An Improved Convolutional Network Architecture Based on Residual Modeling for Person Re-identification in Edge Computing

SHANCHEN PANG¹, SIBO QIAO¹, TAO SONG¹,² (Senior Member, IEEE), JIANLI ZHAO³, and Pan Zheng⁴ (Senior Member, IEEE)

¹College of Computer and Communication Engineering, China University of Petroleum, Qingdao, Shandong, China
²Department of Artificial Intelligence, Faculty of Computer Science, Politecnico University of Madrid, Campus de Montegancedo, Boadilla del Monte, Madrid, Spain
³College of Computer Science and Engineering, Shandong University of Science and Technology, Qingdao, Shandong, China
⁴Department of Accounting and Information Systems, University of Canterbury, Christchurch 8041, New Zealand

Corresponding authors: Tao Song (e-mail: tsong@upc.edu.cn) and Jianli Zhao (e-mail: jlzhao@sdust.edu.cn)

ABSTRACT Person re-identification is an important task in the field of video surveillance that concentrates on identifying the same person across different cameras. Some methods cannot learn effective image representations, due to the low resolution of pedestrian image data sets. In this article, we propose a novel Siamese network architecture with layers specially designed to address the problem of re-identification. The architecture proposed in this work is applied to the edge of the cloud infrastructure, which can accelerate the speed of pedestrian retrieval. Our network outputs a similarity value when a pair of images is given as input, indicating whether the two input images show the same person. Novel elements of our architecture include a residual model layer that includes an "identity block" and a "conv" block, which considerably capture more efficient features between the two input images. A global average pooling layer is adopted to reduce the model complexity before a fully connected layer, which minimizes person retrieval time in edge computing. Our proposed method significantly improves previous on: CUHK03 by 30% in rank-1, Market-1501 by 35% in rank-1. We also demonstrate that the proposed method outperforms most state-of-the-art methods on the two public benchmarks.

INDEX TERMS person re-identification, deep learning, identity block, conv block, edge computing

I. INTRODUCTION

TRADITIONAL manual methods are becoming unable to retrieve person information quickly from the massive amounts of video or image data being generated, due to the increasingly large number of applications of monitoring in various fields. Edge computing, emerging as an effective way to mitigate the problem of long latencies, has attracted more and more attention [1]. Person re-identification refers to the task of person matching across multiple surveillance cameras; that is, given a target person, that person is to be found in image data sets or videos recorded at different times by surveillance cameras at different positions or different angles. Person re-identification is a key technology in many fields, such as intelligent video analysis, image data processing, and human-computer interaction. Person re-identification technology applied to edge computing can greatly improve the speed of person information retrieval.

However, each surveillance camera is affected by factors such as the light intensity, the imaging angle, and the poses of the captured people; consequently, two images of the same person may appear to represent two different people, or two images of different people may appear to represent the same person, thus increasing the difficulty of person re-identification. In addition, such a data set itself usually consists of different images of a single person from multiple cameras; as a result, the representation learned from one image cannot be well generalized to other images of the same person. Moreover, compared with the images used for other tasks, such as object recognition or detection, the images in a person re-identification gallery are often of lower resolution,
and the image quality is not high. Hence, the difficulty of person re-identification is greatly increased.

In recent research, two main approaches to person re-identification have been applied. The first approach is to break down an image into smaller components. The local features of body parts are extracted based on a designed model, and the features of the whole image are then obtained by utilizing these local features. However, since the image resolution is already very low, the local features extracted in this way may not generalize well to the whole image, which will affect the accuracy of the model. In addition, it is extremely difficult to capture the local features, which uses bounding boxes on human body parts. The second approach is to change the complexity of the model, for example, by increasing the number of layers in the model. From a global perspective, extracting the characteristics of the whole image is the main focus of person re-identification research. However, this can easily exacerbate the phenomenon of gradient disappearance, leading to the extraction of invalid image features.

