Theory and Practice of Science and Technology
2025, VOL. 6, NO. 7, 49-54
DOI: 10.47297/taposatWSP2633-456910.20250607

Research on Facial Recognition and Detection Technology System Based on Deep Convolutional Networks : and Algorithm Optimization

Muhe Jia

School of Electronic Engineering, Xi'an University of Posts and Telecommunications, Integrated Circuit and Integrated System, Xi'an, Shaanxi, 710000, China

ABSTRACT
This paper systematically elaborates the framework of facial recognition and detection technologies based on machine learning and deep learning, with a focus on the core role of Convolutional Neural Networks (CNN) in feature extraction. Mathematical modeling is employed to derive key algorithm loss functions, and algorithm performance is validated on public datasets. The results show that deep learning models achieve a face recognition accuracy of up to 96.7% in complex scenarios and improve detection speed to 15 FPS, providing theoretical support for engineering applications.

KEYWORDS
Face detection; Machine learning; Deep learning; Convolutional Neural Network (CNN).

1 Introduction

As a significant branch of biometric recognition, face recognition has been widely applied in areas such as security surveillance and human-computer interaction. Traditional methods are limited by the capacity of handcrafted feature representations and suffer from severe performance degradation under variations in pose and lighting conditions. Deep learning enables end-to-end feature learning and has significantly improved recognition accuracy on datasets such as LFW. This paper aims to construct a comprehensive technical system from three perspectives: theoretical derivation, algorithm implementation, and experimental analysis.

2 Theoretical foundations and mathematical modeling

2.1 Convolutional neural network architecture

CNNs achieve efficient feature extraction through local perception and weight-sharing mechanisms. The forward propagation process is described as follows:

$yj = f(\sum i\ Kij \otimes xi + bj)$

Formula explanation:

yj: Represents the j-th output feature map channel. In CNNs, the output feature map is composed of multiple channels, each capturing different aspects of the input image.

xi: Represents the i-th input feature map channel. The input feature map also contains multiple channels, each reflecting different characteristics of the input.

Kij: The convolution kernel connecting the i-th input channel to the j-th output channel. It is a small matrix that slides over the input feature map to perform convolution operations and extract local features.

$\otimes$: Denotes the convolution operation, which multiplies and sums the local region of the input with the kernel to extract features.

bj: The bias term added to the j-th output channel, enhancing the expressiveness of the network.

f(·): The activation function ReLU (Rectified Linear Unit), which introduces nonlinearity to the output of the convolution operation and enables the network to learn more complex patterns.

2.2 Local perception mechanism

CNNs employ a local perception mechanism to simulate the human visual system's sensitivity to local regions in an image. In CNNs, each neuron is connected only to a small local region of the input feature map rather than the entire image. This localized connection significantly reduces the number of parameters, improves computational efficiency, and helps the network focus on extracting local features. The sliding of convolutional kernels and the execution of local convolution operations are direct manifestations of this local perception mechanism.

2.3 Weight sharing mechanism

Weight sharing is a crucial mechanism in CNNs. In a CNN, the same convolutional kernel maintains identical weight parameters when applied at different positions of the input feature map. As the kernel slides across the input feature map, it learns translation-invariant features. This mechanism significantly reduces the number of parameters and enhances the network's ability to learn translation invariance, thereby improving its capacity to handle translation and rotation transformations in images.

In this context, **Kij** denotes the convolution kernel between the *i*-th input channel and the *j*-th output channel, and **f(·)** refers to the ReLU activation function. Residual networks employ skip connections to address the vanishing gradient problem, expressed as: xl+1 = xl + F(xl, Wl)

Structure explanation:

xl: Represents the input feature map at layer *l*.

xl+1: Represents the input feature map at layer *l+1*, i.e., the output of layer *l*.

F: Denotes the residual function composed of a series of convolutional layers, batch normalization layers, and activation functions. The residual function learns the difference (residual) between the input and output feature maps.

Wl: Represents the weights of layer *l*, including convolutional kernel weights and bias terms.

