# A Face Recognition Method based on Lightweight Neural Network and Multi Hash Recognition Degree Weighting

Shasha Wang

***Abstract*—To address issues such as high computation, long operation time and low accuracy in traditional and lightweight face recognition algorithms, we propose a face recognition method. The method incorporates a lightweight neural network and multi-hash recognition degree weighting. Firstly, the front-end feature extraction in the SSD detection network utilizes the improved MobileNet model instead of the VGG model. The pruned SSD model is employed as the detection network, reducing calculation time and preventing overfitting. Secondly, face matching involves weighing the calculated mean hash similarity and perceived hash similarity using a specific weight. The proposed improved method is trained and evaluated on WiderFace, LFW and FDDB datasets. Experimental results demonstrate that the method enhances running speed by approximately 29% on LFW datasets while achieving a high face recognition rate and maintaining detection performance. This research introduces a novel and high-performance approach to the field of face recognition.**

***Index Terms*—visual-servoing; tracking; biomimetic; redundancy; degrees-of-freedom**

## I. Introduction

FACErecognition plays a crucial role in the field of computer vision, with applications spanning community management, information gathering, and bank security [1]. The facial recognition process involves two key stages: face detection, which identifies facial coordinates, and face matching, which assesses the resemblance between distinct facial images. Traditional facial recognition methods often face the challenge of scanning the entire image, resulting in high computational demands and limited practicality [2]. In recent years, the integration of facial feature extraction and classifiers, utilizing Convolutional Neural Networks (CNNs), has significantly enhanced facial recognition technology, ensuring efficient feature extraction [3-6]. Ren et al. introduced Faster-CNN, a detection network that markedly reduced extraction time for potential regions. Subsequently, Liu and colleagues proposed SSD detection networks, offering faster processing speeds. However, these networks still relied on deep VGG networks for feature extraction, leading to increased computational expenses. Hashing algorithms have become a prevalent approach for comparing image similarities in image processing, including facial matching.

In this study, the authors utilize a modified lightweight network model for extracting facial features. They also employ pruning techniques to reduce the operation time of the detection network. Furthermore, rather than depending on a single hash algorithm, the paper suggests employing the Doha recognition weighting method to tackle the problem of low and unstable similarity in face image calculations. Experimental results indicate that the proposed method improves the robustness of face image matching while maintaining high accuracy.

Manuscript received March 16, 2023; revised January 24, 2024.

This research was supported by the project of the Science and Technology Department of Henan Province (222102210246) and Henan Provincial Social Science Planning Project (2023BZH008).

Shasha Wang is a teacher of Xinyang Normal University, Xinyang, Henan, China , 464000 (corresponding author, e-mail: wss1020@126.com).

## II. Face Recognition

### A. Face detection algorithm

The surge in the popularity of deep learning-based face detection algorithms can be attributed to the impressive learning and comprehension abilities of neural networks. Girshick et al. introduced the Region Proposal+CNN (R-CNN) algorithm, which utilizes convolutional neural networks to extract features from candidate images, eliminating the need for manually designed features. Deng et al. introduced RetinaFace, a proficient face detection algorithm capable of identifying faces of various sizes. Lin introduced RetinaNet to address the imbalance between high and low-frequency categories. Within this object detection network, the Focal Loss function assigns greater importance to the low-frequency category while moderating loss values from the high-frequency category throughout the learning process. These face detection algorithms can be classified into two-stage and single-stage methodologies, distinguished by their unique network structures [7]. The two-stage approach primarily involves feature extraction, classification, and regression of candidate frames. Conversely, the single-stage method employs a single level for face classification, probability estimation, and location coordinate regression.

### B. Calculate image similarity using hash algorithm

Hashing algorithms, including Average Hash (AHash), Perceptual Hash (PHash), and Difference Hash (DHash), are commonly used for calculating image similarity [8]. The Average Hash algorithm primarily leverages low-frequency details within the image. Its specific algorithmic steps are outlined as follows [9]:

*Input:* Image A, Image B.

*Output:* Similarity $Sim_a(A, B)$ between images A and B based on the AHash algorithm.