To address this problem, in this work, we mainly use a convolutional neural network (CNN) to study the problem of re-identification based on a public image data set. The second popular approach introduced above is adopted to build our model. In this paper, we present an improved Siamese network structure for person re-identification based on the reference [2]. Simultaneously, the architecture maintains a low level of complexity that allows it to run quickly on edge devices. The contributions of this paper are as follows:

1. We propose a residual model layers with two residual blocks that can learn more detail high-level features of the two input images. It not only increases the depth of the model, but also reduces the phenomenon of gradient disappearance as much as possible. The two residual blocks, which are an "identity" block and a "conv" block, are proposed based on the concept of residual networks [3]. In order for the features to be comparable across the two images in later layers, residual model layers share weights across the two input views, to ensure that both images use the same filters to compute features.

2. We adopt a global average pooling layer before fully connected layer, thereby reducing the model complexity to some extent. These novel aspects of our architecture endow it with considerable advantages in feature extraction, and experiments prove that our network achieves large improvements over previous methods. Figure 1 shows some examples from our experiments.

The rest of this paper is organized as follows: In Section II, we discuss several related works on person re-identification. In Section III, we introduce the proposed framework of Siamese network modeling. In Section IV, empirical evaluations on two public benchmarks are presented to show the performance of the proposed model. In Section V, we present the conclusion and discussion of this paper.

II. RELATED WORK

A. PREVIOUS WORK ON RE-IDENTIFICATION

Typically, methods for person re-identification is divided into two components: feature extraction from person images and distance metric learning for comparing those features across images. Some researchers focus on finding an improved set of features [4]–[6]. Some researchers focus on finding an improved similarity metric for comparing features [7]–[14]. And some researchers focus on finding a combination of both [15]–[17]. The researcher extracts the color, shape or texture features of the images of interest and then uses distance metric learning to measure the similarity of the extracted features to determine whether two images are of the same person. The basic idea behind the search for better features is to find features that are at least partially invariant to lighting, pose, and viewpoint changes. Features that have been used include variations on color histograms [10], [17] and local binary patterns [10], [17]. The basic idea behind metric learning methods is to find a mapping from feature space to a new space in which feature vectors from same image pairs are closer than feature vectors from different image pairs. Metric learning approaches that have been applied to re-identification include Mahalanobis metric learning [9], KISSME metric learning [10], Discriminative Null Space [11], Linear Similarity Function based on the Mahalanobis distance [15], and Logistic metric learning [13]. Yang et al. change the viewpoint that they consider difference and commonalities between image pairs that different pairs of covariance matrices can be derived from similar pairs, thus making the learning process scalable to a large data set [14]. Our approach is to learn a deep network with residual blocks that simultaneously finds an effective set of features and a corresponding similarity function.

B. PERSON RE-IDENTIFICATION BASED ON DEEP LEARNING

In recent years, significant progress has been achieved in different computer vision areas, including in person re-identification, thanks to the emergence of deep learning, and
in particular deep CNN.

Authors in [2], [18], [19] use Siamese network structure to study the problem of re-identification, combining feature extraction and metric learning. Ahmed et al. propose the neighborhood difference method, which is used to calculate the difference in the feature matrix after two layers of convolution. But, it does not learn an effective set of features of two input images [2]. Varior et al. use long short-term memory (LSTM) instead of the convolutional model; this structure uses contextual information to continuously process image regions to enhance the recognition capabilities for local features [19]. It is effective in learning image similarities in an adaptive manner, but may have efficiency problems under large-scale galleries. In the architecture proposed in [16], an image block matching layer is added to combine the convolution results for two images in different horizontal bands, thereby improving the recognition accuracy.

Also, there has been some attempts to improve person re-identification performance using other network structure. Wang et al. propose a joint learning framework to unify single-image representations (SIRs) and cross-image representations (CIRs) using a CNN [20]. Sun et al. propose SVDNet, which uses singular vector decomposition to optimize the deep learning process, reducing the correlations between weight vectors and producing more identifying features [21]. Zhao et al. propose a new CNN structure called Spindle Net, which considers the structural information of the human body. This is the first study to capture the semantic features of different regions of the body and to use these semantic features for learning [22]. Liu et al. propose an end-to-end comparative attention network (CAN), which learns to selectively focus on certain parts of pairs of person images after receiving only a few glimpses of them [23]. Deng et al. present a "learning via translation" framework. They first translate labeled images from the source domain to the target domain in an unsupervised manner, and then, they train re-identification models on the translated images using supervised methods. Experiments are implemented using the resulting similarity preserving generative adversarial network (SPGAN) to prove its effectiveness [24].

