University of Dayton

eCommons

Electrical and Computer Engineering Faculty Publications

Department of Electrical and Computer Engineering

10-2021

A Unified Framework of Deep Learning-Based Facial Expression Recognition System for Diversified Applications

Sanoar Hossain (0000-0002-1232-7487)

Saiyed Umer (0000-0002-1476-041X)

Vijayan K. Asari (0000-0002-3751-5492)

Ranjeet Kumar Rout (0000-0002-1546-1702)

Follow this and additional works at: https://ecommons.udayton.edu/ece_fac_pub

 Part of the Computer Engineering Commons, Electrical and Electronics Commons, Electromagnetics and Photonics Commons, Optics Commons, Other Electrical and Computer Engineering Commons, and the Systems and Communications Commons

applied sciences

MDPI

Article

A Unified Framework of Deep Learning-Based Facial Expression Recognition System for Diversified Applications

Sanoar Hossain [1,†], Saiyed Umer [1,*,†], Vijayan Asari [2] and Ranjeet Kumar Rout [3]

[1] Department of Computer Science and Engineering, Aliah University, Kolkata 700156, India; snr.hossain12@gmail.com
[2] Electrical and Computer Engineering, University of Dayton, 300 College Park, Dayton, OH 45469-0232, USA; vasari1@udayton.edu
[3] Department of Computer Science and Engineering, National Institute of Technology, Srinagar 190006, India; ranjeetkumarrout@nitsri.net
* Correspondence: saiyedumer@gmail.com
† These authors contributed equally to this work.

Abstract: This work proposes a facial expression recognition system for a diversified field of applications. The purpose of the proposed system is to predict the type of expressions in a human face region. The implementation of the proposed method is fragmented into three components. In the first component, from the given input image, a tree-structured part model has been applied that predicts some landmark points on the input image to detect facial regions. The detected face region was normalized to its fixed size and then down-sampled to its varying sizes such that the advantages, due to the effect of multi-resolution images, can be introduced. Then, some convolutional neural network (CNN) architectures were proposed in the second component to analyze the texture patterns in the facial regions. To enhance the proposed CNN model's performance, some advanced techniques, such data augmentation, progressive image resizing, transfer-learning, and fine-tuning of the parameters, were employed in the third component to extract more distinctive and discriminant features for the proposed facial expression recognition system. The performance of the proposed system, due to different CNN models, is fused to achieve better performance than the existing state-of-the-art methods and for this reason, extensive experimentation has been carried out using the Karolinska-directed emotional faces (KDEF), GENKI-4k, Cohn-Kanade (CK+), and Static Facial Expressions in the Wild (SFEW) benchmark databases. The performance has been compared with some existing methods concerning these databases, which shows that the proposed facial expression recognition system outperforms other competing methods.

Keywords: convolutional neural networks; deep learning; diversified field; facial expression; recognition

check for updates

Citation: Hossain, S.; Umer, S.; Asari, V.; Rout, R.K. A Unified Framework of Deep Learning Based Facial Expression Recognition System for Diversified Applications. *Appl. Sci.* **2021**, *11*, 9174. https://doi.org/10.3390/app11199174

Academic Editor: Monica Perusquia Hernandez

Received: 4 July 2021
Accepted: 28 September 2021
Published: 2 October 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Copyright: © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

Facial expressions [1] are a crucial non-verbal method of indicating meaning and represent a unique, universal way for people to communicate. The facial expression recognition system (FERS) is a contactless recognition system in which the image of a human face of a person can be captured from a distance without any intervention or interruption, even when he/she is moving around, walking, sitting, or performing activities [2]. Facial expressions play an essential role in our daily communication with people and in social interactions [3]. The FERS is mainly used to identify types of human facial expression [4]. According to Ekman et al. [5], there are six basic expressions, including a neutral face as a baseline reference. Figure 1 shows some examples of human facial expressions, e.g., fear (FA), anger (AN), disgust (DI), happy (HA), neutral (NE), sad (SA), and surprise (SU).

Figure 1. Example of seven types of facial expressions for the FER system.

The FERS [6] is an emergent research topic in computer vision research areas. It has comprehensive potential in a diversified field of applications [7] with various challenges in healthcare, education, marketing research, business organization, customer and retail fields, government, entertainment, and within the Internet of Things (IoT). There exists several vendors such as Microsoft, IBM, Amazon, and Google that provide some application programming interfaces using facial expressions but with limited solutions. Here, these diversified fields of application for FERS are discussed in terms of the four major concerns, which are as follows:

- *e-Healthcare:-* The real-time FERS incorporates the healthcare system via the system that is used to analyze and detect the image's visualization of the patient's feelings remotely by identifying facial expressions [8] for patients with different ages, puberty levels, and genders collected from a giant cloud and from social networks. The m-Health provides mobile device-based practices to patients to support their medicine administration and daily healthcare facilities. e-Health is an electronic health service that uses information and communication technology for delivering facilities digitally and for processing patients and doctors through computers for drug administration. Both e-Health and m-Health provide immense support to healthcare industries in building e-Healthcare systems to ensure that patients, doctors, medical professionals, and businesses benefit, as well as to ensure the establishment of a healthy civilization with technological advancements in smart cities. Electronic healthcare systems provide services to patients to physically localize and monitor through recognizing their voice, speech, gesture movement, and facial expressions. Our proposed facial expression recognition system (FERS) will improve the services of healthcare systems. It is a significant challenge to obtain good results in the context of more efficient and less costly health services. Hence, while integrating the FERS into the healthcare framework, all healthcare requirements, such as automated intelligent sensors, sophisticated tools, security, authenticity, access, and privacy, should also be considered.
- *Social IoT:-* Social IoT systems represent an evolution of IoT-based systems. It establishes a platform for interconnecting subjects or objects worldwide through social relationships. It provides better services to users by relaxing the common interests between the users. Now, the services of social IoT are exploited in emotion-recognition as these emotions relate to the social activities of humans in their daily life. Hence, the integration of social IoT services will make life easier with several social care facilities for people [9]. The proposed FERS is useful for developing IoT-based smart devices and appliances. It can be used for several entities such as education, marketing research, retail, government, media and content, gaming, and finance. During online teaching, the facial expressions of students can be compared to their interest and understanding of topics that have been taught to them. The sentiments from online trading and investment strategies will be beneficial for the financial development of the organization. The emotion analysis using customer reviews and shopkeepers' experience will bring good marketing research for the organization.
- *Emotion AI:-* Emotion AI [10] has wide applications in human resource management, such as in any business organization. It helps the human resource management

system (HRMS) during the recruitment of a candidate for selection. This emotion AI considers several traits such as voice and text to analyze the sentiments in candidates.

- *Cognitive AI:-* Cognitive AI [11] provides methods and technologies to build a decision-making system based on the behavior and reasoning ability of a person. It helps a person to make decisions through a system. Job searching, salary prediction, carrier path selection for job-seeker problems, cyber-security with enabled AI, and natural language processing for sentiment analysis problems are under cognitive AI categories. Thus, social interaction, planning, interpretation, decision-making, competence of emotion, and self-learning capabilities are the processes of cognitive AI.

These applications use facial images for recognizing expressions in humans. The psychology of facial expression [12] states that the face is the key to understanding emotions. Linking the face to emotions may be an important idea in the psychology of emotions. The facial expression recognition system works on the facial movements [13], which are described by the facial action coding system (FACS). The FACS breaks down facial expressions into action units that introduce a distinct change in the facial appearance. There are various uses of FACS for discovering disorders in neuropsychiatric and social–emotional development that are performed through psychological research. The FACS is an immediate, powerful, and effective non-verbal communication tool to transit messages and convey emotional information. In most of the implementation cases of FERS, the facial region is analyzed as a texture where numerous techniques such as statistical and structural-based methods have been employed to extract discriminant features [14]. Apart from these techniques, recently, deep learning-based approaches with convolution neural networks [15] have been employed to extract more discriminant and distinctive features to ensure that a better performance can be obtained. However, most of these methods are database-dependent and these databases have been captured spontaneously under controlled environments [16] with tightly controlled illumination, age, and pose variation conditions.

Despite the current state-of-the-art methods for the FERS and their significant progress in effective computing, they still suffer from some limitations: (i) The employed datasets are either laboratory-controlled or wild. These images are captured under unconstrained environments and the images suffer from several challenging issues such as illumination, poor resolution, occlusion, pose, age, and expression variations. Thus, the extraction of the face region from the input images in optimal time is also a challenging issue. (ii) Due to limited domain knowledge, the local to global feature representation schemes generate less discriminative and distinctive patterns. (iii) The assumption of the feature selection might not be valid, i.e., the extraction of local geometric information or action units' geometric features is not valid. Hence, we have proposed a novel deep learning-based framework for the FERS to address these problems and improve its usefulness in diversified fields of applications such as in e-Healthcare, social IoT, and emotion AI. The contributions of the proposed work are as follows:

- We have designed a fast and efficient end-to-end deep learning-based framework using the convolutional neural network approach for learning face representation by adding some extra levels of feature representation schemes to improve the robustness and generalization of the model.
- The obtained predictive model detects and learns powerful high-level features from the input image and extracts more distinctive and discriminant features that provide effective results for the proposed FERS under various illumination changes as well as pose and age variation artifacts.
- To enhance the performance of the FERS, several experiments have been carried out with a trade-off between the batch vs. epoch, data augmentation, progressive image resizing, hyper-parameter tuning, and transfer learning techniques for the better prediction of expression types on the human face and for improvement of the performance as well as robustness of the proposed system.

- The proposed method finds the solution for the challenging issues of FERS. At the same time, a series of experiments have been conducted to reduce the training loss and over-fitting problems that arise due to inadequate training data and bias in the expressions' variation.

