Pedestrian Recognition System of Wild Scene Video Streaming Based on Convolutional Neural Network

Keer Song

The University of Georgia, USA

Abstract. Object detection aims at detecting the position of important objects and identifying them in still images or video frames. Pedestrians are the objects with most attention in images and videos. We study the detection and recognition technology for pedestrian objects of the wild scene video streams. Based on convolutional neural network, we design an online pedestrian recognition system. As non-rigid objects, people may exist in different poses in the image. Moreover, with the interference of complex background, lighting changes and occlusions, it will be very difficult to detect and recognize human objects. Following pedestrian detection in the images, we can conduct research on pedestrian tracking, pedestrian recognition and pedestrian behavior analysis. In addition, pedestrian detection technology also has great commercial value in video monitoring systems, intelligent traffic systems and other fields. In this paper, we propose a network model based on multiple loss functions, which combines the cross-entropy loss function with the improved monitoring signal center loss function to update the model parameters. Finally, the extracted features are more obvious, making the features of the same individual more compact. Compared to networks trained with only cross-entropy loss functions, our method significantly improves recognition accuracy and implements a pedestrian attribute recognition application based on Web technology. And compared with other similar applications, the recognition system we designed is cross-platform and lightweight. It mainly includes image acquisition, pedestrian attribute recognition and data storage modules, which is of great significance for the application of pedestrian recognition technology in the wild scenes.

Keywords: Pedestrian Detection, Pedestrian Recognition, Deep Learning, Convolutional Neural Network.

1. Introduction

The pedestrian recognition technology captures the object face image through the image acquisition device and uses the corresponding algorithm to recognize the object identity. Compared with other biometric technologies, face recognition is more accepted by people because of its characteristics such as non-contact, non-compulsory, simple operation, intuitive results and good concealment. In the past few years, pedestrian recognition technology has made great progress and a large number of pedestrian attribute recognition algorithms and products have appeared. The FRVT 2006 (Face Recognition Vendor Test 2006) and MBGC (Multiple Biometric Grand Challenge) organized by NIST in the USA and the tens of millions level large-scale pedestrian attribute recognition test organized by the First Research Institute of the Ministry of Public Security in China show that the accuracy of pedestrian recognition technology has been greatly improved and can meet the requirements of some practical applications. Pedestrian recognition technology has a wide range of application scenarios. In addition to traditional access control systems, it also can be used in combination with video monitoring system, mobile phone unlocking and computer login identification. In particular, public security technology requirements are currently most urgent. Pedestrian recognition technology can be used to assist public security authorities in solving cases. The pedestrian attribute recognition system, network and face database can be used to determine the identity of criminal suspects, locate them and implement criminal arrests. And it is also used in airports, railway stations and other places. In terms of national public security, it can assist in the security management of long-distance passenger transport and important regional entrances and exits. Therefore, if the field of pedestrian attribute recognition can break through its technical bottleneck, its huge potential market value will be fully seen.

Traditional identity recognition methods are mainly realized through personal recognition objects and personal recognition knowledge. Common personal identifiers include keys, certificates, etc., and knowledge of personal identifiers includes user names, passwords, etc. As we all know, keys, certificates and other signs are easy to be lost or forged, and the knowledge of identifiers is easy to be forgotten or misremembered. A more serious problem is that traditional recognition systems cannot distinguish between the real owner of the identifier and the forger who obtains it. Therefore, traditional identification methods can no longer meet the needs of today's social development. Biometrics technology is a high-tech identity recognition method that includes two parts: physical characteristics and behavioral characteristics. Physical characteristics include facial features, fingerprints, hand shape, genes, body odor, etc. By simulating human nervous system, high-level features with more discriminating ability can be obtained through multi-level nonlinear processing of the original data. Due to the large amount of training data, deep learning methods are also more robust and reduce the interference caused by intra-class changes to a certain extent. However, there is still considerable improvement for safer and more accurate applications in real world.

2. Pedestrian Recognition Based on Deep Convolutional Neural Networks

2.1. Basic Algorithm Framework

In the whole process of pedestrian recognition, the most important thing is the extraction of face features. Different methods can extract features differently. The CNN (Convolutional Neural Network) model is mainly based on supervised learning, obtain face feature vectors with inter-class differences, divide different face images into different categories, train the network for multi-classification task using the cross entropy loss function, eventually remove the Softmax layer and use full connection. The output of the previous layer of SoftMax is used as the facial feature representation. The basic framework of the algorithm proposed in this chapter is shown in Fig. 2 below. The CNN model based on the upper half of the figure only uses Cross-entropy as the loss function. The basic structure of the lower half of the model is the same as that of the upper half, but the Central loss function is added to the fully connected layer of *fc* in Fig. 1. Combining these two loss functions to train the network can further optimize the feature vector obtained from the *fc* layer.

