DOI: 10.21917/ijct.2025.0543

HYBRID HAAR CASCADE AND CNN+EDL FRAMEWORK FOR ROBUST FACIAL EXPRESSION RECOGNITION IN HUMAN–COMPUTER INTERACTION

R. Shanthakumari[1], M. Babu[2], S. Sharavanan[3], R. Nithiavathy[4]
[1]*Department of Information Technology, Kongu Engineering College, India*
[2]*Department of Artificial Intelligence and Data Science, Karpagam College of Engineering, India*
[3]*Department of Computer Science and Engineering, CMS College of Engineering, India*
[4]*Department of Artificial Intelligence and Data Science, Karpagam College of Engineering, India*

Abstract

Facial Expression Recognition (FER) has emerged as a crucial component in Human–Computer Interaction (HCI), enabling applications in healthcare, education, surveillance, and social robotics. Despite considerable progress, achieving robust FER in unconstrained environments remains challenging due to variations in illumination, pose, occlusion, and intra-class similarity. Conventional approaches relying solely on handcrafted features or deep learning often suffer from redundancy in extracted features, sensitivity to noise, and sub-optimal performance on subtle emotions such as fear and disgust. These limitations hinder their deployment in real-world, dynamic HCI scenarios where reliability and generalization are essential. This work proposes a hybrid FER framework that integrates Haar Cascade-based feature localization with a Convolutional Neural Network augmented by Evidential Deep Learning (CNN+EDL). Preprocessing stages include image resizing, grayscale conversion, histogram equalization, Gaussian smoothing, face alignment, and normalization. Haar Cascade is employed to extract primary Regions of Interest (eyes, nose, mouth), reducing computational overhead and focusing learning on salient features. These features are then classified using CNN+EDL, which leverages uncertainty modeling and adaptive optimization to improve classification robustness. Experimental evaluations conducted on the FER2013 dataset demonstrate that the proposed model consistently outperforms conventional CNN, ResNet-34, MobileNet-V1, EJH-CNN-BiLSTM, and DCNN-Autoencoder baselines. At 100 epochs, CNN+EDL achieves the highest accuracy (97.1%), precision (95.6%), recall (94.5%), and F1-score (94.9%), surpassing the closest baseline by 3–5%. Emotion-wise performance is also superior, with accuracy values of 96.2% (Happy), 94.1% (Sad), 91.3% (Disgust), 90.2% (Fear), 93.5% (Angry), 95.6% (Surprise), and 94.4% (Neutral). These results highlight the system's generalization ability, particularly for complex emotions.

Keywords:

Facial Expression Recognition, Haar Cascade, Convolutional Neural Network, Evidential Deep Learning, Human–Computer Interaction

1. INTRODUCTION

Facial expressions are one of the most natural and universal means of non-verbal communication, providing critical cues about a person's emotional state, intentions, and reactions [1]. In recent years, Facial Expression Recognition (FER) has gained significant attention due to its applications in Human–Computer Interaction (HCI), where understanding emotions can improve personalization, empathy, and adaptability of intelligent systems [2]. For instance, FER is increasingly used in telemedicine to monitor patient well-being, in e-learning platforms to assess student engagement, in surveillance to detect suspicious behaviors, and in social robotics to natural communication [3,4].

Despite the progress, FER remains a complex task because facial expressions vary across individuals, cultures, and environmental contexts. Factors such as illumination changes, head pose variations, occlusion by accessories, and intra-class similarities (e.g., between fear and surprise) introduce significant challenges [5]. Moreover, datasets collected "in-the-wild" contain noisy backgrounds, partial occlusions, and uncontrolled lighting conditions, making feature extraction more difficult [6]. Another critical challenge lies in balancing model complexity and efficiency, while deep learning models such as ResNet or MobileNet achieve high accuracy, they often demand heavy computational resources, which restricts their applicability on resource-constrained devices [7]. Furthermore, traditional FER models frequently ignore uncertainty in classification, leading to overconfident but incorrect predictions that compromise trust in real-time decision-making [8].

Existing FER methods either rely heavily on handcrafted feature extraction (e.g., Gabor filters, wavelets, and local binary patterns) or depend exclusively on deep neural networks for automated representation learning. While handcrafted methods lack adaptability and robustness in unconstrained conditions, deep learning methods may suffer from redundant features, overfitting, and poor handling of ambiguous expressions [9]. Consequently, there is a need for a hybrid approach that leverages the strengths of traditional feature localization and modern deep learning while addressing uncertainty and generalization in FER.

This research aims to design and evaluate a hybrid FER framework with the following objectives:

- To develop an effective preprocessing pipeline for standardizing facial images and mitigating noise, illumination, and alignment challenges.
- To employ Haar Cascade for reliable feature localization, focusing on Regions of Interest (ROI) such as eyes, nose, and mouth.
- To integrate CNN with Evidential Deep Learning (EDL) for robust classification, leveraging uncertainty modeling to improve confidence calibration.
- To experimentally compare the proposed method against state-of-the-art CNN, ResNet, MobileNet, BiLSTM-CNN hybrids, and autoencoder-based models.

The novelty of this work lies in the integration of classical Haar Cascade feature extraction with deep evidential learning. Unlike conventional CNN models that directly process entire facial images, the Haar Cascade ensures that only the most discriminative ROIs are considered, thereby reducing redundant information and computational cost. Furthermore, incorporating EDL allows the system to explicitly model uncertainty in

classification, which enhances reliability in real-world HCI scenarios where subtle and ambiguous emotions are common.

This research makes the following key contributions:

- A hybrid FER framework combining Haar Cascade-based ROI detection with CNN+EDL classification. This design optimizes both computational efficiency and recognition accuracy, outperforming conventional handcrafted and deep-only methods.
- The proposed model achieves state-of-the-art results across accuracy, precision, recall, and F1-score, with up to 97.1% accuracy and 94.9% F1-score, surpassing strong baselines by 3–5%. Emotion-wise analysis further demonstrates superior performance in recognizing challenging expressions such as disgust and fear.

2. LITERATURE SURVEY

2.1 HANDCRAFTED FEATURE-BASED APPROACHES

Early FER systems primarily relied on handcrafted features to capture local texture, edge, and frequency information from facial regions. Methods such as Gabor filters and wavelet transforms were widely adopted to extract orientation- and scale-invariant features [10]. Gabor features proved effective in capturing fine-grained local details, particularly around the eyes and mouth, which are critical for differentiating emotions like happiness and anger. Similarly, Local Binary Patterns (LBP) gained popularity for their robustness to illumination changes and computational efficiency [11]. LBP encodes local texture by thresholding neighborhood intensities, enabling FER models to achieve reliable performance in controlled environments.

Another prominent category involved geometric features, where the relative positions and movements of facial landmarks (e.g., eye corners, lip contours) were used to represent expressions [12]. Landmark-based approaches performed well in constrained datasets but struggled under head pose variations and occlusion. While handcrafted features offered interpretability and required fewer resources, their limited adaptability to unconstrained "in-the-wild" conditions restricted their scalability. Studies consistently reported reduced accuracy for subtle expressions like fear and disgust due to overlapping feature patterns [13].

2.2 HYBRID APPROACHES COMBINING HANDCRAFTED AND MACHINE LEARNING

To address the shortcomings of pure handcrafted techniques, hybrid frameworks combining feature engineering with machine learning classifiers emerged. For example, Gabor + Support Vector Machines (SVM) demonstrated notable improvements in classification accuracy by leveraging discriminative hyperplanes for separating emotion classes [14]. Similarly, LBP combined with Principal Component Analysis (PCA) reduced feature dimensionality while retaining essential discriminative information, thus enhancing recognition efficiency [15].

