Research on pedestrian image retrieval based on deep convolution neural network

Jing Fan[*, †]， Jinshuai Qu[†] and Mingmu Chen[†]
[*] Faculty of Metallurgical and Energy Engineering, Kunming University of Science and Technology, Kunming, China
[†] University Key Laboratory of Wireless Sensor Networks in Yunnan Province, Yunnan MinZu University, Kunming, China
E-mail: fanjing9476@163.com Tel: +86-13888040700
E-mail: qujinshuai1989@163.com Tel: +86-13987195292
E-mail: 397246753@qq.com Tel: +86-15287933189

***Abstract*—With the rapid development of information technology, mobile devices are becoming more and more popular. All kinds of multimedia information, such as text information, pictures and videos, are produced. The image information and video information are the most, how to use good image and video information is the problem we face, and the pedestrian retrieval in monitoring video has great help to help the police to break the case, so the retrieval of pictures and video is an urgent problem to be solved. In this paper, the three tuple network based on measurement learning is insufficient to learn the weight of the network, that is to say, we can not give full play to the advantage of training Batch. The relationship of three tuples has been determined before training. Therefore, this paper improves the three tuple loss function, uses the Caffe open source framework to train the deep convolution neural network model, and carries out the experimental evaluation under the two datum data sets, which are Market-1501 and CUHK03 respectively. In Market-1501, the result of the CaffeNet model experiment is mAP=42.18%, and the result of the CUHK03 experiment is mAP=33.46%. The experimental results show that our method is superior to the three tuple loss function.**

I. Introduction

Pedestrians re recognition refers to a series of pedestrians recognition for a series of cameras without overlapping areas. As shown in Figure 1, the video camera A and B's horizons are not overlapped, that is, the images taken by A and B are not overlapped, such as the pedestrian A captured by our camera A, we need to relocate the camera in the B. To catch pedestrians A, may also be in other distributed cameras, pedestrians re recognition do this one thing, this application for the security of the public area and the police to break the criminal investigation of criminal investigation is of great significance. The researcher widely thinks that the problem of pedestrian re recognition is a sub problem of image retrieval, so this article will think that pedestrian re recognition and pedestrian image retrieval refer to the same concept, which will be distinguished from the other.

Due to the changes in the visual angle of the camera, the lower resolution of the image, the change of the pedestrian's posture, the difference of light from the different camera environment, and the occlusion caused by many pedestrians and other things, these problems will challenge the recognition of pedestrians. For example, as shown in Figure 2 (1) (3), because the camera deployment environment is different, the camera parameters are different, so the images are photographed with different light and change angles, and the images of the same pedestrians deployed in different locations are very different. And different pedestrians may look more similar in appearance. Figure 2 (2) shows that the image captured by the camera is low resolution and blurred. The environment of Figure 2 (4) camera deployment may be complex in the road, the airport, the mall, or the station, which will cause our goals to be lost or blocked.

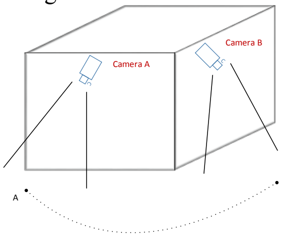


Fig. 1 recognition of pedestrians under multiple cameras

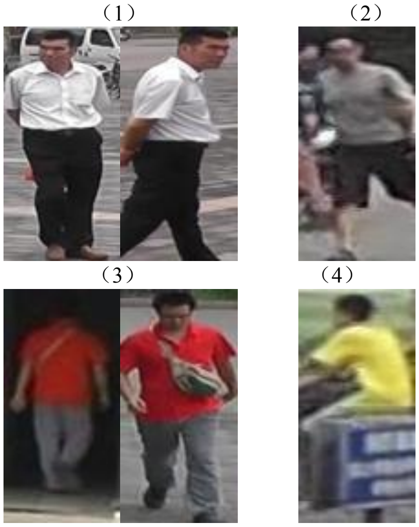


Fig. 2 the challenge of pedestrians recognition

For the image retrieval problem, the traditional underlying visual features are used to express the image, which will have a "semantic gap" with whether we judge whether the image is similar. "Semantic gap" refers to our human judgment that the image is not used to determine the color, edge, and texture of the target object. We human beings believe that the semantic similarity is similar. In order to overcome this "semantic gap", in recent years, the application of deep learning method has achieved good performance automatically from the advanced semantic features of image learning. This article will also combine in-depth learning methods to carry out research.

Feature representation and distance measure are two main components of pedestrian re recognition system. Now the main way to identify pedestrians is to use manual features such as color and texture histogram [1-3]. In order to increase their ability of feature representation, features are designed to capture spatial information [1,4]. For example, Farezena et al. Used the symmetry of pedestrian images to propose a method called Symmetry Driven Accumulation of Local Features (SDALF), which is called symmetric drive (SDALF), which is a good robust [4] for the chaotic background. The graphic structure characteristics based on body structure have also been well studied, aiming to solve individual differences in [5,6].

In addition to manual feature design, there are still some learning characteristics for pedestrians re recognition tasks. For example, Gray and Tao et al [1] have proposed the use of Adaboost algorithm to learn the effective representation of features from the integration of local features. Zhao and others proposed to learn the middle layer features from the hierarchical block of image blocks.

Another important research direction of pedestrian re recognition is distance metric learning. Zheng et al. [7] specifies the distance metric learning as the relative distance comparison (Probabilistic Relative Distance Comparison, PRDC) model, which maximizes this likelihood, and the correct match will have a very small distance between them, and the incorrect matching pair will have a large distance and an incorrect match. The match pair distance is greater than the correct distance. In addition, Mignon and Jurie[8] proposed the Pairwise Constrained Component Analysis (PCCA), which maps the original data to a low dimensional space, where the distance between them is the required attribute. Li et al. [9] introduced a locally adaptive threshold (Locally Adaptive Thresholding Rule, LATR) into the distance metric model, and the literature reports that their methods have achieved good performance in the pedestrian re recognition task. The sort SVM has been proposed to learn a subspace, in which a given image is given, so there is a higher order for the matched image, and the mismatched image has a lower order. There are some general distance metric learning methods, which are rarely used in the background of pedestrians re recognition.

Inspired by the successful application of deep learning in other computer vision, some neural networks have been applied to solve the problem of pedestrian recognition. Yi et al. [10] used deep neural network to learn the similarity of pairs of images, and achieved the best performance at that time. Liu et al. [11] provides a tag set model, which uses the depth belief network model and the nearest neighbor principal component analysis to improve the performance of the pedestrian re recognition by retrieving the connection features of the image set and the retrieved image set. Xu et al. [6] used the clustered sampling algorithm [12] to reidentify pedestrians with templates. Li et al. Proposed a deep learning framework for learning filter pairs, which tried to automatically encode photometric conversion by means of cameras. Our work in this paper differs from the loss function in the main aspect of these works. Because this model mainly fine-tuning the pre-existing training classification model, we first introduce several deep convolutional neural network models.

