This CVPR 2021 paper is the Open Access version, provided by the Computer Vision Foundation.
Except for this watermark, it is identical to the accepted version;
the final published version of the proceedings is available on IEEE Xplore.

Combined Depth Space based Architecture Search For Person Re-identification

Hanjun Li[1,4], Gaojie Wu[1], Wei-Shi Zheng[1,2,3,*]
[1]School of Computer Science and Engineering, Sun Yat-sen University, China
[2]Peng Cheng Laboratory, Shenzhen 518005, China
[3]Key Laboratory of Machine Intelligence and Advanced Computing, Ministry of Education, China
[4]Pazhou Lab, Guangzhou, China
lihj85@mail2.sysu.edu.cn, wugj7@mail2.sysu.edu.cn,wszheng@ieee.org

Abstract

Most works on person re-identification (ReID) take advantage of large backbone networks such as ResNet, which are designed for image classification instead of ReID, for feature extraction. However, these backbones may not be computationally efficient or the most suitable architectures for ReID. In this work, we aim to design a lightweight and suitable network for ReID. We propose a novel search space called Combined Depth Space (CDS), based on which we search for an efficient network architecture, which we call CDNet, via a differentiable architecture search algorithm. Through the use of the combined basic building blocks in CDS, CDNet tends to focus on combined pattern information that is typically found in images of pedestrians. We then propose a low-cost search strategy named the Top-k Sample Search strategy to make full use of the search space and avoid trapping in local optimal result. Furthermore, an effective Fine-grained Balance Neck (FBLNeck), which is removable at the inference time, is presented to balance the effects of triplet loss and softmax loss during the training process. Extensive experiments show that our CDNet (∼1.8 M parameters) has comparable performance with state-of-the-art lightweight networks.

1. Introduction

Person re-identification (ReID) aims to retrieve images of a specific person from different surveillance cameras. Since AlexNet [15] was proposed in the ILSVRC-2012 [3], convolution neural networks (CNNs) have become increasingly popular. With the emergence of complex models [32, 28, 6], people tend to utilize them as backbones to achieve higher performance on the ReID task. However, there are two obvious drawbacks to such implementations. First, they rely deeply on the performance of the backbone and limit researchers to explore more suitable network architectures for ReID. Second, these backbones require large computational resources and time costs at inference time, making them unaffordable for some practical/edge devices with limited computing resources, such as intelligent surveillance cameras. Instead, by deploying a lightweight network across a number of surveillance cameras, only the features these devices extract need to be gathered to retrieve the target person, which is much faster than gathering raw images and processing them with very large backbone networks. For these reasons, we aim to construct a lightweight network that is computationally efficient and more suitable for ReID.

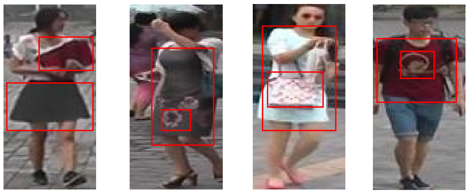


Figure 1. Note that paired salient objects of different sizes can be commonly found in pedestrian images. Our CBlock is designed to explicitly learn such combined patterns.

In recent years, Neural Architecture Search (NAS) has been utilized to search lightweight but effective networks. [50] takes 2000 GPU days to search the NASNet via reinforcement learning, which is far too long for most of researchers. To reduce the expensive search cost, [20] proposes a novel algorithm called differentiable architecture search (DARTS) using gradient descent, which dramatically reduces the search cost to 4 GPU days. Although the searched network architecture is very small, this method still has a few drawbacks. (1) The cell contains numerous complex connections, which are detrimental to parallel computing. [19] also notes that the irregular cell structures

[*] Corresponding author

are not GPU friendly. (2) The algorithm only searches for normal and reduction cells and applies them to different layers. We argue that CNNs tend to concentrate on different pattern information at different depths and thus need to distinguish structures at different layers. (3) During the search, the algorithm computes each branch during forward propagation, even though the contributions of some branches with lower probabilities are negligible, resulting in heavy computational costs. As for the third drawback, [4] chooses to only compute the branch with the maximum weight between two nodes for forward propagation. However, this method may easily become trapped in a single local optimal network architecture since the gradient is mainly updated in the selected branch and other possible branches are gradually ignored; thus, the method cannot make full use of the search space. Apart from the above problems, we observe that most current search spaces are unable to explicitly learn combined pattern features for ReID (see Fig. 1), which are known to have very strong discriminative value.

To address the aforementioned problems and make full use of the advantages of NAS to search lightweight networks, we propose a novel search space called Combined Depth Space (CDS) and a new search strategy called the Top-k Sample Search. In CDS, we design an efficient Combined Block (CBlock) consisting of two independent branches with different kernel sizes for explicitly learning combined pattern information. In this way, our CBlock only has two parallel branches and is thus GPU friendly. Moreover, our Top-k Sample Search computes the top-k branches according to the weights during the forward propagation, avoiding the computation of negligible branches or becoming trapped in a single local optimal network architecture, as is the case for [4]. In this way, we can not only largely reduce the search cost but also obtain a competitive lightweight network. Different from [20], we choose to search the cells for each layer independently.

In addition, we jointly optimize the softmax loss and triplet loss for training, as in many works [8, 22, 24]. Particularly, we further propose a simple but effective Balance Neck (BLNeck) to resolve the inconsistency between the targets of these two losses in the embedding space. In [22], although BNNeck is presented to balance the effects of these losses, it does not always work for arbitrary network architectures (as shown in Table 6). However, the proposed BLNeck has a strong ability to map an embedding space constrained by the triplet loss to one constrained by the softmax loss; thus, the two losses can be optimized harmoniously. The stripe strategy is always used for extracting local features to guide the model to focus on more detailed information. We thus also integrate this idea into our Balance Neck, obtaining a new neck structure called the Fine-grained Balance Neck (FBLNeck) to further improve the performance.