Several loss functions have been adopted for person re-identification. Some like [2], [25], [26] have used positive and negative image pairs through contrastive and binary losses to train their neural network models. Others [27], [28] have employed triplet loss which requires a tuple of anchor, positive and negative images where the training objective is to simultaneously push the positive image towards the anchor while pulling the negative image away from it. Chen et al. propose a quadruplet loss, in analogy to the triplet loss. This method can lead to greater inter-class variations and smaller intra-class variations in the model output [29]. These loss functions are very suitable for person re-identification due to its retrieval nature.

In this work, our network begins with a layer of zero-padding, convolution, batch normalization (BN) [30] and maximum pooling (max-pooling) to preprocess the two input images. To extract person features more quickly and effectively in edge devices, we then alternate the use of novel two residual blocks seven times including "identity" block used five times and "conv" block used two times to learn a set of features for comparing the two input images. A cross-input neighborhood difference layer is used to compare the features from one input image with the features computed in neighboring locations of the other image. This is followed by a subsequent layer that distills these local differences into a smaller patch summary feature. This layer includes a layer of convolution and max-pooling. Next, a novel global average layer is adopted to flatten the feature maps, followed by two fully connected layers with softmax output. Along with our new layers which have learnable parameters in them, our network has two residual blocks as compared to [2], making our network deeper than perviously presented networks for re-identification. In addition, our network introduces a more powerful way to reduce the problem of vanishing gradients and improve the model performance. In extensive experiments, the proposed network significantly outperforms most state-of-the-art methods on two benchmarks: CUHK03 [16] and Market-1501 [31].

III. OUR NETWORK ARCHITECTURE

In this paper, we propose an improved Siamese network architecture to solve the re-identification problem. Given an input pair of images, the architecture is to determine whether or not the two images represent the same person. In brief, our network consists of the following distinct types of layers: preprocessing layers, residual model layers, a neighborhood difference layer, a feature summary layer and a fully connected layer. Figure 2 illustrates our network’s architecture. Each of these layers is explained in the following subsections.

A. PREPROCESSING LAYERS

We need to obtain the relationship between the two input images to determine whether they are of the same person. It has been demonstrated that deep learning can be used to effectively extract high-level image features, thus making image classification tasks more efficient. The brightness, contrast and other properties of an image have a strong effect on its resolution. The same object may exhibit large differences under different brightness and contrast conditions. Therefore, we first need to preprocess each image to reduce the influence of unrelated factors on the performance of the neural network model. In the network proposed in this paper, the preprocessing layers preprocess the input images and extract several low-level features to lay the foundation for the work that follows. The preprocessing layers include one zero-padding layer, one convolutional layer, one BN layer, and one max-pooling layer. In general, we use a BN layer to process the convolution results after each convolutional layer. The goal is to normalize the results to different degrees to increase the efficiency of forward propagation through the network and reduce overfitting. In the preprocessing layers, we first pad the two original input images, which have dimensions of

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/.
160 × 60 × 3, to increase the image dimensions. Each image is padded with 1 × 1 and 3 × 3 kernels on the top, bottom, left, and right sides; thus, we obtain an image map with dimensions of 162 × 66 × 3. Then, we pass the resulting pair of RGB images with dimensions of 162 × 66 × 3 through 64 learned filters with dimensions of 5 × 5 and a stride of 2. The convolutional weights are shared between the two images to ensure that more similar features can be extracted from two images of the same person. We use the BN layer to normalize the feature maps; then, the resulting feature maps are passed through the max-pooling layer, which includes filters with dimensions of 3 × 3 and a stride of 2. The max-pooling layer reduces the number of parameters to some extent. At the end of these operations, each input image is represented by 64 feature maps with dimensions of 39 × 15. The main calculation process of the BN layer is shown below, although the detailed algorithm is not described here:

