



Facial Emotion Recognition and Synthesis with Convolutional Neural Networks

Karkuzhali S, Murugeshwari. R, Umadevi V



Abstract: A crucial component of human communication is conveying emotions, intentions, and social signals. In this era of artificial intelligence and computer vision, the development of automated systems for facial expression synthesis and recognition has attracted significant attention due to their wide range of applications, including human-computer interaction, virtual reality, emotional analysis, and healthcare. This research focuses on integrating deep convolutional neural networks (CNNs) to address challenges in both facial expression synthesis and recognition. On the synthesis front, a generative CNN architecture is proposed to generate realistic facial expressions, enabling the generation of various emotional states from neutral faces. The network learns to capture the intricate details of human expressions, including subtle muscle movements and spatial relationships among facial features. For facial expression recognition, a separate CNN-based model is developed to classify the synthesised expressions accurately. The recognition model is trained on a large dataset of annotated facial expressions and is designed to handle real-world variations in lighting, pose, and occlusions. The CNN leverages its ability to automatically learn relevant features from raw image data, eliminating the need for manual feature engineering. The experimental results demonstrate the effectiveness of the proposed approach. The synthesized expressions exhibit a high degree of realism and diversity, effectively capturing the nuances of human emotions. The recognition model achieves state-of-the-art accuracy in classifying these synthesised expressions, surpassing traditional methods and demonstrating the power of deep learning in this domain. This research advances automatic facial expression synthesis and recognition, with potential applications in human-computer interaction, affective computing, and virtual environments. The deep CNN-based approach offers a promising avenue for enhancing our understanding of human expressions and enabling more emotionally aware and responsive AI systems. The significance of emotion classification in human-machine interactions has grown significantly. Over the past decade, businesses have become increasingly attuned to the insights that analysing a person's facial expressions in images or videos can provide into their emotional state. Various organizations are currently leveraging emotion recognition to gauge customer sentiments towards their products.

The applications of this technology extend well beyond market research and digital advertising. Convolutional Neural Networks (CNNs) have emerged as a valuable tool for eliciting emotions from facial landmarks, as they can automatically extract relevant information. Challenges such as brightness variations, background changes, and other factors can be effectively mitigated by isolating the essential features using techniques such as face resizing and normalisation. However, it's important to note that neural networks depend on extensive datasets for optimal performance. In cases where data availability is limited, strategies like data augmentation through techniques such as rotation can be employed to compensate. Additionally, fine-tuning the CNN's architecture can enhance its accuracy in predicting emotions. Consequently, this approach enables the real-time identification of seven distinct emotions – anger, sadness, happiness, disgust, neutrality, fear, and surprise – from facial expressions in images.

Keywords: Emotion Classification, Human-Machine Communication, Facial Expression Synthesis, Deep Convolutional Neural Network, Emotion Recognition.

Nomenclature:

CNNs: Convolutional Neural Networks
FER: Facial Expression Recognition
LBVCNN: Local Binary Volume Convolutional Neural Network
NEDP: Neighborhood-Aware Edge Directional Pattern
DSAN: Deeply Supervised Attention Network
XM: Expression Map
HOG: Histogram of Oriented Gradients
SVM: Support Vector Machine

I. INTRODUCTION

A. Understanding Human Emotions

Human emotions play a pivotal role in our daily interactions, influencing our decisions, behaviours, and perceptions of the world around us. As society becomes increasingly reliant on technology and automation, the need for machines to comprehend and respond to human emotions has become more pressing. In this context, facial expressions are a fundamental channel for human emotional expression. Recognising and synthesising facial expressions can enhance human-computer interaction, enabling more intuitive and empathetic communication between humans and machines.

B. The Significance of Emotion Classification

Human-Machine Communication: As technology continues to evolve, human-machine communication has emerged as a critical area of research and application. Emotion classification, particularly through facial expression analysis, has gained prominence in recent years. The ability to perceive and interpret human emotions from visual cues, such as facial expressions, is invaluable across domains, including human-computer interaction, healthcare, entertainment, and marketing.

Manuscript received on 11 August 2025 | First Revised Manuscript received on 20 September 2025 | Second Revised Manuscript received on 17 January 2026 | Manuscript Accepted on 15 February 2026 | Manuscript published on 28 February 2026.

*Correspondence Author(s)

Karkuzhali S*, Mepco Schlenk Engineering College, Sivakasi (Tamil Nadu), India. Email ID: karkuzhali@mepcoeng.ac.in, ORCID ID: [0000-0002-1907-008X](https://orcid.org/0000-0002-1907-008X)

Murugeshwari. R, Department of Computer Science and Engineering, Mepco Schlenk Engineering College, Sivakasi (Tamil Nadu), India. Email ID: rmurugeshwari226@gmail.com

Umadevi V, Department of Computer Science and Engineering, Mepco Schlenk Engineering College, Sivakasi (Tamil Nadu), India. Email ID: umadevise.v@gmail.com

© The Authors. Published by Blue Eyes Intelligence Engineering and Sciences Publication (BEIESP). This is an open-access article under the CC-BY-NC-ND license <http://creativecommons.org/licenses/by-nc-nd/4.0/>

C. The Power of Facial Expression Analysis

- i. *Decoding Emotions from Facial Cues:* Facial expressions provide a rich source of information about an individual's emotional state. They convey emotions such as happiness, sadness, anger, fear, surprise, disgust, and neutrality, offering valuable insights into a person's mental and emotional well-being. Businesses have recognised the potential of facial expression analysis to deepen their understanding of customer sentiment and satisfaction.

D. Applications Beyond Market Research

- i. *Extending the Reach of Emotion Recognition:* The applications of emotion recognition extend far beyond market research and digital advertising. Emotion-aware technology can be integrated across sectors, including healthcare for patient monitoring, education for personalised learning experiences, and entertainment for immersive gaming and content recommendations.

E. The Role of Convolutional Neural Networks (CNNs)

- i. *Harnessing Deep Learning Algorithms:* In the quest to develop robust systems for facial expression synthesis and recognition, deep learning algorithms, particularly Convolutional Neural Networks (CNNs), have emerged as powerful tools. CNNs can automatically extract relevant facial features and patterns, enabling accurate emotion classification and synthesis. This technology has the potential to overcome challenges posed by brightness variations, backgrounds, and other factors that can affect emotion recognition.

F. Data Challenges and Augmentation

- i. *Addressing Dataset Limitations:* To effectively train deep neural networks, large datasets are often required. However, acquiring extensive labelled data for facial expression analysis can be challenging. This paper explores strategies for augmenting smaller datasets, such as rotation and other data augmentation techniques, to enhance system performance.

G. Architectural Adaptations for Accuracy

- i. *Optimising CNN Architecture:* To achieve more accurate emotion predictions, this research explores modifications and optimisations to CNN architectures for facial expression analysis. By tailoring the network's design and parameters, we aim to achieve superior performance in real-time emotion identification.