In summary, the contributions of this paper are summarized as follows:

- We propose a novel search space called Combined Depth Space (CDS), in which the CBlocks explicitly learn combined pattern features and are more suitable for ReID.
- We propose a new search strategy called layer-wise Top-k Sample Search, which can largely reduce the search cost over that of other search strategies and make full use of the search space.
- We propose a simple but effective Fine-grained Balance Neck (FBLNeck) for balancing the effects of triplet loss and softmax loss to better leverage their advantages.

The extensive experiments show that our CDNet achieves state-of-the-art performance on both ReID and other tasks among lightweight networks.

2. Related Works

2.1. Lightweight Networks

In recent years, to reduce the computational complexity of CNNs, some researchers have begun to explore effective and small-size models. MobileNets [11, 27, 10] utilize depthwise separable convolution and reduce the number of parameters largely while maintaining comparable performance to standard networks. ShuffleNets [44, 23] utilize group convolution to further reduce the number of parameters. Particularly, [23] argues that an excessive number of branches will impede parallel computation, which we have also addressed in the present work. Other researchers obtain small-size networks via knowledge distillation [39], quantization [5, 41, 13], network pruning [5, 17, 37, 21], and so on. The greatest drawback of these design methods is that numerous experiments need to be conducted to empirically determine the best network structure.

To automate the architecture design process, reinforcement learning and evolution learning have been introduced to search efficient network architectures with competitive accuracy on classification tasks [50, 25, 33]. After [20] proposed their differentiable architecture search algorithm via gradient decent, a number of researchers published extended works [42, 4, 40, 2] with similar algorithms. However, most of these methods adopt the search space and algorithm as DARTS, which has a number of drawbacks, as mentioned in the Introduction. Therefore, we propose a new search space called the Combined Depth Space, which is more efficient and suitable for the task of ReID.

2.2. Person Re-Identification

Recently, most of the proposed ReID models mainly utilize a complex network (*e.g*. ResNet) as the backbone and integrate some special structures to extract extra information to enhance the discriminative features. [31, 46, 36] utilize the stripe strategy to jointly extract both global features and local features and achieve great performance. [45, 29, 35, 49] introduce extra information such as body masks, camera information, and view labels to further improve the performance of the network. Noticeably, most of these methods adopt ResNet as the backbone, which has a large number of parameters and needs considerable computational resources. In practice, however, such a model is difficult to deploy in edge devices such as surveillance cameras with limited computational resources. We thus must avoid utilizing ResNet and aim to build an efficient but lightweight network to satisfy these specific demands.

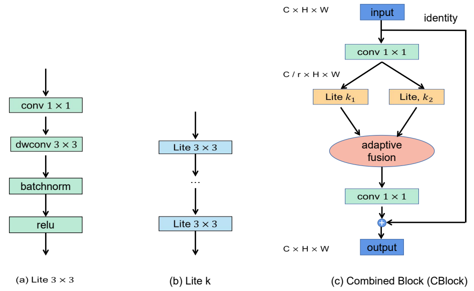
Indeed, some works [18, 16, 48] have been published describing the design of small networks for ReID. [24] introduces a part-aware block into the search space in DARTS and search a lightweight network for ReID. [48] proposes OSNet, which can learn omni-scale features and achieve promising results on both ReID and classification tasks. As a result of the special structure of OSNet, four branches compute the features in parallel at each layer; however, this leads to heavy computational resource consumption despite it only has 2.2 M parameters. We argue that an excessive number of parallel branches tend to extract redundant information; thus, we can discard specific branches to improve efficiency.

3. Methodology

In this section, we describe the proposed CDS in section 3.1. Then, the Top-k Sample Search algorithm is shown in section 3.2. Finally, we introduce the FBLNeck in section 3.3.

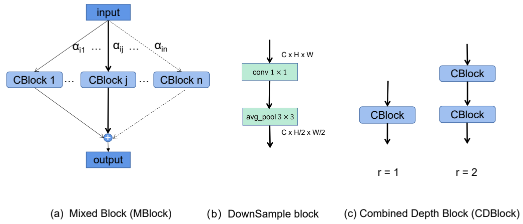
3.1. Combined Depth Space

To explore a more suitable architecture for ReID, we design a new efficient search space called Combined Depth Space (CDS), in which the building blocks explicitly learn combined pattern features. Based on CDS, the depth of the search network can adaptively vary within a specific range. Before defining the CDS, we first introduce the basic blocks. As shown in Fig. 2 (a), we adopt Lite 3×3 from [48] as our basic block.



(a) Lite 3 × 3 (b) Lite k (c) Combined Block (CBlock)

Figure 2. (a) Lite 3 × 3 is the most basic building block. (b) Lite k consists of $\lfloor \frac{k}{2} \rfloor$ Lite 3 × 3. (c) CBlock combines two kinds of kernel (k_1, k_2), and C × H × W denotes the current shape of the tensor.



(a) Mixed Block (MBlock) (b) DownSample block (c) Combined Depth Block (CDBlock)

Figure 3. MBlock consists of different CBlocks. Each branch has its own architecture parameters α_{ij}. (b) A DownSample block is utilized to reduce the spatial size of the feature map between any two stages. (c) CDBlock consists of one or two CBlocks.

