


Binomial Dropout Convolutional Neural Network Classifier-Based Micro-expression Recognition System

Anjani S. D. D ^{1,*}, M. K. Kumar², Chinnam Sabitha³, P. V. V. S. R. Kumari⁴, and Sasi R. Desabathula¹

¹ Sasi Institute of Technology & Engineering, Tadepalligudem, Andhra Pradesh, India

² Department of ECE, Aditya University, Surampalem, Andhra Pradesh, India

³ Koneru Lakshmaiah Education Foundation (KLEF), Vaddeswaram, Andhra Pradesh, India

⁴ Ideal Institute of Technology, Kakinada, Andhra Pradesh, India

Email: anjanihasini@gmail.com (A.S.D.D.); mkishorekumar.hasini@gmail.com (M.K.K.); sabithakiran.ch@gmail.com (C.S.); ramkumari.2001@gmail.com (P.V.V.S.R.K.); rekhasasi08@gmail.com (S.R.D.)

*Corresponding author

Abstract—Facial expression recognition is essential for a variety of applications, including marketing, commerce, security systems, and psychotherapy. The two primary categories of facial expressions are macro-expression and micro-expression. When comparing macro-expression, it can be difficult to recognize micro-expression in real time because it typically happens in high stakes scenarios. So, in this work, auto-detection of facial micro-expressions was proposed by using a Binomial Dropout Convolution Neural Network (BDCNN) classifier technique. Initially, the input video collected from the publicly available source is converted into frames and the keyframes are generated by using the Distance Deviation-based Rood Pattern Search (DDRPS) algorithm. Then, the keyframes are preprocessed to reduce the significance of illumination, face detection, landmark points on the face, and motion magnification. Afterward, the important features are extracted from the preprocessed frames. Finally, the BDCNN classifier was used to recognize the micro-expression. The average accuracy and F-Measure values obtained on the Spontaneous Actions and Micro-Movements (SAMM) dataset are 0.9758 and 0.9247; also, for the Spontaneous Micro-expression Corpus (SMIC) dataset, the values are 0.8989 and 0.8536, similarly, for the Chinese Academy of Sciences Micro-Expression (CASME) II dataset, the accuracy and F-Measures are 0.9784 and 0.9509. The proposed classifier is compared to state-of-art methods to prove its superiority.

Keywords—micro expression, Distance Deviation based Rood Pattern Search (DDRPS) algorithm, Edge Preserving Homomorphic Filter (EPHF), YOLOv3, Linear Active Appearance Model (LAAM), Linear based Eulerian Video Magnification (LEVM), Binomial Dropout Convolutional Neural Network (BDCNN)

I. INTRODUCTION

Emotions are considered the most important part of human life, and they directly affect physical and cerebral activities. Facial Expressions (FE) are a form of non-

verbal communication that conveys an individual's emotional state to the observer. FE's are divided into '2' categories: macro-expressions and Micro-Expressions (ME) [1]. Macro-expressions are the basic emotions that can be recognized by the naked eye [2] and typically last between 0.5 and 4 Seconds [3]. The six main categories of macro-expressions are anger, fear, disgust, sadness, happiness, and surprise [4]. In contrast, macro-expressions are quick, low-intensity facial expressions that typically arise when people intentionally or unintentionally try to disguise their actual feelings [5]. ME typically only affects a few areas of the face and lasts between 1/5 and 1/25 of a second [6]. The main differences between macro-expressions and micro-expressions lie in their intensity and duration [7]. In essence, Micro-Expressions (ME) often reveal the true emotions a person may try to suppress or hide [8]. Micro-Expression (ME) recognition has many practical real-world applications, including clinical diagnosis, education, and security [9].

Micro-expression recognition can be broadly divided into two basic categories. One focuses on obtaining data from a facial video clip by extracting the micro-expression features. The second method is micro-expression classification, which builds a classifier similar to a support vector, decision tree, and k-nearest neighbor [10]. Facial features are retrieved for micro-expressions by using methods like Local Binary Pattern (LBP), Histogram of Oriented Gradients (HOG), and Histogram of Image Gradient Orientation (HIGO) [11]. Recognition of micro-expression involves a few steps. They are as follows:

- (1) The input image is taken from the video.
- (2) Applying pre-processing operations for optimal display.
- (3) Facial landmark detection.
- (4) Applying cropping and resizing of the image.
- (5) Selection of the frame [12].

Manuscript received April 24, 2025; revised June 4, 2025; accepted July 10, 2025; published September 17, 2025.

The available micro-expression datasets are small and collected into two groups: stimulated and spontaneous. The stimulated micro-expression datasets include University of South Florida-High Definition (USF-HD) and Polikovsky, while the spontaneous micro-expression datasets include Spontaneous Micro-expression Corpus (SMIC), Chinese Academy of Sciences Micro-Expression (CASME), CASME II, and Spontaneous Actions and Micro-Movements (SAMM). Spontaneous datasets are used more frequently than the stimulated datasets because they are very close to real conditions [13].

In the early stage, artificial recognition tools like Facial Action Coding System (FACS) [14] and Micro Expression Training Tool (METT) are used to identify facial expressions; due to their expensive cost and low accuracy they have become far away from the practical field. To overcome this, deep learning techniques are employed in Micro-expression to estimate the optimal flow in the facial image [15], and video sequences [16] from three different categories like onset, offset and apex are taken [17]. Though deep learning techniques have a lot of advantages in ME, it has limitations like insufficiency of training data, spotting, and recognition of naturally occurring challenges like the blinking of an eye, and movement of hair will also affect ME identification [18]. Another drawback like the choice of feature in ME is also a difficult task because of the broad classification of ME in feature recognition [19]. In order to overcome these drawbacks, an approach is proposed to recognize the micro-expressions accurately using the Binomial Dropout Convolutional Neural Network (BDCNN) classifier.

II. OBJECTIVES

To use micro-expressions as reliable predictors of true emotional states that would allow for the detection of repressed or hidden feelings for lie detection, diagnostic analysis, and psychological evaluation. The Binomial Dropout Convolutional Neural Network (BDCNN) classifier was used to detect facial micro-expressions which can generalize better and mitigate overfitting. We implemented a custom deep learning model (BDCNN) to detect small, rapid facial expressions that signify emotional expression.

A. Contributions

The following are the main contributions of this work.

A Distance Deviation-based Rood Pattern Search (DDRPS) algorithm is proposed for the motion estimation in the frames.

- An Edge Preserving Homomorphic Filter (EPHF) technique is proposed for the removal of uneven illumination, blurriness, and occlusion problems in video frames.
- A method is developed for the face alignment process, which supports a strong detection system.
- To decrease the computational complexity and to enhance the classification accuracy, a BDCNN classifier is used in this work.

The work is structured as follows: Section II contains a literature review of some of the existing techniques, Section III deals with the proposed technique; Section IV describes the result and discussion of the proposed methodology and Section V ends with the conclusion of the proposed framework.

B. Literature Survey

Li *et al.* [20] developed a multi-scale Joint Feature Module (JFM) network for identifying micro-expressions based on optical flow images to increase accuracy. In this implementation process, firstly, the apex frame and onset-frame were used to generate an optimal flow image that reflected the subtle facial motion information, and it was fed to a multi-scale JFM combined backbone network for extraction of the features. Then finally, the output was combined by using a fusion network that predicted the expression according to its strategy. In addition to this, the joint feature module combined features from various layers are used to identify the micro-expression features with various amplitudes. The developed system attained high performance in terms of Unweighted F1-Score (UF1) and Unweighted Average Recall (UAR). Even though it had obtained good results, it had a limitation in the calculation of optical flow image, which takes place in the onset and apex frame.