The rise time of deep learning was in 2012, and Hinton including its student Krizhrvsky and others took advantage of the convolution neural network to participate in the ILSVRC (ImageNet Large Scale Visual Recognition Competition) large-scale image recognition competition, and more than second use traditional handmade SIFT features to 8.2 percentage points. The gap is very large. Since then, the deep learning craze has finally arrived. Convolution neural network has achieved good results in various image tasks. For example, the Faster-RCNN proposed by Girshick[13] and others has achieved an average accuracy of 78.8% in pedestrian detection tasks (mAP); Sun et al. [14] proposed a series of models on face recognition tasks. The best model, DeepID, has achieved good results and the accuracy rate can be higher than 99%.

In recent years, the research of convolution neural network has been deepened, and various kinds of network structures have been tailored to different specific tasks. The most typical network structures include LeNet, AlexNet, VGGNet, GoogLeNet and ResNet. The embryonic form of the convolution neural network is LeNet, and its main task is handwritten font recognition, so it is then integrated into the bank's check identification system and is well applied, and the network model structure is listed. The model structure of LeNet is complete, and the basic components of the convolution neural network are all, but the ability of its expression is weak, so it can not solve more complex image tasks.

AlexNet is a convolution neural network model adopted by Hinton and its students Krizhevsky and others in the 2012 ILSVRC image recognition competition. AlexNet consists of 5 convolution layers. Each convolution follows the pool layer. Finally, there are 3 full link layers. The detailed network structure is shown in Figure 3. The activation function used in the network model structure is nonlinear correction unit (Rectified Linear Unit Function, ReLU), and it also introduces Dropout, which can prevent the overfitting of the network in training. As mentioned in the second chapter, Dropout is a dormant state of some network neurons in the network learning process by a probability of 50%, that is to say, the output is 0. In order to improve the generalization ability of the network in other test data, the AlexNet network model also proposes the local response normalization (Local Response Normlization, LRN) operation. In addition to these, the network model structure also performs some operations on the enhancement of image data, in which the image is cut by probability, the image is translated, and the image is used as the horizontal mirror and so on.

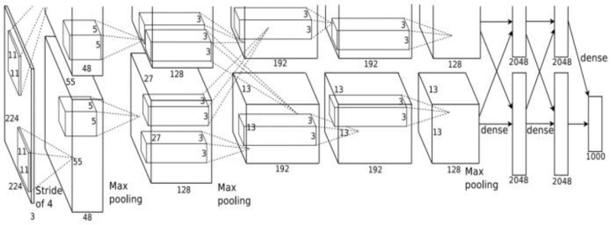


Fig. 3 AlexNet network model structure diagram

GoogleNet[15] got the first network model structure in the image classification task of ImageNet image recognition challenge in 2014. In its internal structure, it adopts a model structure called "Inception" module, which is named Network In Network (NIN), that is to say, one node expansion in the model structure is also a small model structure. The schematic diagram of the network structure of the Inception model is shown in Figure 4. It can be seen from the diagram that the convolution operation of the convolution kernel with different sizes is used for the same input, and the final output is to connect the features of the previous convolution operation to the output. The use of the Inception module extends the depth and horizontally of the model structure to widen the width of the model structure, so that the network can have good sparsity.

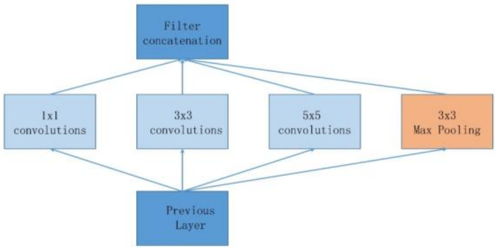


Fig. 4 Inception module structure diagram

The previous discovery from AlexNet to VGGNet is only a simple increase in network depth, which can make a good effect, but when the network has a certain depth, it can not increase the performance of the network so simply as this, and sometimes it will damage the network performance. In order to solve this situation, He, MSRA of Microsoft Research Institute of Asia, creatively developed ResNet in 2015. That is the Deep Residual Network (ResNet) [16]. Compared to the previous model structure of the convolution neural network, the depth residual network ResNet adds a short circuit connection (Shortcut connection) in the non adjacent layer, which is shown in Figure 5. For this link, different functions can be used as activation functions, but Identity Mapping can also be used easily. With this network structure He and others, it is very easy to increase the model network structure by 152 layers, and the network structure of the 152 layer will not appear gradient disappearance and so on. The network can converge well and get very good performance.

It can be observed from the progress of the convolution neural network that we can use the method of increasing the number of network layers and designing more complex model structures to make the network have more expressive features and make the network have good generalization ability.

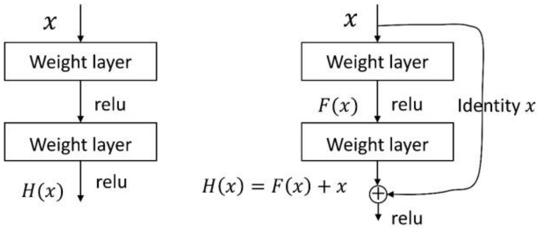


Fig. 5 Left is the ordinary chain CNN structure right side is CNN with residuals module

Aiming at the specific problems of pedestrian re identification, we use AlexNet and 16 level VGGNet respectively as our network model structure.

II. METRIC LEARNING

Now deep learning is done by combining metric learning with end-to-end learning, namely, directly learning the similarity of samples. That is to say, the learning of sample similarity measure, also known as metric learning, usually refers to the characteristics of image data. The ultimate goal of similarity measurement is to use learning and training to reduce the distance of sample features belonging to the same category, and then extend the characteristic distance of samples of different categories to expand [17].

We represent the dimension vectors of the two samples so that vectors can be simply represented as $x_1 = \{x_1^{(1)}, x_1^{(2)}, ..., x_1^{(n)}\}$ and $x_2 = \{x_2^{(1)}, x_2^{(2)}, ..., x_2^{(n)}\}$, Then we represent the distance between the two eigenvectors. The common distances are Euclidean distance, Manhattan distance, martensitic distance, Minkowski distance, cosine distance and so on.