*Step 1:* Resize the images to a size of $n \times n$, resulting in a total of $n^2$ pixels.

*Step 2:* Convert the image $G_a$ to a gray image of size $n \times n$.

*Step 3:* Calculate the average pixel value of the gray image $G_a$ and store it as $p_{avg}$.

*Step 4:* Iterate through each pixel $p_i$ in $G_a$ and compare it with $p_{avg}$. If $p_{avg} \leq p_i$, set the corresponding bit to 1; otherwise, set it to 0. This results in a binary string of $n^2$ bits, which represents the AHash value of the image, denoted as $H_a$.

*Step 5:* Calculate the Hamming distance between the two image hash values to obtain the similarity value $Sim_a(A, B)$.

Using the PHash hash algorithm to calculate the image value involves the application of Discrete Cosine Transform (DCT). The formula for the Discrete Cosine Transform is as follows:

$$F(x) = c(u)\, c(v) \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} f(i,j) \cos(mu) \cos(mv) \quad (1)$$

$$c(u) = \begin{cases} \sqrt{\frac{1}{N}}, u = 0 \\ \sqrt{\frac{2}{N}}, u = 1, 2, \ldots, N-1 \end{cases} \quad (2)$$

Equations (1) and (2) describe the two-dimensional DCT. In the equations, $f(i)$ represents the original signal, $x$ represents the coordinates $(u, v)$, $m$ is given by $\frac{(i+0.5)\pi}{N}$, $F(u)$ represents the coefficients after the DCT transformation, $N$ represents the number of points in the original signal, $c(u)$ represents the compensation coefficient, and the DCT transformation matrix can be orthogonalized.

The specific steps of the PHash algorithm are as follows:

*Input:* image A, image B.

*Output:* PHase based similarity $Sim_p(A, B)$ between images A and B.

*Step1:* $function\,(Img)$.
*Step2:* $Size\,(Img) \leftarrow N \times N$.
*Step3:* $Img \leftarrow 256$ Step graying.
*Step4:* $T_{Img} \leftarrow DCT\,(Img)$.
*Step5:* $T_{Img} \leftarrow T_{Img}\,(\nwarrow n \times n)$.
*Step6:* $u \leftarrow Avg\,(T_{Img})$.
*Step7:* PHash = " ".
*Step8:* **for** $i = 1 : N$.
*Step9:* **for** $j = 1 : N$.
*Step10:* **if** $T_{Img}\,(i, j) \geq u$.
*Step11:* $PHash+ =$ "1".
*Step12:* **else** $PHash+ =$ "0".
*Step13:* $PHash_A \leftarrow function\,(A)$.
*Step14:* $PHash_B \leftarrow function\,(B)$.
*Step15:* $Sim_p(A, B) \leftarrow \frac{n^2 - Dis(PHash_A, PHash_B)}{n^2}$.

Here, $DCT(Img)$ represents the DCT transformation of the image $Img$. $T_{Img}(\nwarrow n \times n)$ represents the top-left $n \times n$ low-frequency part of the matrix $T_{Img}$, where $n \leq N$. $Dis(PHash_A, PHash_B)$ represents the Hamming distance between the PHash values of images A and B.

## III. Improved MobileNet Model

In this paper, the computationally intensive VGG_SSD300 model is alleviated by adopting the more resource-efficient MobileNet model. This substitution involves replacing fully connected layers with convolutional layers, resulting in a significant reduction in computational demands. The adapted model integrates the SSD framework with a modified input size for the purpose of face frame position regression. The input face image dimensions for the enhanced model are configured at 300 × 300 to align with the specifications of the SSD300 network.

