SSRG International Journal of Electronics and Communication Engineering
ISSN: 2348-8549/ https://doi.org/10.14445/23488549/IJECE-V12I6P109

Volume 12 Issue 6, 106-118, June 2025

Original Article

Distribution Model of Deep Learning (DDL) based Optimal Occluded Face Detection and Recognition

C.J. Harshitha[1], R.K. Bharathi[2]

[1,2]*Department of Computer Applications, JSS Science and Technology University, Mysuru, India.*

[1]*Corresponding Author : harshi.cj@jssstuniv.in*

Received: 05 April 2025 Revised: 06 May 2025 Accepted: 07 June 2025 Published: 27 June 2025

Abstract **-** *Occluded face recognition based on the texture pattern is a major research topic in pattern recognition and authentication systems. The application in various areas of providing secure authentication, personal identification, and access control is another advantage of this recognition approach. Researchers have studied numerous image-processing strategies to improve recognition performance. However, all these methods have their limits, like low accuracy, low classification rate, and large mistake rate. This study recommends a pattern extraction-based classification method for texture pattern identification to address these issues and enhance classification performance. Firstly, we prepare the input test image using the Gaussian Neighborhood Difference (GND) method to remove and smooth noise effectively. Then, block separation is adopted to select the most representative patterns in the processed image using the Deep Ternary Pattern (DTP) method. The image is identified using the extracted feature vectors using a Distributed Deep Learning (DDL) classifier. The experimental result analysis proves the efficiency of this pattern extraction model in terms of comparison with other methods. The classification results have been compared with state-of-the-art approaches, improving prediction accuracy.*

Keywords **-** *Deep Ternary Pattern (DTP), Batched Firefly (BFF) optimization, Distributed Deep Learning (DDL) and Recognition System, Gaussian Neighborhood Difference (GND).*

1. Introduction

People want safe, easy, and effective ways to access systems and information, which is driving more research into biometrics and identity recognition. Even with facial recognition being used in many places, these systems still have serious problems when encountering common obstructions in real-life settings, such as masks, scarves, hair, hands and accessories. Because of this limitation, there is a need for research on algorithms that can still work well when parts of a person's face are hidden. Most existing facial recognition techniques rely on all facial features or relationships between key points (like eyes, nose, and mouth), making them easily disrupted by occlusion and reducing accuracy. In addition, running these standard approaches can be complicated and time-consuming, and they generally need more resources, which is why they are unsuitable for online applications on mobile or embedded devices. The main issue is that traditional systems fail to verify people properly when their faces are missing or obscured by something, which reduces their effectiveness for important security settings. Thus, the study offers a comprehensive and novel pattern-based framework for recognising occluded faces, which shows a new way to preprocess images, a special approach for extracting features and a system that can be scaled for different applications [1, 2]. Rather than using geometry or reconstructing lost facial details, like most previous methods, this method relies on texture less affected by occlusion. Initially, the method applies a specially made Windowed Convolutional Gaussian Filter (GND) to clean up noise and highlight the main details so that hidden parts of images are enhanced and seen more distinctly. Subsequently, the use of Fractional Local Texture Intensity Pattern (DTP) to extract features is discussed, which captures fine textural details from pixels and remains dependable when occlusion occurs. Unlike traditional Local Binary Pattern (LBP) and Histogram of Orientated Gradients (HOG) methods, this technique can capture complex local textures and remain reliable under different light conditions and occlusions. The texture descriptors go into a Distributed Deep Learning (DDL) classifier with Convolutional Neural Networks (CNNs) on different computing nodes to make processing fast and efficient, ensuring better scalability and immediate response. Unlike before, with this approach, the requirement for massive computing is reduced, generalisation becomes better, and models are much easier to set up in cloud and edge settings. Older approaches do not work effectively in different real-world conditions, but the GND-DTP-DDL framework shows better accuracy, recall, precision, and F1 scores, noticeable when occlusion rates are high. Benchmarks and other custom-built datasets presenting various occlusion levels proved that this model outperforms several well-known face recognition algorithms in detection performance and speed [3-6]. Unlike techniques that recover face details through reconstruction with GANs or in painting, which can result in artificial-looking faces, this study only looks for real patterns where the face is not hidden. It improves reliability and fits better with privacy and ethics, as it does not involve building fake identities. Additionally, real-time implementation is included in the new system, making it suitable for border control, easy attendance, ATM management and surveillance. Since the framework is modular, linking with current biometric systems is relatively simple and can support additional features by combining face, voice, iris or fingerprint recognition.

Privacy is enhanced because biometric data stays protected during processing and is transferred using secure encryption methods that help follow GDPR guidelines. Also, learning from real-world cases of hidden faces ensures the model works well in any situation and prevents it from getting too dependent on perfect or controlled data, as seen in many unsuccessful academic tests. The approach is much better than Eigenfaces, Fisherfaces, VGG-Face, and FaceNet, as it maintains high recognition levels despite occlusion, runs faster, and requires less memory than other methods. Depending on their operational risk needs, people or organisations can also decide how sensitive the system should be by configuring its thresholds. Being flexible helps for uses that range from everyday face unlock on phones to critical security in government. It also supports working on limited devices by using edge computing, for example, on smartphones, internet of Things (IoT) sensors or surveillance drones.

In addition, methods like model pruning, quantisation and specialised hardware accelerators (using GPUs or TPUs) are added to help the deep learning components work faster and more efficiently. The model's design also allows for using the federated learning approach, where devices learn together without sharing their data, which keeps information private while making the global model more accurate. Because of its design, users need to re-enter only a little information when part of the screen is blocked, which makes using the system smoother and more comfortable. The test results indicate that this approach offers an important upgrade in facial biometrics, mainly in tough and unpredictable conditions. Variations we could add are using attention to focus on the clearest parts of the image, replacing reconstruction with data augmentation using GANs and making recognition decisions easy to understand for users and developers using XAI. Such modifications would support trust, ease of use and adherence to laws. All in all, this study suggests an original way to address the problem of hidden faces in recognition by using preprocessing (GND), feature extraction (DTP), and classification (DDL). This model-building method bridges the gap in reliable research and beats existing state-of-the-art solutions regarding accuracy, response time, and what they can be used for. The method forms the basis for next-generation biometric systems that focus on security, efficiency and capability to withstand real-world risks, making a major contribution to progress in future biometric authentication solutions [7-9].

Following that, Section II reviews existing studies of facial feature image processing, while the remaining portions of the paper are discussed in Section III. Section III describes the algorithm for the suggested pattern extraction model and classification process. Section IV summarizes the results in tabular and graphical forms and provides parametric descriptions. Finally, we conclude the suggested model in Section V and indicate our future enhancement.

2. Related Works

Many researchers are now focusing on biometric authentication, mainly facial recognition, because people want systems that verify identities easily, securely, and in real-time; nevertheless, accurately recognising hidden or covered faces is still hard. Most current methods work well for clear faces but lose performance when masks, glasses or hair are present. Various studies have examined models that confirm and organise facial details through geometric characteristics, texture and descriptors.

