DOI: 10.53469/wjimt.2025.08(08).01

Innovative Multi-Face Recognition Framework Leveraging Hybrid Convolutional Neural Network Architectures

Tingting Zhu, Li Zhou

School of Computer and Software, Jincheng College, Sichuan University, Chengdu 611731, China

Abstract: ***In the contemporary epoch characterized by the relentless progression of technology and the ubiquitous presence of big data, the deployment of facial recognition systems has witnessed a remarkable surge across diverse domains. Among the various approaches, deep learning-based convolutional neural networks (CNNs) have garnered substantial attention and have been increasingly adopted for their exceptional performance in facial recognition tasks. Traditional facial recognition methods often necessitate intricate and time-consuming feature extraction procedures, which are highly dependent on domain expertise and may suffer from limited generalization capabilities. In stark contrast, CNN-based facial recognition offers a paradigm shift. It obviates the need for elaborate manual feature engineering by leveraging the network's inherent capacity to automatically learn hierarchical and discriminative features from raw facial images. Specifically, in the practical implementation of CNN-based facial recognition, the OpenCV library plays a pivotal role in the initial stage of face detection. OpenCV provides a set of robust and efficient algorithms for accurately identifying facial regions within complex image scenes, thereby serving as a crucial preprocessing step. Subsequently, the detected facial images are fed into the CNN architecture for training. Through an iterative optimization process, the CNN model automatically adjusts its internal parameters to minimize the recognition error, ultimately resulting in a well-trained and feasible network model.***

Keywords: Facial Recognition OpenCv Convolutional Neural Network.

1. INTRODUCTION

Facial recognition is widely used in various fields, such as facial attendance, facial socialization, facial payment, suspect tracking, traffic check-in, etc. Facial recognition can be broadly divided into two categories: one is facial detection, which identifies the position and size of a face in an image, and the other is facial recognition, which compares it with existing collected images in a database to determine if it is the same person, and finally completes identity verification. There are various facial recognition algorithms, but traditional methods rely on manually designed features and machine learning algorithms, such as extracting edges, textures, lines, boundaries, and other features from images. Based on these features, the next step of processing is carried out, but the efficiency of such processing is relatively low. Therefore, manual feature extraction and machine learning techniques have been replaced by widely used convolutional neural network training data. The feature and advantage of deep learning neural network algorithms is that they can be trained on a relatively large dataset to learn its surface features. In addition to facial recognition, CNN is also widely used in facial expression recognition, object recognition, age analysis, and other fields.

Empirical studies have demonstrated that CNN-based facial recognition systems, when properly trained and optimized, can achieve superior recognition accuracy and robustness compared to traditional methods, especially in scenarios with varying lighting conditions, facial expressions, and occlusions. This highlights the immense potential of CNNs in advancing the field of facial recognition and paves the way for its wider application in real-world scenarios, such as security surveillance, access control, and human-computer interaction.

Li, Lin, and Zhang (2025) proposed a privacy-preserving framework combining federated learning and differential privacy for personalized advertising, addressing critical data confidentiality concerns [1]. In the domain of urban design, Xu (2025) introduced CivicMorph, a generative modeling approach for public space form development [2]. Concurrently, Tu (2025) presented SmartFITLab, an intelligent platform designed for the execution and validation of 5G field interoperability testing, enhancing network infrastructure robustness [3]. For human resource technology, Xie and Liu (2025) developed EvalNet, a system utilizing sentiment analysis and multimodal data fusion to process recruitment interviews [4]. Zhu (2025) explored language agents with TaskComm, a task-oriented agent aimed at optimizing workflows for small businesses [5]. Further supporting small enterprises, Zhang (2025) investigated reinforcement learning techniques for automated ad campaign optimization in

"Learning to Advertise" [6]. Hu (2025) contributed to 3D content creation for small and medium-sized enterprises with "Learning to Animate," focusing on few-shot neural editors [7]. Developer tooling for large language models was advanced by Zhang (2025) through InfraMLForge, enabling rapid LLM development and scalable deployment [8]. In healthcare AI, Ding and Wu (2024) conducted a systematic review on self-supervised learning applications for processing ECG and PPG biomedical signals [9]. Addressing challenges in recommendation systems, Wang (2025) proposed a joint training method for propensity and prediction models using targeted learning, specifically for data missing not at random (MNAR) scenarios [10]. Lin (2025) addressed product management needs in AI systems by introducing a framework for digital experience observability [11]. Finally, foundational applications were reinforced by Chen (2023), who discussed the utilization of data mining techniques within broader data analysis contexts [12].

