Auto-ReID: Searching for a Part-aware ConvNet for Person Re-Identification

Ruijie Quan[1,2], Xuanyi Dong[1,2], Yu Wu[1,2], Linchao Zhu[1], Yi Yang[1]
[1]University of Technology Sydney [2]Baidu Research
{ruijie.quan, xuanyi.dong, yu.wu-3, linchao.zhu}@student.uts.edu.au;
yi.yang@uts.edu.au

Abstract

Prevailing deep convolutional neural networks (CNNs) for person re-IDentification (reID) are usually built upon the ResNet or VGG backbones, which were originally designed for classification. Because reID has certain differences from classification, the architecture should be modified accordingly. We propose to search for a CNN architecture that is specifically suitable for the reID task. There are three main problems. First, body structural information plays an important role in reID but is not encoded in backbones. Part-based reID models incorporate structure information at the tail of a CNN. Performance relies heavily on human experts and the models are backbone-dependent, requiring extensive human effort when a different backbone is used. Second, Neural Architecture Search (NAS) automates the process of architecture design without human effort, but no existing NAS methods incorporate the structure information of input images. Third, reID is essentially a retrieval task but current NAS algorithms are merely designed for classification. To solve these problems, we propose a retrieval-based search algorithm over a specifically designed reID search space, named Auto-ReID. Our Auto-ReID enables the automated approach to find an efficient and effective CNN architecture that is specifically suitable for reID. Extensive experiments indicate that the searched architecture achieves state-of-the-art performance while requiring less than about 50% parameters and 53% FLOPs compared to others.

1. Introduction

Person re-IDentification (reID) aims to retrieve the images of a person recorded by different surveillance cameras [43, 15, 20, 42]. Due to the success of deep convolutional neural networks (CNNs) in recent years, researchers in this area have mainly focused on improving the representation capability of the features extracted from CNN models [29, 42, 5, 32]. Hundreds of different CNN models have been designed for reID, and the rank-1 accuracy has been improved from 44.4% [42] to 93.8% [32] on the Market-1501 benchmark [42].

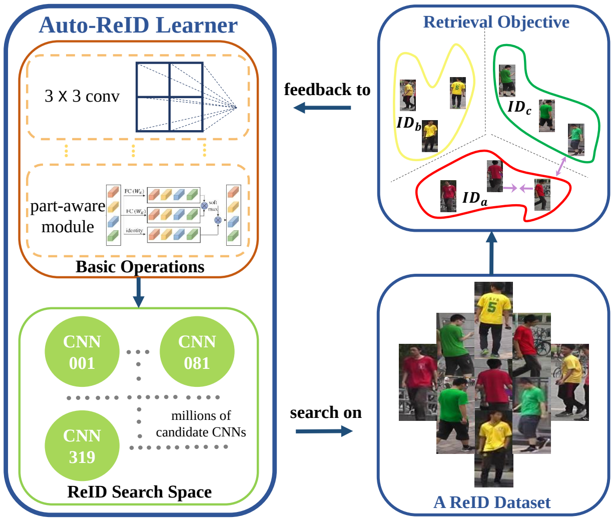

Figure 1. Our Auto-ReID learns to search for a suitable architecture on a specific reID dataset, and it is supervised by the retrieval objective during searching. Auto-ReID finds architecture from a reID search space, which consists of a large number of candidate architectures. These candidates are generated by combining basic operations, such as a 3-by-3 convolutional layer, a 3-by-3 max pooling operation and the proposed part-aware module.

Most recent reID models are based on deep CNNs. They are usually built upon image classification convolutional neural network backbones [39, 15, 29, 30, 32], such as VGG [27], Inception [33], and ResNet [11]. Although classification backbones can be used for retrieval as the inputs of both tasks are images, there are still some differences between the reID task and the classification task. For example, in image classification, the appearance of two objects could be very different, e.g., a cat looks very different from a tree. In contrast, the inputs of the reID task are all person images

[*] Part of this work was done when Ruijie Quan, Xuanyi Dong, and Yu Wu interned at Baidu Research.

with different attributes, e.g., apparel or hair styles. A CNN focusing on recognizing over 1,000 objects [9] should be modified when it is applied to the reID task.

A straightforward method is to manually design reID oriented CNN architecture which is specifically suitable for the reID problem. However, manually designing an exquisite architecture for the reID task may take months [47, 21, 19] even for human experts. This is inefficient and labor intensive. In this paper, we propose an automated approach to search for an optimal CNN architecture that is explicitly suited to the reID task. Our premise is that the CNN backbones designed for classification may have noisy, redundancy and missing components for the reID task. There remain three challenges to automating Neural Architecture Search (NAS) for reID. First, the body structure information plays an important role in reID, which is a major difference between reID and classification [30, 15]. However, no existing NAS approaches search for a CNN architecture that preserves body structural information. Second, reID methods usually encode structural information in a backbone-dependent way. They require extensive manual turning of the hyper-parameters each time a different backbone network is adopted [32, 24]. Third, reID is essentially a retrieval task, but most NAS algorithms are designed for classification. Since retrieval and classification have different objectives, existing NAS algorithms are not directly applicable to the reID problem [47, 21, 18, 19].

In this paper, we propose an approach called Auto-ReID to solve these three challenges. The key contribution of Auto-ReID lies in the design of a new reID search space which encodes body structure information as an operational CNN component. Specifically, we design a part-aware module to enhance the body structure information of a given input feature tensor. Unlike existing part-based reID models, the proposed part-aware module is flexible and can handle features with various input shapes. We use this module as a basic operation for constructing a number of reID candidate architectures. In addition to a typical softmax loss, the proposed Auto-ReID equips the differentiable NAS method [19] with a retrieval loss, making the search results particularly suitable for the reID task. The combination of the proposed reID search space and reID search algorithm enables us to find an efficient and effective architecture for reID in an automated way, as shown in Fig. 1. In sum, our contributions are as follows:

- To the best of our knowledge, this is the first approach that searches neural architectures for the reID task, eliminating the need for effort by human experts in the manual design of CNN models for reID.
- We propose a novel reID search space in which body structure is formulated as a trainable and operational CNN component. The proposed reID search space combines (1) modules that explicitly capture pedestrian body part information, and (2) typical modules that have been used in the standard NAS search space.
- We integrate a retrieval loss into the differentiable NAS algorithm so as to better fit the reID task. A new searching strategy and batch data sampling method are proposed in accordance with the new loss.
- Extensive experiments show that the searched CNN achieves competitive accuracy compared to reID baselines, although this CNN has less than 40% parameters of the reID baselines. By pre-training this CNN on ImageNet, we achieve state-of-the-art performance on four reID benchmarks with only half the number of parameters.