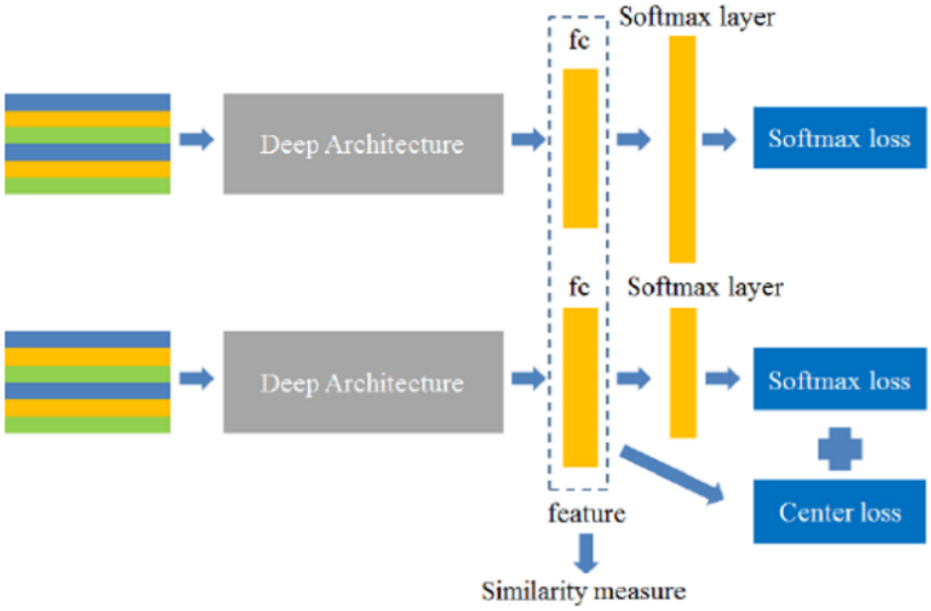


Figure 1. Two ways for Extracting Feature Vectors from Deep Models

Both methods automatically extract features from the original image data by deep convolutional networks. And cosine similarity and Euclidean distance are commonly used to measure the similarity

between different faces in pedestrian attribute recognition. For face retrieval tasks, the similarity between the query image and the images in the database is directly ranked and output.

2.2. Basic CNN Recognition Model

2.2.1. Basic Model Structure

The convolutional layer uses successive small-sized kernels instead of a single large-sized kernel, all of which are 3×3 in size, with stride set to 1, and use zero-padding to keep the output data the same size as the input. Compared to a single large-sized kernel, successive small-sized kernels can also achieve the same extraction effect, and due to the increased number of layers, the activation function further increase the non-linearity and enhance the model expression. In deep convolutional networks, the main consumption of memory is the storage of the output feature maps after each convolution and the parameters of the connections between the layers. In the initial layer, the size of the input image is large, since a small kernel is used and the size of the output data remains the same, the more kernels used, the more feature maps corresponding to the output and the more storage space is required. By reducing the number of partial kernels and the pooling operation above, more kernels can be used to extract and combine high-level features after the size of the input data has been significantly reduced in later layers, which can effectively reduce the space occupied by the feature map and maintain the feature extraction capability of the model. By reducing the number of neural units in the fully connected layers, the parameter occupancy can be effectively reduced. The idea we proposed in this paper reduces the number of neural units in the two fully connected layers of the original VGG model from 4096 to 1024 and 256 respectively, with the Fc7 layer as the final feature representation layer used for faces. It has been shown in some papers that using a smaller number of neural units can still obtain excellent discriminability, and we will also choose an appropriate dimension of the face representation features in later experiments.

During the training process, a Dropout layer is added after the first two fully connected layers. The function of the Dropout layer is to randomly discard some neurons of this layer temporarily according to a certain rate, so that when the network is trained using the batch gradient descent method, the network structure is different, and the weights of the discarded neurons remain the same when updating the weights.

The Dropout layer randomly generates different structure models in each round, and the final trained model results can be seen as a fusion of these models to a certain extent. It makes the neurons within the layer need to be combined with different other neurons each time, weakens the joint adaptation between neurons in the same layer, effectively suppresses the overfitting problem during training, and keeps the number of neuron parameters to be trained not increase, so the model can obtain a better generalization ability.

2.2.2. Similarity Metric

The facial feature vector can be extracted from the trained CNN model. We identify person by comparing the similarity between feature vectors. And the methods proposed in this paper all choose cosine similarity to compare the similarity between faces. The cosine similarity calculates the cosine value of the angle between two vectors, and the formula is defined as below:

$$similarity = \cos(\theta) = \frac{A \cdot B}{\|A\|\|B\|} \tag{1}$$

The value of cosine similarity ranges from -1 to 1. When the two feature vectors are in the same direction, the angle is smaller and the similarity tends more to 1, indicating that the two features are more similar and more likely to be the same person. On the contrary, the larger the angle, the smaller the similarity, indicating that the two features belong to different people.