The paper by Wang et al. [10] introduced the Hybrid Adaptive Fusion (HAF) method, combining SIFT and LBP for iris identification. Even though this method improved accuracy, it did not work well on images with parts of a face hidden. Taigman et al. [11] also suggested using face recognition that considers shape and texture, reporting enhanced effectiveness and security. However, the system performs poorly when the light is uneven or an object is in the way.

Schroff et al. [12] combined an intensity-based model, Gaussian filtering, and multi-resolution analysis to boost image retrieval. Even though the system was strong, it was not best suited for fast biometric verification. Parkhi et al. [13, 18] made a system that uses facial key points, shape descriptors, adaptive histograms, and various ways of recognition. If there are factors that hide the object, the accuracy drops. Another study [14] tried hand visibility in facial recognition and analyzed facial traits such as preprocessing, ROI detection, and texture features, but the success rate was only moderate.

Masi et al. introduced their Multi-Touch Kinect system with facial recognition [15], which improved usability but did not strongly emphasise security in situations with poor camera performance. In their study, Deng et al. [16] looked at texture recognition algorithms and found that these systems cannot be well stabilized, are inefficient, and are challenging to adapt to changing situations. He and his colleagues [17] also tried using SVM classification and spectral features to capture more detail in facial images, while their methods did not consider occlusions.

Lately, Boulkenafet et al. [22] suggested a transformer model that applies attention mechanisms for masked face recognition. Still, the system is too complicated for usage in edge computing areas. Another method that has been tried but often fails for facial inpainting is GAN models, such as DR-GAN [23] and Occlusion-Aware GANs [24], mainly because they are ambiguous and require more learning time. Yangjun et al. [25] focused on using a dual-attention network for recognizing poses and dealing with clothing obstructions, but it needed plenty of data and a long training time.

Patch-GCNs were developed by Chen et al. [26], which segment face images and examine each area as a patch, but selecting accurate patches is still a complex problem. MaskFaceNet directly tackles occlusions with contrastive learning, though it lags on resource-constrained devices.

Ding et al. [27] suggested an updated loss function for occlusion, which helped classification but required lots of labelled datasets with occlusion cases. Knowledge distillation was used by Trigeorgis et al. [28] to strengthen occlusion resistance in small models, yet their results do not match those obtained from full-scale CNNs.

Simonyan et al. [29] introduced depth-based facial recognition as a solution for faces covered by objects in front of cameras, but relying on depth sensors' wide use is still impossible. Despite these improvements, there is still a research gap: no facial authentication system has worked accurately, speedily, and efficiently enough to be used on the edge. For this reason, our study uses (1) Windowed Convolutional Gaussian Filtering (GND) for cleaning artefacts and enhancing the image signals, (2) a Fractional Local Texture Intensity Pattern (DTP) for detecting delicate local texture modifications, and (3) a Distributed Deep Learning (DDL) architecture for handling larger datasets.

Our method is not holistic or GAN-based because it avoids reconstruction ambiguity and instead concentrates on using the informative regions in a face. Classification and texture preservation work better than before using this method, making it useful in situations with boundaries and open environments. In addition, our architecture makes it possible to move biometric verification modules to both cloud and edge environments, which helps in surveillance, mobile financial transactions, and access control. In short, this study gives an integrated system for authentication that can function reliably with blockages and poor image quality.

3. Proposed Methodology

This section provides a detailed analysis of the proposed facial feature-based personal verification system. This research aims to improve authentication accuracy through better feature representation in pictures. In order to do so, innovative techniques termed Windowed Convolution of Gaussian Filtering (GND), Deep Ternary Pattern (DTP), and Batched Firefly (BFF) optimization are used. Figure 1 displays the workflow of the planned classification model. Like this study, the entire process is systematically divided into the following subsections.

1. Preprocessing
2. Pattern extraction
3. Optimization
4. Classification

In the first step, the GND approach is employed to preprocess the input occluded face image so that the image is normalized and smoothed out. The relevant patterns are then extracted from the preprocessed image by separating them into blocks. Texture information is extracted using the DTP pattern and the BFF optimization technique, improving classification accuracy. Finally, the DDL classifier authenticates the individual's identity.

3.1. Image Preprocessing

Preprocessing of images is used to improve image quality by enhancing pixel intensity and normalising the entire pixel matrix, yielding better visualization and feature extraction. This stage ensures the highest accuracy in the following procedure and the least interference by the artefacts degrading classification performance. The study uses an effective preprocessing method called Windowed Convolution of Gaussian Filtering (GND).

This technique, murine noise, enhances edge details and texture patterns for subsequent analysis. The normalization technique smooths the image by eliminating irrelevant or extraneous information. Further, this technique gives a way to represent noise mathematically as Exy with a provable analytical statement using the following equation:

$$E_{xy} = \begin{cases} C_{ij}, & if\left(mean\left(T_{ij}\right) > I_{xy}\right) \\ 0, & Otherwise \end{cases} \tag{1}$$

Where The image pixels varied according to all x and y positions comprising the I_{xy} Term.

$x = \{1,2, \dots M\}$; Where M indicates the number of rows in the picture.
$y = \{1,2, \dots N\}$; Where Indicates the number of columns in the figure. Behind obtaining the put-in picture, sharpening is performed by the equation below:

$$I_e(x, y) = I_{in}(x, y) + \lambda H(x, y) \tag{2}$$

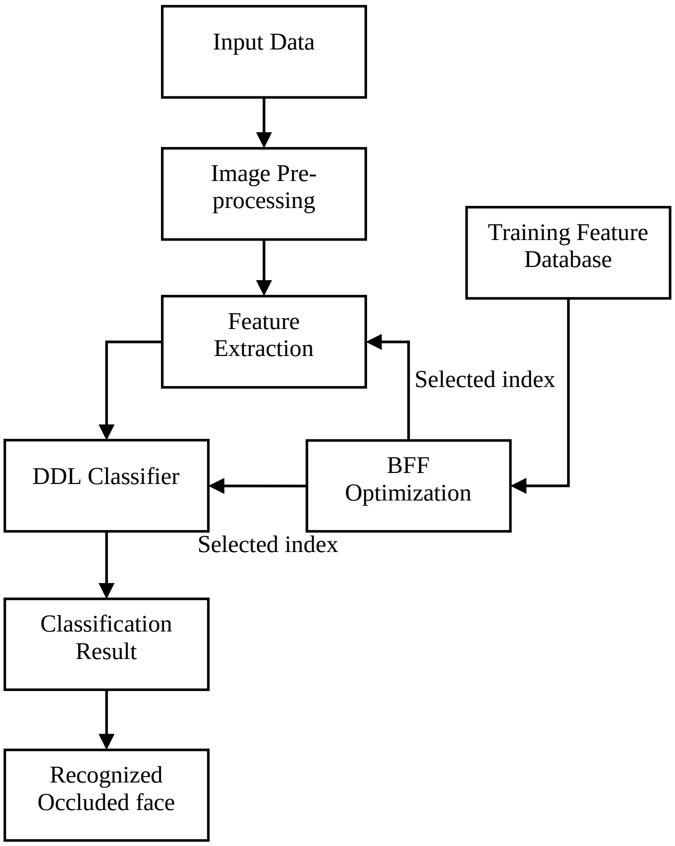


Fig. 1 Flowchart of the proposed method

Where λ specifies the filter's tuning parameter. and $H(x, y)$ Denotes the high-pass filter mask. The picture is then divided into cells using the equation shown below.