The organization of this paper is as follows: Section 2 describes the related work for the proposed system. The proposed facial expression recognition system (FERS) is discussed in Section 3, which describes the face pre-processing techniques and the proposed CNN architectures for the feature computation of both frontal and profile facial images. The database description, experimental results, and discussions are described in Section 4. Finally, Section 5 concludes this paper.

2. Related Work

An automatic facial expression recognition and classification for multi-pose and multi-level face images have revealed to be an attractive and challenging problem since the last thirty years [17]. A literature review stated that early stages of research has focused on several statistical and structural-based methods [14]. In contrast, some [17] template-based and feature-based approaches have also been investigated. The classical methods, such as the Histogram-of-Orientation Gradient (HOG) [18], the Scale Invariant Feature Transform (SIFT) [19], LBP (Local Binary Pattern) [20] features, and some spatio-temporal features (STM-ExpLet [21]), have been adopted by many researchers to obtain texture features in statistical ways; however, these methods require great effort to achieve high performance. Recently, researchers have used convolutional neural networks (CNN, ConvNets) [22] and have achieved great success for large-scale static images and sequences of video recognition [23]. The CNN has been widely applied for the FER system and has significantly improved state-of-the-art practices as well as analyzed the performance of ImageNet classification challenges [22]. Earlier CNN models were used to solve character recognition tasks [24], but nowadays, CNN is widely used in various object recognition problems. Here, the most important ingredient for the success of CNN is the availability of large quantities of training data, i.e., the use of image augmentation techniques [15]. Additionally, the CNN achieves high performance by learning powerful high-level features by combining global appearances to local geometric features rather than conventional handcrafted features. However, the training image samples suffer from the lack of intensity noises, illumination, pose and expression variation, motion blur, low resolution, and occlusion by hair artifacts. The CNN aims towards the application of people-sentiment analysis; application to multi-modal human–machine or computer interactions; and application to intelligent systems with their challenges that arise when capturing images under an unconstrained imaging environment.

Depending on the existing state-of-art methods for face representation and facial expression, recognition could be broadly classified and analyzed into two categories: appearance-based methods and facial action units-based methods. In the appearance-based methods, the entire face region is divided into several blocks or patches and the features are extracted from these patches using the Local Binary Pattern (LBP) [25], Histogram of Oriented Gradients (HOG) [26], and Scale Invariant Feature Transform (SIFT) [19], as explored by Zhao et al. [27] and others. Facial action unit-based methods usually exploit the face geometrical information or face action units-driven representation for facial expression classification. Tian et al. [28] used the positions of facial landmarks for facial action unit recognition and then performed expression classification. The appearance-based method [29] is the most successful and well studied for face recognition. There are several works in which the whole face image captured in controlled-lab conditions was taken as the input image $\mathcal{I}_{m\times n}$ to create a subspace based on the reduction of inconsistent and redundant face space dimensionality reduction techniques [30]; for instance, Fisher LDA, PCA, and LPP [31] had been adopted. A comparative literature review of these methods for facial expression recognition have been done in [32]. The LDA and PCA practically are based on the kernel methods. The Euclidean structure and miscellaneous

learning methods [33] have been employed for face recognition [34]. The computational cost of these techniques is expensive and some of these systems may fail due to the system explicitly exhibiting the exact structure of the manifold. However, these are powerful tools based on statistical signal-modeling, which is known as sparse coding. The sparse coding provides beautiful results for the facial expression recognition [35] system. Instead of these handcrafted features, deep learning methods have been assumed to be a breakthrough in computer vision and have broken the world record in the field of recognition task problems.

Many state-of-the-art methods and deep learning frameworks use hand-labeled points and CNN architecture for both feature extraction and built facial expression recognition systems. Gutta et al. [36] proposed a model with an ensemble radial basis function, a grayscale image, and inductive decision trees for the four classes (i.e., Asian, Caucasian, African, and Oriental) ethnicity recognition problem. Zhang and Wang [37] proposed a method for two-class racial classification using multi-scale LBP (Local Binary Pattern) texture features while combining 2D and 3D texture features. Zhang et al. [38] described two types of features, namely the geometry-based features and Gabor-wavelets-based features for the FER System. Bartlett et al. [39] applied the Gabor filters coupled with feature selection and machine learning techniques for recognizing facial expressions on a human face. In [40], Rose applied Gabor and log-Gabor filters on low-resolution images for facial expression recognition. Wu et al. [41] explored the Gabor motion energy filters [42] to recognize the dynamic facial expressions of individuals. Gabor filters together with genetic algorithms (GA) and SVM for the analysis of six basic facial expressions from video sequences were employed in [25]. In [43], Gu et al. proposed a method for facial expression recognition based on the radial encoding of local Gabor features with classifier synthesis. Almaev et al. [44] proposed a new dynamic feature descriptor called the Local Gabor Binary Patterns from Three Orthogonal Planes (LGBP-TOP) by combining LBP-TOP [45] and Gabor filters.

The major problems that occur during the development of the FER System concern shallow features and bias caused by various cultures and collection conditions. Current datasets have a strong build-in bias and the corresponding proposed methods show that the conditional probability distribution between training and testing datasets are different. We will assess this bias and present novel deep CNN models to address these issues. In our proposed methodology, we considered face recognition as an image classification problem. This face recognition definition has been extended to our work for the classification of facial expressions on human faces. The proposed FER system is based on two backbones: (1) face preprocessing and (2) the design and analysis of features from the proposed CNN architectures. The proposed CNN architecture is built using several convolutional layers, max-pooling, batch normalization, and dropout layers with an optimizer followed by the soft-max classifier for the final classification tasks. Our extensive random experimental results show that our proposed deep-CNN method achieves superior results for facial expression recognition problems for both lab-controlled and real-world databases. The principal issues involved in the facial expression recognition system design are face representation and classifier selection [31]. The face representation concerns extracting feature descriptors from the input face image that minimize the intra-class similarities and maximize the inter-class dissimilarities. In the case of classifier selection, it does not make sense that the high-performance classifiers always find a better separation between different classes even if there are significant similarities. Sometimes, the most sophisticated classifier may fail to execute the facial expression recognition and classification tasks due to inadequate face representations. We cannot achieve high-performance recognition accuracy if we employ good face representation but do not select a good classifier. Hence, the below sections describe the proposed FER system.

3. Proposed Methodology

In this work, we have proposed a facial expression recognition system (FERS) in the diversified fields of applications, such as e-Healthcare, social IoT, emotion AI, and cognitive

AI. The block diagram of the proposed method is demonstrated in Figure 2. Since these fields belong to interdisciplinary research areas, the algorithms and techniques employed during the implementation of these frameworks are interconnected. Thus, the proposed FERS will be used as the common platform for analyzing expressions in the applications of these frameworks. Furthermore, the implementation of a basic FERS is discussed in the following paragraphs.

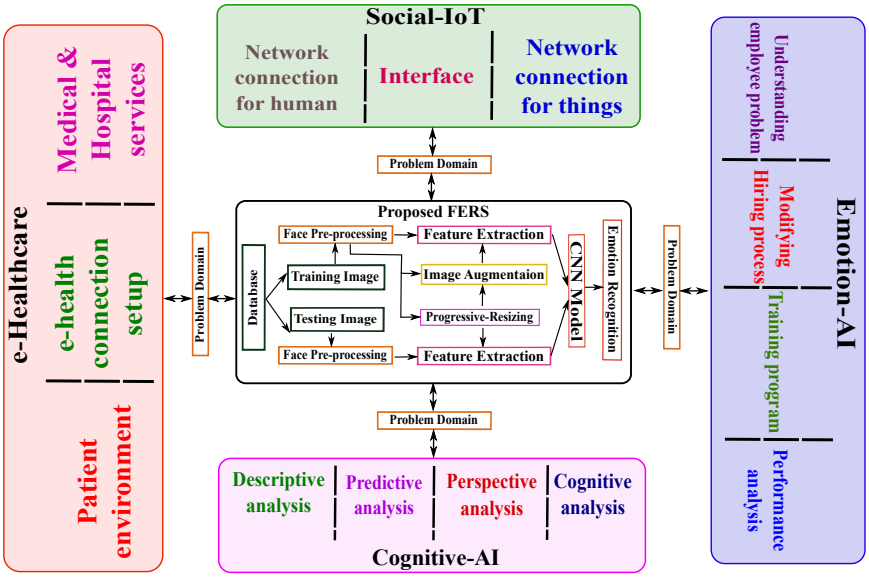

Figure 2. Block diagram of the proposed FERS.

A facial expression recognition system generally consists of face representation, feature extraction, and classifier components. Regarding the importance of face recognition in computer vision research areas, we have proposed robust, efficient, and accurate deep convolution neural network (CNN) models for facial expression recognition systems. Here, an image $\mathcal{I}_{m\times n}$ with a valid face region is used as an input to the system. Our objective is to predict the types of expressions, such fear, anger, disgust, surprise, sadness, happiness, and neutral, from the input face region $\mathcal{I}_{m\times n}$. The proposed facial expression recognition system (FERS) has been implemented in four steps: (i) pre-processing, wherein the bounded box face region is detected from the input image using the tree structure part model [38]; (ii) feature extraction, wherein the global generic CNN features are extracted from the detected bounding box face region and are prepared for the next level through the deep learning model; (iii) the representations are further modified by using multi-stage progressive image resizing followed by transfer learning methods, wherein image augmentation and fine-tuning of parameters have also been adopted; and (iv) classification, which concerns predicting the type of expression classes of the facial region. Each of these steps is described in the block diagram of the proposed system, which is presented in Figure 2. Recently, deep convolution neural network (CNN) techniques have been successfully developed to learn discriminative features in various fields. It is widely being used in deep FER representation. Deep FER suffers from the over-fitting problem due to the lack of sufficient training samples, age variations, head poses, identity bias, and illumination variations. The proposed method focuses on these issues and overcomes the computational complexity of the proposed system.