2. Related Works

In this paper, we focus on searching for a reID CNN model with high performance. We first introduce some recent progress in the reID community in Sec. 2.1; and then explain the relationship between our approach with previous NAS methods in Sec. 2.2.

2.1. Person reID

Person reID algorithms have achieved great success due to the deep learning technique [39, 41, 7, 24, 13, 32, 46]. Xiao et al. [39] propose a pipeline to deep feature representations from multiple datasets. Chen et al. [7] design a quadruplet loss to make deep CNN capture both inter-class and intra-class variations. Su et al. [29] propose a pose-driven deep CNN model to explicitly use the human part cues. Saquib et al. [24] take the body joint maps as additional inputs to enable deep CNN to learn pose sensitive representations. Sun et al. [32] leverage a part-based CNN model and a refined part pooling method to learn discriminative part-informed features. Suh et al. [30] utilize bilinear pooling operation to fuse appearance feature and pose feature for reID.

On the one hand, these deep-based reID algorithms [39, 29, 24, 38, 32, 30] heavily rely on the classification CNN backbone, such as VGG [27], Inception [33], and ResNet [11]. These CNN backbone are specifically designed and experimented on classification datasets, which may not align with reID and limit the performance of reID algorithms. On the other hand, they incorporate reID specific domain knowledge to boost the classic CNN models, such as part cues [32, 30], pose [24], and reID specific loss [7, 12]. In this work, we not only inherit the merit of previous reID methods but also overcome their disadvantages. We automatically find a reID specific CNN architecture over a reID search space.

2.2. Neural Architecture Search

Our work is motivated by recent researches on NAS [19, 47, 48, 2, 10], while we focus on searching for a reID model with high performance instead of a classification model. Most of NAS approaches [19, 47, 48, 2, 21] search CNN on a small proxy task and transfer the found CNN structure to another large target task. Zoph et al. [47, 48] apply reinforcement learning to search CNN, while the search cost is more than hundreds of GPU days. Real et al. [22] modify the tournament selection evolutionary algorithm by introducing an age property to favor the younger CNN candidates. Brock et al. [2] and Bender et al. [1] explore the one-shot NAS approaches, which can evaluate individual CNN candidate without training it. Liu et al. [19] relax the discrete search space so as to search CNN in a differentiable way. Benefited from parameter sharing technique [21, 19], we discard the proxy paradigm and directly search a robust CNN on the target reID dataset. Besides, previous NAS algorithms [19, 47, 48, 2, 21, 1, 21] focus on the classification problem. They are generic and can be readily applied to the reID problem. However, without considering reID specific information, such as semantics [14], occlusion [13], pose [24], and part [32], generic NAS approaches can not guarantee that the searched CNN is suitable for reID tasks. In this work, based on an efficient NAS algorithm [19], we adopt two techniques to modify it for the reID problem. We modified the objective function and training strategy to adapt to the reID problem. In addition, we design a part-aware module and integrate it into the standard NAS search space, which could allow us to find a better CNN and advance the study of the NAS search space.

3. Methodology

In this section, we will show how to search for a reID CNN with high performance. We will first introduce the preliminary background of NAS in Sec. 3.1. Then we propose a new search algorithm for reID, introduced in Sec. 3.2. Furthermore, we design a new reID search space in Sec. 3.3, which integrates our proposed part-aware module and the standard NAS search space. Lastly, we discuss some future direction for reID in Sec. 3.4.

3.1. Preliminaries

Most NAS approaches stack multiple copies of a neural cell to construct a CNN model [48, 18, 22]. A neural cell consists of several different kinds of layers, taking output tensors from previous cells and generating a new output tensor. We follow previous NAS approaches [47, 48, 19] to search for the topology structure of neural cells.

Specifically, a neural cell can be viewed as a directed acyclic graph (DAG) with B blocks. Each block has three steps: (1) take two tensors as inputs, (2) apply two operations on these two tensors, respectively, (3) sum these two tensors. The applied operation is selected from an operation candidate set $\mathcal{O}$. Following some previous works [21], we use the following operations in our $\mathcal{O}$: (1) 3×3 max pooling, (2) 3×3 average pooling, (3) 3×3 depth-wise separable convolution, (4) 3×3 dilated convolution, (5) zero operation (none), (6) identity mapping. The i-th block in the c-th neural cell can be represented as a 4-tuple, i.e., (I_{i1}^c, I_{i2}^c, $\mathcal{O}_{i1}^c$, $\mathcal{O}_{i2}^c$). Besides, the output tensor of the i-th block in the c-th neural cell is:

Algorithm 1 The Auto-ReID Algorithm

```
Input: the architecture parameter $\alpha$ and the operation parameter $\omega$; the training set $\mathbb{D}_T$ and the evaluation set $\mathbb{D}_E$; a class-balance data sampler;
  Split $\mathbb{D}_T$ into the search training set $\mathbb{D}_{train}$ and the search validation set $\mathbb{D}_{val}$
  while not converged do
    Use the sampler to get batch data from $\mathbb{D}_{train}$
    Update $\omega$ via the retrieval loss in Eq. (5)
    Use the sampler to get batch data from $\mathbb{D}_{val}$
    Update $\alpha$ via the retrieval loss in Eq. (5)
  end while
  Obtain the final CNN from $\alpha$ following the strategy in [19]
  Optimize this CNN on the training set $\mathbb{D}_T$ by the standard reID training strategy
  Evaluate the trained CNN on the evaluation set $\mathbb{D}_E$
```

$$I_i^c = \mathcal{O}_{i1}^c(I_{i1}^c) + \mathcal{O}_{i2}^c(I_{i2}^c), \tag{1}$$

where $\mathcal{O}_{i1}^c$ and $\mathcal{O}_{i2}^c$ are selected operations from $\mathcal{O}$ for the i-th block. I_{i1}^c and I_{i2}^c are selected from the candidate input tensors, which consists of output tensors from the last two neural cells and output tensors from the previous block in the current cell.