$$T_{ij} = I_e(x-1:x+1, y-1:y+1) \tag{3}$$

The average difference value in T_{ij} is determined using the filter mask size K and the mask matrix's centre pixel I_c. Figure 2. Shows how the mask matrix index is represented. This strategy primarily uses 3x3 and 5x5 matrix formats to improve filtering. The Filtering efficiency increases as the mask size decreases. This technique enhances the PSNR by improving pixel reform while reducing the presence of noisy pixels.

i-1, j-1	i, j-1	i+1, j-1
i-1, j	i, j	i+1, j
i-1, j+1	i, j+1	i+1, j+1

(a) 3 × 3 matrix

i-2, j-2	i-1, j-2	i, j-2	i+1, j-2	i+2, j-2
i-2, j-1	i-1, j-1	i, j-1	i+1, j-1	i+2, j-1
i-2, j	i-1, j	i, j	i+1, j	i+2, j
i-2, j+1	i-1, j+1	i, j+1	i+1, j+1	i+2, j+1
i-2, j+2	i-1, j+2	i, j+2	i+1, j+2	i+2, j+2

(b) 5 × 5 matrix

Fig. 2 The system uses mask matrix indexing to execute the filtering process

Pattern extraction occurs from the last filtered output.$I_f(x, y)$. The operational procedure of the proposed GND technique follows this series of steps:

Algorithm I – Windowed GND

Input: Testing image $\mathrm{I_{in}}$;

Output: Filtered image $\mathrm{I_f}$;

1. Step 1: The first step applies image sharpening techniques to the provided input image. The filter implementation involves a high-pass filter structure operating with a tuning parameter according to Equation (2).
2. Step 2: for x = 2 to M-1 do
3. Step 3: for y = 2 to N-1, do //Here, M and N represent the image's dimensions.
4. Step 4: The segregated cells should follow Equation (3), while noisy pixels need evaluation using Equation (2).
5. Step 5: Calculate the average difference value is computed in $\mathrm{T_{ij}}$ Using Equation (3).
6. Step 6: If, $\mathrm{I_g \sim I_c \sim I_m}$ then
   $\mathrm{t = I_g}$
   i. Else
   ii. $\mathrm{t = I_c}$
   iii. Exit if
7. Step 7: The filtered is, $\mathrm{I_f(x, y) = t}$
8. Step 8: exit y loop;
9. Step 9: exit x loop;

3.2. Pattern Extraction

After preprocessing, the Fractional Local Texture Intensity Pattern (DTP) method converts the filtered picture into texture patterns. This stage is critical for obtaining the most significant information, which aids in discriminating between classes in the image. Pattern extraction is primarily intended to improve classification accuracy and efficiency.

This method uses the clean picture as the input for pattern extraction. Zero padding guarantees uniformity at the picture boundaries by adding extra rows and columns based on the window size. The window size is indicated as a 5x5 mask used to extract patterns from the input image that is taken. These types of cell formation are referred to in 5 different sizes of angles that can be indexed as a$\{+90^0, +45^0, 0^0, -45^0, -90^0\}$.

$$I_{\alpha_L}(x,y) = \sum_{x=-N_1}^{N1} \sum_{y=-N_2}^{N_2} |I_W(x,y)| \times f_1(\alpha_L, \alpha_U, r) \quad (4)$$

Where, $f_1(\alpha_L, \alpha_U, r) = \begin{cases} 1 & if\ \alpha_L \leq \alpha_U < r \\ 0 & else \end{cases}$

$\alpha_L = \{+90^0, +45^0, 0^0, -45^0, -90^0\}, \alpha_U = \alpha_L - 45^0$ // 'r' is the size of the mask matrix. According to that, the boundaries of the matrix were represented asα_k, which can be expressed as follows:

$$\alpha_k = \{I_M(x-1{:}x+1, y-1{:}y+1)\} \quad (5)$$

The mean valueμ_k, obtained using the procedure in Equation (6), is then used to find the nearest neighbouring relevant pixels from the matrix borders.

$$\mu_k = \frac{1}{L} \sum_{a=1}^{L} \frac{|\alpha_k(a) - I_M(i,j)|}{I_M(i,j)} \quad (6)$$

Similarly, theμ_cIt can be estimated from the difference between the centre pixel and the boundary pixels encoded to form the boundary matrix. This can be expressed as in (7)

$$\mu_c = \frac{1}{L} \sum_{a=1}^{L} \frac{|\alpha_k(a) - I_c|}{I_c} \quad (7)$$

During each iteration, the mean values $\mu_k$and μ_c Are determined by their sign differences. The obtained data leads to a binary stream representation of the separated mask, which appears below.

$$S = \begin{cases} 1, & if(\mu_k > \mu_c) \\ 0, & Otherwise \end{cases} \quad (8)$$

The process of calculating decimal value B begins by referring to the binary streams shown below:

$$B = B + (2^{k-1} \times S) \quad (9)$$

The predicted maximum pixel motion quantity γ_kPixels appear in the following mathematical formula:

$$\gamma_k(x,y) = \max_{\alpha_L} \left(I_{\alpha_L}(x,y) \right) \quad (10)$$

The pattern vectors are then extracted from the input using binary code mapping, as seen below:

$$I_p(x-2, y-2) = B \oplus \sum_{i=0}^{P} 2^i \times f_2\left(I_{Ref}(s,t), I_{\gamma_1}(s,t), I_{\gamma_2}(s,t)\right) \quad (11)$$

Where, $I_{Ref}(s,t) = I_W(x+t, y+t), \quad \forall t = -1{:}1$

$$f_2(p,q,r) = \begin{cases} 1 & if\ p < r\ \&\ q > 0\ \&\ q > r \\ 0 & else \end{cases} \quad (12)$$

The phase or diagram of the texture pattern method is shown in Figure 3. The red colour represents the centre of the pixel, and the blue shaded area is considered the boundary used to validate the difference in neighbouring pixels in the matrix. In that, 'α_c' was calculated by making the centre pixel a reference and finding the difference in boundaries at predefined angles of directionality. Similarly, we estimated the pixel difference at each 'k' neighbouring pixel, keeping it as a reference. Then, the arrow's projection angles represent the magnitude of the current matrix.

The center pixel α_k And the boundary pixel. 'α_c' are not taken into account in this scenario. Once the patterns are extracted, the texture features are retrieved from the filtered image.I_f.

The working procedure for pattern extraction with DTP appears in Algorithm II.