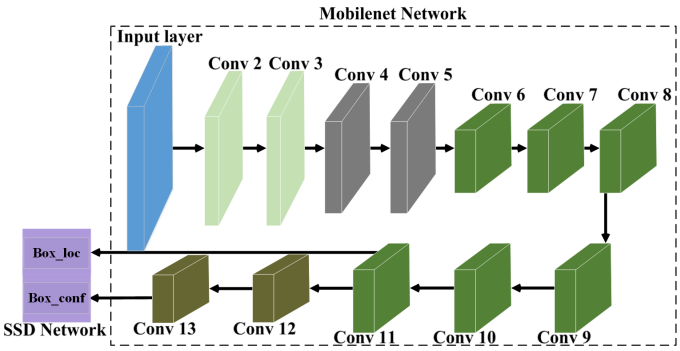


Fig. 1. Improved MobileNet_SSD face detection model. The figure shows the overall framework of the improved MobileNet_SSD face detection model. Conv1 to Conv13 represent the various convolutional layers of MobileNet. Box_loc and Box_conf indicate the classification and regression of prediction boxes generated in Conv11 and Conv13 within the SSD network. The 11th and 13th convolutional layers produce prediction boxes with different heights, namely 4 and 6 boxes, respectively.

The modified network model utilizes multi-group deep decomposition convolutional layers and pointwise convolutional layers to extract features from the input image. These extracted feature images, denoted as "FEATURE," undergo further processing through convolutional layers and fully connected layers. To ensure precise detection, the VGG network incorporates a convolutional layer following the fully connected layer to extract crucial facial feature information. However, for face recognition, the necessity for deep facial feature extraction is not as stringent as in other target detection tasks. Instead, the primary focus is on ascertaining whether the detected image contains a face. Consequently, the proposed enhanced model streamlines the SSD network by singling out two layers from the feature extraction network specifically for face target detection. Fig. 1 presents the improved MobileNet_SSD face detection model. Conv1 to Conv13 represent the various convolutional layers of MobileNet. Box_loc and Box_conf indicate the classification and regression of prediction boxes generated in Conv11 and Conv13 within the SSD network. The model's loss function is obtained by weighting the position $L_{loc}$ and the classification loss $L_{A-S}$. The proposed model in this paper extracts feature maps from Conv11 and Conv13 in the figure, inputting them into the SSD network for classification and regression. Both layers generate face prediction boxes with different aspect ratios, and the number of face prediction boxes differs between them. The former has fewer boxes than the latter, with 4 and 6 boxes, respectively.

To improve the accuracy of face detection, the proposed model adopts an A-Softmax classifier instead of the SoftMax classifier. Prior to this enhancement, the loss function for SoftMax is defined as follows:

$$L = \frac{1}{N} \sum_i L_i = \frac{1}{N} \sum_i -\log \left( \frac{e^{f_{yi}}}{\sum_j e^{f_j}} \right) \quad (3)$$

Among them, $f_j$ represents the $j$-th element of the class vector ($j \in [1, K]$) and $N$ represents the number of training samples. In a convolutional neural network, $f$ is typically represented as the output of a fully connected layer $W$, where $f_j = W_j^T x_i + b_j$ and $f_{yi} = W_{yi}^T x_i + b_{yi}$. Here, $x_i$, $W_j$, and $W_{yi}$ represent the $i$-th training sample, the $j$-th column

of the $W$ matrix, and the $y_i$-th column of the $W$ matrix, respectively. Equation (3) can be expressed as:

$$\begin{aligned} L_i &= -\log\left(\frac{\exp\left(W_{yi}^T x_i + b_{yi}\right)}{\sum_j \exp\left(W_j^T x_i + b_j\right)}\right) \\ &= -\log\left(\frac{\exp\left(\parallel W_{yi} \parallel \parallel x_i \parallel \cos\left(\theta_{yi}, i\right) + b_{yi}\right)}{\sum_j \exp\left(\parallel W_j \parallel \parallel x_i \parallel \cos\left(\theta_j, i\right) + b_j\right)}\right) \end{aligned} \tag{4}$$

In Equation (4), $\theta$ represents the angle between the vectors $x_i$ and $W_j$. Assuming that $\parallel W_j \parallel = 1$ and the bias term for each iteration is $\forall j = 0$, the loss function can be defined as:

$$L = \frac{1}{N}\sum_i -\log\left(\frac{\exp\left(\parallel x_i \parallel \cos\left(\theta_{yi}, i\right)\right)}{\sum_j \exp\left(\parallel x_i \parallel \cos\left(\theta_j, i\right)\right)}\right) \tag{5}$$