To search for the choices of $\mathcal{O}_{i1}^c$ and I_{i1}^c in Eq. (1), we relax the categorical choice of a particular operation as a softmax over all possible operations following [19]:

$$\mathcal{O}_{i1}^c(I_{i1}^c) = \sum_{H \in \mathcal{I}_i^c} \sum_{o \in \mathcal{O}} \frac{\exp(\alpha_o^{(H,i)})}{\sum_{o' \in \mathcal{O}} \exp(\alpha_{o'}^{(H,i)})} o(H), \tag{2}$$

where $\alpha = \{\alpha_o^{(H,i)}\}$ represents the topology structure for a neural cell, named as ***architecture parameters***. Denote the parameters of all operations in $\mathcal{O}$ as ω (named as ***operation parameters***), a typical differentiable NAS approach [19] jointly train ω on the training set and α on the validation set. After training, the strength of H to I_i^c is defined as $\max_{o \in \mathcal{O}, o \neq \text{none}} \frac{\exp(\alpha_o^{(H,i)})}{\sum_{o' \in \mathcal{O}} \exp(\alpha_{o'}^{(H,i)})}$. The $H \in \mathcal{I}_i^c$ with the maximum strength is selected as I_{i1}^c, and the operation with the maximum weight for I_{i1}^c is selected as $\mathcal{O}_{i1}^c$. This paradigm [19] is designed for the classification problem. Inspired from them, we apply several improvements to adapt this paradigm into the person reID problem.

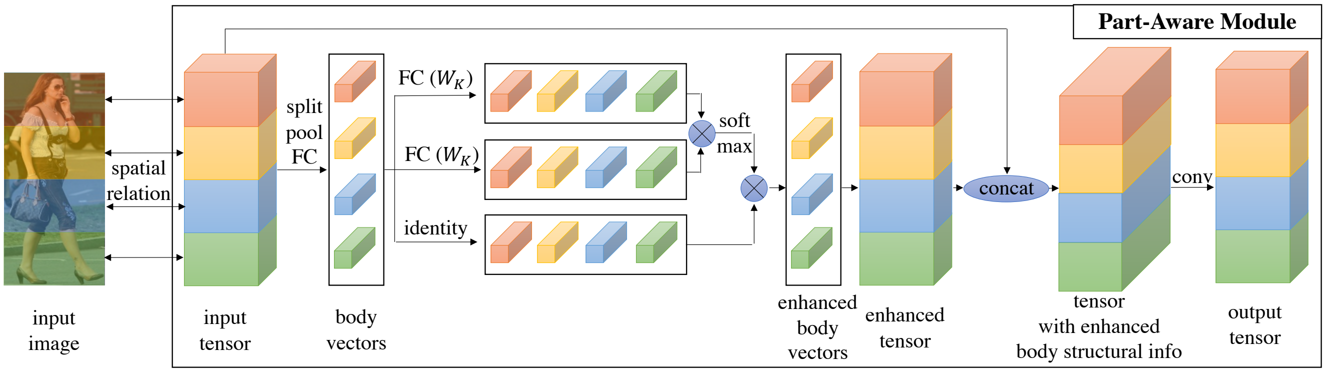

Figure 2. The proposed part-aware module for the reID search space. Given a pedestrian feature tensor, this module can integrate human body structural cues into the input tensor. It first vertically splits the input feature tensor into $M = 4$ body part features, and then averages each part tensor into a vector and uses a linear layer to transform each of them into a new part feature vector, denoted as "body vectors". These M part vectors are interacted via a self-attention mechanism, and each part vectors could include more body part specific information. Later, these M vectors are repeated and concatenated to recover them into the same spatial shape as the input tensor, named as "enhanced tensor". Finally, we fuse this global feature tensor and the original input tensor by a one-by-one convolutional layer.

3.2. ReID Search Algorithm

Prevailing NAS approaches focus on searching for a well-performed architecture in the classification task, in which the softmax with cross-entropy loss is applied to optimizing both α and ω [19, 48]. In contrast, reID tasks aim to learn a discriminative feature extractor during training, so that the extracted feature can retrieve images of the same identity during evaluation. Simply inheriting the cross-entropy loss can not guarantee a good retrieve performance. We need to incorporate reID specific knowledge into the searching algorithm.

Network Structure. We use the macro structure of ResNet [11] for our reID backbone, where each residual layer is replaced by a neural cell [1]. We search the topology structure of neural cells. Denote the feature extracted from the backbone as $\boldsymbol{f}$, we use one embedding layer to transfer the feature $\boldsymbol{f}$ into $\boldsymbol{g}$ following the common setting of reID [32], and we use another linear transformation layer to map the feature $\boldsymbol{g}$ into the logits $\boldsymbol{h}$ with the dimension of C, where C denotes the number of training identities. Two dropout layers [28] are added between $\boldsymbol{f}$&$\boldsymbol{g}$ and $\boldsymbol{g}$&$\boldsymbol{h}$, respectively.