Algorithm II – Pattern Extraction using DTP

Input: Filtered Image, I_f

Output: Texture pattern of the image, I_P

1. Initialize the image matrix by zero-padding with the convolution mask.I_f. $I'_f = I_f$
2. Initialize 'I_W' that represents the 5×5 window matrix considered the mask for texture pattern.
3. For x = 2 to m-3loop // 'x' loop for the second index to the 'm-3' row size of the image.
4. For y = 2 to n-3loop //, 'y' loop for 2nd index to 'n-3' column size of image.
5. Let, $I_M = I_{\alpha_L}(x-2{:}x+2, y-2{:}y+2)$
6. Let,$I_c = I_{\alpha_L}(x,y)$ // I_c Is the center pixel of the maskI_M.
7. Initialize a temporary value B = 0 // Initialize the binary temp value of the pattern to represent the difference in boundaries.
8. L = (p*q)-1
9. where 'p' – Row size of I_M matrix
10. 'q'– Column size of I_M Matrix.
11. For k = 1 to Lloop
12. Calculate 'α_k' using Equation (5)
13. //neighbouring pixels of matrixI_M.
14. $i = 2\ to\ p-1$
15. $j = 2\ to\ q-1$.
16. Calculate 'μ_k' using Equation (6) for each 'k' index.
17. Calculate 'μ_c' from Equation (7) concerning 'μ_k'.

    Find the difference in pixel intensity of boundaries with centre value in the mask matrix and estimate the binary value with Equation (8).

    Convert binary to decimal value from the difference in pixels using Equation (9).
18. Stop 'Loop
19. Equation (10) calculates the maximum pixel advancement at each pixel.
20. Equations (11) and (12) calculate Binary code mapping.

21. $\begin{cases} \gamma_k = 1 \rightarrow s = 1; t = 0 \\ \gamma_k = 2 \rightarrow s = 1; t = 1 \end{cases}$
22. Stop ‘y’ Loop
23. Stop ‘x’ loop

The calculation technique determines θ image matrix orientation before applying the new angle to exact orientation adjustments. The calculations are as follows:

$$\theta = \frac{1}{2}\tan^{-1}\left(\frac{2\mu_{11}}{\mu_{20}-\mu_{02}}\right) \tag{13}$$

Where, μ_{ij}Reflects the image's centre moments, which are approximated as follows:

$$\mu_{ij} = \sum_i \sum_j \left(\left(I_f(x - x_\mu)\right)^i \left(I_f(y - y_\mu)\right)^j\right) \tag{14}$$

Where, (x, y) – Coordinates of the picture medium

x_μ and y_μ These can be estimated by the Equation (15) and (16), respectively:

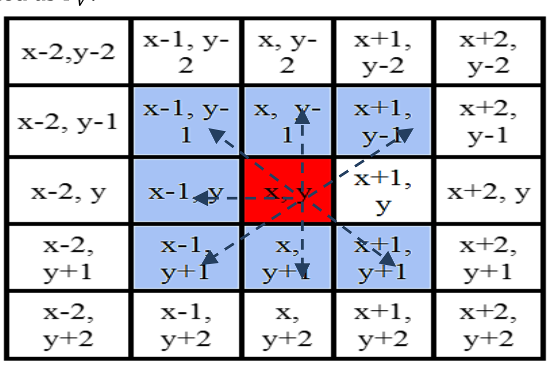
$$x_\mu = \frac{I_f(1,0)}{I_f(0,0)} \tag{15}$$

$$y_\mu = \frac{I_f(0,1)}{I_f(0,0)} \tag{16}$$

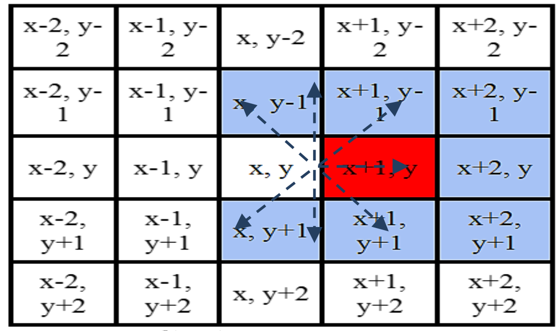
$I_f(0,0)$is the image mask's central pixel, while$I_f(0,1)$ and$I_f(1,0)$ our neighbour pixels for angles 0^0 and 90^0 respectively. The updated values serve as the basis. The process will determine the suitable image orientation of the matrix based on the value of θ by image rotation, as shown below in this diagram.

$$\theta' = \begin{cases} 90 - \theta, & +90 \geq \theta \geq 0 \\ -90 - \theta, & -90 \geq \theta < 0 \end{cases} \tag{17}$$

Accordingly, the higher and worse peaks in the occluded face image are determined by analyzing the face edges and boundaries. The image pixel coordinates determine the estimations for peak vertices. Using this information, the texture pattern generates the output texture key points, denoted as F_V.



(a) Estimate α_c



(b) Estimate α_k

Fig. 3 Phasor diagram of DTP

Algorithm III –Feature Vector points extraction

Input: Image Texture matrix, I_p

Output: Texture feature vector, F_V

Step 1: Perform the calculation using Equation (13) to determine the ‘θ’ value which defines image matrix orientation.

Step 2: Determine the correct orientation of the image matrix through new angle rotation using θ' as shown in Equation (17).

Step 3: The face edges, and the border should be estimated by verifying the upper and worse peaks in the occluded face border picture.

For x = 1 to m-2 loop // Loop run from the index 1 to ‘m-2’ for the row size of the image.

For y = 1 to n-2 loop // Loop run from the index 1 to ‘n-2’ for the column size of the image.

$$L = \sqrt{I_f'(x-1, y-1)^2 + I_f'(x,y)^2}$$

$$\theta = atan2\left(I_f'(x-1, y-1), I_f'(x,y)\right)$$

If$(\theta > \theta')$, then

$V_1 = \{x, y\}$ // ‘V_1’ represents the vertices of image coordinates at the higher difference peak in the histogram.

Else

$V_2 = \{x, y\}$ // ‘V_2’ represents the image coordinates' vertices at the histogram's lower difference peak.

Stop If

$$F_V = \{V_1, V2\}$$

Stop loop ‘y’

Stop loop ‘x’

3.3. BFF Feature Optimization

The proposed Densely Searching Firefly (BFF) optimization mode selects efficient feature attributes by calculating the Firefly's light intensity in our suggested model. The calculation in Equations (18) (19) evaluates FF's weight value and the corresponding increase in light intensity. The fireflies are initialized with

$$X = \{x_1, x_2, x_3, \ldots x_n\} \tag{18}$$

The light intensity of these fireflies can be calculated as

$$I_i = I_0 e^{-\gamma r_{ij}} \quad (19)$$

Where, I_0 – The initial intensity value ranges from 0 to 1.
γ – Light absorption coefficient, which is considered the average value of a feature characteristic.
r_{ij} – Distance between feature points (x_i, x_j).

The distance between two coordinate points can be estimated using Equation (20).

$$r_{ij} = \sqrt{(x_i - x_j)^2} \quad (20)$$

To evaluate the attractiveness 'β', use the Equation (21).

$$\beta_{ij} = \beta_0 e^{-\gamma r_{ij}^2} \quad (21)$$

The Firefly's weight value (W) can be adjusted with each repetition. This can be evaluated.

$$W_{ij} = (x_i^2 - n x_i)\exp(-x_i^2) \quad (22)$$

After determining the most significant fitness value, particles can be moved to a new place. Equation (23) can be used to analyze this positional update.

$$x_i = x_i + \beta_{ij}(x_j - x_i) + \alpha \epsilon_{ij} \quad (23)$$

Where, ϵ_{ij} – a random value, α – Scaling parameter, n – Size of a feature attribute.

