Emotion Recognition using PIZAM-ANFIS by Considering Partial Occlusion and Behind the Mask

Jyoti S. Bedre[1*], Pasupuleti Lakshmi Prasanna[1]

[1]Computer Science and Engineering, KL University, Vaddeswaram, Andhra Pradesh (India)
E-mail: jyoti.phd2020@gmail.com
**Corresponding Author*

Abstract: Emotional expressions, comprising both verbal and non-verbal cues, communicate an individual's emotional state or attitude to others. To understand the complex human behavior, it is essential to analyze physical features across multiple modalities. Recent research has extensively focused on spontaneous multi-modal emotion recognition for human behavior analysis. Nonetheless, accurate Facial Emotion Recognition (FER) is hindered by challenges such as partial facial occlusions from random objects and mask-wearing. This paper proposes a novel classification method, Pizam-ANFIS-based FER, which addresses these issues by incorporating Occlusions and Masks (PAFEROM). The process begins with pre-processing the input image, followed by face detection and cropping using the Viola-Jones Algorithm (VJA). Skin tone analysis and segmentation of facial parts are performed using Local Structural Weighted K-Means Clustering (LSW-KCM). Subsequently, contour formation and edge detection via CGED are conducted, leading to feature extraction. The retrieved features' dimensionality is reduced using PIGA before being processed by CSE for Action Unit (AU) identification. Finally, PizMamdani-Adaptive Neuro Fuzzy Interference System (Pizam-ANFIS) classifies the identified AUs, and reduced-dimensionality features to determine human emotions. Experimental results indicate that the proposed model surpasses existing techniques in both efficacy and accuracy, providing a robust solution for FER in the presence of occlusions and masks.

Keywords: *Local Structural Weighted K-Means Clustering (LSW-KMC) algorithm,* Canny *Gaussian Edge Detector (CGED), PizMamdani (Pizam)-Adaptive Neuro Fuzzy Interference System (Pizam-ANFIS), Correlated Swish Embedding Network (CSE).*

1. Introduction

Facial expressions play a crucial role in human communication by providing essential nonverbal information that complements verbal interactions. Studies suggest that a significant portion of communication, ranging from 60% to 80%, is conveyed through nonverbal cues. These include facial expressions, eye contact, vocal tone, hand gestures, and physical distance [1, 2]. Analyzing these facial expressions has garnered significant attention in research, particularly in the field of FER. FER technology is increasingly utilized in human-computer interaction (HCI) applications, including autopilot systems, education, medical and psychological treatments, surveillance, and psychological analysis in computer vision [3]. By examining human facial expressions, FER systems aim to detect specific emotions such as anger, disgust, fear, happiness, sadness, surprise, and neutral states. The complexity of accurately estimating emotions is heightened by the diversity of human facial features and the variety of possible emotional expressions [4].

Automated facial expression recognition has garnered significant interest in recent years due to its broad spectrum of applications [5]. However, achieving high accuracy in recognizing facial expressions remains challenging because of their subtlety, complexity, and diversity [6].

A critical aspect of effective facial expression recognition is obtaining precise facial representations from the original facial images [7]. This system has two tasks: face detection and facial emotion classification. To extract significant and unique facial features, the human face is first recognized from the acquired image [8]. Then, the emotion represented by the identified face is classified using a FER algorithm. Formerly, researchers have tackled FER using various approaches such as the Multi-layer Perceptron Model, k-nearest Neighbors, and Support Vector Machines (SVM) [9]. These algorithms extracted information through Local Binary Patterns, Eigenfaces, Face-Landmark, and Texture features. Among these techniques, neural networks have gained the most popularity and are now extensively employed for FER [10]. Currently, advanced classifiers like Artificial Neural Networks (ANN), Convolutional Neural Networks (CNN), and Random Forests (RF) are widely used for tasks such as healthcare recognition, biometric identification, handwriting analysis, and facial detection for security purposes [11, 12]. However, achieving precise emotion classification with state-of-the-art classifiers in FER remains challenging due to issues like partial occlusion and the use of masks, which often need to be adequately addressed.

1.1 Problem Statement

Listed below are some of the shortcomings of the existing research approaches used to date:

1. Although current facial expression classifiers have proven practically flawless in analyzing confined frontal faces, they need to improve when analyzing faces that are partially obscured or hidden behind masks, which are frequently seen in the wild.
2. When wearing a face mask that covers the mouth and nose, it is impossible to accurately identify facial expressions of emotion. Classifying facial emotions using the half-face is more complex and challenging since the mouth area is one of the significant variables responsible for emotion detection.
3. Current FER techniques for masked faces often disregard significant facial areas like the forehead. Instead, they isolate only the eye region using landmark detection methods, which ultimately reduces the accuracy of the FER system.

This research suggests an improved FER system using a novel Pizam-ANFIS classifier to overcome these issues. The key research objectives of this system are outlined as follows:

1. A novel Edge Detector has been developed to detect the exact boundaries of organs.
2. A novel dimensionality reduction model is employed to select the interest features to mitigate training time.
3. A novel neural network is employed
4. to categorize the AU present in the mask-covered facial image.
5. A rule-based novel technique is utilized to classify human emotions.

The organization of this paper is as follows: Section 2 offers an in-depth review of related work, emphasizing significant advancements and challenges within the field. Section 3 outlines the proposed methodology and describes the innovative techniques and algorithms. Section 4 presents and discusses the results of the proposed method, focusing on performance metrics and comparative analysis. Lastly, Section 5 concludes the paper by summarizing the findings and suggesting potential directions for future research.

2. Literature Survey

In the realm of facial emotion recognition (FER), Mehendale *et al.* [13] proposed a modular framework using an AdaBoost cascade classifier for face detection and extracting Neighborhood Difference Features (NDF), which were classified with a random forest

classifier to address false detections. Despite outperforming reference methods on the SFEW and RAF datasets, the system's omission of geometric elements led to inaccuracies. Liu *et al.* [14] introduced an FER technique that leveraged landmark curvature and vectorized landmarks, combining SVM classification with a genetic algorithm for feature and parameter selection. While this approach showed balanced performance on the CK+ and MUG datasets, image noise impacted the SVM classifier's accuracy. Alreshidi *et al.* [15] employed Nonlinear Principal Component Analysis (NLPCA) for dimensionality reduction and SVM for emotion classification, achieving high accuracy but struggling with varying input dimensions. Hassan *et al.* [16] utilized graph mining techniques to identify common sub-graphs within emotional classes, enhancing efficiency and accuracy but resulting in a more time-consuming process. Hussain *et al.* [17] developed a deep learning-based FER system structured in three phases: face detection, feature analysis using Keras CNN, and emotion classification. Although this system demonstrated proficiency, errors in facial landmark detection impacted overall accuracy. Houshmand *et al.* [18] proposed a transfer learning approach with pre-trained VGG and ResNet networks for FER under VR headset occlusion, achieving comparable performance but needing refinement in preprocessing steps due to
issues with histogram equalization.