Objective. The classification model usually applies the softmax with cross-entropy loss on $\boldsymbol{h}$ as follows:

$$L_s = \sum_{i=1}^{N} -\log \frac{\exp(\boldsymbol{h}_i[c])}{\sum_{c'=1}^{C} \exp(\boldsymbol{h}_i[c'])}, \tag{3}$$

where $\boldsymbol{h}_i$ indicates the feature $\boldsymbol{h}$ of the i-th sample, and $\boldsymbol{h}_i[c]$ indicates the c-th element in $\boldsymbol{h}_i$. N is the number of samples during training. The reID model usually applies the triplet loss as:

$$L_t = \sum_{i=1}^{N} -||\boldsymbol{f}_i - \boldsymbol{f}_i^p||_2 + ||\boldsymbol{f}_i - \boldsymbol{f}_i^n||_2, \tag{4}$$

where $\boldsymbol{f}_i$ indicates the feature $\boldsymbol{f}$ of the i-th sample. $\boldsymbol{f}_i^p$ indicates the hardest positive feature of $\boldsymbol{f}_i$. In another word, $\boldsymbol{f}_i^p$ is another feature with the maximum L2 distance of $\boldsymbol{f}_i$ and the same identity of $\boldsymbol{f}_i$ in one batch. $\boldsymbol{f}_i^n$ is the hardest negative feature of $\boldsymbol{f}_i$. In another word, $\boldsymbol{f}_i^n$ is another feature with the minimum L2 distance of $\boldsymbol{f}_i$ and the different identity of $\boldsymbol{f}_i$ in one batch. Since the triplet loss is sensitive to the batch data, we should carefully sample training data in each batch. We adopt a class-balance data sampler to sample batch data for triplet loss. This sampler first sample uniformly sample some identities, and then, for each identity, it randomly sample the same number of images, To align with the reID problem and leverage the mutual benefit from the cross-entropy and triplet losses, we consider a mixture retrieval loss of L_s and L_t as follows:

$$L_{ret} = \lambda L_s + (1 - \lambda) L_t, \tag{5}$$

where $\lambda \in [0, 1]$ is a weight balancing L_s and L_t.

We show our overall algorithm (Auto-ReID) in Alg. 1. We first search for a robust reID model by alternatively optimizing α with L_t and ω with L_{ret}. The searched CNN is derived from α based on the same strategy as in [19]. After we find a robust CNN for the reID task, we train and evaluate this CNN in the standard way.

3.3. ReID Search Space with Part-aware Module

The search space covers all possible candidate CNN to be found, and is important for NAS. A standard search space

[1] The detailed configuration will be introduced in experiments

in NAS is “NASNet search space” [48], which contains different kinds of convolutional layers, different kinds of pooling layers, etc. None of these layers can explicitly handle pedestrian information, which requires a delicate design and some unique operations. In this paper, we take the first step to explore a search space that fits the reID problem.

Motivated by the fact that body part information can improve the performance of a reID model [32, 29], we design a part-aware module and combine it with a common search space ($\mathcal{O}$) to construct our reID search space $\mathcal{O}_{reid}$:

- ***part-aware module***
- 3x3 depth-wise separable convolution
- 3x3 dilated convolution
- 3x3 average pooling
- 3x3 max pooling
- skip connection
- zero operation

The part-aware module is shown in Fig. 2. Given an input feature tensor $\boldsymbol{F}$, we first split it into M parts vertically, where we show an example of $M = 4$ in Fig. 2. After we obtain the part features, we average pool each part feature over the spatial dimension and apply a linear transformation to the pooled features, and can thus obtain M local body part feature vectors. Then, we apply a self-attention mechanism [34] on these M part feature vectors. In this way, we can incorporate global information into each part vectors to enhance its body structure cues. Later, we repeat each part vector into its original spatial shape and and concatenate the repeated part features vertically into a body structure enhanced feature tensor. Finally, we fuse this part-aware tensor and the original input feature tensor via channel-wise concatenate, and apply a one-by-one convolutional layer on this fusion tensor to generate the output tensor. Our designed part-aware module can capture useful body part cue and integrate this structural information into the input features. Besides, the parameter size and number of calculation of the proposed part-aware module are similar to the 3x3 depth-wise separable convolution, and thus will not affect the efficiency of the found CNN compared to use a standard NAS search space.

3.4. Discussion

Researchers has trend to move their focus from manual architecture design to automated architecture design in many areas, e.g., classification [48, 19] and segmentation [6]. In the reID community, the breakthrough of the reID performance is usually benefited from improvements on the CNN architecture. We present the first effort towards applying automated machine learning to reID. After so many different architectures are proposed for reID [39, 29, 24, 32, 30], it is more and more difficult to manually find a better architecture. It is time to automatically find a good reID architecture, and to our knowledge, this is the first time that an automated algorithm has matched state-of-the-art performance using architecture search techniques on reID problems.

4. Experiments

We empirically evaluate the proposed method in this section. We will first introduce the used datasets in Sec. 4.1 and implementation details in Sec. 4.2. Then, we will ablatively study different aspects of our Auto-ReID algorithm in Sec. 4.3, and also compare the CNN found by our approach with other state-of-the-art algorithms in Sec. 4.4. Lastly, we make some qualitative analysis in Sec. 4.5.

4.1. Datasets and Evaluation Metrics

Market-1501 [42] is a large-scale person reID dataset which contains 19,372 gallery images, 3,368 query images and 12,396 training images collected from six cameras. There are 751 identities in training set and 750 identities in the test set and they have no overlap. Every identity in the training set has 17.2 images on average.

DukeMTMC-reID [44] is a subset of DukeMTMC [23] and contains 36,411 images of 1,812 identities captured by eight high-resolution cameras. The pedestrian images are cropped by hand-drawn bounding boxes. It consists of 16,522 training images of 702 identities, 2,228 query images and 17,661 gallery images of the other 702 identities.

CUHK03 [15] consists of 1,467 identities and 28,192 bounding boxes. There are 26,264 images of 1,367 identities are used for training and 1,928 images of 100 identities are used for testing. We use the new protocol to split the training and test data as proposed by [45].

MSMT17 [36] is currently the largest person reID dataset, which contains 126,441 images of 4,101 identities in 15 cameras. This dataset is composed of the training set, which contains 32,621 bounding boxes of 1,041 identities and the test set including 93,820 bounding boxes of 3,060 identities. From the test set, 11,659 images are used as query images and the other 82,161 bounding boxes are used as gallery images. This challenging dataset has more complex scenes and backgrounds, e.g., indoor and outdoor scenes, than others.