Wang *et al.* [21] recommended addressing the issue of lacking discriminative spatiotemporal characteristics; it was suggested using a Dual-stream Spatio-Temporal Attention Network (DSTAN). The Spatio-Temporal Appearance Network (STAN) and the spatiotemporal motion network served as the foundation for the spatiotemporal networks in DSTAN (STMN). To extract the spatial information, a multi-scale kernel spatial attention and a global dual-pool channel attention were introduced. The temporal models of STAN and STMN were then updated with the retrieved characteristics. Finally, feature concatenated-Support Vector Machine (SVM) was used to merge the STAN-A and STMN-A attention. The constructed DSTAN networks were used to increase the accuracy of the output. The method's main flaw was that it used SVM, which underperformed when the amount of training data samples exceeded the number of data points in the micro-expression features.

Banerjee *et al.* [22] recommended a mixed feature learning approach for Micro Expression Recognition (MER) to acquire the best accurate categorization. The five main steps of this work are as follows:

1. Pre-processing.
2. Detection of the face.
3. Facial landmark extraction.
4. Feature extraction.
5. Micro Expression (ME) spotting or recognition.

Initially, the video was taken as input and then converted as frames. Next, for pre-processing operations, Grey-Scale (GS) was implemented. After pre-processing, the Viola-Jones Algorithm (VJA) was used to detect the faces of the frames. Weight matrix-based Temporal Accumulated Optical Flow and Improved LBP features (WTAOF-ILPB) were also introduced for extraction. The Levy Flight-based Shuffled Shepherd Optimization

Algorithm was then given the facial picture features that had been retrieved from them. Finally, a Radial Basis Function Neural Network (RBFNN) was employed for the classification of MER. Despite its many benefits, it is extremely challenging to distinguish due to the small variations in micro-expressions and the incomplete and unbalanced dataset.

Li *et al.* [23] recommended a video-based micro-expression recognition. It was developed by using the 3D flow-based Convolutional Neural Network model. In this method, they designed a '12'-layered architecture, which consists of inputs from 3 grayscale frame sequences. In addition to this, a 3D convolution kernel was introduced to extract spatiotemporal feature data. A Dropout fine-tuning regularization technique was added to reduce the overfitting problem and complex adaptation on training data in neural networks. However, obtaining adequate recognition results was extremely challenging because of the scarcity of training data.

Verma *et al.* [24] invented a Lateral Accretive Hybrid Network (LEARNet) to record the minute details of a face expression. In this implemented technique, firstly, generated dynamic images from a micro-expression sequence were captured in one frame. The LEARNet architecture was also used to handle dynamic pictures for training and inference. The prominent characteristics from the expressive areas that were recorded in the previous layers were learned using a hybrid and decoupled learning feature technique. An acceleration layer was included to merge the response of two facial expressions. From experimental results, it was evidenced that the LEARNet had achieved better accuracy rates. The drawback of the system was the usage of different types of layers for dynamic imaging increased the complexity of the hybrid network.

Gan *et al.* [25] came up with Optical Flow Features from Apex frame Network (OFF-Apex Net) were developed to identify face micro-expressions. It was obvious from the name that it blended handcrafted features (such as components derived from optical flow) and a completely data-driven design (i.e., convolutional neural network). In this method, the onset and apex frames were used to compute the horizontal and vertical optical flow features. After that, a neural network was fed with the characteristics to emphasize the important expression data. The output result obtained proved that it had good performance in the recognition of f-measure and accuracy. Though it had a lot of benefits, the usage of Convolution Neural Network (CNN) created a lot of problems like over-fitting, exploding gradient, and class imbalance.

Wang *et al.* [26] to overcome the limited detection accuracy of emotional facial-behavior, a residual network and a novel attention mechanism dubbed micro-attention were applied. Analyzing and pre-processing the database description was done in this. The network was then given the ability to concentrate on the face regions exhibiting micro-expression by integrating novel micro-attention units into each residual block. To reduce the over-fitting issue, a transfer learning strategy was used to train the

network. To demonstrate the potency of the implemented attention units in capturing the underlying face emotion, a high-level feature visualized map was used. The major negative of this technique was that the usage of transfer learning was the problem of the negative transfer of data.

Li *et al.* [27] proposed a deep local holistic network for micro-expression recognition. For extracting rich and local information from Region of Interests (ROIs) associated with micro-expression, Hybrid Convolutional Recurrent Neural Network (HCRNN) is used. Here used Relation-Preserving Recurrent Neural Network (RPRNN) to extract holistic and sparse features from sparse images, and also uses RPCA to extract sparse micro-expression information from original images based on the sparse characteristic of micro-expression. The performance of MER is improved by the Deep Local-Holistic Network, which is fused by HCRNN and RPRNN to capture local holistic, sparse abundant micro-expression information, and this work obtained 60.31% recognition accuracy.

Pan *et al.* [28] to recognize micro-expressions, provide a multi-scale fusion visual attention network model that combines the local attention weights of the multiple-scale feature maps of the eliminating identification attributes network. By understanding the mapping relationship from the apex to the onset frame in micro-expression video sequences, a mix of un-supervised and transfer learning is utilized to address the issue of the tiny ME dataset and lessen the influence of identity traits. After that, by concentrating on multi-scale local attention weights, the local detail characteristics are extracted. Lastly, by combining high weight local and global criteria, micro-expressions are categorized and influence individual qualities on local ROI localization. This work obtained fusing accuracy is 79.45 and F1-Score is 0.7816 respectively.

Tang *et al.* [29] suggested a facial micro-expression recognition technique, based on a CNN-transformer hybrid model. Use a hybrid model of CNN and transformer to extract facial hierarchical features as inputs into a deep model. The area of the facial micro-expression image is segmented in parallel, and then the image is smoothed through thresholding to extract the feature vectors of the facial micro-expressions. These feature vectors are then fed into a CNN-transformer hybrid model to achieve recognition of the facial micro-expressions. They achieved higher mean recognition accuracy-up to 98%.

Paul *et al.* [30] shows a modern frontier of robotic capsicum harvesting while also providing YOLOv8s and YOLOv8s-seg models with detection capabilities, peduncle segmentation, and growth stage classification. The models had great mean Average Precision (mAP) scores and since localization was done using the RealSense D455 camera, it produced 94.1% bloom counting accuracy; additionally, the Android application provided a robust performance and ties into smart agricultural automation.

Facial micro-expression detection has become an important area of study in affective computing because it captures real, involuntary emotional expression. Prior

work has shown the performance of deep-learning models, particularly Convolutional Neural Networks (CNNs), in detecting small facial movements. However, there are still known limitations we face when using CNNs for facial expressions such as the relatively small datasets of images, high intra-class similarity, short duration of expressions, and issues with overfitting to training. While there have been recent examples of methods that implement binomial dropout and a custom architecture (BDCNN) to improve generalization and accuracy, more robust models are necessary that have been tested in real-life settings. These limitations serve as the basis from which we develop optimized architectures to capture the

variability inherent in micro-expressions while improving recognition performance within practical applications.

III. PROPOSED MICRO-EXPRESSION RECOGNITION SYSTEM

In this work, automatic recognition of micro-expressions was proposed by using a Binomial Dropout Convolutional Neural Network (BDCNN) classifier technique. The proposed method undergoes the following steps such as the conversion of video into frames, motion estimation, preprocessing, feature extraction, and classification. Fig. 1 shows the suggested technique's block diagram.