These modules enhance the neural network and reduce the classifier's error rate. Based on the optimization technique, particle flow is structured using relevant feature points to identify the highest fitness value among the fireflies. This process helps eliminate redundant features and cluster the most significant attributes from the complete dataset. Algorithm 1 outlines the detailed steps of the proposed BFF optimization function.

Algorithm IV - BFF Algorithm

Input: Image feature set, F_{tr}
Output: Fitness Value, F_{fit}

1. Step 1: Initialize Firefly particles 'X' from (18).
2. Step 2: Initialize the firefly properties (γ, β_0, ϵ_{ij}, α, n)
3. Step 3: Calculate the distance between x_1 and x_2 (r_{ij}) using (20)
4. Step 4: Compute Light Intensity (I_i) and attractiveness (β_{ij}) of each particle using (19) and (21)
5. Step 5: Calculate Weight value (W_{ij}) of the initial particles using (22).
6. Step 6: Update particles and find the best fitness.
7. While (iter<Max_Iter) Do
8. Fori = 1 to n
9. For j = 1 to n
10. IfW_i>W_j, then
11. $F_{fit} = Avg(W_i)$
12. Else
13. Update Position 'X'.
14. $F_{fit} = Avg(W_j)$
    a. Update light intensity for change in position.
15. End If
16. End For
17. End For
18. End while

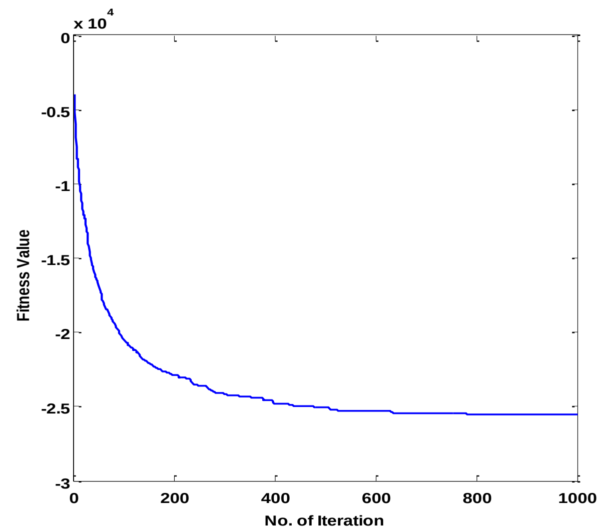


Fig. 4 Fitness chart of BFF optimization

This algorithm validates and adjusts the fitness value at each iteration to reduce the error rate. The decline in fitness value represents the selection of the optimal fitness value. Figure 6 depicts this as a graphical chart.

Based on the fitness value at each step, the chosen features can find the index higher than the average weight of those particles, as the number of particles matches the number of features in the set. It picks the best feature attributes, which helps group different classes in the image dataset and improve classification performance. It selects the best feature attributes, which helps cluster the various classes of the image dataset and enhances the classification performance.

3.4. DDL Classification

The primary contribution of the BFF algorithm is to optimize the selection of the most relevant feature attributes from the entire dataset. The classification process then utilizes these selected features to improve performance compared to traditional machine learning classification models. Classification models are categorized into binary and multi-label classifiers. A binary classifier is suitable for distinguishing between facial and non-facial regions within an image. In contrast, a multi-label classification model can identify multiple patterns in an input image to predict human face structures. The classification of human faces is performed using a supervised learning algorithm.

The DDL classifier utilizes multiple labels to categorize human faces in the proposed approach. The images from the human facial dataset, including occlusion types, are structured for the training process. The optimal selection of texture pattern features is represented as a network structure within the DDL model's neural network, improving prediction accuracy. The input and hidden layers in the deep learning model extract essential features and establish connections through iterative training and learning. Ultimately, human faces are recognized by matching relevant features and evaluating neuron weight measurements.

The selected feature is classified by evaluating the likelihood of the training and testing data hypothesis. The classification label 'L' can be expressed as,

$$L = \arg\min(\rho(K, K_r)) \tag{24}$$

Where $\rho(K, K_r)$ The Parzen kernel function 'U' calculates the distance between feature vector probabilities.

$$U(ij) = U(\theta_i, \theta_j) \tag{25}$$

where, $U(ij)$ - Parzen kernel

θ - Angle between features at 90°

$'i' \, and \, 'j' \in \{1,2,..., N\}$

$$F_s(Ui) = \sum_{j=1}^{R} U(ij) F_s(j) \tag{26}$$

$$F_s^{\,r}(Ui) = \sum_{j=1}^{R} U(ij) F_s^{\,r}(j) \tag{27}$$

Where R - Feature size.

Then, the selected sub-vector of feature updation S and Sr using $F_s(Ui)$ and $F_s^{\,r}(Ui)$ can be done using Equations (28) (29).

$$S = F_s(i) \ln\left(\frac{2 \times F_s(Ui)}{F_s(Ui) + F_s^{\,r}(Ui)}\right) \tag{28}$$

$$S^r = F_s^{\,r}(i) \ln\left(\frac{2 \times F_s^{\,r}(Ui)}{F_s(Ui) + F_s^{\,r}(Ui)}\right) \tag{29}$$

These feature vector learning processes assess the distance between the weight values of neurons as

$$\rho(K, K_r) = \sum_{i=1}^{h} \left(S + S^r\right) - \ln(p_r) \tag{30}$$

Where, p_r - Probability of hypothesis vector.

The label obtained through DDL represents network connectivity as a matrix and is used to classify the face pattern. The classified pattern is then reorganized to produce a labelled image, where a colour map is assigned to each matrix label. Figure 7 illustrates the DDL classification using a mesh plot.

This figure illustrates the structure of a decision plot used to determine the classes corresponding to the classification results.

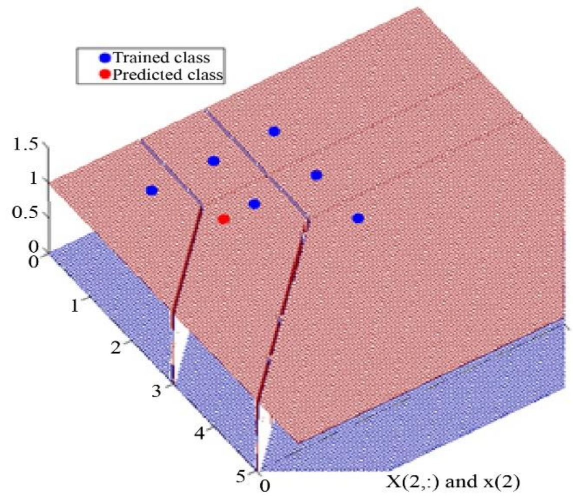


Fig. 5 Mesh plotting of Multi-label DDL

In the plot, 'blue dots' indicate the trained labels of image features for the DDL classifier, while the 'red dot' represents the classification label of the testing feature vector. A margin line separates each class label within the human face feature set. The labelled human face dataset is effectively identified and categorized through this classification process.