Evaluation Metrics. To evaluate the performance of our Auto-ReID and compare with other reID methods, we report two common evaluation metrics: the cumulative matching characteristics (CMC) at rank-1, rank-5 and rank-10 and mean average precision (mAP) on the above four benchmarks following the common settings [42, 36].

Architectures	Pre-train	mAP	Rank-1	Rank-5	Rank-10	Params(M)	FLOPs(G)
ResNet-18 [11]	×	66.0	85.2	94.6	96.5	11.6	1.7
ResNet-34 [11]	×	67.3	87.0	94.8	96.5	21.7	3.4
ResNet-50 [11]	×	68.1	86.4	95.6	96.8	25.1	3.8
DARTS [19]	×	64.5	85.1	94.2	96.2	9.1	1.7
Baseline	×	68.5	87.0	95.4	97.1	11.9	2.0
Baseline + ReID Search Space	×	71.3	**90.0**	96.5	97.7	10.9	1.7
Retrieval + ReID Search Space	×	**72.7**	89.7	**96.7**	**98.0**	11.4	1.8

Table 1. We analyze the effect of each component in our Auto-ReID. All CNN models are trained in the same strategy and do not use ImageNet pre-training for initialization.

4.2. Implementation Details

Search Configurations. For each dataset, we randomly select 50% images from its official training set as the search training set $\mathbb{D}_{train}$ and others as the search validation set $\mathbb{D}_{val}$. We choose the ResNet macro structure to construct the overall network. This network has a 3x3 convolutional head and 4 blocks sequentially, where each block has several neural cells. We denote the number of cells in each block is $\boldsymbol{l}_1$, $\boldsymbol{l}_2$, $\boldsymbol{l}_3$, and $\boldsymbol{l}_4$. We denote $\boldsymbol{l} = [\boldsymbol{l}_1, \boldsymbol{l}_2, \boldsymbol{l}_3, \boldsymbol{l}_4]$. We denote the channel of the first convolutional layer as C, and each block will double the number of channels. The first cell in the 2-th, 3-th, and 4-th block is a reduction cell, and other cells are the normal cell [18, 19]. By default, we use C=32 and $\boldsymbol{l} = [2, 2, 2, 2]$ to search a suitable CNN model.

During searching, we use a input size of 256 × 128, a batch size of 16, the total epoch of 200. We use momentum SGD to optimize ω with the initial learning rate of 0.1 and decrease it to 0.001 in a cosine scheduler. The momentum for SGD is set as 0.9. We use Adam to optimize α with the initial learning rate of 0.02, which is decayed by 10 at 60-th and 150-th epoch. The weight decay for both SGD and Adam is set as 0.0005. The λ is set as 0.5.

In tables, we use "Baseline" to denote the baseline searching algorithm (DARTS 1st order [19]) on the base classification search space. We use "Baseline + ReID Search Space" to denote the baseline searching algorithm on the proposed reID search space. We use "Retrieval + ReID Search Space" to denote the proposed retrieval-based searching algorithm on the reID search space.[2] "Retrieval + ReID Search Space" costs about 1 day to finish one searching procedure on a single NVIDIA Tesla V100 GPU. We run experiments 4 times and select the best architecture following [19], and therefore, the total searching cost is about 4 GPU days.

Configurations			Rank-1	mAP	Params (M)
Search	Train				
	C	$\boldsymbol{l}$			
C=16 $\boldsymbol{l}$=[2,2,2,2]	16	[2,2,2,2]	81.2	58.9	1.1
		[3,4,6,3]	79.5	53.5	1.7
	32	[2,2,2,2]	86.4	66.0	3.8
		[3,4,6,3]	85.4	62.7	5.9
C=16 $\boldsymbol{l}$=[3,4,6,3]	16	[2,2,2,2]	77.3	54.0	1.0
		[3,4,6,3]	81.2	56.2	1.4
	32	[2,2,2,2]	85.9	66.0	3.1
		[3,4,6,3]	87.2	66.5	4.9
C=32 $\boldsymbol{l}$=[2,2,2,2]	16	[2,2,2,2]	80.7	58.3	1.2
		[3,4,6,3]	80.3	56.7	1.9
	32	[2,2,2,2]	87.6	68.3	4.1
		[3,4,6,3]	85.1	64.9	6.6
C=32 $\boldsymbol{l}$=[3,4,6,3]	16	[2,2,2,2]	78.0	55.4	1.2
		[3,4,6,3]	80.0	55.5	1.8
	32	[2,2,2,2]	85.6	66.2	3.9
		[3,4,6,3]	86.5	64.6	6.2

Table 2. We use different configurations in the searching and the training procedures. Apart from the C and $\boldsymbol{l}$, we keep other hyper-parameters the same for different configurations.

Training Configurations. In the training phrase, we use an input size of 384×128, C=64, and $\boldsymbol{l} = [2, 2, 2, 2]$. Following previous works, we use random horizontal flipping and cropping for data augmentation. We set λ as 0.5 for the retrieval loss. For training from scratch, in one batch, our class-balance data sampler will first random select 8 identities and then random sample 4 images for each identity. When using ImageNet pre-training models, it randomly samples 16 identities and then samples 4 images for each identity. We train the model for 240 epochs, using Adam as the optimizer with a momentum of 0.9 and a weight decay of 0.0005. We start the learning rate from 0.0035 and decay it by 10 at the 80-th and 140-th epochs. All experiments are based on PyTorch 1.0.

[2] We search the reduction cell on the reID search space while searching the normal cell on the baseline search space. This strategy is due to (1) we observe that if searching the normal cell in reID search space, the found cell do not have any part-aware module; (2) few part-aware modules are enough to boost the performance, more part-aware modules only bring negligible benefit; (3) only searching the reduction cell on the reID search space can save the searching cost without sacrificing the accuracy.