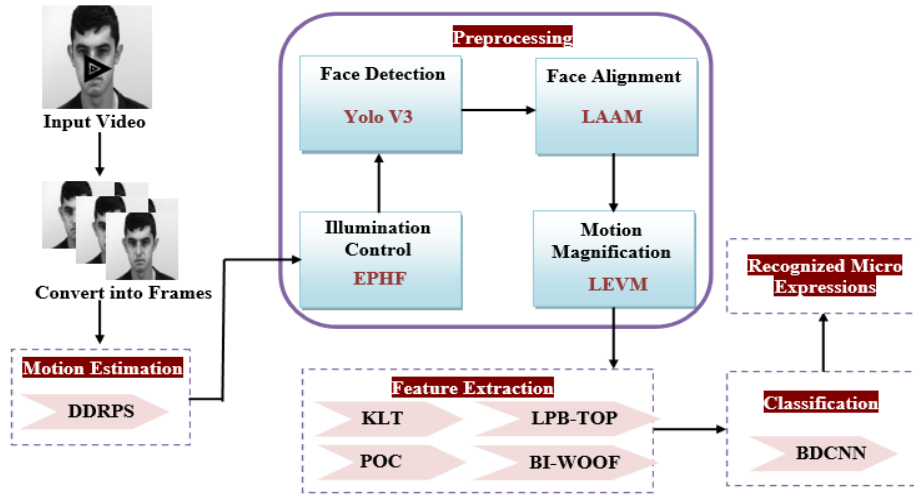


Fig. 1. Block diagram for the suggested technique.

A. Input Video

In this work, the input videos used for the micro-expression recognition system are collected from the publicly available source. Before the recognition process, the input video is first converted into frames for further processing. Let the input video be denoted as (I_{vd}) , which is converted into frames $f(I_{vd})$ is given by in Eq. (1).

$$f(I_{vd}) = [f_1(I_{vd}), f_2(I_{vd}), f_3(I_{vd}), \dots, f_n(I_{vd})] \quad (1)$$

where, $f_n(I_{vd})$ represents the n^{th} number of frames.

B. Motion Estimation

Motion estimate is the procedure of determining the reference frame's best match for the current frame's macro-block. In this work, after the conversion of video into frames $f_n(I_{vd})$, motion estimation was done by using the Distance Deviation-based Rood Pattern Search (DDRPS) algorithm to generate keyframes. DDRPS primarily addresses the extraction of fine-scale (or micro) motion patterns to best identify micro-expressions. BDCNN allows for strong feature learning using dynamic regularization. Taken together, they yield a hybrid framework that employs both extracted motion features and deep learning capabilities for pattern recognition. The Rood Pattern Search (RPS) algorithm is made up of symmetrical search patterns known as rood shape or cross

shape. The Adaptive Rood Pattern (ARP) and the Unity Rood Pattern (URP) are two distinct stages of search that differ primarily in the separation of the search points. The ARP reduces the influence of the neighbor macroblock that has less correlation with the current macroblock in the process of motion vector prediction. To avoid unnecessary computation, Distance Deviation (DD) is calculated instead of the mean absolute difference. Hence, the proposed method is named DDRPS. Distance Deviation (DD) can be used to enhance sensitivity; especially relevant when looking for subtle motion like micro-expressions.

At first, the rood pattern with adjustable arm length has been adopted for each macroblock using its predicted motion behavior. Using the motion vectors of the neighboring blocks, the present block's motion vector is predicted. For MV prediction, a set of four neighboring blocks was used. The chosen set of neighboring blocks is referred to as the Region of Interest (ROI). The size of the rood pattern (χ) can be determined by Eq. (2).

$$\chi = \max(|mv_{pred}(hc)|, |mv_{pred}(vc)|) \quad (2)$$

where, $mv_{pred}(hc)$ and $mv_{pred}(vc)$ represents the horizontal and vertical components of the predicted motion vector respectively.

A sixth point is added to the four-armed root patterns, which is the same as the predicted MV. Then, the Distance Deviation (DD) is computed for all points, which is given in Eq. (3).

$$DD = \sqrt{\frac{\sum \left(f(I_{vd})_{(\pi, \psi)} - \overline{f(I_{vd})_{(\pi, \psi)}} \right)^2}{n}} \quad (3)$$

where, $\overline{f(I_{vd})}$ represents the mean of input frames, (π, ψ) represents the position of the block. The minimum DD point from the prior phase is selected as the centre of the URP, and all of its points are examined.

The process is repeated until the keyframes are generated. The keyframes $K_f(a, b)$ are expressed as in Eq. (4).

$$K_f(a, b) = [K_{f1}, K_{f2}, K_{f3}, \dots, K_{fQ}] \quad (4)$$

where, K_{fQ} represents the Q^{th} number of keyframes.

C. Preprocessing

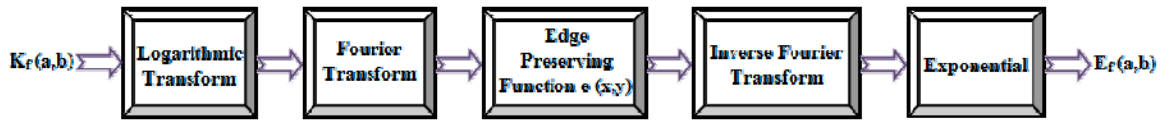


Fig. 2. Architecture of Edge Preserving Homomorphic Filter (EPHF).

According to this method, a key frame $K_f(a, b)$ can be expressed as the multiplication of the illumination component $I(a, b)$ and reflectance component $R(a, b)$, which is expressed as in Eq. (5).

$$K_f(a, b) = I(a, b)R(a, b) \quad (5)$$

For HF to be effective, linear separations for two components are required. In order to separate these two components, natural logarithmic transform is used on both sides of Eq. (5) which is given in Eq. (6).

$$\ln K_f(a, b) = \ln I(a, b) + \ln R(a, b) \quad (6)$$

After the logarithmic transform, the image needs to transform the spatial domain into the frequency domain, which helps to change the brightness by separating high-frequency components from low-frequency components. This can be done by using the Fourier transform, which is expressed as in Eqs. (7) and (8).

$$F \{ \ln K_f(a, b) \} = F \{ \ln I(a, b) \} + F \{ \ln R(a, b) \} \quad (7)$$

$$K_f(x, y) = F_I(x, y) + F_R(x, y) \quad (8)$$

where, $F_I(x, y)$ represents the Fourier transform of $\ln I(a, b)$, $F_R(x, y)$ and $\ln R(a, b)$. $K_f(x, y)$ represents

Preprocessing is the initial step taken to enhance the frames before they are used by the classifier. In this work, the quality of frames has been improved with the steps such as illumination control, face detection, face alignment, and motion magnification.

1) Illumination control

At first, the uneven illumination, light intensity blur, and occlusion problem of the key frames $K_f(a, b)$ are controlled by using Edge Preserving Homomorphic Filter (EPHF). A technique based on an illumination-reflectance image model is called a homomorphic filter. The illumination component belongs to low-frequency components while the reflectance component belongs to high-frequency components. The objective of the Homomorphic Filter (HF) is to reduce the significance of illumination by reducing the low-frequency components of an image. The normal HF introduces other illumination artifacts on the edges of the foreground. So, the edge-preserving function is multiplied with the frequency domain to preserve the edges of the images. Hence, the proposed method is named EPHF. The architecture of EPHF is shown in Fig. 2.

the frequency domain frame.

