Journal of Data Acquisition and Processing
ISSN: 1004-9037
https://sjcjycl.cn/
DOI: 10.5281/zenodo.776907

AN EFFICIENT FRAMEWORK FOR FACE OCCLUSION RECOGNITION USING DEEP CONVOLUTIONAL NEURAL NETWORK TECHNIQUE

Shashidhar V[1] , Dr. R.Balakrishna[2]
[1]Research Scholar, Department of CSE, Rajarajeshwari College of Engineering, Bengaluru.
2 Research Supervisor, Department of CSE, Rajarajeshwari College of Engineering, Bengaluru.
[1]shashigowd@gmail.com, [2]rayankibala@gmail.com

Abstract: Face Recognition has become a major problem in order to identify the theft over surveillance systems**,** especially the occluded Face recognition technology has gain more importance now a days. there are lot of challenges faced by existing surveillance systems available even though various research works has been carried out still many of the existing machine learning techniques fail in identifying the occluded faces. By employing deep learning-based technique, the performance of face recognition tasks has been greatly boosted. The majority of cutting-edge methods still struggle with the verification and discriminating of faces with occlusions. In light of this, this research proposes a unique convolutional neural network that was created specifically for comparing occluded and non-occluded faces for the same identity utilizing CNN and BiLSTM approaches. Based on the architecture of multiple network convolutional neural networks, it could learn both the shared and distinctive properties. The training and testing of the proposed convolutional neural network incorporated the recently disclosed joint loss function and the accompanying alternating minimization strategy. The proposed deep convolutional neural network approach outperforms the state-of-the-art face identification algorithms by 10-15% in terms of several performance characteristics, according to experimental results on the publically accessible datasets (LFW 99.73%, YTF 97.30%, and CACD 99.12%).

Keywords: *Occluded Face Recognition, Deep Neural Network Technique, CNN, BILSTM, Image Processing.*

1. Introduction

In a number of industries, including finance [1, 2], public security [3, 4], and education [5, 6], face recognition has emerged as the go-to biometric technique for personal authentication and identification. Since the early 1990s, a wide range of computer vision-based methods have been suggested that have the ability to extract the low-dimensional representation under particular priors on the characteristics in facial images. Unfortunately, the performance of these methods would be constrained and the assumptions made might not hold true in real-world situations.
Convolutional neural networks (CNN), in particular, have recently gained widespread acceptance as the most cutting-edge method for face detection and verification [7, 8]. As an example, Rajeev et al. [9] and Schroff et al. [10] stated that their suggested method obtained the accuracy of 99.78% and 99.63% on the facial dataset of Labeled Faces in the Wild (LFW) [11], respectively. CNNs have demonstrated great performance in a variety of face recognition

tasks. Yet, it is still difficult for them to achieve adequate accuracy on faces that fluctuate in position, illumination, and occlusion—facial occlusion in particular has traditionally been seen as a very difficult job. One explanation for this phenomena could be the data imbalance in the prevalent facial databases. Despite the fact that the majority of face recognition training datasets contain a sizable number of identities, they still lack challenging facial photos with partial occlusions like sunglasses, hats, and hair. It makes sense to integrate more occluded facial images in the CNN framework's training process as a remedy to this issue.

On the other hand, the loss function might be skewed towards the data distribution, which could negatively impact the training of CNN-based face verification systems and result in subpar performance. For instance, softmax loss would ignore the faces with occlusion by increasing the conditional probability across the board, despite not being especially created for complex data. Many loss functions and various limits on the conventional loss functions have been presented [9, 12–14] to address this issue. We propose an unique CNN architecture trained by manually collecting 6,178 facial picture pairs from 560 different identities with occlusion, keeping in mind the analyses indicated above. Our suggested CNN uses the standard multi-path framework, and the previously introduced maxout operator is used to create its convolutional layers. Following initialization, the CNN architecture's layers—including the convolutional layers—are then tweaked using the gathered facial images.

The unique information from each input image and the shared identity information between one face and its obscured counterpart are respectively contained in two different feature vectors that make up the output layer. Simultaneously, one newly proposed mutual information constraint and the presented maxout operator are integrated to lower the dimensionality of the parameters and eliminate the potential of overfitting that might emerge in small dataset. The objective function for the suggested CNN model could be iteratively tuned for the diverse images throughout both the training and testing processes by adopting an alternative minimization strategy.

We have two methods of evaluation: publicly and privately. Experimental findings show that our mutual information limited CNN framework outperforms cutting-edge face verification methods and learns occlusion-invariant representation.

The temporal features between various images of the same individual are not taken into consideration by the traditional deep learning-based occluded face recognition algorithm. The recognition efficiency is therefore poor. This study examines the geometric and temporal characteristics of several face photographs. It is suggested to use CNN+BILSTM, a convolution neural network occluded face recognition approach based on BiLSTM.

This paper is structured as follows. The Related work is described in Section 2. Section 3 describes the proposed methodology and provides some background information about how the deep convolutional neural network is used to identify the occluded images. The Result and Discussion is presented in Section 4, along with a description of our experimental findings. The paper is concluded in Section 5.

Contributions of the research work are:

The study makes several contributions. The study's key contributions are listed as follows:

- This work presents a novel approach for occluded face recognition using Deep

Convolutional neural network which used CNN with loss function and BiLSTM memory features.