\[ \mu = \frac{1}{m} \sum_{i=1}^{m} x_i \]  
\[ \sigma^2 = \frac{1}{m} \sum_{i=1}^{m} (x_i - \mu)^2 \]  
\[ \hat{x}_i = \frac{x_i - \mu}{\sqrt{\sigma^2 + \epsilon}} \]  
\[ y_i = \gamma \hat{x}_i + \beta \]

where \( x_i \) is the input value, \( \mu \) is the minibatch mean, \( \sigma^2 \) is the minibatch variance, \( \gamma \) is the scale and \( \beta \) is the shift. We use the BN layer to normalize all values in the model in order to accelerate the training convergence process.

### B. Residual Model Layers

According to a large number of experiments, the deeper a deep neural network is, the better the model performance. However, a greater depth will also make the model more prone to gradient degradation, which can prevent any effective image features from being extracted. To ensure the extraction of high-level image features while mitigating the phenomenon of gradient degradation, we designed the residual model layers that are presented in this section. After the preprocessing layers come the residual module layers, through which we use the concept of residual networks to deepen the network model and extract more effective features. In this way, we can eliminate the problem of vanishing gradients and improve the model performance. In this paper, we propose two different modules based on the concept of residual modeling: an "identity block" module and a "conv block" module. These two different residual modules considerably capture more efficient features between the two input images. Simultaneously, they can mitigate the phenomenon of gradient degradation. The term "identity block" refers to the fact that the dimensionality of the value of the input layer is the same as the dimensionality of the intermediate value of the output layer after N convolutions. This is described as follows:

\[ \text{dims}(a^i) = \text{dims}(z^{i+N}) \]  

where \( a \) is the initial activation value, \( z \) is the intermediate value after N convolutions, and \( i \) represents the \( i \)-th layer.

In the "identity block" proposed in this paper, the initial activation value "a" is passed through a "short path" and is added to "z" after the third convolution. Then, we pass the results through the activation function. This process is described as follows:

\[ a^{i+3} = R(z^{i+3} + a^i) \]

where \( R(x) \) is the activation function, which is the rectified linear unit (ReLU) function. The ReLU function is described as follows:

\[ R(x) = \max(0, x) \]

We use the "identity block" module several times in our network. This module contains three convolutional layers, in which weights are shared across the two feature maps (or two views) to ensure that they use the same weights for feature extraction. We repeat such "identity blocks" five times in the residual model layers. The parameters of the first two instances, such as the size of the kernel and the number of filters, are the same. The parameters of the last three instances are also the same. The purpose of using the module multiple times while varying the block parameters is to deepen the network architecture, thus allowing us to extract more prominent feature maps, reduce overfitting, and enhance the robustness of the model.

In the "identity block" module, there are a total of three convolutional layers. In the first two calls to the "identity block" module, the first convolution in the module is accomplished using 16 filters with dimensions of 1 × 1 and a stride of 1. The second convolution uses 16 filters with dimensions of 3 × 3 and a stride of 1. Simultaneously, a padding process is applied to maintain the dimensional invariance of the feature maps. The third convolution is accomplished using 32 filters with dimensions of 3 × 3 and a stride of 1. BN processing is performed after each convolution to increase the speed and accuracy of model training. Finally, the intermediate state value after the three convolutions is added to the initially saved value, and we pass the result through the activation function.

In the latter three calls to the "identity block" module, the values of the module parameters are adjusted. The first convolution in the module uses 32 kernels with dimensions of 1 × 1 and a stride of 1. The second convolution uses 32 filters with dimensions of 3 × 3 and a stride of 1. Simultaneously, a padding process is applied to maintain the dimensional invariance of the feature maps. The third convolution is accomplished using 64 filters with dimensions of 1 × 1 and a stride of 1. Figure 3 illustrates the "identity block" architecture.