3.1. Face Preprocessing

We have implemented a deep learning framework to recognize discrete human facial expression categories in this proposed work. The input face image has been resized to the same size and was normalized to a fixed size face image $\mathcal{I}_{m\times n\times 3}$. Here, these input images are mapped to the same locations, i.e., eye locations, the tip of the nose, etc., are known

as a feature map. At the lowest level of abstraction, it is assumed that preprocessing is a standard term that concerns computing over intensity images. These input and output intensity images are the same as the original data captured by the sensor. A matrix of image function values usually represents an intensity image. The goal of preprocessing is to enhance the expression of the region of interest and to suppress the unwanted, redundant, and inconsistent noises in the image. Image preprocessing methods are classified into four categories according to the size of the pixel neighborhood that is used for the calculation of new pixel brightness: pixel brightness transformations; geometric transformations; certain preprocessing methods that use a local neighborhood of the processed pixel; and image restoration that requires knowledge about the entire image. Here, the required face region is detected from the input image using a tree-structured part model [46]. The detected face has been resized to a fixed image $\mathcal{F}_{n\times n}$. These face images are used as input to the proposed CNN models. During preprocessing, we extracted the face region from each input image $\mathcal{I}_{m\times n}$. Since the facial expressions contained very minute details, it is important to be conscious about analyzing both expressive or non-expressive characteristics of the facial region. During face preprocessing, we applied the tree-structured part model, which works better for both frontal and profile face regions compared to Haar-like features [47]. This model has outstanding performance results compared to the other face detection algorithm in computer vision. The tree-structured part model works on the principle of a mixture of trees with a global mixture of topological viewpoints changing. For an unconstrained image with an unknown face region, this model locates all the facial landmarks in $\mathcal{I}_{m\times n}$. For facial landmark localization, we consider $\mathcal{L}_q^p = (\mathbf{x}_q^p, \mathbf{y}_q^p)$ as the coordinate for the pixel location of part q. Hence, the tree-structured part model computes thirty-nine landmark points for profile faces, while computing sixty-eight landmark points for the frontal faces. These landmark points undergo the computation of four corner points of the face region $\mathcal{F}_{n\times n}$. The face preprocessing steps are shown in Figure 3.

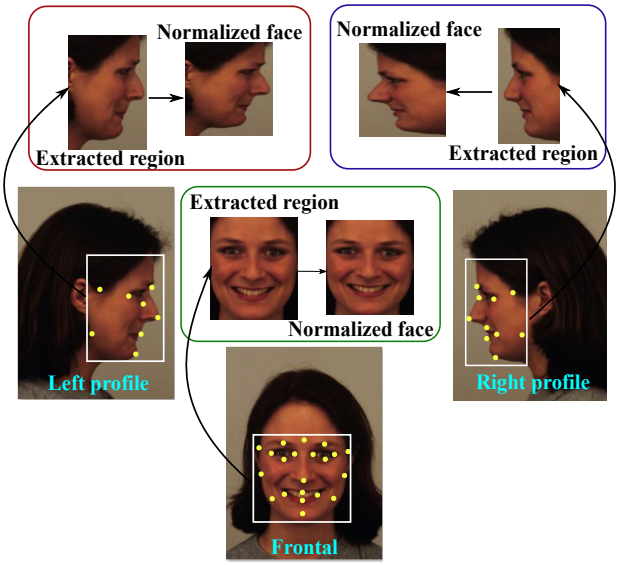

Figure 3. Face pre-processing steps in the proposed FERS.

3.2. Feature Representation for Expression Classification

Feature extraction is a crucial task to extract discriminating features from the input image $\mathcal{F}_{n\times n}$ to ensure that the extracted feature contains more distinctive patterns [48]. Here, the input image may be a grayscaled or RGB color image. In the field of computer vision and in image processing research areas, feature extraction starts from an initial set of measured data and builds the features which are supposed to be informative and non-redundant, facilitating the subsequent learning and generalization steps. In many cases, the texture feature extraction techniques [49] lead to better human interpretations. Moreover, it is related to dimensionality reductions, i.e., when the input image size is

too large to be processed as the representation for that image, it is transformed into a reduced set of features, also called a feature vector. The modern state-of-the-art technique for generic CNN feature representations for facial expression recognition problems could compete with the statistical and structural-based methods of computer vision. These generic CNN features can cope with the articulation and occlusion face images captured in an unconstrained environment and can achieve better performance.

The proposed method describes a complex CNN baseline model with two components, i.e., feature extraction and classification parts. The proposed CNN model uses a convolutional neural network architecture with five to seven deep image perturbation layers. The model performs convolution operations using the ReLU activation function followed by max-pooling and batch normalization operations for feature extraction. Finally, two flatten layers, which are fully connected, are used for classification tasks on the extracted feature maps from the top of the layers. The performance for the proposed CNN has been increased through adding new levels by applying image augmentation and progressive image resizing methods. These also help the model to prevent the over-fitting and imbalanced data problem. Progressive image resizing methods support the model to avoid the use of excessive computational power. Considering only the pre-trained weights of the last few layers are being used, these weights have to be learned properly. We take advantage of image augmentation, batch normalization, the activation function, and regularization methods including the mix-up optimizer and label smoothing techniques. The convolution operation is the primary operator and main building block of a CNN architecture. The term convolution is a mathematical operation that combines two functions and generates a third function. Here, it is used to extract features from the images. In the case of a CNN model, the convolution operation is executed over the input image with the help of a $t \times t$ sized kernel or filter and then generates feature maps. The convolution operation is performed by sliding the filter followed by non-linearity over the input. At every location, matrix multiplication is performed and sums the result onto the feature map. Finally, we used a fine-tuning of the parameters and fusion methods to enhance the performance of the proposed recognition system. Thus, the descriptions of the employed layers for the proposed CNN architecture are as follows:

- *Convolution:-* The Convolution layer is the core building block of a CNN model that performs most of the computation operations. Convolution is a linear matrix operation consisting of some set of kernels or filters $W_{t\times t}$. The kernel is a small-sized matrix of weights that slide over the input [50] and performs element-wise matrix multiplication. The convolution operation essentially performs dot products between some sets of learnable filters $W_{t\times t}$ and local regions of the input image $\mathcal{F}_{n\times n}$, and produces an output matrix of dimension $n' \times n'$. Here, n' is calculated by $n' = \frac{n-t+2\times\mathcal{P}}{\mathcal{S}} + 1$, where $\mathcal{S}$ is the stride that governs how many numbers of cells will be moved by the filter to the right and down, from the top-left corner to the bottom-right corner, in the input image to calculate the next cell in the result. Additionally, $\mathcal{P}$ is the padding that shrinks the height and width of the volumes. Mathematically, the formulation of the convolution operation is denoted as follows [51]: for input feature vector $\mathrm{F} = \mathbf{f}(v)$ and a filter vector $\mathrm{W} = \mathbf{w}(v)$, the convolution operation is obtained as $\mathbf{F} \star \mathbf{W} = \sum_{u\in\mathbf{U}} \mathbf{f}(u)\mathbf{w}(v-u) = \langle \mathbf{f}(u), \mathbf{w}(v-u)\rangle$, where the operator $\star$ denotes the convolution operation and $\langle .,. \rangle$ represents the sliding vector inner product between the input feature $\mathbf{f}(u)$ and the flipped kernel $\mathbf{w}(v-u)$. It measures the similarity between the two vectors. The primary benefits of the convolution operation are: (i) parameter or weight sharing, as a feature detector is used in one part and transfers into other parts of the image; (ii) the fact that it reduces the number of effective parameters and image translation; and (iii) the sparsity of connections, i.e., the hidden layers' input and output dependencies.
- *Max-pooling:-* A pooling operation is a mathematical operation that performs pixel-wise average or median operations to reduce the input image size by half its size. The effective advantages of using pooling operations concern a means of removing

noise, correcting images, and overcoming incidental occlusions [52]. The pooling layer is used to reduce the size of the representation to speed up the process as well as to make some of the features it detects more robust. There are different types of pooling operations, such as average pooling, fractional max-pooling, and max-pooling. Max-pooling is a commonly used pooling operation that is used in most CNN models. Max-pooling calculates the maximum value for patches of a feature map and uses it to create a down-sampled feature map. It is usually used after a convolutional layer. The primary benefits of max-pooling are as follows: (i) it is a translation invariance, i.e., it translates the image by a small amount that does not significantly affect the values of most pooled outputs; (ii) has reduced computational costs; (iii) has faster matching; and (iv) has improved accuracy.