2.2.3. Related Improvement Methods of CNN

Network-in-Network (NIN) Structure: The filter in CNN is a generalized linear regression model. The abstraction level of this model is slightly lower, but the model can obtain a variant with invariant features through abstraction. Lin et al.[1] proposed a model which uses a micro neural network instead of a fully connected layer in CNN, and this approach enhances the performance of the CNN model. The so-called micro-neural network model is a model using multiple neural networks. While the convolution layer of the traditional CNN model is linear, the NIN uses a non-linear convolution layer, and it also uses a multi-layer neural network to replace the generalized linear regression model, the feature plane is obtained through a micro-neural network. Similar to the weight sharing in the convolutional layer, multi-layer neural networks also have local perception and sharing properties for the same feature plane, and the same feature plane has the same multilayer neural network model. The multi-layer neural network model is trained by the feed-forward neural network algorithm, which can achieve compatibility with the CNN. At the same time, the multi-layer neural network is also a deep learning model. And it can handle more complex problems and identify more abstract features. The current CNN model has too many parameters of the fully connected layer and tends to overfitting, so it relies much on regularization techniques. The NIN model uses global average pooling to change the original fully connected layer, greatly reducing the number of parameters in the model by globally pooling the last convolutional layer in the multi-layer neural network and taking the average value for each feature plane then connecting them and finally sending them to the classification layer. The global average pooling can be seen as a regularization parameter to avoid overfitting. In addition, this method is obtained by summing the spatial information, so it has a strong robustness. Lin et.al [2] have applied this algorithm on the dataset SVHN and MNIST to show the effectiveness of this algorithm. Xu et al. proposed the ML-DNN network model based on the NIN model, which has been experimentally confirmed that it has a better performance [3].

Spatial Transformer Network (STN): Although CNN model is already a very powerful classification model, it does not mean that it is not affected by the diversity of data. Jaderberg et al. [4] used spatial transformer network to solve the problem of data diversity in network space. This model consists of three parts: the localization network model, grid generator and sampler which can be applied to the convolution layer, the input layer or other layers without changing the internal structure of the original CNN model. The spatial transformer and alignment of the STN model are adaptive, making the CNN model invariant to other transformations such as translation, rotation, and scaling. And STN hardly affects the training speed of the CNN model.

Deconvolution: The deconvolution network model proposed by Zeiler et al. [5] has similar ideas to the CNN, there is only somewhat different with the operation process. The operation of CNN is bottom-up, with the input signal going through a series of convolutional operations, non-linear transformations and downsampling in turn. The idea of the deconvolutional neural network is just the opposite. The information of each layer is transmitted from the upper layer to the lower layer, and the input signal is reconstructed through the convolution operation of the filter and the feature plane learned by itself. Zeiler [6] used the deconvolution model to visualize the features of each layer in a CNN model through learning, which is very helpful for analyzing the performance of the network structure model. A deconvolutional network model is also equivalent to a convolutional model, going through the same two steps of convolution and pooling, except that the process of deconvolution is reversed and each layer of the model has an additional layer of deconvolution. After the convolution process, activation function and pooling operation, while using the output features as the input of the lower layer, also perform an inverse process similar to pooling for the corresponding deconvolution layer, ensure that the output of the nonlinear function and deconvolution operation is non-negative, and finally form a new reconstructed feature. The features learned by CNN can be visualized through the deconvolution process. Zeiler [7] also concluded that the learning features of CNN are invariant to translation and scaling, but excluding the case where the model has very strong symmetry, deconvolution is not invariant to rotation. Zhao et al. [8] propose an automatic coding structure with stacked variable positions, which consists of convolutional structure and deconvolutional structure,

its input is encoded using a convolutional network model and the deconvolution is used to reconstruct the network model. The feature of this model is that there is a "content-position" automatic coding machine at each stage, which consists of a convolutional layer and a max-pooling layer, so that the two variables have an intersection through the max-pooling layer. The variable output by the largest layer is used as the input of its next layer, and the position information of the largest layer is to be sent to the deconvolution part. The stacked variable position auto-encoding mechanism includes three loss functions: reconstruction loss, discriminant loss and intermediate reconstruction loss. This method achieves high accuracy in a variety of supervised and semi-supervised models, and it is particularly suitable for the cases where there are a large number of unlabelled categories and a small number of labelled categories in the data, and this model is also suitable for video applications.

2.3. CNN Model Based on Multiple Loss Functions

2.3.1. Cross-entropy Loss Function

The cross-entropy loss function for classification task is defined as Eq. (2):

$$L_s = -\sum_{i=1}^{m} \log\left(e^{W_{yi}^T x_i + b_{yi}} \Big/ \sum_{j=1}^{n} e^{W_j^T x_i + b_j} \right) \tag{2}$$

Where x is the output value of the second to last fully connected layer, W is the weight of the last fully connected layer, b is the corresponding bias value, m is the batch size, and n is the number of categories. Each output vector is nonlinearly scaled by $\exp(x)$ when training for classification, making it easier to converge on the multi-classification problem, while after the categorie features are almost separated, the loss value is lower and it is no longer tends to further separate the features of different categories. But merely distinguishing different categories in the training set is not enough to meet the needs to compare feature vector similarity in the pedestrian attribute recognition task.