- ❖ The proposed model is based on CNNs loss function is added to extract important features of the face with loss part in order to recognize occluded faces.
- ❖ The BiLSTM technique is also incorporated to the proposed model to capture the sequence features of the image and extract the hidden temporal features, so as to further improve the accuracy of feature selection
- ❖ The Loss function, Mutual information function are used to classify the images, thus improving training efficiency and classification accuracy
- ❖ The performance evaluation is carried out using various performance parameters like accuracy, F-measure, Precision, and Recall by comparing with the existing techniques.

2. Literature Review

The depth and width of the developed networks have greatly increased since the creation of the first well-known LeNet-5 convolutional neural network [18–19]. Network models have evolved from the early three to seven layers to the dozens or even hundreds of layers at this point in order to adapt to increasingly difficult image recognition jobs. Convolutional neural networks had a consistent structure in their early versions. After each convolutional layer, people typically link a pooling layer or one or more thick layers. For a very long period, the neural network's ability to develop was constrained by this type of comparable network topology. VGGNet helped people think differently. By stacking different sized convolutional layers, Simonyan et al. improved the network's ability to express itself and shown in trials its superior recognition effect [20].

People now have new perspectives on "depth" thanks to the development of "Network in Network" (NiN). To boost the network's expressiveness, Lin et al. replaced the conventional dense layer with additional 1 1 convolutional layers that connected each layer in the network [21]. Future generations benefited greatly from their method of constructing more intricate networks by placing smaller networks within larger networks. A multilayer neural network can learn combinations of local to global information from images to perform image processing tasks. For instance, the first layer might discover a set of pixels to recognize corners. The second layer might pick up on a corner combination to recognize an item's shape or basic object parts.

By combining contours in this way, the deeper network layer is able to identify more intricate object properties. A multidimensional serial feature extraction module was presented based on the NiN concept. It may enhance the learning capacity of the two dimensions (space and channel) by expanding the depth and width of the neural network. It is a great concept to extract space and channel dimensional features through deeper network structures and connect them in series. The higher the feature complexity that can be represented, the deeper and wider the deep neural network is. Complex models are not always useful, though, because of issues like disappearing gradients, over fitting, a lack of training data, and other issues including real processing power limitations. The data's and the application's properties were reasonable as a result.

The eight-layer convolutional neural network used by AlexNet, which was proposed in 2012, won the image recognition competition that same year by a significant margin. Its results demonstrated that features learned by neural networks were superior to those initially extracted artificially. Convolutional neural networks were then created and became widely used, including VGGNet, Google Net, Residual Net, DenseNet, and others [22–24]. Although the module and network structure of these convolutional neural networks have been changed, feature extraction is still based on a single space dimension.

It is challenging to meet people's needs to extract characteristics in only one dimension in the face of the complexity of picture recognition tasks. Hu et al. therefore started researching the size of image channels in 2017. In order to speed up learning and improve the network's capacity for representation, they created a Squeeze-and-Excitation block, which models each channel's dynamic nonlinear dependence using global information [25]. This work investigates the feature extraction of the channel under the assumption of conventional space feature extraction. The obtained feature maps will be changed in accordance with the learned features of the picture channel in order to suppress the unnecessary features and boost the information-rich features. All of the image's attributes are naturally combined through multidimensional feature extraction. It completely considers how the qualities are related internally, which is more conductive to precise and thorough feature expression.

Deep learning [5] has become very popular in recent years, particularly in the field of facial recognition. It differs from the conventional approaches. The deep learning-based face recognition system is more open to the inclusion of facial features and is highly adaptable to variations in lighting and expression. Deep learning-based facial recognition algorithms that are often used include MTCNN [6], DeepID [7], FaceNet [8], and DeepFace [9].

Using explicit 3D face modelling and piecewise affine transformation applied to frontal faces, DeepFace [9] enhances the face alignment approach. Nevertheless, because the model must be reconstructed with every request, a lot of memory is needed. From the information provided above, it is clear that a convolutional neural network is a popular deep learning technique for face recognition. It has good self-organization and adaptive capabilities and, through learning, can implicitly express a number of face recognition laws. Although it is more versatile, it nevertheless has some shortcomings that reduce its effectiveness.

For face identification, Tao et al. [10] suggested combining CNN with metric learning approaches, which use a multi-Inception structure to extract facial features. Yet it necessitates an excessively large sample of data. Hu et al[11] .'s fusion of the subspace's extracted multi-layer features enhanced the effect of face recognition. But, the operation is sluggish and there are too many settings. L-SoftMax was substituted for SoftMax in MobileNet by Wu and Zhang [6] in order to prevent over fitting and produce better classification outcomes. However, it makes the network deeper and lengthens the running time, which is inefficient. An R-CNN face recognition algorithm, which combines CNN and ResNet, was proposed by Ren and Xue [12]. Unfortunately, the network's size is unwieldy and necessitates the use of numerous datasets.

Conventional facial recognition algorithms frequently need a lot of effort and money to build, and they are challenging to adapt to the actual world. Convolutional neural networks have been extensively utilized in the field of image identification with the resurrection of deep learning [12–17]. Currently, there are two types of anti-occlusion face recognition algorithms:

conventional algorithms and deep learning algorithms. Convolutional neural networks have quickly advanced since the LeNet neural network's outstanding performance on the MNIST handwriting dataset. Subsequently, as convolutional neural networks like AlexNet, Visual Geometry Group Net (VGGNet), GoogLeNet, and others continued to increase the accuracy of the ImageNet identification challenge, people started employing these networks for object detection and picture classification for the first time.

