Hindawi
Complexity
Volume 2019, Article ID 5069026, 16 pages
https://doi.org/10.1155/2019/5069026

Research Article

Pedestrian Re-Recognition Algorithm Based on Optimization Deep Learning-Sequence Memory Model

Feng-Ping An [1,2]

[1]*School of Physics and Electronic Electrical Engineering, Huaiyin Normal University, Huaian 223300, China*
[2]*School of Information and Electronics, Beijing Institute of Technology, Beijing 100081, China*

Correspondence should be addressed to Feng-Ping An; anfengping@163.com

Received 2 March 2019; Revised 3 July 2019; Accepted 5 September 2019; Published 14 October 2019

Guest Editor: Ivan Marsa-Maestre

Pedestrian re-recognition is an important research because it affects applications such as intelligent monitoring, content-based video retrieval, and human-computer interaction. It can help relay tracking and criminal suspect detection in large-scale video surveillance systems. Although the existing traditional pedestrian re-recognition methods have been widely applied to address practical problems, they have deficiencies such as low recognition accuracy, inefficient computation, and difficulty to adapt to specific applications. In recent years, the pedestrian re-recognition algorithms based on deep learning have been widely used in the pedestrian re-recognition field because of their strong adaptive ability and high recognition accuracy. The deep learning models provide a technical approach for pedestrian re-recognition tasks with their powerful learning ability. However, the pedestrian re-recognition method based on deep learning also has the following problems: First, the existing deep learning pedestrian re-recognition methods lack memory and prediction mechanisms, and the deep learning methods offer only limited improvement to pedestrian re-recognition accuracy. Second, they exhibit overfitting problems. Finally, initializing the existing LSTM parameters is problematic. In view of this, this paper introduces a revertive connection into the pedestrian re-recognition detector, making it more similar to the human cognitive process by converting a single image into an image sequence; then, the memory image sequence pattern reidentifies the pedestrian image. This approach endows deep learning-based pedestrian re-recognition algorithms with the ability to memorize image sequence patterns and allows them to reidentify pedestrians in images. At the same time, this paper proposes a selective dropout method for shallow learning. Selective dropout uses the classifier obtained through shallow learning to modify the probability that a node weight in the hidden layer is set to 0, thereby eliminating the overfitting phenomenon of the deep learning model. Therefore, this paper also proposes a greedy layer-by-layer pretraining algorithm for initializing LSTM and obtains better generalization performance. Based on the above explanation, this paper proposes a pedestrian re-recognition algorithm based on an optimized LSTM deep learning-sequence memory learning model. Experiments show that the pedestrian re-recognition method proposed in this paper not only has strong self-adaptive ability but also identifies the average accuracy. The proposed method also demonstrates a significant improvement compared with other mainstream methods because it can better memorize and learn the continuous motion of pedestrians and effectively avoid overfitting and parameter initialization in the deep learning model. This proposal provides a technical method and approach for adaptive pedestrian re-recognition algorithms.

1. Introduction

In recent years, with the rapid development of the social economy and the continuous updating of science and technology, our lives have undergone rapid changes and many social problems have gradually been exposed, among which security issues are the most prominent [1]. Security video surveillance constitutes a large-scale distributed monitoring system, and the amount of monitoring data is exploding [1–3]. To both participants and leaders of the society, in-depth research on human behavior in videos is important. Simultaneously, pedestrian recognition is an important research topic in intelligent monitoring, content-based video retrieval, human-computer interaction, and other applications. The study of pedestrian re-recognition has been actively promoted in industries such as

transportation and public security criminal investigations [4–6]. Pedestrian re-recognition can be applied to relay tracking and suspect detection in large-scale video surveillance systems, and it is highly significant in improving the intelligence and functionality of video surveillance systems [7].

Pedestrian re-recognition research originated with a multicamera tracking study [8]. Gheissar proposed the concept of pedestrian re-recognition at the 2006 CVPR and used color and significant edge line histogram features to achieve pedestrian reidentification [9]. In 2007, the first dataset dedicated to the study of pedestrian re-recognition VIPeR was published [10]. Since then, pedestrian recognition has received increasing attention from researchers. At prestigious international conferences such as CVPR, ICCV, and AAAI, pedestrian recognition research results are published every year, and their numbers have increased year over year [11]. In recent years, a large number of pedestrian re-recognition research results have been published in the top internationally renowned journals, such as the International Journal of Computer Vision, IEEE Transactions on Pattern Analysis and Machine Intelligence, and IEEE Transactions on Image Processing [12–15]. Among these publications, Vezzani et al. [16] and Bedagkar-Gala et al. [17] provided a review of pedestrian re-recognition research progress in 2013 and 2014. To facilitate the study of pedestrian re-recognition, many datasets specifically designed to test the performance of pedestrian recognition algorithms have also been published. Table 1 lists information such as the release time, number of rows, number of pictures, and number of camera views in some of these commonly used datasets. These datasets provide a unified comparison platform for evaluating the performances of pedestrian recognition algorithms. These data were all collected from videos captured by actual surveillance cameras. Pedestrian re-recognition experiments conducted on these datasets can better simulate the cross-camera pedestrian identity matching task in actual monitoring scenarios. Pedestrian re-recognition methods can be divided into those based on manually designed features and those based on deep learning techniques [18, 19].

The following is an analysis and summary of the pedestrian re-recognition method based on manual design, which mainly includes pedestrian detection, pedestrian image feature extraction, and distance metrics. Pedestrian re-recognition methods based on manual design are devoted to feature extraction research and pedestrian image distance measurement. Manually designed feature extraction methods commonly use colors and textures for feature extraction. For example, HSV/RGB histograms are used to obtain image color information and the local binary pattern (LBP) [26], and Gabor filter features can be used. The texture information of the captured image, the histogram of the oriented gradient (HOG) [27], and the scale-invariant feature transform (SIFT) [28] can capture image shape information. In practice, the basic features of different categories, such as color, texture, and shape, are usually concatenated to obtain more discriminative feature representations. However, concatenation also causes the final feature expression vector to have a relatively high dimension. Additionally, representative distance metrics for pedestrian image distance measurements in the pedestrian re-recognition field include the following. Liao and Li [29] proposed a metric learning model based on the log-logistic loss function to solve the accelerated proximal gradient (APG) and the strategy of positive and negative samples for asymmetric weighting and introduced the semipositive constraints of the metric matrix. To reduce the high computational cost in the general metric learning method solution, Koestinger et al. [30] proposed the Keep It Simple and Straightforward MEtric (KISSME) learning method in which the metric matrix has an efficient closed-form solution that does not require iterative optimization. However, KISSME is sensitive to the dimensions of the feature expression vectors. Many researchers have proposed ways to improve KISSME's performance [31, 32]. Liao et al. [33] proposed a cross-view quadratic discriminant analysis (XQDA) algorithm that learned a metric matrix while also learning a more discriminative projection matrix to reduce the feature dimension. However, such methods still suffer from low recognition accuracy, low computational efficiency, and weak adaptive ability.

Because of the above problems in the traditional pedestrian re-recognition method, many scholars have begun to study pedestrian re-recognition methods based on deep learning. The earliest research work on deep learning in the field of pedestrian re-recognition began in 2014 with Li et al. [34, 35]. Increasing deep learning-based work has emerged in the field of pedestrian re-recognition [36–42]. In general, two types of deep learning models, verification and classification models, are widely used in pedestrian re-recognition tasks.

(1) Pedestrian Re-Recognition Based on the "Verification" Model. Li et al. [35] proposed adding a series of patch-matching layers after the patch-based convolution response to more accurately constrain the similarity between pedestrian images. Ahmed et al. [11] learned a "cross-image" representation by calculating the distance between a sample and its neighborhood, improving the Siamese structure and enhancing model robustness. Wu et al. [43] improved the discriminative ability of deep learning models by increasing the number of layers in the network structure and reducing the size of the convolution filter in the Siamese model. Varior et al. [44] proposed inserting a gating function to capture effective small details after the convolutional layer in the Siamese model. Liu et al. [14] proposed integrating a soft attention model into the Siamese model to adaptively focus on the more important local parts. Su et al. [45] proposed a three-stage learning process, which includes the predicting attributes and the use of the loss function of the triples on the basis of attributes to train the deep learning model. Ding et al. [46] proposed a novel and effective triplet generation scheme and an optimized gradient descent method based on the triplet-based network structure to change the human re-recognition training process. They obtained a pedestrian deep feature representation with stronger discriminating ability, which improved the pedestrian recognition performance.

Table 1: Common datasets for pedestrian re-recognition research.

Dataset	Public time	Number of people	Number of cameras	Labeling method	Evaluation index
VIPeR [20]	2007	632	2	Hand	CMC
GRID [21]	2009	119	2	Hand	CMC
3DPeS [22]	2011	192	8	Hand	CMC
CUHK03 [23]	2014	1467	2	Hand/DPM	CMC
Market-1501 [24]	2015	1501	6	Hand/DPM	CMC/mAP
DukeMTMC-reID [25]	2017	1812	8	Hand/DPM	CMC/mAP

(2) Pedestrian Re-Recognition Based on the "Classification" Model. In the deep learning structure based on the verification model, the supervised information constrains the similarity between samples, but no specific annotation exists that corresponds to a certain sample. In contrast, the deep learning structure based on the "classification" model directly utilizes the specific annotation information of the pedestrian image, and the annotation content can be more fully utilized. Zheng et al. [47, 48] proposed a deep learning structure using the standard "classification" model and learned a highly discriminative pedestrian identity embedding in the pedestrian subspace, compared to the traditional feature plus distance metric learning paradigm. That model improved the pedestrian recognition performance by more than 20%. Su et al. [45] proposed using the singular vector decomposition strategy to decorrelate the learned weight vectors after the last fully connected layer in the deep learning model structure and improve the discriminative ability of the learned deep features. Zhong et al. [49] proposed a random erasure strategy based on the classification model, which both reduced the risk of overfitting during training and improved the robustness and discriminative ability of the model. Zheng et al. [50] proposed a pedestrian-aligned network based on the deep learning structure of the classification model. During deep learning model training, the alignment between pedestrians was realized. Additionally, this approach improves the ability to express pedestrians. Hermans et al. [51] verified that the ternary loss function achieves better matching accuracy than does the binary classification loss function on multiple large pedestrian re-recognition datasets. Subsequently, scholars proposed a variety of deep learning-based pedestrian re-recognition methods that have achieved various effects [52–55].