Fig. 2 below show the 2D feature mapping obtained from training a classification CNN model using the cross-entropy loss function on the MNIST dataset, with different colours representing different classes. As it can be seen that the model effectively distinguishes the categories, but these feature vectors are not well measured by cosine similarity or Euclidean distance.

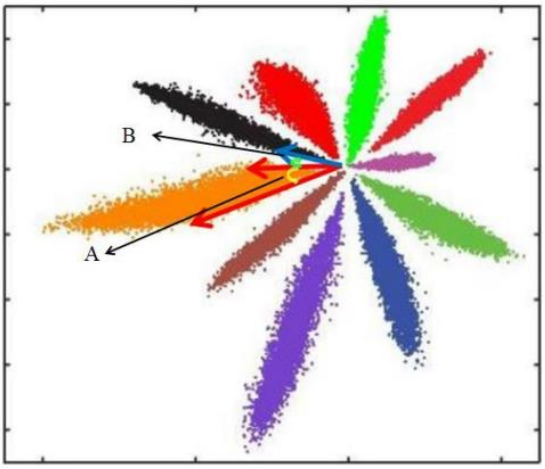


Figure 2. 2D Feature Map for 10-class Classification on MNIST using Cross-entropy Loss Function

Taking the cosine similarity as an example, in Fig. 2-4 $\angle$A represents the angle between two feature vectors belonging to the same class, while $\angle$B represents the angle between two different classes. There may be cases that $\angle$A is larger than $\angle$B. At this time, the similarity between

different categories is instead greater. When there are more categories, this case is more likely to occur in the problem of pedestrian attribute recognition, which affects the recognition accuracy.

2.3.2. Improved Center Loss Function

Due to the shortcomings of the above Cross-entropy loss function, we introduce the Center Loss function in this paper to make the intra-class distance more tight and the inter-class more separated. The idea is to add an auxiliary loss function, maintain a class centre variable for each class, and add the distance between the training sample's feature vector and the corresponding class centre to the loss function, so that the sample can move closer to its class centre position when solving optimization. The loss function is defined as follows:

$$L_C = \frac{1}{2}\sum_{i=1}^{m}\left\|x_i - c_{yi}\right\|^2 \tag{3}$$

Where c_{y_i} is the maintained corresponding class center vector, and its dimension is the same as the feature vector output by the second to last fully connected layer. The distance between the training sample output feature vector x_i and the class center is calculated using Euclidean distance. Combine this loss function with the Cross-entropy loss function can get the overall loss function:

$$L = L_s + \lambda L_c = -\sum_{i=1}^{m}\log\frac{e^{W_{yi}^T x_i + b_{yi}}}{\sum_{j=1}^{n} e^{W_j^T x_i + b_j}} + \frac{1}{2}\sum_{i=1}^{m}\left\|x_i - c_{yi}\right\|^2 \tag{4}$$

The two loss functions are balanced by one parameter, and since the Center loss mainly has an effect on supervised training, λ is usually less than 1. Each round of mini-batch will update each class centre, the formula is as follows:

$$c_j^{i+1} = c_j^t - \alpha\Delta c_j^t = c_j^t - \alpha\left(\sum_{i=1}^{m}\delta\left(y_i = i\right)\cdot\left(c_j - x_i\right)\right)\Big/\left(1+\sum_{i=1}^{m}\delta\left(y_i = i\right)\right) \tag{5}$$

Where $\delta(y_i = j)$ indicates that the value is 1 when the condition in the parentheses is true, otherwise it is 0. Therefore, when updating the class center, all samples' average distance of this class contained in the mini-batch is updated, and a coefficient α is added as the learning rate of the class center to reduce the effect of some wrongly labeled samples. Other parameters are updated through backpropagation under the combined effect of the two loss functions. The parameters of the last fully connected layer W and the parameters θ before the second to last layer are updated as follows:

$$W^{t+1} = W^t - \beta\cdot\frac{\partial L^t}{\partial W^t} = W^t - \beta\cdot\frac{\partial L_S^t}{\partial W^t} \tag{6}$$

Based on the original Center loss, this paper improves on the cross-age recognition problem. As the number of images contained in different ages of a person is different, it may make the class centre more biased towards the age with more samples. Therefore, when training the model for different images of a person at different ages, we give different weight values for updating the corresponding class centres, and the formula for the change of the class center becomes:

$$\Delta \bar{c}_j^t = \left(\sum_{i=1}^{m} \delta(y_i = i) \cdot (c_j - x_i) \right) \Big/ \left(1 + \sum_{i=1}^{m} \delta(y_i = i) \right) \tag{7}$$