Deep learning is an end-to-end algorithm that does not require the creation of numerous intricate components as do conventional algorithms. Deep learning is a data-driven technique with strong generalizability, especially for various types of occlusions. Deep learning is a constantly developing algorithm that can correct errors in routine applications. This paper designs a deep convolutional neural network that incorporates multidimensional serial feature extraction modules for occluded faces in order to reduce the time costs of the design process of conventional algorithms and significantly increase the recognition accuracy under unrestricted conditions. The multidimensional serial feature extraction module suggested in this work is more resilient for obstructed facial images when compared to conventional approaches.

3. Proposed Methodology

In order to extract the bidirectional time series characteristics of the images in this section, our network is built using the neural network described in figure 1 is structure and enhanced with BiLSTM features. After that, an attention mechanism is included to extract the photos' key details. The deep neural network is created at the end.

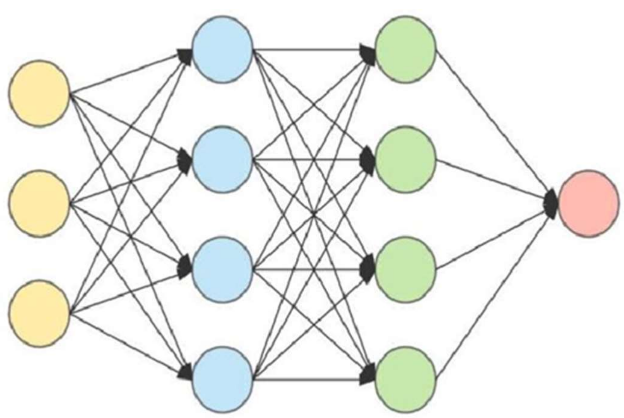


Figure 1: Basic Neural Network Architecture

A neural network is made up of many neurons, where each link is given a weight. These neurons operateconcurrently. The neural network updates its weights during the learning phase in order to forecast the right input and give the desired function of output.

Equation 1 provides a mathematical expression for sigmoid optimization.

$$Sigm(a) = 1/(1 + e^{-a})$$

(1)

Equation 2 provides the mathematical details of the Hyperbolic Tangent (Tanh) optimization technique.

$$tanh(a) = 2/(1 - e^{-2a})$$

(2)

Equation 3 describes how the Rectilinear Unit (Relax) optimization approach operates.

$$r(a) = \max(0)$$

(3)

3.1 Architecture of the Network

We provide a unique CNN-based picture classification method to handle the challenge of obstructed facial verification. The suggested CNN design is similar to the CNN described in [29], for example, they are both multiple networks. However, they differ from one another in terms of size, layer count, and loss functions. Additionally, the CNN in [29] was used to emphasize the small differences between the various bananas' ripening stages, whereas the proposed architecture was primarily aimed to extract the shared information between the occluded part and the non-occluded part of the face images.

First, the publicly accessible facial datasets LFW [11] and VGG Face [30] are used to train the proposed CNN architecture and its associated parameters. The manually gathered 6178 photos, which include the non-occluded faces and the corresponding occluded images, are then used to fine-tune the initialization model. According to Figure 2, 8 convolutional layers with matching maxout operators and 3 fully connected layers were used for each input facial image.

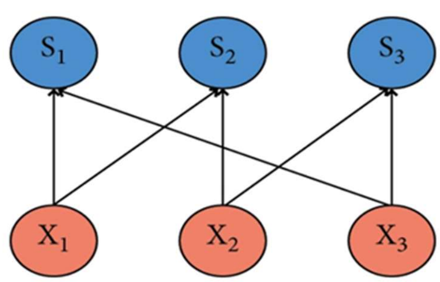


Figure 2: Full Connection of the Network

As illustrated in Figure 3, the network implementation can be broken down into four stages: network building, network training, network validation, and network application. The multidimensional feature serial extraction module, the 1*1 convolutional layers, the batch normalization layer, and the activation function adjustment are the four components that make up the internal architecture of the network. Serial extraction module for multidimensional features. The combination of parallel and serial processing was used in the multidimensional feature extraction module.

There were two sections in the module structure: "space" and "channel." The module's initial section is in charge of separating the spatial information from the image features. This component's design incorporates Google's Inception module. In this section, the occluded faces' features were extracted using five parallel lines. As opposed to the conventional convolution layer, different-sized kernels are used to extract the image information. This allows for the simultaneous extraction of numerous local features and the collection of responses to these features. Four 1*1, 3*3, and 7*7 pooling layer kernels were employed in the design as the extraction parameters for five parallel lines. The VGGNet demonstrates that stacking repeated convolutional blocks can enhance calculation performance while also reducing parameter scale and increasing network depth.