Then, the frames are multiplied with the edge-preserving function to preserve the edge of an image, which corresponds to convolution operation in the spatial domain. The filtered frame in the frequency domain is given in Eqs. (9) and (10).

$$B_f(x, y) = K_f(x, y)e(x, y) \quad (9)$$

$$B_f(x, y) = e(x, y)F_I(x, y) + e(x, y)F_R(x, y) \quad (10)$$

where, $B_f(x, y)$ represents the filtered image in the frequency domain and $e(x, y)$ represents the edge-preserving function, which is used to sharpen the edge of the frame that is expressed as in Eq. (11).

$$e(x, y) = e^{\left(\frac{(x-y)^2}{\alpha} \right)} \quad (11)$$

where, α represents the scale parameter.

Next, the filtered image in the spatial domain is obtained by taking the inverse Fourier transform. Thereby, the Eq. (10) can be written as in Eqs. (12) and (13).

$$B_f(a, b) = F^{-1} \{ B_f(x, y) \} \quad (12)$$

$$B_f(a,b) = F^{-1} \{e(x,y)F_l(x,y) + e(x,y)F_r(x,y)\} \quad (13)$$

Finally, the desired enhanced frame $E_f(a,b)$ is obtained by taking the exponential of $B_f(a,b)$, which is given in Eqs. (14) and (15).

$$E_f(a,b) = \exp \{B_f(a,b)\} \quad (14)$$

$$E_f(a,b) = \hat{I}(a,b) \hat{R}(a,b) \quad (15)$$

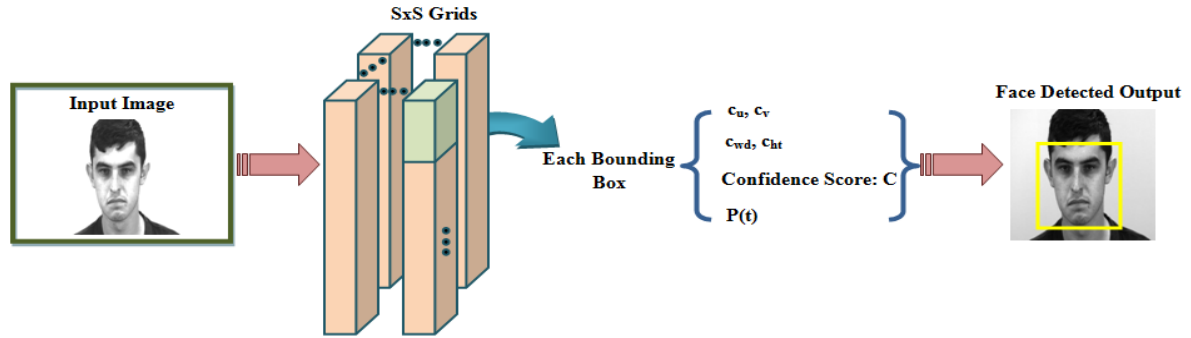


Fig. 3. Architecture of YOLOv3.

Initially, the frame $E_f(a,b)$ is divided into a grid. If the face falls in one of the grids in the frame, the grid is responsible for predicting the face. Each grid will forecast three bounding boxes, each of which contains five parameters and the likelihood that each category will occur. The following Eqs. (16)–(19) provides the bounding box coordinates prediction method.

$$c_u = \gamma(p_u) + g_u \quad (16)$$

$$c_v = \gamma(p_v) + g_v \quad (17)$$

$$c_{wd} = s_{wd} e^{p_{wd}} \quad (18)$$

$$c_{ht} = s_{ht} e^{p_{ht}} \quad (19)$$

where, γ represents the sigmoid activation function, c_u , c_v , c_{wd} , and c_{ht} are the center coordinates and dimension of the bounding box obtained from prediction, p_u , p_v , p_{wd} and p_{ht} are the prediction output of the model, g_u and g_v are the grid cell coordinates and s_{wd} , s_{ht} are the size of bounding box before prediction.

Next, using the Intersection Over Union (IOU) function, the required box is selected. IOU is the proportion of the intersection and union of the ground truth box and the bounding box, which is given in Eq. (20).

$$IOU = \frac{B_{gt} \cap B_{bb}}{B_{gt} \cup B_{bb}} \quad (20)$$

where, $\hat{I}(a,b)$ and $\hat{R}(a,b)$ are the illumination and reflectance components of the output image.

2) Face detection

Next, using YOLOv3, the face in the improved frame was found. A convolutional neural network called YOLOv3 is made up of two fully connected layers, four maximum pooling layers, and twenty-four convolution layers. To determine the bounding box and categorize the target inside the box, it makes use of all the image's features. Fig. 3 depicts the YOLOv3 architecture.

where, B_{gt} represents the ground truth box and B_{bb} represents the bounding box.

The confidence score C indicates the likelihood that the target defect is present in the bounding box as well as instances in which the bounding box and the ground truth coincide, which is given in Eqs. (21) and (22).

$$C = P(t) * IOU(gt) \quad (21)$$

The output $Y_{(\beta, \theta)}$ of YOLOv3 is given by:

$$Y_{(\beta, \theta)} = \begin{cases} f_{(det)} & P(t) = 1 \\ f_{(ndet)} & P(t) = 0 \end{cases} \quad (22)$$

where, $P(t)$ indicates if a target is present in the current grid's bounding box, $f_{(det)}$ and $f_{(ndet)}$ represent the detection and not detection of face. The method's loss function is divided into four sections, including the classification error, the confidence error, the bounding box's centre error, and its width and height errors. Sum variance is used to calculate the bounding box's centre error, width error, and height error, which is given in Eq. (23).

$$loss = \sum_{\beta=1}^m \sum_{\theta=1}^m (Y_{(\beta, \theta)} - \hat{Y}_{(\beta, \theta)})^2 \quad (23)$$

where, $\hat{Y}_{(\beta, \theta)}$ represents the estimated frames, (β, θ) depicts the height and width of the bounding box, and m represents the number of frames.

3) Face alignment

After the detection of the face in the frames $f_{(\text{det})}$, the landmark points on the face were detected by using the Linear Active Appearance Model (LAAM). Active Appearance Models (AAM) are linear models of the shape and the texture of an object. While the texture refers to the amount of pixels, which is often expressed by intensities or colors, the shape is a vector produced by concatenating the position elements of the designated landmarks. Normal AAM has the limitation of less number of samples per class. So, Linear Discriminant Analysis (LDA) is used for alignment. LDA is used to find a proper way to represent the face vector space of the original dataset.

At first, the shape of the detected face $f_{(\text{det})}$ is modeled based on the labeled landmark. Let the shape S as a vector containing ϕ landmarks coordinates are given in Eq. (24).

$$S_{(f_{(\text{det})})} = [q_1, q_2, q_3, \dots, q_\phi] \quad (24)$$

Then, LDA is used to normalize the shapes and project them onto the created shape subspace, which is given in Eq. (25).

$$S = \mu(S) + L_{DA}(S) \bullet S_p \quad (25)$$

where, $\mu(S)$ denotes the mean of shape, S_p represents the shape parameter in the shape subspace, and $L_{DA}(S)$ is the matrix consisting of a set of ortho normal base vectors by using LDA.

The linear discriminant analysis is given by in Eq. (26).

$$L_{DA} = \left| \frac{l_{f_{(\text{det})}}^T h_b l_{f_{(\text{det})}}}{l_{f_{(\text{det})}}^T h_w l_{f_{(\text{det})}}} \right| \quad (26)$$

where $l_{f_{(\text{det})}}$ denotes the linear transformation that maps $f_{(\text{det})}$. Then h_b and h_w are the inter-class and intra-class scatter matrix. Afterward, the images are warped to mean shape based on the corresponding points to produce shape-free patches.

Next, the texture in the frame is generated by applying scaling ε and offset δ which is given in Eq. (27).

$$T_{(f_{(\text{det})})} = \frac{(T_i - \delta \bullet 1)}{\varepsilon} \quad (27)$$

where, T_i represents the intensity of texture.