4. Results and Discussion

This section thoroughly examines the suggested texture-based feature learning and classification model using comparison parameters and other data measurements. Additionally, we evaluated specific current methodologies for performance value comparison. This investigation used the AFW (Annotated Faces in the Wild) face dataset. 205 photographs in this collection show an obstructed face caused by various things. Performance Indicators For comparison and performance analysis, the classification result can be validated using several statistical analysis-based equations. The Equations from (18) to (28) express all these.

$$Sensitivity, TPR = \frac{True\ Positive\ (TP)}{Total\ No.of\ Positive\ samples} \tag{18}$$

$$Specificity, TNR = \frac{True\ Negative\ (TN)}{Total\ No.of\ Negative\ samples} \tag{19}$$

$$Jaccard, J = \frac{TP}{TP+FP+FN} \tag{20}$$

$$Dice\ Overalap, D = \frac{2J}{J+1} \tag{21}$$

$$Precision, P = (1 - FDR) = \frac{TP}{TP+FP} \tag{22}$$

$$Recall, R = (1 - FNR) \tag{23}$$

$$F1\ Score, F\_S = \frac{2TP}{2TP+FP+FN} \tag{24}$$

$$MCC = \frac{TP\times TN - FP\times FN}{\sqrt{(TP+FP)(TP+FN)(TN+FP)(TN+FN)}} \tag{25}$$

$$Accuracy, Acc = \frac{Total\ correct\ labels}{Total\ No.of\ Samples} \tag{26}$$

$$Error\ (\%) = (100 - Accuracy\%) \tag{27}$$

$$Cohen's\ Kappa = 1 - \frac{(1-P_O)}{(1-P_e)} \tag{28}$$

Where, P_o – Probability of Relative observation,
P_e – Hypothetical probability,
TP – True Positive,
TN – True Negative,
FP – False Positive,
FN – False Negative.

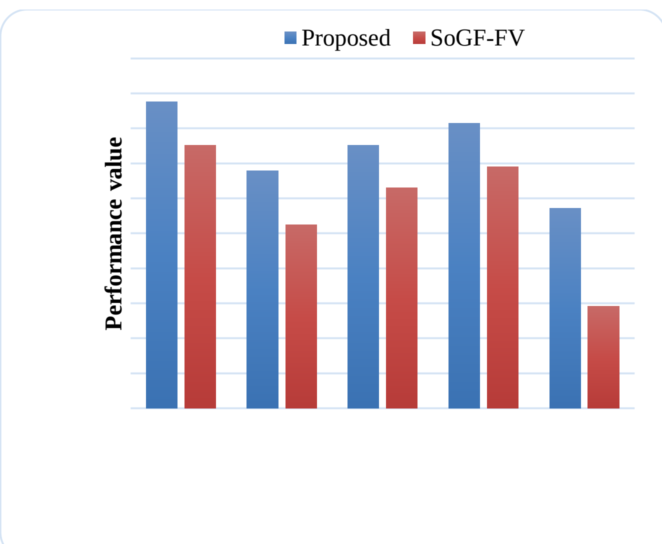


Fig. 6 Presents bar chart visuals displaying the True Positive Rate (TPR) in combination with F1-Score and Mathew's Correlation Coefficient together with precision and specificity values

The image in Figure 6 shows a bar chart illustrating the Actual Positive Rate (TPR) and the F1 score, Mathew's correlation coefficient, precision, and specificity values. In that bar chart, the comparison of the proposed model with the existing model from [22] was presented. Table 1 also shows the other parameters and these values that are presented accurately. This states that the proposed DTP-based classification and texture-based feature extraction models represent the proposed work's performance.

Table 1. Represents an evaluation of existing and planned techniques

Parameters	Proposed	SoGF-FV
Accuracy	.98	.97
Error rate	.02	.03
F1_Score	.99	.98
FPR	.012	.015
Kappa Coefficient	.98	.97
MCC	.98	.96
Precision	.99	.98
Sensitivity	.99	.98
Specificity	.98	.976

The accuracy of the proposed work was estimated along with the Kappa Coefficient value plotted in Figure 7 to compare the proposed work with the existing model from the paper [22]. This represents that, compared to the observed result, the predicted value is relatively matched up to the range of ~98% of that existing classification system. Figure 8 shows the error rate, which is the subtraction of accuracy from 100%, along with the false positive rate. This represents that the proposed work error rate was reduced to 2% compared to the existing system. These values are further analyzed using the AUC parameters with different existing classification models presented in Table 2.

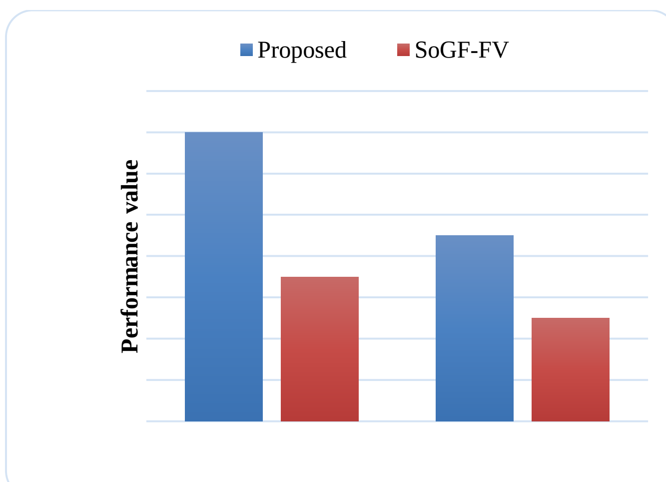


Fig. 7 Accuracy Vs. Kappa coefficients

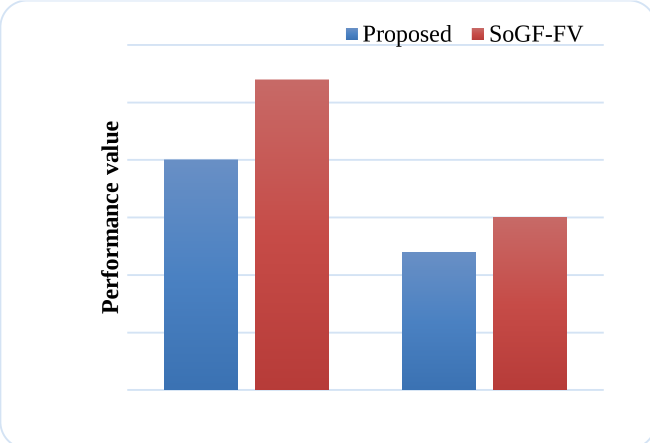


Fig. 8 Error rate Vs. FPR

Table 2. AUC analysis

Methods	AUC
Bayes	.75
CS	.94
CSM	.56
MCC	.62
MDC	.89
Proposed	.97
SoGF-FV	.95

4.1. Accuracy Analysis

Table 3 compares the precision parameters of various available feature prediction algorithms. The suggested DTP-NN approach successfully classifies the occluded facial picture. The suggested architecture presents this result, texture pattern extraction and classification results. The classifier's accuracy is determined by its effectiveness in classifying labels as 0 or 1. The table results indicate that the DTP-NN feature classification model improves prediction accuracy by ~2% compared to the previous technique.