The term "conv block" refers to the fact that the dimensionality of the initial activation value "a" is different from...
the dimensionality of the intermediate state value “z” of the output layer after N convolutions. Therefore, we need to perform a convolution on the initial activation value such that the dimensions of both are the same. This is described as follows:

\[ \text{dims}(a^i) \neq \text{dims}(z^{i+N}) \] (8)

In the proposed "conv block" structure, the initial layer’s activation value \(a^i\) is similarly passed through a "short path" to be added to the output layer’s intermediate value \(z^{i+3}\) after three convolutions, but due to the difference in the dimensions, a convolution operation must be additionally applied to \(a^i\). Thus, we can obtain \(a_{\text{short\_cut}}^i\), which has the same dimensions as \(z^{i+3}\). Finally, we sum \(a_{\text{short\_cut}}^i\) and \(z^{i+3}\) and pass the result to the activation function to obtain \(a^{i+3}\). This can be described as follows:

\[ a_{\text{short\_cut}}^i = C(a^i) \] (9)

\[ a^{i+3} = R(z^{i+3} + a_{\text{short\_cut}}^i) \] (10)

where \(C(x)\) represents one convolution.

In the network proposed in this paper, we call the "conv block" module twice. This module contains four convolutions, in which the weights are shared across both views or feature maps.

When the "conv block" module is called for the first time, the first convolution in the module is accomplished using 16 filters with dimensions of \(1 \times 1\) and a stride of 1. The second convolution uses 16 filters with dimensions of \(3 \times 3\) and a stride of 1. Simultaneously, a padding process is applied to maintain the dimensional invariance of the feature maps. These features are then passed through the third convolution, which uses 32 filters with dimensions of \(1 \times 1\) and a stride of 1. The purpose of the first three convolution operations is to extract image features. Since the dimensionality of the output feature maps is not the same as that of the initial input maps, the fourth convolution is required to adjust the dimensions of the initial input. Then, we add the adjusted result to the result
of the three convolutions. After that, we pass the resulting value through the activation function.

When the "conv block" module is called for the second time, the values of the module parameters are adjusted. The first convolution in the module is accomplished using 32 filters with dimensions of $1 \times 1$ and a stride of 2. The second convolution uses 32 filters with dimensions of $3 \times 3$ and a stride of 1. These features are then passed through the third convolution, which uses 64 filters with dimensions of $1 \times 1$ and a stride of 2. Figure 4 illustrates the "conv block" architecture.

---

**D. FEATURE SUMMARY LAYER**

Through the previous layers, we have obtained a complicated representation of the difference between the two images. Then, a feature summary layer is used to summarize the complex relations calculated by the neighborhood difference layer and to learn the spatial relationships between the neighborhood difference maps. The feature summary layer performs the following mapping: $K \in \mathbb{R}^{37 \times 3 \times 13 \times 3 \times 64} \rightarrow L \in \mathbb{R}^{37 \times 13 \times 64}$. This layer convolves the previously obtained complex features using 64 filters with dimensions of $3 \times 3$ and a stride of 3. Now, we wish to learn the spatial relationships across the neighborhood difference maps; therefore, the weights are not shared in the feature summary layer. After that, we pass the results through the ReLU function.

To extract the main features and reduce the number of model parameters, we use max-pooling to reduce the height and width by a factor of 2. Thus, we obtain two feature mappings:

$$L \in \mathbb{R}^{37 \times 13 \times 64} \rightarrow G \in \mathbb{R}^{27 \times 9 \times 64} \text{ and } L' \in \mathbb{R}^{37 \times 13 \times 64} \rightarrow G' \in \mathbb{R}^{27 \times 9 \times 64}.$$  

