0**	**57.1**	**54.2**	**84.7**	**68.6**	**71.0**	**43.3**
SVDNet [52]	ICCV'17	ResNet	82.3	62.1	41.5	37.3	76.7	56.8	-	-
PDC [25]	ICCV'17	Inception	84.1	63.4	-	-	-	-	58.0	29.7
HAP2S [53]	ECCV'18	ResNet	84.6	69.4	-	-	75.9	60.6	-	-
DPFL [54]	ICCVW'17	Inception	88.6	72.6	40.7	37.0	79.2	60.6	-	-
DaRe [32]	CVPR'18	DenseNet	89.0	76.0	63.3	59.0	80.2	64.5	-	-
PNGAN [55]	ECCV'18	ResNet	89.4	72.6	-	-	73.6	53.2	-	-
GLAD [51]	ACM MM'17	Inception	89.9	73.9	-	-	-	-	61.4	34.0
KPM [16]	CVPR'18	ResNet	90.1	75.3	-	-	80.3	63.2	-	-
MLFN [1]	CVPR'18	ResNeXt	90.0	74.3	52.8	47.8	81.0	62.8	-	-
DuATM [4]	CVPR'18	DenseNet	91.4	76.6	-	-	81.8	64.6	-	-
Bilinear [26]	ECCV'18	Inception	91.7	79.6	-	-	84.4	69.3	-	-
G2G [56]	CVPR'18	ResNet	92.7	82.5	-	-	80.7	66.4	-	-
DeepCRF [57]	CVPR'18	ResNet	93.5	81.6	-	-	84.9	69.5	-	-
PCB+RPP [5]	ECCV'18	ResNet	93.8	81.6	63.7	57.5	83.3	69.2	-	-
SGGNN [58]	ECCV'18	ResNet	92.3	82.8	-	-	81.1	68.2	-	-
Mancs [59]	ECCV'18	ResNet	93.1	82.3	65.5	60.5	84.9	71.8	-	-
OSNet (ours)	-	OSNet	**94.8**	**84.9**	**72.3**	**67.8**	**88.6**	**73.5**	**78.7**	**52.9**

Table 3. Results (%) on big ReID datasets. It is clear that OSNet achieves the best performance on all datasets, surpassing the published state-of-the-art ReID methods by a clear margin. It is noteworthy that *OSNet has only 2.2 million parameters*, which are far less than current best-performing ResNet-based methods. -: not available. †: model trained from scratch. ‡: reproduced by us.

strate the versatility of our OSNet: It enables effective feature tuning from generic object categorisation tasks and offers robustness against model over-fitting when trained from scratch on datasets of moderate sizes. (3) Compared with ReID models that deploy a multi-scale/multi-stream architecture, namely those with a Inception or ResNeXt backbone [2, 25, 54, 51, 1, 4], OSNet is clearly superior. As analysed 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 AG.

Results on Small Datasets VIPeR and GRID are very challenging datasets for deep ReID approaches because they have only hundreds of training images - training on the large ReID datasets and fine-tuning on them is thus necessary. Table 4 compares OSNet with six state-of-the-art deep ReID methods. On VIPeR, it can be observed that OSNet outperforms the alternatives by significant margins – more than 11.4% at R1. GRID is much more challenging than VIPeR because it has only 125 training identities (250 images) and extra distractors. Further, it was captured by real (operational) analogue CCTV cameras installed in busy public spaces. JLML [28] is currently the best published method on GRID. It is noted that OSNet is marginally better than JLML on GRID. Overall, the strong performance of OSNet on these two small datasets is indicative of its practical usefulness in real-world applications where collecting large-scale training data is unscalable.

Ablation Study Table 5 evaluates our architectural design choices where our primary model is model 1. T is the stream cardinality in Eq. 2. (1) *vs. standard convolutions:* Factorising convolutions reduces the R1 marginally by 0.4% (model 2 vs. 1). This means our architecture design maintains the representational power even though the model size is reduced by more than $3\times$. (2) *vs. ResNeXt-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, with 1.9%/3.1% difference in R1/mAP. This further validates the necessity of our omni-scale design. (3)

Method	Backbone	VIPeR	GRID
MuDeep [8]	Inception	43.0	-
DeepAlign [30]	Inception	48.7	-
JLML [28]	ResNet	50.2	37.5
Spindle [24]	Inception	53.8	-
GLAD [51]	Inception	54.8	-
HydraPlus-Net [3]	Inception	56.6	-
OSNet (ours)	OSNet	**68.0**	**38.2**

Table 4. Comparison with deep learning approaches on VIPeR and GRID. Only Rank-1 accuracy (%) is reported. -: not available.