Then, the texture is ultimately projected onto the texture subspace based on LDA that is expressed as in Eq. (28).

$$T = \mu(T) + L_{DA}(T) \bullet T_p \quad (28)$$

where, $\mu(T)$ represents the mean of texture, $L_{DA}(T)$ is the matrix consisting of a set of orthonormal base vectors by

using LDA and T_p represents the texture parameter in the texture subspace.

Finally, the correlation between the shape and the texture is used to generate a combined appearance model, which is analyzed by LDA. Thereby, the shape and texture can be described as in Eqs. (29) and (30).

$$S = \mu(S) + M_s \bullet A \quad (29)$$

$$T = \mu(T) + M_T \bullet A \quad (30)$$

where, A is the appearance parameter vector that controls both the shape and the texture, while M_s and M_T are the matrices that describe the mode of variation derived from the training set.

Finally, the output obtained from the face alignment step is a video(V) that is given as input to the further process.

4) Motion magnification

After face alignment, the Eulerian Video Magnification (EVM) is used to magnify subtle motions in the video(V) by enhancing motion differences. It can make subtle movements and adjustments in video visible and louder. EVM makes the real motion more obvious and allows us to see movements that are hidden from the human eye. EVM techniques are divided into two types: linear-based EVM and phase-based EVM. The linear-based EVM is concentrated in this work for motion magnification. Motion in video is proportional to the intensity variation over the first-order expansion of the Taylor series in LEVM. To reveal hidden information, a video sequence is considered as input, spatial decomposition is used, and frames are filtered by a temporal filter.

The intensity can be expressed as a function of displacement $\delta(tm)$ such that:

$$X_V(r, tm) = H(r + \delta(tm)) \quad (31)$$

where $X_V(r, tm)$ denotes the intensity of the original video frame at a certain position r and time tm .

The video frame is approximated on the basis of first-order Taylor's series. The displaced frame $H(r + \delta(tm))$ in first-order Taylor's series is given in the Eq. (32).

$$X_V(r, tm) \approx H(r) + \delta(tm) \frac{\partial H(r)}{\partial r} \quad (32)$$

Then, the temporal filtering is performed on each spatial frequency band by performing a band pass filter to $X_V(r, tm)$ for extracting the frequency band of interest.

The output of the temporal band pass filtering $U(r, tm)$ process is given in Eq. (33).

$$U(r, tm) = \delta(tm) \frac{\partial H(r)}{\partial r} \quad (33)$$

where, $U(r, tm)$ is amplified by the amplification factor λ and added back to the original frame.

The resultant frame is given by in Eqs. (34)–(36).

$$\hat{X}_V(r, tm) = X_V(r, tm) + \lambda U(r, tm) \quad (34)$$

$$\hat{X}_V(r, tm) = H(r) + (1 + \lambda) \delta(tm) \frac{\partial H(r)}{\partial r} \quad (35)$$

Therefore, the output of LEVM is given by:

$$\hat{X}_V(r, tm) = H(r + (1 + \lambda) \delta(tm)) \quad (36)$$

From the output, if the motion occurred by means of facial expression, the further process will be carried out otherwise it will be ignored.

D. Feature Extraction

After preprocessing, the feature extraction process was done to extract the important feature for classification purposes. In this work, the features such as Local Binary Pattern on Three-Orthogonal Planes (LBP-TOP), Bi-Weighted Oriented Optical Flow (Bi-WOOF), Kanade Lucas Tomasi (KLT), and Phase-Only Correlation (POC) features are extracted from the frames (Z) of ($\hat{X}_V(r, tm)$) for detecting the expression of the subtle motion in the face. The features are explained below.

LBP-TOP: LBP-TOP is an operator used to describe the local texture characteristic of the frame. The goal of LBP-TOP is to compare the value of a frame's central pixel to the value of its neighboring pixel. Let the pixel k with the intensity value v_k , radius (rad), and m neighboring pixels. If the intensity value of a given pixel is greater than the intensity value of the centre pixel, "1" is assigned to the pixel; otherwise, "0" is assigned to the pixel. The LBP-TOP L_{BP} is given by in Eq. (37).

$$L_{BP} = \sum_{q=0}^{m-1} \tau(v_k({}^z n_q) - v_k(Z)) 2^q \quad (37)$$

where, $v_k({}^z n_q)$ represents the intensity value of q^{th} neighboring pixel in the frame (Z), $v_k(Z)$ represents the intensity value of the center pixel, τ represents the sign function, and 2^q denotes the weight that corresponds to the neighboring pixel location.

BI-WOOF: BI-WOOF utilizes onset frames and apex frames to perform emotion recognition. From the frames, the orientation ϕ and magnitude ω of the flow vector computed from the spotted apex frame and neutral reference frame are calculated by using the Eqs. (38) and (39).

$$\phi = \tan^{-1} \frac{B_z}{A_z} \quad (38)$$

$$\omega = \sqrt{(A_z^2 + B_z^2)} \quad (39)$$

where, A, B represents the flow vector from the frames z . Then, the magnitude and optical strain values are used to weigh the orientation values locally and globally to form the Bi-WOOF features (Bi_{woof}).

KLT: KLT is a typical sparse optical flow algorithm used in feature tracking. It works by identifying good feature points in the facial area in the first frame. These feature points are tracked across all frames. The tracking of the frame is dependent on the movement of the feature centre in two successive frames, which are given by in Eqs. (40)–(42).

$$\hat{X}_{V(z)} = \hat{X}_{V(z-1)} + (\rho_z - \rho_{z-1}) \quad (40)$$

$$\rho_z = \frac{1}{|fp_z|} \sum fp_z \quad (41)$$

$$\rho_{z-1} = \frac{i}{|fp_{z-1}|} \sum fp_{z-1} \quad (42)$$

where, $\hat{X}_{V(z)}$ and $\hat{X}_{V(z-1)}$ represent the face area in two adjacent frames; ρ_z and ρ_{z-1} represent the position center of features in two consecutive frames respectively; fp_z and fp_{z-1} represent the feature points in current and previous frames.

POC: POC is used to compare the similarity of two frames' blocks. The inverse 2D discrete Fourier transformation is used in POC. The position and height of the correlation peaks can be used to calculate the translation and similarity between the two images. The POC (P_{oc}) is given in Eqs. (43) and (44).

$$P_{oc} = \begin{cases} 1 & \text{if } \left(\hat{X}_{V(z)} = \hat{X}_{V(z-1)} \right) \\ 0 & \text{otherwise} \end{cases} \quad (43)$$

Hence, the final extracted features $F_{n_{(ext)}}$ are given by:

$$F_{n_{(ext)}} = [L_{BP}, Bi_{woof}, J_z, P_{oc}] \quad (44)$$

Then, the extracted features are given to the classifier for classifying the micro expressions.

E. Classification

After feature extraction, the classification process was done to classify the micro expression as Sadness, Happiness, Depression, Disgust, Surprise, and Fear by using the BDCNN Classifier. An artificial neural network called a CNN is used to recognize images. An input layer, an output layer, and a hidden layer with several convolutional layers, pooling layers, fully connected

layers, and normalization layers are the important layers of a CNN. To overcome the overfitting problem in CNN, dropout is used in a fully connected layer and the weight matrix or kernel is calculated based on Binomial

Distribution. Therefore, the network will be accurate even in the absence of certain information. Hence, the proposed method is named BDCNN. The architecture of BDCNN is shown in Fig. 4.