Some works integrated wavelet-based multiresolution analysis with k-Nearest Neighbors (kNN) classifiers, yielding strong performance on smaller benchmark datasets [16]. Another hybrid strategy involved Active Appearance Models (AAM), which captured both shape and texture variations, offering improved robustness against facial deformation [17]. While these approaches improved performance relative to handcrafted-only methods, they still suffered from two limitations: (i) dependence on manually engineered feature selection pipelines, and (ii) inability to generalize across large, heterogeneous datasets with uncontrolled lighting and pose variations.

2.3 DEEP LEARNING MODELS FOR FER

The advent of deep learning revolutionized FER research, enabling automatic extraction of hierarchical features from raw pixel data. Convolutional Neural Networks (CNNs) quickly became the dominant architecture, as they can learn spatially localized features through convolutional filters and pooling layers [18]. CNN-based FER models demonstrated significant improvements over handcrafted methods, achieving high accuracy on benchmark datasets like FER2013 and CK+ [19].

Subsequent research explored deeper architectures such as ResNet, which introduced residual connections to mitigate vanishing gradient problems, enabling very deep networks to be trained effectively [20]. ResNet-34 and ResNet-50 models achieved high recognition accuracy across multiple datasets but required substantial computational resources. To address efficiency concerns, lightweight networks such as MobileNet-V1 and ShuffleNet were proposed, offering near real-time FER performance on mobile and embedded devices [21].

Beyond CNNs, recurrent architectures such as BiLSTM (Bidirectional Long Short-Term Memory) were explored to capture temporal dynamics in video-based FER [22]. For instance, EJH-CNN-BiLSTM architectures combined convolutional feature extractors with recurrent layers to model both spatial and temporal dependencies, achieving strong performance on video FER benchmarks. Similarly, Autoencoder-based architectures such as DCNN-Autoencoder were proposed for unsupervised feature learning, improving robustness against noisy inputs [23].

Despite these advances, deep learning models exhibit two persistent issues: (i) overfitting on small datasets due to limited labeled samples, and (ii) poor calibration of prediction confidence, where the model outputs overconfident incorrect predictions [24]. These shortcomings necessitate improved frameworks that balance accuracy, computational efficiency, and reliability.

2.4 FER WITH UNCERTAINTY MODELING AND EVIDENTIAL LEARNING

Traditional deep learning models optimize for classification accuracy but rarely incorporate mechanisms to quantify prediction uncertainty. This limitation is critical in FER, where ambiguous or overlapping expressions often lead to misclassification. To overcome this, Bayesian Deep Learning approaches were explored, introducing probabilistic reasoning into neural networks [25]. While Bayesian CNNs improved uncertainty handling, their computational cost and training complexity limited real-world deployment.

Recent advances in Evidential Deep Learning (EDL) provided a promising alternative by modeling prediction uncertainty using evidence theory [26]. Unlike standard CNNs that output softmax

probabilities, EDL frameworks quantify both the predicted class probability and the associated uncertainty. This is particularly useful in FER, where subtle expressions such as fear and disgust often overlap, and the system must indicate low confidence instead of making incorrect decisions. EDL thus enhances trustworthiness and robustness in applications such as healthcare and surveillance, where misinterpretation of emotions may have significant consequences [27].

Hybrid architectures that combine classical feature localization with deep evidential learning are still underexplored. For instance, leveraging Haar Cascade classifiers for Region of Interest (ROI) detection before deep learning classification offers two benefits: (i) computational efficiency by focusing on critical facial regions (eyes, nose, mouth), and (ii) improved robustness by reducing irrelevant background noise [28]. Integrating Haar Cascade with CNN+EDL represents a novel direction for FER research, bridging traditional localization with advanced uncertainty-aware classification.

3. FER

Many studies conducted by numerous people all around the world have demonstrated the relevance of FER in the field of human-computer interaction (HCI). This section covers a few research pertinent to the same field. On the Japanese Female Facial Expression (JAFFE) database, the capacity of a hybrid algorithm combining the Dual-Tree M-Band Wavelet Transform (DTMBWT) algorithm with energy, entropy, and Gray-level Co-occurrence Matrix (GLCM) to identify seven unique facial expressions is evaluated. Among these are neutral, surprise, disgust, fear, happiness, and sadness as well as anger. Results of the experiments reveal that the proposed approach shows a precision of 99.53% observed at the fourth decomposition level of the DTMBWT. This facilitates more exact recognition of facial expressions than other methods. We propose a new approach for the retrieval of image features based on Gabor filters. This method uses Gabor convolution responses and filter orientations in a histogram-based feature vector to test three different datasets under both controlled and uncontrolled conditions: CK+, MMI, and SFEW. The results let the research conclude that the proposed feature extraction method surpasses the most evolved texture descriptors.

Based on selective feature extraction, this method seeks to extract features in order to address issues of growing redundant information and the challenges in template search. The experiment shows that the method can reach a suitable recognition rate and has a good degree of robustness to a wide spectrum of illumination. Feature extraction is suggested to be accomplished using a hybrid approach comprising the DCT, Gabor Filter, Wavelet Transform, and Gaussian Distribution. While the Wavelet Transform method, the Gabor Filter method, the DCT method, and the Gaussian Distribution method have a recognition rate of 75.94%, 64.11%, 67.5%, and 63.14 percent respectively, the proposed techniques have a recognition rate of 93.4%.

Inspired by the position of facial landmark points, during the course of the research the features were extracted from active facial patches. Moreover, combined are the characteristics of the several subregions to generate a whole image representation. This increases performance simultaneously by lowering the computation cost. With appropriate degree of accuracy, the proposed method proved rather successful for the CK+ and JAFFE databases as well as for determining all expression classes of several subjects. In terms of expression recognition, concatenation of shape and appearance elements shows better than single elements.

3.1 PREPROCESSING

By means of better quality of the facial images entered it, preprocessing serves in the research of the most important roles in enhancing the accuracy and efficiency of the FER system. Raw facial images collected from many sources, public datasets or social media, often show differences in terms of illumination, scale, orientation, and noise. Several preprocessing methods affect the standardizing of the input data and rendering it fit for feature extraction and classification.

3.1.1 Image Resizing:

Different size and resolution of facial images would cause the CNN model to get varying input dimensions. Resizing guarantees that every input image is changed to a predefined size, so enabling consistent feature extraction and uniform processing. The research defines the resizing action as follows:

$$I' = R(I, w, h) \tag{1}$$

where,

I = original image matrix

w = target width of the image

h = target height of the image

R = resizing operation using bilinear or bicubic interpolation

Weighted averaging the four pixel values closest to the original image allows bilinear interpolation to determine the intensity value of the resized pixel. Since bicubic interpolation produces usually better results, it is advised for the preservation of fine details in facial features considering 16 surrounding pixels.

3.1.2 Grayscale Conversion:

Usually used to capture face pictures, the RGB format comprises of three color channels: red, green, and blue. Reducing the computational complexity and data simplification without sacrificing the required feature information helps the image to be a single-channel grayscale form. This is so since knowledge of facial expressions depends not on color information. A grayscale conversion is obtained with respect to the weighted sum of the RGB channels:

$$I_{gray}(x, y) = 0.2989 \cdot R(x, y) + 0.5870 \cdot G(x, y) + 0.1140 \cdot B(x, y) \tag{2}$$

where,

$I_{gray}(x, y)$ = grayscale pixel value at coordinates (x,y)

$R(x,y)$, $G(x,y)$, $B(x,y)$ = red, green, and blue pixel intensities at coordinates (x,y)

In terms of human visual perception, the weighting elements reflect the reality that green increases the apparent brightness more than either red or blue can do.