Model	Architecture	Market1501	
		R1	mAP
1	$T = 4$ + unified AG (**primary model**)	93.6	81.0
2	$T = 4$ w/ full conv + unified AG	94.0	82.7
3	$T = 4$ (same depth) + unified AG	91.7	77.9
4	$T = 4$ + concatenation	91.4	77.4
5	$T = 4$ + addition	92.0	78.2
6	$T = 4$ + separate AGs	92.9	80.2
7	$T = 4$ + unified AG (stream-wise)	92.6	80.0
8	$T = 4$ + learned-and-fixed gates	91.6	77.5
9	$T = 1$	86.5	67.7
10	$T = 2$ + unified AG	91.7	77.0
11	$T = 3$ + unified AG	92.8	79.9

Table 5. Ablation study on architectural design choices.

Multi-scale fusion strategy: To justify our design of the unified AG, we conduct experiments by changing the way how features of different scales are aggregated. The baselines are concatenation (model 4) and addition (model 5). The primary model is better than the two baselines by more than 1.6%/2.8% at R1/mAP. 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) is lost. Empirically, unifying AG improves by 0.7%/0.8% at R1/mAP (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), both the R1 and the mAP decline by 1%. As feature channels encapsulate sophisticated correlations and can represent numerous visual concepts [61], it is advantageous to use channel-specific weights. (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 [8]. As a result, the R1/mAP drops off by 2.0%/3.5% compared with that of dynamic gates (primary model). Therefore, adapting the scale fusion for individual input images is essential. (7) *Evaluation on stream cardinality:* The results are substantially improved 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 β and the image resolution multiplier γ. 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 82M 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 proposed models, most of which are $100\times$ bigger in size. This indicates that OSNet has a great potential for efficient deployment in resource-constrained devices such as a surveillance camera with an AI processor.

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 concatenate the gating vectors of four streams in the last bottleneck, perform k-means clustering on test images of Mix4, and select top-15 images closest to the cluster centres. Fig. 5 shows four example clusters where images within the same cluster exhibit similar patterns, i.e., combinations of global-scale and local-scale appearance.

β	# params	γ	Mult-Adds	Market1501	
				R1	mAP
1.0	2.2M	1.0	978.9M	94.8	84.9
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 6. Results (%) of varying width multiplier β and resolution multiplier γ for OSNet. For input size, $\gamma = 0.75$: 192×96; $\gamma = 0.5$: 128×64; $\gamma = 0.25$: 64×32.

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 [62], 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. 6 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 ReID due to the low resolution of surveillance images. Therefore, 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 requirement for accurate ReID. More examples can be found in the Supplementary Material.

4.2. Evaluation on Person Attribute Recognition

Although person attribute recognition is a category-recognition problem, it is closely related to the person ReID problem in that omni-scale feature learning is also critical: some attributes such as 'view angle' are global; others such as 'wearing glasses' are local; heterogeneous-scale features are also needed for recognising attributes such as 'age'.

Datasets and Settings We use PA-100K [3], the largest person attribute recognition dataset. PA-100K contains 80K training images and 10K test images. Each image is annotated with 26 attributes, e.g., male/female, wearing glasses, carrying hand bag. Following [3], we adopt five evaluation metrics, including mean Accuracy (mA), and four instance-based metrics, namely Accuracy (Acc), Precision (Prec), Recall (Rec) and F1-score (F1). Please refer to [63] for the detailed definitions. Implementation is detailed in the

Figure 5. Image clusters of similar gating vectors. The visualisation shows that our unified aggregation gate is capable of learning the combination of homogeneous and heterogeneous scales in a dynamic manner.

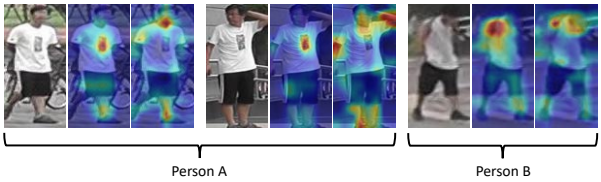

Figure 6. Each triplet contains, from left to right, original image, activation map of OSNet and activation map of single-scale baseline. These images indicate that OSNet can detect subtle differences between visually similar persons.

Method	PA-100K				
	mA	Acc	Prec	Rec	F1
DeepMar [64]	72.7	70.4	82.2	80.4	81.3
HydraPlusNet [3]	74.2	72.2	83.0	82.1	82.5
OSNet	**74.6**	**76.0**	**88.3**	**82.5**	**85.3**

Table 7. Results (%) on pedestrian attribute recognition.

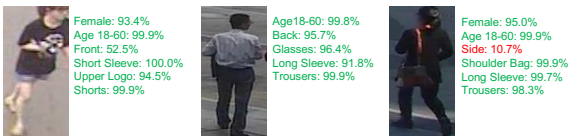

Figure 7. Likelihoods on ground-truth attributes predicted by OSNet. Correct/incorrect classifications based on threshold 50% are shown in green/red.