From the above analysis and summary, given their powerful learning abilities, deep learning models provide an effective solution to pedestrian re-recognition tasks. However, some problems still exist in the deep learning-based pedestrian re-recognition methods. First, the existing deep learning pedestrian re-recognition methods lack memory and prediction mechanisms. They can increase the accuracy of pedestrian recognition by adding layers to the convolutional neural network, but such improvements are quite limited. Second, the deep learning pedestrian recognition methods have an overfitting problem. Third, while the existing deep learning pedestrian re-recognition methods have achieved good results in various tasks using long short-term memory (LSTM), the problem of how to initialize the LSTM parameters is not well resolved because the objective function used in training is nonconvex and involves many local minima. Therefore, during deep LSTM training, the main challenge lies in effectively initializing the LSTM parameters. Therefore, this paper introduces a revertive connection into the pedestrian re-recognition detector based on the human cognitive process. Using this approach, the single image is converted into an image sequence, and a memory image sequence pattern is used for pedestrian recognition. This approach allows the deep learning-based pedestrian re-recognition algorithm to memorize image sequence patterns and thus gain the ability to reidentify pedestrians in pedestrian images. This paper proposes a selective dropout method based on shallow learning. This technique uses the classifier obtained by shallow learning to modify the probability that a node's weight in the hidden layer will be set to 0, thereby eliminating the overfitting phenomenon. Furthermore, this paper proposes a greedy layer-by-layer pretraining algorithm to initialize the LSTM. It trains the model in a layer-by-layer fashion through greedy strategies, using each layer of the unsupervised learning process to preserve the input information. Then, using gradient-based optimization, the entire network undergoes supervised fine tuning based on the final task to achieve better generalizability. Therefore, the parameters learned at this stage are better able to initialize the network in subsequent supervised learning tasks. Based on the above explanation, this paper proposes a pedestrian re-recognition algorithm based on an optimized LSTM deep learning-sequence memory learning model.

Section 2 describes the deep learning model of the LSTM-shallow learning selective dropout proposed in this paper. It mainly introduces the greedy strategy applied to the LSTM training process. Section 3 elaborates on the sequence memory learning model proposed in this paper. Section 4 constructs the pedestrian re-recognition algorithm based on the optimized LSTM deep learning-sequence memory learning model proposed in this paper. Section 5 analyzes the model proposed in this paper using examples and compares the results with those of popular mainstream person re-recognition algorithms. Finally, the full text is summarized and discussed.

2. Deep Learning Model Based on LSTM-Shallow Learning Selective Dropout

The first half of this section will elaborate on the training of long- and short-term memory networks through a multilayer autoencoder and propose a greedy layer-by-layer LSTM training model. It can solve the problem of parameter initialization of

deep learning models. Then, in order to better eliminate or avoid the overfitting problem of the deep learning model, the second half of this section will introduce the shallow learning selective dropout method to solve the problem. On this basis, in order to make better use of pedestrians, the characteristics of image memory and sequence are re-recognized. In Section 3, a pedestrian re-recognition model based on sequence memory learning is proposed and embedded into the deep learning model proposed in this section. Finally, the model is used for pedestrian re-recognition in various complex scenes. The specific framework is shown in Figure 1.

It can be known from the above that the CNN structure of deep learning is complex and training is difficult [56, 57]. Specific to this part, first, Section 2.1 elaborates on how to construct a greedy layer-by-layer LSTM training model to solve the problem of parameter initialization of deep learning models. Then, Section 2.2 elaborates on how to use the shallow learning selective dropout method to solve the overfitting problem of the deep learning model.

2.1. LSTM Unsupervised Training. Using a random initialization approach during training can easily cause a deep learning model to converge to a poor local minimum, which results in slower convergence and lower performance. Unsupervised layered pretraining using the stacked autoencoder method can better solve these problems. At the same time, the LSTM autoencoder shows a good ability to learn sequential representations.

2.1.1. Autoencoder and Multilayer Autoencoder Training. An autoencoder is trained to encode the input x into some representation $c(x)$ that can reconstruct the input by $f(c(x))$, where $c(\cdot)$ represents the encoder and $f(\cdot)$ represents the decoder. In general, the loss function of such an automatic encoder can be defined as a cross-extraction error, which is $-\sum_i p_i \log \widehat{p}_i - \sum_i (1 - p_i)\log(1 - \widehat{p}_i)$ or the Euclidean distance $\sum_i (x_i - \widehat{x}_i)^2$.

A stacked automatic encoder can be used to initialize a deep multilayer network. The basic training steps are as follows:

(1) The first layer is trained as an automatic encoder to minimize the reconstruction error of the original input.

(2) The output of the autoencoder is then used as the input to the next layer, which is also trained as an autoencoder.

(3) Step (2) is iterated to initialize the required number of layers.

(4) The output of the last hidden layer is entered into the new supervisory layer.

(5) All the parameters of the deep structure are fine tuned using a supervised or unsupervised loss function.

2.1.2. LSTM Training and Autoencoding

(1) LSTM Training. The recurrent neural network (RNN) has achieved great success in sequence learning tasks. However,

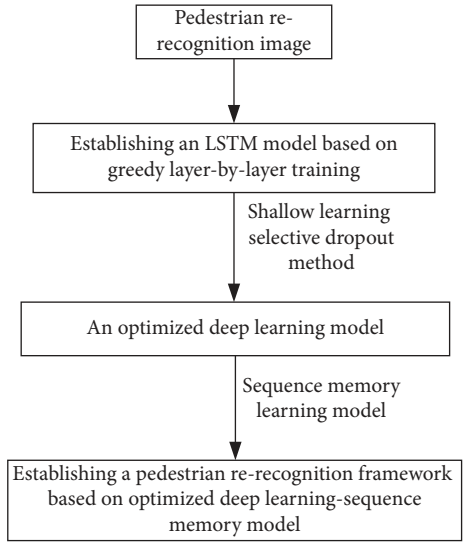

Figure 1: Pedestrian re-recognition framework based on LSTM-sequence memory learning.

RNNs can have gradient vanishing or exploding problems, which cause modeling difficulties. One of the most effective methods for solving these problems is to use the LSTM architecture. The LSTM network introduces a new structure called the memory unit to store long-term dependencies. The storage unit has three main elements: an input gate, a forget gate, and an output gate. The input gate writes input information to the memory, and the forget gate and the output gate determine whether the information is saved or released from memory at each decision point. The variants of LSTM do not show large performance differences [56, 57]. Therefore, this paper uses the common LSTM described in [56, 57], in which the gate, memory unit, and hidden layer output are calculated as follows:

$$\begin{aligned} i^t &= \text{sigmoid}\left(W^{xi}x^t + W^{hi}h^{t-1} + b^i\right), \\ o^t &= \text{sigmoid}\left(W^{xo}x^t + W^{ho}h^{t-1} + b^o\right), \\ f^t &= \text{sigmoid}\left(W^{xf}x^t + W^{hf}h^{t-1} + b^f\right), \\ g^t &= \tanh\left(W^{xg}x^t + W^{hg}h^{t-1} + b^g\right), \\ c^t &= f^t \odot c^{t-1} + i^t \odot g^t, \\ h^t &= o^t \odot \tanh\left(c^t\right), \end{aligned} \tag{1}$$

where W is the weight; b is the corresponding deviation vector; x^t, h^t, and c^t represent the input, output, and memory units, respectively, at time step t; h^{t-1} and c^{t-1} are the output and memory units at time step $t-1$; i^t, o^t, and f^t are the input, output, and forgetting gates; e represents a dot product operation; and $\text{sigmoid}(x) = 1/(1 + e^{-x})$ and $\tanh(x) = (e^x - e^{-x})/(e^x + e^{-x})$ represent nonlinear activation functions.

(2) LSTM Autoencoder. This consists of two LSTMs, one for encoding and one for decoding. The input to the model is a series of vectors (features or videos). The encoder LSTM will read all input sequences and encode them into fixed-length hidden outputs and memory units. The memory unit and the

concealed output of the encoder LSTM are then copied to the decoder LSTM, which are output as a reconstructed decoded sequence for the input sequence. The decoded sequence should be identical to the input sequence in either the original or reverse order. Reversing the target sequence should be easier because the model only needs to capture correlations in a small range. Therefore, this paper uses this structure to perform unsupervised pretraining on sequence classification tasks. Conversely, it can reconstruct the original sequence from the input sequence; therefore, the model needs to preserve the general structure and long-term correlation of the input sequence. This paper uses this model to learn the initialization of sequence-to-sequence learning tasks.

2.1.3. Greedy Layer-by-Layer LSTM Training. Using a random initialization of the standard gradient descent makes it difficult to train a deep neural network because the singular value of the Jacobian matrix of each layer is greater than 1; layer activations and changes in the gradient can easily lead to gradient vanishing or exploding problems. In this paper, the output of the layer $i+1$ is represented as z^{i+1} and the output of the layer i is z^i. The Jacobian matrix associated with the layer i is defined as follows:

$$J^i = \frac{\partial z^{i+1}}{\partial z^i} = \left\{W^i; f\right\}, \tag{2}$$

where the W^i value represents the weight of the layer i and f is the activation function. In practice, if W_s^i does not initialize properly according to a different f (making J~1), the gradient may exhibit different amplitudes on different layers, which can result in poor condition numbers (sensitive to small errors in the input) and slower training speeds. In a deep LSTM model, the gradient flows in two directions: between the same layers of the LSTM and between different layers of the LSTM. That is, in an LSTM layer l, the Jacobian matrix is

$$J^i = \frac{\partial h_l^{t+1}}{\partial h_l^t} = \{W_l; f\}. \tag{3}$$

Additionally, between the LSTM layer l and the LSTM layer $l+1$, the Jacobian matrix is

$$J_l^{l+1} = \frac{\partial h_l^{t+1}}{\partial h_l^t} = \{W_l; W_{l+1}; f\}, \tag{4}$$

where W_l and W_{l+1} are the weights in layers l and $l+1$, respectively, and h_{l+1}^t and h_l^t are the hidden atoms of layers $l+1$ and l, respectively, at time t. The relationships between h_l^{t+1} and h_l^t and h_{l+1}^t and h_l^t are not typical and cannot be expressed as explicit functions. Therefore, this paper cannot obtain an appropriate random initialization expression for the weight to avoid the vanishing or exploding gradient problem.