2. DATA COLLECTION

The collection of facial recognition is carried out through electronic devices such as cameras. This article will use 33 students in the class, each using OpenCV+Haar features to extract 10 color images through a camera. Each photo has different expressions, actions, positions, lighting, etc. Place them in the same directory and name them name_ 00.TIF~name_09.TIF uniformly.

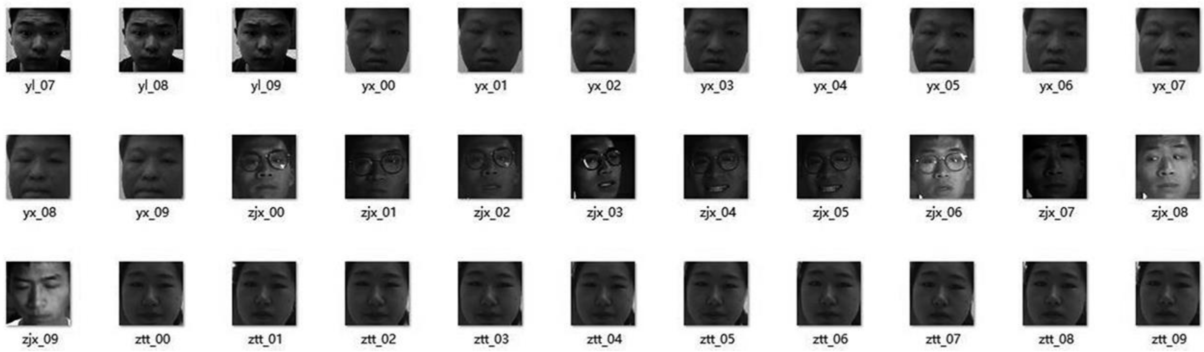

Figure 1: Collection of facial image data

3. DATA PREPROCESSING

3.1 Face detection based on OpenCV

Opencv is a cross platform computer vision library that enables a wide range of image processing [1] and has high accuracy in face detection.

Using OpenCV's FaceDetector to detect faces and extract them from videos, for the face detection part, Haar feature classifiers can be used, including those for glasses, head, mouth, nose, and other areas. In this article, the haarcascad_ron-talface_default.xml classifier is used.

3.2 Modifying Image Size and Data Enhancement

3.2.1 Modifying Dimensions

Use Haar features to uniformly extract facial features from photos, and modify and save the photos to a size of 60 * 60.

3.2.2 Data augmentation

For some images, the lighting is too weak, causing the face and background in the photo to blend together, and their pixel points will be very close. The deformation of the face itself will also cause obstacles to computer recognition. Therefore, enhancing the original image can improve the accuracy of the task to a certain extent. Use the flip() function in the OpenCV library to horizontally flip the original samples, use the contrast-brightness_demo() function to adjust brightness, and expand the data to 1320 images.

3.3 Label and divide the training set, validation set, and testing set

3.3.1 Labeling

Use the imread() function in the OpenCV library to obtain the matrix information of the image, use the get_dummies() function in the Pandas library to perform one hot encoding and set labels for the data, and finally integrate the labels with the image information into a DataFrame format for easy subsequent processing.

3.3.2 Dividing the Training Set

Take 70% of the processed data as the training set, 20% as the testing set, and 10% as the validation set. In order to make the samples for each training different and prevent overfitting, the shuffle function is used to shuffle the training set.

4. A BRIEF DISCUSSION ON CNN

4.1 Introduction to CNN

Convolutional Neural Network (CNN) [2] is a type of neural network that includes convolutional computation and has a deep structure. It is one of the algorithms in deep learning.

4.2 Main Structure

4.2.1 Input Layer

In the convolutional neural network for image processing, its input layer is usually a pixel matrix of an image, designed in three dimensions, such as 28 * 28 * 3, where 28 is the image size, 3 is the depth or channel of the image, and color images are usually represented by three channels: R, G, and B.

4.2.2 Convolutional Layer

Usually used for feature processing. Filter, Convolutional kernels or convolutional layer filters are three-dimensional data with the same depth as the input image. Set the step size, slide the filter over the width and height of the input data, and then calculate the inner product of each channel's filter with any point in the input data. After the output of the convolution kernel, there is also an activation function to introduce nonlinearity into the network.

4.2.3 Pooling layer

Compressing the matrix size reduces computational resource consumption and effectively controls overfitting to improve generalization.