- CBlock. As shown in Fig. 2(c), our CBlock utilizes two different kernels with different receptive fields to jointly learn various scale patterns. We then elaborately fuse the features learned and obtain more discriminative features. Therefore, let CK ={(3,5), (3,7), (3,9), (5,7), (5,9), (7,9)}; given the combination $(k_i, k_j) \in CK$, we can obtain 6 types of CBlock. To reduce the number of parameters and computation costs, we replace the depthwise convolution k × k with $\lfloor \frac{k}{2} \rfloor$ Lite 3×3 (see in Fig. 2(b)) since their final feature maps share the same receptive field, and there are fewer parameters with the latter than with the former. Let g denote Lite 3×3. Given input x and kernel size k, we can formulate the computation process of Lite k as follows:

$$Lite(x, k) = g(x) \circ \lfloor \frac{k}{2} \rfloor \tag{1}$$

op ∘ t means that op is run t times consecutively. Because the features learned from the two kernels are heterogeneous, it would be improper to simply sum them. As [48] introduces an Adaptive Fusion Gate, we compute the weights of each channel according to the input and then compute the channel-wise weighted summation of $Lite(x, k_1)$ and $Lite(x, k_2)$. The two conv 1×1 are utilized to squeeze and restore the channels at a ratio r.

- MBlock. Our MBlock is equivalent to the cell in DARTS. However, we simplify the search cell and only need to select the appropriate combination of kernels rather than determine the complex connections between the inner nodes. Specifically, here, the combination of CBlocks belongs to the set of CK, that is, MBlock only has six candidate operations. As shown in Fig. 3(a), each branch has its own weight α_{ij}, representing the importance of the current CBlock j.

DARTS only searches one normal cell and one reduc-

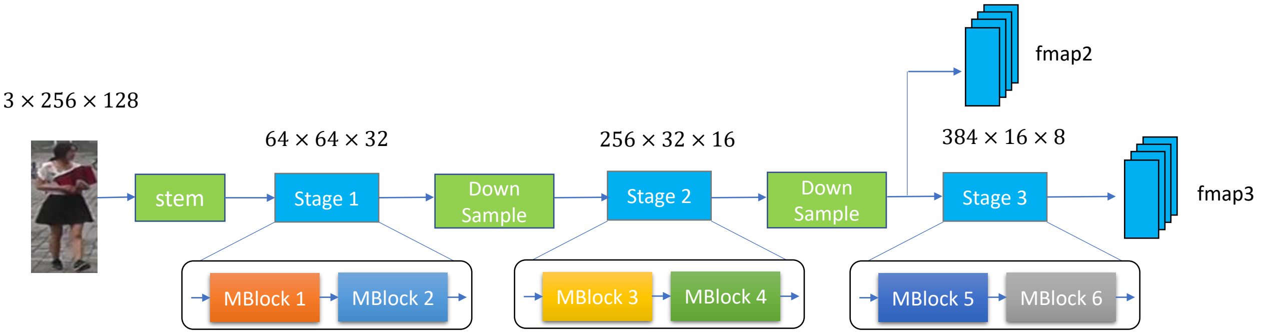


Figure 4. The macro architecture of the search network for person re-identification. It includes a stem and three stages in total. Each stage stacks two MBlocks. Between every two stages, we insert a DownSample block to reduce the spatial size. The description c × h × w over each stage denotes the number of channels and the height and width of the in-tensor.

tion cell and shares the inner structure across different layers. We argue that blocks at different depths of the neural network may focus on different information; thus, it would be sub-optimal to simply search two cells and stack the searched cells until comparable performance is obtained. Because of our efficient search strategy (described in section 3.2), we design a macro framework first, as shown in Fig. 4, and search independent cells for each layer during the neural architecture search. At the beginning of the search network, we adopt a normal convolution stem consisting of a 7×7 standard convolution with stride 2 and 3×3 max pooling with stride 2. As we can see, there are 6 MBlocks to be searched at different depths. Unlike DARTS, we utilize a fixed reduction block, the DownSample block (Fig. 3(b)).

Based on the aforementioned search space, termed Combined Space (CS), we call the searched architecture network CNet. Since each MBlock has 6 candidate operations, we can easily determine that the total size of the CS is $6^6 \approx 10^{4.7}$. Compared with the space of the normal NAS methods, the size of our CS is relatively small; nevertheless, it contains a sufficient number of efficient and high-quality structures.

- CDBlock. Note that our search network only has 6 MBlocks, which is relatively shallower than ResNet with its 16 building blocks. We can simply add one or two MBlocks to each stage to deepen our network, but this will greatly increase the number of parameters and computations. To effectively deepen the network, an appropriate number of blocks should be allocated at each stage rather than randomly or uniformly. To achieve this, we introduce a depth factor into CS, forming a new search space termed Combined Depth Space (CDS). We redefine CK as CDK = {(3,5,1), (3,7,1), (3,9,1), (5,7,1), (7,9,1), (3,5,2), (3,7,2), (3,9,2), (5,7,2), (7,9,2)}; let tuple $(k_1, k_2, r) \in CDK$. k_1, k_2 denote the kernel size for two branches in CBlock. r denotes the number of times CBlock is repeated. Given the redefined CDK, we can construct CDBlock as shown in Fig. 3(c). Therefore, we can replace the CBlock in the original MBlock with CDBlock to construct a new MBlock. Similarly, we can easily compute the size of the CDS as $12^6 \approx 10^{6.5}$. After considering the depth factor, the depth of our network can range from 6 to 12, making our network more flexible. With the CDS, we can adaptively search for the optimal network depth rather than determine the network depth manually.

3.2. Top-k Sample Search

We denote the architecture parameters as α = $\{\alpha_1, \alpha_2, ..., \alpha_6\}$, where each α_i is a n-dim vector that denotes the weights of each branch in the i_{th} MBlock. Specially, n is 6 for the CS and 12 for the CDS. We denote the network parameters of the search network as w. Therefore, our goal is to find the optimal architecture parameters α^* that can achieve a minimum validation loss L_V after w is updated by minimizing the training loss L_T, as shown in Eq.2.

$$\begin{aligned} min_\alpha \quad & L_V(w^*(\alpha), \alpha) \\ & s.t. \quad w^*(\alpha) = argmin_w L_T(w, \alpha) \end{aligned} \tag{2}$$

Obviously, the first impulse is to solve the top equation above by updating w until it converges on the training set and then updating α until it converges on the validation set. These two steps are repeated until the search network reaches a state of convergence. However, attempting to achieve convergence in each iteration is time consuming, making this method unsuitable for finding the optimal result. Similar to [4, 40], w is instead updated using single batch training data, and α is updated using single batch validation data at each epoch. After updating w and α alternatively for an adequate number of epochs, the optimal value α^* for Eq. 2 is ultimately approximated.