The learning method of depth learning combined with similarity measure is generally learning directly from the training data pairs. The training data input for this method is usually a pair or the three tuple, so we will need to improve the network model of the convolution neural network. Under this input requirement, there are two network structures, Siamese Network and Triplet Network.

A. Siamese Network

The learning object of twin network structure is data pair, so it is usually used to determine whether the two pictures belong to the same class, so it is a verification problem. Twin network structure was used for face verification in 2005 and achieved good results. Since then, the twin network structure has been used in many directions of image processing.

The schematic diagram of the twin network model is shown in Figure 6. From the above diagram, we can see that the weight of the two sub networks in the model structure is shared, that is, the two input training data are actually passed through the same network. The main function of the sub network of the model is to map the data from high dimension to low dimension, and finally to compare the similarity of two low dimensional data in low dimensional space. Set and represent the model training data pairs to indicate whether the sample is similar or not, and when the same category is set to

0, the different classes are set to 1. Set and represent the data after the model is mapped to the low dimension. Finally, a mathematical expression is used to represent similarity.

$$E_W(x_1,x_2) = \| H_W(x_1) - H_W(x_2) \| \qquad (1)$$

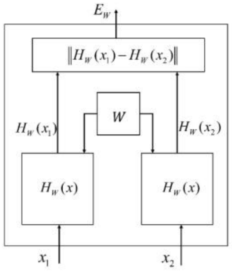


Fig.6 Siamese Network

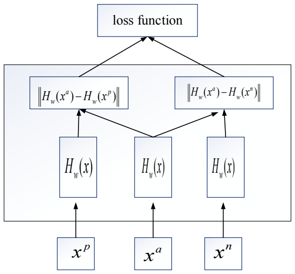


Fig.7 Triplet Network

When the twinning network model is trained, the network needs to satisfy certain relations to explain that the network model is useful. That is to say, m belongs to certain values, generally equal to 1, and for any x_1 and x_2. The relationship between the samples in the low dimensional space is to meet the $E_W(x_1,x_2)+m<E_W(x_1,x_2^{'})$. x_1 and x_2 are similar training data pairs. x_1 and $x_2^{'}$ are different classes of training data pairs.

So we can define the loss function of the model structure as follows:

$$J(W,(y,x_1,x_2)^i) = (1-y)J_H(E(x_1,x_2)^i) + yJ_I(E(x_1,x_2)^i) \qquad (2)$$

The $(y,x_1,x_2)^i$ in the formula represents the i training data pair and matching tag. J_H represents the loss function of matching data pairs, and J_I represents the loss function of mismatched data pairs. The definition of J_H and J_I is to meet the smaller and smaller process of training and learning so as to ensure the training effect. So we often use the formula to express the loss function concretely.

$$J(W,(y,x_1,x_2)^i) = (1-y)\frac{1}{2}(E_W)^2 + y\frac{1}{2}\{\max(0,m-E_W)\}^2 \qquad (3)$$

In the formula, m is the minimum difference between the distance of negative sample pairs and the distance of positive sample pairs.

It is very simple to embed the convolution neural network network into the similarity measure learning. In Figure 6, the twin network schematic of the network can be replaced by the subnetwork of the network model structure to CNN. So far, some literature has proposed the twin network structure with CNN, and applies it to image processing tasks, such as face verification, human posture estimation and so on.

B. Network model structure of three tuples

In similarity learning, there is another kind of network structure called three tuple, also called Triplet network, in addition to the twin network structure introduced above. The learning feature of Triplet network structure is the ability to learn fine grained data matching, that is, after learning, the model can be specific to a person's matching. Google [18] uses three tuple network model structures to study the FaceNet for face recognition, and the accuracy of the benchmark set is ninety-nine point six percent.

Compared with the twin network model, the three tuple network model consists of three subnetworks in total, so it is able to enter three tuples and a schematic diagram of the network model of the three tuple, as shown in Figure 7. For the three input data shown above, we will call it the three tuple. The input data x^a in the figure above is called the Anchor sample. x^p is called a positive sample (Positive sample). Positive samples and anchor samples belong to one category. x^n is called negative sample (Negative sample), and it belongs to different classes with anchor samples. We map the three input data of the three tuple network model to low dimensional space after the network, and then calculate the distance between the positive sample and the anchor sample in the low dimensional space in the low dimensional space and the distance between the negative sample and the low dimensional space of the anchor sample. Finally, the loss is calculated.

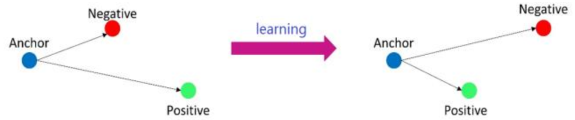


Fig. 8 A schematic diagram of three tuple model learning effect

The main purpose of the design of the three tuple network model structure is to train the network, and then calculate the sample in the low dimensional space. We hope that the distance between the anchor sample and the positive sample is less than the anchor point sample and the negative sample, as shown in Figure 8. In the whole training set, we represent the i three tuple as (x_i^a, x_i^p, x_i^n), which is the same as the above. After the action of the network, the features are expressed as $f(x_i^a)$, $f(x_i^p)$ and $f(x_i^n)$ respectively. Then, according to the above introduction, we hope that the feature representation of the three tuple satisfies the following relations:

$$\| f(x_i^a) - f(x_i^p) \|_2^2 + \alpha < \| f(x_i^a) - f(x_i^n) \|_2^2 \qquad (4)$$

$$\forall \left(f(x_i^a), f(x_i^p), f(x_i^n) \right) \in D \qquad (5)$$

The D in the upper form indicates that the training of all the three tuples is composed of all the data, and the α indicates that the distance between the anchor sample feature and the negative sample feature subtracts the minimum value of the difference between the sample features of the anchors and the positive sample features.The mathematical expression of the triplet loss loss function commonly used in the three tuple network model is as follows:

$$J(x_i^a, x_i^p, x_i^n) = \frac{1}{2}\sum_{i=1}^{N} \max\{\| f(x_i^a) - f(x_i^p) \|_2^2 + \alpha - \| f(x_i^a) - f(x_i^n) \|_2^2, 0\} \qquad (6)$$

After losing, we need to calculate partial derivatives when using gradient descent method to update weights. The formula is as follows:

$$\frac{\partial J(x_i^a, x_i^p, x_i^n)}{\partial f(x_i^a)} = 2\left(f(x_i^n) - f(x_i^p) \right) \qquad (7)$$

$$\frac{\partial J(x_i^a, x_i^p, x_i^n)}{\partial f(x_i^p)} = 2\left(f(x_i^p) - f(x_i^a)\right) \tag{8}$$

$$\frac{\partial J(x_i^a, x_i^p, x_i^n)}{\partial f(x_i^n)} = 2\left(f(x_i^a) - f(x_i^n)\right) \tag{9}$$

In the three tuple network model structure, how to build a three tuple of three samples plays a vital role in the training effect of the network and the speed of convergence. If the training data is randomly composed of three tuples, then assuming that the data set has n samples, then all the samples have n^3 possibilities. For deep learning, the number of training data is not intended to be very large, and the more lethal is that a large part of the combined three tuples are in the back. It is useless to propagate update parameters, so it is very important to build useful three tuples. Hard positive and hard negative are widely used to select effective samples. Hard positive literally means the most difficult positive sample, meaning that the most difficult sample is hard negative, which is the most difficult negative sample from the anchor point sample, and the most recent negative sample from the anchor point sample. The choice of hard positive and hard negative can be combined offline and online. It is selected from a Batch during training.