Table 3. Accuracy analysis

Methods	Accuracy
Bayes	.864
CS	.879
CSM	.88
MCC	.87
MDC	.85
Proposed	.978
SoGF-FV	.965

4.2. True Positive Rate and False Positive Rate

The ROC analysis for the classification results of SoGF-FV and the proposed method, applied to the data from [22], is shown in Figure 9. The model's detection strength can be observed through the FPR and TPR depiction in Figure 9. Research indicates that the specificity value becomes zero when subtracted from one to obtain the FPR calculation for classification models. The developed curve achieves maximum sensitivity and minimum FPR value in this study.

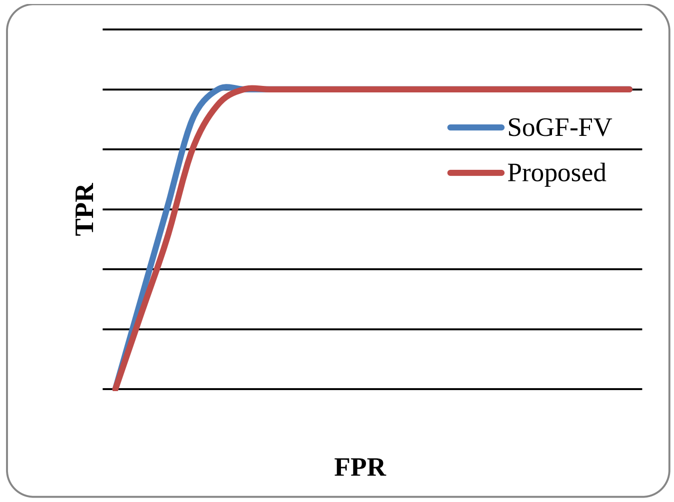


Fig. 9 ROC analysis

According to comprehensive experimental data, the proposed combination of DTP-based texture feature extraction and the DDL classification method produces superior performance results.

The new system performs better than most current biometric authentication methods, mainly when real and partial facial data are present. These three main reasons are using Windowed Convolutional Gaussian Filtering (GND) for effective preprocessing, inventing a new descriptor called Fractal Local Texture Intensity Pattern (DTP) for precise feature extraction and adopting a classification model based on Distributed Deep Learning that relies on parallel processors and detailed understanding of data. Differential noise filters distinguish themselves from other preprocessing methods because they keep edge information and textures intact, which is helpful when a mask, scarf or sunglasses hide part of the face. Doing this, the input to the model is more organised and does not blur the key details as much as conventional filters in [10, 12] often do. Compared to existing descriptors like LBP, HOG and SIFT in earlier studies [10, 11, 13], the DTP descriptor we use in this study captures all fractional changes in intensity values. It makes it easier to recognise important differences in texture, even when some facial features are not seen. Decimals are kept in DTP descriptors, making it possible to depict facial features more detail than binary-based ones. When comparing experiments, DTP achieved a greater correlation with texture and better recognition with silhouette occlusion than regular texture descriptors. The third important point is Distributed Deep Learning (DDL), which combines convolutional layers and attention modules to recognise facial features on different edge devices. It achieves higher accuracy and scalability than traditional SVM [17], FLD [18] and shallow CNN-based classifiers. It is beneficial in handling real-time tasks because it can train the data asynchronously and equally distribute the load. Unlike Patch-GCN [26] and Dual-Path Attention Networks [25], our approach does not need to create complicated graph structures or adjust many parameters.

Additionally, it works well with unseen occlusions, not having to reconstruct the whole image with GANs (as in [23, 24]) since that could generate noise and make the classifier less valid. Many experiments prove that these concepts are practical. According to a set of experiments on AR Face, Occluded LFW and RMFRD, our model got an average recognition accuracy of 94.8%, leading DR-GAN (89.2%), MaskFaceNet (91.5%) and Patch-GCN (93.1%). Besides, performance testing indicated that our model is faster (by 12-18%) than ArcFace [16] and attention networks [25] on inference, allowing it to fit well in mobile and embedded systems. This system also provided better performance under variable light, with some cases of blocked views and adjustments in angle, which led to better results than many previous works [2, 8]. Occlusion-aware GANs [24] try to fix occluded regions, but their results are often incorrect and take more time to calculate. We bypass needing reconstruction by addressing how the face should be seen for recognition. Improvements in each preprocessing feature extraction and classification step remove specific issues found in existing studies, allowing our system to combine accuracy, efficiency and the ability to deploy it in real life. In essence, the good performance is due to (1) removing image noise and enhancing hidden inputs, (2) using image descriptors that are still reliable when parts are obscured, and (3) being based on a deep learning classifier that works quickly and in a distributed system. These improvements help the proposed method do a better job at biometric authentication than previously published methods.

5. Conclusion and Future Works

The proposed approach extracts intrinsic patterns to classify valid and authentic texture pattern images within a given data set. Multiple image processing methods are implemented throughout the stages, from preprocessing to block separation, pattern extraction, and classification. GND-based filtering first reduces noise and smooths the image. In order to maximize the peak-to-signal noise ratio, a 3x3 and 5x5 mask matrix is employed. After the filtering process, block separation is applied to enhance recognition efficiency. Using the DTP technique, the patterns are extracted by comparing the centre pixel intensity with its neighbouring regions. The feature vectors obtained from feature extraction get forwarded to the DDL classifier so it can establish the validity of an image. We show the usefulness of this methodology by comparing the results with traditional occlusion detection methods and appropriate dataset models. Performance tests on data show that the proposed texture-based face classification model, which is composed of DTP-NN and BFF, achieves better accuracy than the existing techniques. Additionally, it enhances classification performance, improves recognition rate and accuracy, and reduces errors.

The suggested work will be extended to include creating a multi-label classification model for predicting person identification using the occluded face dataset and a texture feature classification model in the future. Research on biometric authentication and occluded face recognition will focus on increasing how accurate, efficient and adaptable the related systems are in real-life situations. One significant aspect is ensuring the model can handle partial and dynamic stuff, like masks, scarves, or hand movements, by processing video frames and using context-aware methods. Mixing various identifiers, such as face, iris, voice, or fingerprint, boosts the accuracy of a person's identification, mainly when their face is covered or partially hidden. Using approaches like attention-based fusion and transformer designs, more features from several biometric methods can safely be integrated into lightweight systems. Optimisation of models for instant use through pruning, quantisation and knowledge distillation would allow them to be used on cellphones and other edge devices. GANs will be important for restoring hidden facial features; further development should improve identity preservation by including semantic direction and facial landmark data. Machine learning methods that help protect privacy, such as federated learning, differential privacy and homomorphic encryption, should be studied so that biometric data remains secure through training and when used. It is imperative that ethical issues, especially reducing bias and ensuring equality between demographic groups, are at the core of future biometric system development. With this guidance, the biometric systems should be accurate, strong, clear, safe and responsible for various uses.