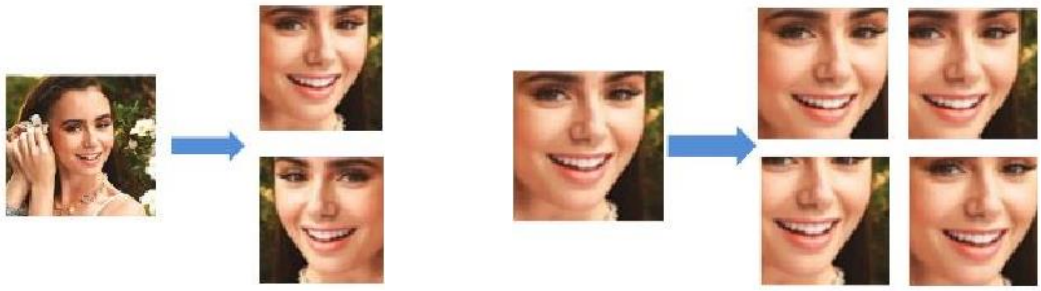
The images of a person's middle age range correspond to a larger weight value, and the weight value gradually decreases towards the edge age range. Specifically, the weight of each person's middle age range is set to 1, and the corresponding weight is multiplied by a coefficient of 0.9 for each year's change to the edge. As the improved Center loss also only acts as a supervision signal when training and does not affect the network structure of the front feature extraction.

2.4. Experimental Results

2.4.1. CACD Dataset and Image Preprocessing

The CACD dataset was the largest cross-age face dataset at that time. The researchers selected celebrities of different ages on IMDb.com as the objects to be collected, and finally selected the top 50 celebrities each year from the celebrities born between 1951 and 1990, for a total of a total of 2,000 celebrities. Then they use the celebrity "name + year" as the keyword to search for pictures of these celebrities through Google. The images of each celebrity is collected in the years from 2004 to 2013, so the age span was ten years. However, the pictures obtained by this method also contain a lot of noise. For example, the retrieved pictures contain multiple people or some celebrities have very few pictures published in certain years. The CACD dataset includes a total of 163,446 face images for ages from 16 to 62. In order to improve the recognition of the model, it is necessary to perform some preprocessing operations on the image data before training.

In order to further enhance the generalization ability of the model, image enhancement technology is used to construct more training samples during the training process to suppress overfitting more effectively. For the data in training set, two main image enhancement techniques are applied: horizontal flipping and random cropping. Since the face has a certain symmetry, flipping the image horizontally can make the trained model more robust to different angles of the same person. Fig. 3 shows the original image of a face in CACD and the corresponding image after alignment processing of face detection and horizontal flipping.



(a) (b)

Figure 3. Face Image Detection Alignment Processing. (a) Horizontal Flipping; (b)Random Cropping Processing

After the detection and alignment processing, the image size of the face region is 256×256, and the image size is randomly cropped to 224×224 as the input of the final training, as shown in Fig. 2-7, it can double the number of training samples, which can make the network model not sensitive to the translation transformation of some positions or even facial occlusion and effectively improves the generalization ability of the model.

2.4.2. Experimental Result and Analysis

In this experiment, a total of 120 celebrities ranked from 3 to 5 in each of the 40 years are selected as the test set, and the images of these 120 celebrities in 2013 are used as the query images. In addition, the remaining images were divided into three groups, 2004 to 2006, 2007 to 2009 and 2010 to 2012,

as the database images to be retrieved. And the effectiveness of retrieval on different age ranges is tested respectively.

Effect of Batch Normalization: In this subsection, we mainly compare the impact of batch normalization layer on deep network training experimentally. ModelA remains as it is, while ModelB adds a batch normalization layer after the convolutional layer and the fully connected layer. Then we use the same SGD optimization algorithm and a learning rate of 0.1 to train these two multi-class models on the CACD dataset.

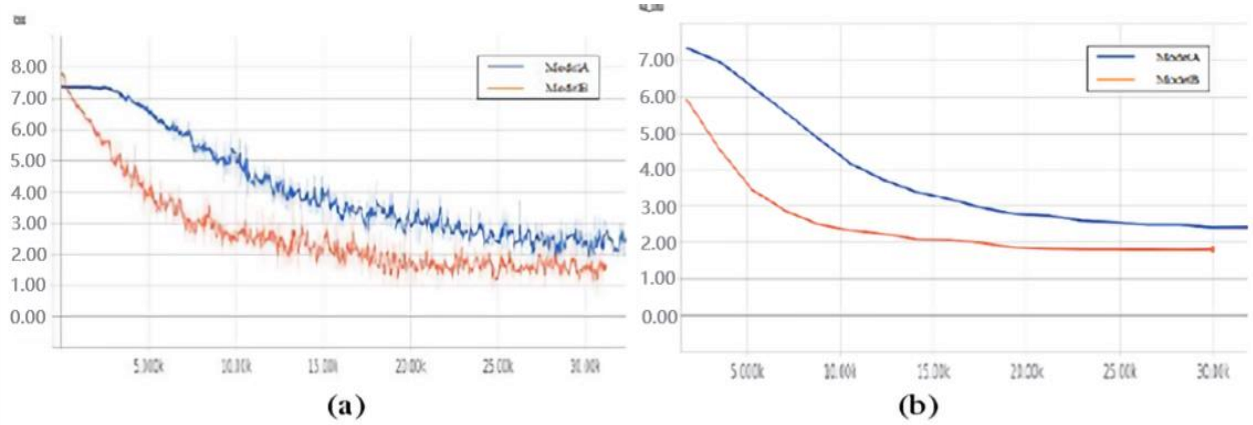


Figure 4. Loss Curves of Two Network Classification Models (a) Training Set Loss Curve; (b) Validation Set Loss Curve