**E. FULLY CONNECTED LAYER**

Finally, we use a fully connected layer to combine the feature representations and map the learned distributed feature representation to the sample label space. Before applying the fully connected layer, to obtain high-order relationships, we combine the feature maps $G$ and $G'$ to obtain a summary feature map with dimensions of $27 \times 9 \times 128$. We do not use the traditional flattening method to abruptly simplify the 128 feature maps into a single feature vector. Instead, we apply a global average pooling layer to each feature map. Thus, each feature map is reduced to one value in the feature vector. This global average pooling operation does not destroy the spatial structure information of the original feature matrix. Then, the resulting 64-dimensional feature vector is passed through the ReLU function. Finally, we pass the feature vector to a fully connected layer containing 2 softmax units to calculate the probabilities that the two images represent either the same person or different people. The softmax function is described as follows:

$$s(x) = \frac{\exp(x_i)}{\sum_{i=1}^{n} \exp(x_i)} \quad (11)$$

---

**IV. EXPERIMENTS**

**A. DATA SETS**

To evaluate our proposed method, experiments are conducted on two benchmarks, CUHK03 [16] and Market-1501 [31]. The CUHK03 data set [16] contains 1467 identities captured by 6 different surveillance cameras, comprising a total of 13164 images. Each person in CUHK03 [16] is captured from two different views. On average, there are 4.48 images of each person from each view. We use 1267 of the identities...
as the training set and the remaining 200 identities for validation and testing. Market-1501 [31] is composed of 12936 training images and 19732 gallery images, representing 1501 identities. Each person is captured by at most 6 cameras, comprising 5 high-resolution cameras and one low-resolution camera. We use 1000 of the identities as the training set and the remaining 501 identities for testing and validation. We present a more intuitive visualization of the data sets in Figure 5 and Figure 6. From these figures, it is evident that the data sets contain multiple images of each person and that each image records a person in a walking pose. The poses of the people in the data sets are made more complex by the fact that the images contain not only people but also other objects, such as bicycles and backpacks. Similar to the problem of object detection or recognition, the person re-identification problem is also affected by occlusion. However, person re-identification is a more challenging task. This is because the images in person re-identification data sets have lower resolution, making it more difficult to identify whether images show the same person or different people.

It is well known that deep learning requires large numbers of samples for training. Notably, there are not nearly as many positive pairs as negative pairs; this situation can lead to data imbalance and overfitting. Therefore, to reduce overfitting, we apply data augmentation techniques to our data sets. We increase the number of images by flipping, rotating,
deforming, scaling, and performing other transformations on each original image; thus, approximately 5 images are added for each original image. We use 30,000 pairs of images as the training set; 15,000 pairs of images of the same person from two different views are randomly selected as positive samples, and 15,000 pairs of images of different people are randomly selected as negative samples. Similarly, we use 10,000 pairs of images as the validation set; 5000 pairs of images of the same person from two different views are randomly selected as positive samples, and 5000 pairs of images of different people are randomly selected as negative samples. For the test set, we apply the same process as for the validation set, i.e., we use 10,000 pairs of images, comprising 5000 randomly selected pairs of images of the same person from two different views as positive samples and 5000 randomly selected pairs of images of different people as negative samples.

B. TRAINING THE NETWORK

For these experiments, we use the TensorFlow and Keras deep learning framework to implement our architecture. Model training converges within approximately 7 hours on NVIDIA GTX1080Ti GPUs. For a pair of images representing the same person, we attach a positive tag (1); otherwise, we attach a negative tag (0). The whole data set is divided into many minibatches to train the model. This approach can improve the training speed and reduce overfitting. To compute the loss function of a minibatch, 32 images are randomly sampled from the data sets such that the ratio of the sampled positive and negative images is approximately 1:1. The cross entropy is adopted as the loss function to avoid trapping in local minima during optimization. We choose the Adam optimization algorithm instead of the traditional stochastic gradient descent algorithm to optimize the model because of the fuzzy characteristics of the data sets. The
FIGURE 7: CMC curves obtained on the CUHK03 data set. Qdlet: Quadruplet, Qdlet+Marg: Quadruplet+Marg, AlexNet: CAN (AlexNet), VGG-16: CAN (VGG-16).