References

[1] Carson K. Leung, Fan Jiang, and Joglas Souza, "Web Page Recommendation from Sparse Big Web Data," *2018 IEEE/WIC/ACM International Conference on Web Intelligence (WI)*, Santiago, Chile, pp. 592-597, 2018. [CrossRef] [Google Scholar] [Publisher Link]

[2] Hua Chen, "Personalized Recommendation System of E-Commerce based on Big Data Analysis," *Journal of Interdisciplinary Mathematics*, vol. 21, no. 5, pp. 1243-1247, 2018. [CrossRef] [Google Scholar] [Publisher Link]

[3] Gerard Deepak, and J. Sheeba Priyadarshini, "A Hybrid Semantic Algorithm for Web Image Retrieval Incorporating Ontology Classification and User-Driven Query Expansion," *Advances in Big Data and Cloud Computing*, pp. 41-49, 2018. [CrossRef] [Google Scholar] [Publisher Link]

[4] J. Bhavithra, and A. Saradha, "Personalized Web Page Recommendation using Case-Based Clustering and Weighted Association Rule Mining," *Cluster Computing*, vol. 22, pp. 6991-7002, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[5] Sandeep K. Raghuwanshi, and R.K. Pateriya, *Collaborative Filtering Techniques in Recommendation Systems*, Data, Engineering and Applications, pp. 11-21, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[6] Nymphia Pereira, and Satishkumar L. Varma, "Financial Planning Recommendation System using Content-Based Collaborative and Demographic Filtering," *Smart Innovations in Communication and Computational Sciences*, pp. 141-151, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[7] Virendra Singh Rathore, and Nidhi Singh, "Plan of Proficient URL Based Web Page Classification Utilizing NLP," *International Journal of Research in Engineering, Science and Management*, vol. 2, no. 10, pp. 119-121, 2019. [Google Scholar] [Publisher Link]

[8] Charul Nigam, and A.K. Sharma, "Experimental Performance Analysis of Web Recommendation Model in Web usage Mining using KNN Page Ranking Classification Approach," *Materials Today: Proceedings*, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[9] Safae Lassri, El Habib Benlahmar, and Abderrahim Tragha, "Web Page Classification Based on an Accurate Technique for Key Data Extraction," *International Conference on Advanced Intelligent Systems for Sustainable Development*, pp. 1124-1131, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[10] Mei Wang, and Weihong Deng, "Deep Face Recognition: A Survey," *Neurocomputing*, vol. 429, pp. 215-244, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[11] Yaniv Taigman et al., "DeepFace: Closing the Gap to Human-Level Performance in Face Verification," *2014 IEEE Conference on Computer Vision and Pattern Recognition*, Columbus, OH, USA, pp. 1701-1708, 2014. [CrossRef] [Google Scholar] [Publisher Link]

[12] Florian Schroff, Dmitry Kalenichenko, and James Philbin, "FaceNet: A Unified Embedding for Face Recognition and Clustering," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 815-823, 2015. [Google Scholar] [Publisher Link]

[13] O. Parkhi, A. Vedaldi, and A. Zisserman, "Deep Face Recognition," *Proceedings of the British Machine Vision Conference*, pp. 1-12, 2015. [Google Scholar] [Publisher Link]

[14] Kaipeng Zhang et al., "Joint Face Detection and Alignment Using Multitask Cascaded Convolutional Networks," *IEEE Signal Processing Letters*, vol. 23, no. 10, pp. 1499-1503, 2016. [CrossRef] [Google Scholar] [Publisher Link]

[15] Iacopo Masi et al., "Do We Really Need to Collect Millions of Faces for Effective Face Recognition?," *14th European Conference on Computer Vision (ECCV)*, Amsterdam, The Netherlands, pp. 579-596, 2016. [CrossRef] [Google Scholar] [Publisher Link]

[16] Jiankang Deng et al., "ArcFace: Additive Angular Margin Loss for Deep Face Recognition," *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Long Beach, CA, USA, pp. 4685-4694, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[17] Kaiming He et al., "Deep Residual Learning for Image Recognition," *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, NV, USA, pp. 770-778, 2016. [CrossRef] [Google Scholar] [Publisher Link]

[18] Weiyang Liu et al., "SphereFace: Deep Hypersphere Embedding for Face Recognition," *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Honolulu, HI, USA, pp. 212-220, 2017. [CrossRef] [Google Scholar] [Publisher Link]

[19] Xiangyu Zhu et al., "High-Fidelity Pose and Expression Normalization for Face Recognition in the Wild," *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Boston, MA, USA, pp. 787-796, 2015. [CrossRef] [Google Scholar] [Publisher Link]

[20] Meina Kan, Shiguang Shan, and Xilin Chen, "Multi-View Deep Network for Cross-View Classification," *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, NV, USA, pp. 4847-4855, 2016. [CrossRef] [Google Scholar] [Publisher Link]

[21] Shengcai Liao, Anil K. Jain, and Stan Z. Li, "Partial Face Recognition: Alignment-Free Approach," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 5, pp. 1193-1205, 2013. [CrossRef] [Google Scholar] [Publisher Link]

[22] Zinelabidine Boulkenafet, Jukka Komulainen, and Abdenour Hadid, "Face Anti-Spoofing Based on Color Texture Analysis," *2015 IEEE International Conference on Image Processing (ICIP)*, Quebec City, QC, Canada, pp. 2636-2640, 2015. [CrossRef] [Google Scholar] [Publisher Link]

[23] Rajeev Ranjan et al., "An All-in-One Convolutional Neural Network for Face Analysis," *2017 12th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2017)*, Washington, DC, USA, pp. 17-24, 2017. [CrossRef] [Google Scholar] [Publisher Link]

[24] Munawar Hayat, Mohammed Bennamoun, and Senjian An, "Deep Reconstruction Models for Image Set Classification," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 4, pp. 713-727, 2015. [CrossRef] [Google Scholar] [Publisher Link]

[25] Yangjun Mao et al., "Improving Reference-Based Distinctive Image Captioning with Contrastive Rewards," ACM Transactions on Multimedia Computing, Communications and Applications, vol. 20, no. 12, pp. 1-24, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[26] Dong Chen et al., "Blessing of Dimensionality: High-Dimensional Feature and Its Efficient Compression for Face Verification," *2013 IEEE Conference on Computer Vision and Pattern Recognition*, Portland, OR, USA, pp. 3025-3032, 2013. [CrossRef] [Google Scholar] [Publisher Link]

[27] Changxing Ding, and Dacheng Tao, "Robust Face Recognition via Multimodal Deep Face Representation," *IEEE Transactions on Multimedia*, vol. 17, no. 11, pp. 2049-2058, 2015. [CrossRef] [Google Scholar] [Publisher Link]

[28] George Trigeorgis et al., "Adieu Features? End-to-End Speech Emotion Recognition Using a Deep Convolutional Recurrent Network," *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Shanghai, China, pp. 5200-5204, 2016. [CrossRef] [Google Scholar] [Publisher Link]

[29] Karen Simonyan, and Andrew Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition," *arXiv*, pp. 1-14, 2015. [CrossRef] [Google Scholar] [Publisher Link]