To obtain proper weight initialization in the deep LSTM model, this paper first uses the LSTM autoencoder to learn to ensure the constant gradient weight and activation value streams in a single LSTM layer. The hidden layer output of the previous layer is then used recursively as the input to the next LSTM autoencoder, ensuring a constant gradient flow across the LSTM layer by adjusting the weights. Finally, the supervised task gradient descent learning is started from the initialized parameters, to avoid vanishing or exploding gradient problems. Consequently, the model learns better and faster than a randomly initialized model.

Moreover, this layer-by-layer training process affects each LSTM layer by using the previous representation of the remembered sequence and reconstructing the original input from the representation. The extracted information becomes more abstract from the lower layers to the higher layers; thus, the model retains the most interesting and compact information and discards uncorrelated noise from the input. This approach helps the model avoid local optimal solutions in the parameter space. Therefore, this process can be considered as better than random initialization of deep LSTM networks during training. The process of training a multilayer LSTM is similar to that of training a stacking autoencoder. The specific steps are as follows:

(1) The first LSTM layer is trained as an LSTM autoencoder using the model in Figure 2. The input sequence is also used as an input to decode the LSTM.

(2) The hidden atom encoding the LSTM is input to the next LSTM autoencoder. To help the model learn the input sequence, the LSTM must be decoded to recover the original input sequence.

(3) The iterative process in step (2) initializes the required number of additional LSTM layers.

(4) The hidden output of the last LSTM layer is input into the supervisory layer.

(5) Finally, all the parameters of this deep structure are fine tuned by a supervised loss function.

According to the above process, the sequence pretraining framework in the sequence classification and sequence learning tasks of this paper is as follows. For sequence classification tasks, the input signal or feature $\{v_1 v_2 v_3\}$ is read into the encoder LSTM in the original sequence, and the decoder LSTM needs to reconstruct the input $\{v_1' v_2' v_3'\}$ in the reverse sequence. It is easier to reconstruct the signal in a small range. For sequence learning tasks, the goal is to predict the next element in the target sequence. Therefore, the decoded output needs to be the same sequence as the real input sequence $\{WXY\}$. However, in order to reduce the length of the dependency between the input and the predicted output $\{W'X'Y'\}$, correspondingly, inverting the input sequence is a similar process.

2.2. Shallow Learning Selective Dropout Method

2.2.1. Selective Dropout. Dropout technology is a key technology in deep learning that can effectively prevent network overfitting. In each batch of sample training, the weights of some neurons in the hidden layer are reduced to 0, thereby increasing the sparsity of the network. The weights

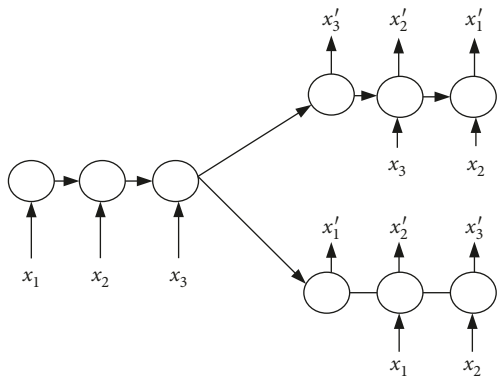

Figure 2: Two types of LSTM autoencoders: one that reconstructs the previous element and the other that is the next element in the sequence (x_i is an element in the input sequence, and x_i' is the prediction of x_i; the circles represent LSTM units).

are set to 0 randomly in dropout, which is an embodiment of the average model idea.

However, the values of each convolution kernel are not equivalent. For example, for object recognition, convolution kernels that describe edges and corners are often more important than convolution kernels that describe planes. Thus, a deep learning model that trains more convolution kernels to describe edges may be more efficient for describing objects. Therefore, randomly setting values to 0 during dropout is not the best approach; instead, a kernel should be set to 0 based on the importance of its weight. That is, the probability at which more important convolution kernels are set to zero should be less than that of less important convolution kernels. Based on this idea, this section presents a selective dropout method based on shallow learning. It uses the classifier obtained through shallow learning to modify the probability that the weight of a node in the hidden layer will be set to zero. The specific algorithmic process is as follows.

Premise: the feature map α_i^l and the weight ω_i^l of a hidden layer l in the deep network are inputs ($i \in [0, n]$), the probability parameter is λ, and the weight ω_i^l after selective dropout is the output.

(1) The first training session is conducted using the standard dropout. All the weights randomly set to 0 in the recording process correspond to the position $D_{\omega_i^l}$ of the node.

(2) The feature map $\{\alpha_i^l\} \subset F$ corresponding to the weight value set to 0 in each hidden layer in the network and the feature map $\{\alpha_i^l\} \subset \text{all} - F$ corresponding to the weight value not set to 0 are used as positive and negative sample values, respectively. Thus, 0 and not-0 are set as positive and negative sample labels, and the map is sent to the support vector machine (SVM) for training.

(3) A second training session is conducted in which each node in each hidden layer uses an SVM to determine whether the weight of that node should be set to zero. Nodes classified as positive samples (whose weight value is set to 0) have an increased probability of being set to 0 by λ times.

(4) Dropout is again performed on each layer; however, the probability of setting each node to 0 will be different. Finally, each layer is assigned the weight ω_i^l after the selective dropout.

From the network training perspective, the dropout approach allows each batch of samples to correspond to different network structures. Different network structures rely on the shared hidden layer weights, which increases the diversity of the network. From the training model perspective, updating the stochastic model each time is macroscopically an average model idea, which enhances model robustness. The random weight approach of dropout no longer relies on the interactions of the hidden layer nodes; consequently, it prevents special cases where certain features are valid only when other specific features exist. This gives dropout the ability to adapt to changes, which greatly reduces network overfitting and enhances its generalizability.

2.2.2. Deep Learning Model Guided by Shallow Learning. The existing deep learning models typically include one or more convolutional layers, a fully connected layer at the top, associated weights, and pooling layers. The specific deep learning model used here is different, and its structure is also different. This paper mainly adjusts and optimizes the three aspects of artificial guidance, network structure, and activation function involved in the establishment and training of typical convolutional neural network models.

(1) Selective Dropout. The specific selective dropout algorithm is described in detail in the previous section. Through the improved SVM-based selective dropout, the network is artificially oriented toward maintaining the original sparsity and finally improves the learning performance, as shown in Figure 3.

The region of interest (RoI) pooling layer in the deep learning model uses a single downsampling layer that allows different-sized feature maps to be normalized to the same size by downsampling. Multiscale is important in some traditional Levy operators. In deep learning, the RoI pooling layer takes a similar form. The convolutional layer is transformed into $4 \times 4 \times n$, $2 \times 2 \times n$, and $l \times l \times n$ fixed-sized feature maps through three different downsamplings. $n = 256$ for the ZF model, and $n = 512$ for the VGG model. Then, the feature maps are connected end to end in a certain order to form a $(16 + 4 + 1) \times n$-dimensional feature vector. Finally, the input of the pooled layer is connected to the fully connected layer. The downsampling process adaptively measures the size and stride by controlling the pooled convolution template. It is scaled to the front convolutional output to ensure that the downsampled subgraph has a fixed size.

(2) Randomly Corrected Linear Unit. The rectified linear unit (ReLU) has better predictive energy in its sparse activation (one-sided suppression) characteristics and a wide excitatory boundary, as shown in the following formula:

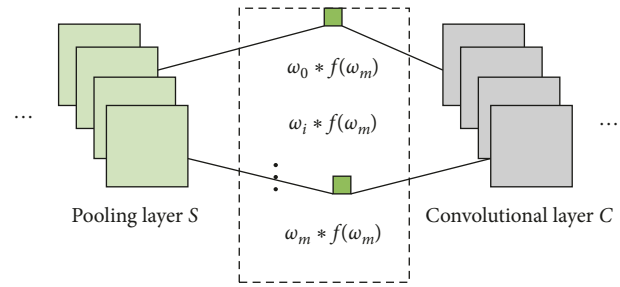

Figure 3: Selective dropout process schematic diagram.

$$\text{ReLU}: f(x) = \begin{cases} x, & x > 0, \\ 0, & x \le 0. \end{cases} \tag{5}$$

ReLU not only guarantees the nonlinear characteristics but also simulates the positive activation response of the nerve well; however, the complete inhibition of negative activation may result in information loss. In 2015, Russakovsky et al. proposed a modification of ReLU with the variant PReLU [58] to improve the negative activation response. In PReLU, the negative activation response is no longer completely suppressed but only considerably reduced. The formula for PReLU is as follows:

$$\text{PReLU}: f(x) = \begin{cases} x, & x > 0, \\ \dfrac{x}{\alpha}, & x \le 0, \end{cases} \quad \alpha \in (1, +\infty), \tag{6}$$

where the parameter $\alpha \in (1, +\infty)$ is obtained during training, and a fixed value is used for testing. The activation characteristics of PReLU reduce overfitting on small-scale data and greatly improve the training effect. This technology has outperformed humans on the ImageNet classification dataset.