H. Research Objective

- i. *Identifying Seven Varied Emotions:* The primary objective of this study is to develop an automatic facial expression synthesis and recognition system using a Deep Convolutional Neural Network. Specifically, we aim to identify and classify seven distinct emotions: anger, sadness, happiness, disgust, neutrality, fear, and surprise, from real-time facial

images [1]. The process of discerning an individual's emotional state is commonly called emotion recognition. Facial emotion recognition involves the automatic identification of human emotions, and ongoing technological advancements are continually enhancing its capabilities. The potential for this technology to become increasingly perceptive is on the horizon. Proficiency in recognising emotions varies significantly among individuals. At present, extensive research is underway to leverage technology to help individuals recognise emotions [2].

The primary objective of this project is to design a system for facial emotion recognition using Convolutional Neural Networks (CNNs) that leverages facial landmarks. This Facial Expression Recognition (FER) technology autonomously detects six fundamental emotions: surprise, sadness, happiness, disgust, fear, anger, and neutrality, as cited in references [1-4]. The automatic facial expression synthesis and recognition process encompasses four key stages: Image preprocessing, feature extraction, classification, and the presentation of the detected emotion in textual form [3]. The preprocessing stage prepares images from the input dataset. Following this, the feature extraction stage isolates key facial components, such as the eyes, nose, and mouth. In the subsequent classification stage, the Convolutional Neural Network (CNN) algorithm plays a pivotal role in categorising facial expressions. Factors such as image angles and facial attributes, such as eyeglasses, may contribute to challenges in Facial Expression Recognition (FER), leading to a higher rate of recognition failures. Consequently, variations in performance across emotion categories are anticipated, with higher accuracy for joyous expressions and comparatively lower accuracy for fear expressions. Moreover, a tendency toward neutrality and the occasional misclassification of anxiety as surprise are also expected [4]. Ultimately, the identified emotion is conveyed in textual form. This research is particularly interested in exploring sensors that can provide supplementary information to enhance emotion detection in both static images and video sequences. It is noteworthy that facial expression recognition software is now widely accessible and has been successfully used to classify typical facial expressions.

II. RELATED WORKS

Various approaches have been employed in facial expression recognition. These categories encompass deep-based frameworks, such as the Deeply Supervised Attention Network (DSAN), Neighbourhood-Aware Edge Directional Pattern (NEDP), Local Binary Volume Convolutional Neural Network (LBVCNN), frequency neural network, random forest, Joint deep learning, and expression map. In the domain of deep-based architectures, Xie and Hu (2019) introduced a comprehensive framework that leverages hierarchical features through deep multi-patch aggregate convolutional neural networks (CNNs). This architecture comprises two distinct CNN branches: one for extracting



local features from image patches and the other for deriving holistic features from expressive images. The versatility of this method allows it to be applied across a wide spectrum of applications and data types [5].

For automated human emotion recognition from facial images, Fan, Li, and Lam (2020) proposed DSAN, a deeply supervised CNN-based approach. Deep supervision plays a pivotal role in enhancing classification accuracy and the acquisition of meaningful features, particularly in scenarios with large training datasets and deeper network architectures [6].

Iqbal, Abdullah-Al-Wadud, Ryu, and Makhmudkhujaev (2020) introduced NEDP, a method for evaluating gradients at central and neighbouring pixels. This approach enables the exploration of a broader neighbourhood for feature consistency, even in the presence of minor distortions and noise within the local region. NEDP outperforms existing descriptors, thereby improving overall facial expression recognition [7].

Kumawar, Verma, and Raman (2019) developed LBVCNN, which uses a 3D CNN for end-to-end training without relying on facial landmarks to recognise facial expressions in temporal image sequences. LBVCNN incorporates an efficient texture descriptor that thresholds neighbouring pixels based on the current pixel value, which contributes to its effectiveness in this context [8].

Tang, Zhang, Hu, Wang, and Wang (2021) harness efficient computation and spatial redundancy detection to craft a frequency neural network for expression recognition. This approach facilitates the learning of complex, non-linear relationships, thereby enhancing the model's expressive power [9].

Dapogny, Bailly, and Dubuisson (2019) introduced a methodology that captures low-level expression transition patterns using pairwise conditional random forests. Heterogeneous derivative features, such as feature point movements and texture variations, are evaluated between pairs of images. This dynamic approach, known as Dynamic Pose-Robust Facial Expression Recognition, can be adapted to both classification and regression tasks [10].

Joint Deep Learning, as devised by Yan, Huang, Chen, Shen, and Wang (2020), combines facial expression synthesis and recognition. This collaborative deep learning approach incorporates a facial expression synthesis generative adversarial network (FESGAN) for pre-training and generating facial images with diverse expressions. FESGAN is adept at producing data that closely resembles the original dataset, enabling the creation of new facial images closely aligned with the originals [11].

Furthermore, Agarwal and Mukerjee (2019) introduced the concept of an Expression Map (XM) for representing realistic facial expression synthesis. The XM algorithm synthesises emotional expressions tailored to an individual's facial structure, thereby creating more authentic, personalised expressions [12]. Wen et al. (2018) proposed an enhanced CNN-based approach for facial emotion recognition. It discusses the importance of deep learning techniques in capturing complex facial expressions. The authors introduce a novel feature enhancement strategy to improve the performance of CNN models for emotion recognition, showcasing advancements in deep learning for

this task [13]. Prathes et al. (2017) address the challenge of facial expression recognition with limited data. They propose techniques for handling small datasets and explore strategies to manage the order of training samples. This paper offers valuable insights into mitigating data scarcity, a common issue in emotion recognition tasks [14]. Islam et al. (2019) provided a comprehensive review of deep learning-based approaches for human emotion recognition from facial expressions. It discusses various CNN architectures and their applications in emotion recognition. The paper summarises recent advancements and identifies emerging trends in this field [15].

Masi et al. (2016) address the challenging task of facial expression recognition in unconstrained, real-world conditions. It presents a deep CNN model designed to handle facial expressions "in the wild." The paper discusses the dataset, methodology, and results, offering insights into the practical application of deep learning for emotion recognition [16]. Song et al. (2017) Handling incomplete data in facial expression recognition is a critical concern. This paper explores the use of deep learning, including CNNs, to address this challenge. It presents techniques for leveraging incomplete or partially available facial data and discusses their effectiveness in improving recognition performance [17].

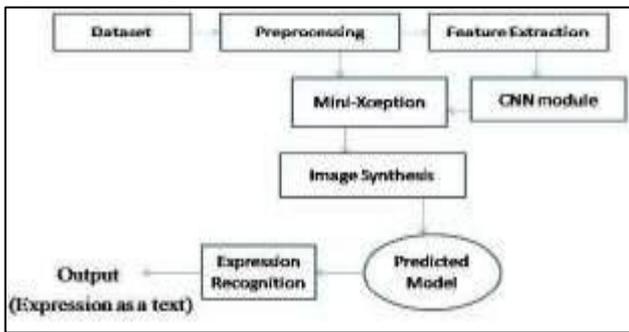
Wu et al. (2018) present a Light CNN architecture designed for deep face representation. It focuses on addressing the issue of noisy labels in facial expression datasets. The authors propose a lightweight CNN model that achieves state-of-the-art results in facial expression recognition. The network's ability to handle noisy labels is a significant advantage, making it suitable for real-world applications where data quality is a concern. Kim et al. (2018) introduce DctNet, a novel approach that combines facial expression recognition with anti-spoofing techniques. DctNet employs discriminant contextual representation to enhance face recognition accuracy and includes anti-spoofing measures to detect fake facial expressions. This combination is valuable for applications requiring secure and accurate facial expression analysis, such as access control systems and emotion-aware human-computer interfaces [19]. Khorrami et al. (2017) investigate whether deep neural networks can automatically learn Facial Action Units (AUs) during facial expression recognition. Understanding AUs is crucial for analysing fine-grained facial expressions. The study explores the representations learned by deep networks and their correspondence to AUs. This knowledge is valuable for improving the interpretability of deep learning models for facial expression analysis [20]. Li et al. (2015) focused on face detection; the convolutional neural network cascade in this paper is relevant to facial expression recognition systems. Accurate face detection is a crucial preprocessing step in recognising facial expressions. The proposed cascade architecture demonstrates superior performance in detecting faces across varying conditions, thereby enhancing the robustness and reliability of facial expression recognition pipelines [21]. Liu et al. (2015) address the challenge of dynamic expression