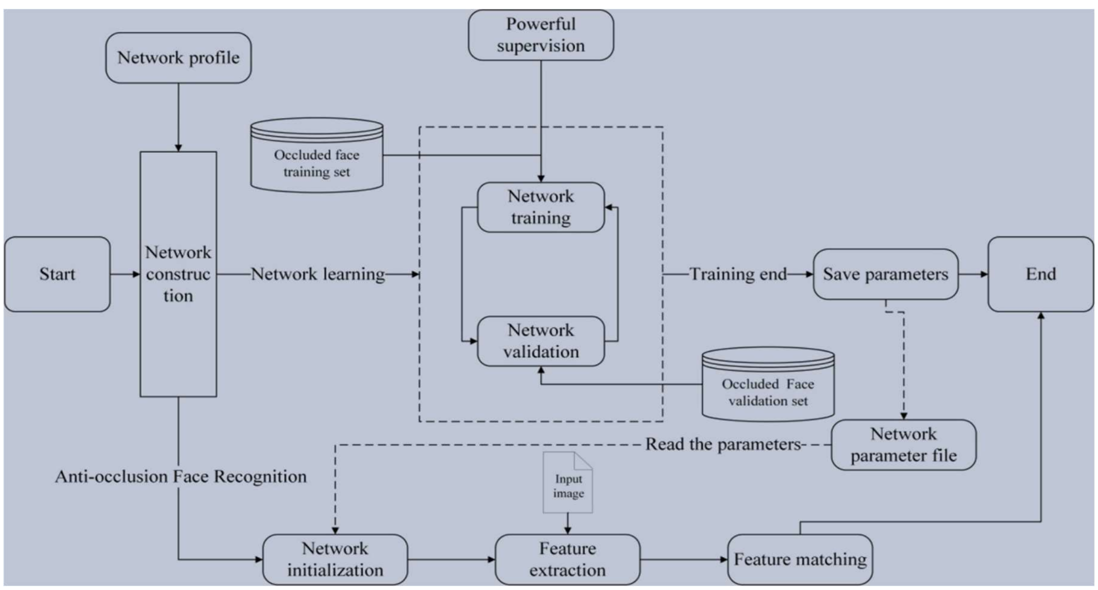


Figure 3: Implementation of Network using proposed Deep Neural Network technique

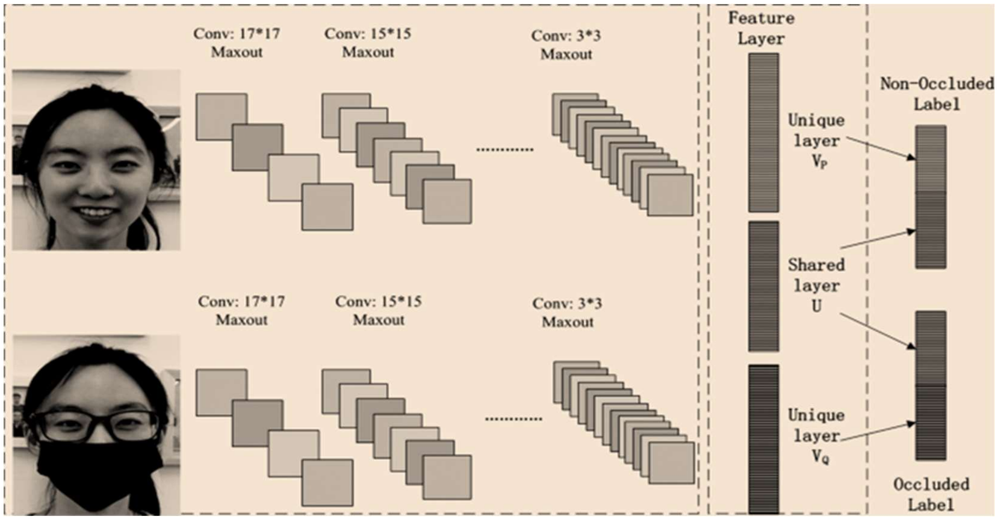


Figure 4: Proposed Deep Neural Network Architecture

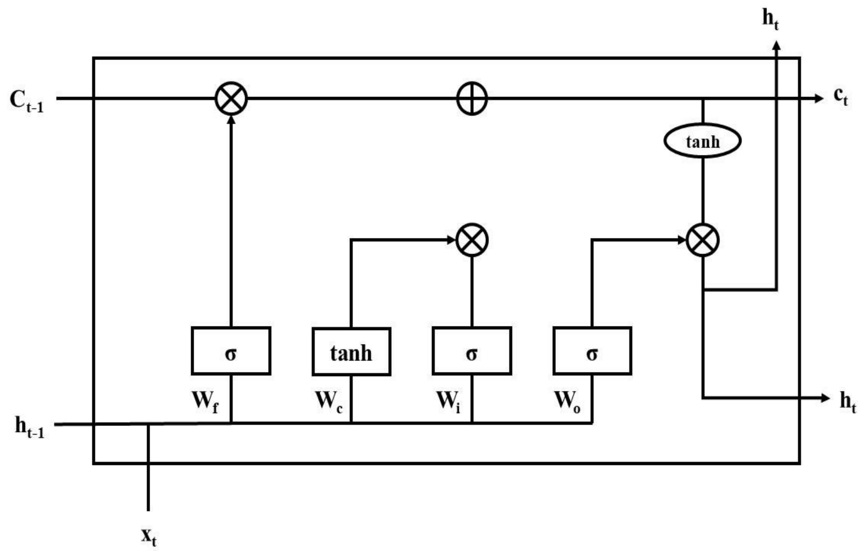
Two different share parameters processed using the neural network technique shown in Figure 4 for both occluded and un-occluded faces. Furthermore, the output feature layer is used to extract both the common features from the input face pair and the distinctive characteristics from each input single image. The following is a list of each CNN channel's specifics.

- Convolutional layer 1. There are 48 kernels (size 17×17, stride of 2) in the first convolutional layer, which is combined with one maxout operator and one max pooling layer.
- Convolutional layer 2. There are 96 kernels (size 15×15, stride of 2) in the second layer, which is combined with one maxout operator and one max pooling layer.
- Convolutional layer 3. There are 128 kernels (size 13×13, stride of 2) in the third layer, which is combined with one maxout operator and one max pooling layer.
- Convolutional layer 4. There are 128 kernels (size 11×11, stride of 2) in the fourth layer, which is combined with one maxout operator.
- Convolutional layer 5. There are 128 kernels (size 9×9, stride of 2) in the fifth layer, which is combined with one maxout operator.
- Convolutional layer 6. There are 128 kernels (size 7×7, stride of 2) in the sixth layer, which is combined with one maxout operator.
- Convolutional layer 7. There are 384 kernels (size 5×5, stride of 2) in the seventh layer, which is combined with one maxout operator.
- Convolutional layer 8. There are 384 kernels (size 3×3, stride of 2) in the eighth layer, which is combined with one maxout operator.