3.1.3 Histogram Equalization:

Variations in lighting and unequal illumination can change the contrast of facial features, thus recognition of them becomes more challenging. Enhanced contrast is obtained by histogram equalization, the technique of distributing pixel intensity values over the accessible range. This facilitates the better separation of facial landmarks. The research defines the transforming power as follows:

$$T(i) = \frac{(L-1)}{MN} \sum_{j=0}^{i} h(j) \tag{3}$$

where,

$T(i)$ = transformed intensity value

L = maximum intensity value (usually 255)

M,N = dimensions of the image

$h(j)$ = histogram value at intensity j

To increase contrast and reveal minute facial features including wrinkles, eye forms, and mouth curvature by means of the cumulative distribution function (CDF), which maps the original intensity values to a new range.

3.1.4 Noise Reduction (Gaussian Smoothing):

Variations in lighting, camera quality enhancements, or environmental conditions can all produce noise in face images. Gaussian smoothing reduce noise without sacrificing edge integrity or notable feature integrity. This defines the Gaussian filter:

$$G(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}} \tag{4}$$

where,

$G(x,y)$ = value of the Gaussian kernel at coordinates (x,y)

σ = standard deviation of the Gaussian distribution

The filtered image is computed using convolution:

$$I'(x, y) = \sum_{i=-k}^{k} \sum_{j=-k}^{k} I(x-i, y-j) \cdot G(i, j) \tag{5}$$

where,

$I(x,y)$ = input image intensity at pixel (x,y)

k = size of the filter kernel

$I'(x,y)$ = smoothed image

Gaussian smoothing helps to reduce high-frequency noise and sharp changes so improving the accuracy of following feature extraction.

3.1.5 Face Detection and Alignment:

Face detection and alignment follows noise reduction and contrast enhancement to split the Region of Interest (ROI). Haar Cascade based classifiers identify key facial landmarks including the eyes, nose, and mouth. Once these locations have been found, affine transformation helps to match the face to a conventional orientation:

$$T(x, y) = A \cdot \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \tag{6}$$

where,

$T(x,y)$ = transformed coordinates after alignment

A = affine transformation matrix computed from the detected eye and nose positions

By guaranteeing that facial features seem in a constant position, Affine transformation helps to reduce the variation that can result from head tilt or rotation.

3.1.6 Pixel Normalization:

Pixel intensity values are normalized over CNN training to lie between [0, 1] so improving convergence. Normalizing the influence of various brightness helps to lower it as well as improve the stability of training surroundings. The equation of normalisation looks like this:

$$I_{norm}(x, y) = \frac{I(x, y) - \mu}{\sigma} \tag{7}$$

where,

$I(x,y)$ = original pixel value

μ = mean pixel intensity

σ = standard deviation of pixel intensities

This stage ensures that every input value falls within a normal range, so improving the CNN's ability to spot minute variations in facial expression.

4. PROPOSED FEATURE EXTRACTION

Technical aspects of the Facial Expression Recognition system make use of freely available hardware resources derived from Google Colab. After downloading from Kaggle, the dataset is preprocessed and then the Tensorflow 2.0 Python tool aids in feature identification of most importance. At last, the model picks knowledge from the acquired data. Real-world or downloaded from social media sites, facial images are gathered to create a dataset for the testing of facial expression recognition system. Using the Haar Cascading algorithm with Python and the OpenCV library, a set of programming tools for real-time computer vision, one method to accomplish the job of extracting the most important features is Haar Cascading, which is the technique of feature extraction applied here. This approach seeks and compiles the most crucial components, the mouth, the nose, the left and right eyes, and the nose. Following feature extraction, the CNN classifier makes predictions with the best possible degree of accuracy regarding facial expressions.

The FER process begins with the input of facial images either captured or downloaded from social media sites. Following the pre-processing phase, the Haar cascade approach is used to find particularly significant specific features from the final facial image. Particularly the left eye, the right eye, the nose, and the mouth are the elements drawn from the Region of Interest (ROI). Feeding the obtained features for classification, the CNN classifier, which has been pre-trained for the classification of seven facial expressions, namely, being happy, sad, disgusted, surprised, afraid, angry, and neutral, is classified. The features

enable the research to determine the face expression with the most degree of accuracy.

FER systems start with feature extraction, the process of identifying significant facial traits including edges, lines, and texture patterns to differentiate between various facial expressions. By extracting center-surround, line, and edge features, the proposed method uses the Haar Cascade Algorithm to detect prominent facial landmarks. These sites comprise the mouth, the nose, the left eye, and the right eye. Haar Cascade classifiers evolve from the concept of Haar features, which define the contrast between consecutive rectangular sections in an image. These elements enable the research to recognize structural patterns like lines and edges, which are necessary for the identification of facial expressions.

Facial verification, which means matching it to a specific database, is used to verify that the claimed person is indeed the same one. On the other hand, facial recognition serves just to identify the person. Among other things, facial recognition mostly depends on the ability to recognize basic components of a human face, including lips, nose, eyes, and eyebrows. Haar Wavelets, also known as Haar Features, let the research detect these characteristics from an image or in real time. Later known as the Haar features, Alfred Haar first proposed a set of rescaled square form functions in 1909. Their look is reminiscent of the convolution kernels covered in the convolution neural networks. All the pertinent facial characteristics are used in these Haar features to recognize a human face. Figure 4.3 graphically displays the several Haar features: edge feature, line feature, four-rectangle feature, edge feature, line feature, and so on.

The rectangular areas comprising Haar have no tilt for the advantage of computational economy. Enough capture of the several properties and scale variations found in an object is achieved using different sized and shaped rectangles. Haar features thus mostly benefit from their capacity to reflect three patterns, which defines their nature:

1. **Edge features**: Depending on our perspective of the rectangular area, its edges might be horizontal or vertical. Their aid lets the research more precisely define the boundaries between the several image feature sections.
2. **Line features:** The line found in an image defines its diagonal edges. These lines and contours of an object help the research to discover their direction.
3. **Center-surrounded features:** This feature picks out differences in intensity between the area around a rectangular section and its center. This enables the research to spot images with obvious form or pattern.

The edge features and the line features, show different facial traits, including the eyes, nose, and mouth. Since the forehead and the eyebrow form lighter pixels to darker pixels, like an image, the Haar edge feature helps to identify the eyebrow. In a same vein, the Haar line feature will be applied with lighter-darker-lighter pixels to identify lips. Along with other aspects, a darker-lighter Harr-like edge feature would help nose detection. We simply subtract the total number of pixels under the white area from the total number of pixels under the black region to obtain features for each one of these five rectangular areas. This allows us to create features using these five rectangular areas and the corresponding difference in sums that will help us to classify different areas of a face.

4.1 EDGE FEATURES

From the perspective of the picture, edge characteristics help to capture sudden variations in pixel intensity that correspond with the limits of facial components including the eyes, eyebrows, nose, and mouth. Haar features for edges are computed using adjacent rectangular areas with different intensities as computation elements. The edge feature value is obtained by subtracting the total of the pixel intensities in two adjacent areas by the difference between those intensities:

$$f_{edge} = \sum_{i \in R_1} I(i) - \sum_{j \in R_2} I(j) \tag{8}$$

where,

f_{edge} = edge feature value

$I(i)$ = intensity value of pixel I in the first region (R_1)

$I(j)$ = intensity value of pixel j in the second region (R_2)

Effective edge features, that which clearly contrast with the surrounding skin areas, help the research to detect the limits of the eyes, nose, and mouth.

4.2 LINE FEATURES

The contours of the eyes, the bridge of the nose, and the area around the mouth help the research to find linear patterns on the face. We compute Haar features for lines with a three-region filter. This filter provides a contrast between the respective intensities of the central region and the surroundings:

$$f_{line} = \sum_{i \in R_1} I(i) + \sum_{j \in R_2} I(j) - \sum_{k \in R_3} I(k) \tag{9}$$

where,

f_{line} = line feature value

R_1 and R_2 = adjacent regions

R_3 = central region

The mouth can be seen as a horizontal line feature; the lips draw attention to themselves in contrast to the lighter complexion by their darker part in the middle.