To further improve randomness, Russakovsky et al. [58] proposed that randomized ReLU (RReLU) is equivalent to a random version of PReLU. The main improvement is that, during training, the parameters $\alpha \sim U(l, u)$ are randomly generated by uniform distribution, as shown in the following formula:

$$\text{RReLU}: f(x) = \begin{cases} x, & x > 0, \\ \dfrac{x}{\alpha}, & x \le 0. \end{cases} \quad \alpha \sim U(l, u), \tag{7}$$

During testing, the parameter α has a fixed value $(l + u)/2$. After a large number of experimental statistical analyses, the general training values are $\alpha \sim U(3, 8)$, and the test has fixed $\alpha = 5.5$. The random nature of RReLU further enhances the model's resistance to overfitting. This is also the main reason for the adoption of this activation function in this paper because it can better adaptively react to and accomplish pedestrian re-recognition.

3. Pedestrian Re-Recognition Models Based on Sequence Memory Learning

This section proposes a pedestrian re-recognition method based on sequence memory learning. In order to further simulate human memory and prediction mechanism, this method designed a sequence learning model based on memory prediction to transform pedestrian images into image sequences. At the same time, the order of the sequence and the pattern of the memory sequence are learned to achieve fast and accurate memorization and recognition of the pedestrian image.

3.1. Sequence Generation and Feature Extraction. The purpose of sequence generation is to convert a single pedestrian image into a sequence of images for sequence learning. Therefore, the pedestrian image is divided into $M \times N$ nonoverlapping grids, and the subimages in the grid are connected in series to form a sequence of images of length T. In the odd rows, the image sequence runs from left to right; in the even rows, the image sequence runs from right to left. This serial path guarantees the information correlation between subimages and contributes to sequence learning.

Assuming that the pedestrian image is P, after conversion, the image sequence is expressed as $\{(1), \ldots, (T)\}$. Assume that the size of the pedestrian image is 128×48 pixels and that M and N are set to 8 and 3, respectively. Then, the size of each subimage is 16×16 pixels, and the length of the image sequence is 24 $(T = M * N)$.

The CNN and the human visual system are multilayered structures. Therefore, based on the sequence memory learning model, a CNN is used to extract features from each subimage in the image sequence to form a feature sequence of length T. Consider the accuracy and efficiency of pedestrian re-recognition. Based on the sequence memory learning model, the deep learning model proposed in Section 2 of this paper is selected for feature extraction. The size of each subimage in the image sequence is 16×16 pixels. Therefore, there is only one element in each feature map. Then, each feature in the feature sequence is represented as $x(t) \in R^{256 \times 1}$, $t = 1, \ldots, T$.

3.2. Sequence Order Exchange. Sequence order exchange is an important step in the sequence memory learning model based on memory prediction. It simulates human eye tracking to a certain extent. The purpose of sequence order exchange is to rearrange the feature sequences so that the priority based on the sequence memory causes the learning model to memorize important pedestrian features. At the same time, sequence order exchange helps speed up the pedestrian recognition because only part of the feature sequence needs to be input into the sequence-based memory learning model for it to output accurate prediction labels.

In the sequence order exchange process, the feature sequence $\{(1), \ldots, (T)\}$ is converted into the reordered feature sequence $\{\hat{x}(1), \ldots, \hat{x}(T)\}$, where $\hat{x}(t) \in \mathbb{R}^{256 \times 1}$ represents the feature vector extracted by the t-th subimage in the image sequence. In the sequence-based memory

learning model, the order of the sequences is exchanged as follows:

$$\widehat{X} = XW, \tag{8}$$

where $X = [(1), \ldots, (T)]$, $\widehat{X} = [\widehat{x}(1), \ldots, \widehat{x}(T)]$, and $W \in \mathbb{R}^{T\times T}$ is a swap matrix with a constraint $W^T W = I$. Additionally, in each column of the switching matrix W, only one element is set to 1, and the remaining elements are set to 0. Then, if and only if $(i) = 1$, $\widehat{x}(j) = x(i)$.

A schematic diagram of the sequence order exchange process is shown in Figure 4. Based on the order after the exchange, the pedestrian image can be divided into three regions. After the order exchange, the trunk and arm characteristics are more obvious. In pedestrian recognition, the torso and arm characteristics are more important than are those of the head and legs. Therefore, based on the sequence memory learning model, the trunk and arm characteristics are preferentially memorized and recognized. In addition, the sequence-based memory learning model uses random initialization. After model training is complete, although the exchange matrices are not identical, they are highly similar to some extent.

3.3. Memory Storage. Based on the sequence memory learning model, memory storage is realized by using the LSTM model; that is, the sequence pattern of the pedestrian is memorized. The LSTM model takes the reordered feature sequence $\{\widehat{x}(1), \ldots, \widehat{x}(T)\}$ as the input and produces an output sequence $\{(1), \ldots, (T)\}$. In this model, an implicit layer containing 8 memory blocks is configured; each memory block has 16 memory cells that share the same input and output gates.

In the forward process of the LSTM model, all the hidden layer neurons and output layer neurons can be activated at any time. This section describes how all neurons are calculated when $\widehat{x}(t)$ is entered into the LSTM, $t = 1, \ldots$. First, the memory cell input $y_c(t) \in \mathbb{R}^{128\times 1}$, the input gate activation value $y_{in}(t) \in \mathbb{R}^{16\times 1}$, and the output gate activation value $y_{out}(t) \in \mathbb{R}^{16\times 1}$ are calculated:

$$y_c(t) = g\left(\omega_c^f \widehat{x}(t) + \omega_c^\tau y(t-1) + b_c\right), \tag{9}$$

$$y_{in}(t) = f\left(\omega_{in}^f \widehat{x}(t) + \omega_{in}^\tau y(t-1) + b_{in}\right), \tag{10}$$

$$y_{out}(t) = f\left(\omega_{out}^f \widehat{x}(t) + \omega_{out}^\tau y(t-1) + b_{out}\right), \tag{11}$$

where ω_c^f, ω_{in}^f, and ω_{out}^f represent the forward weight of the memory cell, the forward weight of the input gate, and the forward weight of the output gate, respectively; ω_c^τ, ω_{in}^τ, and ω_{out}^τ represent the memory cell return weight, the input gate return weight, and the output gate return weight, respectively; b_c, b_{in}, and b_{out} represent the memory cell offset, the input gate offset, and the output gate offset, respectively; and $y(t-1) \in \mathbb{R}^{128\times 1}$ represents the memory cell output when $\widehat{x}(t-1)$ is input to the long short-term memory model. The memory cell activation function uses $g(x) = (4/(1+e^{-x})) - 2$. The input and output gate activation functions use the sigmoid function $f(x) = 1/(1+e^{-x})$. Then, the memory cell intermediate state $sc(t) \in \mathbb{R}^{128\times 1}$ is calculated as follows:

$$sc(t) = sc(t-1) + y_c(t) \odot \text{expand}\left(y_{in}(t)\right), \tag{12}$$

where e denotes a point multiplication operation between matrices. The expand(x) copies and extends each element of the vector x 8 times, ensuring that the input of each memory cell is correctly multiplied by the corresponding input gate activation value. The intermediate state of all memory cells is initialized to zero. Then, the output $y(t) \in \mathbb{R}^{128\times 1}$ of the memory cell is calculated as follows:

$$y(t) = h(sc(t)) \odot \text{expand}\left(y_{out}(t)\right), \tag{13}$$

where the activation function h is defined as $h(x) = 2/(1+e^{-x}) - 1$. Finally, the output value $z(t) \in [0, 1]$ of the long- and short-term memory model is calculated using the following formula:

$$z(t) = f(\omega y(t) + b), \tag{14}$$

where ω and b represent the output layer weight and offset, respectively. If $z(t)$ is close to 1, the input image is determined to be a pedestrian image; if $z(t)$ is close to 0, the input image is determined to be a background image.

3.4. Joint Learning. This section proposes a joint learning method that enables a sequence-based memory learning model to simultaneously learn the pedestrian sequence order and the memory pedestrian sequence pattern. To achieve this goal, the objective function L is built by the following formula:

$$L = \frac{1}{2}\sum_{t=1}^{T}(z(t) - \widetilde{z})^2 + \frac{\lambda}{4}\left\|W^T W - I\right\|_F^2, \tag{15}$$

where the first term L_l of the objective function L is a loss term, which calculates an error between the output sequence $\{z(1), \ldots, z(T)\}$ and the real tag $\widetilde{z}$, and the second term L_c of the objective function L is a restriction term which limits the form of the switching matrix W and makes the switching matrix W an orthogonal matrix. To minimize the objective function L, this section uses the timed backpropagation algorithm to train LSTM and the backpropagation algorithm to update the switching matrix W.

The LSTM training method using the timed backpropagation algorithm is detailed in Section 2.1. Therefore, this section describes only how to update the switching matrix, W. The gradient of the exchange matrix W with respect to the objective function L is calculated according to the stochastic gradient descent algorithm:

$$\begin{aligned}\frac{\partial L}{\partial W} &= \frac{\partial L_l}{\partial W} + \frac{\partial L_c}{\partial W} \\ &= \frac{\partial L_l}{\partial \widehat{X}}\frac{\partial \widehat{X}}{\partial W} + \frac{\partial L_c}{\partial W^T W}\frac{\partial W^T W}{\partial W} \\ &= X^T \frac{\partial L_l}{\partial \widehat{X}} + \lambda\left(W^T W - I\right)W,\end{aligned} \tag{16}$$

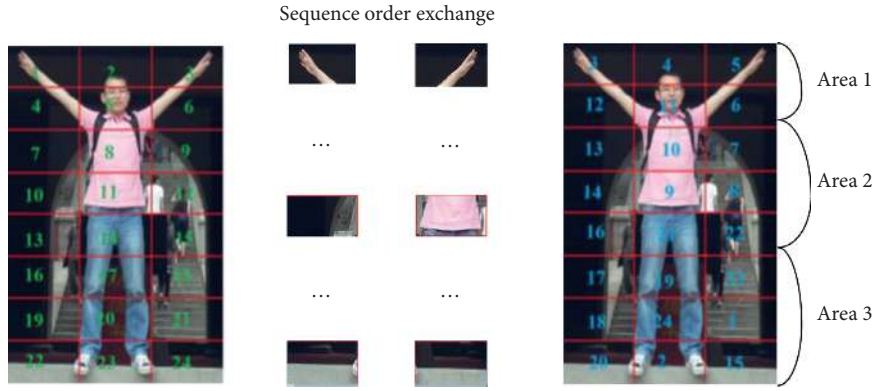

Figure 4: Sequence order exchange schematic diagram.