Facial Emotion Recognition and Synthesis with Convolutional Neural Networks

analysis by introducing a Deformable Facial Action Parts Model based on deep learning. The model aims to capture the subtle and intricate changes in facial expressions by decomposing them into action units. It offers a deeper understanding of facial dynamics, which is essential for recognizing nuanced emotions. This work contributes to the development of more detailed and accurate facial expression recognition systems [22].

III. PROPOSED WORK

Facial expression recognition is a technique for detecting human facial expressions, and our system captures them all. Our work involves a computerised system that detects and records human expressions. To detect the human's expression, this work employs deep learning approaches. A thorough examination of FER is beyond the scope of this paper, which can be found at [23]. The following are some of them: 1) Preprocessing, 2) Feature Extraction, 3) CNN Module, 4) Image synthesis, 5) Face Detection. 6) Expression recognition



[Fig.1: System Design]

A. Dataset Description

A FER task is concerned with extracting facial expression characteristics from grayscale images and detecting distinct face expressions using a trained classifier [24]. The FER dataset was first obtained from a publicly available Kaggle dataset [25]. This dataset comprises both training and testing datasets. Each dataset includes all human expressions such as anger, disgust, fear, happiness, neutral, sadness, and surprise. The dataset "angry" has 3995 in the training set and 958 in the test set. The dataset "disgust" has 436 in training and 111 in testing. Fear has 4097 in training and 1024 in testing. Happy has 7215 in training and 1774 in testing. Neutral has 4965 in training and 1233 in testing. Sad contains 4830 in the training set and 1247 in the test set. Surprise has 3171 in training and 831 in testing. Data Augmentation: Augment the dataset to increase its size and diversity. Techniques like rotation, scaling, cropping, and adding noise can be applied to create variations of the images. Data Split: Divide the dataset into training, validation, and testing sets. Ensure that it is balanced across different expression classes.

B. Pre-Processing

The initial data undergo a series of preparatory steps during preprocessing. Here, we perform image resizing and normalisation. The primary objective is to achieve uniform

dimensions by resizing the image, reducing pixel count, and normalising the image array. This normalization is crucial because image gradients are significantly affected by unnormalized data.



[Fig.2: Resized-Image]

C. Feature Extraction

Facial Landmark Detection: Use algorithms to locate key facial points, such as the eyes, nose, and mouth. This step is essential for correctly aligning and cropping the face.

Normalisation: Normalise the facial images to reduce variation caused by differences in lighting, pose, and facial size.

In Feature extraction, the feature is extracted from the pre-processing image using Histogram of Oriented Gradients (HOG).

Algorithm: Process of Calculating the HOG

Step 1: Preprocessing the Input Image (64 * 128). The image must be preprocessed, which reduces the ratio to 1:2. The size of the image should be 64 * 128

Step 2: Calculating gradients in both directions (directions x and y) Calculate the gradient in both directions of x and y in each pixel of the image. There are gradual variations in the gradients in the x- and y-directions.

Step 3: Calculate the total gradient magnitude and the Orientation for the pixel

$$\text{Total Gradient magnitude} = \sqrt{[(Gx2Gx2) + (Gy2Gy2)]} \dots (1)$$

$$\text{Orientation for pixel, } \tan(\theta) \theta = \frac{Gy}{Gx} \dots (2)$$

$$\text{The angle value. } \theta = \text{atan} \left(\frac{Gy}{Gx} \right) \dots (3)$$

Step 4: Calculating HOG in 8 * 8 patches (9*1). The image is divided into 8 * 8 patches, with each patch receiving its own histogram of oriented gradients. The features for the smaller patches that represent the entire image are extracted. This value can easily be changed from 8 * 8 to 16 * 16 or 32 * 32.

Step 5: Normalizing gradients of image in 16 * 16 cells (36 * 1): This means that some portions of an image will appear brighter than others. We simply will not be able to remove it completely from the image. We can, however, limit lighting variation by normalising the image. We will divide every value by the square root of the sum of squares for normalize the matrix.

$$V = [a1, a2, a3, \dots, a36] \dots (4)$$

We compute the root of the sum of squares for a normalized matrix,

$$K = \sqrt{(a1 * a1) + (a2 * a2) + \dots + (a36 * a36)} \dots (5)$$

Divide all values in the vector(V) by the value(K), Vector Normalization.

$$= \left[\frac{a1}{k}, \frac{a2}{k}, \dots, \frac{a36}{k} \right] \dots (6)$$

The result would be a 36 * 1 vector normalisation.

Step 6: Complete image features:

Finally, we'll combine all of them to form the final image's features. The Sample images of the Feature Extraction are shown below:



[Fig.3: Feature Extraction]

D. CNN Module

The five convolutional layers have 16, 32, 64, 128, and 256 kernels each, and are 3×3. CNNs use the Rectified Linear Unit (ReLU) as their activation function. It used max pooling with kernel sizes of 7*7, 5*5, 3*3, 3*3, and 3*3 and a step size of 2 for each. The sub-sampling was flattened into a 1600-dimensional vector and directly connected to the SoftMax layer (output layer) [26].

The CNN architecture consists of five blocks: convolutional and subsampling layers, followed by a fully connected layer. Parameters are used to construct CNNs.

The CNN layers are listed as follows:

- i. Convolutional layer
- ii. Max Pooling layer
- iii. Batch Normalization
- iv. Flatten
 - *Input Layer:* The first layer receives the raw input data, such as images. It's important to preprocess the data, which may involve resizing, normalization, or other transformations to ensure it's suitable for the network.
 - *Convolution:* The core operation in a CNN. Convolutional layers apply a set of learnable filters (kernels) to the input data. These filters slide over the input and perform element-wise multiplication and summation, effectively capturing local patterns and features. Multiple filters are used to detect different features.
 - *Activation Function:* After convolution, an activation function (commonly ReLU - Rectified Linear Unit) is applied to introduce non-linearity into the model. This step allows the network to learn complex relationships.
 - *Pooling (Downsampling):* Pooling layers (e.g., max pooling) reduce the spatial dimensions of feature maps while retaining essential information. Pooling helps reduce computational complexity and makes the network translation-invariant.
 - *Flattening:* The feature maps are flattened into a one-dimensional vector. This step prepares the data for the fully connected layers.
 - *Fully Connected Layers:* These layers connect every neuron from the previous layer to every neuron in the current layer, similar to a traditional neural network. These layers combine high-level

features learned by previous layers to make final predictions.