C. *Improved three tuple loss function*

In the classic implementation of the three tuple method mentioned in the previous section, a Batch with B is selected for three tuples, then they are made up of $3B$ images, in which the mapping of the image is calculated, and the contribution to the loss is only B . But we can see that the $3B$ image can be composed of $(6B^2 - 4B)$ pairs of three tuples, and it would be very wasteful to use only B item loss. So the classic three - tuple method doesn't make full use of Batch when training, because the triplet training data is random sampling, so it's equivalent to a fixed three tuple during training. To solve this problem, we improved the three tuple loss function. The mathematical expressions are as follows:

$$J = \frac{1}{2|P|}\sum_{i\in P}\max\left(0, J_i\right)^2 \tag{10}$$

$$J_i = \max_{\substack{p\in P\\ p\neq i}}(D_{i,p}) + \max_{n\in N}(\alpha - D_{i,n}) \tag{11}$$

Among them, P is the set of positive samples of training set, and N is the set of negative samples pairs of training set. The loss function above is faced with the problem that this function is very smooth. To solve this problem, we will optimize an upper bound of the entire function.

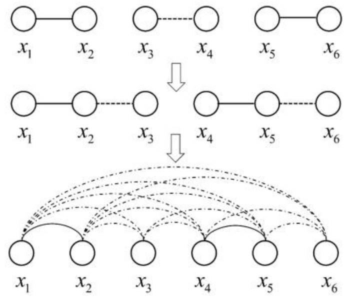


Fig. 9 Batch utilization rate change process

In order to make full use of training Batch, a key idea is to enhance the utilization of Batch so as to achieve the use of all data pairs. Figure 9 shows this full use process, from the use of the twin network image pairs to the three tuples of the three tuple network to use all the images of the three tuple network, in which the real line is a positive sample pair and the dotted line is a negative sample pair. This means calculating the distance between all images of each batch. We can do this fast and fast, assuming that there is a Batch, $X \in R^{m\times c}$, a two norm of a single Batch element that contains the c dimension of a m sample, the ordinary column vector of a single Batch element of a single Batch element is $\tilde{x} = [\| f(x_1)\|_2^2, \ldots, \| f(x_m)\|_2^2]^T$, and the square of all the elements of the Batch can be effectively calculated $D^2 = \tilde{x}1^T + 1\tilde{x}^T - 2XX^T$, of which $D_{ij}^2 = \| f(x_i) - f(x_j)\|_2^2$. However, what we need to pay attention to is that both the positive sample pair and the negative sample pair are sampled randomly in Figure 8, so the sample is effective for the information carried, that is to say, these samples are not the more difficult positive samples and negative samples.

For the above problem, we randomly sampled C individuals, and then each person randomly sampled H sheets and then formed $P = CH$ picture in a Batch. So we can search for the most difficult positive and negative samples in a Batch (which can be seen as a mild and difficult case, because it is not the most difficult in the whole world), which is a difference between our three tuples, and the three tuple is fixed after sampling and cannot be searched in the Batch.

As mentioned earlier, the loss function defined by us is not smooth, so we optimize an upper bound function. Its mathematical expression is as follows:

$$\tilde{J} = \frac{1}{2|P|}\sum_{i\in P}\max\left(0, \tilde{J}_i\right)^2 \tag{12}$$

$$\tilde{J}_i = \log\left(\sum_{\substack{p\in P\\ p\neq i}}\exp\left\{D_{i,p}\right\}\right) + \log\left(\sum_{n\in N}\exp\left\{\alpha - D_{i,n}\right\}\right) \tag{13}$$

Among them, the P and N definitions are the same as in the previous article. The backpropagation update algorithm is shown in algorithm 1.1, where the partial derivatives are as follows:

$$\frac{\partial \tilde{J}}{\partial D_{i,p}} = \frac{1}{|P|}\tilde{J}_i \, 1[\tilde{J}_i > 0]\frac{\exp\{D_{i,p}\}}{\sum_{\substack{p\in P\\ p\neq i}}\exp\{D_{i,p}\}} \tag{14}$$

$$\frac{\partial \tilde{J}}{\partial D_{i,n}} = \frac{1}{|P|} \tilde{J}_i \, 1[\tilde{J}_i > 0] \frac{-\exp\{\alpha - D_{i,n}\}}{\sum_{n \in N} \exp\{\alpha - D_{i,n}\}} \tag{15}$$

```
Input: $D$, $\alpha$
Output: $\partial \tilde{J} / \partial f(x_i), \forall i \in [1, m]$
Initialization: $\partial \tilde{J} / \partial f(x_i) = 0, \forall i \in [1, m]$
for $i = 1, \ldots, m$ do
   for $p = i + 1, \ldots, m,\ s.t. (i, p) \in P$ do
   $\partial \tilde{J} / \partial f(x_i) \leftarrow \partial \tilde{J} / \partial f(x_i) + \partial \tilde{J} / \partial D_{i,p} * \partial D_{i,p} / \partial f(x_i)$
   $\partial \tilde{J} / \partial f(x_p) \leftarrow \partial \tilde{J} / \partial f(x_p) + \partial \tilde{J} / \partial D_{i,p} * \partial D_{i,p} / \partial f(x_p)$
   end
   for $n = 1, \ldots, m,\ s.t. (i, n) \in N$ do
   $\partial \tilde{J} / \partial f(x_i) \leftarrow \partial \tilde{J} / \partial f(x_i) + \partial \tilde{J} / \partial D_{i,n} * \partial D_{i,n} / \partial f(x_i)$
   $\partial \tilde{J} / \partial f(x_n) \leftarrow \partial \tilde{J} / \partial f(x_n) + \partial \tilde{J} / \partial D_{i,n} * \partial D_{i,n} / \partial f(x_n)$
   end
end
```