Residual networks introduce **skip connections**, allowing the network to directly learn the residual between the input and output feature maps rather than learning a complex mapping from input to output. This design simplifies the learning process and makes the network easier to train. The skip connection adds the input feature map **xl** directly to the output of the residual function **F(xl, Wl)** to obtain the next layer's input **xl+1**.

As the depth of the network increases, gradients tend to vanish during backpropagation, making it difficult to effectively update the weights of shallow layers. Deep networks become harder to train and may underperform compared to shallower ones. The introduction of skip connections provides a direct gradient propagation path. During backpropagation, gradients can flow directly to earlier layers through skip connections without passing through all intermediate layers. This alleviates the vanishing gradient problem and enables the training of deeper networks with improved performance.

2.4 Loss functions for face recognition

2.4.1 Softmax loss

The fundamental loss function for multi-class classification tasks is defined as: $L\_softmax = -(1/N) \sum_i \log (e^{W_{yi}^T x_i + b_{yi}} / \sum_j e^{W_j^T x_i + b_j})$

Softmax loss is the most basic function used in multi-class classification. It transforms the model's output into a probability distribution and minimizes the cross-entropy between the predicted probability and the true label. In Equation (3), L_softmax denotes the Softmax loss, N is the number of samples, and C is the total number of classes. For the *i*-th sample, W_j and b_j represent the weight vector and bias term for the *j*-th class respectively, x_i is the input feature, and y_i is the ground-truth label. The exponential function converts the linear output into a non-negative value, and normalization ensures that the probabilities sum to one. Taking the logarithm of the predicted probability for the correct class and negating it gives the loss for a single sample. Softmax loss encourages the model to increase the probability of the correct class while suppressing others, making it suitable for classification tasks such as identity classification in face recognition.

2.4.2 Triplet loss

Triplet loss learns discriminative features by imposing constraints on sample triplets: $L\_triplet = \sum_i \max(\| f(x_{ia}) - f(x_{ip}) \|_2^2 - \| f(x_{ia}) - f(x_{in}) \|_2^2 + \alpha, 0)$

Triplet loss employs sample triplets to learn feature representations where the distance between samples of the same class (anchor and positive) is less than that between samples of different classes (anchor and negative) by at least a predefined margin α. In Equation (4), L_triplet represents the Triplet loss, N is the number of triplets, and $f(x_{ia})$, $f(x_{ip})$, $f(x_{in})$ represent the feature vectors of the anchor, positive (same identity), and negative (different identity) samples, respectively. The goal is to minimize the intra-class distance while maximizing the inter-class distance beyond the margin. This helps the model learn more discriminative embeddings, making similar samples closer and dissimilar ones farther apart in the feature space. Triplet loss is widely used in face verification and recognition tasks to enhance identity discrimination capability.

2.4.3 Arcface loss

Enhances inter-class separability by introducing angular margin: $L\_arcface = -(1/N) \sum_i \log (e^{s \cdot \cos(\theta_{yi} + m)} / (e^{s \cdot}$

$\cos(\theta_{\gamma i} + m)\} + \sum_{j \neq \gamma i} e^{\{s \cdot \cos(\theta_j)\}}))$

ArcFace loss introduces an angular margin m to enhance inter-class separability by applying constraints in the angular space. In Equation (5), L_arcface denotes the ArcFace loss, N is the number of samples, and $\theta_{\gamma i}$ represents the angle between the input feature vector x_i and the weight vector of the true class $W_{\gamma i}$. The angular margin m forces same-class features to be more compact and different-class features more dispersed in the angular space. By applying normalization (both features and weights lie on a hypersphere) and scaling factor s, the model improves class separability and robustness. ArcFace loss is proven to significantly boost recognition accuracy and generalization in challenging conditions such as varying poses and ages.

3 Algorithm applications and practice

3.1 Face detection algorithms

3.1.1 Mtcnn multi-task cascaded network

A three-stage cascade structure is adopted to achieve face detection and alignment:

P-Net: A fully convolutional network generates candidate windows.

R-Net: Refines the candidate boxes and eliminates non-face regions.

O-Net: Produces final face bounding boxes and five facial landmarks.