Supplementary Material.

Results Table 7 compares OSNet with two state-of-the-art methods [64, 3] on PA-100K. It can be seen that OSNet outperforms both alternatives on all five evaluation metrics. Fig. 7 provides some qualitative results. It shows that OSNet is particularly strong at predicting attributes that can only be inferred by examining features of heterogeneous scales such as age and gender.

4.3. Evaluation on Object Categorisation

Datasets and settings CIFAR10/100 [11] has 50K training images and 10K test images, each with the size of 32×32. OSNet is trained following the setting in [65, 66]. Apart from the default OSNet in Table 1, a deeper version is constructed by increasing the number of staged bottlenecks from 2-2-2 to 3-8-6. Error rate is reported as the metric.

Results Table 8 compares OSNet with a number of state-of-the-art object recognition models. The results suggest that, although OSNet is originally designed for fine-grained object instance recognition task in ReID, it is also highly competitive on object category recognition tasks. Note that CIFAR100 is more difficult than CIFAR10 because it contains ten times fewer training images per class (500 vs. 5,000). However, OSNet's performance on CIFAR100 is stronger, indicating that it is better at capturing useful patterns with limited data, hence its excellent performance on the data-scarce ReID benchmarks. We have also conducted experiments on the larger-scale ImageNet 1K object recognition task. The results (see the Supplementary Material) show that our OSNet outperforms similar-sized lightweight models including SqueezeNet [36], ShuffleNet [40] and MobileNetV2 [10]. The overall results show that omni-scale feature learning is beneficial beyond ReID and should be considered for a broad range of visual recognition tasks.

Method	Depth	# params	CIFAR10	CIFAR100
pre-act ResNet [65]	164	1.7M	5.46	24.33
pre-act ResNet [65]	1001	10.2M	4.92	22.71
Wide ResNet [66]	40	8.9M	4.97	22.89
Wide ResNet [66]	16	11.0M	4.81	22.07
DenseNet [67]	40	1.0M	5.24	24.42
DenseNet [67]	100	7.0M	**4.10**	20.20
OSNet	78	2.2M	4.41	19.21
OSNet	210	4.6M	4.18	**18.88**

Table 8. Error rates (%) on CIFAR datasets. All methods here use translation and mirroring for data augmentation. Pointwise and depthwise convolutions are counted as separate layers.

Architecture	CIFAR10	CIFAR100
$T = 1$	5.49	21.78
$T = 4$ + addition	4.72	20.24
$T = 4$ + unified AG	**4.41**	**19.21**

Table 9. Ablation study on OSNet on CIFAR10/100.

Ablation Study We compare our primary model with model 9 (single-scale baseline in Table 5) and model 5 (four streams + addition) on CIFAR10/100. Table 9 shows that both omni-scale feature learning and unified AG contribute positively to the overall performance of OSNet.

5. Conclusion

We presented OSNet, a lightweight CNN architecture that is capable of learning omni-scale feature representations. Extensive experiments on six person ReID datasets demonstrated that OSNet achieved state-of-the-art performance, despite its lightweight design. We also evaluated OSNet on both single-label object categorisation tasks and a multi-label attribute recognition task. The superior performance of OSNet on these tasks suggests that OSNet is of

wide interest to visual recognition problems beyond ReID.

References

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

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

[3] 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.

[4] 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.

[5] 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.

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

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

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

[9] 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.

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

[11] A. Krizhevsky and G. Hinton, "Learning multiple layers of features from tiny images," tech. rep., Citeseer, 2009.

[12] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *CVPR*, 2009.

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

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

[15] R. R. Varior, M. Haloi, and G. Wang, "Gated siamese convolutional neural network architecture for human re-identification," in *ECCV*, 2016.

[16] 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.

[17] Y. Guo and N.-M. Cheung, "Efficient and deep person re-identification using multi-level similarity," in *CVPR*, 2018.

[18] A. Subramaniam, M. Chatterjee, and A. Mittal, "Deep neural networks with inexact matching for person re-identification," in *NIPS*, 2016.

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

[20] X. Zhang, H. Luo, X. Fan, W. Xiang, Y. Sun, Q. Xiao, W. Jiang, C. Zhang, and J. Sun, "Alignedreid: Surpassing human-level performance in person re-identification," *arXiv preprint arXiv:1711.08184*, 2017.

[21] 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.

[22] 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.

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

[24] 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.

[25] 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.

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

[27] 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.

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

[29] 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.

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

[31] 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.

[32] 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.

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

[34] 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.

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

[36] 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.

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

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

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

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

[41] M. Lin, Q. Chen, and S. Yan, "Network in network," *arXiv preprint arXiv:1312.4400*, 2013.

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

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

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

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

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

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

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

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

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

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

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

[53] 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.

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

[55] 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.

[56] 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.

[57] 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.