Methods	Venue	Backbone	Params (M)	Market-1501				DukeMTMC-reID			
				R-1	R-5	R-10	mAP	R-1	R-5	R-10	mAP
TriNet [12]	*arXiv17*	ResNet-50	25.1	84.9	94.2	-	69.1	-	-	-	-
AOS [13]	*CVPR18*	ResNet-50	> 25.1	86.4	-	-	70.4	79.1	-	-	62.1
DPFL [8]	*ICCV17-W*	Inception-V3	> 23.9	88.6	-	-	72.6	79.2	-	-	60.6
MLFN [3]	*CVPR18*	ResNeXt-50	> 25.0	90.0	-	-	74.3	81.0	-	-	62.8
DuATM [26]	*CVPR18*	DenseNet-121	> 8.0	91.4	97.0	-	76.6	81.8	90.1	-	64.5
GSRW [25]	*CVPR18*	ResNet-50	> 25.1	92.7	96.9	98.1	82.5	80.7	88.5	90.8	66.4
DNN_CRF [4]	*CVPR18*	ResNet-50	26.1	93.5	97.7	-	81.6	84.9	92.3	-	69.5
PCB [32]	*ECCV18*	ResNet-50	27.2	92.3	97.2	98.2	77.4	81.9	89.4	91.6	65.3
Mancs [35]	*ECCV18*	ResNet-50	> 25.1	93.1	-	-	82.3	84.9	-	-	71.8
Baseline	-	-	11.9	93.8	98.1	98.8	83.4	86.3	93.6	95.3	71.7
Auto-ReID			13.1	**94.5**	**98.5**	**99.0**	**85.1**	**88.5**	**94.1**	**95.7**	**75.1**
TriNet (RK) [12]	*arXiv17*	ResNet-50	25.1	86.7	-	-	81.1	-	-	-	-
AOS (RK) [13]	*CVPR2018*	ResNet-50	> 25.1	88.7	-	-	83.3	-	-	-	-
AACN (RK) [40]	*CVPR2018*	GoogleNet	8	88.7	-	-	83.0	-	-	-	-
PSE+ECN (RK) [24]	*CVPR2018*	ResNet-50	> 25.1	90.3	-	-	84.0	85.2	-	-	79.8
PCB (RK) [32]	*ECCV18*	ResNet-50	27.2	95.1	-	-	91.9	-	-	-	-
Baseline (RK)	-	-	11.9	94.8	97.6	98.3	93.5	90.6	**95.0**	**96.3**	88.6
Auto-ReID (RK)			13.1	**95.4**	**97.9**	**98.5**	**94.2**	**91.4**	94.7	96.1	**89.2**

Table 3. Comparisons with state-of-art reID models on Market-1501 and DukeMTMC-reID. "R-1" indicates the rank-1 accuracy, and so as "R-5" and "R-10". "RK" indicates the re-ranking technique [45].

4.3. Ablation Study

To investigate the effect of each component in our Auto-ReID, we perform extensive ablation studies on the Market-1501 dataset. We show the results in Table 1 and Table 2.

We compare four searching options in Table 1 without using ImageNet pre-training. We make several observations: (1) DARTS [19] is searched on a small classification dataset, in which the found CNN is worse than a simple reID model based on ResNet-18. (2) By directly searching on the reID dataset ("Baseline"), we find a better CNN, which outperforms ResNet-18 by 2% mAP but has similar numbers of parameters and FLOPs. (3) By searching on the proposed reID search space, the performance of the searched CNN can be significantly improved by about 2.8% mAP and 3% rank-1 accuracy. (4) Replacing the classification searching loss with the retrieval searching loss, we can obtain a much better CNN. This CNN further boost the mAP by 1.4%. In addition, we find that this CNN contains more part-aware modules than the CNN searched via classification. Compared to a strong ResNet-50 reID baseline, the CNN found by our Auto-ReID achieves a much higher mAP (72.7 vs. 68.1), whereas the parameters and FLOPs are only about half of that of ResNet-50.

The effect of different configurations. We try different configurations in the searching and training procedure. We use the "Baseline + ReID Search Space" and keep other hyper-parameters the same. Results are shown in Table 2. First, a higher number of channels during training will yield better accuracy and mAP. Second, more layers (a larger $\boldsymbol{l}$) can result in a better performance only when the value of $\boldsymbol{l}$ during searching is the same as the value during training. In another word, a neural cell searched by $\boldsymbol{l} = [2, 2, 2, 2]$ is more suitable for an architecture with $\boldsymbol{l} = [2, 2, 2, 2]$. Third, if we use C=64 or $\boldsymbol{l}$=[3,4,6,3] for experiments in Table 1, we might find a better CNN. Consider the efficiency, we use a small C and $\boldsymbol{l}$ during searching.

4.4. Comparison with State-of-the-art ReID Models

Since all state-of-the-art reID algorithms pre-train their models on ImageNet, we also pre-train our searched CNN on ImageNet for a fair comparison. Most our results in Table 3, Table 4 and Table 5 use ImageNet pre-training by default. "Auto-ReID" in these tables refers to the best CNN found by our searching algorithm on Market-1501.

Results on Market-1501. Table 3 compares our method with other state-of-the-art reID models. Our baseline searching algorithm finds a CNN, which achieves a rank-1 accuracy of 93.8% and a mAP of 83.4%, it outperforms other state-of-the-art reID algorithms. Our Auto-ReID further boosts the performance of "Baseline". The CNN found by our Auto-ReID achieves a rank-1 accuracy of 94.5% and a mAP of 85.1%. Note that this CNN reduces the parameters of ResNet-50 based reID models by more than 45%, whereas it obtains a much higher accuracy and mAP than them. This experiment demonstrates that our automated architecture search approach can find an efficient and effective model, which successfully removes the noises, redundancies, and missing components in other typical backbones.

Methods	Labeled		Detected	
	Rank-1	mAP	Rank-1	mAP
BOW+XQDA [42]	7.9	7.3	6.4	6.4
LOMO+XQDA [17]	14.8	13.6	12.8	11.5
SVDNet [31]	40.9	37.8	41.5	37.3
HA-CNN [16]	44.4	41.0	41.7	38.6
AOS [13]	-	-	47.7	43.3
MLFN [3]	54.7	49.2	52.8	47.8
PCB [32]	-	-	59.7	53.2
Mancs [35]	69.0	63.9	65.5	60.5
Baseline	75.0	70.1	70.5	66.5
Auto-ReID	**77.9**	**73.0**	**73.3**	**69.3**

Table 4. Comparison of accuracy and mAP with the state-of-the-art reID models on CUHK03. Note that we use the new evaluation protocol reported in [45].