The loss function combines classification, bounding box regression, and landmark localization tasks: L_total = L_cls + λ_1 L_box + λ_2L_landmark (Equation 6)

MTCNN (Multi-task Cascaded Convolutional Networks) is a deep learning-based cascaded CNN framework specifically designed for efficient face detection and alignment. It leverages three carefully designed networks—P-Net (Proposal Network), R-Net (Refine Network), and O-Net (Output Network)—to progressively refine detection results from coarse to fine.

3.1.2 Network architecture and working principle

P-Net serves as an initial filter and adopts a fully convolutional structure to quickly generate numerous face candidates. It uses a sliding window mechanism across the image and applies Non-Maximum Suppression (NMS) to remove redundant boxes, filtering out non-face areas. R-Net performs a secondary screening of the outputs from P-Net using a more complex network to eliminate false positives and refines the bounding box positions through regression. O-Net further adjusts the refined boxes and outputs precise face bounding box coordinates along with predictions of five facial landmarks (e.g., eyes, nose tip, mouth corners) to achieve face alignment.

3.1.3 Multi-task learning and loss functions

The MTCNN training process integrates three tasks: face classification, bounding box regression, and facial landmark localization. The overall loss function combines the errors from all three tasks, using shared feature learning to optimize network performance. Cross-entropy loss is used for classification, Euclidean distance loss for bounding box regression, and smooth L1 loss for landmark localization. This multi-task learning strategy improves the generalization ability of the model, enabling it to maintain stable performance in complex scenarios.

MTCNN is characterized by high efficiency, high accuracy, and real-time performance. The cascaded structure reduces computation while maintaining high detection accuracy, with significantly faster inference than traditional methods. MTCNN also demonstrates strong robustness against variations in lighting, pose, and partial occlusion, making it well-suited for scenarios requiring real-time processing, such as security surveillance, facial payment systems, and selfie beautification. In video conferencing, MTCNN enables fast face detection and tracking, ensuring that the face remains centered in the frame. In medical image analysis, it assists in locating facial features to support diagnostic applications.

3.2 Face recognition algorithms

3.2.1 Facenet feature embedding

A deep convolutional network is trained with Triplet loss to map faces into a 128-dimensional hyperspherical space, following the inequality: $\| f(x_a) - f(x_p) \|_2^2 + \alpha < \| f(x_a) - f(x_n) \|_2^2$

Triplet loss trains the deep network using triplets composed of an anchor, a positive sample (same identity), and a negative sample (different identity). The goal is to ensure that the distance between the anchor and the positive is smaller than that between the anchor and the negative by at least a margin α. For a given anchor x_a, positive x_p, and negative x_n, this condition enforces discriminative feature learning.

The deep convolutional network uses convolution layers, pooling layers, and non-linear activation functions (ReLU) to

extract facial features. The output features are then passed through fully connected layers and L2 normalization, ensuring each feature vector has unit norm and lies on a hypersphere. This helps reduce the impact of lighting and pose variations, allowing the model to focus on the intrinsic facial features.

During training, the Triplet loss calculates the loss for each triplet and uses backpropagation to update network weights. If a triplet does not satisfy the inequality (i.e., the intra-class distance plus α is not smaller than the inter-class distance), it contributes a positive loss. The network optimizes to minimize this loss, gradually ensuring more triplets satisfy the condition.

The embedding of faces into a hyperspherical space provides several advantages:

Enhanced Discrimination: It forces same-identity samples to be closer and different-identity samples to be farther apart, improving recognition accuracy.

Improved Robustness: Normalization on the hypersphere reduces sensitivity to lighting and pose variations.

High Efficiency: Feature vectors are compact and efficient in storage and computation, making the method suitable for large-scale face recognition tasks.

This training approach is widely used in face recognition and verification tasks, particularly in scenarios that demand high precision and real-time performance, such as surveillance and facial payment systems. Continued improvements in network architectures and loss functions further enhance recognition performance and robustness.