While SoftMax [10] is capable of capturing angular boundary features, its effectiveness in recognizing these features is suboptimal. A-SoftMax introduces a novel approach to consolidate angular boundaries by utilizing decision boundaries. This can be formulated as follows:

$$L_{A-S} = \frac{1}{N}\sum_i -\log\left(\frac{y}{y + \sum\limits_{j \neq y_i} \alpha(x)}\right) \tag{6}$$

where $y = \exp\left(\parallel x_i \parallel \varphi\left(\theta_{yi}, i\right)\right)$, $\alpha(x) = \exp\left(\parallel x_i \parallel \varphi\left(\theta_j, i\right)\right)$, $\varphi\left(\theta_{yi}, i\right) = (-1)^k \cos\left(m\theta_{yi}, i\right) - 2k$, $\theta_{yi,i} \in [k\pi/m, (k+1)\pi/m]$, and $k \in [0, m-1]$ , $m$ is an integer greater than or equal to 1, controlling the size of the corner edge.

The loss function of the proposed model in this paper is a combination of the position loss $L_{loc}$ and the classification loss $L_{A-S}$:

$$L\left(x, c, l, g\right) = \frac{1}{N}\left(L_{A-s}\left(x, c\right) + \alpha L_{loc}\left(x, l, g\right)\right) \tag{7}$$

Among the variables used in the equation, $N$ represents the total number of matched positive samples. If $N$ is equal to 0, then the loss $L$ is set to 0. The variables $x$ and $c$ represent the quantity of indices and the confidence degree of classification, respectively. The variables $l$ and $g$ represent the predicted box and the ground truth box, respectively. The variable $\alpha$ represents the weight of the position loss. The position loss $L_{loc}$ is calculated using the Smooth $L1$ loss between the predicted box and the ground truth box, and it is defined as follows:

$$smooth_{L1}\left(x\right) = \begin{cases} 0.5x^2, |x| < 1 \\ |x| - 0.5, otherwise \end{cases} \tag{8}$$

## IV. Face Matching is Performed By Dohahi Similarity Weighting

In face recognition, face matching is equally important as face detection. Image similarity indices are commonly used to determine whether two images depict the same person's face. While using hashing algorithms for single storage can calculate face similarity, it often falls short in meeting real-world demands due to the adaptable nature of face images. To overcome this constraint, this paper proposes a technique to assign a specific weight to the computed face image similarity. This weighted approach aims to obtain a more accurate and reliable face image similarity value. The similarity values calculated by the AHash algorithm ($Sim_a\left(A, B\right)$) and the PHash algorithm ($Sim_p\left(A, B\right)$) range between 0 and 1, with higher values indicating higher similarity between images. Traditionally, the determination of whether two images are similar is made by comparing the algorithm's results with a threshold. However, relying solely on this single comparison can lead to significant errors in judging image similarity. To address this issue, the paper proposes an error-based threshold method for image similarity evaluation.

By incorporating the weighted similarity values and the error-based threshold approach, the proposed algorithm aims to provide more accurate and robust judgments of image similarity in face matching tasks. The specific calculation formula is:

$$f(f_1, f_2) = \lambda f_1 + \beta f_2, 0 < \lambda < 1, 0 < \beta < 1 \tag{9}$$

Among, $\lambda$, $\beta$ are the weight coefficient, $f_1$ is the mean hash similarity, $f_2$ is the perceived hash similarity, and $f(f_1, f_2)$ is the final face image similarity value.

The specific process of the multi-hash similarity weighting algorithm for determining the similarity or dissimilarity between two face images, given a similarity threshold $\alpha$ and an error $\delta$, can be outlined as follows:

*Input:* image A, B.
*Output:* similar or dissimilar.
*Step1:* $Sim_a\left(A, B\right) \leftarrow AHash_A, AHash_B \leftarrow A, B.$
*Step2:* **if** $Sim_a\left(A, B\right) \leq \alpha.$
*Step3:* **Output** similarity.
*Step4:* **else if** $\alpha < Sim_a\left(A, B\right) \leq \alpha + \delta.$
*Step5:* $Sim_p\left(A, B\right) \leftarrow PHash_A$, $PHash_B \leftarrow A, B.$
*Step6:* **if** $Sim_p\left(A, B\right) \leq \alpha.$
*Step7:* **Output** similarity.
*Step8:* **else Output** dissimilarity.