Methods	Rank-1	Rank-5	Rank-10	mAP
GoogleNet [33]	47.6	65.0	71.8	23.0
PDC [29]	58.0	73.6	79.4	29.7
GLAD [37]	61.4	76.8	81.6	34.0
PCB [32]	68.2	81.2	85.5	40.4
Baseline	74.7	86.1	89.5	48.2
Auto-ReID	**78.2**	**88.2**	**91.1**	**52.5**

Table 5. Comparison of accuracy and mAP with the state-of-the-art reID models on MSMT17.

Note that our Auto-ReID is orthogonal to other reID techniques, such as re-ranking [45], as shown in Table 3. Using the same augmentation technique [45], our Auto-ReID also outperforms other reID models. For example, "PCB (RK)" achieves the mAP of 91.9%, whereas our "Auto-ReID (RK)" achieves the mAP of 94.2%, which is higher than it by 2.3%. Although other techniques can further improve the performance of Auto-ReID, we do not discuss more since it is not the focus of this paper.

To validate the transferable ability of the searched architecture, we apply the Market-1501 searched architecture to other reID datasets. **Results on DukeMTMC-reID:** This CNN architecture is superior to the state-of-the-art models by 3.3% mAP and 3.6% accuracy. **Results on CUHK03 in Table 4:** There are two types of person bounding boxes: manually labeled and automatically detected. On both settings, our Auto-ReID obtains a significantly higher accuracy and mAP than our models. **Results on MSMT17 in Table 5:** Auto-ReID outperforms the PCB [32] by the mAP of 12% and the accuracy of 10%. In sum, we made the following conclusions: (1) the searched architecture on one reID dataset can be successfully transferred into another dataset; (2) the automated CNN by our Auto-ReID consistently outperforms state-of-the-art reID models on all four datasets. (3) Our Auto-ReID consistently outperforms the baseline searching on all four datasets.

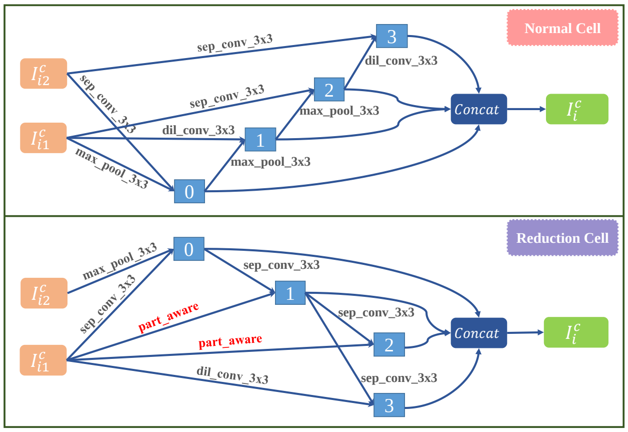

Figure 3. The normal cell and the reduction cell used in Table 3, Table 4, and Table 5. This topology structure is complex and hard to designed by human expert.

Since MSMT17 is much larger than Market-1501, it might be possible to find a better CNN. We also experiment to search on MSMT17 and transfer the found architecture to Market-1501. The architecture found on MSMT17 obtains a slightly worse result than "Auto-ReID" in Table 3. This might imply that we should search for a different reID model for a different dataset. However, since most of the hyper-parameters are tuned on Market-1501, it requires to tune these parameters when applying our searching algorithm to other datasets.

Future Work. Our Auto-ReID takes the first step to automate the reID model design. The proposed reID search space only considers one possible reID specific module. More carefully designed basic reID modules can help us find a better reID architecture. In addition, the proposed searching algorithm is a simple extension to the existing NAS algorithm. We should consider more reID specific knowledge to design more efficient and effective searching algorithms.

4.5. Visualization

To better understand what we found during searching, we display one of our searched architectures in Fig. 3. We show both the normal cell and the reduction cell. These automatically discovered cells are quite complex and are difficulty to be found by a human expert with manual tuning. Manually design a similar architecture as in Fig. 3 will cost months, which is inefficient and labor intensive. This further shows that it is necessary to automate the reID architecture design.

5. Conclusion

In this paper, we propose an automated neural architecture search for the reID tasks, and we name our method as Auto-ReID. The proposed Auto-ReID involves a new reID search space and a new retrieval-based searching algorithm. The proposed reID search space incorporates body structure information into the candidate CNN in the search space. Specifically, it combines a typical classification search space and a novel part-aware module. Since reID is essentially a retrieval task but current NAS algorithms are merely designed for classification. We equip DARTS with a retrieval loss, making it particularly suitable for reID. In experiments, the CNN architecture found by our Auto-ReID significantly outperforms all state-of-the-art reID models on four benchmarks.

References

[1] G. Bender, P.-J. Kindermans, B. Zoph, V. Vasudevan, and Q. Le. Understanding and simplifying one-shot architecture search. In *ICML*, 2018. 3

[2] A. Brock, T. Lim, J. M. Ritchie, and N. Weston. SMASH: one-shot model architecture search through hypernetworks. In *ICLR*, 2018. 3

[3] X. Chang, T. M. Hospedales, and T. Xiang. Multi-level factorisation net for person re-identification. In *CVPR*, 2018. 7, 8

[4] 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. 7

[5] D. Chen, Z. Yuan, B. Chen, and N. Zheng. Similarity learning with spatial constraints for person re-identification. In *CVPR*, 2016. 1

[6] L.-C. Chen, M. Collins, Y. Zhu, G. Papandreou, B. Zoph, F. Schroff, H. Adam, and J. Shlens. Searching for efficient multi-scale architectures for dense image prediction. In *NeurIPS*, 2018. 5