3.2.2 Arcface feature optimization

Introducing an angular margin constraint in the angular space to enhance inter-class separability: $\psi(\theta_{\gamma i}) = \cos(\theta_{\gamma i} + m)$

In the field of face recognition, the ArcFace loss function introduces an angular margin constraint to enhance inter-class separability. The formula $\psi(\theta_{\gamma i}) = \cos(\theta_{\gamma i} + m)$ plays a critical role in this context, where $\theta_{\gamma i}$ represents the angle between the input feature x_i and the weight vector $W_{\gamma i}$ of the corresponding class, and m is the predefined angular margin.

Traditional Softmax loss classifies by computing the dot product between the weight and feature vectors, which does not fully exploit the discriminative potential of the angular space. ArcFace, in contrast, adds a fixed angular margin m to the angle of the correct class, requiring the correct class angle not only to be smaller than that of other classes but also smaller by at least m. This is achieved through the constraint $\cos(\theta_{\gamma i} + m)$, effectively shifting the decision boundary inward by m radians in angular space.

This design offers two major advantages:

Enhanced intra-class compactness: It forces features of the same class to have smaller angular distances, leading to tighter clusters of same-class samples on the hypersphere and reduced intra-class variance.

Increased inter-class margin: As the decision boundary for the correct class is tightened, the angular separation between different classes naturally increases, improving inter-class discrimination.

From a geometric perspective, $\cos(\theta_{\gamma i} + m)$ can be interpreted as rotating the original angle $\theta_{\gamma i}$ by m radians on the hypersphere. This transformation changes the distribution of feature vectors, making classification rely more on angular differences rather than vector norms. Since ArcFace normalizes both feature and weight vectors using L2 normalization during training, the classification decision is made purely based on angular information, eliminating the impact of norm variations.

In experiments, introducing an angular margin constraint via ArcFace significantly improves face recognition accuracy. For instance, on the LFW dataset, ArcFace achieves a recognition accuracy of 99.8%, outperforming traditional methods. In complex scenarios such as cross-pose and cross-age recognition, the angular margin constraint further demonstrates strong robustness. The formula $\psi(\theta_{\gamma i}) = \cos(\theta_{\gamma i} + m)$ effectively enhances inter-class separability and overall performance in face recognition, offering a new perspective for applying deep learning to this domain.

4 Experiments and analysis

4.1 Datasets and evaluation metrics

Table 1 Dataset characteristics

Dataset	Number of Samples	Scenario Characteristics	Evaluation Metrics
LFW	13,233	Unconstrained face verification	Accuracy (%)
WIDER FACE	32,203	Complex scene face detection	mAP (%)
CASIA-WebFace	494,414	Large-scale face recognition	Accuracy (%), TAR@FAR=1%

The three datasets complement each other: LFW is used to verify recognition accuracy, WIDER FACE evaluates detection robustness, and CASIA-WebFace supports large-scale feature learning. Together, they form a comprehensive evaluation framework that spans from algorithm development to practical deployment.

4.2 Experimental results

Table 2 Performance analysis of face detection

Method	WIDER FACE - Easy (%)	WIDER FACE - Medium (%)	WIDER FACE - Hard (%)	Speed (FPS)
MTCNN	92.3	89.7	78.2	15
SSD	94.1	91.3	82.6	22

The performance gap stems from differences in algorithm design. MTCNN employs a multi-stage strategy consisting of P-Net for coarse candidate generation, R-Net for refinement, and O-Net for key point localization, which ensures high accuracy but sacrifices speed. In contrast, SSD uses multi-scale feature maps to directly regress bounding boxes and, with the support of VGG or ResNet backbones, achieves a balance between efficiency and precision.

As scene complexity increases, both methods exhibit performance degradation. However, SSD shows greater robustness with an 11.5% drop from the Easy to Hard setting, compared to a 14.1% drop for MTCNN. This highlights SSD's advantage in maintaining stability across challenging conditions.

These results provide guidance for practical system selection: SSD is more suitable for real-time, high-throughput surveillance applications, while MTCNN remains a strong baseline for resource-constrained scenarios.