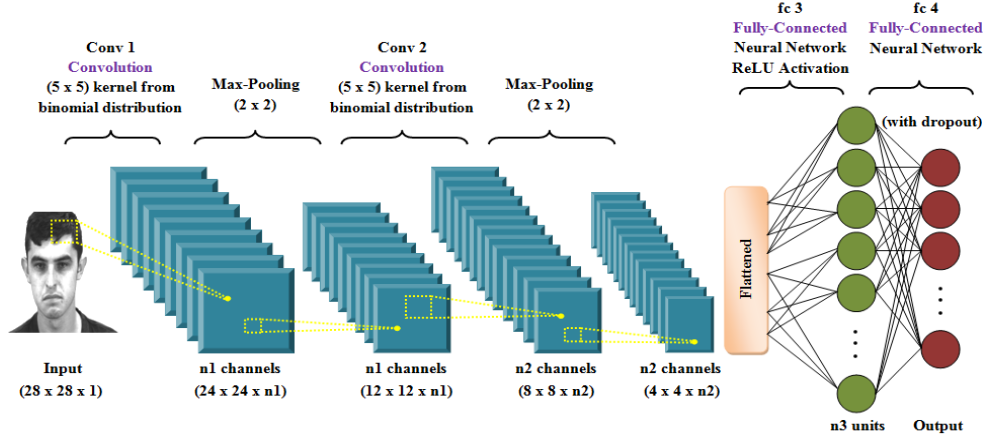


Fig. 4. Architecture of Binomial Dropout Convolutional Neural Network (BDCNN).

The steps of BDCNN are explained below:

At first, the extracted features $F_{n(ext{ext})}$ are given as input to the convolution layer. The convolution layer is the foundational component of CNN. It has a number of filters, each of which convolves with the input to produce an activation map. The convolution layer Con_{lyr} is expressed as in Eq. (45).

$$Con_{lyr} = \eta \left(\sum_{i_{(ch)}} \sum_{i,j} F_{n(ext{ext})} W \right) \quad (45)$$

where, $i_{(ch)}$ represents the channel index, (i, j) is the coordinates of extracted features $F_{n(ext{ext})}$, $\eta(\cdot)$ represents the ReLU activation function, and W represents the convolutional kernel, which is calculated based on the binomial distribution. The binomial distribution summarizes the number of trials, or observations, when each trial has the same chance of achieving a specific value. Thereby, the convolutional kernel is given by in Eq. (46).

$$W = \left(\frac{u}{v} \right) p^v o^{u-b} \quad (46)$$

where, u represents the number of trials, v represents the number of successes desired, p represents the probability of success in one trial, and o represents the failure in one trial.

Followed by the convolution layer, a pooling layer that is also known as down sampling is used to reduce the complexity for further layers. It subsamples a small area of the convolutional output as input to create a single output. The output of the pooling layer is given in Eq. (47).

$$Pol_{lyr} = \text{down}(\max(Con_{lyr})) \quad (47)$$

where, $\text{down}(\cdot)$ denotes the down sampling that retains the maximum value in the pooled area. After all the convolution pooling layers, the output is converted into a single long linear vector by using flattening. Next, to overcome the over-fitting problem, the network uses the dropout layer for training. The Dropout layer is a mask that removes some neurons' contributions to the next layer while leaving all others alone.

Then, the flattened output (ft) is given as input to the fully connected layer where the softmax function SF is used to produce classified output which is given by in Eq. (48).

$$SF = \frac{\exp(ft)}{\sum \exp(ft)} \quad (48)$$

where, (ft) represents the flattened output.

The Pseudo-code of BDCNN is as follows:

Input: Extracted Features $F_{n(ext{ext})}$
Output: Classified output

Begin
Initialize extracted features $F_{n(ext{ext})}$, maximum iterations i_{\max}
Set Initial iteration $i_t = 1$
While ($i_t \leq i_{\max}$) **do**
 Assign weight values by binomial distribution
 $W = \left(\frac{u}{v} \right) p^v o^{u-b}$
 For every $F_{n(ext{ext})}$ **do**
 Compute convolution layer con_{lyr}
 Compute pooling layer pol_{lyr}
 End For
 Flatten the output
 Compute dropout
 Apply softmax function
 $SF = \frac{\exp(ft)}{\sum \exp(ft)}$
 End While
 Return Classified output
End

Finally, the classifier recognizes the micro-expression as sadness, happiness, depression, disgust, surprise, fear, positive-negative, and others

IV. RESULT AND DISCUSSION

In this segment, the performance of the proposed system is compared with other existing methods in terms of performance metrics like sensitivity, specificity, precision, recall, and F-Measure. The output results are calculated by taking the average of different categories of micro facial expressions like happiness, sad, disgust, positive, and negative. Different types of data sources namely CASME II, SMIC, and SAMM are used for the proposed micro facial expression recognition system, and it is implemented in the working platform of MATLAB (version R 2020a)

A. Dataset Description

In this, micro facial expression like happiness, sadness, regression, and disgust, are taken from different data sources namely CASME II, SMIC, and SAMM. This SMIC dataset comprises 164 video sequences of facial micro-expressions taken from 16 people while they were in a frontal stance. These scenes are captured with a standard visual camera in a controlled environment (VIS). Generally, SMIC consists of 51 positive videos, 70 negative sequence videos, and 43 surprise sequence videos. Likewise, the facial micro-expressions are distributed for CASME II, which includes 63 video sequences of disgust, 27 video sequences of regression, 32 video sequences of happiness, 25 video sequences of surprise, 7 video sequences of sadness, 2 video sequences of fear, and 100 video sequences for others. Finally, SAMM consists of 159 samples of video sequences from 32 subjects and seven main classes like contempt, disgust, fear, anger, sadness, happiness, and surprise.

B. Performance of the Proposed BDCNN in the Three Different Datasets

Here, the results obtained by the proposed BDCNN approach are discussed on the three datasets such as CASME II, SMIC, and SAMM datasets.

1) Evaluation of proposed BDCNN in CASME II dataset

Here, the performance of the proposed BDCNN algorithm for micro-expression on CASME II is discussed. The sample image outputs obtained from the proposed model when applied to the CASME II dataset are given in Fig. 5 as follows.

The sample micro-expressions such as disgust, happiness, fear, repression, surprise, sad, and other expression frames are given as input to the proposed and its corresponding output (Fig. 5).

The experimental output of the proposed BDCNN with the expressions is obtained in terms of accuracy and F-Measure.

Table I depicts the experimentally analyzed output values of the proposed BDCNN approach on the benchmark dataset. Accuracy was calculated to identify how correctively a system recognizes the micro-expression. The value of 1 means, that the expressions were recognized without any error. Here, the proposed model recognized the expressions of happiness, sadness, repression, and fear without any error. Hence, it can be concluded that the performance of the proposed model works significantly on the CASME II dataset.

TABLE I. EXPERIMENTAL OUTPUTS OBTAINED ON THE CASME II DATASET

Expressions	Accuracy	F-Measure
Disgust	0.96226	0.92308
Happiness	1	1
Fear	1	1
Repression	1	1
Surprise	0.96226	0.83333
Sadness	1	1
Others	0.92453	0.9

1) Evaluation of suggested method on SAMM dataset

The frames obtained from the videos available in the SAMM dataset are given to the suggested model for the classification of micro-expressions such as anger, contempt, disgust, fear, happiness, sadness, and surprise. The output sample images obtained after the preprocessing are given in Fig. 6.



Fig. 5. (a) Sample Micro Expressions obtained from the videos of the CASME II dataset. (b) Output images of EPHF. (c) YOLOv3 output images.