Algorithm 1 Reverse propagation gradient update

III. METRIC LEARNING

A. System design

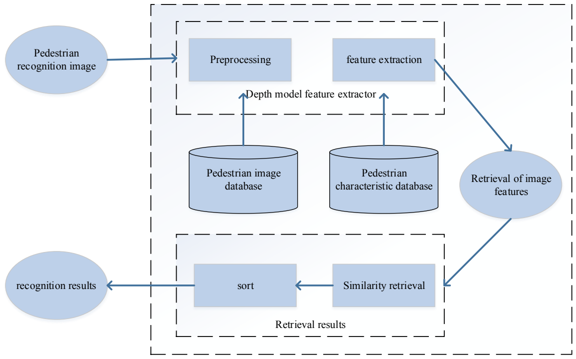

Pedestrian recognition image
Preprocessing
feature extraction
Depth model feature extractor
Pedestrian image database
Pedestrian characteristic database
Retrieval of image features
recognition results
sort
Similarity retrieval
Retrieval results

Fig. 10 Pedestrian re recognition system framework

The pedestrian data studied in this paper is to intercept the pedestrian from the monitoring video manually or by using the existing pedestrian image detection technology. Only the pedestrian re recognition operation is carried out alone. Therefore, in the actual monitoring environment, when deploying pedestrians to identify the system, it is necessary to include pedestrian detection components, pedestrians re identification parts and so on. But pedestrian detection technology is not in our research scope. We only carry out pedestrian re recognition, that is pedestrian image retrieval. Our system flow is shown in Figure 10.

B. Algorithm evaluation index

The algorithm is effective and the performance is exactly what it needs to be done. In general, we use the same data set to compare the accuracy of the algorithm to evaluate the performance of the algorithm, and the validity of the algorithm can be obtained. In this experiment, our first evaluation index is the CMC curve.

In the experiment, the general data set is divided into training data and test data, and usually the former is several times the latter. The retrieval task for re recognition of pedestrians is more special, and the test data is divided into two halves, and it is divided according to the camera from the image source. One part is used as a retrieval test set, also known as a search set, in which the images belong to the same camera, the other is a retrieved detection set, and also known as a candidate set, in which the images belong to one or more cameras.

In the process of the experiment, we extract the pedestrian picture of the query search set and the gallery candidate set, then calculate the Euclidean distance between the two set features, and make the distance in the gallery from the small to the large. Retrieval performance is usually represented by Cumulative Match Characteristic (CMC). The CMC curve represents the probability that the retrieved pedestrian image should be included in the foreground picture of the candidate set after sorting, that is, to retrieve the correct probability. The calculation formula is as follows:

$$CMC(R) = \frac{1}{N} \sum_{i=1}^{N} \begin{cases} 1 & r_i \le R \\ 0 & r_i \ge R \end{cases} \tag{16}$$

Among them, N represents the pedestrian images to be retrieved, and also indicates the search and sorting of N times, and r_i represents the position of the i pedestrian search, and R is the TOPK of the sort. The experimental results show the experimental results with Top1, top5 and TOP10.

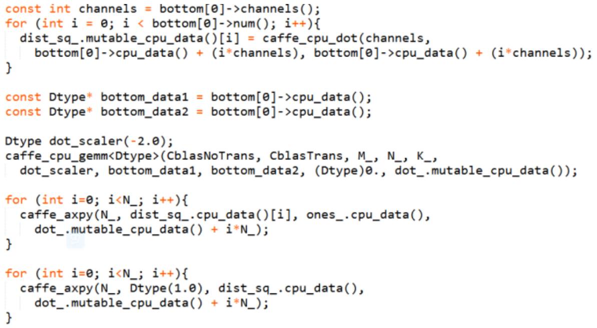
Another important indicator of pedestrian re recognition is mAP (mean Average Precision), which is a response to the average performance of all test sets. The calculation formula is as follows:

$$mAP = \int_0^1 P(R) dR \tag{17}$$

Among them, R refers to the recall rate, $P(R)$ is the recall rate is R accuracy rate.

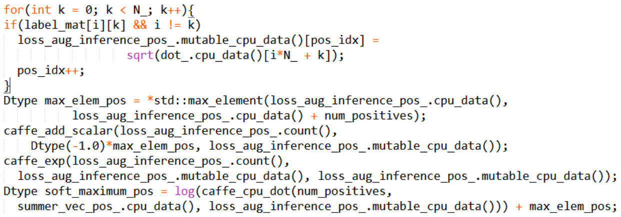
C. How to implement the experiment simulation

The experiment is based on the open source deep learning framework (Caffe) to train the model, because Caffe has the network data input layer, convolution layer, pool layer, the forward propagation and reverse propagation of the full connection layer, so our main work is to implement the loss layer of the framework, which is to make our improved loss function real. Now. We will introduce how to implement the loss layer specifically, but we will omit some unimportant code.



```
const int channels = bottom[0]->channels();
for (int i = 0; i < bottom[0]->num(); i++){
  dist_sq_.mutable_cpu_data()[i] = caffe_cpu_dot(channels,
    bottom[0]->cpu_data() + (i*channels), bottom[0]->cpu_data() + (i*channels));
}

const Dtype* bottom_data1 = bottom[0]->cpu_data();
const Dtype* bottom_data2 = bottom[0]->cpu_data();

Dtype dot_scaler(-2.0);
caffe_cpu_gemm<Dtype>(CblasNoTrans, CblasTrans, M_, N_, K_,
  dot_scaler, bottom_data1, bottom_data2, (Dtype)0., dot_.mutable_cpu_data());

for (int i=0; i<N_; i++){
  caffe_axpy(N_, dist_sq_.cpu_data()[i], ones_.cpu_data(),
    dot_.mutable_cpu_data() + i*N_);
}

for (int i=0; i<N_; i++){
  caffe_axpy(N_, Dtype(1.0), dist_sq_.cpu_data(),
    dot_.mutable_cpu_data() + i*N_);
}
```