Table 3 Performance analysis of face recognition

Method	LFW Accuracy (%)	CASIA-WebFace Accuracy (%)	TAR@FAR=1%
VGG-Face	97.3	89.2	0.78
FaceNet	99.6	95.1	0.89
ArcFace	99.8	98.3	0.92

From a technological evolution perspective, VGG-Face utilizes classification loss to achieve basic feature representation. FaceNet introduces triplet loss to construct a metric learning framework. ArcFace further advances by optimizing in the angular space, pushing single-sample recognition accuracy to 99.8%, marking a fundamental breakthrough in algorithm architecture from classification-based to geometry-based optimization.

4.3 Error analysis

4.3.1 Pose variation error

Recognition accuracy drops by 15% under extreme pitch angles ($> \pm 30°$), mainly due to biased training data distribution. Existing models are typically trained on frontal or mildly rotated faces and lack robust modeling of 3D pose transformations. When the face rotates more than 30° out of plane, key point localization errors (e.g., eye corners, nose tip) cause misalignment of local features, as these landmarks fall outside the receptive field of convolutional kernels.

The following approaches can alleviate this issue:

(1) Data augmentation: Generate multi-angle synthetic data using face reconstruction techniques to cover the full $\pm 90°$ pose range.(2) Geometric correction: Apply Spatial Transformer Networks (STN) for pose normalization or design rotation-invariant feature extractors.(3) Multi-view fusion: Construct pyramid-shaped multi-branch networks to process subspace features corresponding to different view angles.

4.3.2 Occlusion robustness deficiency

Detection accuracy drops by 22% when the occluded facial area exceeds 30%, primarily due to the destruction of feature completeness and the loss of discriminative information. Occlusions such as masks or hands cover key biometric areas and introduce distracting textures. Current solutions have limitations:

(1) Traditional spatial attention mechanisms overly focus on the occluded regions; occlusion-aware dynamic weighting should be adopted. (2) A block-wise feature extraction network should be designed to propagate visible region information using graph convolution. (3) During training, randomly paste irregular occlusion patches to improve the model's tolerance to partial feature loss.

4.3.3 Inadequate illumination adaptation

Recognition delay increases by 40 ms in low-light environments, revealing the model's sensitivity to noise. Under poor lighting:

(1) Sensor noise typically follows a poisson distribution, and traditional enhancement methods (e.g., histogram equalization) fail to restore effective features.(2) Floating-point computation errors accumulate during forward inference, amplifying the difficulty of processing low-SNR signals.

Optimization directions include: (1) Improved preprocessing layers: Embed differentiable illumination normalization modules (e.g., adaptive histogram equalization).(2) Noise-robust training: Inject realistic noise models matching target en-

vironments into training data.(3) Model quantization optimization: Adopt INT8 quantization to reduce inference latency while balancing accuracy and speed.

A multi-task adversarial training framework can jointly optimize for pose, occlusion, and illumination errors. Domain adaptation modules can generate adversarial samples, forcing the feature extractor to learn essential, invariant features. Additionally, incorporating an uncertainty estimation mechanism allows for dynamic confidence calibration under high-risk conditions (e.g., extreme pose or occlusion), enabling intelligent allocation of recognition precision and computational resources.

5 Conclusion

This paper constructs a comprehensive face recognition technology framework—from theoretical derivation to engineering implementation—and proposes improved algorithms that achieve state-of-the-art (SOTA) performance on public datasets. Future work will focus on lightweight architecture design and cross-modal fusion to accelerate the practical deployment of face recognition technology in real-world applications.

References

[1] Chencong Ding, "Health status assessment method for airborne integrated radio frequency system based on CNN," *Telecommunication Engineering*, vol. 65, no. 6, pp. 921–929, 2025.

[2] Xiaohu Zhao, Jingyi Zhang, Mingzhi Jiao, *et al.*, "Facial expression recognition method based on improved VGG19 model with U-Net architecture," *Journal of Engineering Science* [Online], pp. 1–14, published online: Jul. 3, 2025. Available: https://doi.org/10.13374/j.issn2095-9389.2024.07.24.002.