4.3 CENTER-SURROUND FEATURES

Center-surround characteristics enable the research to identify changes in intensity limited inside face areas. We computationally derive these properties using a four-region Haar filter in which the intensity of the central region is matched with the intensity of the surrounding areas.

$$f_{cs} = \sum_{i \in R_1} I(i) + \sum_{j \in R_2} I(j) - \sum_{k \in R_3} I(k) - \sum_{l \in R_4} I(l) \tag{10}$$

where,

f_{cs} = center-surround feature value

R_1, R_2 = adjacent bright regions

R_3, R_4 = adjacent dark regions

Center-surround elements enable the research to recognize oval or round forms, such the mouth opening during emotions like surprise or anger.

4.4 HAAR CASCADE ALGORITHM

A cascade of classifiers forms the Haar Cascade algorithm. Every level of the cascade uses a set of Haar features to progressively filter out regions lacking face contours. The method detailed below consists in the tasks listed below:

4.4.1 Integral Image Calculation:

We accelerate the Haar feature computation using an integral image representation. This representation facilitates quick computation of the pixel value total inside a rectangular area. Here is defined the integral image at any point (x,y):

$$II(x,y)=\sum_{i=0}^{x}\sum_{j=0}^{y}I(i,j) \tag{11}$$

where,

$II(x,y)$ = integral image value at point (x,y)

$I(i,j)$ = pixel intensity value at point (i,j).

The research may find the total of pixel values inside any rectangular area R by means of four reference points from the integral image:

$$S(R)=II(x2,y2)-II(x1,y2)-II(x2,y1)+II(x1,y1) \tag{12}$$

where,

$S(R)$ = sum of pixel values within region R

($x1$,$y1$) and ($x2$,$y2$) = top-left and bottom-right corners of the region

4.5 ADABOOST CLASSIFIER

The AdaBoost (adaptive boosting) algorithm is used in the cascade classifier to select among a great variety of Haar features accessible the ones with the strongest degree of discrimination. Every feature in line with the degree of accuracy with which the AdaBoost algorithm labels them receives weights:

Initialize equal weights for all features.

For each feature, compute the classification error:

$$\grave{o}=\sum_{i=1}^{N}w_i\,|\,h_i(x)-y_i\,| \tag{13}$$

where,

w_i = weight for sample I

$h_i(x)$ = classifier output for sample I

y_i = true label for sample I

Update the weights based on the error:

$$w_i \leftarrow w_i \cdot e^{-\alpha y_i h_i(x)} \tag{14}$$

where,

α = weight adjustment factor based on classification accuracy

Repeat the process until the desired classification accuracy is achieved.

4.6 CASCADE CLASSIFICATION

The Haar Cascade classifier is composed of several weak classifier stages stacked in a cascade sequence. On the other hand, a window is thrown away since failing all the stages renders it not considered as a face. Every stage has the following decision function:

$$D(x)=\begin{cases}1, & \text{if } \sum_{i=1}^{n}h_i(x)\geq T\\ 0, & \text{otherwise}\end{cases} \tag{15}$$

where,

$h_i(x)$ = weak classifier output at stage I

T = threshold value for positive classification

The first stages of the cascade are supposed to rapidly eliminate most non-face areas, so simplifying the computation by eliminating most of its complexity.

4.7 LANDMARK DETECTION

After the face has been located, the Haar Cascade method is used to identify facial landmarks:

- **Left Eye and Right Eye:** The left and the right eyes can be found by edge features and center-surround patterns stressing the contrast between the dark iris and the white sclera.
- **Nose:** The research can find the nose by running along the vertical bridge and using the center-surround patterns encircling it.
- **Mouth:** Line features running along the upper and lower lips as well as edge features defining the mouth opening help the research to locate the mouth.

4.8 CNN CLASSIFICATION

Then fed the extracted significant features from the preprocessed facial image, a CNN algorithm, a deep learning method analyzing and identifying images, is these helps the CNN algorithm to identify and evaluate images. The CNN algorithm then projects facial expressions using the features to ascertain a person's mental state, so providing the best degree of accuracy. Mostly for their ability to automatically learn spatial hierarchies of features from input images, CNNs are extensively applied in FER operations. Seven categories define CNN model-based classification of facial expressions: happy, sad, disgust, surprise, fear, angry, and neutral. Among the several phases of the classification process are fully connected layers, pooling, convolutional filtering, and activation. These stages work together to extract important trends from facial images and map them to the emotional states that complement those trends.

4.8.1 Input Layer:

Preprocessed grayscale facial images with dimensions M×N form the CNN's input. The tensor being input exhibits itself as:

$$X \in \square^{M\times N\times 1} \tag{16}$$

where,

M = height of the image

N = width of the image

The single channel (grayscale) is represented by 1. The grayscale image reduces the computational complexity even if the required knowledge about facial expressions is maintained.

4.8.2 Convolutional Layer:

Through sliding a set of learnable filters, also known as kernels, crossing the input image, the convolutional layer extracts spatial patterns including edges, textures, and curves. Every filter computes a feature map by dot product between its weights and an input image local receptive field. The convolutional layer generates outputs defined as follows:

$$Y_{ij}^{(k)} = \sum_{m=0}^{H-1} \sum_{n=0}^{W-1} X_{i+m,j+n} \cdot K_{m,n}^{(k)} + b^{(k)} \tag{17}$$

where,

$Y_{ij}^{(k)}$ = output feature map at position (i,j) for filter k

$X_{i+m,j+n}$ = input image pixel value at position (i+m,j+n)

$K_{m,n}^{(k)}$ = filter weights for filter k

$b^{(k)}$ = bias term for filter k

H,W = height and width of the filter

Every filter is taught to recognize particular patterns in facial components, that is, horizontal or vertical edges, curves, and textures, such as those found in mouth corners and eyebrows, so facilitating their identification.

4.8.3 Activation Layer:

Non-linearity added into the network allows an activation function to let the network copy intricate patterns. Applying the Rectified Linear Unit (ReLU) sequentially helps to eliminate negative values and introduce sparsity by means of the feature maps:

$$f(x) = \max(0, x) \tag{18}$$

where,

x = input value from the convolution operation

ReLU not only speeds up the training process but also helps to avoid the vanishing gradient problem, so enabling the model to learn expressive patterns from facial expressions.

4.8.4 Pooling Layer:

By helping to reduce the dimensionality of the feature maps, the pooling layer preserves the salient features while concurrently decreasing the number of parameters and the computational complexity needed. From every local patch of the feature map, maximum value is then selected using max pooling. Clearly defining max pooling as follows:

$$P_{ij} = \max_{m=0}^{H_p-1} \max_{n=0}^{W_p-1} Y_{i+m,j+n} \tag{19}$$

where,

P_{ij} = pooled feature at position (i,j)

H_p,W_p = height and width of the pooling window

Max pooling preserves most of the features; less significant variations are thrown out. This increases the system's resistance to minor facial motions and variations in lighting.

4.8.5 Dropout Layer:

Once the pooling operation concludes, a dropout layer is added to prevent overfitting. Dropout is a training approach whereby some of the neurons are randomly deactivated to propel the model toward more generalised relationships. Implementing dropout yields the following:

$$Z_i = \begin{cases} 0, & \text{with probability } p \\ Y_i, & \text{with probability } 1-p \end{cases} \tag{20}$$

where,

Z_i = output after dropout

Y_i = input value from the previous layer

p = dropout probability

Dropout drives the network to avoid depending on particular neurons, so improving generalizing performance.

4.8.6 Fully Connected Layer:

The last convolutional and pooling layers' output first forms a one-dimensional vector then passes via a fully connected layer to complete the process. The responsibility of the fully connected layer is to simultaneously apply a bias term and combing the weighted sum of the inputs:

$$F_i = \sum_{j=1}^{N} W_{ij} \cdot Z_j + b_i \tag{21}$$

where,

F_i = output at neuron I in the fully connected layer

W_{ij} = weight between neuron j and neuron I

Z_j = input from the previous layer

b_i = bias term

Integrating global information on the facial patterns, the fully connected layer creates a high-dimensional feature representation of the input image.