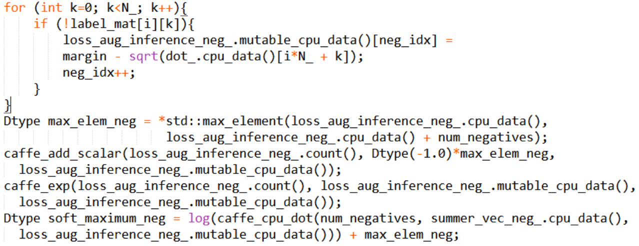
First, calculate the distance between the input images $D_{ij}^2 = \| f(x_i) - f(x_j) \|_2^2$, the variable is dot_, the code is as follows:



```
for(int k = 0; k < N_; k++){
if(label_mat[i][k] && i != k)
  loss_aug_inference_pos_.mutable_cpu_data()[pos_idx] =
                  sqrt(dot_.cpu_data()[i*N_ + k]);
  pos_idx++;
}
Dtype max_elem_pos = *std::max_element(loss_aug_inference_pos_.cpu_data(),
          loss_aug_inference_pos_.cpu_data() + num_positives);
caffe_add_scalar(loss_aug_inference_pos_.count(),
    Dtype(-1.0)*max_elem_pos, loss_aug_inference_pos_.mutable_cpu_data());
caffe_exp(loss_aug_inference_pos_.count(),
  loss_aug_inference_pos_.mutable_cpu_data(), loss_aug_inference_pos_.mutable_cpu_data());
Dtype soft_maximum_pos = log(caffe_cpu_dot(num_positives,
  summer_vec_pos_.cpu_data(), loss_aug_inference_pos_.mutable_cpu_data())) + max_elem_pos;
```

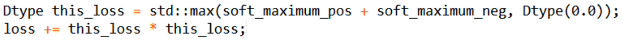
Secondly, the realization of formula (12), this part of the calculation we divide into about (13) and two parts of the implementation, the first to the left, the left part of the variable is soft_maximum_pos, the code as follows:

Furthermore, the right part of formula (13), the right part of variable soft_maximum_neg, is implemented as follows:



```
for (int k=0; k<N_; k++){
    if (!label_mat[i][k]){
        loss_aug_inference_neg_.mutable_cpu_data()[neg_idx] =
        margin - sqrt(dot_.cpu_data()[i*N_ + k]);
        neg_idx++;
    }
}
Dtype max_elem_neg = *std::max_element(loss_aug_inference_neg_.cpu_data(),
                       loss_aug_inference_neg_.cpu_data() + num_negatives);
caffe_add_scalar(loss_aug_inference_neg_.count(), Dtype(-1.0)*max_elem_neg,
  loss_aug_inference_neg_.mutable_cpu_data());
caffe_exp(loss_aug_inference_neg_.count(), loss_aug_inference_neg_.mutable_cpu_data(),
  loss_aug_inference_neg_.mutable_cpu_data());
Dtype soft_maximum_neg = log(caffe_cpu_dot(num_negatives, summer_vec_neg_.cpu_data(),
  loss_aug_inference_neg_.mutable_cpu_data())) + max_elem_neg;
```

Finally, the (13) formula of the loss function is implemented as follows:



```
Dtype this_loss = std::max(soft_maximum_pos + soft_maximum_neg, Dtype(0.0));
loss += this_loss * this_loss;
```

D. *Caffe training parameters setting and tuning model*

The experiment used in this experiment is to pre train two skeleton networks under ImageNet, AlexNet and VGGNet (V) introduced in the section C, and AlexNet also known as CaffeNet (C). Fig. 10 shows the change in the number of CaffeNet iterations and the loss function, and it can be seen from the diagram that the fine-tuning model is iterative to 3000 times and has begun to converge.

Base_lr in Table 1 is the initial learning rate, lr_policy is the learning rate change strategy, gamma is the change of the ratio, the gamma is the change ratio, the stepsize is the number of iterations after the learning rate needs to change, display is the number of iterations to print the information on the screen, max_iter is the maximum number of training iterations, momentum is the initial momentum, snapshot is every other time. After many iterations, the training model needs to be saved immediately. Snapshot_prefix is the path to save the model. Solver_mode indicates that the training is GPU or CPU.

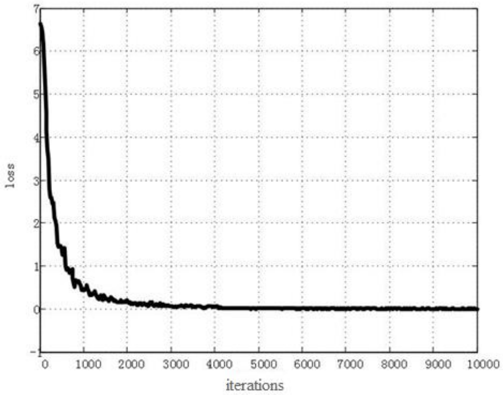


Fig. 10 The curve of the number of iterations and the change of the loss function

All experiments in this paper are based on the deep learning framework Caffe:

Table 1 parameter setting when training model

	Parameter name		Parameter value
✧	base_lr	✧	0.001
✧	lr_policy	✧	“step”
✧	gamma	✧	0.1
✧	stepsize	✧	3000
✧	display	✧	50
✧	max_iter	✧	10000
✧	momentum	✧	0.9
✧	weight_decay	✧	0.0002
✧	snapshot	✧	3000
✧	snapshot_prefix	✧	“output/”
✧	solver_mode	✧	GPU

E. *Data set*

In this paper, we use two data sets to conduct experimental evaluation of pedestrian re recognition, including Market-1501[19], CUHK03[20]. Market-1501 contains 1501 pedestrians. There are 19732 pedestrian pictures in the candidate set. There are 3368 pictures of the pedestrians to be retrieved, and the training set is 12936, all of which are monitored by 6 cameras. All of the bounding boxes are detected by the DPM detector, as shown in fig.11. The CUHK03 dataset contains 13164 pictures of 1467 pedestrians. Each of the pedestrians was captured by two cameras. CUHK03 provides a bounding box obtained by manually annotating the bounding box and DPM, and we use the latter in the experiment. Similar to Market-1501, the experiment also divides CUHK03 datasets into training sets and test sets, which contain 767 pedestrians and 700 pedestrians respectively. Examples are shown in Figure 12. At the time of testing, we randomly selected one picture from each camera to retrieve the picture, and the rest of the pictures constituted the candidate set. We ensure that every retrieved pedestrian is captured by two cameras, so that cross camera retrieval can be completed.

Fig.11 Market-1501 pedestrian picture example

Fig.12 CUHK03 pedestrian picture example

F. *Analysis of experimental results*

After the model is trained, all the probe and gallery images are input into the convolution neural network model, and the network is propagating forward. We will extract the FC6 and FC7 in the network, that is, the full connection layer, as our feature mapping, and then evaluate the results. Only to extract the features of FC6 and FC7 is that the forward propagation of the network forward through the convolution layer to the layer by layer abstraction of the full connection layer has been able to achieve the ability to represent the sample, which is the expectation of our training model.