Monisha *et al.* [19] introduced a real-time FER system using CNN for classification, demonstrating high accuracy but encountering recognition errors due to limited training data. Akhand *et al.* [20] utilized transfer learning within a Deep Convolutional Neural Network (DCNN), progressively enhancing FER accuracy but failing to preserve edge information crucial for detailed emotion recognition. Saha *et al.* [21] employed the Cosine Similarity-Based Harmony Search Algorithm (SFHSA) for feature selection, optimizing feature vectors and improving classification accuracy, albeit with a time-consuming training process. Gautam *et al.* [22] combined HOG and SIFT for feature extraction with CNN for classification, outperforming existing methods but struggling with the limitations of 2D data in handling facial pose variations. Castellano *et al.* [23] focused on recognizing emotions from masked faces using ResNet, achieving high accuracy with eye region analysis but increasing computational demands due to skip connections. Wally *et al.* [24] developed an Occlusion-Aware Student Emotion Recognition system utilizing CNN and FCNN, which faced overfitting issues due to limited data. Elsayed *et al.* [25] showcased a hybrid CNN with LBP for feature extraction in masked faces, demonstrating improved recognition but facing challenges with imbalanced and noisy data. Mukhiddinov *et al.* [26] applied synthetic masks to input images, emphasizing head and upper facial features for FER, achieving higher accuracy but encountering orientation issues with landmark features. Finally, Zhu *et al.* [27] introduced HDCNet, leveraging a feature constraint methodology to mine attention consistency features, improving classification accuracy but posing substantial computational demands due to Class Activation Mapping.

3. Proposed Framework for FER

This research proposes a novel Pizam-ANFIS for effective human emotion recognition using visual features. Two key processes—face detection and classification—are finished in order to identify the facial mood. Features from the face are retrieved and fed into a trained network for emotion classification. The block diagram for the suggested model is illustrated in Figure 1.

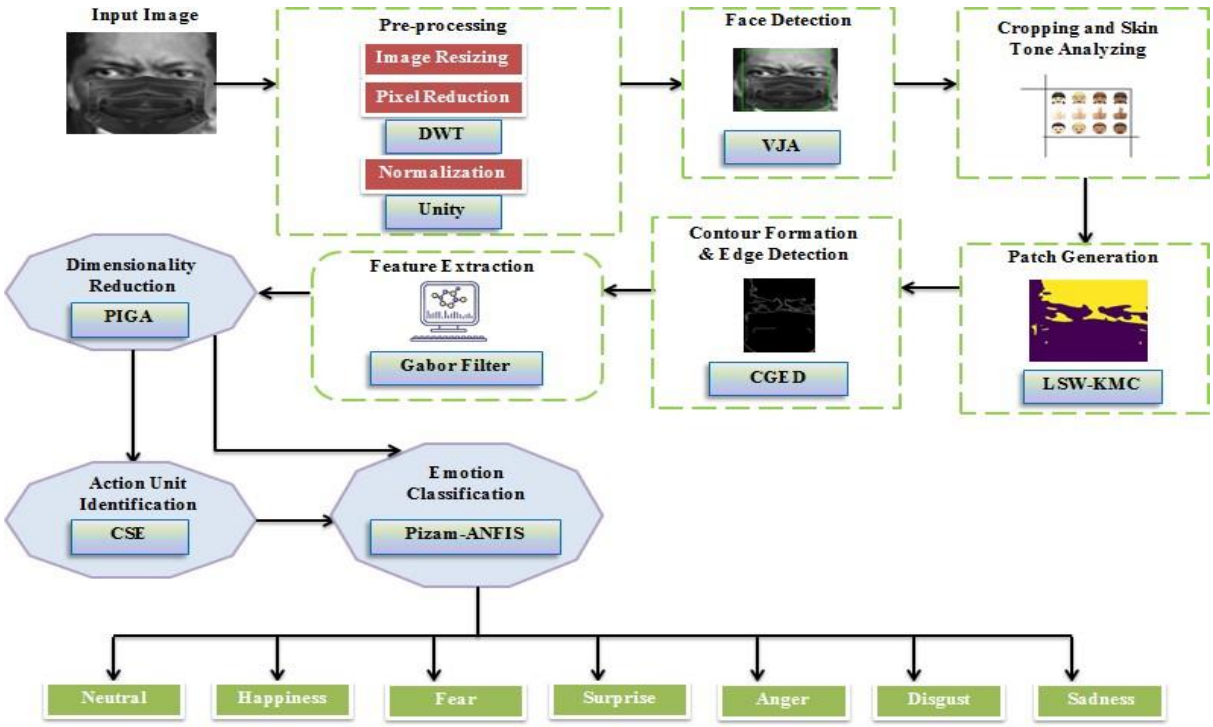

Figure 1: Schematic of the projected framework

3.1 Pre-processing

In this section, an Image with emotion is taken as input and further injected into the pre-processing because of the presence of unwanted things. The Input Face Expression Image undergoes the pre-processing operation in three stages: image resizing, pixel reduction, and normalization.

3.1.1 Image resizing

The accuracy and computation time of the processing system can be adversely affected by unwanted pixels in the input image. The input image (I) is resized to 256×256 pixels using bilinear interpolation to address this. This method is particularly recommended for continuous data sets lacking distinct boundaries. Bilinear interpolation is a resampling method that computes a new pixel value by averaging the four nearest pixel values, weighted by their distances. This technique provides a smoother and more precise representation of the image. The resized image (I_{resize}) is,

$$I_{resize} = \frac{\Psi^R S^L + \Psi^L S^R + \Psi^T S^B + \Psi^B S^T}{\Psi^R + \Psi^L + \Psi^T + \Psi^B} \quad (1)$$

Here, ψ^R, ψ^L, ψ^T, and ψ^B are the corresponding distances from the missing pixel, and S^L, S^R, S^B, and S^T are the left, right, top, and bottom source pixels.

3.1.2 Pixel Reduction

After resizing the image, we remove noisy pixels from the resized image (I_{resize}) by utilizing the Discrete Wavelet Transform (DWT). DWT is selected due to its ability to achieve a higher

compression ratio. This process involves decomposing the image into coefficients (sub-bands) and then compared to a set threshold (T_{thres}). The coefficients that fall below this threshold are considered noiseless pixels and are retained in the image. In contrast, those above the threshold are identified as noisy pixels and are subsequently removed. This method ensures that only the low-low frequency sub-bands, which contain the essential image information with reduced noise, are preserved. The resulting pixel-reduced image (I_{red}) can be represented as follows:

$$I^{red} = \begin{cases} noisy\ pixel, if\ (I^{red})^{\rho} > T_{thres} \\ noiseless\ pixel, if\ (I^{red})^{\rho} < T_{thres} \end{cases} \quad (2)$$

3.1.3 Normalization

Unity normalization transforms the pixel-reduced image (I^{red}) into a range of pixel values. Unity normalization has better and faster execution. In order to reduce the inner-class feature mismatch, which can be seen as intensity offsets, image normalization is a crucial pre-processing approach. The normalized image can be denoted as I^{nor}.

$$I^{nor} = \frac{I^{red}}{||v||} \quad (3)$$

Here, $||v||$ denotes the vector of the pixels.

3.2 Face Detection

In this step, we detect the face from the pre-processed image using the VJS to facilitate the determination of the region of interest and subsequent feature extraction. The VJS process entails sliding feature boxes across the image and computing the difference in the summed pixel values between adjacent regions, represented as (d). This difference is then compared to a threshold value (T_f) to determine if an object, such as a face, has been detected. This method simplifies the identification of the region of interest and ensures accurate feature extraction from the detected face. The detected face (I^{face}) is computed as follows,

$$T_f = \begin{cases} face\ detected\ if\ T_f > d \\ not\ detected\ if\ T_f < d \end{cases} \quad (4)$$

3.3 Cropping and skin tone analysis

The detected face image is cropped to remove all the unwanted things from the image, such as the background, and to keep only relevant information in the image. After that, skin tone analysisis done to differentiate the parts presented over the face. Then, the image of the skin analyzed is denoted as I^{skin}.