- ❖ Fully connected layer 1. 512 neurons combined with ReLU.
- ❖ Fully connected layer 2. 512 neurons combined with ReLU.
- ❖ Fully connected layer 3. 512 neurons combined with ReLU.

The LSTM may permanently store the important information depending on the cell and forget gate. The classification of arrhythmia signals must take into account both recent and old data. The LSTM model addresses problems with long-term dependency as a result by utilizing the self-feedback strategy of hidden layers [16, 17]. In order to preserve information, the LSTM model employs memory cells and three gates—input, forget, and output gates—which helps to solve the issue of long-term features [18, 19]. Figure 5 depicts the Bi-LSTM model's architecture.

Figure 5: LSTM architecture



The output of the LSTM cell is marked by ht, the previous moment output of the LSTM cell is denoted by h_{t1}, and the LSTM input data is denoted by x_t at time t. This is a calculation:

I As stated in equation, the Wc denotes weight matrix, b_c denotes bias, and c_t signifies the candidate memory cell ct (5).

$$\tilde{c}t = \tanh(Wc.\ [h_{t-1}, x_t] + b_c) \tag{5}$$

ii) The input gate is calculated; it controls the current input data update of the memory cell state value. The bias is denoted by the letters bi, the weight matrix by the letters Wi, and the sigmoid function by the letters, as shown in equation (6).

$$it = \sigma(W_i.\ [h_{t-1}, x_t] + b_i) \tag{6}$$

The LSTM model can easily update, reset, read, and maintain long-term data since it is based on memory cells and control gates. The internal parameter sharing mechanism of the LSTM model controls the output dimensions based on the parameters for the weight matrix dimensions.

The Bi-LSTM model depicted in the image analyses the token sequence both forward and backward using two LSTM units. One LSTM unit processes the token sequence from right to left, while the other does it from left to right. The prior hidden state, h_{t1}, is used as the foundation for computing the hidden unit function, h, of a hidden forward layer at each time step, t. The future hidden state h_{t+1} and the current step x_t at the input are used to compute the hidden unit function h of a hidden backward layer. Concatenating h and h of the forward and backward representations of context to create a long vector.

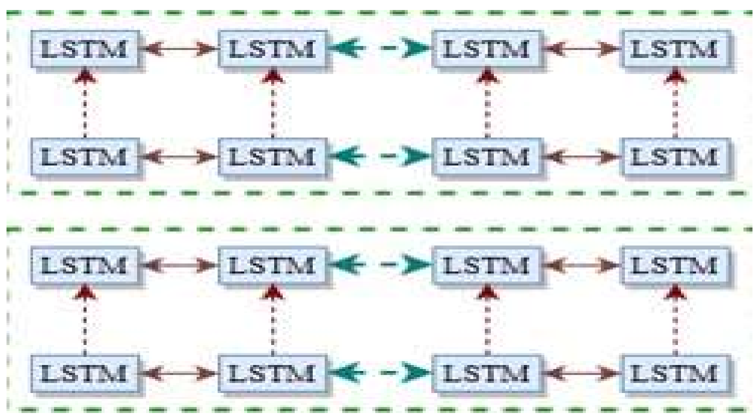


Figure 6: LSTM Model

3.2 Mutual Information using softmax Loss function

Firstly, the widely used softmax loss function is formulated as follows.

$$L_s = \sum_{i=1}^{m} log \frac{e^{x\,(xi+bwi)}}{e^{W\,(xi+b \quad)}} \tag{7}$$

Where Xi bd stands for the characteristic of the jth class of the ith input image. The bias is represented by b Rn, while Wj bd is the jth column of the weights W bdn in the final fully linked layer. And m and n, respectively, stand for the batch size and the number of identities

Let IQ and IP stand in for the occluded face and the unobscured facial image, respectively. The following statement provides a definition of the general feature extraction procedure.

$$X_i = \text{Conv}\ (Ii, \theta i\ )\ (\ i \in \{P,Q\}) \tag{8}$$

Where Conv stands for the proposed CNN's feature extraction function, Xi for the associated output feature, and for the feature mappings the proposed CNN needs to learn during training. The idea behind the suggested solution is that the non-occluded image and its occluded counterpart should have a common component. In order to represent the shared information of the features and the unique feature, we introduce three separate matrices (U, VP, and VQ), which can be expressed as follows.

$$F_i = \frac{F_{share}}{F_{uni}} = \frac{Ui\ Xi}{Vi\ Xi} \tag{9}$$

3.3 Optimization of the Occluded Images

Then, the final objective function of the proposed deep neural technique could be expressed as follows according to the Lagrange multiplier,

$$\mathbf{L}(\mathbf{F}, \mathbf{c}, \boldsymbol{\theta}, \mathbf{U}, \mathbf{V}) = \sum_{i}^{u,v} \mathbf{Softmax}\left(\boldsymbol{a}^{n-k}\mathbf{Fi}, \mathbf{c}, \boldsymbol{\theta}, \mathbf{U}, \mathbf{Vi}\right) + \boldsymbol{\lambda}\mathbf{i}\sum \mathbf{i} \in \{\mathbf{U}, \mathbf{V}\}$$
(10)

where λi denotes the Lagrange multiplier for xi. By using the alternating minimization algorithm and the back-propagation mechanism, the θ, U, and Vi can be iteratively optimized. The gradients of U and Vi can be expressed as:

Algorithm 1: Proposed Deep Neural Network Technique

Input: Dataset with Occluded Images
Output: Identification of Occluded face and real images

1. Input Dataset
2. Split the dataset Sd
3. Split Dtrain into train and validation set to train the algorithm and optimize the parameters
4. Data preprocessing of the dataset
5. Normalization the data by removing the outliers
6. Initialize the proposed deep neural network layer_ Nb
7. Embedding layer to convert categorical features to embedding vectors (Eb)
8. Dense layer to process the numerical features (Dn)
9. BiLSTM layers enhance extracting the important features for predicting the images
10. Build CNN layer. Eq. (1-4)
11. Dropout layers to prevent over fitting
12. Batch normalization layer
    a. As described in Equations 5-6.
13. Mutual function Information
    a. As shown in Eq. (7-9)
14. Optimization of the Occluded Images Described in Eq(10).
15. Fully connected layers
16. λi Calculation for each of the dataset used
17. End for
18. Evaluate the Performance parameters for the proposed model

4. Results and Discussion

In this section, we introduce modules such as datasets and pre-processing, experimental environment and parameter setting, experimental model, experimental results, comparison with other algorithms, and ablation experiment.

4.1. Datasets and Pre-processing

Several tests were run to determine how well our suggested face verification approach performed on various publicly accessible face recognition benchmarks, such as LFW [11], YouTube Faces (YTF) [32], and Cross-Age Celebrity Dataset (CACD) [33]. This section illustrates the analysis and the experimental findings.

4.1 Dataset and pre-processing

The CNN model's training dataset had a substantial impact on how well the corresponding tasks were performed [34]. As a result, numerous face recognition datasets have been provided. The 2007 publication LFW [11] has 13,233 facial pictures from 5749 unique individuals. The most widely used measure for assessing how well deep learning systems perform in unrestricted environments, its accuracy has reached around 100% [9]. Yet, there aren't enough challenging situations and the faces in LFW are primarily frontal without a severe pose or illumination. There are 2.6M and 3.32M faces in VGG-Face [30] and VGG-Face2 [35], respectively, drawn from 2622 and 9131 identities. These two datasets, in contrast to LFW, are private and both include faces with pose-related variants. The largest publicly accessible face recognition dataset is MS-Celeb-1M [36]. There are 100,000 well-known celebrities represented by 10 million facial photos with some annotation noise. The 4.7M faces in MegaFace [37] come from 672,057 different people. Additionally, it offers two subsets of the photographs that can be used to confirm the changes in position and age. IJB-A [38], which only comprises 25,809 faces of 500 distinct people, is regarded as a challenging face recognition database because it was created for combined face detection and recognition tasks and includes both photos and videos of faces with diverse poses.

In addition to the LFW [11] dataset, we gathered 6178 facial pictures, both occluded and unclouded, for training the suggested CNN architecture (examples given in Figure 7(a)(b)(c)). For each identity, around 6 photos are collected. We enlarge the original dataset using data augmentation techniques such as translations (varying from 10 pixels to 100 pixel with a gap of 10 pixels) and vertical and horizontal reflections in order to increase the diversity of the input images and reduce the possibility of potential overfitting in the small-scale facial images. Images are then downsized to 256*256 after the data augmentation process.

(a) **(b)** **(c)**

Figure 7 (a) (b) and (c) Conversion of occluded to face identification using Proposed Deep Neural Network technique

4.2 Training and evaluating

The samples were manually divided into various categories in accordance with the identities. A training dataset is chosen from 50% of the photos, an evaluation dataset is picked from 30% of the images, and a testing dataset is chosen from 20% of the images. The back-propagation method, which initially calculates the minimization of the squared difference between the classification ground truth and the associated output prediction, is used in the training phase to refine the proposed framework. The learning rate begins at 0.01 and the training process requires 105 iterations in total. It is carried out on a high performance GPU and implemented in Tensorflow [39]. It takes barely a fraction of a second for each iteration. The threshold comparison is used for the face verification in the testing phase, and the similarity score is derived with the cosine distance of two output characteristics.

4.3 Experiments on the LFW

We conducted comparison experiments between the state-of-the-art methods and ours on three publicly available datasets in order to assess the performance of the proposed face verification technique. These methods included DeepID3 [7], L2 Softmax [9] (3.7M), FaceNet [10] (200M), VGG-Face [30] (2.6M), Baidu [40] (1.3M), Deep Face [41] (4M), Range Loss [42] (1.5M), and Deep Visage [43] (4. We first carried out the experiments on 6000 pairs of facial photos in LFW, adhering to the methodology "unrestricted with labelled outside data," and the experimental findings are displayed in Table 1.