Fig. 4 shows the loss curves for the two models. It is clear that the batch normalization operation accelerates the convergence of the model, and the loss values decreasing more rapidly in early stages and finally tending to converge to lower values. The experiments show that the batch normalization operation can effectively solve this problem. From the retrieval results on the CACD test set, the first ranking recognition rate is compared between the model obtained from 20 iterations of ModelA training set and the model obtained from 6 iterations of ModelB. The first recognition rates of ModelA on database of different years are 82.4%, 76.5%, and 74.46%, respectively, ModelB are 88.3%, 86%, and 82.6%, which visually reflects the difference between the features extracted by the two models.

Effect of Feature Dimension: High-dimensional feature vectors can converge faster for supervised classification training, but the vectors are relatively sparse and may contain more noise, while low-dimensional feature vectors are relatively tighter, but the accuracy improves slower during training. Under the same condition that the SGD optimization algorithm is used for model training, the feature dimensions are selected as 128, 256, 512, and 1024 for experiments. Higher-dimensional features have lower retrieval accuracy, with the highest dimensional 1024-dimensional features having the lowest MAP values for each year interval. As the dimensionality of the feature vectors gradually decreases, the retrieved MAP values are all improved. The improvement is more obvious when the feature dimension is 256 and the average retrieval accuracy reaches 63.42% on the database. When the feature dimension is further reduced to 128 dimensions, the MAP value does not continue to improve and the result of the model is slightly lower than the 512-dimensional model. It may be due to the low dimension that the features are not effectively compressed to 128 dimensions when the model is trained for the 1584-class multi-classification task, and the loss value of the validation set loss curve is relatively large when it tends to converge during training.

3. Pedestrian Recognition System

3.1. Overall System Design

The overall system design is the primary basis to build the system by analyzing the system requirements for the pedestrian recognition task and its sub-modules in the application scenario and giving the specific system architecture and process. In this paper, the pedestrian identification system

will collect images through the Web front-end, send the image data and corresponding request information to the server-side for processing, and then feed the recognition results back to the front-end. The overall architecture of the system is shown in Fig. 3-1 and it is divided into three main parts, including the image acquisition module, the pedestrian recognition module and the data storage module. The connection flow between the modules is shown in the figure below.

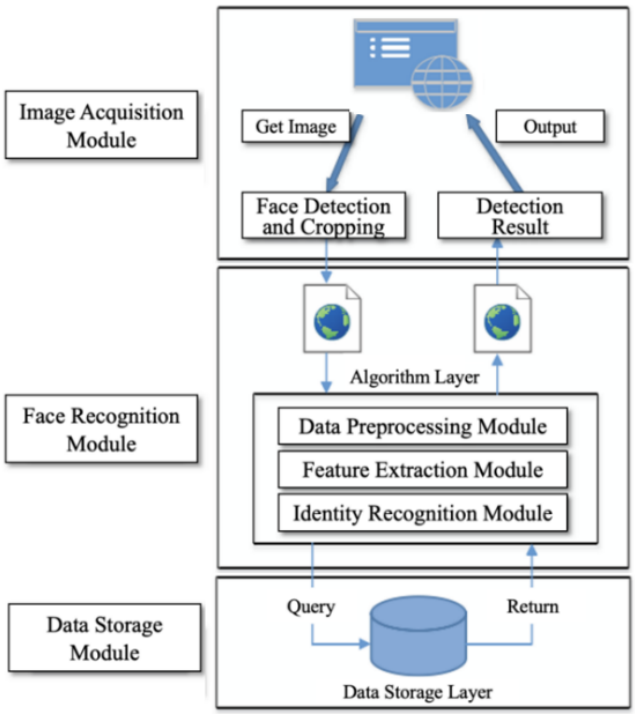


Figure 5. Overall Architecture of the System

3.2. Function Module Implementation

3.2.1. Image Acquisition Module

Image Data Acquisition Display: First, we need to upload images or access video streams. HTML5 can use multimedia interaction on the front-end. HTML5 feature WebRTC (Web Real Time Communication) provides the interface *MediaDevices.getUserMedia()* to request permission from users on various devices to obtain media input, including video sources, which can be obtained from the camera on the device. Further, after the video stream object has been obtained, if we want to input the video stream to the front-end page, we should create an URL object for the video stream and assign it to the video tag of the page for live playback.