During the search, we only compute the top-k branches according to α_i for MBlock i. Given input x and MBlock

i, we denote the forward propagation as $F(i,x)$ and the j_{th} branch in MBlock as f_j. Let n denote the number of branches; the probability distribution of MBlock i, $p_i = [p_{i1}, p_{i2}, ..., p_{in}]$, is obtained from Eq.3,

$$p_{ij} = \frac{exp(\alpha_{ij})}{\sum_{j=1}^{n} exp(\alpha_{ij})} \tag{3}$$

We then construct a binary vector $h_i = [h_{i1}, h_{i2}, ..., h_{in}]$ via Eq. 4, where h_{ij} is 1 if the j_{th} branch is selected and 0 if it is dropped in forward propagation.

$$h_{ij} = \begin{cases} 1, & p_{ij} \geq v_k \\ 0, & p_{ij} < v_k \end{cases} \tag{4}$$

where v_k denotes the k_{th} largest value in p_i. Subsequently, we can formulate the forward propagation of MBlock i as follows:

$$F(i,x) = \sum_{j=1}^{n} h_{ij} f_j(x) \tag{5}$$

However, since h_i is a discrete distribution from a probability, $F(i,x)$ cannot back propagate gradients to α_i via h_i. To allow back propagation, we aim to build a bridge between h and α.

$$m_i = h_i^* - p_i^* \tag{6}$$

$$\hat{h_i} = m_i + p_i \tag{7}$$

where p_i^*, h_i^* denotes a copy of p_i, h_i without gradients. Thus m_i is a normal vector without gradients. We then replace h_{ij} in Eq. 5 with $\hat{h_{ij}}$. Since the gradients of p_i need to be computed, the gradients of $\hat{h_i}$ in Eq.7 also need to be computed. In this way, we can compute the gradient of α_{ij} in MBlock i as follows:

$$\frac{\partial F(i,x)}{\partial \alpha_{ij}} = \sum_{k \in topk} \left(\frac{\partial F(i,x)}{\partial \hat{h_{ik}}} \frac{\partial \hat{h_{ik}}}{\partial p_{ik}} \frac{\partial p_{ik}}{\partial \alpha_{ij}} \right) \tag{8}$$

The Eq. 5 and Eq. 8 show that we treat the top-k branches equally during forward propagation (each weight is 1) but update the architecture parameters according to its real weight p. Our search algorithm can be summarized as Algorithm 1.

3.3. Fine-grained Balance Neck

To efficiently tackle the problem that the targets of the triplet and softmax losses are inconsistent in the embedding space, we propose the effective Fine-grained Balance Neck (FBLNeck). As shown in Fig. 5, the FBLNeck consists of two parts. The upper part is named Balance Neck (BLNeck), which only utilizes global information, and the lower part is named Fine-grained Neck (FNeck), which partitions fmap2 into two stripes and extracts local information.

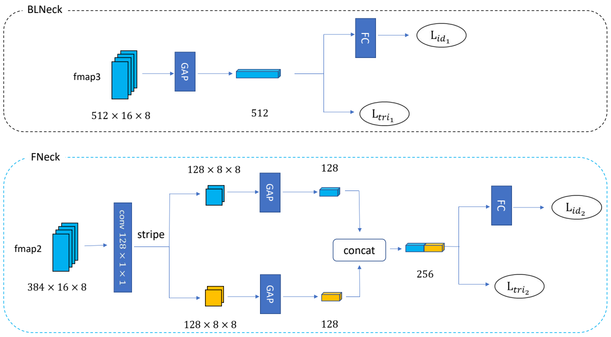


Figure 5. The schematic of the Fine-grained Balance Neck consisting of BLNeck and FNeck. fmap2 and fmap3 are shown in Fig. 4. The main idea is to insert a fully connected layer between feats, which is constrained by the triplet loss, and feats, which is constrained by the softmax loss. Before partitioning fmap2, we first squeeze the channel to 128-dim.

Note that FNeck also has similar structure as BLNeck for aforementioned losses. It is worth noting that our local features are extracted at a relatively shallow layer rather than at the last layer selected by most works. This is because the shallow layers have a smaller receptive field than the deeper layers and thus tend to focus on fine-grained features. Additionally, we desire to avoid using the same global features as above and instead make full use of earlier features, which would lead to greater discriminability. Importantly, the fc layer mainly transforms the triplet-friendly embedding space into a softmax-friendly embedding space in both BLNeck and FNeck. During the search training, we do not utilize FNeck; thus, our search loss can be formulated as follows:

$$L_{search} = L_{tri1} + L_{id1} \tag{9}$$

Algorithm 1 Top-k Sample Search

```
Input: Split the training set into two disjoint sets: $D_{train}$
    and $D_{val}$; Given the total search epoch $E$, architecture
    parameter learning rate $\eta_\alpha$, network parameter learning
    rate $\eta_w$, initialize $\alpha$ and $w$.
Output: Derived the final architecture from the learned $\alpha$.
for $e = 1$ to $E$ do
    for batch data $(X_t, Y_t)$ in $D_{train}$ do
        Compute the training loss $L_T(X_t, Y_t)$
        Update $w$: $w^* = w - \eta_w \bigtriangledown_w L_T$
        Sample batch data $(X_v, Y_v)$ from $D_{val}$
        Compute the validation loss $L_V(X_v, Y_v)$
        Update $\alpha$: $\alpha^* = \alpha - \eta_\alpha \bigtriangledown_\alpha L_V$
```