where a represents the gradient of the reordered feature sequence with respect to the loss term L in the objective function L, $\partial L_l/\partial \widehat{X} = [(\partial L_l/\partial \widehat{x}(1)), \ldots, (\partial L_l/\partial \widehat{x}(T))]$. Therefore, how to calculate $\partial L_l/\partial \widehat{x}(t)$ is the key to updating the exchange matrix, $t = 1, \ldots, T$. For convenience, this paper defines the following two items:

$$l(t) \triangleq \frac{1}{2}(z(t) - \widetilde{z})^2,$$
$$D(t) \triangleq \sum_{i=t}^{T} l(i), \tag{17}$$

where $l(t)$ represents the error between the memory model output $z(t)$ and the real label $\widetilde{z}$ at time t and $D(t)$ represents the cumulative error between the memory model output sequence $\{z(1), \ldots, z(T)\}$ and the real tag $\widetilde{z}$ from time t to time T. Then, $D(t)$ can be re-represented according to the following formula:

$$D(t) = \begin{cases} l(t), & t = T, \\ l(t) + D(t+1), & t = 1, \ldots, T-1. \end{cases} \tag{18}$$

Subsequently, $t = T$, and $\partial L_l/\partial \widehat{x}(t)$ is calculated as follows:

$$\frac{\partial L_l}{\partial \widehat{x}(t)} = \frac{\partial D(t)}{\partial \widehat{x}(t)} = \frac{\partial D(t)}{\partial y(t)} \frac{\partial y(t)}{\partial \widehat{x}(t)} = \frac{\partial l(t)}{\partial y(t)} \frac{\partial y(t)}{\partial \widehat{x}(t)}. \tag{19}$$

Here, $\partial l(t)/\partial y(t)$ and $\partial y(t)/\partial \widehat{x}(t)$ can be directly calculated according to formulas (9)–(14). In addition, when $t = 1, \ldots, T-1$, it is calculated as follows:

$$\begin{aligned} \frac{\partial L_l}{\partial \widehat{x}(t)} &= \frac{\partial l(t)}{\partial \widehat{x}(t)} + \frac{\partial D(t+1)}{\partial \widehat{x}(t)} \\ &= \frac{\partial l(t)}{\partial y(t)} \frac{\partial y(t)}{\partial \widehat{x}(t)} + \frac{\partial D(t+1)}{\partial y(t)} \frac{\partial y(t)}{\partial \widehat{x}(t)} \\ &= \frac{\partial l(t)}{\partial y(t)} \frac{\partial y(t)}{\partial \widehat{x}(t)} + \frac{\partial D(t+1)}{\partial y(t+1)} \frac{\partial y(t+1)}{\partial y(t)} \frac{\partial y(t)}{\partial \widehat{x}(t)}. \end{aligned} \tag{20}$$

After calculating $\partial L_l/\partial \widehat{x}(t+1)$, $\partial D(t+1)/\partial y(t+1)$ and $\partial y(t+1)/\partial y(t)$ can be directly calculated according to formulas (9) to (13). After obtaining $\partial L/\partial W$, the switching matrix W is updated according to the following formula:

$$W = W - \alpha \frac{\partial l}{\partial W}, \tag{21}$$

where α is the learning rate.

However, the above method does not ensure that only one element in each column of the switching matrix W is 1 and that the remaining elements are 0. Therefore, the normalization operation is performed on the switching matrix W each time the switching matrix W is updated, the maximum element in each column of the switching matrix W is set to 1, and the remaining elements are set to 0.

4. Pedestrian Re-Recognition Algorithm Based on the Optimized LSTM Deep Learning-Sequence Memory Model

Based on the above content, this section designs a pedestrian re-recognition algorithm to optimize LSTM deep learning-sequence memory models. First, the LSTM network adaptive training method is used to solve the parameter initialization problem. Then, shallow learning selective dropout technology is used to solve the overfitting problem in the deep learning model training process. Next, a sequence memory learning model is built. Finally, a pedestrian re-recognition algorithm based on the optimized LSTM network deep learning-sequence memory learning model is proposed to realize efficient pedestrian re-recognition in general scenes. The basic steps of the proposed pedestrian re-recognition algorithm are as follows:

(1) Using the optimized deep learning model proposed in Section 2, we extract 256 feature maps from the pedestrian image and use the candidate region extraction algorithm to generate RoIs from the pedestrian image. Because the re-recognition target is a pedestrian, the re-recognition algorithm for pedestrians can be selected as the candidate region extraction algorithm.

(2) A local feature map corresponding to the region of interest is extracted from the entire feature map according to the position of each region of interest. Then, because the sizes of the RoIs are inconsistent, the local feature map is input into the RoI pooling layer.

(3) The fixed-sized 256 feature map is converted into a feature sequence and input into the sequence-based learning model for sequence pattern identification. When the output value is close to 1, the RoI is considered to be a pedestrian; when the output value is close to 0, the RoI is determined to be background.

5. Example Analysis

5.1. Experimental Evaluation Criteria. In this paper, the average accuracy rate and the rank-1 matching rate are used as performance indicators. These two indicators are used to evaluate the effect of the pedestrian re-recognition algorithm. The mAP indicator evaluates the performance of the pedestrian search model in a manner similar to detection: it reflects the accuracy of person detection from the library image. The top-1 match rate treats pedestrian search as a sorting and positioning problem. If the overlap rate of the top-predicted bounding box with the true bounding box is above a threshold of 0.6, it is considered to be successfully matched.

5.2. Example 1. This example uses the INRIA pedestrian dataset [59] and the Caltech pedestrian dataset [60] to verify the proposed pedestrian re-recognition algorithm. Additionally, to prove the feasibility and effectiveness of the proposed method, the method is compared with the main popular re-recognition model.

5.2.1. Experimental Process. Training a sequence-based memory learning model requires sufficient training samples and appropriate training parameters. For positive samples, 1,000 pedestrian images were selected from the INRIA pedestrian dataset [60] and 2,000 pedestrian images were selected from the VIPeR [10] and PRID2011 datasets [61]. Negative samples were randomly taken from images that did not contain pedestrians. All sample images were horizontally flipped and scaled to multiple scales to form a training set. Because the sizes of the samples were not uniform, after extracting the feature map of the sample using the CNN, the RoI pooling layer was used to generate 256 fixed-sized feature maps. During training, the learning rate α was set to 0.01 and the parameter λ of the joint objective function was set to 0.001.

5.2.2. INRIA Pedestrian Dataset. In 2005, Dalal et al. established the INRIA pedestrian dataset [60], which is a classic pedestrian recognition dataset. Moreover, the background of the pedestrian image of the INRIA pedestrian dataset is more complicated. Therefore, many pedestrian recognition methods compare the effects of pedestrian detection using this dataset. The INRIA pedestrian dataset provides training sets and test sets. The training set has 614 images containing pedestrians, which were used to select positive samples, and 1,218 images without pedestrians, which were used to select negative samples. The test set contains 288 different-sized inspection images, for a total of 588 pedestrians. Using the INRIA pedestrian dataset, the proposed method was compared and analyzed with other popular re-recognition methods. The experimental results are shown in Table 2.

For the INRIA pedestrian dataset, Table 2 shows both the average accuracy rate and the rank-1 matching rate. The recognition results of the pedestrian re-recognition algorithm based on the optimized LSTM network deep learning-sequence memory learning model proposed in this paper are superior to those of the existing popular mainstream person re-recognition methods. These results validate the rationality and effectiveness of the proposed method.

5.2.3. Caltech Pedestrian Dataset. Caltech is an image database created by the California Institute of Technology, which contains two datasets, Caltech101 and Caltech256. Similar to the INRIA pedestrian dataset, the Caltech pedestrian dataset also includes training and test sets. The training set has 1000 images with pedestrians, and the test set has 300 pedestrian recognition images. On the Caltech pedestrian dataset, we compare the method proposed in this paper with other popular mainstream re-recognition methods. The experimental results are shown in Table 3.

Table 3 shows that, on the mAP and top-1 metrics, the method in this paper outperforms the other popular mainstream pedestrian re-recognition methods. In terms of mAP, the proposed method scores 0.8% higher than Faster R-CNN. Compared to the LDCF method, the proposed method's mAP increased by 1.4%, while compared to the ACF-Caltech method, its mAP increased by 7%. These results further validate the pedestrian re-recognition method proposed in this paper. From the top-1 results, the pedestrian re-recognition method proposed in this paper is 1.4% higher than Faster R-CNN. Compared to the LDCF method, the proposed method increased the top-1 score by 2.5%, and compared to the ACF-Caltech method, the proposed method increased the top-1 score by 7.4%. These results also fully demonstrate the superiority of the proposed method in terms of the average accuracy and top-1 metrics.

5.3. Example 2

5.3.1. Test Dataset Description. To further test and analyze the performance of the pedestrian re-recognition method proposed in this paper, this example will use two more complex and more challenging public datasets CUHK-SYSU [71] and PRW [48] for experimental verification and analysis.

(1) CUHK-SYSU. This dataset [71] is a large pedestrian search dataset with different shooting scenes, containing 18,184 scene images. These scene images contain a total of 8432 different pedestrians and 96143 marked bounding boxes. Each selected reidentified pedestrian appears in at least two images taken from different perspectives. The images have large differences in perspectives, illumination conditions, resolutions, pedestrian occlusions, and backgrounds. They reflect the diversity of actual pedestrian

Table 2: Comparison of different methods on the INRIA pedestrian dataset.

Method type	INRIA pedestrian dataset	
	Rank-1 (%)	mAP (%)
LatSVM [62]	89.3	75.8
VF [63]	92.6	85.4
Fast R-CNN [64]	95.8	88.9
Faster R-CNN [65]	96.2	88.7
Ours	98.4	90.1

Table 3: Comparison of different methods on the Caltech pedestrian dataset.