Table 1: Experimental results of LTW Dataset

Methods	**Images**	**Single loss**	**Accuracy (%)**	**Time (sec)**
DeepID3	–	No	99.47	1.3
L_2 Softmax	3.7M	Yes	99.60	1.5
FaceNet	200M	Yes	99.63	2.6
VGG-Face	2.6M	No	98.95	1.9
Baidu	1.3M	No	99.13	1.8
Deep face	4M	No	97.35	1.2
Proposed Deep Neural Network	2.62M	No	99.73	1.2

Table 2: Performance Parameters

Methods used	**Accuracy**	***F*1-score**	**Recall**	**Precision**
FaceNet	94.8	0.62	0.71	0.49
VGG-Face	91.3	0.91	0.72	0.88
Baidu	92.23	0.88	0.86	0.87

Proposed Deep Neural Network	98.16	0.742	0.798	0.68

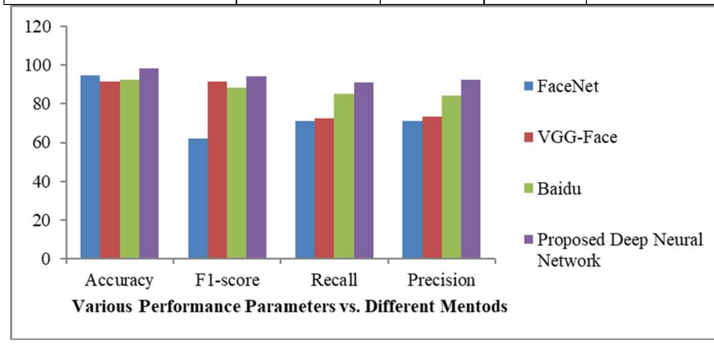


Figure 9: Performance Evaluation of Proposed vs. Existing Methods

Furthermore, to evaluate the influence of different value of λ in Eq. (7), we carried out experiments with our method on the abovementioned datasets, and the experiments are shown in Figure. 10.

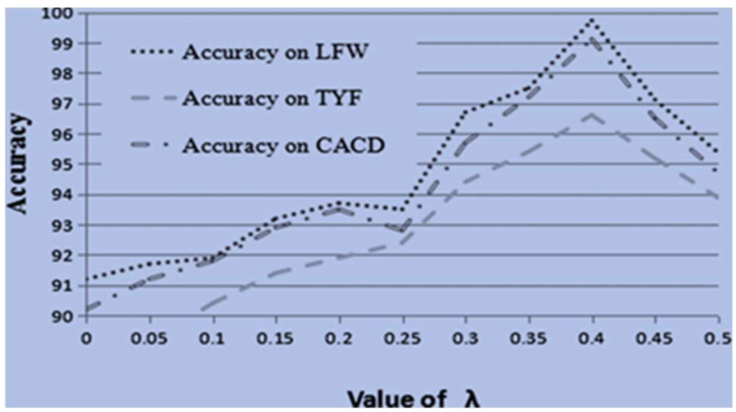


Figure 10: Experiments on the datasets with different λ

As shown in Figure. 10, while the value of λ is greater than 0.40, the accuracy of the proposed method would start to degenerate. It demonstrates that the value of λ should be set to around 0.40.

5. Conclusion and Future Scope

In this paper, a novel loss function with the BiLSTM feature is provided together with a two-channel CNN architecture based on deep neural network techniques. The proposed CNN and BiLSTM architecture uses two parameter-sharing CNN channels to process the non-occluded facial image and the occluded facial image, respectively. Both the shared feature and the unique feature could be retrieved in a feature layer at the network's conclusion. The alternating minimization technique iteratively optimizes the mutual information regularized softmax loss. Using numerous publicly accessible face picture datasets, we conducted comparison experiments between the state-of-the-art approaches and ours to assess the performance of the suggested strategy. The proposed strategy outperforms the most recent methods in accuracy, according to experimental results. This essay makes a number of contributions. The face verification challenge is first implemented using an unique deep neural network technique. The unique loss function is being included into the CNN architecture for the second time, most likely for the first time. We will continue to develop more uses of the proposed CNN architecture in our upcoming works. For instance, we would test the accuracy of the suggested CNN on more realistic images, like as hazy photos. We will keep gathering additional face photos and construct a publicly accessible dataset in order to meet this goal.

REFERENCES

1. D. S. Bartoo, *Financial services innovation: opportunities for transformation through facial recognition and digital wallet patents* (Dissertations & Theses - Gradworks, Ann Arbor, 2013).
2. M. Ketcham, N. Fagfae, *The algorithm for financial transactions on smartphones using two-factor authentication based on passwords and face recognition. International Symposium on Natural Language Processing* (Springer, Cham, New York, 2016).
3. J. Xiao, Research on application of face recognition in area of public security. Comput. Sci. 43(11A) 127–132 (2016).
4. D. Chawla, M. C Trivedi. A comparative study on face detection techniques for security surveillance (SpringerSingapore, 2018), pp. 531–541.
5. Q. Zhao, M. Ye, *The application and implementation of face recognition in authentication system for distance education. International Conference on NETWORKING and Digital Society* (IEEE, Los Alamitos, 2010).
6. D. Yang, A. Alsadoon, P. W. C. Prasad, et al., An emotion recognition model based on facial recognition in virtual learning environment. Procedia Comput. Sci. 125:, 2–10 (2018).
7. Y. Sun, D. Liang, X. Wang, et al., DeepID3: face recognition with very deep neural networks. Comput. Sci. abs/1502.00873: (2015).
8. M. A. Hasnat, J. Bohn, J. Milgram, et al., von Mises-Fisher mixture model-based deep learning: application to face verification. arXiv: 1706.04264 (2017).
9. R. Ranjan, C. D. Castillo, R. Chellappa, L2-constrained softmax loss for discriminative face verification. arXiv preprint arXiv: 1703.09507 (2017).
10. F. Schroff, D. Kalenichenko, J. Philbin, et al., FaceNet: a unified embedding for face recognition and clustering[J]. Computer vision and pattern recognition, 815–823 (2015).
11. L. J. Karam, T. Zhu, Quality labeled faces in the wild (QLFW): a database for studying face recognition in real-world environments[J]. Proceedings of SPIE - The International Society for Optical Engineering. 9394:93940B-93940B-10 (2015).
12. J. Deng, Y. Zhou, S. Zafeiriou, *Marginal loss for deep face recognition. IEEE Conference on Computer Vision and Pattern Recognition Workshops* (IEEE Computer Society, Los Alamitos, 2017).
13. W. Liu, Y. Wen, Z. Yu, et al., Large-margin softmax loss for convolutional neural networks. International Conference on International Conference on Machine Learning, 507–516 (2016).
14. J. Deng, J. Guo, S. Zafeiriou, ArcFace: additive angular margin loss for deep face recognition. arXiv: 1801.07698 (2018).
15. A. Krizhevsky, I. Sutskever, Hinton G.E., in *NIPS*. Imagenet classification with deep convolutional neural networks (MIT PressCambridge, 2012), pp. 1097–1105.