- *Fully Connected Layers:-* It has been stated that fully connected layers and convolutional layers are distinct, but it has been observed that fully connected layers are a special case of convolutional layers [53]. In our proposed CNN model, we used two fully connected layers denoted as $\mathcal{FC}_1$ and $\mathcal{FC}_2$. Here, n_2 neurons in $\mathcal{FC}_2$ have full connections to all activation n_1 in $\mathcal{FC}_1$. The activation function can be computed with a matrix multiplication followed by a bias offset. Let $x \in \mathbf{R}^{n_1 \times 1}$ represent the single output vector of layer $\mathcal{FC}_1$ and let $\mathbf{W} \in \mathbf{R}^{n_1 \times n_2}$ denote the weight matrix of the $\mathcal{FC}_2$. Suppose w_i is the weight vector of the corresponding i_{th} neuron of the column vector of $\mathbf{W}$ in layer $\mathcal{FC}_2$ [54]. Then, the output of $\mathcal{FC}_2$ is obtained by $\mathbf{W}^{\mathcal{T}} \times x$. The output of fully connected layers is independent of the input image size. Fully connected layers of a CNN architecture will reduce the full image size, compute the single vector of class scores, and produce a resulting vector of size $[1 \times 1 \times \mathcal{C}_i]$.
- *Dense Layers:-* The dense layer is a type of fully connected connection layer in deep neural networks [55]. In a dense layer, all input layers are connected to the output layers by a weight. It performs linear operations with $\mathcal{X}_{inputs}$ parameters and generates $\mathcal{X}_{output}$ parameters [56] that are also connected to the next layer as inputs. It utilizes dense connections between layers with matching feature map size $\mathcal{X}'_l = g'\left(\mathbf{W}^T \mathcal{X}'_{l-1}\right)$, where g' is the activation function, e.g., ReLU defined as $p(x) = max(0, x)$.
- *Batch Normalization:-* Batch is used to normalize the inputs of the previous layers at each batch, maintaining the values in a comparable range with the mean equal to 0 and the standard deviation equal to 1. This helps the CNN model to prevent skews at any one particular point and increases the computation speed. We applied the batch normalization after every convolution layer and then passed these values to the ReLU activation function. Batch normalization acts as a regularizer and allows the model to use higher learning rates [57]. It is used in various image classification problems and achieves higher accuracy with fewer training steps. Batch normalization also has a beneficial effect on the gradient flow through the network by reducing the dependence of gradients on the scale of their parameters or initial values. It also regularizes the model and reduces the need for dropout layers. We calculated the batch normalization mathematically as follows: For a a mini-batch χ of size m and with values of $x^{(l)}$, i.e., activation and omit l for clarity, the mini-batch is expressed as $\chi = (x_{1...m})$. θ and ψ are the learning parameters, $(\hat{x}_{1....m})$ are normalized values, and $y_{1....m}$ are their corresponding linear transformations denoted by batch normalizing transform, i.e., $\mathcal{BN}_{\theta,\psi} : x_{1...m} \rightarrow y_{1...m}$. Thus, consider the following: mini-batch mean, $\preceq_{\succ} = \frac{1}{m}\sum_{i=1}^{m} x_i$; mini-batch variance, $\supset_{\succ}^{\in} = \frac{1}{m}\sum_{i=1}^{m}(x_i - \mu_\chi)^2$; normalization, $\hat{x}_i = \frac{x_i - \mu_\chi}{\sqrt{\supset_{\succ}^{\in} + \zeta}}$; and scale and shift, $y_i \leftarrow \theta\hat{x}_i + \psi = \mathcal{BN}_{\theta,\psi}(x_i)$.
- *Regularization:-* Regularization strategies are designed to reduce the test error of a machine learning algorithm, possibly at the expense of the training error [58]. The popular regularization methods that exist in the field of deep learning [59] are dropout, R1-regularization, and discriminative regularization, among others. We employed the dropout regularization technique on the penultimate layer $\alpha = [\vec{a}_1, \vec{a}_2, ..., \vec{a}_{\mathrm{F}}]$ (F are the numbers of filters) for our proposed deep CNN model with constrain: ℓ_2

norms of the weight vector [60]. The dropout regularization technique drops a unit during the training time with a specified probability. Dropout prevents co-adaptation of the network's hidden units by randomly dropping out a portion or setting the hidden units to zero during forward and backward propagation. The neural network becomes too reliant on particular connections. Instead of using $\gamma = \omega \times \alpha + \delta$ for output hidden unit γ in forward propagation, here, dropout uses $\gamma = \omega \times (\alpha \otimes \beta) + \delta$, where the operator '$\otimes$' performs element-wise matrix multiplication and $\beta \in \mathbf{R}^{\mathrm{F}}$ is the masking vector of the Bernoulli random variable. At test time, all units are present and the learned weight vectors are scaled by $\mathcal{P}$ such that $\vec{\omega} = \mathcal{P} \times \omega$, where $\vec{\omega}$ represents the class score computed without dropout. The advantage of using dropout is that it prevents artificial neural networks from over-fitting. Intuitively, dropout can be thought of as creating an implicit ensemble of neural networks. This means that a selected subset of units for each training sample, including their incoming and outgoing connections, are temporarily removed from the network. Suppose a dropout probability of 0.5 is used; in this case, roughly half of the activation in each layer is deleted for every training sample, thus preventing hidden units from relying on other hidden units present.

- *Optimisation:-* The proposed FERS problem has been solved by stochastic optimization methods to optimize our CNN models. In this study, we used the popular first-order gradient-based Adam optimizer of the stochastic objective function. The popular optimization methods used for solving FERS problems are Adagard, SGD, RMSProp, SGD with momentum, AggMo, Demon, Demon CM, DFA, and Adadelta optimization methods. They use their stochastic mini-bath method. This method estimates the learning rate based on lower-order momentum. Adam [61] uses only the first two moments of gradient $\widetilde{v}_t$ and the learning rate or steps size η. The weight updates for the Adam optimizer are mathematically calculated as $w_t = w_{t-1} - \eta \frac{\hat{h}}{\sqrt{\widetilde{v}_t - \epsilon}}$, where ϵ is a smaller number. The primary advantages of using the Adam optimizer are that it works well and is suitable for problem-solving for large training data sets. Adam can handle non-stationary objective functions as in RMSProp while overcoming the sparse gradient issue drawbacks that appear in RMSProp. Adam is favorable compared to other stochastic optimizers. The implementation of Adam is straightforward and computationally efficient with less memory required.

The proposed CNN architectures are based on several blocks as discussed in the previous section. Here, an input image $\mathcal{F}_{n_H \times n_W}$ is convolved with a set of kernels of size $t \times t$. These convolution layers are called feature maps. The feature maps are stacked to provide multiple filters on the input. We used 3×3 sized filters with a stride of 1 for each convolution layer. The activation function for each convolution layer was ReLU. The computational complexity of the CNN models was reduced by using $d \times d$ pooling layers, which reduces the output size from one layer to the next in the hidden network layers. To select maximum elements, we used 2×2 max-pooling operations to preserve the important features [62]. Hence, these layers reduce the size of the input image by half. To feed the pooled output from the stacked featured map to the final layer, the maps were flattened into one column. The final layers of the CNN had two fully connected layers with M number of nodes each. Fully connected layers also used the ReLU activation function. These two layers were regularized by using the dropout layers with the regularization technique. Finally, the Softmax layer was employed, followed by two fully connected layers, and the number of nodes of this layer was equal to the number of expression classes.

During the feature representation of images using deep learning approaches, it was observed that the CNN models obtained better representation when patterns were analyzed from the multi-resolution of images. Additionally, increasing some layers in the architecture while increasing the resolution of the images results in more deeply analyzing some hidden patterns in the feature maps. Inspired by these observations, we applied multi-resolution of the facial images with varying layers in different CNN architectures. During feature representations, we considered facial images with three different resolutions such

that the original facial image $\mathcal{F}_{n\times n}$ was down-sampled to $\mathcal{F}_{n_1\times n_1}$, $\mathcal{F}_{n_2\times n_2}$, and $\mathcal{F}_{n_3\times n_3}$, $n_3 = 2\times n_2 = 4\times n_1$. Here, for facial images, namely $\mathcal{F}_{n_1\times n_1}$, $\mathcal{F}_{n_2\times n_2}$, and $\mathcal{F}_{n_3\times n_3}$, three different CNN architectures, namely CNN_1, CNN_2, and CNN_3, were proposed. These architectures are shown in Figures 4–6, whereas the detailed descriptions of these architectures, including the employed input–output hidden layers, the output shapes of the convoluted images, and the input image sizes and parameters generated at each layer, are shown in Tables 1–3, respectively, to allow for greater understanding and clarity about the models.

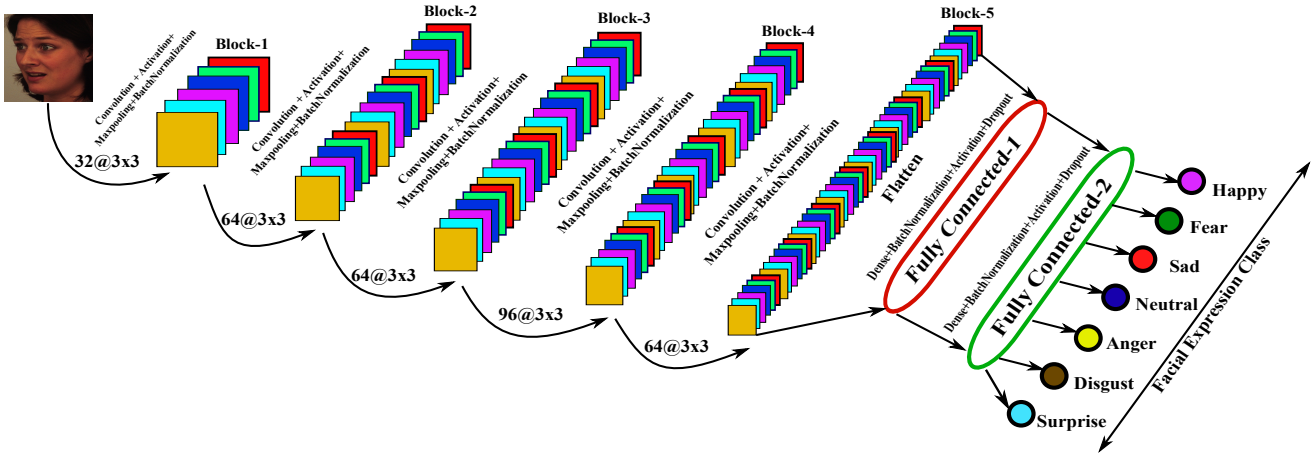

Figure 4. CNN_1 architecture for the proposed FER system, that takes the input image $\mathcal{F}$ of size $(n_1 \times n_1 \times 3)$.