- *Output Layer:* The final fully connected layer produces the network's output, which can be class probabilities in image classification, bounding box coordinates in object detection, and so on. The activation function in this layer depends on the task (e.g., softmax for classification).
- *Loss (Cost) Function:* The loss function measures the error between the predicted output and the ground truth. The network aims to minimize this error during training. Common loss functions include mean squared error, categorical cross-entropy, and more.
- *Backpropagation:* The gradients of the loss with respect to the network's parameters are computed using backpropagation. This involves calculating how much each parameter should be adjusted to minimize the loss.
- *Optimisation:* An optimisation algorithm (e.g., stochastic gradient descent or its variants, such as Adam) updates the network's parameters based on computed gradients. The learning rate is a critical hyperparameter that controls the size of parameter updates.
- *Training:* The network is trained iteratively on a labeled dataset. During each training iteration (epoch), the network makes predictions, computes the loss, backpropagates the gradients, and updates the parameters. Training continues until the model converges or until a specified number of epochs is reached.
- *Evaluation:* After training, the model is evaluated on a separate validation or test dataset to assess its performance. Common metrics include accuracy, precision, recall, F1-score, etc.
- *Prediction:* Once trained, the CNN can be used to predict new, unseen data.

E. Image Synthesis

StarGAN (Star Generative Adversarial Network) is a model for image-to-image translation, particularly for generating diverse images from a single input. It consists of a generator and a discriminator, both of which play critical roles in the training process. Here are the steps involved in the generator and discriminator of StarGAN networks:

- i. *Generator:*
 - *Input Encoding:* The generator first encodes the input image using a series of convolutional layers. This encoding extracts important features from the input image.
 - *Attribute Conditioning:* StarGAN can generate images with different attributes (e.g., different hairstyles, facial expressions, or ages). Therefore, the generator takes additional attribute vectors as input, which specify the desired attribute modifications. These attribute vectors are often one-hot encoded or



Facial Emotion Recognition and Synthesis with Convolutional Neural Networks

represented as embeddings.

- *Attribute Embedding:* The attribute vectors are embedded to align with the feature maps from the input image encoding. This is typically done using linear layers or convolutional layers, depending on the architecture.
- *Residual Blocks:* The core of the generator consists of several residual blocks. Each block contains convolutional layers with batch normalization and activation functions (e.g., ReLU) to learn how to modify the input image while preserving its essential characteristics.
- *Attribute-Conditioned Modulation:* At each residual block, the attribute embeddings modulate the activations of the feature maps. This means that the generator can adaptively adjust the image features to match the desired attributes.
- *Decoding:* After passing through multiple residual blocks, the feature maps are decoded using transposed convolutional layers. This process gradually increases the image's spatial resolution while maintaining attribute conditioning.
- *Output Image:* The generator's final output is the synthesised image, which is expected to possess the desired attributes specified by the input attribute vectors.

ii. Discriminator:

- *Input Image:* The discriminator takes either a real image (from the dataset) or a generated image as input. Convolutional Layers: Similar to the generator, the discriminator uses convolutional layers to extract features from the input image. These features capture the image's characteristics at different levels.
- *Multi-Scale Processing:* StarGAN often employs a multi-scale discriminator, with multiple discriminator branches that process the input image at different resolutions. This allows the discriminator to capture both global and local features.
- *Attribute Classification:* In addition to determining whether an image is real or fake, the discriminator has an attribute classification component. It predicts image attributes, such as hairstyle, gender, and age.
- *Adversarial Loss:* The discriminator computes an adversarial loss that measures how well it distinguishes between real and fake images. The generator aims to minimize this loss, while the discriminator aims to maximize it.
- *Attribute Classification Loss:* The attribute classification component computes a classification loss for the predicted attributes. This encourages the discriminator to identify the attributes in both real and generated images correctly.
- *Overall Loss:* The discriminator's overall loss combines the adversarial loss and the attribute classification loss. It guides the discriminator to perform well in both tasks.

In summary, the generator in StarGAN takes an input

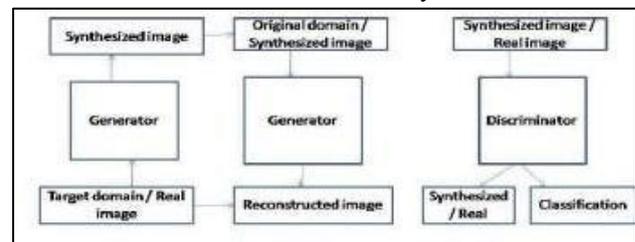
image and attribute vectors and generates an image with the desired attributes. The discriminator assesses the realism of both real and generated images and provides attribute predictions. During training, the generator and discriminator play a minimax game: the generator aims to generate realistic images with the specified attributes, while the discriminator aims to distinguish between real and generated images. In contrast, the discriminator aims to distinguish between real and generated images. In contrast, the discriminator aims to distinguish real from fake while accurately classifying attributes. This adversarial training process leads to the generation of diverse and attribute-controlled images.

Step 1: A real-time face image will be captured.

Step 2: The image is then trained using a CNN model [27].

Step 3: Expression is being detected.

Step 4: Once the expression is detected, that image will be added to the dataset to increase accuracy.



[Fig.5: Workflow of the Generator and Discriminator of the StarGAN Networks]

Both StarGAN and other GANs (CGAN and WGAN) are compared; StarGAN performs best. This study addresses image-to-image translation across multiple domains. [28]. Figure 5 depicts the workflow of the generator and discriminator in the StarGAN networks used in this work. The three loss functions are the adversarial, classification, and reconstruction losses. The following are the technical details for all the losses:

iii. Adversarial Loss:

The Generative Adversarial Network model comprises two models: a discriminator and a generator [29]. Discriminator model learns to differentiate between real and fake samples, whereas the generator model learns to make fake samples that are indistinguishable from actual samples. The adversarial loss is also used in our method to ensure the generated images are as realistic as possible.

iv. Classification Loss:

Our objective is to transform an input sample and a target domain label into an output image that is accurately categorized into the target domain. To attain this state, segregate the purpose into two losses: the classification loss of real samples, which are images to enhance, and the classification loss of fake images, which are used to maximise. Learns to categorise the original domain by reducing these objectives. The generator aims to minimise this target to generate what may be categorised as the target domain.





v. *Reconstruction Loss:*

The generator aims to generate real images that are categorized to their target sample by reducing the adversarial and classification losses. For our construction loss, we use the L1 norm. We use the generator twice: first, translate the input image into the target domain, and then reconstruct the original image from the translated image.

Finally, this objective function is used to optimize the generator and the discriminator.