3.4 Patch Generation

In patch generation, the different facial parts are segmented from I^{skin} to encourage extracting discriminative features from the minute parts using the LSW-KMC algorithm. K means is favored over other segmentation methods because of its ease of use and rapid computation speed. However, the spatial Euclidean distance-based characterization of the relationship between the image pixels and cluster center is more difficult since this distance alone is insufficient to understand the general characteristics. In order to get over the drawbacks above, the weighted sum of the image pixels was used to estimate the distance between each image pixel and the cluster center. After that, the structural similarity index calculates a local distance measurement to determine how far apart two image pixels are from one another in the overall image. This local distance computation reflects not only the physical relationship between two

picture pixels but also the relationship connected to luminance and contrast, as well as the structure of the image pixels revolving around them. As a result, LSW-KMC serves as the inspiration for the proposed KMC. The steps of LSW-KMC are listed as:

a) Initializing the pixels $\rho^j \in I^{skin}$, presented as,

$$\rho^j = \{\rho^1, \rho^2, \rho^3, \ldots\ldots \rho^N\} where, j = 1,2.,3, \ldots\ldots N \quad (5)$$

Here, j denotes the count of pixels of the skin tone detected image.

b) Select the number of clusters that are defined by their centroids. Initially, the precise centers of the pixels are unknown, so the centroids C_m are chosen randomly to establish each cluster.

$$C_i = C_1, C_2, \ldots\ldots C_M \qquad i = 1,2, \ldots\ldots\ldots\ldots M \quad (6)$$

Here, i represents the centroid (cluster centre).

c) Calculate the weighted sum of the image pixels(⬚) by considering the essential distance $\left(d\left(\rho^j,\ C_i\right)\right)$.

$$S = \sum_{r=1}^{Z} W_R d\left(\rho^{jr}, C_{ir}\right) \quad (7)$$

Here, W_R denotes the weight associated with the distance $\left(d\left(\rho^j,\ C_i\right)\right)$, ρ^{jr} represents the value of the point in the image located around the ρ^j, C_{ir}denotes centroids, and Z denotes the number of points in the skin tone detected image.

d) W_R is determined by looking at the coordinate distance ⬚between $\rho^{jr}$and ρ^j. Therefore, the weights are,

$$W_R = \frac{1}{(1+d_r)^{Cpara}} \quad (8)$$

Here, C_{para} represents the control parameter.

e) Measure the structural similarity of the image. It considers the degree of similarity of luminance, contrast, and structure of the pixel and cluster center. The SSIM index$(D \in S)$ between pixels and cluster center is defined as,

$$D = \frac{\left(2\lambda_{\rho^j}\lambda_{C_i}+\chi_1\right)\left(2\sigma_{\rho^j C^i}+\chi_2\right)}{\left(\lambda_{\rho^j}{}^2\lambda_{C_i}{}^2+\chi_1\right)\left(\sigma_{\rho^j}{}^2\sigma_{C_i}{}^2+\chi_2\right)} \quad (9)$$

Here, λ_{ρ^j} and λ_{C_i} denote the mean of $\rho^j$and C_i respectively, $\sigma_{\rho^j C^i}$ signifies the cross-correlation between $\rho^j$and C_i, $\sigma_{\rho^j}{}^2$ and $\sigma_{C_i}{}^2$specifies standard deviation of $\rho^j$and C_i, respectively, χ_1 and $\chi_2$are the positive constants.

f) Assign each pixel to a cluster where the distance between the pixel and the centroid is minimized.

This process continues iteratively until the clusters stabilize and no further changes occur. This segmentation identifies and outlines standard and disease-affected regions in the resulting image, denoted as I_{seg}. The pseudocode for the proposed LSW-KMC means is:

Input: Face detected image I^{skin}
Output: Segmented image I^{seg}

Begin
Initialize ρ^n, number of clusters C_m, iteration $(iter)$, maximum iteration $(iter_{max}())$
Perform clustering
Select the number of centroids
Set $iter = 1$
While $iter \leq iter_{max}$
For each pixel, **do**
Calculate the weighted sum of image pixels
Compute distance *D*

$$D = \frac{\left(2\lambda_{\rho^i}\lambda_{C_i} + \chi_1\right)\left(2\sigma_{\rho^j C^i} + \chi_2\right)}{\left(\lambda_{\rho^i}{}^2\lambda_{C_i}{}^2 + \chi_1\right)\left(\sigma_{\rho^i}{}^2\sigma_{C_i}{}^2 + \chi_2\right)}$$

End for
Check all the pixels are presented under the cluster
If $(\rho^n == undercluster)$ **{**
Stop criteria
} Else {
Set $iter = iter + 1$
} End if
End While
Return segmented image
End

3.5 Contour Formation and Edge Detection

Here, the contour is formed over I^{seg} using CGED to extract the facial parts more effectively from the occluded and mask-covered input images. For simplicity, the existing Canny Edge Detection (CED) is chosen for the proposed work. However, a drawback of the CED is that the default Sobel Operators are restricted to a fixed 3-by-3 window. This limitation can be problematic, particularly in noisy images, potentially compromising the final output. Our work employs a broader 5-by-5 Sobel Operator window to address this issue. This adjustment aims to produce a smoother image and reduce susceptibility to noise, thereby enhancing the effectiveness of the CED. In addition, the horizontal and vertical gradients are calculated using a Gaussian kernel rather than CED's standard convolution kernel to save time. Denoise image before detecting the edge ofthe image usually use the 5-by-5 Sobel Operator to reduce noise, according to (10),

$$I^{den} = \sqrt{\beta_o^2 + \beta_t^2} \tag{10}$$

To calculate the gradient intensity (B), use the Gaussian kernel and determine the edge direction (ϕ). Typically, the gradient direction is categorized into four angles: 0, 45, 90, and 135 degrees. This process is defined by equations (11) and (12),

$$B = \exp\left(\frac{-\left\|\beta_o - \beta_t\right\|^2}{2\sigma^2}\right) \tag{11}$$

$$\varphi = \tan^{-1}\left(\frac{\beta_o}{\beta_\iota}\right) \tag{12}$$

Where, β_o and β_ι denote the pixel values in the o-axis and ι-axis, respectively, σ denotes the signum function. After the gradient and magnitude calculation, the entire image is scanned, unwanted pixel intensities are suppressed to 0, and the edges present are given as $E_\hbar, \hbar = 1,2,..., fin$. Next, the hysteresis threshold is selected as high(Up_l) and low(Lo_l). These thresholds analyze whether all the detected edges are edges or not. The thresholding function is given as,

$$I^{edge} = \begin{cases} Sure\,edge & if\ \hbar > Up_l \\ Valid\,edge & if\ Up > \hbar > Lo_l \\ non\,edge & else \end{cases} \tag{13}$$

Where $\hbar$ depicts the edge, if the edge $\hbar$ lies between, then Up_l and Lo_l connected to $Sure\,edge$ is considered a valid edge. If the edge $\hbar$ does not connect to the sure edges and below, then Lo_l it is removed from the image as a non-edge. Finally, the edge-detected image is denoted as I^{edge}.