The algorithm starts by calculating the similarity between the two images using the AHash algorithm. If the similarity is below or equal to the threshold $\alpha$, the algorithm directly outputs similarity. However, if the similarity is within the range $\alpha < Sim_a\left(A, B\right) \leq \alpha + \delta$, the algorithm calculates the similarity using the PHash algorithm. If the PHash similarity is below or equal to the threshold $\alpha$, the algorithm outputs similarity. Otherwise, it outputs dissimilarity.

This approach allows for a flexible determination of similarity, taking into account different hashing algorithms and their respective similarity values. By incorporating both the similarity threshold $\alpha$ and the error $\delta$, the algorithm provides a more nuanced judgment of the similarity between face images, distinguishing between similar and dissimilar cases based on the calculated similarities.

## V. Experimental Results

### A. Experimental setup

The experiments detailed in this paper were conducted on an Ubuntu system, which served as the operating platform for executing the tests and performing the assessments.

Regarding the network training parameter configurations, the learning policy is customized to meet the experiment's specific requirements. More precisely, the initial learning

rate is set at 0.001, determining the size of steps taken by the model to adjust its internal parameters during training. Additionally, the weight decay is set to 0.0004, controlling the level of regularization applied to the network's weights to guard against overfitting.

The proposed method's performance is evaluated using both Convolutional Neural Networks (CNN) and MobileNet architectures. These architectural choices are highly regarded in the realm of computer vision tasks, particularly in tasks such as face detection and recognition, due to their effectiveness in extracting significant features from images.

The assessment is conducted on three established benchmark datasets: LFW (Labeled Faces in the Wild) [11], WIDER Face [12], and FDDB (Face Detection Data Set and Benchmark) [13]. These datasets hold significant recognition in both face detection and face recognition domains, providing standardized metrics for evaluating various algorithms and methodologies. By subjecting the proposed method to tests on these datasets, researchers can assess its precision, resilience, and ability to generalize across different scenarios.

### *B. Data set*

The LFW dataset is widely used as a benchmark in the field of face recognition. It comprises 5,749 images, each tagged with a name and a distinct identifier, totaling 13,233 faces. These images showcase celebrities sourced from the internet. This dataset presents a diverse collection of facial images, making it a valuable resource for evaluating the performance of face recognition algorithms.

The WIDER Face dataset stands as another widely used resource for face detection and recognition. It encompasses 393,703 annotated face images within a total of 32,203 images. This dataset incorporates a range of environmental variables such as occlusion, blurring, and multi-scale factors. Additionally, it features 61 intricate scenes, introducing even more demanding scenarios for face annotation and assessment. The WIDER Face dataset is acclaimed for its large-scale and intricate scenes, making it well-suited for the evaluation of face detection algorithms. Fig. 2 in the paper provides a partial glimpse into the WIDER Face dataset, offering insight into its intricate and diverse nature.

The FDDB dataset is specially curated for face detection tasks, featuring images captured in natural settings with diverse face poses, resolutions, shadows, and varying degrees of focus. It encompasses both color and grayscale images, totaling 2,845 face images with 5,171 labeled face regions. The face regions in the FDDB dataset are annotated with precise face coordinates and labels. Serving as a comprehensive benchmark, the FDDB dataset is invaluable for evaluating face detection algorithms. Fig. 3 in the paper showcases a selection from the FDDB dataset, highlighting the range of face poses and conditions captured within the dataset.

### *C. The performance evaluation*

In the paper, Fig. 4 presents a comparative analysis of the accuracy between MobileNet_SSD, CNN [14], and MobileNet using the LFW dataset. The graph illustrates how the accuracy of each model evolves with increasing iterations during the training process. It is evident from the graph that as the number of iterations grows, all three models

Fig. 2. WIDER Face data set sample example.

Fig. 3. FDDB data set sample example.