4.2.4 Fully Connected Layer

Connect all local parts together through a fully connected layer and perform the final classification to obtain the result.

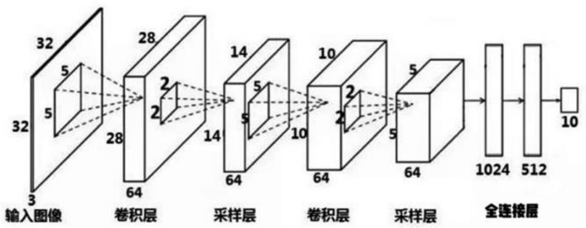

Figure 2: Convolutional Neural Network Structure Diagram

5. BUILDING CONVOLUTIONAL NETWORKS

Batch value: The initial training batch and the number of training samples in each batch.

Input layer: x represents image information, y represents classification information.

Convolutional layer: Using 3 * 3 sized convolution kernels to continuously convolve images and obtain feature maps, starting from the second layer, the filter size changes to the depth of the image after convolution in the previous layer, and using the ReLU activation function to introduce nonlinearity into the network. The convolution step size of the last layer changes from 1 to 2.

Pooling layer: Reduce the size of matrix information by half.

Fully connected layer: The size and depth of the image are calculated while filling the boundaries. Multiple neurons are set up in the fully connected layer, and the convolutional feature map is passed in. The final output layer consists of 33 neurons.

Softmax layer: Use a softmax classifier to map the output results to a range of 0-1, and determine the highest score as the correct classification.

After building the basic network framework and obtaining the results, it is necessary to calculate its loss value, and then propagate the error through backpropagation, continuously update the weights using gradient descent, and iteratively train the network. Each iteration needs to shuffle the training set to enhance the model's generalization ability. The obtained classification is compared with the validation set to output the validation accuracy, and finally the validation accuracy is output through the test set.

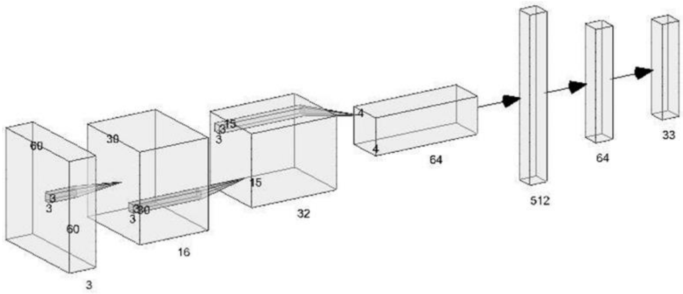

Figure 3: Convolution Process Structure Diagram

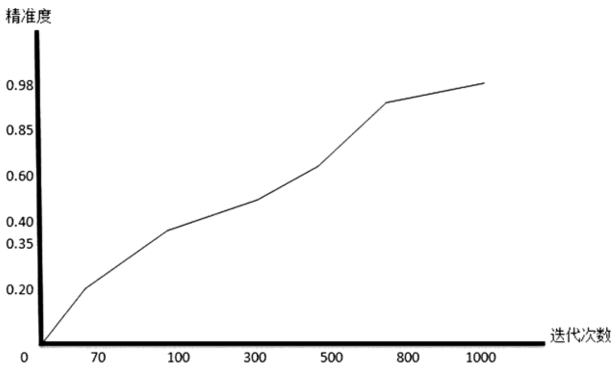

Figure 4: Accuracy variation curve

```
epoch: 936 valid_acc:0.98
epoch: 937 valid_acc:0.98
epoch: 938 valid_acc:0.98
epoch: 939 valid_acc:0.98
epoch: 940 valid_acc:0.98

epoch: 941 valid_acc:0.98
epoch: 942 valid_acc:0.98
epoch: 943 valid_acc:0.98
epoch: 944 valid_acc:0.98
```

Figure 5: The final accuracy chart of the network model

6. OPTIMIZE THE NETWORK

6.1 Optimizing Training Batches

We use small batches of training data and set the initial batch size. As the number of training sets is not too large, we can reduce the batch size appropriately, but it should not be too small to avoid slow training speed. Sizes such as 20, 30, 50, and 100 can be selected through training to choose the best batch. In this article, we trained the best batch with 100.

6.2 Optimizing the size, number, and number of convolutional kernels

When training the convolutional layer, you can set the length and width of the convolutional kernel by yourself. In this paper, a size of 3 * 3 is preferred. In the first layer of convolution, the depth of the convolutional kernel should be kept consistent with the image channel at 3, and the number of filters should be adjusted by yourself. The first layer of the network uses 16 filters, the second layer has 32 filters, and the third layer has 64 filters. The network can also be adjusted by increasing or decreasing the number of convolutional layers, and this article uses three layers.