Fig. 6. Input images from the SAMM dataset and its corresponding preprocessed outputs in the proposed model. (i) Sample input frames. (ii) Illumination-controlled images. (iii) Detected face in the images.

The presentation of the suggested framework is numerically analyzed in terms of accuracy, precision, recall, and F-Measure for the various micro-expressions available in the SAMM dataset.

The simulation output values of the proposed BDCNN classifier are shown as the graphical representation in Fig. 7. From Fig. 7, it is evident that the proposed model recognizes the micro-expressions of disgust, fear, surprise, and sadness without any discrimination. Even though, some of the expressions failed to be recognized by the proposed classifier. The average performance of the model shows better performance on the SAMM dataset in terms of accuracy and F-Measure, which is given in Table II.

2) Evaluation of dataset SMIC

Similar to the analyzes of the CASME and SAMM datasets, the proposed model also used the SMIC dataset

to analyze the efficiency of the proposed ME model. Thus, the output obtained after the preprocessing steps of the incoming image frame is shown in Fig. 8.

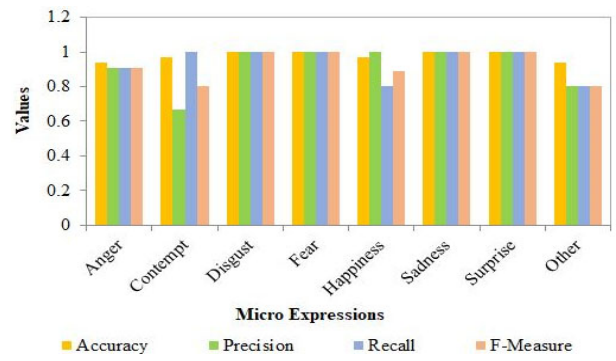


Fig. 7. Experimental output values obtained on SAMM dataset.



Fig. 8. Experimental output images of the preprocessing techniques on the dataset SMIC. (a) Input micro expression frames. (b) Images after illumination control by EPHF technique. (c) Images obtained on the output of YOLOv3.

TABLE II. COMPARATIVE ANALYSIS OF THE SUGGESTED BDCNN CLASSIFIER WITH THE OTHER PREVAILING APPROACHES

Methods	CASME II		SMIC		SAMM	
	Acc (%)	F-Score (%)	Acc (%)	F-Score (%)	Acc (%)	F-Score (%)
STM-Net [18]	84.84	84.80	78.66	79.73	81.13	74.36
OFF-ApexNet [25]	88.28	86.97	67.68	67.09	68.18	54.23
DSTAN [21]	75	73	77	78	-	-
LFSSOA-RBFNN [22]	89.34	91.24	-	-	-	-
ResNet with micro attention [26]	65.9	53.9	49.4	49.6	48.5	40.2
Proposed BDCNN	97.84	95.09	89.89	85.36	97.58	92.47

Here, the numerical analysis of the suggested BDCNN classifier for the recognition of micro-expression is performed on the SMIC dataset. The SMIC dataset contains video files of micro expressions such as negative, positive, and surprise reactions. The performance is analyzed based on performance metrics such as Negative Predictive Value (NPV), Mathews Correlation Coefficient (MCC), accuracy, precision, recall, F-Measure, and sensitivity. The pictorial representation based on the results obtained when experiments are performed on the SMIC dataset is shown in Fig. 9.

Here, the efficiency of the suggested BDCNN algorithm is analyzed based on various performance measures. Although the proposed model wrongly predicts a few of the emotions compared to the other existing works, the performance of the suggested technique is superior; it can be evident from Table II. Hence, it can be said that the average performance of the proposed method is also dominant in the recognition of micro expressions on the SMIC dataset.

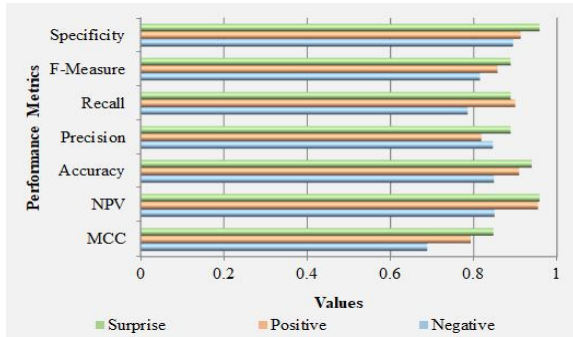


Fig. 9. Analyzed results of the suggested method on the SMIC dataset.

C. Comparative Analysis of the Suggested BDCNN

Here, the performance of the proposed Binomial Dropout Convolutional Neural Network (BDCNN) classifier is analyzed and compared with the state of the arts approaches based on the accuracy and F-Measure values obtained in the CASME II, SMIC, and SAMM datasets.

Table II depicts the comparative analysis of the suggested Micro-Expression recognition model with the state of arts ME recognition models such as Single Scale Multiscale-Network (STM-Net), Optical Flow Features from Apex frame Network (OFF-ApexNet), Dualstream Spatio Temporal Attention Network (DSTAN), Levy Flight based Shufed Shepherd Optimization Algorithm-Radial Basis Function Neural Network (LFSSOA-RBFNN) and Resolution Network with micro attention. Here, the recognition accuracy (Acc) values of the

proposed method are 97.84% on the CASME II dataset, 89.89% on the SMIC dataset, and 97.58% on the SAMM dataset, which are very much higher than the other compared approaches. Also, the F-Measure value of the proposed BDCNN algorithm on the SAMM dataset is 70.51% higher than the OFF-ApexNet and 24.35% higher than the STM-Net. Similarly, the F-Measure values of the proposed algorithm on the CASME II and SMIC datasets are also higher when compared to other methods. This means that the precision and recall values are also high since the F-Measure is calculated by the combination of precision and recall. From these observations, it can be concluded that the suggested method is dominant over the other methods in the recognition of Micro-Expressions.

V. CONCLUSION

Facial expressions can provide a great source of information in our day-to-day social communication. Basically, facial micro-expression is classified as macro and micro facial expressions. In this, micro facial expressions cannot be easily identified by humans, extremely difficult to capture and analyze the micro-expressions and their low intensity makes it hard to distinguish from neutral expressions or noise. In order to identify the Micro facial expression method, a modified BDCNN is proposed. In this proposed method, Distance Deviation Based Root Pattern algorithm and Edge Preserving homomorphic filter are introduced for motion estimation. Finally, based on the proposed technique, the classifier produces the result, such as sadness, happiness, disgust, surprise, and fear and the average of this is analyzed by using the parameters like accuracy, recall, specificity, F-Measure, precision, NPV, and MCC. The obtained outcomes are accuracy (97.84%) for CASME II, which is higher when compared to the existing LFSSOA-RBFNN algorithm, STM-Net, OFF-ApexNet, DSTAN, and ResNet with micro attention. Likewise, the F-Measure of the proposed system is 85.36 % for the SMIC dataset, which is higher when compared to ResNet with micro attention, DSTAN, OFF-ApexNet, and STM-Net. Similarly, other metrics also vary for other datasets. This shows that the proposed system has high performance when compared with the existing method. However, there are known limitations associated with the use of CNNs for facial expression like the relatively small data set of images, high intra-class similarity, short duration of expressions, and the tendency to overfit to training. In the future, modified systems neural networks are developed to explore the relationship between facial muscle movements (Action units) and human emotions.

DATA AVAILABILITY STATEMENT

Data sharing is not applicable to this article as no datasets were generated or analyzed during the current study.

CONFLICT OF INTEREST

The authors declare that they have no conflict of interest in the submission of this article for publication.