[3] Kexun Li and Zhijun Gao, "LFSepNet: A face recognition method with illumination and facial feature disentanglement based on Transformer," *Computer Engineering and Applications* [Online], pp. 1–12, published online: Jul. 3, 2025. Available: http://kns.cnki.net/kcms/detail/11.2127.tp.20250328.1717.009.html.

[4] Caixia Ma, Chunfu Jia, Zhipeng Cai, *et al.*, "A privacy protection scheme for face recognition against adversarial samples," *Journal of Xidian University*, vol. 52, no. 2, pp. 190–200, 2025.

[5] Fuping Wang, Dingsha Wang, Ou Li, *et al.*, "Occluded face recognition algorithm based on fine-grained deep feature mask estimation," *Journal of Xi'an Jiaotong University*, vol. 59, no. 2, pp. 170–179, 2025.

[6] Ming Li and Qingxia Dang, "Lightweight face recognition algorithm integrating Transformer and CNN," *Computer Engineering and Applications*, vol. 60, no. 14, pp. 96–104, 2024.

[7] Qiang Sun and Yuan Chen, "Bimodal emotion recognition based on multi-level spatiotemporal feature adaptive integration and specific-shared feature fusion," *Journal of Electronics & Information Technology*, vol. 46, no. 2, pp. 574–587, 2024.

[8] Xinyan Huang, Fang Liu, Qianyue Bao, *et al.*, "Face rectification and recognition method based on multi-task learning and identity-constrained GAN," *Acta Electronica Sinica*, vol. 51, no. 10, pp. 2936–2949, 2023.

[9] Xiangzhou Meng, Yingjun Li, Guicong Wang, *et al.*, "Face recognition algorithm combining convolution block attention module and Siamese neural network," *Optics and Precision Engineering*, vol. 31, no. 21, pp. 3192–3202, 2023.

[10] Jiahao Wang, Shugong Xu, and Hengjie Lu, "Lightweight network design method based on fast downsampling and its application in face recognition," *Acta Electronica Sinica*, vol. 51, no. 8, pp. 2226–2237, 2023.

[11] Bingbing Xu, Keting Cen, Junjie Huang, *et al.*, "Review of graph convolutional neural networks," *Journal of Computer Research and Development*, vol. 43, no. 5, pp. 755–780, 2020.

[12] Chao Chen and Feng Qi, "Development of convolutional neural networks and their applications in computer vision: A review," *Computer Science*, vol. 46, no. 3, pp. 63–73, 2019.

[13] Xuan Tian, Liang Wang, and Qi Ding, "A review of deep learning-based image semantic segmentation methods," *Journal of Software*, vol. 30, no. 2, pp. 440–468, 2019.

[14] Shun Zhang, Yihong Gong, and Jinjun Wang, "Development of deep convolutional neural networks and their applications in computer vision," *Journal of Computer Research and Development*, vol. 42, no. 3, pp. 453–482, 2019.

[15] Xudong Li, Mao Ye, and Tao Li, "A review of object detection based on convolutional neural networks," *Application Research of Computers*, vol. 34, no. 10, pp. 2881–2886, 2891, 2017.

[16] Yandong Li, Zongbo Hao, and Hang Lei, "A survey on convolutional neural networks," *Computer Applications*, vol. 36, no. 9, pp. 2508–2515, 2565, 2016.

[17] Liang Chang, Xiaoming Deng, Mingquan Zhou, *et al.*, "Convolutional neural networks in image understanding," *Acta Automatica Sinica*, vol. 42, no. 9, pp. 1300–1312, 2016.

[18] Hongtao Lu and Qinchuan Zhang, "A review of the application of deep convolutional neural networks in computer vision," *Journal of Data Acquisition and Processing*, vol. 31, no. 01, pp. 1–17, 2016.

[19] Baocai Yin, Wentong Wang, and Lichun Wang, "A review of deep learning research," *Journal of Beijing University of Technology*, vol. 41, no. 01, pp. 48–59, 2015.

[20] Xianchang Chen, *Research on Deep Learning Algorith*ms and Applications Based on Convolutional Neural Networks, Master's thesis, Zhejiang Gongshang University, 2014.SIST,volume 179, First Online: 1 May 2020, pp 117–122.