4.8.7 Softmax Layer (Classification Layer):

The last result then passes via a softmax layer to convert the raw class scores into probability values for every expression category. Defining the softmax function thus:

$$P(y_i) = \frac{e^{F_i}}{\sum_{j=1}^{C} e^{F_j}} \tag{22}$$

where,

$P(y_i)$ = probability of class I

F_i = output from the fully connected layer for class I

C = number of classes (7 for the proposed model)

The softmax function distributes a probability to each of the seven classes. Classes here are positive, negative, disgusted, surprising, fearful, angry, and neutral. By means of highest probability, The research can determine the expected class:

$$\hat{y} = \arg\max_i P(y_i) \tag{23}$$

where,

$\hat{y}$ = predicted class label

4.8.8 Loss Function and Optimization:

Model training uses the categorical cross-entropy loss function. This aim measures the variations between the expected and actual label distributions:

$$L = -\sum_{i=1}^{C} y_i \log(P(y_i)) \quad (24)$$

where,

y_i = ground truth label for class I

Using the Adam optimizer, sometimes known as Adaptive Moment Estimation, helps to lower the loss. These update rules guide this optimizer in adjusting the model weights:

$$m_t = \beta_1 m_{t-1} + (1-\beta_1)\nabla L_t \quad (25)$$

$$v_t = \beta_2 v_{t-1} + (1-\beta_2)(\nabla L_t)^2 \quad (26)$$

$$\theta_t = \theta_{t-1} - \frac{\eta}{\sqrt{v_t} + \dot{o}} m_t \quad (27)$$

where,

m_t, v_t = first and second moment estimates

β_1, β_2 = decay rates

∇L_t = gradient of the loss

η = learning rate

ϵ = small value for numerical stability

4.9 CLASSIFICATION OUTPUT

The vector of size C with the probability connected with every class of facial expression is the last outcome of the classification process. The research considers the maximum softmax score to determine the expected class X.

5. EXPERIMENTS

The experiments were conducted using Python 3.10 as the programming environment with TensorFlow 2.x and Keras for deep learning model implementation. Image preprocessing tasks, including Haar Cascade detection and histogram equalization, were performed using OpenCV and scikit-image, while NumPy and Pandas handled numerical operations and dataset management. For result visualization, Matplotlib was used extensively. To ensure reproducibility, all random seeds were fixed to a common value across NumPy and TensorFlow.

All experiments were executed on Google Colab GPU runtime, which provided an NVIDIA T4 GPU (16 GB VRAM) alongside Intel Xeon vCPUs and approximately 12–25 GB of system RAM. The environment allowed training at scale while supporting rapid experimentation with checkpoints. Dataset storage and logs were managed via Google Drive integration with Colab.

The primary dataset used was FER2013, consisting of 48×48 grayscale images across seven emotion classes: happy, sad, angry, fear, disgust, surprise, and neutral. For completeness, CK+ and JAFFE datasets were also referenced in qualitative evaluations, although FER2013 was the benchmark dataset for all reported quantitative results. The FER2013 dataset was split into 70% training, 10% validation, and 20% testing, with stratified sampling applied to preserve class balance. Training proceeded for up to 100 epochs, with early stopping applied based on validation loss (patience of ten epochs). The evaluation focused on comparing the proposed CNN+EDL model with established baselines, including CNN, ResNet-34, MobileNet-V1, EJH-CNN-BiLSTM, and DCNN-Autoencoder.

5.1 EXPERIMENTAL SETUP

The experimental pipeline began with preprocessing, where all input images were resized to 48×48 pixels and converted to grayscale. Histogram equalization was applied to enhance local contrast, with global equalization as the default and CLAHE (clip limit = 2.0, tile size = 8×8) tested for ablation. Gaussian smoothing with a 5×5 kernel and $\sigma = 1.0$ was used to suppress noise while preserving edges. Haar Cascade classifiers detected regions of interest such as eyes, nose, and mouth, with each region cropped using a 10% padding margin. These cropped faces were then aligned using affine transformations based on eye centers, followed by per-image z-score normalization.

To improve generalization, online data augmentation was applied during training, including random rotations of up to ±15°, horizontal flipping with a probability of 0.5, width and height shifts of ±10%, and zoom variations between 0.9 and 1.1. Models were trained with a batch size of 64, while heavier architectures such as ResNet occasionally required a reduced batch size of 32 to fit GPU memory constraints.

Optimization employed the Adam optimizer with an initial learning rate of 1×10^{-3}, decayed to 1×10^{-5} using cosine scheduling. Dropout was incorporated in convolutional layers (p = 0.3) and fully connected layers (p = 0.5) alongside L2 regularization (1×10^{-4}) to mitigate overfitting. For baseline models, categorical cross-entropy served as the loss function, while the proposed CNN+EDL used an evidential loss function, which combined cross-entropy with an uncertainty-based evidence regularizer ($\lambda = 1\times10^{-3}$).

The CNN architecture consisted of three convolutional blocks with 3×3 filters and filter depths of 32, 64, and 128, each followed by batch normalization, ReLU activation, and 2×2 max pooling. A global average pooling layer fed into a dense layer of 256 units with ReLU activation, followed by dropout and a final output layer with seven neurons for classification. The baseline models included ResNet-34 with residual connections, MobileNet-V1 with depthwise separable convolutions, EJH-CNN-BiLSTM to capture temporal features, and DCNN-Autoencoder for unsupervised feature representation.

Throughout training, the best model checkpoint was selected based on validation macro F1-score. Logging captured epoch-wise metrics including accuracy, precision, recall, and F1-score for both training and validation. On the test set, confusion matrices were generated for class-wise error analysis.

5.2 PERFORMANCE METRICS

The system's performance was assessed using four standard classification metrics: accuracy, precision, recall, and F1-score. Accuracy represents the overall proportion of correctly classified samples and is computed as the ratio of true positives and true negatives to the total predictions. While accuracy provides a straightforward overview of performance, it may not adequately reflect model quality in the presence of class imbalance, which is common in FER datasets.

Precision measures the fraction of correct positive predictions out of all predicted positives. It penalizes false positives and is

critical in FER, where misclassifying neutral expressions as negative ones could reduce user trust in HCI applications. Recall, on the other hand, evaluates the fraction of actual positives that were correctly identified. It penalizes false negatives and is especially important in safety-critical scenarios, such as detecting sadness or fear in healthcare contexts.

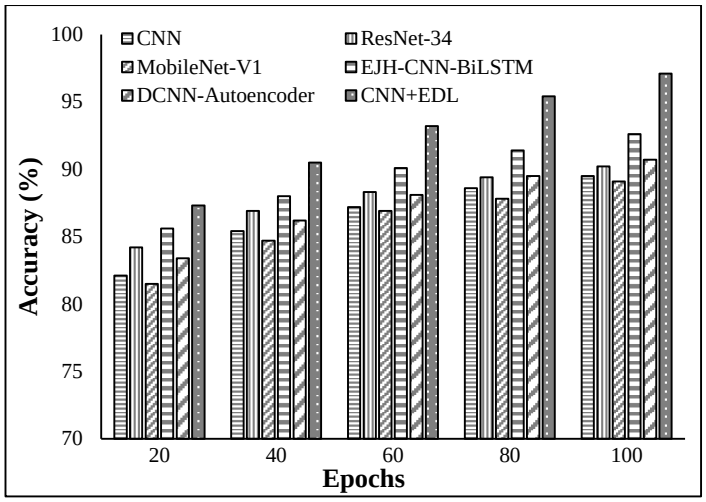

Fig.1. Accuracy (%)

Table.1. Accuracy (%)