Face Detection: In order to bring a good interactive experience, it is necessary to detect the face of the output video through canvas technology and render the detected face area on the video in real time. Face detection is assisted by introducing a lightweight javascript library, *tracking.js*, which implements the Viola-Jones face detection algorithm on the web side. It can continuously return the coordinates of the face area in the video to determine if the face has been detected through the monitoring events binding to the video stream detection.

Image Capture: In the case that a face is detected, i.e. the area coordinates returned are not empty, users can choose to perform the face image cropping to reduce the size of the image when transferred and improve response time. The video itself does not provide a screenshot interface, but we can use the *drawImage()* method in Canvas technology to achieve this function.

3.2.2. Pedestrian Attribute Recognition Module

The system uses a lightweight web application framework *flask* to provide this service. Once started, it will monitor the localhost: 5000 port by default and respond to HTTP protocol requests. During initialisation, the CNN model is trained with KERA loaded. Then route the request through

the *app.route()* method, i.e. the defined function is binding to the URL. The main URL bindings include *app.route("/")* and *app.route("*recognition*/", methods=["POST"])*, the first one returns to the front-end main page, the second one provides pedestrian attribute recognition service. Through the front-end request recognition service API, we can call the corresponding recognition functions to perform the following operations:

Data Preprocessing: First, we extract the image data and related information from the requests, perform the necessary format check on the images, convert the binary file into image matrix, and perform image preprocessing before recognition. OpenFace is used to align the faces of the image, and the images is converted to a size of 224 x 224, the same size as the input to the model. Then use the zero mean of the image's each channel. During processing, if there are data transfer interruptions, format errors and no human face is detected, the corresponding error code should be returned to the front end for processing.

Feature Extraction: After getting the correct face image data, input the image data into the CNN model for prediction, and extract the 256-dimensional feature vector of the face quickly.

Identity Recognition: The server-side database contains feature vectors extracted from faces with known identities. During the identity recognition process, the similarity between the feature vector of the image to be recognized and the feature vector in the database needs to be calculated. The system uses cosine similarity as the similarity metric. And then the similarities are ranked. For the image with the greatest similarity to the image to be recognized, an SVM classifier or a specified threshold is used to determine whether the image is of the same person and to obtain the identity information corresponding to the image for the same person. Otherwise, the image can not be successfully recognized.

3.2.3. Data Storage Module

The data storage module is implemented using Mysql database for effectively managing available identity information and facilitating quick identity recognition. Key data tables include *person*, *feature*, and *group*, as shown in Fig. 6 below.

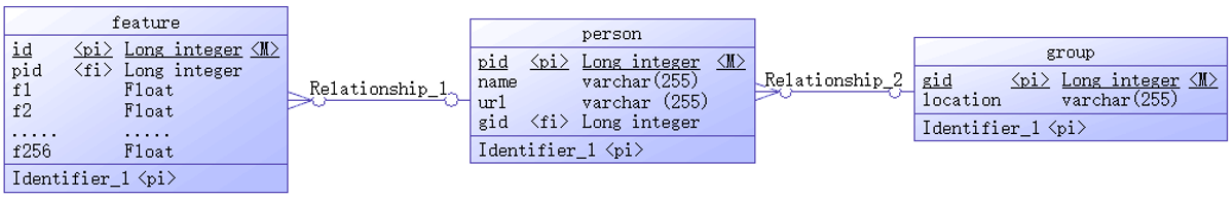
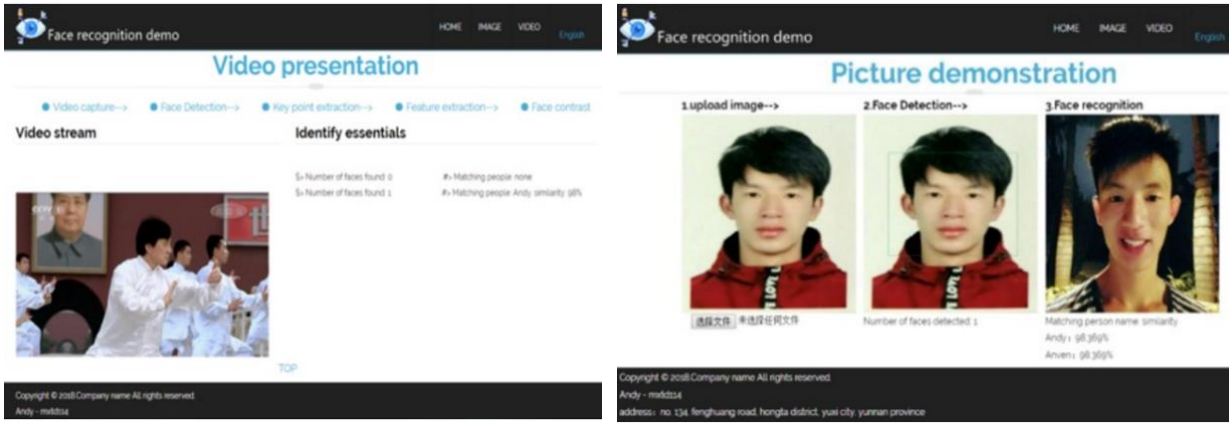