During the training of CNet or CDNet, we calculate the objective loss as Eq. 10,

$$L_{train} = L_{tri1} + L_{id1} + L_{tri2} + L_{id2} \tag{10}$$

4. Experiments

4.1. Search for CNet and CDNet

Model	Top-k	Param	Time	rank-1	mAP
CNet	top-1	1.4M	30.4ms	93.3	82.6
	top-2	1.4M	40.0ms	**93.6**	**83.5**
	top-3	1.3M	51.2ms	93.1	81.9
	top-4	1.4M	57.7ms	92.9	82.3
CDNet	top-1	2.1M	27.3ms	93.4	83.2
	top-2	1.8M	25.3ms	**93.7**	**83.7**
	top-3	1.9M	48.4ms	93.6	83.5
	top-4	1.6M	43.1ms	93.2	83.0

Table 1. The performance of different searched architectures evaluated on Market1501. The time is the average time required to process one image during the search. M:Million

Different from normal NAS, we search the network architectures on the Market1501 [47]. According to Algorithm 1, we further split the training set of Market1501 into a new training set and a validation set. Additional details on the data preparation and experimental configuration are described in the supplementary materials. Considering the computational costs involved, we conduct a series of Top-k Sample Search, $k \in$ {1,2,3,4}. After the search, we derive the architecture according to the learned α. For each MBlock, we select j_{th} CBlock or CDBlock with the maximum weight eventually. Based on CS and CDS, the architectures derived from the search network are called CNet and CDNet, respectively. We then evaluate the searched architectures on Market1501. As shown in Table 1, the top-2 sample search yields the best architecture for both CNet and CDNet. Top-1 sample search tends to become trapped in a local architecture and cannot produce the optimal architecture. As the number of computational branches increases, the performance of the searched architectures worsens because the excess branches may exhibit a competitive relationship. Specially, we compute all branches of the search network like DARTS and the time cost as in Table 1 is 100ms and 83.4ms for CNet and CDNet respectively. Taking search cost and performance into account, we suggest that the top-2 sample search is the best choice. The speed of the top-2 sample search is 2.5× greater than that of DARTS. We show the final architectures of CNet and CDNet via the top-2 sample search in Table 2.

4.2. Evaluation on Person Re-Identification

Datasets and implementation details We conduct experiments on three widely used person ReID datasets: Market1501 [47], DukeMTMC [26] and MSMT17 [38]. For all experiments, all images are resized to a resolution of 256 × 128, and Random Erasing Augment in [22] is utilized to imitate occlusion. During the training of CDNet and CNet, we adopt a training scheme similar to that in [48]. We set the number of stripes at FNeck to 2. Finally, all models are trained with triplet loss and softmax loss unless stated otherwise.

Layer	CNet	CDNet
	(k_1, k_2)	(k_1, k_2, r)
1	(5,7)	(3,5,1)
2	(7,9)	(3,7,2)
3	(7,9)	(5,7,2)
4	(7,9)	(5,9,1)
5	(7,9)	(5,7,2)
6	(3,5)	(5,7,1)

Table 2. The architectures of CNet and CDNet. k_1, k_2 denotes the kernel size of the two branches in CBlock, and r denotes the number of times CBlock is repeated within CDBlock.

Result analysis As shown in Table 3, among the models that are not pretrained with ImageNet, both CNet and CDNet achieve competitive performances.The margins for mAP are even larger. Compared with that of Auto-ReID, which incorporates a part-aware super block into the search space of DARTS and searches for an architecture similar to DARTS, the rank-1 accuracy and mAP of CNet are better by 3.9% and 10.8%, respectively, on Market1501 with ∼5× fewer parameters. Hence, these results indirectly demonstrate that our search spaces (CS, CDS) are superior to the search space in DARTS. Furthermore, although we utilize the Lite 3×3 basic block proposed in OSNet, our CNet outperforms OSNet in terms of mAP by a large margin on both Market1501 and DukeMTMC with much fewer parameters. CNet can be interpreted as a subnet of OSNet, which verifies that many branches of OSNet are redundant and useless. In contrast, CNet is more effective and computationally economical without redundant branches. Although HA-CNN utilizes an attention mechanism and BagofTricks utilizes many training tricks, they are still inferior to our models without ImageNet pretraining. CDNet is the enhanced version of CNet with an increased adaptive depth, which further improves the performance with only a small increase in the number of parameters. Since pretraining on ImageNet yields better performance for most networks, we also pretrained CDNet on ImageNet. As shown in the bottom half of Table 3, CDNet still achieves competitive performance among the models listed on all datasets with only 1.8 M parameters. Compared with OSNet, CDNet achieves higher mAP on all datasets, suggesting that our model is more robust in identifying difficult positive identities.

4.3. Visualization of Combined Pattern Learning

CBlock is the core building block of CDNet and is designed to explicitly learn combined patterns in images. To

Model	Param(M)	Market1501		DukeMTMC		MSMT17	
		rank-1	mAP	rank-1	mAP	rank-1	mAP
ShuffleNet*[44]	~1.9	84.8	65.0	71.6	49.9	41.5	19.9
MobileNet*[27]	~2.14	87.0	69.5	75.2	55.8	50.9	27.0
OSNet[48]	2.2	93.6	81.0	84.7	68.6	71.0	43.3
HA-CNN[18]	2.7	91.2	75.7	80.5	63.8	-	-
Auto-ReID[24]	11.4	89.7	72.7	-	-	-	-
BagofTrick$^+$[22]	~26	83.7	65.8	76.0	62.2	-	-
CNet(ours)	1.44	93.6	83.5	86.0	73.2	73.3	47.7
CDNet(ours)	1.8	93.7	83.7	86.7	73.9	73.7	48.5
PCB(+RPP)[31]	~26	93.8	81.6	83.3	69.2	68.2	40.4
VPM[30]	~26	93.0	80.8	83.6	72.6	-	-
BagofTricks[22]	~26	94.5	85.9	86.4	76.4	-	-
IANet[9]	~26	94.4	83.1	-	-	75.5	46.8
CtF[34]	~26	93.7	84.9	87.6	74.8	-	-
SCSN[1]	~26	95.7	88.5	90.1	79.0	83.0	58.0
OSNet[48]	2.2	94.8	84.9	88.6	73.5	78.7	52.9
Auto-ReID[24]	11.4	94.5	85.1	88.5	75.1	-	-
CDNet(ours)	1.8	95.1	86.0	88.6	76.8	78.9	54.7