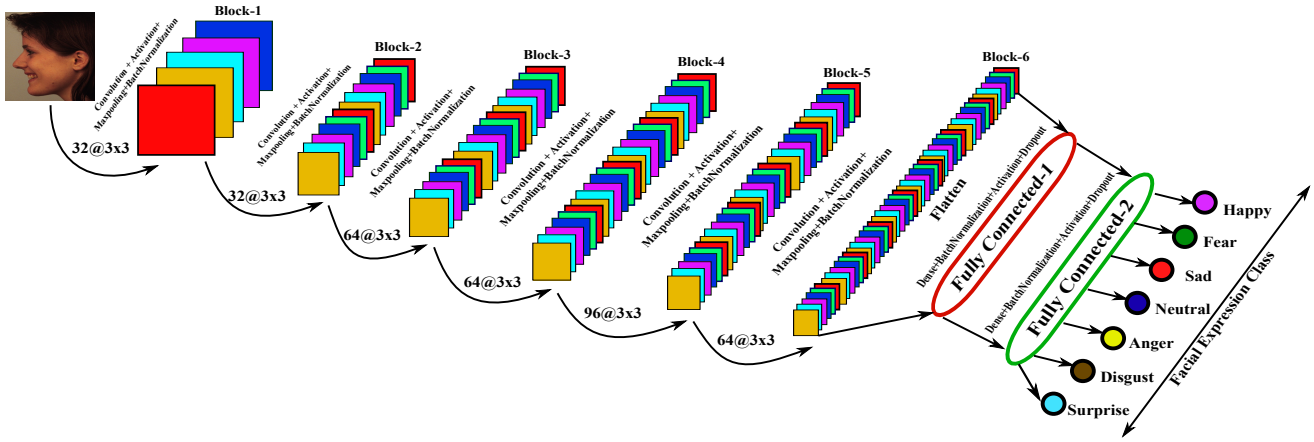

Figure 5. CNN_2 architecture for the proposed FER system, that takes the input image $\mathcal{F}$ of size $(n_2 \times n_2 \times 3)$.

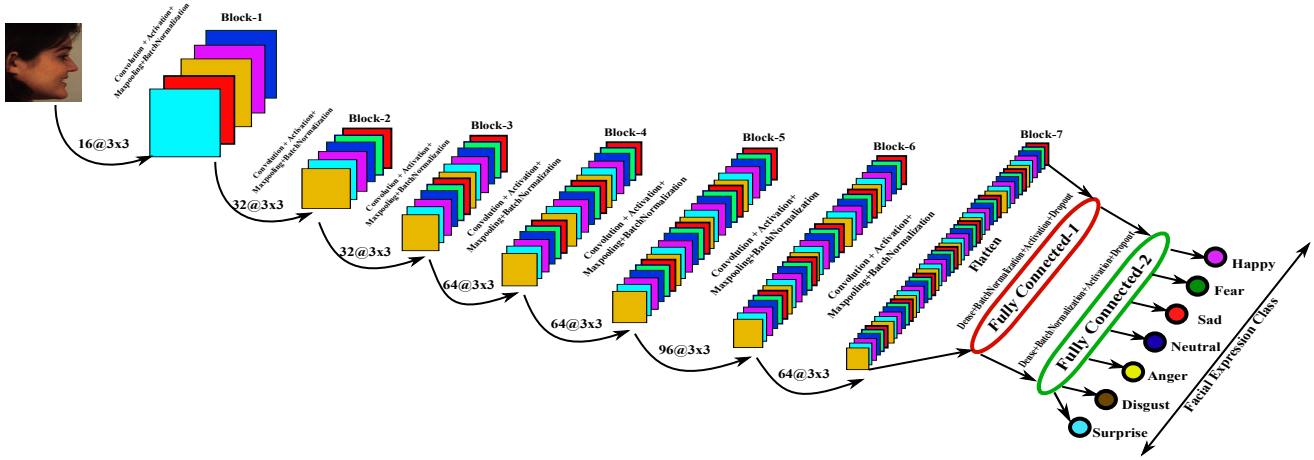

Figure 6. CNN_3 architecture for the proposed FER system, that takes the input image $\mathcal{F}$ of size $(n_3 \times n_3 \times 3)$.

Table 1. The proposed CNN architecture for the input size 48 × 48 with layers, output shape, image size, and parameters.

Layer	Output Shape	Image Size	Parameters	Layers	Output Shape	Image Size	Parameters
Block-1				**Block-3**			
Conv2D (3 × 3)@32	$(n, n, 32)$	(48, 48, 32)	((3 × 3 × 3) + 1) × 32 = 896	Conv2D (3 × 3)@96	$(n_2, n_2, 96)$	(12,12,96)	((3 × 3 × 64) + 1) × 96 = 55,392
Batch Norm	$(n, n, 32)$	(48, 48, 32)	4 × 32 = 128	Batch Norm	$(n_2, n_2, 96)$	(12,12,96)	4 × 96 = 384
Activation ReLU	$(n, n, 32)$	(48, 48, 32)	0	Activation ReLU	$(n_2, n_2, 96)$	(12,12,96)	0
Maxpool2D (2 × 2)	$(n_1, n_1, 32)$ $n_1 = n/2$	(24, 24, 32)	0	Maxpool2D (2 × 2)	$(n_3, n_3, 96)$ $n_3 = n_2/2$	(6,6,96)	0
Dropout	$(n_1, n_1, 32)$	(24, 24, 32)	0	Dropout	$(n_3, n_3, 96)$	(6,6,96)	0
Block-2				**Block-4**			
Conv2D (3 × 3)@64	$(n_1, n_1, 64)$	(24, 24, 64)	((3 × 3 × 32) + 1) × 64 = 18,496	Conv2D (3 × 3)@96	$(n_3, n_3, 96)$	(6,6,96)	((3 × 3 × 96) + 1) × 96 = 83,040
Batch Norm	$(n_1, n_1, 64)$	(24, 24, 64)	4 × 64 = 256	Batch Norm	$(n_3, n_3, 96)$	(6,6,96)	4 × 96 = 384
Activation ReLU	$(n_1, n_1, 32)$	(24, 24, 64)	0	Activation ReLU	$(n_3, n_3, 96)$	(6,6,96)	0
Maxpool2D (2 × 2)	$(n_2, n_2, 64)$ $n_2 = n_1/2$	(12, 12, 64)	0	Maxpool2D (2 × 2)	$(n_4, n_4, 96)$ $n_4 = n_3/2$	(3,3,96)	0
Dropout	$(n_2, n_2, 64)$	(12, 12, 64)	0	Dropout	$(n_4, n_4, 64)$	(3,3,96)	0
Block-5							
Conv2D(3 × 3)@64	$(n_4, n_4, 64)$	(3,3,64)	55,360				
BatchNorm	$(n_4, n_4, 64)$	(3,3,64)	256				
ActivationReLU	$(n_4, n_4, 64)$	(3,3,64)	0				
Maxpool2D(2 × 2)	$(n_5, n_5, 64)$, $n_5 = n_4/2$	(1,1,64)	0				
Dropout	$(n_5, n_5, 64)$	(1,1,64)	0				
Layer	**Output Shape**	**Image Size**	**Parameter**				
Flatten	(1, $n_5 \times n_5 \times 64$)	(1, 64)	0				
Dense	(1, 256)	(1, 256)	(1 + 64) × 256 = 16,640				
Batch Normalization	(1, 256)	(1, 256)	1024				
Activation Relu	(1, 256)	(1, 256)	0				
Dropout	(1, 256)	(1, 256)	0				
Dense	(1, 256)	(1, 256)	(256 + 1) × 256 = 65,792				
Batch Normalization	(1, 256)	(1, 256)	1024				
Activation Relu	(1, 256)	(1, 256)	0				
Dropout	(1, 256)	(1, 256)	0				
Dense	(1, 7)	(1, 7)	(256 + 1) × 7 = 1799				
Total parameters for the input image size:			300,871				
Total number of trainable parameters:			299,143				
Non-trainable parameters:			1728				

Table 2. The proposed CNN architecture for the input size 96 × 96 with layers, output shape, image size, and parameters.

Layer	OutputShape	ImageSize	Parameters
Block-1			
Conv2D(3 × 3)@32	$(n, n, 32)$	(96, 96, 32)	896
BatchNorm	$(n, n, 32)$	(96, 96, 32)	128
ActivationReLU	$(n, n, 32)$	(96, 96, 32)	0
Maxpool2D(2 × 2)	$(n_1, n_1, 32)$, $n_1 = n/2$	(48, 48, 32)	0
Dropout	$(n_1, n_1, 32)$, $n_1 = n/2$	(48, 48, 32)	0

Block-2				**Block-4**			
Conv2D (3 × 3)@32	$(n_1, n_1, 32)$	(48,48,32)	((3 × 3 × 3) + 1) × 32 = 9248	Conv2D (3 × 3)@96	$(n_3, n_3, 96)$	(12,12,96)	((3 × 3 × 64) + 1) × 96 = 55,392
Batch Norm	$(n_1, n_1, 32)$	(48,48,32)	4 × 32 = 128	Batch Norm	$(n_3, n_3, 96)$	(12,12,96)	4 × 96 = 384
Activation ReLU	$(n_1, n_1, 32)$	(48, 48, 32)	0	Activation ReLU	$(n_3, n_3, 96)$	(12,12,96)	0
Maxpool2D (2 × 2)	$(n_2, n_2, 32)$ $n_2 = n_1/2$	(24, 24, 32)	0	Maxpool2D (2 × 2)	$(n_4, n_4, 96)$ $n_4 = n_3/2$	(6,6,96)	0
Dropout	$(n_2, n_2, 32)$	(24, 24, 32)	0	Dropout	$(n_4, n_4, 96)$	(6,6,96)	0