Adam optimizer has several important parameters. For two of these parameters, beta_1 and beta_2, the preferred values are 0.9 and 0.999, respectively, according to the literature. By contrast, accurate convergence will not be reached during training if a fixed learning rate is used. Hence, it is necessary to constantly vary the value of the learning rate. We adopt an initial learning rate of α = 0.01, and we update the learning rate as follows:

$$\alpha = \alpha \times \frac{1}{1 + \text{decay} \times \text{num\_itera}}$$  \hspace{1cm} (12)$$

where α is the learning rate, decay is the loss value (0.0005 in this paper), and num\_itera is the number of iterations.

The loss function is calculated as follows:

$$\text{Loss}(\theta) = -\frac{1}{m} \left[ \sum_{i=1}^{m} y_i^t \log h_\theta(x^i) + (1 - y_i^t) \log(1 - h_\theta(x^i)) \right]$$  \hspace{1cm} (13)$$

where m is the number of samples, $y_i^t$ is the i-th true label, and $h_\theta(x^i)$ is the i-th predicted label.

For performance evaluation on both data sets, cumulative matching characteristic (CMC) curves are employed to measure the network performance. A CMC curve represents the probabilities that a queried identity will appear in candidate lists of different sizes. No matter how many ground-truth matches there are in the gallery, only the first ground-truth match is used in the CMC calculations, and we report the single-shot results on all data sets.

C. COMPARISON WITH THE STATE OF THE ART

The results obtained in extensive experiments on these two public data sets are shown in Table 1 and Table 2. In these tables, we compare our method against eSDC [4], KISSME [10], FPNN [16], KLFDA [17], SIRCIR [20], JoinRe-id [2], GOG [32], Quadruplet [22], Quadruplet+Marg [29], CAN (AlexNet) [23], CAN (VGG-16) [23], DGD [33], SVDNet [21] and Spindle [22]. As shown in Table 1 and Table 2, our results are superior to those of most of the methods listed above, including most deep networks. Figure 7 and Figure 8 plot the CMC curves for the rank-1 accuracy for all of these methods on the CUHK03 and Market-1501 data sets. Our method outperforms JoinRe-id, a previous deep learning method, by a large margin. Our rank-1 accuracy is higher than that of the previous state-of-the-art method (84.60% vs. 54.74% on CUHK03 and 80.20% vs. 45.00% on Market-1501), thus verifying the effectiveness of our network architecture. By contrast, Spindle achieves better performance than our method on CUHK03 because Spindle considers body landmark information. Even so, our method still outperforms Spindle on Market-1501 (80.20% vs. 76.90%). Our method uses two different types of residual blocks for feature extraction, and we leverage global representations of person images. This is why our method outperforms Spindle on the Market-1501 data set. Similarly, the rank-1 accuracy of SVDNet is higher than that of our method on the Market-1501 data set but lower than that of our method on the CUHK03 data set. SVDNet uses CaffeNet and ResNet as its backbone; we will investigate integrating this strategy into our method in future research.

V. CONCLUSION AND DISCUSSION

In this paper, we present a novel Siamese network architecture for person re-identification based on the concept of residual modeling. And two different residual network modules are proposed to learn an effective set of features. The two residual blocks are adopted to our architecture, significantly improving the performance on CUHK03 and Market-1501. The global average pooling is used before fully connected layer to flatten the convolutional feature maps, preserving the spatial structure and content information of the feature maps. Extensive experiments show that our network architecture outperforms most state-of-the-art methods (especially JoinRe-id) on two benchmark data sets.

However, our architecture cannot reduce total energy consumption and keep fast person retrieval well in edge devices. In the future, we will use the method of weight pruning to lightweight the model, just like [34], to give the model fast processing capability in edge devices. In addition, the work presented in [35]–[39] can provide a new energy-saving protocol that can be adopted when our model is implemented in practice. And we can incorporate different methods or models to consider the human body structure, thereby improving...
the global feature representation by leveraging local features extracted from human body parts. May be we can learn some experiences from [40]–[46]. In this way, our model may become more robust and achieve higher performance.

ACKNOWLEDGMENTS
This work was supported by the National Natural Science Foundation of China (61572522, 61572523, and 61672033), Key Research and Development Program of Shandong Province (No. 2017GGX10147).

REFERENCES


***