Table 3. Performance on the Market1501, DukeMTMC and MSMT17 datasets. All of the models listed in the top half of the table are trained from scratch, and those in the bottom half are pretrained on ImageNet. * denotes that the result comes from [48]. + denotes that we reproduce the result. - denotes that the result is unavailable. The number of parameters is counted at inference time. Best and second best results are colored with red and blue respectively. It is clear that our models surpass most published models by a clear margin with the fewest number of parameters.

verify that the proposed CDNet can learn discriminative combined features, we visualize the last feature maps at stage1, stage2 and stage3. The top left images in Fig. 6 show a girl dressed in a black skirt with a red book in her hand. CDNet primarily captures the combined information on the skirt and the book. For the person in the bottom left images, CDNet mainly focuses on the handbag. Other than this salient object combination, plain patterns on clothing can also be captured by CDNet, as shown for the person in the top right images. Moreover, it can be observed that the legs of all identified persons are captured by CDNet, since the legs and the background can also be seen as a salient combination. Obviously, CDNet can effectively capture dominating information and ignore useless background information.

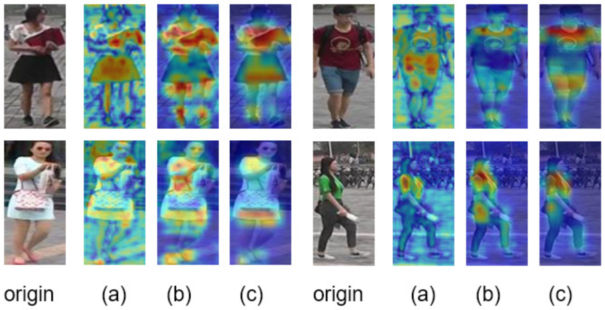


Figure 6. Visualization of the learning representation from CDNet. The column origin corresponds to the original image. Columns (a), (b), and (c) correspond to the last feature maps of stage1, stage2 and stage3, respectively.

4.4. Evaluation on Classification Task

The CIFAR-100 [14] consists of 50K training images and 10K test images comprising 100 categories. The size of each image is 32 × 32. Random horizontal flip and random crop are utilized for data augmentation. We normalize the pixel values as in [4], and the other training settings are the same as in section 4.2. As shown in Table 4, although GDAS and DARTS are originally searched for classification, they do not have significant advantages on CIFAR-100. With the exception of DARTS, our CDNet outperforms the other lightweight networks by a clear margin. In particular, CDNet is better than OSNet designed for ReID by 1.83%. Obviously, the superior performance on classification task demonstrates the benefit of learning combined pattern information.

4.5. Ablation Study

- Best partitions for FNeck . Related works [31, 46, 36] split the final feature map to extract local features and subsequently enhance the discriminability by combining them with global features.However, they all only utilize the last

Model	Param(M)	Error(%)
GDAS[4]	2.5	18.13
DARTS*[20]	3.4	17.54
DensNet[12]	7.0	20.20
pre-act ResNet[7]	10.2	22.71
Wide ResNet[43]	11.0	22.07
OSNet[48]	2.2	19.21
CDNet(ours)	2.3	17.83

Table 4. Error rates on CIFAR-100. * indicates that the result for DARTS is obtained from GDAS.

feature map; thus, there is much similar information between global features and the local features, actually does little to improve discriminability.We propose exploring the use of shallower information for two reasons. First, it can avoid using the same information as global features, instead making full use of other features. Second, the shallower layer has a relatively small receptive field; thus, the local features are more fine-grained. In Table 5, we explore the suitable number of partitions for FNeck. By comparing the first two rows, we see that the features of the shallower layer indeed greatly contribute to improving performance. Both rank-1 accuracy and mAP are improved with a small increase in the feature-dim when the number of partitions increases to 2. As the number of partitions further increases, the mAP does not notably improve. Although CDNet achieves the best rank-1 accuracy (94.2%) for 4 partitions, this would require considerable computational resources. Therefore, we choose 2 partitions for all experiments without further comment.

Partition	Feature-dim	rank-1	mAP
0	512	93.1	81.5
1	640	93.0	82.6
2	768	93.7	**83.7**
3	896	93.2	82.7
4	1024	**94.2**	83.1

Table 5. Effect of the number of FNeck partitions. The experiments are conducted with CDNet on Market1501. The entry for 0 partitions refers to an implementation without FNeck.

-Effect of FBLNeck. To effectively make full use of the combination of the triplet loss and softmax loss, we propose a simple but effective neck structure called FBLNeck to balance these two losses for optimization. Although the BNNeck proposed in [22] can balance these two losses to achieve training convergence, it cannot guarantee optimal results. We first introduce BNNeck into our model, and the result can be seen in the first row in Table 6. The use of BLNeck improves the rank-1/mAP by 2.1%/3.5% over the use of BNNeck. Obviously, our BLNeck can better balance the effects of the triplet loss and softmax loss. We then further combine BLNeck and FNeck as FBLNeck, which can extract both local and global features. As shown in the third row in Table 6, both the rank-1 accuracy and mAP are greatly improved, which demonstrates that the fine-grained information extracted from shallower depths by FNeck is helpful. It is worth noting that our FBLNeck can be removed at inference times, thus making the model more lightweight and efficient. Experiments about introducing FBLNeck into OSNet are shown in the supplementary materials.