Method type	mAP (%)	Top-1 (%)
Ours	89.2	93.9
Faster R-CNN [66]	88.4	92.5
SCF-AlexNet [67]	88.4	92.6
LDCF [68]	87.8	91.4
UDN [69]	84.0	89.7
ACF-Caltech [70]	82.2	86.5

application scenarios. This experiment used the training set and test set partitioning method provided by the dataset itself. The training set contains 11,206 images, of which 5,532 require re-recognizing the pedestrian image. The test set includes a total of 2,900 pedestrians, and the total number of library images is 6,978.

(2) PRW. This dataset [48] was extracted from a 10-hour video taken on a college campus. The dataset includes 11,816 video frames taken by six cameras. The 11,816 frames were manually tagged, and 43,110 bounding boxes were provided. Of these, 34,344 bounding boxes were assigned to 932 different pedestrians. The dataset also provides a standard method for training and test set partitioning. The training set provides 5,134 frames with a total of 482 different pedestrians. The test set contains 2,057 reidentified pedestrians and 6,112 library images.

5.3.2. Experimental Process. In this paper, the Edge Boxes [68] method is used to generate a suggested pedestrian bounding box to provide information for subregion partitioning. The parameter α (which controls the sampling variation, bounding box's translation, and sampling bounding box width ratio) has a step size of 0.65, and the parameter β (which controls the intersection over union (IoU) threshold in the NMS) is set to 0.7. Using these settings, approximately 1,000 suggested bounding boxes were generated, and the first 300 bounding boxes were selected for subsequent subarea partitioning.

The feature extraction process uses the deep learning model presented in Section 2 of this paper. For the input region at each time step, the ROI pooling layer is applied to the conv4-3 convolution map to normalize all the feature maps to the same size of $14 \times 14 \times 1024$. For re-recognition pedestrian images, this paper extracts them as $14 \times 14 \times 1024$ convolution features in the same way. These feature maps are then input into the architecture proposed in Section 3 of this paper. This study uses the Theano deep learning framework to implement the proposed deep learning model.

The experimental platform was equipped with an NVIDIA GeForce GTX GPU, an Intel i7-5790 CPU, and 64 GB of memory. The deep learning framework of this paper requires 45 and 38 hours of training time on the CUHK-SYSU and PRW datasets, respectively. The initial learning rate was set to 0.001, and the attenuation rate of the weight update was set to 0.9. In addition, this paper added data by performing a random two-dimensional geometric transformation. The pedestrian recognition speed of the proposed method was close to real time. For a library image, the recognition model proposed in this paper takes approximately 1 second to output the final recognition result. The main computational cost of pedestrian re-recognition lies in the result ordering of each library image. For CUHK-SYSU, with the number of library images set to 200, the proposed method takes approximately 15 seconds to calculate the cosine similarity between the search results of all library images and the sorted query. For PRW with 6,112 library images, it required approximately 12 minutes to sort the search results for all the library images.

5.3.3. Performance Comparison between This Method and Other Mainstream Methods. Here, the performance of the proposed method is compared with the performance of other mainstream pedestrian recognition methods. These methods include the end-to-end pedestrian recognition framework proposed by Xiao et al. [71] and the method proposed by Zheng et al. [48] and other mainstream methods, as shown in Table 4.

(1) Experimental Results on the CUHK-SYSU Dataset. Table 4 shows the pedestrian recognition performance of CUHK-SYSU with the library image size set to 200, where the CNN represents the detector portion (Faster R-CNN using ResNet-50 was the feature extractor [65]) and IDNet represents the "pedestrian re-recognition" part of the network in the OIM framework [67]. Compared with CNN + IDNet, OIM improves the performance by introducing the joint optimization detection and pedestrian re-recognition components, but it still uses the identification strategy that involves two independent stages of detection and re-recognition in the pedestrian re-recognition process. In contrast, the pedestrian re-recognition proposed in this paper is a memory recognition method that solves the problem of accurate pedestrian positioning through the deep learning model of the LSTM-shallow learning selective dropout. It solves the problem of accurate pedestrian re-recognition by introducing a learning model based on sequence memory. The results shown in Table 4 verify that the pedestrian re-recognition method proposed in this paper outperforms the other mainstream methods on the mAP and top-1 evaluation indexes.

In addition, the mAP experiment was performed by the three methods: OIM, LOMO-XQDA, and the proposed method, under different library image settings—the library image size was variously set to [100, 200, 500, 1500, 3000,

Table 4: CUHK-SYSU pedestrian dataset recognition comparison results for different methods.

Method type	mAP (%)	Top-1 (%)
Ours	80.1	83.8
ACF [69] + LOMO [33] + XQDA [33]	55.5	63.1
ACF + IDNet [48, 71]	56.5	63.0
CNN + LOMO + XQDA [33]	68.9	74.1
CNN + IDNet [48, 71]	68.6	74.8
OIM [71]	75.5	78.7

6000]. The experimental results are shown in Figure 5, which shows that as the number of library images increases, the mAP gradually decreases, but the method proposed in this paper remains superior to the other methods. This method outperforms OIM at each library image size setting by approximately 4%.

(2) Experimental Results on the PRW Dataset. These experiments were also performed on the PRW dataset to compare the performance of this method with that of other mainstream methods. The comparison results are shown in Table 5.

Among the comparison methods, AlexNet [10] was used as the basic network of the R-CNN detector. VGGNet [72] and ResNet [73] have more parameters and are deeper than the AlexNet layer. However, according to the relevant discussion in [48], AlexNet achieves better performance than integrating different recognizers on the deformable part model (DPM) and aggregated channel features (ACFs). The specific results are shown in Table 5. Compared with the results of OIM, our method increases the mAP and top-1 scores by 5.5% and 7.0%, respectively. In addition, all the other mainstream methods use five bounding boxes for each library image. However, our method achieves better performance by retaining only one bounding box for each library image during testing.

The above results also show that the proposed method is better able to identify the generated attention maps on the test set samples from the PRW and CUHK-SYSU datasets. Additionally, the method proposed in this paper effectively decreases the search area to the correct target pedestrian area guided by the original memory containing the reidentified person.

5.4. Example 3

5.4.1. Test Dataset Description. In order to more effectively analyze the performance of the pedestrian reidentification method proposed in this paper, this example will use the current largest pedestrian reidentification dataset Market-1501 [74] for experimental verification and analysis. The dataset is captured by six cameras with different viewing angles (5 of which are 1280×1080 HD cameras and 1720×576 SD cameras). Some examples are shown in Figure 6. The dataset contains 32,668 pedestrian images of 1501 people, each of which appears under at least two cameras and may have multiple images under one camera. The training set and test set of the Market-1501 dataset. The

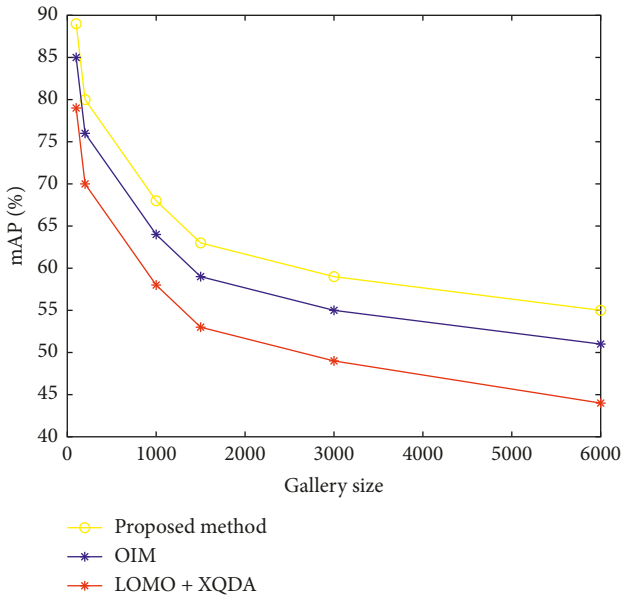

Figure 5: Comparison of mAP performance of different methods under different library image settings.

Table 5: PRW pedestrian dataset recognition comparison results for different methods.

Method type	mAP (%)	Top-1 (%)
Ours	26.8	56.9
DPM-Alex + IDE + CWS [48, 71]	20.3	47.4
ACF-Alex + IDE + CWS [48, 71]	17.8	45.2
OIM [71]	21.3	49.9

Figure 6: Market-1501 dataset example diagram.

training set has 751 people, including 12,936 images; the test set has 750 people, including 19,732 images. The test set contains a gallery set and a query set. All images of the

gallery set are detected by the DPM detector. The pedestrian detection rectangle of all query images in the query set is manually drawn.

5.4.2. Experimental Process. All the images used to train pedestrians in this experiment were subjected to data enhancement such as horizontal flip, blur, random crop, and pan. The division of the training set and the test set is completely in accordance with the standards proposed in [74]. Feature extraction uses the deep learning model presented in Section 2 of this paper. For the input area of each time step, the ROI pooling layer is applied on its conv4-3 convolution map. It normalizes all feature maps to the same size of $14 \times 14 \times 1024$. For querying pedestrian images, this paper extracts their size as $14 \times 14 \times 1024$ convolution features in the same way. These feature maps are then fed back into the architecture proposed in Section 3 of this paper. This paper uses the Theano deep learning framework to implement related models. The basic configuration of this experimental platform is as follows: NVIDIA GeForce GTX GPU, Intel i7-5790 CPU, and memory of 64 GB. The initial learning rate was set to 0.001, and the attenuation rate of the weight update was set to 0.9.

5.4.3. Performance Comparison between This Method and Other Mainstream Methods. In order to better demonstrate the recognition ability of this method, this method is compared with other main popular reidentification methods, including the pedestrian re-recognition algorithm based on the dual-flow network proposed by Suh et al. [75], the artificial semantic analysis-deep learning pedestrian reidentification algorithm proposed by Kalayeh et al. [76], the deep learning pedestrian reidentification method based on the partial convolutional baseline network proposed by Sun et al. [77], and other mainstream methods [78, 79], as shown in Table 6.