6.3 Optimize the number and layers of neurons in the fully connected layer

Setting the number of neurons and the number of fully connected layers in the fully connected layer can appropriately increase and improve accuracy. If it is too large, it can also lead to slower training speed. The network has 512 neurons in the first layer and 64 neurons in the second layer

6.4 Optimizing Learning Rate

In backpropagation gradient descent, the learning rate or learning step size is usually set to 0.1, 0.01, 0.001, etc. If it is too small, it will cause slow training speed, and if it is too large, it will lead to crossing the optimal value. By tuning the parameters, the optimal learning rate of the network is found to be 0.001

6.5 Optimization iteration times

In training the network, multiple iterations are required to calculate accuracy, and the number of iterations needs to be set by oneself. If it is too small, it will result in low accuracy. Therefore, a larger number of iterations is chosen to optimize the network model. This article uses 2000 iterations.

7. REAL TIME FACIAL RECOGNITION

In this project, to achieve real-time face recognition, it is necessary to use the OpenCv library method to obtain the face data that needs to be recognized in real time.

A model with an accuracy of up to 98% on the validation set has been trained using convolutional neural networks, and now we need to use this trained model to predict faces in real time. So first, we need to call the camera with OpenCv to capture real-time video streams, and then use OpenCv's face detection model to detect faces in the video window and capture facial images. Perform some processing on the captured facial images, such as changing the size to the size required by the network. Then pass this image into the network, use the model's parameters for recognition, obtain a probability for each person's category, and finally return the name of the person with the highest probability.

Display the person's name in the video window by using a function to annotate the image.

After multiple tests, the accuracy of real-time recognition is still very high, which proves that the established model has strong learning ability for image features and can accurately recognize every person.

8. CONCLUSION

With the rapid development of technology, in this era of big data, facial recognition has brought great convenience to society with its efficient recognition accuracy. Neural network-based multi face recognition has good feature extraction ability and strong learning ability. By using the OpenCv library to extract and detect faces, and then using the captured face data to train the network, face recognition can be achieved. However, due to the small number of images, the accuracy obtained from the training samples is not high enough, and the later accuracy will increase with the increase of the number of samples in each category. During the process of training a network, overfitting may sometimes occur due to the limited amount of data and complex model. This can be achieved by limiting the fitting ability by obtaining more data, reducing the number of network layers and neurons, etc.

REFERENCES

[1] Li, X., Lin, Y., & Zhang, Y. (2025). A Privacy-Preserving Framework for Advertising Personalization Incorporating Federated Learning and Differential Privacy. arXiv preprint arXiv:2507.12098.
[2] Xu, Haoran. "CivicMorph: Generative Modeling for Public Space Form Development." (2025).
[3] Tu, Tongwei. "SmartFITLab: Intelligent Execution and Validation Platform for 5G Field Interoperability Testing." (2025).
[4] Xie, Minhui, and Boyan Liu. "EvalNet: Sentiment Analysis and Multimodal Data Fusion for Recruitment Interview Processing." (2025).
[5] Zhu, Bingxin. "TaskComm: Task-Oriented Language Agent for Efficient Small Businesses Workflows." (2025).
[6] Zhang, Yuhan. "Learning to Advertise: Reinforcement Learning for Automated Ad Campaign Optimization for Small Businesses." (2025).
[7] Hu, Xiao. "Learning to Animate: Few-Shot Neural Editors for 3D SMEs." (2025).
[8] Zhang, Yuhan. "InfraMLForge: Developer Tooling for Rapid LLM Development and Scalable Deployment." (2025).
[9] Ding, C.; Wu, C. Self-Supervised Learning for Biomedical Signal Processing: A Systematic Review on ECG and PPG Signals. medRxiv 2024.
[10] Wang, Hao. "Joint Training of Propensity Model and Prediction Model via Targeted Learning for Recommendation on Data Missing Not at Random." AAAI 2025 Workshop on Artificial Intelligence with Causal Techniques. 2025.
[11] Lin, Tingting. "Digital Experience Observability in AI-Enhanced Systems: A Framework for Product Managers." ResearchGate, Mar (2025).
[12] Chen, Rensi. "The application of data mining in data analysis." International Conference on Mathematics, Modeling, and Computer Science (MMCS2022). Vol. 12625. SPIE, 2023.