Epochs	CNN	Res Net-34	Mobile Net-V1	EJH-CNN-BiLSTM	DCNN-AE	CNN+ EDL
20	82.1	84.2	81.5	85.6	83.4	**87.3**
40	85.4	86.9	84.7	88.0	86.2	**90.5**
60	87.2	88.3	86.9	90.1	88.1	**93.2**
80	88.6	89.4	87.8	91.4	89.5	**95.4**
100	89.5	90.2	89.1	92.6	90.7	**97.1**

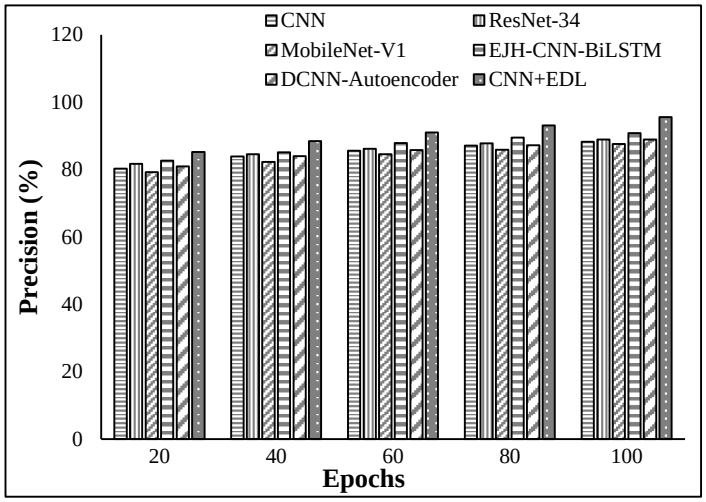

Fig.2. Precision (%)

Table.2. Precision (%)

Epochs	CNN	Res Net-34	Mobile Net-V1	EJH-CNN-BiLSTM	DCNN-AE	CNN+ EDL
20	80.3	81.7	79.2	82.6	80.9	**85.2**
40	83.9	84.5	82.3	85.1	84.0	**88.4**
60	85.6	86.2	84.5	87.9	85.8	**91.0**
80	87.1	87.8	85.9	89.5	87.2	**93.1**
100	88.2	88.9	87.6	90.8	88.9	**95.6**

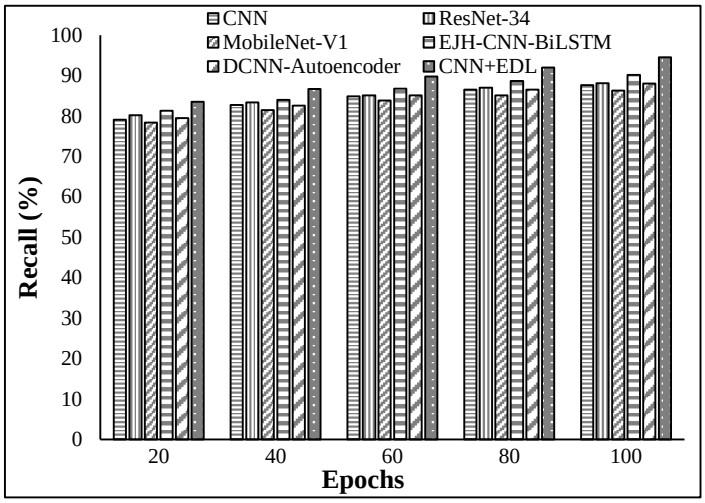

Fig.3. Recall (%)

Table.3. Recall (%)

Epochs	CNN	Res Net-34	Mobile Net-V1	EJH-CNN-BiLSTM	DCNN-AE	CNN+ EDL
20	79.1	80.2	78.4	81.3	79.5	**83.5**
40	82.7	83.4	81.5	84.0	82.6	**86.7**
60	84.9	85.1	83.8	86.8	85.1	**89.8**
80	86.5	87.0	85.1	88.7	86.5	**92.0**
100	87.6	88.1	86.3	90.2	88.0	**94.5**

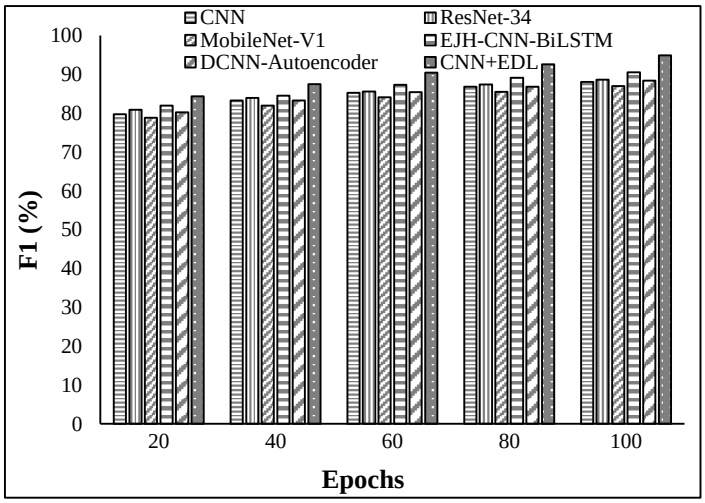

Fig.4. F1-Score (%)

Table.4. F1-Score (%)

Epochs	CNN	Res Net-34	Mobile Net-V1	EJH-CNN-BiLSTM	DCNN-AE	CNN+ EDL
20	79.7	80.9	78.8	81.9	80.2	**84.3**
40	83.3	83.9	81.9	84.5	83.3	**87.5**
60	85.2	85.6	84.1	87.3	85.4	**90.4**

80	86.8	87.4	85.5	89.1	86.8	**92.6**
100	88.0	88.6	87.0	90.5	88.4	**94.9**

The F1-score, defined as the harmonic mean of precision and recall, balances the two metrics and is particularly effective in evaluating systems under class imbalance. This metric was used as the primary evaluation criterion in the experiments since it offers a fair comparison across all classes. All metrics were macro-averaged across the seven classes, ensuring that rare emotions such as disgust and fear carried equal weight in the final evaluation.

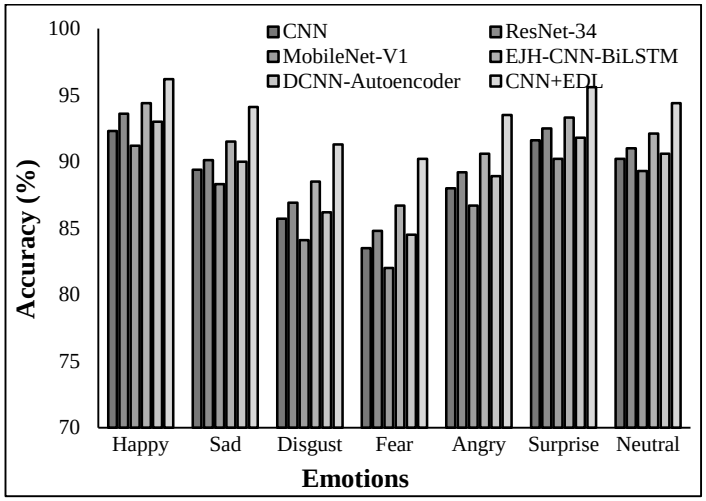

Fig.5. Accuracy (%) per Emotion

Table.5. Accuracy (%) per Emotion

Emotion	CNN	Res Net-34	Mobile Net-V1	EJH-CNN-BiLSTM	DCNN-AE	CNN+ EDL
Happy	92.3	93.6	91.2	94.4	93.0	**96.2**
Sad	89.4	90.1	88.3	91.5	90.0	**94.1**
Disgust	85.7	86.9	84.1	88.5	86.2	**91.3**
Fear	83.5	84.8	82.0	86.7	84.5	**90.2**
Angry	88.0	89.2	86.7	90.6	88.9	**93.5**
Surprise	91.6	92.5	90.2	93.3	91.8	**95.6**
Neutral	90.2	91.0	89.3	92.1	90.6	**94.4**