3.5 Contour Formation and Edge Detection

After performing edge detection, the next step is to extract features to obtain detailed information from the input image. Texture features are extracted using the GF, a linear filter selected for its frequency and orientation representations that closely mimic the human visual system. The GF consists of a sinusoidal plane wave modulated by a Gaussian kernel function. According to the convolution theorem, the Fourier transform of a harmonic function and the Fourier transform of a Gaussian function combine to produce the impulse response of a Gabor filter. This filter captures orthogonal directions with both real and imaginary components. The process involves applying the GF to the input image to obtain the sinusoidal plane wave response, modulating this response with the Gaussian kernel function to capture both frequency and orientation information, and combining the Fourier transforms of the harmonic and Gaussian functions to generate the GF's impulse response. The real and imaginary components representing orthogonal directions are then extracted. These Gabor features (f_1) are crucial for accurately capturing the texture information from the image, thereby enhancing the overall feature extraction process.

$$f_1 = \exp\left(-\left(\rho^i\right)^2 + \left(\rho^i\right)^2 / 2\varpi^2\right) * \cos\left(2\pi / \lambda\right)\rho^i \tag{14}$$

Here, λ and ϖ denotes the wavelength and effective width, respectively. Additionally, various features such as geometrical features, appearance features, temporal features, HOG, SIFT, and Speeded-Up Robust Features (SURF) are extracted. The comprehensive set of extracted features (f_k) can be summarized as follows:

$$f_k = \{f_1, f_2, f_3, ... f_K\} \quad where, k = 1,2,3,.....K \tag{15}$$

Here, K denotes the number of features.

3.7 Dimensionality reduction

In this step, the dimensionality of features is reduced f_k to a lower-dimensional space using PIGA, which selects the most critical features to minimize training time during classification. Principal Component Analysis (PCA) is employed for its straightforward computation process and ability to eliminate correlated features. Principal Components aim to capture the maximum variance among the features. However, traditional PCA may lose some information compared to the original feature set due to the arbitrary selection of principal components. The research incorporates the Information Gain (IG) mechanism to address this issue and determine the optimal number of principal components. IG is an entropy-based feature estimation method that evaluates each feature individually, calculates its information gain, and assesses its importance on the class label. Each extracted feature is assigned a score ranging from 1 to 0, indicating its relevance from most to least important for setting the number of principal components. This approach ensures that principal component selection is fair and effective, preserving essential information while reducing dimensionality.

Covariance matrix construction: The PIGA constructs a covariance matrix for the recognition process to get the eigenvectors. The covariance matrix $(\Re)$ construction is formulated as,

$$\Re = \frac{1}{K}\sum_{k=1}^{K}(f_k)(f_k)^T \tag{16}$$

Where, $(\ )^T$ depicts matrix transpose.

Eigenvalue calculation: The eigenvalue is calculated from the features as,

$$E = \vartheta((1/K)\times f_k) \tag{17}$$

Where, E depicts the eigenvalue and $\vartheta(\ )$ depicts the decomposition function, which is given as,

$$\vartheta = D_{co}D_{main} \tag{18}$$

Here, $D_{co}D_{main}$ depicts the decomposition of two matrices of the features.

Eigenvector estimation: For the features with high eigenvalues, the eigenvector (V) is calculated using the formula,

$$V = \Re - \varsigma.E \tag{19}$$

Here, ς indicates a random constant value.

Obtaining Principal Components: After the eigenvalues are estimated, the features with high Eigenvalues are derived as the principal components. The Principal components are calculated using IG,

$$p_{com} = V \times \phi_{cen} \tag{20}$$

Where, ϕ_{cen} depicts the kernel center. Thus, the selected features $\left(F_z^{sel}\right)$ are given as,

$$F_z^{sel} = \left[F_1^{sel}, F_2^{sel}, F_3^{sel}, ..., F_Z^{sel}\right] \quad (21)$$

Where Z^{th} represents the number of features.

3.8 Action Unit Identification

Here, the CSE network determines the human AUs F_z^{sel} for quickly identifying emotions during training. Human action units encompass various expressions and movements such as slit, eyes closed, squint, blink, wink, and others. They also incorporate actions such as raising the inner brow, raising the outer brow, lowering the brow, raising the upper lid, wrinkling the nose, raising the cheeks, tightening the lids, and drooping the lids. CNN is employed for its ability to handle high-dimensional data without significant information loss. However, in existing CNNs, many neurons still need to be updated because the ReLU activation function does not preserve negative values due to its monotonic and linear nature. The proposed method utilizes Hard Swish (HS), which is nonmonotonic and smooth, to address this issue. The nonmonotonic property of HS stabilizes the network's gradient, allowing it to maintain small negative values. Additionally, the CNN's embedding and correlated interference modules are crucial for effective recognition. These enhancements ensure that the network can better capture and process the nuances of human action units, leading to more accurate and robust recognition. The correlated interference module received and processed the discriminative AU features' estimations from the embedded module. It calculates the correlations between the differentiating characteristics. As a result, the planned CNN is known as CSE.

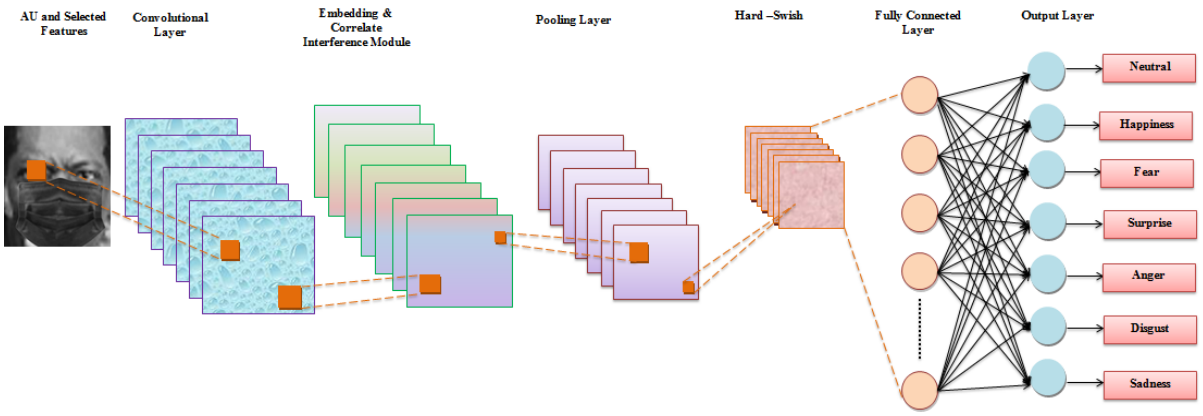

Figure 2: Architecture of the proposed CSE

3.8.1 Input layer: The input layer of a neural network is composed of artificial input neurons that feed the initial data into the system, setting the stage for processing by the successive layers of artificial neurons.

3.8.2 Convolution *Layer:* In the convolution layer, an element-wise product is calculated between each element of the kernel and the input array at every position within the tensor. These products are subsequently summed to generate the output value for the corresponding location in the output array. This process is repeated with multiple kernels to generate diverse feature maps. Then, convolution (L_{con}) is expressed as,

$$L_{con} = \sum_u \sum_u \left(F_z^{sel}\right)(g-u, h-u) * w(u,u) \quad (22)$$

Where, g and h are the input matrix dimension size, $w(u,u)$ represents the kernel having $u \times u$

dimension size. The convolution parameters can reduce the model's complexity.