Block-3				**Block-5**			
Conv2D (3 × 3)@64	$(n_2, n_2, 64)$	(24, 24, 64)	((3 × 3 × 32) + 1) × 64 = 18,496	Conv2D (3 × 3)@96	$(n_4, n_4, 96)$	(6,6,96)	((3 × 3 × 96) + 1) × 96 = 83,040
Batch Norm	$(n_2, n_2, 64)$	(24, 24, 64)	4 × 64 = 256	Batch Norm	$(n_4, n_4, 96)$	(6,6,96)	4 × 96 = 384
Activation ReLU	$(n_2, n_2, 32)$	(24, 24, 64)	0	Activation ReLU	$(n_4, n_4, 96)$	(6,6,96)	0
Maxpool2D (2 × 2)	$(n_3, n_3, 64)$ $n_3 = n_2/2$	(12, 12, 64)	0	Maxpool2D (2 × 2)	$(n_5, n_5, 96)$ $n_5 = n_4/2$	(3,3,96)	0
Dropout	$(n_3, n_3, 64)$	(12, 12, 64)	0	Dropout	$(n_5, n_5, 64)$	(3,3,96)	0

Block-6			
Conv2D(3 × 3)@64	$(n_5, n_5, 64)$	(3,3,64)	55,360
BatchNorm	$(n_5, n_5, 64)$	(3,3,64)	256
ActivationReLU~	$(n_5, n_5, 64)$	(3,3,64)	0
Maxpool2D(2x2)	$(n_6, n_6, 64)$, $n_6 = n_5/2$	(1,1,64)	0
Dropout	$(n_6, n_6, 64)$	(1,1,64)	0
Layer	**Output Shape**	**Image Size**	**Parameter**
Flatten	(1, $n_6 \times n_6 \times 64$)	(1, 64)	0
Dense	(1, 256)	(1, 256)	(1 + 64) × 256 = 16,640
Batch Normalization	(1, 256)	(1, 256)	1024
Activation Relu	(1, 256)	(1, 256)	0
Dropout	(1, 256)	(1, 256)	0
Dense	(1, 256)	(1, 256)	(256 + 1) × 256 = 65,792
Batch Normalization	(1, 256)	(1, 256)	1024
Activation Relu	(1, 256)	(1, 256)	0
Dropout	(1, 256)	(1, 256)	0
Dense	(1, 7)	(1, 7)	(256 + 1) × 7 = 1799
Total parameters for the input image size:			310,247
Total number of trainable parameters:			308,455
Non-trainable parameters:			1792

Table 3. The proposed CNN architecture for the input size 192 × 192 with layers, output shape, image size, and parameters.

Layers	Output Shape	Image Size	Parameters
Block-1			
Conv2D(3 × 3)@16	$(n, n, 16)$	(192, 192, 16)	448
BatchNorm	$(n, n, 16)$	(192, 192, 16)	64
ActivationReLU	$(n, n, 16)$	(192, 192, 16)	0
Maxpool2D(2 × 2)	$(n_1, n_1, 16)$, $n_1 = n/2$	(96, 96, 16)	0
Dropout	$(n_1, n_1, 16)$	(96, 96, 16)	0
Block-2			
Conv2D(3 × 3)@32	$(n_1, n_1, 32)$	(96, 96, 32)	4640
BatchNorm	$(n_1, n_1, 32)$	(96, 96, 32)	128
ActivationReLU	$(n_1, n_1, 32)$	(96, 96, 32)	0
Maxpool2D(2 × 2)	$(n_2, n_2, 32)$, $n_2 = n_1/2$	(48, 48, 32)	0
Dropout	$(n_2, n_2, 32)$	(48, 48, 32)	0

Layers	Output Shape	Image Size	Parameters	Layers	Output Shape	Image Size	Parameters
Block-3				**Block-5**			
Conv2D (3 × 3)@32	$(n_2, n_2, 32)$	(48,48,32)	((3 × 3 × 3) + 1) × 32 = 9248	Conv2D (3 × 3)@96	$(n_4, n_4, 96)$	(12,12,96)	((3 × 3 × 64) + 1) × 96 = 55,392
Batch Norm	$(n_2, n_2, 32)$	(48,48,32)	4 × 32 = 128	Batch Norm	$(n_4, n_4, 96)$	(12,12,96)	4 × 96=384
Activation ReLU	$(n_2, n_2, 32)$	(48,48,32)	0	Activation ReLU	$(n_4, n_4, 96)$	(12,12,96)	0
Maxpool2D (2 × 2)	$(n_3, n_3, 32)$ $n_3 = n_2/2$	(24, 24, 32)	0	Maxpool2D (2 × 2)	$(n_5, n_5, 96)$ $n_5 = n_4/2$	(6,6,96)	0
Dropout	$(n_3, n_3, 32)$	(24, 24, 32)	0	Dropout	$(n_5, n_5, 96)$	(6,6,96)	0
Block-4				**Block-6**			
Conv2D (3 × 3)@64	$(n_3, n_3, 64)$	(24, 24, 64)	((3 × 3 × 32) + 1) × 64 = 18,496	Conv2D (3 × 3)@96	$(n_5, n_5, 96)$	(6,6,96)	((3 × 3 × 96) + 1) × 96 = 83,040
Batch Norm	$(n_3, n_3, 64)$	(24, 24, 64)	4 × 64 = 256	Batch Norm	$(n_5, n_5, 96)$	(6,6,96)	4 × 96 = 384
Activation ReLU	$(n_3, n_3, 32)$	(24, 24, 64)	0	Activation ReLU	$(n_5, n_5, 96)$	(6,6,96)	0
Maxpool2D (2 × 2)	$(n_4, n_4, 64)$ $n_4 = n_3/2$	(12, 12, 64)	0	Maxpool2D (2 × 2)	$(n_6, n_6, 96)$ $n_6 = n_5/2$	(3,3,96)	0
Dropout	$(n_4, n_4, 64)$	(12, 12, 64)	0	Dropout	$(n_6, n_6, 64)$	(3,3,96)	0
Block-7							
Conv2D(3 × 3)@64	$(n_6, n_6, 64)$	(3,3,64)	55,360				
BatchNorm	$(n_6, n_6, 64)$	(3,3,64)	256				
ActivationReLU~	$(n_6, n_6, 64)$	(3,3,64)	0				
Maxpool2D(2 × 2)	$(n_7, n_7, 64)$, $n_7 = n_6/2$	(1,1,64)	0				
Dropout	$(n_7, n_7, 64)$	(1,1,64)	0				
Layer	**Output Shape**	**Image Size**	**Parameter**				
Flatten	(1, $n_7 \times n_7 \times 64$)	(1, 64)	0				
Dense	(1, 256)	(1, 256)	(1 + 64) × 256 = 16,640				
Batch Normalization	(1, 256)	(1, 256)	1024				
Activation Relu	(1, 256)	(1, 256)	0				
Dropout	(1, 256)	(1, 256)	0				
Dense	(1, 256)	(1, 256)	(256 + 1) × 256 = 65,792				
Batch Normalization	(1, 256)	(1, 256)	1024				
Activation Relu	(1, 256)	(1, 256)	0				
Dropout	(1, 256)	(1, 256)	0				
Dense	(1, 7)	(1, 7)	(256 + 1) × 7 = 1799				
Total parameters for the input image size:			314,503				
Total number of trainable parameters:			312,679				
Non-trainable parameters:			1824				

3.3. Factors Affecting the Performance of the Proposed FERS

- *Data Augmentation:-* The data augmentation technique is used to expand the training samples in order to improve the performance of recognition and the ability to generalize the models. In machine learning, image augmentation techniques artificially increase the amount of training data by applying transformation methods to the existing data [63]. The classical augmentation techniques that were employed are bilateral filtering, unsharp filtering, horizontal flip, vertical flip, Gaussian blur,

additive Gaussian noise, image scale, image cropping, translation, image rotation, shear mapping, image zooming, image filling, and contrast normalization methods from [15] for the purpose of image augmentation. The whole training images were flipped horizontally by applying simple image data augmentation techniques. In this work, we applied these techniques for each resolution of the images.

- *Fine Tuning:-* Fine-tuning allows for higher-order feature representations in the base model to make them more relevant for the face recognition tasks. For example, VGG used many layers and generated a higher dimensional feature vector, and thus thw inference was quite costly at run-time due to huge parameters. In this case, fine-tuning techniques were applied when freezing some layers and the number of parameters, and the model was retrained to reduce computational overheads.
- *Progressive Resizing:-* Progressive image resizing is an eminent technique that sequentially resizes all images while training the CNN models on smaller, i.e., tinier images to larger image sizes. The progressive resizing technique is used to train a CNN with $n \times n$ image size, saving the weights, and then the CNN is retrained again for other iterations with the images of increased sizes greater than n. This technique was used for super-resolution [64], where low-resolution images gradually increased to the image with a higher resolution during training processes. The advantages of using progressive resizing are that it improves generalization and reduces overfitting problems.
- *Transfer Learning:-* The principle concept behind transfer learning for facial expression recognition and classification problems is that a model trained on large data sets for one problem is effectively used as a generic model in some way on other related problems. The model that has been trained earlier is known as the pre-trained model. Our proposed deep learning convolution neural network model uses a transfer learning technique in which the weights of the pre-trained model and/or a set of layers from the pre-trained model CNN_1 are used for the new model CNN_2 to solve similar problems. Similarly, the weights of CNN_2 have been adopted to solve the CNN_3 model. The benefits of using transfer learning are that it reduces the training time and can result in lower generalization errors.
- *Scores Fusion:-* In the proposed system, three CNN architectures have been proposed. These architectures take images of different sizes as inputs. Thus, during the recognition of facial expressions on the test sample F, there are three different classification score vectors, namely $s_1 = (a_1^1, a_2^1, \cdots, a_7^1)$, $s_2 = (a_1^2, a_2^2, \cdots, a_7^2)$, and $s_3 = (a_1^3, a_2^3, \cdots, a_7^3)$, where each a_j^i is the classification score by the CNN_i architecture and for j^{th} expression class. These classification scores are fused together using score-level post-classification fusion approaches [14] to increase the performance of the recognition system. In this work, two score-level fusion techniques, namely *Sum-rule* and *Product-rule*, were employed. The *Sum-rule* and *Product-rule* techniques are defined as follows:

$$\max_{i \neq j,\ k=\{1,\cdots,7\}} \{a_k^i + a_k^j\} \tag{1}$$

$$\max_{i \neq j,\ k=\{1,\cdots,7\}} \{a_k^i * a_k^j\} \tag{2}$$

4. Experimentation

In this section, the experimentation of the proposed FER system is discussed and for this purpose, four challenging benchmark facial expression databases were experimented on. Each database was randomly divided into 50% of the dataset for training, while the remaining 50% was used for testing purposes. Finally, this partitioning of the datasets were done ten times and the average performance was reported, corresponding to each database. As there were no particular benchmark datasets specifically built for the healthcare scenario,

social IoT, emotion AI, and cognitive AI, the employed datasets were assumed as backbones for e-Healthcare, social IoT, emotion AI, and cognitive-AI diversified applications as discussed in this paper. The proposed system has not been tested in a real-time scenario. Still, the employed datasets were very challenging. The proposed method can accept and handle all the unconstrained situations of facial expression recognition in the real-time strategy for e-Healthcare, social IoT, emotion AI, and cognitive AI applications.