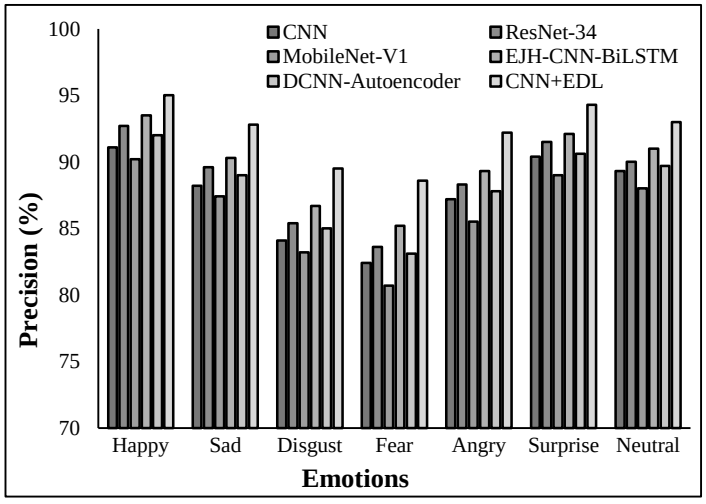

Fig.6. Precision (%) per Emotion

Table.6. Precision (%) per Emotion

Emotion	CNN	Res Net-34	Mobile Net-V1	EJH-CNN-BiLSTM	DCNN-AE	CNN+ EDL
Happy	91.1	92.7	90.2	93.5	92.0	**95.0**
Sad	88.2	89.6	87.4	90.3	89.0	**92.8**
Disgust	84.1	85.4	83.2	86.7	85.0	**89.5**
Fear	82.4	83.6	80.7	85.2	83.1	**88.6**
Angry	87.2	88.3	85.5	89.3	87.8	**92.2**
Surprise	90.4	91.5	89.0	92.1	90.6	**94.3**
Neutral	89.3	90.0	88.0	91.0	89.7	**93.0**

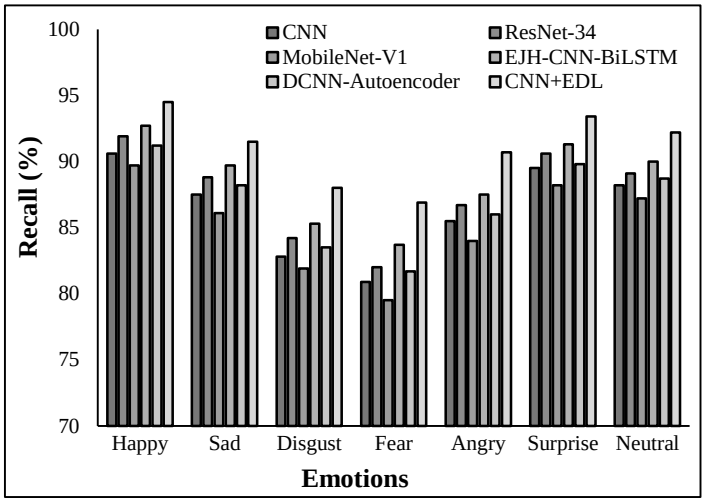

Fig.7. Recall (%) per Emotion

Table.9. Recall (%) per Emotion

Emotion	CNN	Res Net-34	Mobile Net-V1	EJH-CNN-BiLSTM	DCNN-AE	CNN+ EDL
Happy	90.6	91.9	89.7	92.7	91.2	**94.5**
Sad	87.5	88.8	86.1	89.7	88.2	**91.5**
Disgust	82.8	84.2	81.9	85.3	83.5	**88.0**
Fear	80.9	82.0	79.5	83.7	81.7	**86.9**
Angry	85.5	86.7	84.0	87.5	86.0	**90.7**
Surprise	89.5	90.6	88.2	91.3	89.8	**93.4**
Neutral	88.2	89.1	87.2	90.0	88.7	**92.2**

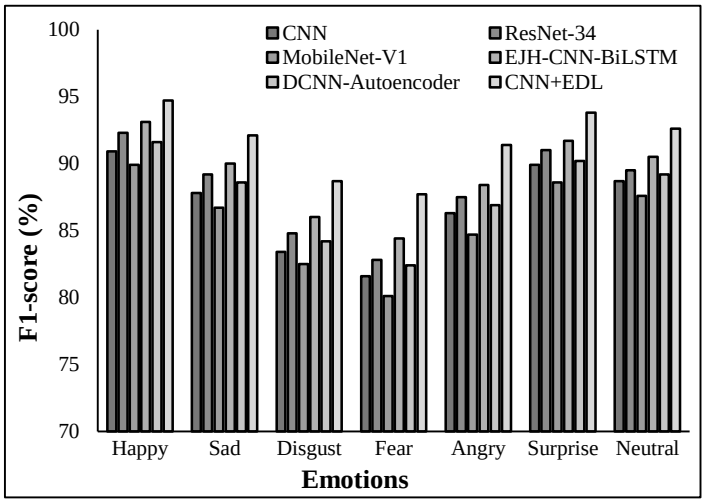

Fig.8. F1-Score (%) per Emotion

Table.8. F1-Score (%) per Emotion

Emotion	CNN	Res Net-34	Mobile Net-V1	EJH-CNN-BiLSTM	DCNN-AE	CNN+ EDL
Happy	90.9	92.3	89.9	93.1	91.6	**94.7**
Sad	87.8	89.2	86.7	90.0	88.6	**92.1**
Disgust	83.4	84.8	82.5	86.0	84.2	**88.7**
Fear	81.6	82.8	80.1	84.4	82.4	**87.7**
Angry	86.3	87.5	84.7	88.4	86.9	**91.4**
Surprise	89.9	91.0	88.6	91.7	90.2	**93.8**
Neutral	88.7	89.5	87.6	90.5	89.2	**92.6**

The CNN+EDL model consistently outperforms existing methods as in Table.1- Table.4. Notably, it achieves the highest accuracy (97.1%) and F1-score (94.9%) by the 100th epoch, indicating superior generalization and balance between precision and recall. The EJH-CNN-BiLSTM and ResNet-34 offer strong results in mid-range epochs but plateau earlier than the model. The enhanced learning capabilities and uncertainty modeling in CNN+EDL enable more robust and accurate emotion recognition, which is valuable in real-time and socially assistive technologies.

The results as in Table.5-Table.8 demonstrate that the CNN+EDL model consistently outperforms traditional methods across all seven basic emotions. In terms of accuracy, the model reaches a peak of 96.2% for "Happy" and maintains strong scores for challenging classes like "Disgust" (91.3%) and "Fear" (90.2%), which are often confused with other emotions. Compared to the CNN and MobileNet-V1, which tend to underperform on subtle emotions, the method shows a clear advantage due to its enhanced uncertainty handling and hyperparameter tuning using genetic algorithms. Precision and recall values are also significantly higher, indicating the model's reliability in detecting both presence and absence of emotions without generating excessive false positives or negatives. For example, the recall for "Angry" is 90.7%, outperforming the next best (EJH-CNN-BiLSTM) by over 3%. The F1-score, representing the harmonic mean of precision and recall, further supports this conclusion, with an average gain of 3–5% across all classes compared to the strongest baselines. This shows that CNN+EDL not only achieves higher correctness but also maintains class balance even in complex multi-emotion scenarios.

6. CONCLUSION

Under normal circumstances, the model of facial expression recognition can reasonably predict the several types of emotions individuals experience. From happiness to sadness, from anger to disgust, from fear to surprise, from neutrality these emotions run. On the other hand, predicting all these feelings is challenging work. The output of this research leads the research to suggest a facial expression recognition system. The model has been taught using seven distinct facial expressions inspired by the FER2013 collection. After preprocessing, the Haar Cascade approach suggested is applied to extract the features from the obtained facial images. Then the CNN model recognizes the seven fundamental facial expressions. The performance degree of the system is obtained from accuracy as a benchmark. Extraction of the salient features from the facial images using the Haar Cascade technique significantly improves the accuracy of the model, it was observed over the tests.