It can be seen from Table 6 that the pedestrian re-recognition framework proposed in [78] has the lowest recognition effect whether it is mAP or top-1. This is because the method does not model based on the characteristics of the pedestrian, and the model uses less data for training. It directly leads to poor recognition in the latter. The deep learning-based pedestrian re-recognition algorithm proposed in [79] is superior to the method proposed in [78] for both mAP and top-1. This is mainly due to the better integration of pedestrian image characteristics in the method modeling process proposed in [79]. The accuracy of mAP obtained by the method in [75–77] is above 80%, and the accuracy of top-1 is over 90%. It further shows that the deep learning-based method can better integrate the characteristics of pedestrian images and can train a more suitable pedestrian recognition model.

The pedestrian reidentification method proposed in this paper has achieved the best results in both mAP and top-1 among all pedestrian reidentification algorithms. In terms of mAP, the method in this paper is 1% higher than that in [76, 77] and 2.9%, 8.6%, and 14.5% higher than that in [75, 78, 79], respectively. In terms of top-1, the method in this paper is 2.3% and 3.6% higher than that in [76, 77] and 4.4%, 6.2%, and 16.3% higher than that in [75, 78, 79], respectively. This is mainly because the pedestrian re-recognition algorithm proposed in this paper introduces an optimized deep learning model. It solves the problem of parameter initialization and overfitting of deep learning models. Therefore, it can give full play to the advantages and characteristics of the deep learning model. At the same time, the pedestrian re-recognition algorithm proposed in this paper introduces the sequence memory learning model. It can fully obtain the memory characteristics and sequence characteristics of pedestrian images and further explore the pedestrian image characteristics. Therefore, the pedestrian recognition algorithm proposed in this paper achieves the best recognition performance.

Table 6: Market-150 pedestrian dataset recognition comparison results for different methods.

Method type	mAP (%)	Top-1 (%)
Ours	82.5	96.1
PCB + CNN [77]	81.6	93.8
SPReID [76]	81.3	92.5
Part-aligned + CNN [75]	79.6	91.7
GLAD + deep learning [79]	73.9	89.9
Pose-transfer GAN [78]	58.0	79.8

6. Conclusion

The current pedestrian re-recognition methods based on deep learning have various problems, including a lack of memory and prediction mechanisms, ineffective LSTM parameter initialization, and overfitting problems. Based on the human cognitive process, this paper simulates human memory and prediction mechanisms by designing a memory learning model to transform pedestrian images into image sequences. The model also learns memory sequences and patterns to achieve fast and accurate memory and recognition of pedestrian images. In addition, to perform pedestrian re-recognition efficiently, a pedestrian re-recognition model based on sequence memory learning is designed. This model is then applied to the candidate region-based pedestrian re-recognition framework to identify the patterns of image sequences in each candidate region individually. In addition, this paper proposes a selective dropout method based on shallow learning that uses the classifier obtained by shallow learning to modify the probability that the weight of a node in the hidden layer will be set to 0 in an attempt to eliminate the overfitting phenomenon of the deep learning model. Finally, this paper proposes a greedy layer-by-layer pretraining algorithm to initialize the LSTM. To incorporate all these improvements, this paper proposes a pedestrian re-recognition algorithm based on an optimized LSTM deep learning-sequence memory model.

The basic idea of the pedestrian re-recognition algorithm proposed in this paper is as follows: First, it uses the deep learning model proposed in this paper to extract 256 feature maps from a pedestrian image; then, it uses the candidate

region extraction algorithm to generate some RoIs from the pedestrian image. Next, the pedestrian recognition algorithm is used as the candidate region extraction algorithm. Following this, according to the position of each RoI, a local feature map corresponding to the RoI is extracted from the entire feature map. Finally, these feature maps are transformed into a feature sequence and input into the sequence memory learning model for sequence pattern recognition. When the output value is close to 1, the RoI is considered a pedestrian; when the output value is close to 0, the RoI is considered a background.

In this paper, the proposed algorithm was verified by three experiments that included training and testing on four different pedestrian datasets and then compared with other existing popular re-recognition and detection algorithms. The experimental results showed that the proposed method not only achieves a higher average accuracy than do other mainstream methods but also exceeds top-1 scores clearly than those of the other mainstream algorithms.

Data Availability

The data and code used to support the findings of this study are included within the article.

Conflicts of Interest

The author declares no conflicts of interest.

Acknowledgments

This study was funded by the National Natural Science Foundation of China (No. 61701188), China Postdoctoral Science Foundation (No. 2019M650512), and Natural Science Foundation of Shanxi (No. 201801D221171).

References

[1] J. Ngiam, A. Khosla, M. Kim, N. Juhan, L. Honglak, and Y. N. Andrew, "Multimodal deep learning," in *Proceedings of the 28th International Conference on Machine learning (ICML-11)*, pp. 689–696, Bellevue, WA, USA, June-July 2011.

[2] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*, MIT Press, Cambridge, UK, 2016.

[3] J. P. Gee, "Learning by design: good video games as learning machines," *E-learning and Digital Media*, vol. 2, no. 1, pp. 5–16, 2005.

[4] N. Srivastava, E. Mansimov, and R. Salakhudinov, "Unsupervised learning of video representations using LSTMs," in *Proceedings of the International Conference on Machine Learning*, pp. 843–852, Lille, France, July 2015.

[5] Y. Wu, Y. Lin, X. Dong, Y. Yan, W. Ouyang, and Y. Yang, "Exploit the unknown gradually: one-shot video-based person re-recognition by stepwise learning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5177–5186, Salt Lake City, UT, USA, June 2018.

[6] Z. Qiu, T. Yao, and T. Mei, "Learning deep spatio-temporal dependence for semantic video segmentation," *IEEE Transactions on Multimedia*, vol. 20, no. 4, pp. 939–949, 2017.

[7] X. Liu, W. Liu, T. Mei, and H. Ma, "A deep learning-based approach to progressive vehicle Re-identification for urban surveillance," in *Computer Vision-ECCV 2016*, vol. 9906, pp. 869–884, Springer, Cham, Switzerland, 2016.

[8] X. Wang, "Intelligent multi-camera video surveillance: a review," *Pattern Recognition Letters*, vol. 34, no. 1, pp. 3–19, 2013.

[9] N. Gheissari, T. B. Sebastian, and R. Hartley, "Person reidentification using spatiotemporal appearance," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 1528–1535, New York, NY, USA, June 2006.

[10] D. Gray and H. Tao, "Viewpoint invariant pedestrian recognition with an ensemble of localized features," in *Lecture Notes in Computer Science*, vol. 5302, pp. 262–275, Springer, Berlin, Germany, 2008.

[11] E. Ahmed, M. Jones, and T. K. Marks, "An improved deep learning architecture for person re-recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3908–3916, Boston, MA, USA, June 2015.

[12] L. Wu, Y. Wang, J. Gao, and X. Li, "Deep adaptive feature embedding with local sample distributions for person reidentification," *Pattern Recognition*, vol. 73, pp. 275–288, 2018.

[13] R. Zhao, W. Oyang, and X. Wang, "Person re-recognition by saliency learning," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 2, pp. 356–370, 2016.

[14] H. Liu, J. Feng, M. Qi, J. Jiang, and S. Yan, "End-to-end comparative attention networks for person re-identification," *IEEE Transactions on Image Processing*, vol. 26, no. 7, pp. 3492–3506, 2017.

[15] G. Lisanti, I. Masi, and A. D. Bagdanov, "Person re-recognition by iterative re-weighted sparse ranking," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 8, pp. 1629–1642, 2014.

[16] R. Vezzani, D. Baltieri, and R. Cucchiara, "People reidentification in surveillance and forensics: a survey," *ACM Computing Surveys*, vol. 46, no. 2, pp. 29–37, 2013.

[17] A. Bedagkar-Gala and S. K. Shah, "A survey of approaches and trends in person re-identification," *Image and Vision Computing*, vol. 32, no. 4, pp. 270–286, 2014.

[18] R. Zhao, W. Ouyang, and X. Wang, "Learning mid-level filters for person re-recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 144–151, Columbus, OH, USA, June 2014.

[19] W. Chen, X. Chen, and J. Zhang, "Beyond triplet loss: a deep quadruplet network for person re-recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 403–412, Honolulu, HI, USA, July 2017.

[20] M. E. Monroe, N. Tolić, N. Jaitly, J. L. Shaw, J. N. Adkins, and R. D. Smith, "VIPER: an advanced software package to support high-throughput LC-MS peptide identification," *Bioinformatics*, vol. 23, no. 15, pp. 2021–2023, 2007.

[21] C. C. Loy, T. Xiang, and S. Gong, "Multi-camera activity correlation analysis," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1988–1995, Miami, FL, USA, June 2009.

[22] S. Zhang, E. Staudt, and T. Faltemier, "A camera network tracking (CamNeT) dataset and performance baseline," in *Proceedings of the IEEE Winter Conference on Applications of Computer Vision*, pp. 365–372, Waikoloa, HI, USA, January 2015.

[23] M. Gou, S. Karanam, W. Liu, C. Octavia, and J. R. Richard, "DukeMTMC4ReID: a large-scale multi-camera person Re-

recognition dataset," in *Proceedings of the IEEE Conference on CVPR Workshops*, pp. 10–19, Honolulu, HI, USA, July 2017.

[24] H. Zhao, M. Tian, S. Sun et al., "Spindle net: person re-recognition with human body region guided feature decomposition and fusion," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1077–1085, Honolulu, HI, USA, July 2017.

[25] L. Wei, S. Zhang, and W. Gao, "Person transfer GAN to bridge domain gap for person re-identification," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 79–88, Salt Lake City, UT, USA, June 2018.

[26] T. Ahonen, A. Hadid, and M. Pietikainen, "Face description with local binary patterns: application to face recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, no. 12, pp. 2037–2041, 2006.

[27] O. Oreifej, R. Mehran, and M. Shah, "Human identity recognition in aerial images," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 709–716, San Francisco, CA, USA, June 2010.

[28] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.