3.8.3 Nonlinear activation function: The HS activation function is used for this purpose. The main task of using nonlinearity is to adjust or cut off the generated convolution output. The activation function is expressed in the mathematical representation as,

$$A = L_{con}\frac{R6\left(L_{con}+3\right)}{6} \tag{23}$$

Where, A denotes the output of the HS activation function and R denotes the ReLu activation function.

3.8.4 Embedding Module and Correlated Interference Module: In this step, the features derived from the nonlinear function are fed into the embedding module. This module utilizes a deeper convolutional network as a feature extraction mechanism, enhancing the capacity for feature representation by extracting discriminative AU features. Then, the output of the embedding module is calculated as,

$$A^{emb} = Embed\left\{(A) * w(u,u)\right\} \tag{24}$$

Here, $Embed\{\ \}$signifies the embedding function. The discriminative AU features are given into the correlated interference module, which efficiently calculates the correlation between the features, and it is represented as,

$$A^{corr} = \frac{\sum x^{A^{corr}} y^{A^{corr}} - \sum x^{A^{corr}} \sum y^{A^{corr}}}{\sqrt{\left(\sum\left(x^{A^{corr}}\right)^2 - \left(\sum x^{A^{corr}}\right)^2\right)\left(\sum\left(y^{A^{corr}}\right)^2 - \left(\sum y^{A^{corr}}\right)^2\right)}} \tag{25}$$

Here, A^{corr} specifies the output of the correlated interference module.

3.8.5 Pooling Layer: A pooling layer executes a standard down-sampling action that reduces the in-plane dimensionality of the feature maps. It outputs the maximum value found within the pooling filter, using this value as the result. The pooling (L_{pool}) operation can be expressed as:

$$L_{pool} = \frac{A^{corr} - w}{S} + 1 \tag{26}$$

Where, S represents the kernel strides. The process keeps on going until it reaches the last layer.

3.8.6 Fully connected layer: The final convolution or pooling layer's output feature maps are often flattened, becoming a one-dimensional array of numbers. The last completely linked layer has an equal number of output nodes corresponding to the number of classes. Calculating the flattened output as,

$$L_{fully} = L_{pool} - (w(u \times u) - 1) \tag{27}$$

Where, L_{fully} is the output of the fully connected layer.

3.8.7 Softmax layer:

The activation function, primarily used in the output layer, normalizes the real values in the range (0, 1) from the last fully connected layer into target class probabilities. This is achieved using the softmax function, which is defined by the following equation,

$$L_{soft} = \frac{e^{L_{fully}}}{\sum L_{fully}} \tag{28}$$

Where, L_{soft} is the output of the softmax activation function. Later, the loss function is evaluated using the below equation,

$$loss = \left\| O^{target} - L_{soft} \right\| \tag{29}$$

Here, O^{target} specifies the target output. Finally, the identified AU is denoted as (L_{soft}). The pseudocode of the proposed CSE is,

---

Input: Dimension reduced features $\left(F_z^{sel}\right)$

Output: Action units $\left(L_{soft}\right)$

Begin

Initialize parameters L_{con} , $w(u,u)$ L_{pool}

Compute weight value

While $j = 1 to Z$

For $j = 1$

Compute convolution operation η

Evaluate activation function

$$A = L_{con} \frac{R6\left(L_{con} + 3\right)}{6}$$

Compute Embedding Module

Perform Correlated Interference Module

End for

While $j = 2 to Z$

For $j = 2$

Compute convolution operation η

Evaluate activation function

$$A = L_{con} \frac{R6\left(L_{con} + 3\right)}{6}$$

Compute pooling operation L_{pool}

End for

End while

Flattening all the layers

Evaluate softmax activation function L_{soft}

If $\left(O^{target} \neq O^{L_{soft}}\right)$

Stop criteria

} **else {**

Set $iter = iter + 1$

} End if

Return L_{soft}

End

3.9 Emotion Classification

The Pizam-ANFIS is used to categorize the types of emotions by taking the input as selected features and action units from the occluded and mask-covered input images once the action units have been identified. The Adaptive Neuro-Fuzzy Inference System (ANFIS) is a computational and predictive model that integrates the fuzzy Sugeno method with an adaptive neural network system. However, the adapted Sugeno fuzzy interference system introduces computational complexity while designing the higher-order fuzzy models. To avoid this issue, the Mamdani fuzzy interference system in the defuzzification process is induced with modification in the existing ANFIS. It uses the center of gravity technique for the defuzzification process, and the bell membership is replaced with the Piz membership function, which reduces the computational complexity and produces effective outcomes.

Here, the second layer performs the fuzzification process, with the nodes in this layer being adaptive. The fuzzified output for the ι^{th} layer Φ_ι is,

$$\Phi_\iota = \mu_1\left(\eta_{W_h}\right) \tag{30}$$

$$\Phi_\iota = \mu_2\left(\eta_{W_v}\right) \tag{31}$$

Where, μ_1 and μ_2 represent input node, W_h and W_v denotes value of weights, η denotes Piz membership function (layer1), and it is calculated as,

$$\eta = \frac{1}{1+\left(\frac{AF^{points} - P^1}{P^2}\right)^2} \tag{32}$$

Here, $^{\text{points}}$ denote the feature and AU points. In the third layer, the output signals from the previous layers are multiplied. This layer processes the outputs from the second layer ε_ι, resulting in:

$$\varepsilon_\iota = \mu_1\left(\eta_{W_h}\right) * \mu_2\left(\eta_{W_v}\right) \tag{33}$$

The output of each node represents the firing strength of the rules. In the fourth layer, the output, described as the normalized firing strength $\left(\varepsilon_\iota\right)^*$, is mathematically represented using the Radial Basis Function (RBF) as follows,

$$\left(\varepsilon_\iota\right)^* = \sum_i \eta_{W_h} \zeta\left(\mu_2\left(\eta_{W_v}\right), \varepsilon_\iota\right) + b \tag{34}$$

Here, ζ and b denote kernel and bias. The consequent part of the fuzzy rules is executed in the fourth layer. The nodes in this layer are adaptive, and the node function is formulated as follows,

$$\left(\bar{\varepsilon}_t\right)^* = \frac{\left(\varepsilon_i\right)^*}{\left(\phi_i \mu_1 + a_i \mu_2 + L_i\right)} \tag{35}$$

Where, ϕ_i a_i and L_i denote linear adaptive parameters, $\left(\bar{\varepsilon}_t\right)^*$ represent defuzzification using the Mamdani interference system's defuzzification process. Finally, the last layer predicts the emotion of the human (Γ), and it is represented as,

$$\Gamma = \sum \left(\varepsilon_i\right)^* \left(\phi_i \mu_1 + a_i \mu_2 + L_i\right) \tag{36}$$

After training the proposed network, the image, which has to be tested, is given to the system for testing. By testing the data, the output layer classifies the emotions as Neutral, Happiness, Fear, Surprise, Anger, Disgust, and Sadness.

4. Results and Discussion

This section details the experiments performed on the PYTHON platform to validate the proposed scheme's performance. The experiments utilized a synthetic dataset created from publicly available sources. The dataset was split into two segments: 80% of the images were allocated for training, while the remaining 20% were reserved for testing. Sample images from the dataset were processed and incorporated into the operation, as depicted in Figure 3.