16. K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv: 1409.1556, 1–9 (2014).
17. C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, A. Rabinovich, et al., Going deeper with convolutions. Cvpr, 1–9 (2015).
18. K. He, X. Zhang, S. Ren, J. Sun, in *CVPR*. Deep residual learning for image recognition (IEEELos Alamitos, 2016), pp. 770–778.
19. M. Kan, S. Shan, X. Chen, in *CVPR*. Multi-view deep network for cross-view classification (IEEELos Alamitos, 2016), pp. 4847–4855.
20. I. Masi, S. Rawls, G. Medioni, P. Natarajan, in *CVPR*. Pose-aware face recognition in the wild (IEEELos Alamitos, 2016), pp. 4838–4846.
21. Y. Sun, X. Wang, Tang X., in *ICCV*. Hybrid deep learning for face verification (IEEELos Alamitos, 2013), pp. 1489–1496.
22. Y. Sun, X. Wang, Tang X., in *CVPR*. Deep learning face representation from predicting 10,000 classes (IEEELos Alamitos, 2014), pp. 1891–1898.
23. R. Ranjan, S. Sankaranarayanan, C. D. Castillo, Chellappa R., in *FG 2017*. An all-in-one convolutional neural network for face analysis (IEEELos Alamitos, 2017), pp. 17–24.
24. X. Qi, L. Zhang, Face recognition via centralized coordinate learning. arXiv preprint arXiv: 1801.05678 (2018).
25. W. Liu, Y. Wen, Z. Yu, M. Yang, in *ICML*. Large-margin softmax loss for convolutional neural networks (ACMNew York, 2016), pp. 507–516.
26. S. Sankaranarayanan, A. Alavi, C. D. Castillo, R. Chellappa, in *BTAS*. Triplet probabilistic embedding for face verification and clustering (IEEELos Alamitos, 2016), pp. 1–8.
27. S. Sankaranarayanan, A. Alavi, R. Chellappa, Triplet similarity embedding for face verification. arXiv preprint arXiv: 1602.03418 (2016).
28. Y. Zhang, J. Lian, M. Fan, et al., Deep indicator for fine-grained classification of bananas ripening stages. Eurasip. J. Image Video Process. 2018(1), 46 (2018).
29. O. M. Parkhi, A. Vedaldi, A. Zisserman. Deep face recognition (British Machine Vision AssociationDurham, 2015), pp. 41.1–41.12.
30. J. Lian, Y. Zheng, W. Jiao, et al., Deblurring sequential ocular images from multi-spectral imaging (MSI) via mutual information[J]. Med. Biol. Eng. Comput.56(6), 1107–1113 (2018).
31. L. Wolf, T. Hassner, Maoz I., in *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*. Face recognition in unconstrained videos with matched background similarity (IEEELos Alamitos, 2011), pp. 529–534.
32. B. -C Chen, C. -S Chen, W. H Hsu, in *European Conference on Computer Vision*. Cross-age reference coding for age-invariant face recognition and retrieval (SpringerNew York, 2014), pp. 768–783.
33. E. Zhou, Z. Cao, Q. Yin, Naive-deep face recognition: touching the limit of LFW benchmark or not?Comput. Sci. abs/1501.04690: (2015).
34. Q. Cao, L. Shen, W. Xie, et al., VGGFace2: a dataset for recognising faces across pose and age, 67–74 (2017).
35. Y. Guo, L. Zhang, Y. Hu, X. He, J. Gao, in *ECCV*. Ms-celeb-1m: a dataset and benchmark for large-scale face recognition (SpringerNew York, 2016), pp. 87–102.

36. I. Kemelmacher-Shlizerman, S. M. Seitz, D. Miller, E. Brossard, *The megaface benchmark: 1 million faces for recognition at scale* (IEEE, Los Alamitos, 2016).
37. B. F. Klare, B. Klein, E. Taborsky, A. Blanton, J. Cheney, K. Allen, P. Grother, A. Mah, A. K. Jain, in *CVPR*. Pushing the frontiers of unconstrained face detection and recognition: Iarpa janus benchmark a (IEEELos Alamitos, 2015), pp. 1931–1939.
38. M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, et al., Tensorflow: large-scale machine learning on heterogeneous distributed systems. arXiv preprint arXiv: 1603.04467 (2016).
39. J. Liu, Y. Deng, T. Bai, et al., Targeting ultimate accuracy: face recognition via deep embedding. arXiv: 1603.04467 (2015).
40. Y. Taigman, M. Yang, M. Ranzato, L. Wolf, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. Deepface: closing the gap to human-level performance in face verification (IEEELos Alamitos, 2014), pp. 1701–1708.