Architecture	Feature-dim	rank-1	mAP
+BNNeck	512	91.0	77.6
+BLNeck	512	93.1	81.1
+FBLNeck	768	**93.7**	**83.7**

Table 6. Effect of each component of FBLNeck. The backbone is the body of CDNet. All experiments are conducted on Market1501.

Network	Param(M)	rank-1	mAP
CDNet	1.80	**93.7**	**83.7**
CDNet_std	1.79	91.8	79.7
CDNet_v35	1.61	93.0	82.0

Table 7. The effect of combined patterns and search. std denotes that the CBlock is changed to a single branch. v35 indicates that the kernel combination is fixed to (3,5) with the same number of layers as in CDNet. All experiments are conducted on Market1501.

- Effect of combination and search. To verify the effect of the kernel combination pattern, we change the structure of CBlock to a single branch with the same number of building blocks; thus, the constructed network is called CDNet_std. As shown in Table 7, compared with those of the original CDNet, both rank-1 and mAP of CDNet_std drop dramatically, indicating that our combined pattern learning is much effective. Both CDNet and CDNet_v35 share the same number of blocks in each stage, but the latter fixes the kernel combination to 3×3 and 5×5 in each CBlock. Ultimately, the latter suffers from its lack of combination variety, resulting in suboptimal performance.

5. Conclusion

In this paper, we introduce the Combined Depth Space, and obtain a lightweight and efficient network called CDNet via top-2 sample search, which is effectively for ReID. Our experiments show that the proposed Fine-grained Balance Neck effectively balances the effects of triplet loss and softmax loss. The extensive experiments also further demonstrate that CDNet outperforms state-of-the-art lightweight networks proposed for person re-identification task.

6. Acknowledgements

This work was supported partially by the NSFC(U1911401,U1811461), Guangdong NSF Project (No. 2020B1515120085, 2018B030312002), Guangzhou Research Project (201902010037), and Research Projects of Zhejiang Lab (No. 2019KD0AB03), and the Key-Area Research and Development Program of Guangzhou (202007030004).

References

[1] Xuesong Chen, Canmiao Fu, Yong Zhao, Feng Zheng, Jingkuan Song, Rongrong Ji, and Yi Yang. Salience-guided cascaded suppression network for person re-identification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3300–3310, 2020. 7

[2] Yukang Chen, Gaofeng Meng, Qian Zhang, Shiming Xiang, Chang Huang, Lisen Mu, and Xinggang Wang. Renas: Reinforced evolutionary neural architecture search. In *Proceedings of the IEEE Conference on computer vision and pattern recognition*, pages 4787–4796, 2019. 2

[3] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. 1

[4] Xuanyi Dong and Yi Yang. Searching for a robust neural architecture in four gpu hours. In *Proceedings of the IEEE Conference on computer vision and pattern recognition*, pages 1761–1770, 2019. 2, 4, 7, 8

[5] Song Han, Huizi Mao, and William J. Dally. Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding. In *ICLR 2016 : International Conference on Learning Representations*. 2

[6] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 1

[7] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Identity mappings in deep residual networks. In *European conference on computer vision*, pages 630–645. Springer, 2016. 8

[8] Alexander Hermans, Lucas Beyer, and Bastian Leibe. In defense of the triplet loss for person re-identification, 2017. 2

[9] Ruibing Hou, Bingpeng Ma, Hong Chang, Xinqian Gu, Shiguang Shan, and Xilin Chen. Interaction-and-aggregation network for person re-identification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 9317–9326, 2019. 7

[10] Andrew Howard, Ruoming Pang, Hartwig Adam, Quoc Le, Mark Sandler, Bo Chen, Weijun Wang, Liang-Chieh Chen, Mingxing Tan, Grace Chu, Vijay Vasudevan, and Yukun Zhu. Searching for mobilenetv3. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 1314–1324, 2019. 2

[11] Andrew G Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*, 2017. 2

[12] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4700–4708, 2017. 8

[13] Benoit Jacob, Skirmantas Kligys, Bo Chen, Menglong Zhu, Matthew Tang, Andrew Howard, Hartwig Adam, and Dmitry Kalenichenko. Quantization and training of neural networks for efficient integer-arithmetic-only inference. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2704–2713. 2

[14] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009. 7

[15] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Communications of the ACM*, 60(6):84–90, 2017. 1

[16] Hussam Lawen, Avi Ben-Cohen, Matan Protter, Itamar Friedman, and Lihi Zelnik-Manor. Compact network training for person reid. In *Proceedings of the 2020 International Conference on Multimedia Retrieval*, pages 164–171. 3

[17] Hao Li, Asim Kadav, Igor Durdanovic, Hanan Samet, and Hans Peter Graf. Pruning filters for efficient convnets. In *ICLR (Poster)*, 2016. 2

[18] Wei Li, Xiatian Zhu, and Shaogang Gong. Harmonious attention network for person re-identification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2285–2294, 2018. 3, 7

[19] Ming Lin, Hesen Chen, Xiuyu Sun, Qi Qian, Hao Li, and Rong Jin. Neural architecture design for gpu-efficient networks. *arXiv preprint arXiv:2006.14090*, 2020. 1

[20] Hanxiao Liu, Karen Simonyan, and Yiming Yang. Darts: Differentiable architecture search. *arXiv preprint arXiv:1806.09055*, 2018. 1, 2, 8

[21] Zhuang Liu, Mingjie Sun, Tinghui Zhou, Gao Huang, and Trevor Darrell. Rethinking the value of network pruning. In *International Conference on Learning Representations*, 2018. 2