4.1. Database Used

The first employed database was Karolinska-directed emotional faces (KDEF) [42] which contains seventy different subjects (thirty-five male and thirty-five female) with five different pose variations labeled with seven basic expression categories. Here, we used only 1210 samples as training sets, whereas 1213 samples were used as testing sets, as only these samples were available from the license agreement downloaded site. Figure 7 shows some examples from this database.

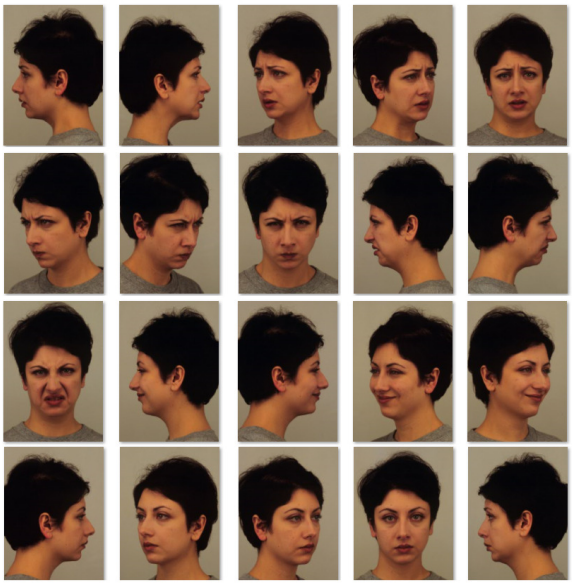

Figure 7. Some image samples from the KDEF database.

The second employed database is the GENKI database [65], which is composed of 4000 facial images that have been labeled as two classes: (i) happy and (ii) non-happy. Additionally, for this database, two thousand images were randomly selected as training sets, while the remaining two thousand images were considered for the testing set. Some examples of this database are shown in Figure 8.

The Extended Cohn-Kanade (CK+) [66] is our third database, which is composed of 593 video sequences from 123 subjects captured between the ages of 18 to 50 years. Here, only 309 image sequences were labeled with six basic expressions. During the experimentation, we randomly split this database into the training and testing sets. Figure 9 presents some image samples from the CK+ database.

Figure 8. Some image samples from the GENKI database.

Figure 9. Some image samples from the CK+ database.

Our fourth database was Static Facial Expressions in the Wild (SFEW) [26], which was created from the AFEW video database by selecting the keyframes based on facial point clustering. The challenging SFEW dataset contains 700 images which were divided into the training set (346 images) and testing set (354 images). This database has seven facial expression classes, namely afraid, anger, disgust, happiness, neutral, sadness, and surprise. Figure 10 presents image samples from the SFEW database.

Figure 10. Some image samples from the SFEW database.

Table 4 presents the detailed descriptions of the KDEF, GENKI, CK+, and SFEW facial expression databases. The objectives of the selection of these databases were (i) to obtain expressions that were common and generic in people; (ii) to ensure that the other expressions used in the affecting computing research areas were composed of a mixture of these basic facial expressions; and (iii) to ensure that the good recognition system for these expressions would be very beneficial for several real-world applications, such as in e-Healthcare frameworks,in the social Internet of Things (IoT), and in emotion AI in business organizations.

Table 4. Presentation of the employed facial expression databases.

Database	Class	Training	Testing
KDEF	7	1210	1213
GENKI	2	2000	2000
CK+	6	663	146
SFEW	7	346	354

4.2. Results and Discussion

This section describes and explains the experimentation of the proposed facial expression recognition system (FERS). The proposed FERS was implemented using Python 3.7.9 version, Tensorflow 2.3.1 version, Keras 2.4.3 version, CUDA version 11.2, and NVIDIA-SMI 460.79 Driver Version in Windows 10 Pro 64-bit, Intel(R) Core(TM)-i7-9700 CPU, 3.30 GHz(8 CPU) Processor, and in a 8 GB NVIDIA GeForce RTX 2070 SUPER XLA GPU device with 16 GB RAM. During experimentation, we employed both gray-scaled and RGB-colored images as some databases have RGB images while others only have gray-scaled images. During image preprocessing, from each input image $\mathcal{I}$, we detected the face region by applying the methods discussed in Section 3.1. Furthermore, the detected face region $\mathcal{F}$ was normalized to a fixed size image $\mathcal{F} \in \mathbb{R}^{200\times200}$. For recognizing the expression classes on the human face, in this work, we employed deep learning-based approaches where three convolutional neural network (CNN) architectures (Figures 4–6) were designed. These CNN architectures were trained in such a way that they would perform both feature computation and expression classification tasks. For better understanding the functionality of these architectures, at first, we started the experiment using CNN_1 architecture (Figure 4), where the input to this system is an image $\mathcal{F} \in \mathbb{R}^{n_1\times n_1}$, $n_1 = 48$, i.e., the training $\mathcal{F}_{48\times48}$ samples were used to train CNN_1 architecture while the performance of the trained CNN_1 model was evaluated using the remaining testing samples. Learning the parameters in any CNN architecture is a very important task and depends on two factors, i.e., epochs and batches. Both these factors affect the learning capabilities of the architecture during the training of samples in the network. Thus, a trade-off between epochs and batches was established, which improved the performance of FERS using CNN_1 architecture (Figure 4). The demonstration of the performance with the trade-off between epochs and batches is shown in Figure 11 and from this figure, it is observed that the performance gradually improved with increasing epochs (best performance at nearly 700 epochs) while keeping 16 batches fixed.

Inspired by the experiment shown in Figure 11, another experiment was conducted using CNN_1 architecture while keeping the fixed batch = 16 with varying epochs with respect to the KDEF, CK+, and SFEW database; the performance is shown in Figure 12. From this figure, we can observe that during epochs between 700 and 800, the performance for each database was good.

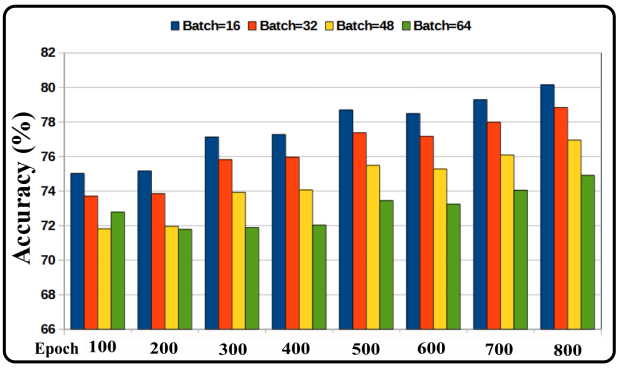

Figure 11. Effect on the performance of the proposed FERS due to the trade-off between epochs and batches using CNN_1 architecture for the CK+ database.

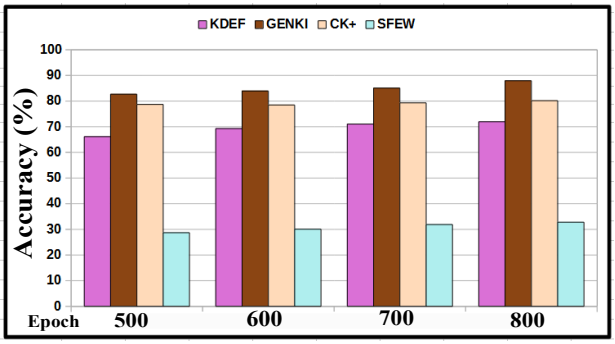

Figure 12. Effect on the performance of the proposed FERS while keeping the batch fixed with varying epochs using CNN_1 architecture for the KDEF, CK+, and SFEW database.

4.2.1. Effect of Data Augmentation Techniques

The data augmentation techniques were applied to training samples to increase the number of samples. The increased training samples learned the parameters of CNN_1 architecture well and obtained a better performance. Moreover, in order to adapt the diversity of the training data and to avoid overfitting problems, data augmentation plays an important role. In this work, each sample of the training images was horizontally and then vertically flipped. Then, Affine transformations such as rotation, scaling, zooming, and shearing operations were performed. For the data augmentation technique, we employed the methods mentioned in [15], which derives seventeen images for each sample. Figure 13 shows the effect of data augmentation techniques on the performance of FERS using CNN_1 architecture and from this figure, we can observe that the performance of the proposed FERS increases due to the employed data augmentation techniques.