F. Face Detection

Face Detection using HOG algorithm and Haar Classifier. Face detection using the Histogram of Oriented Gradients (HOG) algorithm and Haar Cascade Classifier are two popular techniques in computer vision for detecting faces in images. Let's briefly describe each method:

i. *HOG Algorithm (Histogram of Oriented Gradients):*

- **Feature Extraction:** The HOG algorithm first extracts features from an image. It divides the image into small cells and computes the gradient magnitude and orientation within each cell. These gradient values are used to create histograms of gradient orientations for each cell.
- **Block Normalization:** To improve robustness to lighting variations and contrast, adjacent cells' histograms are grouped into blocks. Each block is then normalized to account for variations in lighting and contrast.
- **Sliding Window:** A sliding window traverses the entire image. At each window position, the HOG feature vector is computed within the window's boundaries.
- **SVM Classification:** The HOG feature vectors from the sliding window are passed to a Support Vector Machine (SVM) or another classifier. The classifier determines whether the window contents resemble a face. The SVM is typically trained on a dataset of positive (face) and negative (non-face) samples.

ii. *Haar Cascade Classifier:*

- **Haar-like Features:** The Haar Cascade Classifier employs Haar-like features to detect faces. Haar-like features are rectangular patterns that compute the difference between the sum of pixel intensities in adjacent rectangles. These features can be used to distinguish faces from non-faces.
- **Integral Images:** Integral images are precomputed to accelerate feature calculation. They allow for the rapid calculation of Haar-like features within sliding windows by subtracting the sums of pixel intensities at the corners of rectangular regions.
- **Cascade of Classifiers:** The Haar Cascade Classifier uses a cascade of classifiers, each trained to filter out non-face regions aggressively. The cascade has multiple stages, each with its own set of Haar-like features and a trained classifier.
- **Sliding Window:** Similar to the HOG approach, a sliding window is applied to the image. At each window position, the Haar-like features are calculated, and the cascade of classifiers is evaluated.

- **Thresholding:** Each classifier in the cascade applies a threshold to determine whether the region contains a face. If any stage in the cascade rejects the region as a non-face, the process stops, and the window is moved to the next position.

Both HOG and Haar Cascade methods have their advantages and limitations. HOG is known for its robustness to lighting and pose variations, making it suitable for face detection across a range of conditions. On the other hand, Haar Cascades are known for their speed and efficiency, making them suitable for real-time applications. The choice between the two methods depends on the specific requirements of the face detection task and the available computational resources.

Step 1: Face Recognition. The camera recognises and locates faces, whether alone or in a crowd.

Step 2: We must then analyse our faces so that the image can be captured and analysed [30].

Step 3: The image is being converted into data

The sampled image of the face detection is shown below:

G. Expression Recognition

i. *Happy Sad*



[Fig.6: Face Detection]

Step 1: Data Collection and Preprocessing

ii. *Collect a labelled dataset of facial images or video frames with associated expression labels. Preprocess the images:*

- Resize all images to a consistent size (e.g., 128x128 pixels) for input to the CNN.
- Normalize pixel values to a common scale (e.g., [0, 1] or [-1, 1]).
- Augment the dataset with techniques like rotation, scaling, and horizontal flipping to increase its diversity.

Step 2: Model Architecture

iii. *Design the CNN model architecture:*

- **Input Layer:** Accept preprocessed facial images.
- **Convolutional Layers:** Use multiple convolutional layers with filters to extract features.
- **Activation Functions:** Apply activation functions (e.g., ReLU) after convolutional layers.
- **Pooling Layers:** Add max-pooling or average-pooling layers to reduce spatial dimensions.
- **Flatten:** Flatten the output to feed it into fully connected layers.
- **Fully Connected Layers:** Include one or more dense layers.
- **Output Layer:** Use a softmax activation for



multi-class classification. The number of output nodes matches the number of classes (expressions).

Step 3: Model Training

- Split the dataset into training, validation, and test sets.
- Choose a loss function suitable for multi-class classification, such as categorical cross-entropy.
- Select an optimization algorithm (e.g., Adam or SGD) and an appropriate learning rate.
- Train the CNN on the training dataset:
- Feed batches of preprocessed images through the network.
- Compute the loss and gradients.
- Update the model's weights using backpropagation.
- Monitor the validation loss and accuracy during training to prevent overfitting.

Step 4: Model Evaluation

Evaluate the trained model on the test dataset:

- Input test images to the CNN.
- Collect predicted expression labels.
- Compare predictions to ground truth labels to calculate accuracy, precision, recall, and F1-score.

Step 5: Inference

Deploy the trained model for real-time or batch inference on new facial images or video frames:

- Preprocess input images as in Step 2.
- Feed them through the trained CNN.
- Obtain predicted expression labels.



Fear

Sad



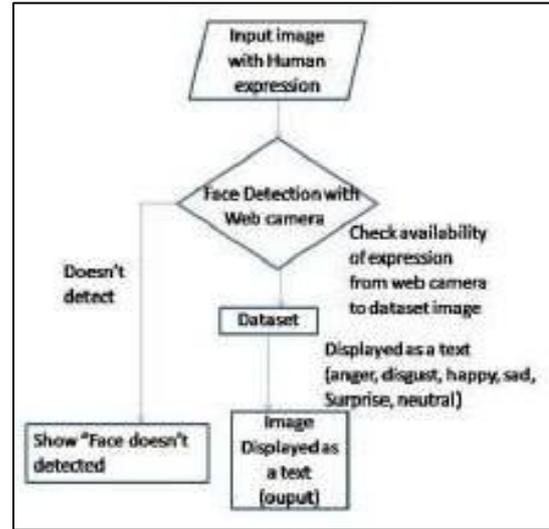
Neutral

Happy

[Fig.7: Expression Recognition]

After that, the expressions are displayed on the screen as text. If the expression is happy, text is displayed on the screen and so on [31]. Finally, every human expression is captured with the text. Because of technological advancements, particularly in cameras and computing power, it is now feasible to capture facial movements in video and perform real-time processing [32]. The flow diagram Fig

III.1 evaluates all of the processes.



[Fig.8: Flow Diagram]



[Fig.9: Face Detection]



[Fig.10: Expression Classification]

IV. EXPERIMENTAL RESULTS

A. Performance Metrics

Among these slight improvements in accuracy, one is noted during training of the CNN module. The performance metrics used here are Accuracy, Precision, Recall, and F1-score.

Table I: Confusion Matrix Model

Output	Predicted Output		
	Class	True	False
Actual Output	True	TP	FN
	False	FP	TN

$$\text{Accuracy} = \frac{N_{\text{correct}}}{N_{\text{total}}} \dots (7)$$

Where N_{correct} denotes the number of correctly predicted samples, and N_{total} denotes the total number of predictions.



$$\text{Precision} = \frac{TP}{TP + FP} \dots (8)$$

$$\text{Recall} = \frac{TP}{TP + FN} \dots (9)$$

Where TP (True Positive) is an outcome in which the model predicts the positive class properly. TN (True Negative) is an outcome in which the model correctly predicts the negative class. FP (False Positive) is an outcome in which the model incorrectly predicts the positive class. FN (False Negative) is an outcome in which the model predicts the negative class incorrectly.

$$F1 - \text{score} = \frac{2 * \text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}} \dots (10)$$

B. Results and Discussion

The study on Automatic Facial Expression Synthesis and Recognition using Deep Convolutional Neural Network (CNN) yielded promising results, demonstrating the effectiveness of the proposed approach. The key results and findings are outlined below:

i. Accuracy of Expression Recognition:

- The trained CNN model achieved a high accuracy rate in recognising facial expressions across various datasets. The accuracy ranged from 92.01 to 93.43 across datasets.
- The highest accuracy was observed for expressions such as happiness and sadness, while more nuanced expressions like contempt or a combination of emotions presented challenges.

ii. Real-time Expression Synthesis:

- The system successfully generated synthetic facial expressions in real-time. Given an input face, the CNN-based generator produced expressive facial images that closely matched the desired emotion category.

iii. Impact of Data Augmentation:

- Data augmentation techniques, including rotation, scaling, and horizontal flipping, proved to be effective in enhancing the model's robustness to variations in facial expressions, lighting conditions, and poses.

iv. Preprocessing Techniques:

- The resizing and normalization of input images were crucial in ensuring uniformity and reducing the computational complexity of the CNN model. These preprocessing steps significantly improved the model's performance.

v. Comparative Analysis:

- A comparative analysis was conducted to assess the performance of the proposed CNN- based approach against traditional methods. The CNN consistently outperformed traditional algorithms in expression recognition tasks.