[7] W. Chen, X. Chen, J. Zhang, and K. Huang. Beyond triplet loss: a deep quadruplet network for person re-identification. In *CVPR*, 2017. 2

[8] Y. Chen, X. Zhu, and S. Gong. Person re-identification by deep learning multi-scale representations. In *ICCV-W*, 2017. 7

[9] 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. 2

[10] X. Dong and Y. Yang. Searching for a robust neural architecture in four gpu hours. In *CVPR*, 2019. 3

[11] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 1, 2, 4, 6

[12] A. Hermans, L. Beyer, and B. Leibe. In defense of the triplet loss for person re-identification. *arXiv preprint arXiv:1703.07737*, 2017. 2, 7

[13] H. Huang, D. Li, Z. Zhang, X. Chen, and K. Huang. Adversarially occluded samples for person re-identification. In *CVPR*, 2018. 2, 3, 7, 8

[14] M. M. Kalayeh, E. Basaran, M. Gökmen, M. E. Kamasak, and M. Shah. Human semantic parsing for person re-identification. In *CVPR*, 2018. 3

[15] W. Li, R. Zhao, T. Xiao, and X. Wang. Deepreid: Deep filter pairing neural network for person re-identification. In *CVPR*, 2014. 1, 2, 5

[16] W. Li, X. Zhu, and S. Gong. Harmonious attention network for person re-identification. In *CVPR*, 2018. 8

[17] S. Liao, Y. Hu, X. Zhu, and S. Z. Li. Person re-identification by local maximal occurrence representation and metric learning. In *CVPR*, 2015. 8

[18] C. Liu, B. Zoph, M. Neumann, J. Shlens, W. Hua, L.-J. Li, L. Fei-Fei, A. Yuille, J. Huang, and K. Murphy. Progressive neural architecture search. In *ECCV*, 2018. 2, 3, 6

[19] H. Liu, K. Simonyan, and Y. Yang. DARTS: Differentiable architecture search. In *ICLR*, 2019. 2, 3, 4, 5, 6, 7

[20] S. Paisitkriangkrai, C. Shen, and A. Van Den Hengel. Learning to rank in person re-identification with metric ensembles. In *CVPR*, 2015. 1

[21] H. Pham, M. Y. Guan, B. Zoph, Q. V. Le, and J. Dean. Efficient neural architecture search via parameter sharing. In *ICML*, 2018. 2, 3

[22] E. Real, A. Aggarwal, Y. Huang, and Q. V. Le. Regularized evolution for image classifier architecture search. In *AAAI*, 2019. 3

[23] E. Ristani, F. Solera, R. Zou, R. Cucchiara, and C. Tomasi. Performance measures and a data set for multi-target, multi-camera tracking. In *ECCV-W*, 2016. 5

[24] M. Saquib Sarfraz, A. Schumann, A. Eberle, and R. Stiefelhagen. A pose-sensitive embedding for person re-identification with expanded cross neighborhood re-ranking. In *CVPR*, 2018. 2, 3, 5, 7

[25] 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. 7

[26] 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. 7

[27] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In *ICLR*, 2015. 1, 2

[28] G. H. A. K. I. S. Srivastava, Nitish and R. Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *JMLR*, 2014. 4

[29] 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. 1, 2, 5, 8

[30] Y. Suh, J. Wang, S. Tang, T. Mei, and K. Mu Lee. Part-aligned bilinear representations for person re-identification. In *ECCV*, 2018. 1, 2, 5

[31] Y. Sun, L. Zheng, W. Deng, and S. Wang. Svdnet for pedestrian retrieval. In *ICCV*, 2017. 8

[32] 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. 1, 2, 3, 4, 5, 7, 8

[33] 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. 1, 2, 8

[34] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin. Attention is all you need. In *NeurIPS*, 2017. 5

[35] 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. 7, 8

[36] L. Wei, S. Zhang, W. Gao, and Q. Tian. Person trasfer gan to bridge domain gap for person re-identification. In *CVPR*, 2018. 5

[37] L. Wei, S. Zhang, H. Yao, W. Gao, and Q. Tian. Glad: Global-local-alignment descriptor for pedestrian retrieval. In *ACM Multimedia*, 2017. 8

[38] Y. Wu, Y. Lin, X. Dong, Y. Yan, W. Ouyang, and Y. Yang. Exploit the unknown gradually: One-shot video-based person re-identification by stepwise learning. In *CVPR*, 2018. 2

[39] T. Xiao, H. Li, W. Ouyang, and X. Wang. Learning deep feature representations with domain guided dropout for person re-identification. In *CVPR*, 2016. 1, 2, 5

[40] J. Xu, R. Zhao, F. Zhu, H. Wang, and W. Ouyang. Attention-aware compositional network for person re-identification. In *CVPR*, 2018. 7

[41] L. Zhang, T. Xiang, and S. Gong. Learning a discriminative null space for person re-identification. In *CVPR*, 2016. 2

[42] L. Zheng, L. Shen, L. Tian, S. Wang, J. Wang, and Q. Tian. Scalable person re-identification: A benchmark. In *CVPR*, 2015. 1, 5, 8

[43] W.-S. Zheng, S. Gong, and T. Xiang. Transfer re-identification: From person to set-based verification. In *CVPR*, 2012. 1

[44] Z. Zheng, L. Zheng, and Y. Yang. Unlabeled samples generated by gan improve the person re-identification baseline in vitro. In *ICCV*, 2017. 5

[45] Z. Zhong, L. Zheng, D. Cao, and S. Li. Re-ranking person re-identification with k-reciprocal encoding. In *CVPR*, 2017. 5, 7, 8

[46] Z. Zhong, L. Zheng, Z. Luo, S. Li, and Y. Yang. Invariance matters: Exemplar memory for domain adaptive person re-identication. In *CVPR*, 2019. 2

[47] B. Zoph and Q. V. Le. Neural architecture search with reinforcement learning. In *ICLR*, 2017. 2, 3

[48] B. Zoph, V. Vasudevan, J. Shlens, and Q. V. Le. Learning transferable architectures for scalable image recognition. In *CVPR*, 2018. 3, 4, 5