The experiment first compares the performance of the proposed method and benchmark method [21], using CaffeNet (C), respectively.VGGNet (V) model and data set in Market-1501 and CUHK03. After that, we compare the loss based on triplet and the method in this paper.

This paper comprehensively evaluated the proposed method in comparison with the above two data sets and benchmark methods. The overall results are shown in table 2. The benchmark (C) FC6 in the table indicates that the benchmark method is characterized by the FC6 layer of CaffeNet, and the other is similar. The performance enhancement in the two skeleton network models is still obvious: when CaffeNet is used as a skeleton network, the Rank-1 in Market-1501 is raised from 57.45% to 67.28%, and mAP is raised from 31.53% to 42.18. In CUHK03, Rank-1 increased from 21.30% to 48.19%, and mAP increased from 19.70% to 33.46%. When using VGGNet as skeleton network, Market-1501's Rank-1 increased from 69.63% to 78.41%, and mAP increased from 43.62% to 55.82%. In CUHK03, Rank-1 increased from 51.65% to 62.52, and mAP increased from 34.28% to 42.16%.

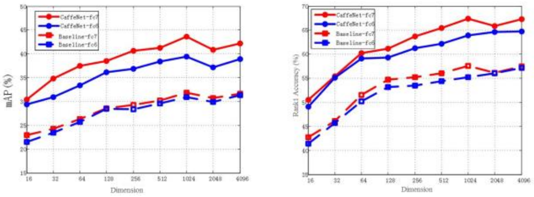


Fig.13 mAP and Rank1 based on CaffeNet

Table 2 comparison of the benchmark method and the performance of this method

Model &	Di me nsi	Market-1501				CUHK03			
Feature	on	R-1	R-5	R-10	mAP	R-1	R-5	R-10	mAP
Benchmark	40	57.16	77.61	84.44	31.31	20.56	55.18	66.78	17.60
(C)FC6	96								
Benchmark	40	57.45	77.67	84.23	31.53	21.30	56.98	67.19	19.70
(C)FC7	96								
Text	40	64.73	83.55	89.01	38.88	46.84	74.21	84.13	30.32
(C)FC6	96								
Text	40	***67.28***	***85.10***	***90.05***	***42.18***	***48.19***	***75.34***	***86.17***	***33.46***
(C)FC7	96								
Benchmark	40	67.90	84.71	89.61	40.77	49.89	78.54	86.02	32.39
(V)FC6	96								
Benchmark	40	69.63	85.27	89.55	43.62	51.65	81.63	86.29	34.28
(V)FC7	96								
Text	40	75.87	86.74	90.69	51.28	60.37	81.15	88.56	40.97
(V)FC6	96								
Text	40	***78.41***	***87.19***	***92.36***	***55.82***	***62.52***	***84.89***	***89.14***	***42.16***
(V)FC7	96								

We also change the dimension of output features. The results of market-1501 based on CaffeNet and VGGNet are shown in figures 13 and14 respectively. It can be seen from the graph that with the increase of the output dimension, the Rank1 accuracy and mAP also increase, and the 2048 dimension to the output dimension tends to be gentle. Moreover, the mAP and Rank1 of any backbone FC7 are superior to those of FC6.

To illustrate that this method is better than the triplet loss method, the next method is based on the above two data sets and triplet loss methods. Because the triplet method is better than the benchmark method, we are no longer in parity. The results are shown in table 3. As shown in table 3 above, when CaffeNet is used as skeleton network, the Rank-1 of Market-1501 is increased from 60.26% to 67.28%, and mAP is increased from 37.54% to 42.18. In CUHK03, Rank-1 increased from 31.68% to 48.19%, and mAP increased from 27.67% to 33.46%. When using VGGNet as skeleton network, Market-1501's Rank-1 increased from 73.78% to 78.41%, and mAP increased from 49.73% to 55.82%. In CUHK03, Rank-1 increased from 56.59% to 62.52, and mAP increased from 38.20% to 42.16%.

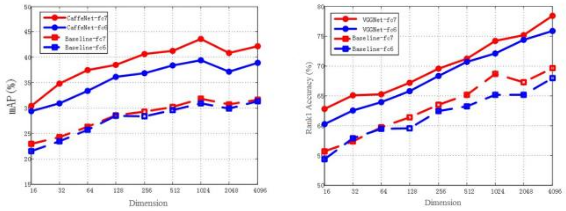


Fig.14 mAP and Rank1 based on VGGNet

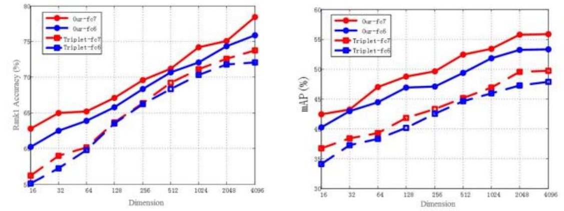


Fig.15 mAP and Rank1 based on CaffeNet

Table 3 Triplet method and the performance comparison of this method

Model & Feature	Dime-nsion	Market-1501				CUHK03			
		R-1	R-5	R-10	mAP	R-1	R-5	R-10	mAP
Tri(C)FC6	4096	59.47	79.12	86.49	37.21	30.45	65.09	77.27	24.96
Tri(C)FC7	4096	60.26	80.16	86.87	37.54	31.68	65.28	78.49	27.67
Text (C)FC6	4096	64.73	83.55	89.01	38.88	46.84	74.21	84.13	30.32
Text (C)FC7	4096	***67.28***	***85.10***	***90.05***	***42.18***	***48.19***	***75.34***	***86.17***	***33.46***
Tri(V)FC6	4096	72.10	85.09	89.61	47.87	54.37	80.18	86.48	37.59
Tri(V)FC7	4096	73.78	85.89	89.55	49.73	56.59	83.04	86.70	38.20
Text (V)FC6	4096	75.87	86.74	90.69	51.28	60.37	81.15	88.56	40.97
Text (V)FC7	4096	***78.41***	***87.19***	***92.36***	***55.82***	***62.52***	***84.89***	***89.14***	***42.16***

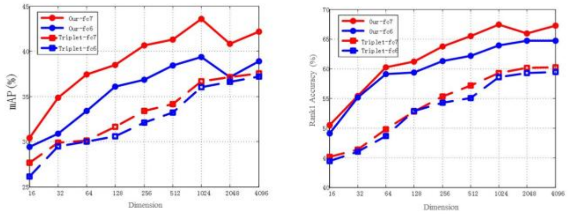


Fig.16 mAP and Rank1 based on VGGNet

In the same way, the output characteristic dimension is also changed. The results of market-1501 based on CaffeNet and VGGNet are shown in figures 15 and 16 respectively. It can be seen from the graph that with the increase of the output dimension, the Rank1 accuracy and mAP also increase, and the 2048 dimension to the output dimension tends to be gentle. Moreover, the mAP and Rank1 of any backbone FC7 are superior to those of FC6.