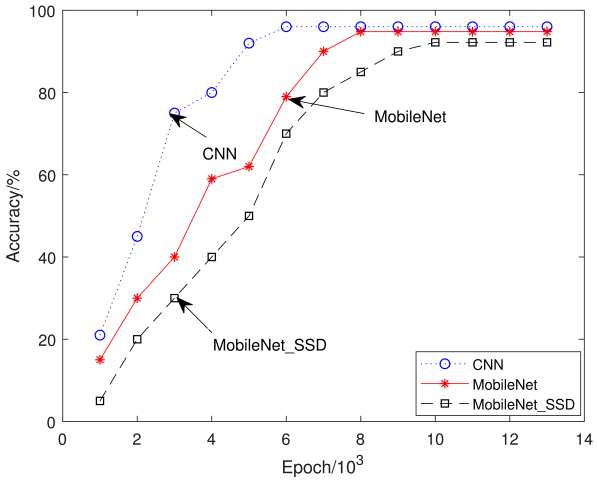


Fig. 4. Comparison of accuracy of the three models. The figure illustrates the accuracy trend of the three models on the LFW data set as the number of iterations increases.

tend to reach a point of convergence. Among these methods, CNN achieves the earliest convergence, attaining a consistent accuracy of 96% between 4,000 and 5,000 iterations. The highest accuracy rate among the three methods, once stabilized, belongs to CNN. MobileNet_SSD and the enhanced MobileNet method take approximately 8,000 and 10,000 iterations, respectively, to converge. The enhanced method achieves a convergence accuracy of around 94.8%, while the accuracy of the improved MobileNet model is 92.2% upon convergence. These findings indicate the viability of the pro-

TABLE I
THE COMPARISON WAS MADE BETWEEN THE THREE MODELS IN TERMS OF THE NUMBER OF PARAMETERS AND RECOGNITION ACCURACY ON THE LFW DATA SET.

| The network structure | Parameter quantity | Accuracy |
|---|---|---|
| CNN | 25000000 | 96% |
| MobileNet | 3600000 | 92.2% |
| MobileNet_ SSD | 3000000 | 94.8% |

TABLE II
COMPARISON OF RUNNING SPEED BETWEEN THE TWO MODELS.(MS)

| model | Min | Max | Avg |
|---|---|---|---|
| MobileNet | 136.31 | 147.63 | 141.97 |
| MobileNet_ SSD | 106.63 | 119.24 | 112.93 |

posed approach, demonstrating promising results. In contrast to the conventional CNN approach, both MobileNet_SSD and the adapted MobileNet method exhibit a decline in accuracy. Nonetheless, MobileNet_SSD holds an edge in computational speed and reduced network parameters. While the accuracy of the MobileNet_SSD model surpasses that of the adapted MobileNet model by two percentage points, it doesn't reach the level of the traditional CNN method.

Table I provides a comparison of parameter count and recognition accuracy between MobileNet_SSD, CNN, and MobileNet models using the LFW dataset. The objective is to assess the proposed model's performance in terms of reducing network parameters and training time. The findings indicate that the proposed model successfully decreases parameters while enhancing accuracy when compared to the original MobileNet.

Furthermore, we conducted tests to compare the operational speed of both the original MobileNet model and the enhanced version. As indicated in Table II, the outcomes affirm that the proposed model significantly decreases the network's training duration and boosts the operational velocity of the initial model by approximately 29%.

Fig. 5 and Fig. 6 showcase the performance of the MobileNet_SSD model on selected face images from the WIDER Face and FDDB data sets, respectively. These images present a range of difficulties including compactness, pose variations, and occlusions. For each identified face frame in the images, a confidence score for face detection is provided, shown in the upper right corner. The visuals demonstrate that the suggested MobileNet_SSD model excels at identifying faces even in demanding situations involving occlusions and crowded spaces, underscoring its robust detection capabilities.