[29] S. Liao and S. Z. Li, "Efficient PSD constrained asymmetric metric learning for person re-recognition," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 3685–3693, Las Condes, Chile, December 2015.

[30] M. Koestinger, M. Hirzer, P. Wohlhart, M. R. Peter, and B. Horst, "Large scale metric learning from equivalence constraints," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2288–2295, Providence, RI, USA, June 2012.

[31] B. Nguyen and B. De Baets, "Kernel distance metric learning using pairwise constraints for person Re-identification," *IEEE Transactions on Image Processing*, vol. 28, no. 2, pp. 589–600, 2019.

[32] S. Bak and P. Carr, "Deep deformable patch metric learning for person re-identification," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 28, no. 10, pp. 2690–2702, 2018.

[33] S. Liao, Y. Hu, and X. Zhu, "Person re-recognition by local maximal occurrence representation and metric learning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2197–2206, Boston, MA, USA, June 2015.

[34] D. Yi, Z. Lei, S. Liao, and S. Z. Li, "Deep metric learning for person re-identification," in *Proceedings of the IEEE Conference on Pattern Recognition*, pp. 34–39, Stockholm, Sweden, June 2014.

[35] W. Li, R. Zhao, T. Xiao, and X. Wang, "DeepReID: deep filter pairing neural network for person re-recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 152–159, Columbus, OH, USA, June 2014.

[36] S.-Z. Chen, C.-C. Guo, and J.-H. Lai, "Deep ranking for person re-identification via joint representation learning," *IEEE Transactions on Image Processing*, vol. 25, no. 5, pp. 2353–2367, 2016.

[37] H. Shi, Y. Yang, X. Zhu et al., "Embedding deep metric for person re-identification: a study against large variations," in *Computer Vision-ECCV 2016*, vol. 9905, pp. 732–748, Springer, Cham, Switzerland, 2016.

[38] D. Li, X. Chen, Z. Zhang, and K. Huang, "Learning deep context-aware features over body and latent parts for person re-recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 384–393, Honolulu, HI, USA, June-July 2017.

[39] J. Lin, L. Ren, J. Lu, and J. Feng, "Consistent-aware deep learning for person re-recognition in a camera network," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5771–5780, Honolulu, HI, USA, June-July 2017.

[40] L. Wu, C. Chunhua Shen, and A. v. d. Hengel, "Deep linear discriminant analysis on Fisher networks: a hybrid architecture for person re-identification," *Pattern Recognition*, vol. 65, pp. 238–250, 2017.

[41] J. You, A. Wu, X. Li, and W. S. Zheng, "Top-push video-based person re-recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1345–1353, Las Vegas, NV, USA, June-July 2016.

[42] S. Paisitkriangkrai, C. Shen, and A. Van Den Hengel, "Learning to rank in person re-recognition with metric ensembles," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1846–1855, Boston, MA, USA, June 2015.

[43] L. Wu, C. Shen, and A. Hengel, "Personnet: person re-recognition with deep convolutional neural networks," 2016, http://arxiv.org/abs/1601.07255.

[44] R. R. Varior, M. Haloi, and G. Wang, "Gated siamese convolutional neural network architecture for human re-identification," in *Computer Vision-ECCV 2016*, vol. 9912, pp. 791–808, Springer, Cham, Switzerland, 2016.

[45] C. Su, S. Zhang, J. Xing, W. Gao, and Q. Tian, "Deep attributes driven multi-camera person re-identification," in *Computer Vision-ECCV 2016*, vol. 9906, pp. 475–491, Springer, Cham, Switzerland, 2016.

[46] S. Ding, L. Lin, G. Wang, and H. Chao, "Deep feature learning with relative distance comparison for person re-identification," in *Pattern Recognition*, vol. 48, pp. 2993–3003, no. 10, Springer, Cham, Switzerland, 2015.

[47] L. Zheng, Z. Bie, Y. Sun et al., "MARS: a video benchmark for large-scale person re-identification," in *Computer Vision-ECCV 2016*, vol. 9910, pp. 868–884, Springer, Cham, Switzerland, 2016.

[48] L. Zheng, H. Zhang, and S. Sun, "Person re-recognition in the wild," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1367–1376, Honolulu, HI, USA, July 2017.

[49] Z. Zhong, L. Zheng, and G. Kang, "Random erasing data augmentation," 2017, http://arxiv.org/abs/1708.04896.

[50] Z. Zheng, L. Zheng, and Y. Yang, "Pedestrian alignment network for large-scale person re-recognition," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 10, pp. 10–29, 2018.

[51] L. Maaten and G. Hinton, "Visualizing data using t-SNE," *Journal of Machine Learning Research*, vol. 9, pp. 2579–2605, 2008.

[52] G. Antipov, S. A. Berrani, and N. Ruchaud, "Learned vs. hand-crafted features for pedestrian gender recognition," in *Proceedings of the 23rd ACM International Conference on Multimedia*, pp. 1263–1266, Brisbane, Australia, October 2015.

[53] E. Ustinova, Y. Ganin, and V. Lempitsky, "Multi-region bilinear convolutional neural networks for person re-recognition," in *Proceedings of the 14th IEEE International Conference on Advanced Video and Signal Based Surveillance*, pp. 1–6, Lecce, Italy, August-September 2017.

[54] S. Bai, X. Bai, and Q. Tian, "Scalable person re-recognition on supervised smoothed manifold," in *Proceedings of the IEEE*

Conference on Computer Vision and Pattern Recognition, pp. 25307–32539, Honolulu, HI, USA, July 2017.

[55] G. Song, B. Leng, Y. Liu, C. Hetang, and S. Cai, "Region-based quality estimation network for large-scale person re-recognition," *Thirty-Second AAAI Conference on Artificial Intelligence*, pp. 7347–7354, New Orleans, LA, USA, February 2018.

[56] W. Zaremba and I. Sutskever, "Learning to execute," 2014, http://arxiv.org/abs/1410.4615.

[57] K. Xu, X. Shen, T. Yao, X. Tian, and T. Mei, "Greedy layer-wise training of long short term memory networks," in *Proceedings of the IEEE International Conference on Multimedia & Expo Workshops (ICMEW)*, pp. 1–6, San Diego, CA, USA, July 2018.

[58] O. Russakovsky, J. Deng, H. Su et al., "Imagenet large scale visual recognition challenge," *International Journal of Computer Vision*, vol. 115, no. 3, pp. 211–252, 2015.

[59] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 886–893, San Diego, CA, USA, June 2005.

[60] G. Griffin, A. Holub, and P. Perona, *Caltech-256 Object Category Dataset*, California Institute of Technology, Pasadena, CA, USA, 2007.

[61] B. Alipanahi, M. Biggs, and A. Ghodsi, "Distance metric learning vs. Fisher discriminant analysis," in *Proceedings of the 23rd National Conference on Artificial Intelligence*, pp. 598–603, Chicago, IL, USA, July 2008.

[62] P. Felzenszwalb, D. McAllester, and D. Ramanan, "A discriminatively trained, multiscale, deformable part model," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1–58, Anchorage, AK, USA, June 2008.

[63] S. Schulter, P. Wohlhart, and C. Leistner, "Alternating decision forests," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 508–515, Portland, OR, USA, June 2013.

[64] R. Girshick, "Fast R-CNN," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 1440–1448, Las Condes, Chile, December 2015.

[65] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: towards real-time object detection with region proposal networks," in *Advances in Neural Information Processing Systems*, pp. 91–99, 2015.

[66] Z. Duan, J. Lan, Y. Xu, B. Ni, L. Zhuang, and X. Yang, "Pedestrian detection via Bi-directional multi-scale Analysis," in *Proceedings of the 2017 ACM on Multimedia Conference*, pp. 1023–1031, Mountain View, CA, USA, October 2017.

[67] J. Hosang, M. Omran, R. Benenson, and B. Schiele, "Taking a deeper look at pedestrians," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4073–4082, Boston, MA, USA, June 2015.

[68] W. Nam, P. Dollár, and J. H. Han, "Local decorrelation for improved pedestrian detection," *Advances in Neural Information Processing Systems*, pp. 424–432, 2014.

[69] W. Ouyang and X. Wang, "Joint deep learning for pedestrian detection," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 2056–2063, Sydney, Australia, December 2013.

[70] P. Dollar, R. Appel, S. Belongie, and P. Perona, "Fast feature pyramids for object detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 36, no. 8, pp. 1532–1545, 2014.

[71] T. Xiao, S. Li, and B. Wang, "Joint detection and identification feature learning for person search," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3415–3424, Honolulu, HI, USA, July 2017.

[72] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014, http://arxiv.org/abs/1409.1556.

[73] K. He, X. Zhang, and S. Ren, "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 770–778, Las Vegas, NV, USA, June-July 2016.

[74] L. Zheng, L. Shen, L. Tian, S. Wang, J. Wang, and Q. Tian, "Scalable person re-identification: a benchmark," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 1116–1124, Las Condes, Chile, December 2015.

[75] Y. Suh, J. Wang, S. Tang, T. Mei, and K. Lee, "Part-aligned bilinear representations for person re-identification," in *Proceedings of the European Conference on Computer Vision*, pp. 402–409, Munich, Germany, September 2018.

[76] M. M. Kalayeh, E. Basaran, M. Gökmen, M. E. Kamasak, and M. Shah, "Human semantic parsing for person re-identification," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1062–1071, Salt Lake City, UT, USA, June 2018.

[77] 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 *Proceedings of the European Conference on Computer Vision*, pp. 480–496, Munich, Germany, September 2018.

[78] J. Liu, B. Ni, Y. Yan, P. Zhou, S. Cheng, and J. Hu, "Pose transferrable person re-identification," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4099–4108, Salt Lake City, UT, USA, June 2018.

[79] L. Wei, S. Zhang, H. Yao, W. Gao, and Q. Tian, "GLAD: global-local-alignment descriptor for pedestrian retrieval," in *Proceedings of the 25th ACM International Conference on Multimedia*, pp. 420–428, Mountain View, CA, USA, October 2017.