Figure 3: Sample images of a human face with an emotion (a) input images (b) Face detectedimage (c) Patch generated image (d) Edge detected image

4.1 Performance analysis of proposed CSE-Pizam-ANFIS

To thoroughly assess channel estimation performance, the anticipated CSE-Pizam-ANFIS algorithm was benchmarked against several well-established methods. These included the

ANFIS, CNN, Long Short-Term Memory (LSTM) network, and ANN. The efficacy and advantages of the CSE-Pizam-ANFIS approach in channel estimation were effectively validated by conducting a comprehensive comparison with these existing algorithms.

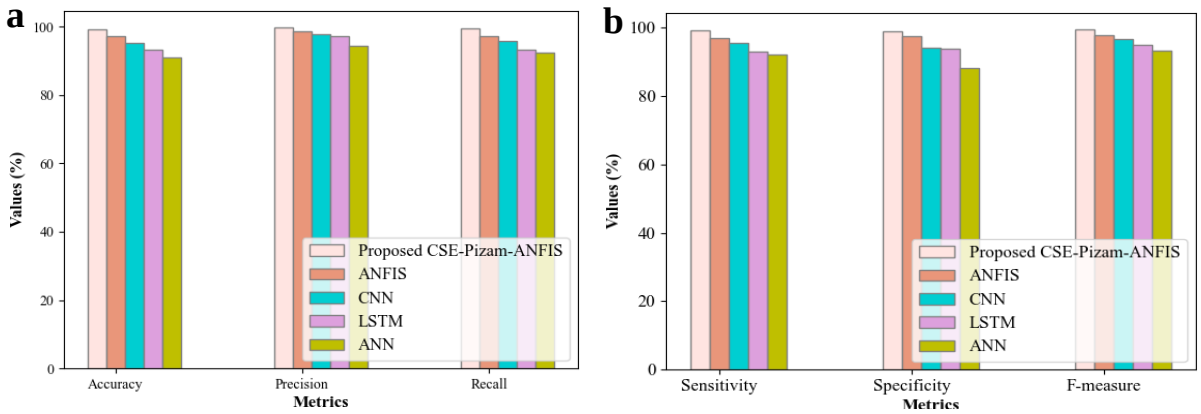

Figure 4: Illustrative comparison of the proposed and existing models (right-hand side): (a) Accuracy, Precision, Recall metrics, (b) Sensitivity, Specificity, and F-measure parameters.

Figure 4 presents a detailed assessment of the proposed CSE-Pizam-ANFIS model's performance in comparison to existing models, focusing on key metrics such as accuracy, precision, recall, sensitivity, specificity, and f-measure. Higher values in these metrics indicate more efficient model performance. The proposed CSE-Pizam-ANFIS model achieves an impressive accuracy of 99.28%, which is notably higher than the accuracy of the existing models: ANFIS at 97.22%, CNN at 95.24%, LSTM at 93.34%, and ANN at 90.97%.

In addition to accuracy, the proposed model excels in other metrics. It records a precision of 99.67%, a recall of 99.35%, a sensitivity of 99.35%, a specificity of 99.09%, and an f-measure of 99.51%. These values surpass those of the existing models, demonstrating the superior performance of the proposed model across all evaluated aspects. This comprehensive analysis underscores the effectiveness of the proposed model in AU classification and emotion classification tasks, significantly outperforming current alternatives.

Table 1: Performance evaluation of proposed and existing models

Techniques	FPR	FRR	FNR	PPV	NPV	MCC
Proposed CSE-Pizam-ANFIS	0.00900901	0.006451613	0.006452	0.996764	0.9821429	0.9871895
ANFIS	0.02484472	0.11764706	0.029412	0.985075	0.9515152	0.971719
CNN	0.05932203	0.04290429	0.042904	0.976431	0.8951613	0.8845861
LSTM	0.0610687	0.068965517	0.068966	0.971223	0.8601399	0.8544533
ANN	0.11764706	0.077192982	0.077193	0.942652	0.8450704	0.7963936

Table 1 provides a comprehensive performance evaluation of the proposed and existing models using various metrics: False Positive Rate (FPR), False Rejection Rate (FRR), False Negative Rate (FNR), Positive Predictive Value (PPV), Negative Predictive Value (NPV), and Matthews

Correlation Coefficient (MCC). Higher values of FPR, FRR, and FNR indicate improved performance of the proposed model, while lower values of PPV, NPV, and MCC demonstrate its higher efficiency. For example, the proposed model shows a 63.78% improvement in FPR compared to ANFIS, 84.81% compared to CNN, and 92% compared to ANN. Additionally, the FRR of the proposed model is 94.51% better than that of LSTM and other existing models. Similarly, the FNR, PPV, NPV, and MCC metrics for both the proposed and existing models have been analyzed and compared. This detailed analysis reveals the superior efficiency and performance of the developed AU and emotion recognition system.

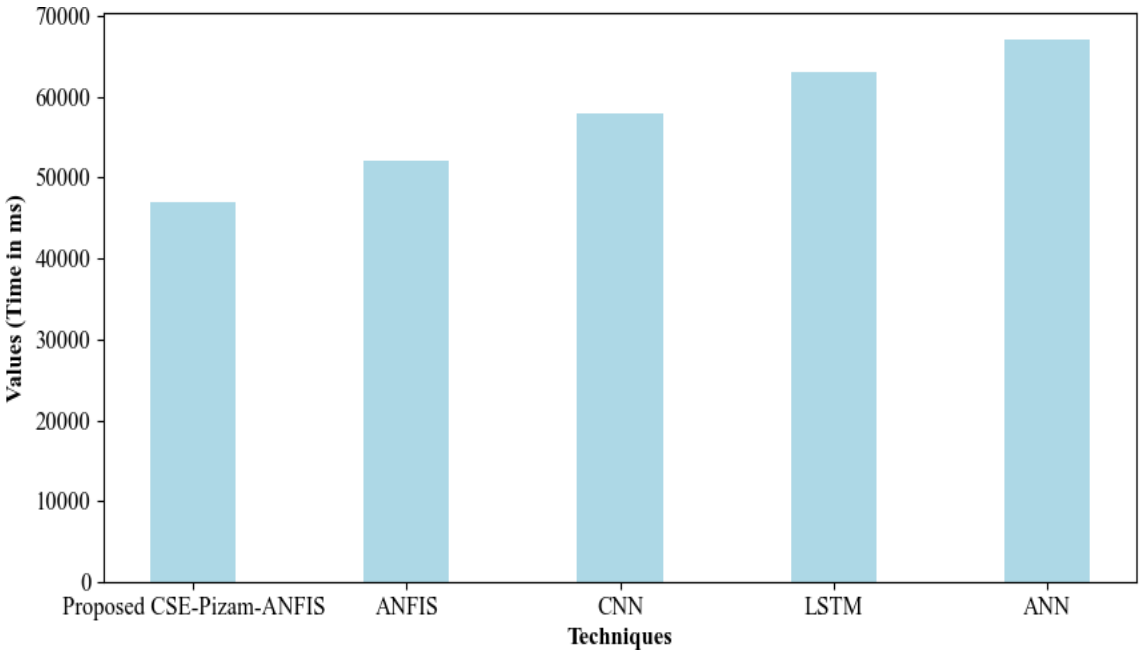

Figure 5: Computational time analysis

Figure 5 illustrates the computational time analysis, comparing the proposed and existing models. Attaining a Lower time for the proposed model indicates the efficient time of the proposed model. Here, the training time of the proposed model is 47015ms, whereas the existing ANFIS (52009ms), CNN (58006ms), LSTM (63006ms), and ANN (67010ms) take more time to train the proposed model. This can be achieved by inducing the HS and embedding a correlated interference module to stabilize the network's gradient and efficiently recognize action units. Additionally, the Piz membership function and the Mamdani defuzzification method were introduced, which aids in the classification of emotions for computational complexity.