[22] Hao Luo, Youzhi Gu, Xingyu Liao, Shenqi Lai, and Wei Jiang. Bag of tricks and a strong baseline for deep person re-identification, 2019. 2, 6, 7, 8

[23] Ningning Ma, Xiangyu Zhang, Hai-Tao Zheng, and Jian Sun. Shufflenet v2: Practical guidelines for efficient cnn architecture design. In *Proceedings of the European conference on computer vision (ECCV)*, pages 116–131, 2018. 2

[24] Ruijie Quan, Xuanyi Dong, Yu Wu, Linchao Zhu, and Yi Yang. Auto-reid: Searching for a part-aware convnet for person re-identification. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3750–3759, 2019. 2, 3, 7

[25] Esteban Real, Alok Aggarwal, Yanping Huang, and Quoc V Le. Regularized evolution for image classifier architecture search. In *Proceedings of the aaai conference on artificial intelligence*, volume 33, pages 4780–4789, 2019. 2

[26] Ergys Ristani, Francesco Solera, Roger Zou, Rita Cucchiara, and Carlo Tomasi. Performance measures and a data set for multi-target, multi-camera tracking. In *European Conference on Computer Vision*, pages 17–35. Springer, 2016. 6

[27] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4510–4520, 2018. 2, 7

[28] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 1

[29] Chunfeng Song, Yan Huang, Wanli Ouyang, and Liang Wang. Mask-guided contrastive attention model for person re-identification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1179–1188, 2018. 3

[30] Yifan Sun, Qin Xu, Yali Li, Chi Zhang, Yikang Li, Shengjin Wang, and Jian Sun. Perceive where to focus: Learning visibility-aware part-level features for partial person re-identification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 393–402, 2019. 7

[31] Yifan Sun, Liang Zheng, Yi Yang, Qi Tian, and Shengjin Wang. Beyond part models: Person retrieval with refined part pooling (and a strong convolutional baseline). In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 480–496, 2018. 3, 7

[32] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9, 2015. 1

[33] Mingxing Tan, Bo Chen, Ruoming Pang, Vijay Vasudevan, Mark Sandler, Andrew Howard, and Quoc V Le. Mnasnet: Platform-aware neural architecture search for mobile. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2820–2828, 2019. 2

[34] Guan'an Wang, Shaogang Gong, Jian Cheng, and Zengguang Hou. Faster person re-identification, 2020. 7

[35] Guangcong Wang, Jianhuang Lai, Peigen Huang, and Xiaohua Xie. Spatial-temporal person re-identification. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 8933–8940, 2019. 3

[36] Guanshuo Wang, Yufeng Yuan, Xiong Chen, Jiwei Li, and Xi Zhou. Learning discriminative features with multiple granularities for person re-identification. In *Proceedings of the 26th ACM international conference on Multimedia*, pages 274–282, 2018. 3, 7

[37] Yulong Wang, Xiaolu Zhang, Lingxi Xie, Jun Zhou, Hang Su, Bo Zhang, and Xiaolin Hu. Pruning from scratch. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(7):12273–12280, 2020. 2

[38] Longhui Wei, Shiliang Zhang, Wen Gao, and Qi Tian. Person transfer gan to bridge domain gap for person re-identification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 79–88, 2018. 6

[39] Ancong Wu, Wei-Shi Zheng, Xiaowei Guo, and Jian-Huang Lai. Distilled person re-identification: Towards a more scalable system. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1187–1196, 2019. 2

[40] Bichen Wu, Xiaoliang Dai, Peizhao Zhang, Yanghan Wang, Fei Sun, Yiming Wu, Yuandong Tian, Peter Vajda, Yangqing Jia, and Kurt Keutzer. Fbnet: Hardware-aware efficient convnet design via differentiable neural architecture search. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 10734–10742, 2019. 2, 4

[41] Jiaxiang Wu, Cong Leng, Yuhang Wang, Qinghao Hu, and Jian Cheng. Quantized convolutional neural networks for mobile devices. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4820–4828. 2

[42] Zhaohui Yang, Yunhe Wang, Xinghao Chen, Boxin Shi, Chao Xu, Chunjing Xu, Qi Tian, and Chang Xu. Cars: Continuous evolution for efficient neural architecture search. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1829–1838, 2020. 2

[43] Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. *arXiv preprint arXiv:1605.07146*, 2016. 8

[44] Xiangyu Zhang, Xinyu Zhou, Mengxiao Lin, and Jian Sun. Shufflenet: An extremely efficient convolutional neural network for mobile devices. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6848–6856, 2018. 2, 7

[45] Zhizheng Zhang, Cuiling Lan, Wenjun Zeng, and Zhibo Chen. Densely semantically aligned person re-identification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 667–676, 2019. 3

[46] Feng Zheng, Cheng Deng, Xing Sun, Xinyang Jiang, Xiaowei Guo, Zongqiao Yu, Feiyue Huang, and Rongrong Ji. Pyramidal person re-identification via multi-loss dynamic training. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8514–8522, 2019. 3, 7

[47] Liang Zheng, Liyue Shen, Lu Tian, Shengjin Wang, Jingdong Wang, and Qi Tian. Scalable person re-identification: A benchmark. In *Proceedings of the IEEE international conference on computer vision*, pages 1116–1124, 2015. 6

[48] Kaiyang Zhou, Yongxin Yang, Andrea Cavallaro, and Tao Xiang. Omni-scale feature learning for person re-identification. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3702–3712, 2019. 3, 6, 7, 8

[49] Zhihui Zhu, Xinyang Jiang, Feng Zheng, Xiaowei Guo, Feiyue Huang, Weishi Zheng, and Xing Sun. Aware loss with angular regularization for person re-identification. *arXiv*, pages arXiv–1912, 2019. 3

[50] Barret Zoph and Quoc V Le. Neural architecture search with reinforcement learning. *arXiv preprint arXiv:1611.01578*, 2016. 1, 2