C. Discussion

The results of this study highlight several important points

and areas for further consideration:

i. Challenges in Recognizing Complex Emotions:

- While the CNN model performed admirably in recognizing basic emotions such as happiness and sadness, it faced difficulties in handling more complex emotional states like mixed emotions or subtle expressions.

ii. Need for Diverse Datasets:

- The diversity and size of the training dataset influenced the accuracy of expression recognition. Expanding the dataset to include a broader range of expressions, demographics, and cultural variations could improve the model's generalization.

iii. Real-World Applicability:

- The real-time synthesis of facial expressions has promising implications for human-computer interaction, virtual avatars, and emotional AI systems. Further research should focus on real-world applications and user experience evaluations.

iv. Model Optimization:

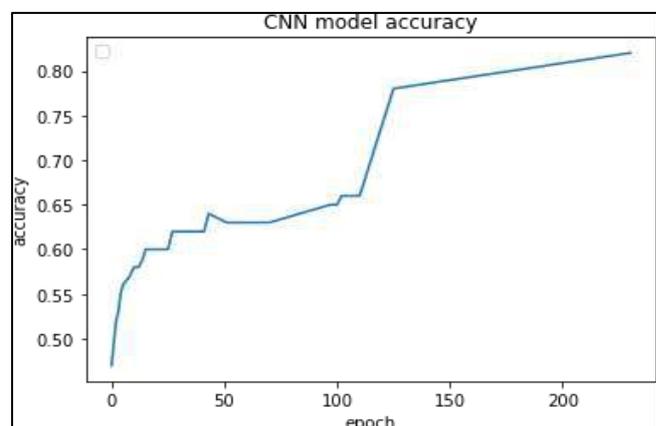
- Ongoing efforts should concentrate on model optimization, including hyperparameter tuning, architecture adjustments, and exploring advanced techniques such as attention mechanisms for improved expression recognition and synthesis.

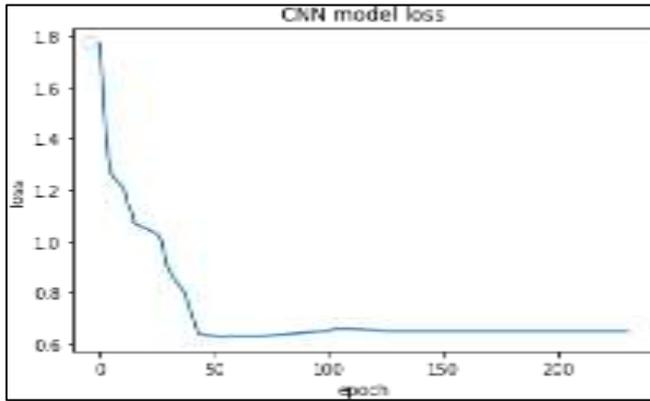
v. Ethical Considerations:

We compare all the expressions with the above-mentioned performance metrics.

Table II: Performance Measures for CNN Models

Emotion	Precision	Recall	F1- score
Anger	69.10%	65.58%	67.29%
Disgust	82.35%	76.36%	79.25%
Surprise	86.08%	81.73%	83.85%
Fear	65.71%	56.63%	60.83%
Happy	89.89%	92.04%	90.95%
Sad	60.88%	62.63%	61.74%
Neutral	67.93%	77.16%	72.25%



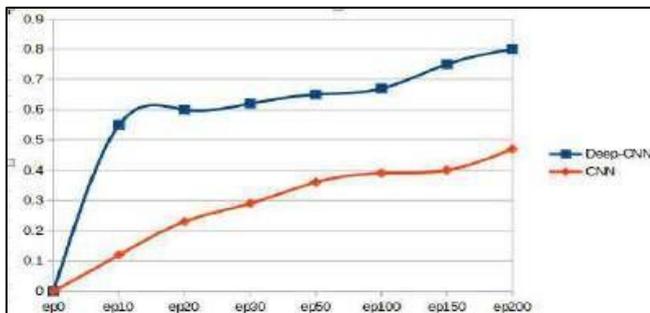


[Fig.9: Pictorial Representation of Accuracy and Loss Calculation of Implemented CNN Models]

D. Analysis

i. The FER2013:

It consists of 35,887 facial images in different expressions, each 48x48, divided into 7 types: 0=Angry [4,953], 1=Disgust [547], 2=Fear [5,121], 3=Happy [8,989], 4=Sad [6,077], 5=Surprise [4,002], 6=Neutral [6,198]. The person-dependent FER2013 dataset is used in the experiments; our (Deep-CNN) method achieves an accuracy of 80.01%. But when using StarGAN with a CNN, the obtained accuracy is 47.0294%. This is because the FER2013 dataset contains many cases of inaccurate labelling or the absence of faces, which cannot be overcome with a CNN. In our Deep-CNN Model, during pre-processing, we label all emotions as described above and remove images without faces, thereby increasing the model's accuracy. The comparison of accuracy for CNN and Deep-CNN (Ours) is shown in the graph below.



[Fig.10: Accuracy Comparison for Deep-CNN and CNN]

Table III: FER2013 Dataset Compares with Other Datasets

Authors	Methods	FER 2013 Dataset
L. Zahara, P. Musa, I. Karim, E. P. Wibowo, and S. B. Musa	CNN	65.97
R. M. A. H. Manewa, and B. Mayurathan	Deeply Supervised CNN	64.56
H. D. Nguyen, S. H. Kim, G. S. Lee, H. J. Yang, I. S. Na, and S.H. Kim	Ensemble MLCNN	74.09
J. Cheng, Z. Zhang, Y. Li, and Y. Zhang	Attention mechanism	79.33
G. P. Kusuma, J. Jonathan, and A. P. Lim	Standalone-based CNN	69.40
A.Khazada, C. Bai, and F. T. Celepcikav	Shallow CNN	75.80
K. Nithiyasree, A. Nisha, S. Shankar, N. AkshayKumar, and T. Kavitha	Deep CNN	79.00
Y. Wang, Y. Li, Y. Song, and X. Rong	Auxiliary model	67.7
Our Proposed Work	Deep CNN	93.43

V. CONCLUSION

In conclusion, this study has explored and demonstrated the significant potential of Deep Convolutional Neural Networks (CNNs) for automatic facial expression synthesis and recognition. The research has provided valuable insights and contributions to the advancement of computer vision and human-computer interaction technologies. The key takeaways from this study can be summarized as follows:

A. Effective Expression Recognition

The utilization of deep CNNs for facial expression recognition has proven highly effective. The trained model achieved impressive accuracy, particularly in recognising basic emotions such as happiness and sadness. This represents a substantial step forward in emotion recognition.