4.2 Performance analysis of patch generation

To highlight the advantages of our proposed model, we evaluated the performance of the LSW-KMC. We compared it to existing models using metrics such as the Jaccard Index, Dice score, and Clustering Time.

Table 2: Jaccard Index

Method	Jaccard Index
Proposed LSW-KMC	0.03263298
K Means	-0.0690296
FCM	-0.068997
K Medoid	-0.0690228
CLARA	-0.069018

Table 3: Clustering time analysis

Method	Clustering Time (ms)
Proposed LSW-KMC	38010
K Means	43005
FCM	47011
K Medoid	53010
CLARA	56012

Table 2 presents a detailed analysis of the Jaccard Index for both the proposed and existing models. The Jaccard Index measures the similarity between pixel groupings across different clusters, with one indicating that two clusters have perfectly extracted the same pixels and 0 indicating no overlap. The proposed model achieved a Jaccard Index of 0.03263298, demonstrating superior performance compared to the existing models, which showed lower coefficients. This result underscores the enhanced effectiveness of our new patch generation technique in accurately identifying similar clusters.

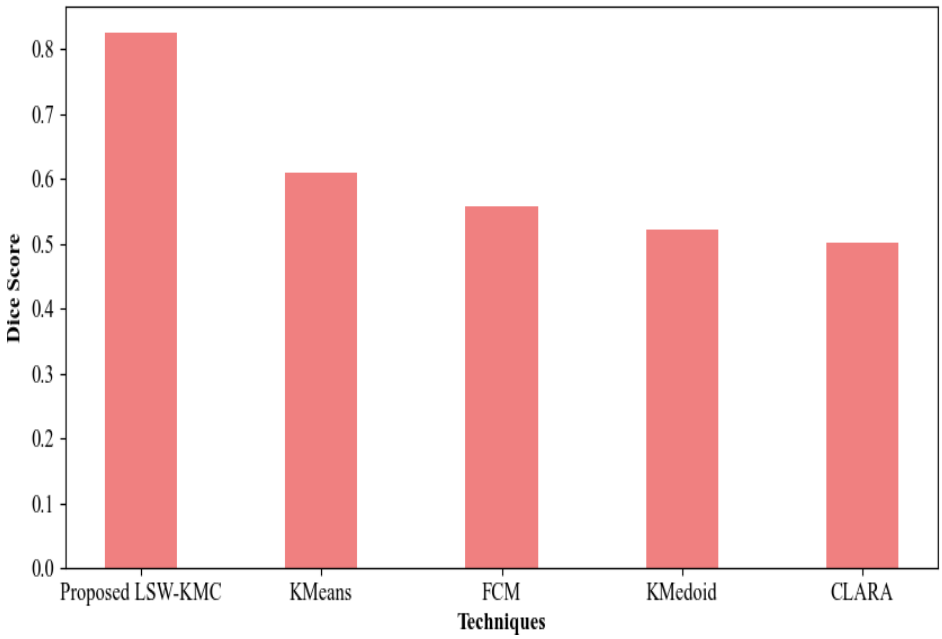

Figure 6: Dice score analysis.

Figure 6 comprehensively analyzes the Dice scores, comparing the proposed model with existing models. The Dice coefficient, which measures the pixel-wise agreement between predicted segmentation and the corresponding ground truth, shows that the proposed model achieved a Dice score of 0.8245. This performance significantly surpasses that of the existing models: K-Means with a score of 0.60973, Fuzzy c-Means (FCM) with 0.5582, K-Medoid with 0.52096, and Clustering Large Applications (CLARA) with 0.5007. This analysis highlights the superior performance of the proposed method. Furthermore, Table 3 presents the performance results, underscoring the proposed model's efficiency in terms of clustering time. The proposed model's clustering time is 38010ms, showing an improvement of 4995ms over K Means, 9001ms over FCM, and 18002ms over CLARA. This indicates that the LSW-KMC technique generates facial parts with greater accuracy and in a shorter time frame. The overall

success of the proposed model is attributed to the careful selection and modification of existing patch generation techniques, as established in previous studies. By refining these existing models, the proposed approach effectively generates accurate facial parts more efficiently.

4.3 Comparative Evaluation of the Suggested and Earlier Approaches

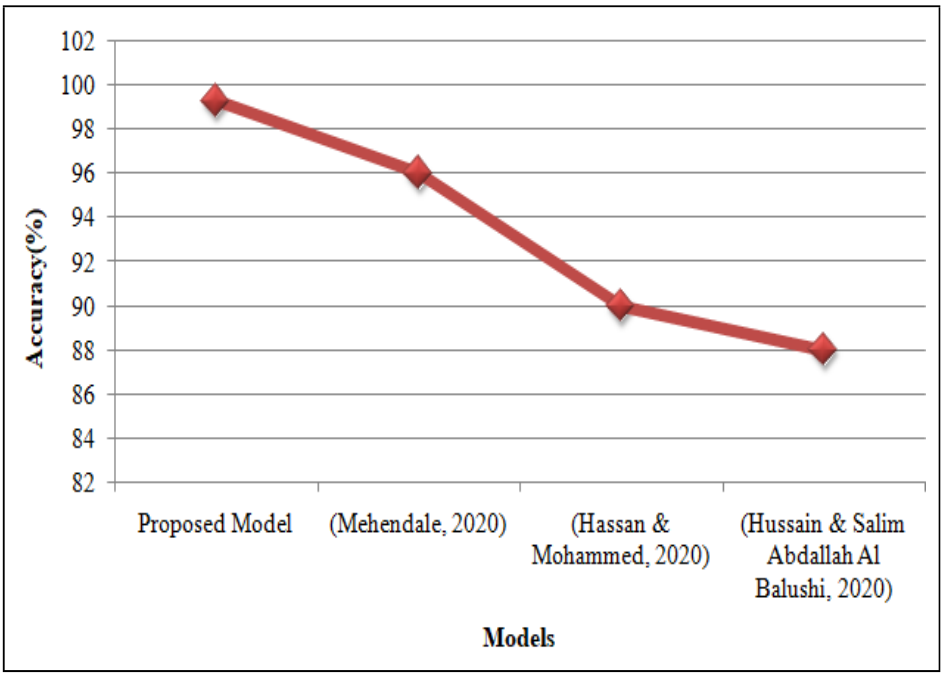

Figure 7: Comparison of Accuracy between Proposed and Existing models

In this section, the performance of our proposed methodology with existing hybrid approaches developed by Hussain & Salim Abdallah Al Balushi (2020), Hassan & Mohammed (2020), and Mehendale (2020) based on their classification accuracy is compared. Figure 7 illustrates the accuracy performance of our proposed framework under various conditions. Our model consistently demonstrated superior performance across all tested scenarios. The existing models utilized different techniques: Hussain & Salim Abdallah Al Balushi (2020) employed a graph mining scheme, Hassan & Mohammed (2020) used a CNN model, and Mehendale (2020) applied FERC. In contrast, our proposed PAFEROFA model achieved higher accuracy in emotion classification, which is attributed to using CSE and PIGA techniques for recognizing AUs and selecting optimal features for training, respectively. Therefore, it is evident that the overall performance of our proposed methodology surpasses that of the existing techniques.