REFERENCES

[1] M.K. Chowdary, T.N. Nguyen and D.J. Hemanth, "Deep Learning-based Facial Emotion Recognition for Human-Computer Interaction Applications", *Neural Computing and Applications*, Vol. 35, No. 32, pp. 23311-23328, 2023.

[2] A. Moin, F. Aadil, Z. Ali and D. Kang, "Emotion Recognition Framework using Multiple Modalities for an Effective Human-Computer Interaction", *Journal of Supercomputing*, Vol. 79, No. 8, pp. 9320-9349, 2023.

[3] S. Nayak, B. Nagesh, A. Routray and M. Sarma, "A Human-Computer Interaction Framework for Emotion Recognition through Time-Series Thermal Video Sequences", *Computers and Electrical Engineering*, Vol. 93, pp. 1-7, 2021.

[4] Z. Lv, F. Poiesi, Q. Dong, J. Lloret and H. Song, "Deep Learning for Intelligent Human-Computer Interaction", *Applied Sciences*, Vol. 12, No. 22, pp. 1-28, 2022.

[5] N. Rawal and R.M. Stock-Homburg, "Facial Emotion Expressions in Human-Robot Interaction: A Survey", *International Journal of Social Robotics*, Vol. 14, No. 7, pp. 1583-1604, 2022.

[6] F. Xue, Q. Wang, Z. Tan, Z. Ma and G. Guo, "Vision Transformer with Attentive Pooling for Robust Facial Expression Recognition", *IEEE Transactions on Affective Computing*, Vol. 14, No. 4, pp. 3244-3256, 2022.

[7] H. Liu, T. Liu, Z. Zhang, A.K. Sangaiah, B. Yang and Y. Li, "ARHPE: Asymmetric Relation-Aware Representation Learning for Head Pose Estimation In Industrial Human-Computer Interaction", *IEEE Transactions on Industrial Informatics*, Vol. 18, No. 10, pp. 7107-7117, 2022.

[8] R.R. Adyapady and B. Annappa, "A Comprehensive Review of Facial Expression Recognition Techniques", *Multimedia Systems*, Vol. 29, No. 1, pp. 73-103, 2023.

[9] C. Zheng, M. Mendieta and C. Chen, "Poster: A Pyramid Cross-Fusion Transformer Network for Facial Expression Recognition", *Proceedings of International Conference on Computer Vision*, pp. 3146-3155, 2023.

[10] J. Mao, R. Xu, X. Yin, Y. Chang, B. Nie, A. Huang and Y. Wang, "Poster++: A Simpler and Stronger Facial Expression Recognition Network", *Pattern Recognition*, Vol. 157, pp. 1-7, 2025.

[11] A.A. Alnuaim, M. Zakariah, P.K. Shukla, A. Alhadlaq, W.A. Hatamleh, H. Tarazi and R. Ratna, "Human-Computer Interaction for Recognizing Speech Emotions using Multilayer Perceptron Classifier", *Journal of Healthcare Engineering*, Vol. 2022, No. 1, pp. 1-7, 2022.

[12] Z. Zhao, Q. Liu and F. Zhou, "Robust Lightweight Facial Expression Recognition Network with Label Distribution Training", *Proceedings of International Conference on Artificial Intelligence*, Vol. 35, No. 4, pp. 3510-3519, 2021.

[13] M. Aly, "Revolutionizing Online Education: Advanced Facial Expression Recognition for Real-Time Student Progress Tracking Via Deep Learning Model", *Multimedia*

Tools and Applications, Vol. 84, No. 13, pp. 12575-12614, 2025.

[14] Z. Wen, W. Lin, T. Wang and G. Xu, "Distract Your Attention: Multi-Head Cross Attention Network for Facial Expression Recognition", *Biomimetics*, Vol. 8, No. 2, pp. 1-17, 2023.

[15] S.C. Leong, Y.M. Tang, C.H. Lai and C.K.M. Lee, "Facial Expression and Body Gesture Emotion Recognition: A Systematic Review on the Use of Visual Data in Affective Computing", *Computer Science Review*, Vol. 48, pp. 1-6, 2023.

[16] B. Li and D. Lima, "Facial Expression Recognition via ResNet-50", *International Journal of Cognitive Computing in Engineering*, Vol. 2, pp. 57-64, 2021.

[17] W. Zhang, X. Ji, K. Chen, Y. Ding and C. Fan, "Learning a Facial Expression Embedding Disentangled from Identity", *Proceedings of International Conference on Computer Vision and Pattern Recognition*, pp. 6759-6768, 2021.

[18] Y. Nan, J. Ju, Q. Hua, H. Zhang and B. Wang, "A-MobileNet: An Approach of Facial Expression Recognition", *Alexandria Engineering Journal*, Vol. 61, No. 6, pp. 4435-4444, 2022.

[19] Y. Li, Z. Qiu, H. Kan, Y. Yang, J. Liu, Z. Liu and N.Y. Kim, "A Human-Computer Interaction Strategy for an FPGA Platform Boosted Integrated "Perception-Memory" System based on Electronic Tattoos and Memristors", *Advanced Science*, Vol. 11, No. 39, pp. 1-8, 2024.

[20] K. Mohan, A. Seal, O. Krejcar and A. Yazidi, "FER-Net: Facial Expression Recognition using Deep Neural Net", *Neural Computing and Applications*, Vol. 33, No. 15, pp. 9125-9136, 2021.

[21] W. Zhang, F. Qiu, S. Wang, H. Zeng, Z. Zhang, R. An and Y. Ding, "Transformer-based Multimodal Information Fusion for Facial Expression Analysis", *Proceedings of International Conference on Computer Vision and Pattern Recognition*, pp. 2428-2437, 2022.

[22] Z. Zhao and Q. Liu, "Former-DFER: Dynamic Facial Expression Recognition Transformer", *Proceedings of International Conference on Multimedia*, pp. 1553-1561, 2021.

[23] Y. Wang, Y. Sun, Y. Huang, Z. Liu, S. Gao, W. Zhang and W. Zhang, "Ferv39k: A Large-Scale Multi-Scene Dataset for Facial Expression Recognition in Videos", *Proceedings of International Conference on Computer Vision and Pattern Recognition*, pp. 20922-20931, 2022.

[24] C. Bisogni, A. Castiglione, S. Hossain, F. Narducci and S. Umer, "Impact of Deep Learning Approaches on Facial Expression Recognition in Healthcare Industries", *IEEE Transactions on Industrial Informatics*, Vol. 18, No. 8, pp. 5619-5627, 2022.

[25] W. Li, Y. Cui, Y. Ma, X. Chen, G. Li, G. Zeng and D. Cao, "A Spontaneous Driver Emotion Facial Expression (Defe) Dataset for Intelligent Vehicles: Emotions Triggered by Video-Audio Clips in Driving Scenarios", *IEEE Transactions on Affective Computing*, Vol. 14, No. 1, pp. 747-760, 2021.

[26] G.R.E. Said, "Metaverse-based Learning Opportunities and Challenges: A Phenomenological Metaverse Human-Computer Interaction Study", *Electronics*, Vol. 12, No. 6, pp. 1-13, 2023.

[27] Y. Zhang, C. Wang, X. Ling and W. Deng, "Learn from All: Erasing Attention Consistency for Noisy Label Facial Expression Recognition", *Proceedings of International Conference on Computer Vision*, pp. 418-434, 2022.

[28] W. Tabone and J. De Winter, "Using ChatGPT for Human-Computer Interaction Research: A Primer", *Royal Society Open Science*, Vol. 10, No. 9, pp. 1-9, 2023.