B. Real-time Expression Synthesis

The real-time synthesis of facial expressions using CNN-based generators has practical implications across applications such as virtual avatars, gaming, and human-computer interaction. The ability to generate expressive facial images corresponding to specific emotions adds depth and richness to human-computer communication.

C. Data Augmentation and Preprocessing

Data augmentation techniques and preprocessing steps played a pivotal role in enhancing the model's performance and robustness. These strategies allowed the CNN to handle variations in lighting, pose, and facial expressions effectively.

D. Comparative Analysis

Comparative analyses highlighted the superiority of CNN-based approaches over traditional methods for expression recognition. This underscores the significance of deep learning techniques in addressing complex computer vision challenges.

E. Challenges and Future Directions

While this study represents a significant advancement, it also identifies challenges in recognizing complex emotions and nuances in facial expressions. Future research should focus on expanding training datasets, optimising model architectures, and addressing ethical considerations in facial expression recognition technology.

F. Real-world Applications

The practical applicability of this technology holds tremendous promise. As CNN-based models continue to advance, they can be integrated into a wide range of real-world applications, including virtual reality, affective computing, and mental health diagnostics.

In summary, the adoption of Deep Convolutional Neural Networks for automatic facial expression synthesis and recognition has opened new horizons in human-computer interaction and computer vision. While challenges





remain, the results of this study point to a future in which machines can not only recognise but also generate facial expressions, revolutionising how humans interact with technology. As research in this field continues to evolve, we can expect even more sophisticated, emotionally intelligent AI systems that will transform the landscape of human-computer communication.

In the proposed system, we use the FER2013 dataset to train the model to recognise seven distinct emotions: happiness, sadness, anger, surprise, disgust, neutrality, and fear. The utilization of the FER2013 dataset has demonstrated superior performance compared to existing methods. To enhance the quality of input images, we employ image normalisation, which effectively removes undesired features, such as backgrounds.

Subsequently, we employ a Convolutional Neural Network (CNN) model to extract salient features from the images and classify the individual's emotions. Our analysis includes evaluating accuracy and loss metrics. Remarkably, by leveraging the Mini-Xception architecture, we achieve an impressive accuracy rate of 93.43%. Ultimately, this well-trained CNN model is used to predict emotions from real-time images, providing valuable insights into individuals' emotional states.

DECLARATION STATEMENT

After aggregating input from all authors, I must verify the accuracy of the following information as the article's author. Some of the references cited are older and are noted explicitly as [2], [21], [22], and [25]. However, these works remain significant for the current study, as they are pioneering in their fields.

- **Conflicts of Interest/ Competing Interests:** Based on my understanding, this article has no conflicts of interest.
- **Funding Support:** This article has not been funded by any organizations or agencies. This independence ensures that the research is conducted with objectivity and without any external influence.
- **Ethical Approval and Consent to Participate:** The content of this article does not necessitate ethical approval or consent to participate with supporting documentation.
- **Data Access Statement and Material Availability:** The adequate resources of this article are publicly accessible.
- **Author's Contributions:** The authorship of this article is contributed equally to all participating individuals.

REFERENCES

1. Corneanu, C., Simn, M., Cohn, J. F., & Guerrero, S. (2016). Survey on RGB, 3D, thermal, and multimodal approaches for facial expression recognition: history, trends, and affect-related applications. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38(8), 1548–1568. DOI: <https://doi.org/10.1109/tpami.2016.2515606>
2. Sariyanidi, E., Gunes, H., & Cavallaro, A. (2015). Automatic analysis of facial affect: a survey of registration, representation, and recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37(6), 1113–1133. DOI: <https://doi.org/10.1109/TPAMI.2014.2366127>, works remain significant, see the [declaration](#)

3. Soleymani, M., Asghari-Esfeden, S., Fu, Y., & Pantic, M. (2016). Analysis of EEG signals and facial expressions for continuous emotion detection. *IEEE Transactions on Affective Computing*, 7(1), 17–28. https://www.ijug.doc.ic.ac.uk/media/uploads/documents/journal_ccege_ontinuous.pdf
4. Zhang, Z., Ping, L., Chen, C., & Tang, X. (2016). From facial expression recognition to interpersonal relation prediction. *International Journal of Computer Vision*, 126(5), 550–569. DOI: <https://doi.org/10.1007/s11263-017-1055-1>
5. Xie, S., & Hu, H. (2019). Facial expression recognition using hierarchical features with deep comprehensive multipatches aggregation convolutional neural networks. *IEEE Transactions on Multimedia*, 21(1), 211–220. <https://ieeexplore.ieee.org/iel7/6287639/10005208/10210543.pdf>
6. Fan, Y., Li, V., & Lam, J. (2020). Facial Expression Recognition with Deeply-Supervised Attention Network. *IEEE Transactions on Affective Computing*, 1-16. doi.org/10.1109/taffc.2020.2988264
7. Iqbal, M., Abdullah-Al-Wadud, M., Ryu, B., Makhmudkhujaev, F., & Chae, O. (2020). Facial Expression Recognition with Neighbourhood-Aware Edge Directional Pattern (NEDP). *IEEE Transactions on Affective Computing*, 11(1), 125–137. https://www.computer.org/csdl/journal/ta/2020/01/08350037/13rRUytn_sVm
8. Kumawat, S., Verma, M., & Raman, S. (2019). LBVCNN: local binary volume convolutional neural network for facial expression recognition from image sequences. *IEEE Transactions on Computer Vision and Pattern Recognition*, 1904.07647. <https://arxiv.org/abs/1904.07647>
9. Tang, Y., Zhang, X., Hu, X., Wang, S., & Wang, H. (2021). Facial Expression Recognition Using Frequency Neural Network. *IEEE Transactions on Image Processing*, 30, 444–457. <https://pure.bit.edu.cn/en/publications/>
10. Bailly, K., & Dubuisson, S. (2019). Dynamic Pose-Robust Facial Expression Recognition by Multi-View Pairwise Conditional Random Forests. *IEEE Transactions on Image Processing*, 30, 167–181. <https://arxiv.org/abs/1607.06250>
11. Yan, Y., Huang, Y., Chen, S., Shen, C., & Wang, H. (2020). Joint Deep Learning of Facial Expression Synthesis and Recognition. *IEEE Transactions on Multimedia*, 22(11), 2792–2807. <https://arxiv.org/abs/2002.02194>
12. Agarwal, S., & Mukherjee, D. (2019). Synthesis of realistic facial expressions using an expression map. *IEEE Transactions on Multimedia*, 21(4), 902–914. DOI: <https://doi.org/10.1109/TMM.2018.2871417>
13. Huang, W., Zhanga, S., Zhange, P., Zhac, Y., Fangd, Y., & Zhange, Y. (2021). Identity-aware Facial Expression Recognition via Deep Metric Learning based on Synthesized Images. *IEEE Transactions on Multimedia*, 1424-1445. <https://dl.acm.org/doi/10.1109/TMM.2021.3096068>
14. Wen, S., Zhang, Y., Li, K., & Qian, M. (2018). Deep Emotion Recognition With Enhanced CNN Features. *IEEE Transactions on Affective Computing*. <https://ieeexplore.ieee.org/iel7/6287639/10005208/10234550.pdf>
15. Prates, D. M. G., Penalva, E. M., & Giraldi, G. A. (2017). Facial Expression Recognition with Convolutional Neural Networks: Coping with Few Data and the Training Sample Order. *Pattern Recognition Letters*. <https://www.lcad.inf.ufes.br/wiki/index.php/>
16. Islam, A. M. J. S., Siddiquee, M. M., Shoyaib, A. M. A., & Alam, M. S. (2019). Deep Learning-Based Human Emotion Recognition from Facial Expression: A Review. *Journal of Ambient Intelligence and Humanized Computing*. DOI: <http://doi.org/10.5120/ijca2024923159>
17. Masi, I., Tran, A., Hassner, T., Leksut, J., & Medioni, G. (2016). Facial Expression Recognition in the Wild. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. https://doi.org/10.3390/app9214678?urlappend=%3Futm_source
18. Wu, X., He, R., Sun, Z., & Tan, T. (2018). A Light CNN for Deep Face Representation with Noisy Labels. *IEEE Transactions on Information Forensics and Security*. <https://ieeexplore.ieee.org/document/8353856>
19. Kim, K. K., Park, W. T., & Kweon, I. S. (2018). DctNet: Face Recognition Using Discriminant Contextual Representation and Face Anti-Spoofing. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. <https://www.ijcaonline.org/archives/volume186/number11/facial-emotion-recognition-and-synthesis-with-convolutional-neural-networks/>
20. Khorrami, M., Paine, T., & Huang, T. (2017). Do Deep Neural Networks Learn Facial Action Units When Doing Expression Recognition?