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

[59] 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.

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

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

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

[63] D. Li, Z. Zhang, X. Chen, H. Ling, and K. Huang, "A richly annotated dataset for pedestrian attribute recognition," *TIP*, 2016.

[64] D. Li, X. Chen, and K. Huang, "Multi-attribute learning for pedestrian attribute recognition in surveillance scenarios," in *ACPR*, 2015.

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

[66] S. Zagoruyko and N. Komodakis, "Wide residual networks," in *BMVC*, 2016.

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

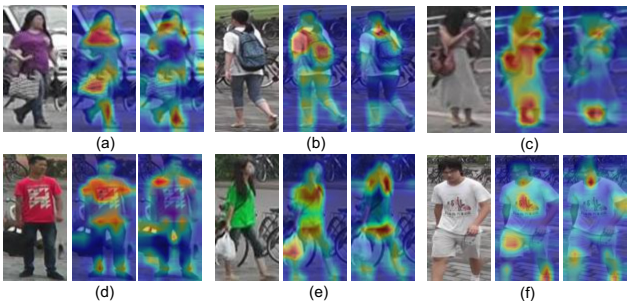

Figure 8. Visualisation of activation maps obtained by OSNet (middle one in each triplet) and the single-scale baseline (right one in each triplet).

Supplementary

A. More Visualisation on Person ReID

In addition to Fig. 6 in the main paper, Fig. 8 here provides more examples of activation maps to support the claim that OSNet can learn discriminative features with homogeneous and heterogeneous scales. It can be observed from Fig. 8 that OSNet is able to identify local patterns with their context as the focus of attention. For example, Figs. 8(d) and (f) show that both the T-shirts and the logos are selected for feature extraction. In contrast, the single-scale baseline tends to only focus on local regions while ignoring the contextual information. This renfers the model more susceptible to occlusion and ambiguity of small local patterns.

B. Implementation Details on PA-100K

A sigmoid-activated attribute prediction layer is added on the top of OSNet. Following [64, 3], we use the weighted multi-label classification loss for supervision. For data augmentation, we adopt random translation and mirroring. OSNet is trained from scratch with SGD, momentum of 0.9 and initial learning rate of 0.065 for 50 epochs. The learning rate is decayed by 0.1 at 30 and 40 epochs.

C. Evaluation on ImageNet

In Sec. 4.3 of the main paper, we have reported results of OSNet on the object category recognition tasks of CIFAR10/100. In this section, the results on the larger-scale ImageNet 1K category dataset (LSVRC-2012 [12]) are discussed.

Implementations OSNet is trained with SGD, initial learning rate of 0.4, batch size of 1024 and weight decay of 4e-5 for 120 epochs. For data augmentation, we use random 224×224 crops on 256×256 images and random mirroring. To benchmark, we report single-crop[5] top1 accuracy on the LSVRC-2012 validation set [12].

[5] 224×224 centre crop from 256×256.

Method	β	# params	Mult-Adds	Top1
SqueezeNet [36]	1.0	1.2M	-	57.5
MobileNetV1 [9]	0.5	1.3M	149M	63.7
MobileNetV1 [9]	0.75	2.6M	325M	68.4
MobileNetV1 [9]	1.0	4.2M	569M	70.6
ShuffleNet [40]	1.0	2.4M	140M	67.6
ShuffleNet [40]	1.5	3.4M	292M	71.5
ShuffleNet [40]	2.0	5.4M	524M	73.7
MobileNetV2 [10]	1.0	3.4M	300M	72.0
MobileNetV2 [10]	1.4	6.9M	585M	74.7
OSNet (ours)	0.5	1.1M	424M	69.5
OSNet (ours)	0.75	1.8M	885M	73.5
OSNet (ours)	1.0	2.7M	1511M	**75.5**

Table 10. Single-crop top1 accuracy (%) on ImageNet-2012 validation set. β: width multiplier. M: Million.

Results Table 10 shows that OSNet outperforms the alternative lightweight models by a clear margin. In particular OSNet$\times$1.0 surpasses MobiltNetV2$\times$1.0 by 3.5% and MobiltNetV2$\times$1.4 by 0.8%. It is noteworthy that MobiltNetV2$\times$1.4 is around 2.5$\times$ larger than our OSNet$\times$1.0. OSNet$\times$0.75 performs on par with ShuffleNet$\times$2.0 and outperforms ShuffleNet$\times$1.5/$\times$1.0 by 2.0%/5.9%. These results give a strong indication that OSNet has a great potential for a broad range of visual recognition tasks. Note that although the model size is smaller, our OSNet does have a higher number of mult-adds operations than its main competitors. This is mainly due to the multi-stream design. However, if both model size and number of Multi-Adds need to be small for a certain application, we can reduce the latter by introducing pointwise convolutions with group convolutions and channel shuffling [40].