Figure 6. Database E-R Diagram

Person is used to store the identity information of the person, including the corresponding ID, name, group number and image address of the person; *Group* is used to store the grouping information, including group ID and group name location. Take the pedestrian attribute recognition of examination room scenario as an example, the location corresponds to the name of the examination room. During the recognition process, feature matching is performed between people corresponding to *gid* according to the room name information transmitted from the front-end to reduce the range of the retrieved people and improve the accuracy and response speed. Features are used to store a 256-dimensional feature vector, which is extracted in advance from each person's image. When an individual is added to the *person* table, the elements in the element table is updated accordingly. Based on the request data sent by the front-end, the corresponding set of feature vector is quickly retrieved through the database and compared with the feature vector obtained from the face images uploaded by the front-end. The corresponding identity information and the image in the database will be found in the *person* table according to the *PID* and sent to the home page for display.

3.3. Function Test

Once the video stream is obtained, the captured video image will be displayed on the web page. The web application will automatically start real-time face detection based on the video stream.



(a) (b)

Figure 7. Face Detection and Recognition System . (a) Video Recognition System; (b) Image Recognition System

4. Conclusion

Pedestrian recognition has attracted the attention of many researchers due to its wide range of practical application scenarios. At the same time, the accuracy of pedestrian recognition is often influenced by many factors. To address the recogniztion problems caused by age change, in this paper, we study the effectiveness of recogniztion methods bases on deep learning for this problem. And we design and implement a pedestrian recognition application using Web technology. We also introduce the design goal and development environment of this application and detail the implementation method of each module in this application, and test its basic functions. The application is lightweight and cross-platform, which can meet the daily needs of pedestrian recognition.

References

[1] Chuang Y U, Yang S, Kou H. Research on moving pedestrian recognition algorithm based on convolutional neural network [J]. Journal of Shenyang Normal University(Natural Science Edition), 2019.

[2] Dai X, Duan Y, Hu J, et al. Near Infrared Nighttime Road Pedestrians Recognition Based on Convolutional Neural Network [J]. Infrared Physics & Technology, 2018.

[3] Yang C, Zhang J, Chen Q, et al. UL-CNN: An Ultra-Lightweight Convolutional Neural Network Aiming at Flash-Based Computing-In-Memory Architecture for Pedestrian Recognition [J]. Journal of Circuits, Systems and Computers, 2020.

[4] Enyuan Yang, Rong Xie. Pedestrian Recognition based on Human Semantics and PCA-HOG [C]// International Conference on Mechatronics Engineering and Information Technology, 2019.

[5] Al-Hazaimeh O M, Al-Smadi M. Automated Pedestrian Recognition Based on Deep Convolutional Neural Networks[C]// 2019 the 4th International Conference on Multimedia and Image Processing (ICMIP). 2019.

[6] Wang J T, Yan G L, Wang H Y, et al. Pedestrian recognition in multi-camera networks based on deep transfer learning and feature visualization[J]. Neurocomputing, 2018, 316 (NOV.17): 166 - 177.

[7] Deng J, Zhu J. Research on pedestrian detection algorithms based on video [C]// International Conference on Computer Design and Applications. 2010.

[8] Cheng H U, Chen L, Zhang X, et al. Pedestrian Attribute Recognition Based on Convolutional Neural Network in Surveillance Scenarios[J]. Modern Computer, 2018.

[9] Heo D, Lee E, Ko B C. Pedestrian Detection at Night Using Deep Neural Networks and Saliency Maps[J]. Journal of Imaging Science and Technology, 2017, 61 (6).

[10] Matsushima K, Uchimura K, Esaki S. Pedestrian Recognition Based on Boosting using Combination of Feature Detector[J]. IEICE technical report, 2009, 109 (325): 61 - 66.

[11] Pop D O, Rogozan A, F Nashashibi, et al. Fusion of Stereo Vision for Pedestrian Recognition using Convolutional Neural Networks. 2017.

[12] Yu T, Wang C, X Liu, et al. Adaptive superpixel-based multi-object pedestrian recognition[J]. Machine Vision and Applications, 2021, 32 (1).

[13] Zhao L, Song L B, Geng M X, et al. Pedestrian Recognition and Tracking of Dancing Robot Based on Laser Scanner. 2016.

[14] Heuel S, Rohling H. Pedestrian Recognition based on 24 GHz Radar Sensors[C]// WIT 2010; International workshop on intelligent transportation. Department of TelecommunicationsrnHamburg University of TechnologyrnHamburg, 2010.

[15] Huang X P. Research on Inverted Pendulum Control Based on Genetic Neural Network [M]. Chongqing University Press, 2014.