To assess the detection performance of our proposed model, we conducted a comparative analysis with several established algorithms, specifically, MTCNN, FacenessNet, CNN, and MobileNet_SSD utilizing the FDDB dataset. Receiver Operating Characteristic (ROC) curves were computed and depicted, as illustrated in Fig. 7. The ROC curve delineates the balance between True Positive Rate (TPR) and False Positive Rate (FPR). A higher TPR signifies a greater portion of accurately identified positive samples, while a lower FPR indicates fewer incorrectly identified negative samples. The Area Under the ROC Curve (AUC) serves as a metric for the overall performance of each approach,

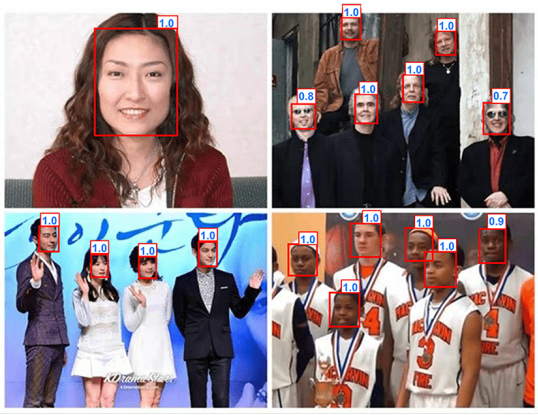

Fig. 5. Face detection results on the WIDER Face data set.

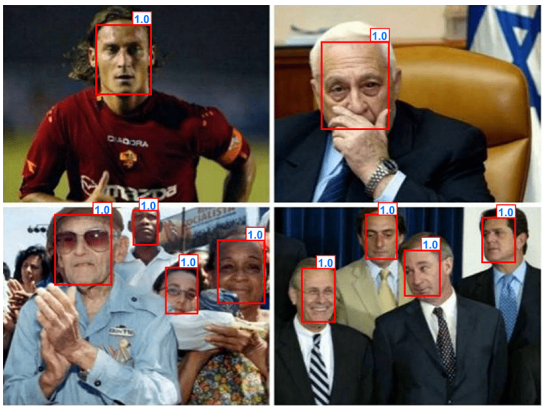

Fig. 6. Face detection results on the FDDB data set.

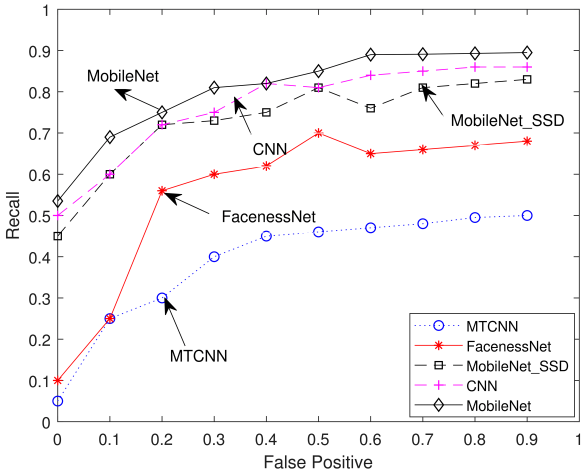


Fig. 7. The ROC curve comparison diagram. The figure illustrates the test performance of the proposed model and a comparison method on the FDDB data set.

with a higher value (closer to 1) denoting superior detection accuracy and resilience. From the graph, it is evident that our proposed model achieves a TPR of over 92% on the FDDB dataset, demonstrating suitability for various real-world scenarios. Upon comparing the methods based on ROC curves, it is apparent that the MobileNet method exhibits a narrower feature distribution and weaker randomness compared to the other approaches. Nonetheless, our proposed MobileNet_SSD displays a reasonably moderate AUC value,

indicating its proficiency in accurately distinguishing face regions from non-face regions. In terms of TPR, our proposed MobileNet_SSD achieves an approximate value of 0.535 at an FPR of 0. This further underscores the high detection accuracy of our proposed method. While its AUC value may not be the highest among all methods, it surpasses that of the modified MobileNet algorithm.

## VI. Conclusions

In this paper, we introduce a face recognition method that leverages lightweight neural networks and the dohadic weighting technique. Our approach incorporates pruned SSD with a weighted mean hash similarity value and perceptual hash similarity value in the detection network. Through multiple experiments, we demonstrate that the proposed method achieves an average face recognition accuracy of approximately 95% and an average recognition time of about 112 ms. However, it is important to acknowledge some limitations of our proposed approach, particularly when dealing with low-definition images, as they can lead to reduced recognition performance. Future work should focus on optimizing the proposed model to address this challenge and further improve the overall face recognition rate.