AUTHOR CONTRIBUTION

Anjani Suputri Devi D: Define the problem statement, methodology. M Kishore Kumar: Algorithm analysis. Chinnam Sabitha: Drawing Figures. P. V. V. Sree Rama Kumari: Helping in algorithm analysis. Sasi Rekha D: Simulations of results. All authors had approved the final version.

REFERENCES

- [1] S. T. Liong, J. See, R. C. W. Phan *et al.*, "Hybrid facial regions extraction for micro-expression recognition system," *J. Signal Process. Syst.*, vol. 90, no. 4, pp. 601–617, 2017.
- [2] U. Saeed, "Facial micro-expressions as a soft biometric for person recognition," *Pattern Recognit. Lett.*, vol. 143, pp. 95–103, 2021.
- [3] X. Ben, Y. Ren, J. Zhang *et al.*, "Video-based facial micro-expression analysis: A survey of datasets, features and algorithms," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 9, pp. 5826–5846, 2021.
- [4] X. Jia, X. Ben, H. Yuan *et al.*, "Macro-to-micro transformation model for micro-expression recognition," *J. Comput. Sci.*, vol. 25, pp. 289–297, 2017.
- [5] J. Wu, Y. Min, X. Yang *et al.*, "Micro-expression recognition algorithm based on information entropy feature," *J. Shanghai Jiaotong Univ.*, vol. 25, no. 5, pp. 589–599, 2020.
- [6] S. T. Liong, J. See, K. S. Wong, and R. C. W. Phan, "Less is more: Micro-expression recognition from video using apex frame," *Signal Process. Image Commun.*, vol. 62, pp. 82–92, 2017.
- [7] H. X. Xie, L. Lo, H. H. Shuai *et al.*, "An overview of facial micro-expression analysis: Data, methodology and challenge," *IEEE Trans. Affect. Comput., Early Access*, vol. 14, no. 3, pp. 1857–1875, 2020. doi: 10.1109/TAFFC.2022.3143100
- [8] J. Pei and P. Shan, "A micro-expression recognition algorithm for students in classroom learning based on convolutional neural network," *Trait. Signal*, vol. 36, no. 6, pp. 557–563, 2019.
- [9] Y. Zong, W. Zheng, Z. Cui *et al.*, "Toward bridging micro-expressions from different domains," *IEEE Trans. Cybern.*, vol. 50, no. 12, pp. 5047–5060, 2019.
- [10] Y. Zong, X. Huang, W. Zheng *et al.*, "Learning from hierarchical spatiotemporal descriptors for micro-expression recognition," *IEEE Trans. Multimedia*, vol. 20, no. 11, pp. 3160–3172, 2018.
- [11] X. Li, X. Hong, A. Moilanen *et al.*, "Towards reading hidden emotions: A comparative study of spontaneous micro-expression spotting and recognition methods," *IEEE Trans. Affect. Comput.*, vol. 9, no. 4, pp. 563–577, 2016.
- [12] K. H. Liu, Q. S. Jin, H. C. Xu *et al.*, "Micro-expression recognition using advanced genetic algorithm," *Signal Process. Image Commun.*, vol. 93, no. 3, pp. 1–10, 2021.
- [13] H. Pan, L. Xie, Z. Lv *et al.*, "Hierarchical support vector machine for facial micro-expression recognition," *Multimedia Tools Appl.*, vol. 79, no. 41–42, pp. 31451–31465, 2020.
- [14] M. Aouayeb, W. Hamidouche, C. Soladie *et al.*, "Micro-expression recognition from local facial regions," *Signal Process. Image Commun.*, vol. 99, no. 1, pp. 1–13, 2021.
- [15] Q. Li, S. Zhan, L. Xu *et al.*, "Facial micro-expression recognition based on the fusion of deep learning and enhanced optical flow," *Multimedia Tools Appl.*, vol. 78, no. 19, pp. 29307–29322, 2019.
- [16] H. Pan, L. Xie, Z. Wang *et al.*, "Review of micro-expression spotting and recognition in video sequences," *Virtual Real. Intell. Hardw.*, vol. 3, no. 1, pp. 1–17, 2021.
- [17] R. Guermazi, T. B. Abdallah, and M. Hammami, "Facial micro-expression recognition based on accordion spatio-temporal representation and random forests," *J. Vis. Commun. Image Represent.*, vol. 79, pp. 1–16, 2021.
- [18] J. Wang, X. Pan, X. Li *et al.*, "Single trunk multi-scale network for micro-expression recognition," *Graph. Vis. Comput.*, vol. 4, pp. 1–9, 2021.
- [19] Y. J. Liu, B. J. Li, and Y. K. Lai, "Sparse MDMO learning: A discriminative feature for micro-expression recognition," *IEEE Trans. Affect. Comput.*, vol. 12, no. 1, pp. 254–261, 2018.
- [20] X. Li, G. Wei, J. Wang *et al.*, "Multi-scale joint feature network for micro-expression recognition," *Comput. Vis. Media*, vol. 7, no. 3, pp. 407–417, 2021.
- [21] Y. Wang, Y. Huang, C. Liu *et al.*, "Micro-expression recognition via dual-stream spatiotemporal attention network," *J. Healthc. Eng.*, 2021. doi: 10.1155/2021/7799100
- [22] R. Banerjee, S. De, and S. Dey, "WTAOF-ILPB based feature learning and LFSSOA-RBFNN based classification for facial micro-expression recognition," *Wirel. Pers. Commun.*, vol. 127, no. 3, pp. 2285–2304, 2021. doi: 10.1007/s11277-021-08794-5
- [23] J. Li, Y. Wang, J. See *et al.*, "Micro-expression recognition based on 3D flow convolutional neural network," *Pattern Anal. Appl.*, vol. 22, no. 12, pp. 1331–1339, 2018.
- [24] M. Verma, S. K. Vipparthi, G. Singh *et al.*, "LEARNet: Dynamic imaging network for micro expression recognition," *IEEE Trans. Image Process.*, vol. 29, pp. 1618–1627, 2019.
- [25] Y. S. Gan, S. T. Liong, W. C. Yau *et al.*, "OFF-ApexNet on micro-expression recognition system," *Signal Process. Image Commun.*, vol. 74, pp. 129–139, 2019.
- [26] C. Wang, M. Peng, T. Bi *et al.*, "Micro-attention for micro-expression recognition," *Neurocomputing*, vol. 410, pp. 354–362, 2020.
- [27] J. Li, T. Wang, and S. J. Wang, "Facial micro-expression recognition based on deep local-holistic network," *Appl. Sci.*, vol. 12, no. 9, p. 4643, 2022. doi: 10.3390/app12094643
- [28] H. Pan, H. Yang, L. Xie *et al.*, "Multi-scale fusion visual attention network for facial micro-expression recognition," *Front. Neurosci.*, vol. 17, 2023. doi: 10.3389/fnins.2023.1216181
- [29] Y. Tang, J. Yi, and F. Tan, "Facial micro-expression recognition method based on CNN and transformer mixed model," *Int. J. Biometrics*, vol. 16, no. 5, 2024. doi: 10.1504/IJBM.2024.140771
- [30] A. Paul, R. Machavaram, D. Kumar, and H. Nagar, "Smart solutions for capsicum harvesting: unleashing the power of YOLO for detection, segmentation, growth stage classification, counting, and real-time mobile identification," *Comput. Electron. Agric.*, vol. 219, 108832, 2024.

Copyright © 2025 by the authors. This is an open access article distributed under the Creative Commons Attribution License ([CC-BY-4.0](https://creativecommons.org/licenses/by/4.0/)), which permits use, distribution and reproduction in any medium, provided that the article is properly cited, the use is non-commercial and no modifications or adaptations are made.