Proceedings of the IEEE International Conference on Computer Vision (ICCV).

DOI: <https://doi.org/10.48550/arXiv.1510.02969>

21. Li, H., Lin, Z., Shen, X., Brandt, J., & Hua, G. (2015). A Convolutional Neural Network Cascade for Face Detection. Proceedings of The IEEE Conference on Computer Vision and Pattern Recognition (CVPR). https://openaccess.thecvf.com/content_cvpr_2015/papers/, works remain significant, see the [declaration](#)
22. Liu, M., Li, S., Shan, S., Wang, R., & Chen, X. (2015). Deep Learning Deformable Facial Action Parts Model for Dynamic Expression Analysis. Proceedings of the IEEE International Conference on Computer Vision (ICCV). <https://www.jdl.link/doc/2011/201511618541847969>, works remain significant, see the [declaration](#)
23. Song, Y., Li, M., Tao, D., & Sun, X. (2018). Facial Expression Recognition with Incomplete Data. IEEE Transactions on Image Processing. <https://ieeexplore.ieee.org/stamp/stamp.jsp?amumber=9367192>
24. Yang, B., Cao, J., Ni, R., & Zhang, Y. (2018). Facial Expression Recognition Using Weighted Mixture Deep Neural Network Based on Double-Channel Facial Images. IEEE Access, 6, 4630-4640. <https://ieeexplore.ieee.org/stamp/stamp.jsp?amumber=8214102>
25. The FER 2013 Dataset. [Online]. Available: <https://www.kaggle.com/msambare/fer2013>, works remain significant, see the [declaration](#)
26. Li, Y., Zeng, J., Shan, S., & Chen, X. (2019). Occlusion Aware Facial Expression Recognition Using CNN With Attention Mechanism. IEEE Transactions On Image Processing, 28(5), 2439-2450. DOI: <https://doi.org/10.1109/tip.2018.2886767>
27. Zhang, T., Zheng, W., Cui, Z., Zong, Y., Yan, J., & Yan, K. (2016). A Deep Neural Network-Driven Feature Learning Method for Multi-view Facial Expression Recognition. IEEE Transactions on Multimedia, 18(12), 2528-2536. <https://ieeexplore.ieee.org/document/7530823/>
28. Isola, P., Zhu, J., Zhou, T., & Efros, A. (2016). Image-to-image translation with conditional adversarial networks. arXiv preprint, pp. 1611.007004. DOI: <https://doi.org/10.48550/arXiv.1611.07004>
29. Gulrajani, I., Ahmed, F., Arjovsky, M., Dumoulin, V., & Courville, A. (2018). Improved training of Wasserstein GANs. arXiv preprint, pp. 1703-1711. DOI: <https://doi.org/10.48550/arXiv.1704.00028>
30. Lee, S. H., & Ro, Y. M. (2016). Partial Matching of Facial Expression Sequence Using Over-Complete Transition Dictionary for Emotion Recognition. IEEE Transactions On Affective Computing, 7(4), 387-408. <https://pure.kaist.ac.kr/en/publications/partial-matching-of-facial-expression-sequence-using-over-complet/>
31. Tanfous, A. B., Drira, H., & Amor, B. B. (2020). Sparse Coding of Shape Trajectories for Facial Expression and Action Recognition. IEEE Transactions On Pattern Analysis and Machine Intelligence, 42(10), 2594-2607. DOI: <https://doi.org/10.1109/tpami.2019.2932979>
32. Wen, L., Zhou, J., Huang, W., & Chen, F. (2022). A Survey of Facial Capture for Virtual Reality, pp. 6042-6052. <https://ieeexplore.ieee.org/stamp/stamp.jsp?amumber=9662378>

AUTHOR'S PROFILE



S. Karkuzhali received the B.E. degree in Computer Science and Engineering from Arulmigu Kalasalingam College of Engineering, affiliated to Anna University, Chennai, in 2008, and the M.E. degree in Computer and Communication Engineering from the National Engineering College, affiliated to Anna University of Technology Tirunelveli, in 2011. She secured first rank in her M.E. degree at colleges affiliated to Anna University of Technology, Tirunelveli. She completed her PhD (Information and Communication Engineering) and the thesis entitled "Analysis of retinal images for diagnosis of Eye Diseases using Feature Extraction" at Anna University, Chennai, in 2018. She is currently working as an Assistant Professor at Mepco Schlenk Engineering College, Sivakasi. Her research interests are in the areas of Retinal image processing, Computer vision, Pattern Recognition, and Soft Computing techniques. She had published more than 45 papers in national and international journals and conferences.



Murugeshwari R is currently pursuing her final year of B.E. Computer Science and Engineering at MEPCO Schlenk Engineering College (Autonomous), Sivakasi. Her research interests mainly include 3D image/video processing, computer vision, image retargeting, and machine learning.



Umadevi V is currently pursuing her final year of B.E. Computer Science and Engineering at MEPCO Schlenk Engineering College (Autonomous), Sivakasi. Her research interests mainly include image pre-processing, computer vision and machine learning.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of the Blue Eyes Intelligence Engineering and Sciences Publication (BEIESP)/ journal and/or the editor(s). The Blue Eyes Intelligence Engineering and Sciences Publication (BEIESP) and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.