Experimental results show that our method is effective and the performance is improved.

IV. CONCLUSIONS

In order to illustrate our experimental results and how the theory corresponds, we propagate the 5000 test pictures in front of the model, extract the 4096 dimensional feature vectors of the FC7 layer, and then reduce the 4096 dimension eigenvectors to the 2 dimensional eigenvectors, then draw the corresponding pictures in the 2 dimensional Cartesian coordinates. From the picture, the distance between the same person in the 2 dimensional space is relatively close, and the distance between different people is far cheaper. So it can well illustrate that our theory is effective in drawing close distance and expanding different distances.

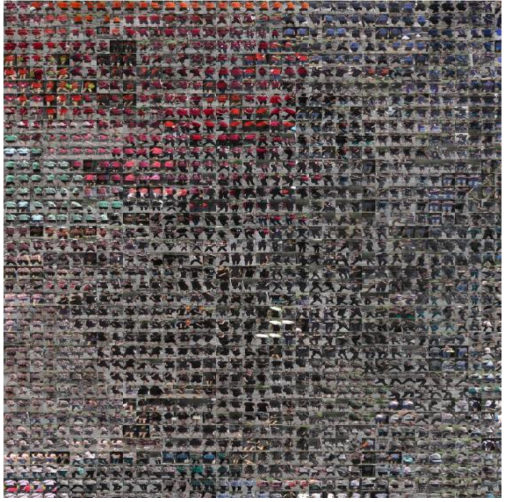


Fig.17 Market-1501 test set feature mapping

This paper first introduces pedestrian recognition in sub domain of image retrieval, and then presents some related work. Because the basic neural network structure used in this paper is CaffeNet and VGGnet, the development process of convolution neural network is introduced. In view of the shortage of three tuple network based on the weight of learning network, it is not to give full play to the advantage of training Batch. The relationship of three tuples has been determined before training. Therefore, this paper improves the loss function of three tuples and evaluates it on two open data sets. The experimental results show that our method is superior to the three tuple loss function.

ACKNOWLEDGMENT

This work is supported by the National Natural Science Foundation of China (grant No.61540063, 61163061, 60963026), the Application Basic Research Fund of Yunnan Province (grant No.2016FD058) , Yunnan Provincial Department of Education Science Research Fund Project(grant No.2017ZDX045) , Yunnan MinZu University project(No. 2017QN02) and Program for Innovative Research Team (in Science and Technology) in University of Yunnan Province.

REFERENCES

[1] D. Gray, H. Tao, “Viewpoint invariant pedestrian recognition with an ensemble of localized features, in: ECCV, ” Springer, 2018, pp. 262–275.

[2] M. Hirzer, C. Beleznai, P. M. Roth, and H. Bischof. “Person reidentification by descriptive and discriminative classification”J. In Image Analysis, Springer, 2017, pp. 91–102.

[3] L. Lin, P. Luo, X. Chen, K. Zeng, “Representing and recognizing objects with massive local image patches,”J. Pattern Recognit. 45 (1) (2016) 231–240.

[4] M. Farenzena, L. Bazzani, A. Perina, V. Murino, M. Cristani, Person reidentification by symmetry-driven accumulation of local features[C]. in: CVPR, IEEE, 2010, pp. 2360–2367.

[5] L. Lin, X. Wang, W. Yang, J. Lai, Discriminatively trained and-or graph models for object shape detection[J]. IEEE Trans. Pattern Anal. Mach. Intell. 37 (5) (2015) 959–972.

[6] Y. Xu, L. Lin, W.-S. Zheng, X. Liu, Human re-identification by matching compositional template with cluster sampling[C]. in: IEEE International Conference on Computer Vision (ICCV), IEEE, 2013, pp. 3152–3159.

[7] W.-S. Zheng, S. Gong, T. Xiang, Person re-identification by probabilistic relative distance comparison[C]. in: CVPR, IEEE, 2017, pp. 649–656.

[8] A. Mignon, F. Jurie, Pcca: A new approach for distance learning from sparse pairwise constraints[C]. in: CVPR, IEEE, 2012, pp. 2666–2672.

[9] Z. Li, S. Chang, F. Liang, T.S. Huang, L. Cao, J.R. Smith, Learning locally-adaptive decision functions for person verification[C]. in: CVPR, IEEE, 2016, pp. 3610–3617.

[10] D. Yi, Z. Lei, S.Z. Li, Deep metric learning for practical person re-identification, CoRR abs/1407.4979.

[11] H. Liu, B. Ma, L. Qin, J. Pang, C. Zhang, Q. Huang, Set-label modeling and deep metric learning on person re-identification[J]. Neurocomputing 151 (2017) 1283–1292.

[12] L. Lin, X. Liu, S.-C. Zhu, Layered graph matching with composite cluster sampling[J]. IEEE Trans. Pattern Anal. Mach. Intell. 32 (8) (2010) 1426–1442.

[13] R. B. Girshick. Fast R-CNN[C]. ICCV.2015: 1440-1448.

[14] Sun Y, Liang D, Wang X, et al. DeepID3: Face recognition with very deep neural network[J]. arXiv preprint arXiv: 1502.00873, 2015.

[15] Szegedy C, Liu W, Jia Y, et al. Going deeper with convaluton[C] . IEEE Conference on Computer Vision and Pattern Recognition (CVPR). 2015: 1-9.

[16] He K, Zhang X, Ren S, et al. Deep residual learning for image recognition[C]. IEEE Conference on Computer Vision and Pattern Recognition (CVPR). 2016:770-778.

[17] E.P. Xing, M.I. Jordan, S. Russell, A.Y. Ng, Distance metric learning with application to clustering with side-information[J]. in: NIPS, 2002, pp. 505–512.

[18] Schroff F, Kalenichenko D, Philbin J. FaceNet: A unified embedding for face recognition and clustering[C]. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2015: 815-823

[19] L. Zheng, L. Shen, L. Tian, S. Wang, J. Wang, and Q. Tian. Scalable person re-identification: A benchmark[C]. In ICCV, 2015.

[20] W. Li, R. Zhao, T. Xiao, and X. Wang. Deepreid: Deep filter pairing neural network for person re-identification[C]. In CVPR,2018.

[21] L. Zheng, Y. Yang, and A. G. Hauptmann. Person reidentification: Past, present and future[J]. arXiv preprint arXiv:1610.02984, 2016.