5. Conclusion

FER is a crucial method for assessing emotional states. However, traditional recognition models often struggle with accuracy due to challenges like partial occlusion and wearing face masks. To address these issues, we have developed a novel FER method. The process begins with pre-processing the input image and detecting the face. Differential parts of the face are then extracted using the LSW-KMC method, which identifies a similarity score of 0.0326 within a time frame of 38010ms. Following this, feature extraction and selection are performed using the PIGA technique, which is known for its high efficiency. These selected features classify Action Units (AUs) with a trained neural network model. Later, the features and AUs are fed into the Pizam-ANFIS classifier to determine the emotions. Our proposed CSE-Pizam-ANFIS model achieved an impressive accuracy of 99.28% and a computation time of 47015ms.

The proposed FER system demonstrated high efficacy, even under the challenging conditions of partial occlusion and face masks. Therefore, our proposed model outperforms existing methods. Currently, the model is designed to recognize emotions in individual subjects. Future research will focus on advancing emotion recognition from video inputs involving multiple people.

6. Declaration

Availability of Data and Material
The data presented in this study are available upon request from the corresponding author.

Competing Interests
The authors have no relevant conflicts of interest to disclose.

Funding
The author received no funding support from any agencies or firms.

Author's Contribution
JSB gathered publicly available datasets for emotion recognition (happy, sad, angry, etc.) and developed Python methods to assess the model's performance. This involves metrics like accuracy, precision, and recall for each emotion category. JSB wrote the complete manuscript and replied to the reviewer's comments. PLP supervised the experiments, reviewed drafts of the manuscript, commented on the manuscript, and provided guidance for submission of this manuscript.

Acknowledgement
Not applicable.

References

[1] W. Mellouk and W. Handouzi, "Facial emotion recognition using deep learning: Review and insights," *Procedia Computer Science,* vol. 175, p. 689–694, 2020.

[2] M. R. Appasaheb Borgalli and D. S. Surve, "Deep learning for facial emotion recognition using custom CNN architecture," *Journal of Physics: Conference Series,* vol. 2236, pp. 1-12, 2022.

[3] N. Pratap and Shwetank, "Development of spectral signatures and classification using hyperspectral face recognition," *Journal of Interdisciplinary Mathematics,* vol. 23, pp. 453-462, 2020.

[4] E. C. Valverde, M. Udurume and W. Lim, "Performance analysis of a deep learning-based facial emotion recognition system on edge device," *Neural Computing and Applications,* p. 1–2, 2021.

[5] S. Li and W. Deng, "Deep facial expression recognition: A survey.," *IEEE Transactions on Affective Computing,* vol. 13, p. 1195–1215, 2022.

[6] A. R. Khan, "Facial emotion recognition using conventional machine learning and deep learning methods: Current achievements, analysis and remaining challenges," *Information,* vol. 13, pp. 1-17, 2022.

[7] M. Dirik, "Optimized anfis model with hybrid metaheuristic algorithms for facial emotion recognition," *International Journal of Fuzzy Systems*, 2022.

[8] S. Minaee, M. Minaei and A. Abdolrashidi, "Deep-emotion: Facial expression recognition using attentional convolutional network," *Sensors,* vol. 21, pp. 1-16, 2021.

[9] E. S. Salama, R. A. El-Khoribi, M. E. Shoman and M. A. Wahby Shalaby, "A 3D-convolutional neural network framework with ensemble learning techniques for multi-modal emotion recognition," *Egyptian Informatics Journal,* vol. 22, p. 167–176, 2021.

[10] F. M. Alamgir and M. S. Alam, "An artificial intelligence driven facial emotion recognition system using hybrid deep belief rain optimization," *Multimedia Tools and Applications,* vol. 82, p. 2437–2464, 2023.

[11] R. Kareem and K. M. M. Al-Abrahemee, "Modification artificial neural networks for solving singular perturbation problems," *Journal of Interdisciplinary Mathematics,* vol. 25, pp. 1535-1549, 2022.

[12] M. Kumar, S. K. Khatri and M. Mohammadian, "Breast cancer identification and prognosis with machine learning techniques - An elucidative review," *Journal of Interdisciplinary Mathematics,* vol. 23, pp. 503-521, 2020.

[13] N. Mehendale, "Facial emotion recognition using convolutional neural networks (FERC)," *SN Applied Sciences,* vol. 2, 2020.

[14] X. Liu, X. Cheng and K. Lee, "GA-SVM-based facial emotion recognition using facial geometric features.," *IEEE Sensors Journal,* vol. 21, p. 11532–11542, 2021.

[15] A. Alreshidi and M. Ullah, "Facial emotion recognition using hybrid features," *Informatics,* vol. 7, p. 1–13, 2020.

[16] A. K. Hassan and S. N. Mohammed, "A novel facial emotion recognition scheme based on graph mining," *Defence Technology,* vol. 16, p. 1062–1072, 2020.

[17] S. A. Hussain and A. Salim Abdallah Al Balushi, "A real time face emotion classification and recognition using deep learning model," *Journal of Physics: Conference Series,* vol. 1432, 2020.

[18] B. Houshmand and N. M. Khan, "Facial expression recognition under partial occlusion from virtual reality headsets based on transfer learning," *Proceedings-IEEE 6th International Conference on Multimedia Big Data,* p. 70–75, 2020.

[19] G. Monisha, G. Yogashree, R. Baghyalaksmi and P. Haritha, "Enhanced automatic recognition of human emotions using machine learning techniques.," *Procedia Computer Science,* vol. 218, p. 375–382, 2023.

[20] M. A. H. Akhand, S. Roy, N. Siddique, M. A. S. Kamal and T. Shimamura, "Facial emotion recognition using transfer learning in the deep CNN," *Electronics,* vol. 10, 2021.

[21] S. Saha, M. Ghosh, S. Ghosh, S. Sen, P. K. Singh, Z. W. Geem and R. Sarkar, "Feature selection for facial emotion recognition using cosine similarity-based harmony search algorithm," *Applied Sciences,* vol. 10, pp. 1-22, 2020.

[22] C. Gautam and K. Seeja, "Facial emotion recognition using Handcrafted features and CNN," *Procedia Computer Science,* vol. 218, p. 1295–1303, 2023.

[23] G. Castellano, B. De Carolis and N. Macchiarulo, "Automatic facial emotion recognition at the COVID-19 pandemic time," *Multimedia Tools and Applications,* vol. 82, p. 12751–12769, 2023.

[24] S. Wally, A. Elsayed, I. Alkabbany, A. Ali and A. Farag, "Occlusion aware student emotion recognition based on facial action unit detection," *Arxiv,* pp. 1-14, 2023.

[25] Y. ELsayed, A. ELSayed and M. A. Abdou, "An automatic improved facial expression

recognition for masked faces," *Neural Computing and Applications*, vol. 35, p. 14963–14972, 2023.

[26] M. Mukhiddinov, O. Djuraev, F. Akhmedov and A. Mukhamadiyev, "Masked face emotion recognition Based on facial landmarks," *Sensors*, vol. 23, p. 1–23, 2023.

[27] X. Zhu, J. Sun, G. Liu, C. Shen, Z. Dai and L. Zhao, "Hybrid domain consistency constraints-based deep neural network for facial expression recognition," *Sensors*, vol. 23, p. 1–16, 2023.