## References


[1] Lei Cao, Qiang Wang, Runjia Shi and Zhongjin Jiang, "Method for vehicle target detection on SAR image based on improved RPN in Faster-RCNN," *Journal of Southeast University (English Edition)*, vol. 51, no. 1, pp. 87-91, 2021.

[2] Jiesheng Wang, Yanlang Ruan and Bowen Zheng, "Face recognition method based on improved gabor wavelet transform algorithm," *IAENG Internaitonal Journal of Computer Science*, vol. 46, no. 1, pp. 12-24, 2019.

[3] Shasha Wang ,Daohua Liu ,Zhipeng Yang , Chen Feng and Ruiling Yao ," A convolutional neural network image classification based on extreme learning machine," *IAENG Internaitonal Journal of Computer Science*, vol.48, no.3, pp.799-803, 2021.

[4] Pengfei Ke, Maoguo Cai and Tao Wu,"Face recognition algorithm based on improved convolutional neural network and ensemble learning," *Computer Engineering*, vol.46, no.2, pp. 262-267, 2020.

[5] Yong Soon Tan, Kian Ming Lim, Connie Tee, Chin Poo Lee, and Cheng Yaw Low, "Convolutional neural network with spatial pyramid pooling for hand gesture recognition," *Neural Computing and Applications*, vol. 33, no. 10, pp. 5339-5351, 2020.

[6] Kuan Li, Yi Jin , Muhammad Waqar, Ruize Han and Jiongwei Chen, "Facial expression recognition with convolutional neural networks via a new face cropping and rotation strategy," *The Visual Computer*, vol. 36, no. 2, pp. 391-404, 2020.

[7] Xiaoyan Zhou, Ke Wang and Lingyan Li, "Overview of target detection algorithm based on deep learning," *Electronic Measurement Technology*, vol.40, no.11, pp.89-93, 2017.

[8] Linsheng Ma and Yan Zhao , "Image hashing algorithm based on QBFM moments and three-dimensional structure ," *Application Research of Computers*, vol. 39, no. 3, pp. 949-955, 2022.

[9] Liang Deng, Genglin Xu, Mengjie Li and Zangjin Chen , "Fast face recognition based on deep learning and multiple hash similarity weighting ," *Computer Science*, vol.47, no.9, pp.163-168, 2020.

[10] Qiang Zhang, Jibin Yang, Xiongwei Zhang and Tieyong Cao , "CS-Softmax: A cosine similarity-based softmax loss function ," *Journal of Computer Research and Development*, vol.59, no.4, pp.936-949, 2022.

[11] S Tabaie, F Rza Aria, B Flipo and M Jahazi , "Dissimilar linear friction welding of selective laser melted Inconel 718 to forged Ni-based superalloy AD730 (TM): Evolution of strengthening phases ," *Journal of Materials Science Technology*, vol.96, no.1, pp.248-261, 2022.

[12] Ismail Oztel, Go Yolcu and Cemil Oz , "Performance comparison of transfer learning and training from scratch approaches for deep facial expression recognition ," in *the Proceedings of 2019 4th International Conference on Computer Science and Engineering*, IEEE, 2019, pp. 1-6.

[13] Feng Ye, Xingwen Zhao, Eenlai Gong and Lijun Hang , "Tiny face detection based on deep learning inreal-times scenes ," *Computer Engineering and Applications*, vol. 55, no. 12, pp. 162-168, 2019.

[14] Jiangyu Lan, Yinggang Xie, Hui Wang and Guangjun Liu , "A face recognition system based on improved convolutional neural network ," in *the Proceedings of 2019 2nd International Conference on Algorithms, Computing and Artificial Intelligence*, ACM, 2019, pp.244-249.

[15] Jingze Xu, Zuohong Wu, Yan Xu and Jianhang Zeng , "Face recognition based on PCA, LDA and SVM algorithms ," *Computer Engineering and Applications*, vol.55, no.18, pp.34-37, 2019.