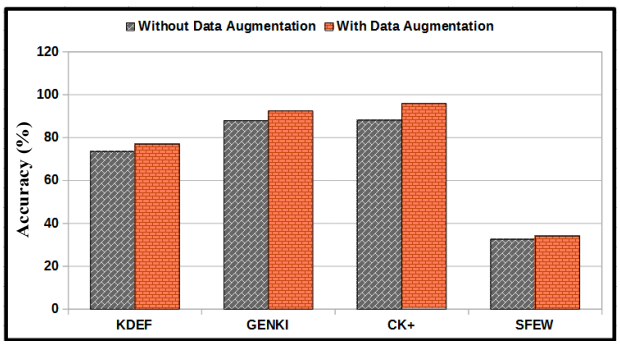

Figure 13. Effect on the performance of the proposed FERS using proposed data augmentation techniques.

4.2.2. Effect of Progressive Image Resizing

The progressive image resizing technique has been discussed in Section 3.3. During experimentation, the preprocessed face region $\mathcal{F} \in \mathbb{R}^{200\times200}$ was further down-sampled into $n_1 \times n_1 \times 3$, $2n_1 \times 2n_1 \times 3$, and $4n_1 \times 4n_1 \times 3$ size images. Here, we already considered $n_1 = 48$ in the above experiments. Hence, in progressive image resizing, we down-sampled $\mathcal{F} \in \mathbb{R}^{200\times200}$ to $\mathcal{F} \in \mathbb{R}^{48\times48}$, $\mathcal{F} \in \mathbb{R}^{96\times96}$, and $\mathcal{F} \in \mathbb{R}^{192\times192}$. In other words, the CNN_1 architecture (Figure 4) was trained with $\mathcal{F}_{48\times48\times3}$ images. Then, $\mathcal{F}_{96\times96\times3}$ images were used to train CNN_2 architecture (Figure 5). Lastly, the CNN_3 architecture was trained with $\mathcal{F}_{192\times192\times3}$ images. The purpose behind learning these architectures with the increasing image sizes concern the fact that (i) the high-resolution images are trained in the network; (ii) the effect of multi-resolution approaches can be introduced in the network such that the texture patterns at the higher level of abstraction will be reflected during the learning of parameters; and (iii) the system will provide deeper information that would be beneficial for the hierarchical representations of features. Hence, the use of progressive image resizing not only increases the performance of the recognition system but also reduces the overfitting problems. The effect of progressive image resizing on the performance of the proposed system is reported in Table 5 and from this table, we can observe that for the KDEF, GENKI, CK+, and SFEW databases, the proposed FERS exhibits a better performance for $\mathcal{F}_{192\times192\times3}$ images than for both $\mathcal{F}_{96\times96\times3}$ and $\mathcal{F}_{48\times48\times3}$ images. Moreover, it is evident that both progressive image resizing and data augmentation techniques together are very effective for the proposed CNN models for recognizing facial expression on facial regions.

Table 5. Effect of the progressive image resizing on the performance of the proposed FERS where the first, second, and third row for each database shows the accuracy in percentage using CNN_1 ($\mathcal{F} \in \mathbb{R}^{48\times48}$), CNN_2 ($\mathcal{F} \in \mathbb{R}^{96\times96}$), and CNN_3 ($\mathcal{F} \in \mathbb{R}^{192\times192}$) models, respectively.

Database	Data Augmentation	No Data Augmentation
KDEF	75.95	71.67
	78.18	74.92
	80.92	78.21
GENKI	92.45	87.91
	94.13	89.77
	95.59	91.16
CK+	95.89	88.15
	96.22	92.67
	96.71	95.20
SFEW	34.05	32.56
	34.91	33.11
	35.72	33.34

4.2.3. Effect of Transfer Learning

For this technique, we used two different approaches: (i) in the first approach, we freshly trained CNN_1, CNN_2, and CNN_3 architectures with the corresponding image sizes, i.e., refreshed models were used; (ii) in the second approach, we used the trained CNN_1 model as a retrained model for CNN_2 such that only upper layers of the CNN_2 model were trained with $\mathcal{F}_{96\times96\times3}$ images. Similarly, the upper layers of the CNN_3 architecture were trained by $\mathcal{F}_{192\times192\times3}$ images, while for the remaining layers, the weights of the trained CNN_2 model were used. Figure 14 presents the effect of transfer learning approaches on the performance of the proposed FERS. Here, only the performance of the CNN_3 model trained with $\mathcal{F}_{192\times192\times3}$ images is shown using both progressive image resizing and data augmentation techniques.

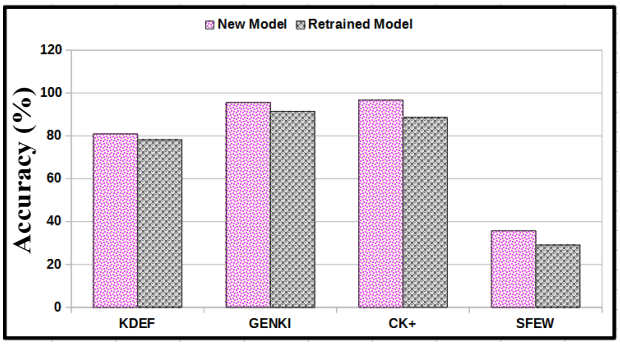

Figure 14. Effect of transfer learning on the performance of the proposed FERS.

4.2.4. Effect of Score Fusion

Score fusion techniques defined in Equations (1) and (2) were applied on the classification scores obtained by the CNN_1, CNN_2, and CNN_3 architectures, and the results are reported in Table 6.

Table 6. Effect of score-level fusion approaches on the performance of the proposed FERS.

Method	**KDEF**	**GENKI**
CNN_1	75.95	92.45
CNN_2	78.18	94.13
CNN_3	80.92	95.59
Sum-Rule	81.53	96.03
Product-Rule	**82.63**	**96.75**
Method	**CK+**	**SFEW**
CNN_1	95.89	34.05
CNN_2	96.22	34.91
CNN_3	96.71	35.72
Sum-Rule	97.07	36.15
Product-Rule	**97.32**	**36.79**

To better understand the performance of the proposed FERS, the confusion matrices are shown in Figure 15, corresponding to the KDEF, GENKI, CK+, and SFEW facial expression database. Here, each confusion matrix represents the product-rule-based fusion performance of the proposed FERS.

4.2.5. Comparison

To compare the performance of the proposed methodology, we computed features from the competing methods and obtained the performance under the same training–testing protocol. Here, the performance of methods of Vgg16 [67], ResNet50 [68], that of Zavare et al. [42], Inception-v3 [69], and that of Rao et al. [70] were compared with the performance of the proposed system for the KDEF database, as presented in Table 7. For the GENKI database the performance of the proposed system was compared with Vgg16, ResNet50, Inveption-v3, that of An et al. [29], that of Zhang et al. [71], and that of Gao et al. [72], and the competing methods are presented in Table 8. Similarly, we compared the performance of the proposed system with that of Sun et al. [73], ResNet50, and Inveption-v3 for the CK+ database in Table 9. For the SFEW database, the performance of Liu et al. [74], Vgg16, ResNet50, and the Inveption-v3 methods were compared in

Table 10. The comparison of performance, as presented in Tables 7–10, shows the superiority of the proposed system.

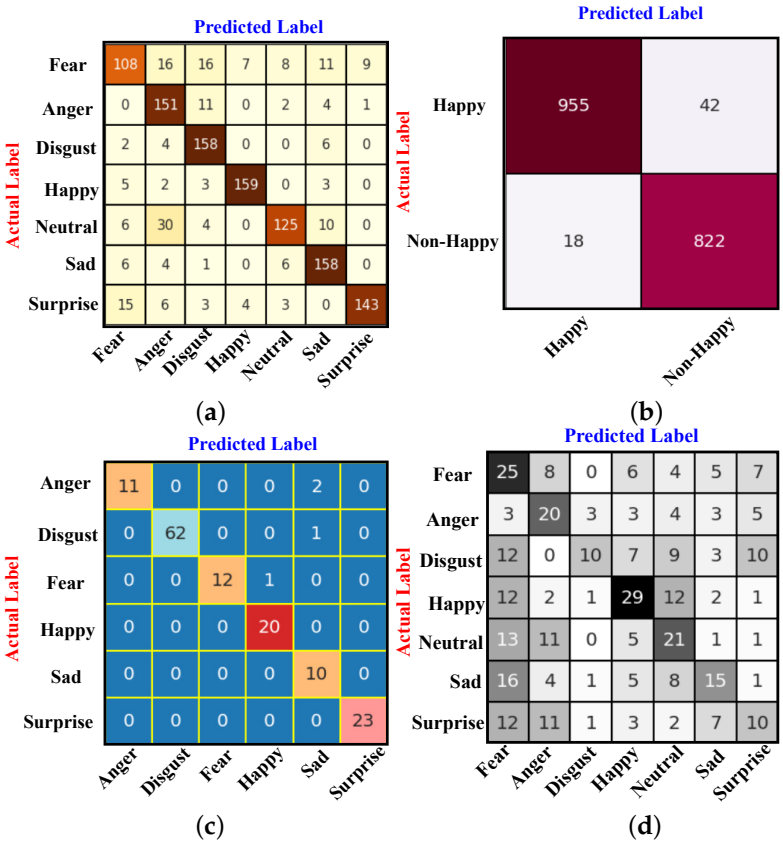

Figure 15. Confusion matrix for the (**a**) KDEF, (**b**) GENKI, (**c**) CK+, and (**d**) SFEW dataset after the product-rule-based fusion performance of the proposed FERS.

Table 7. Performance comparison of the proposed FERS for the KDEF database.

Method	Accuracy (%)	Remarks
Vgg16 [67]	65.08	Images used (980), expression class (7), train/test split
ResNet50 [68]	72.32	Images used (980), expression class (7), train/test split
Zavare et al. [42]	72.55	Images used (980), expression class (7) Images type (frontal), 10-fold cross validation
Inception-v3 [69]	75.04	Images used (980), expression class (7), train/test split
Rao et al. [70]	74.05	Images used (720), expression class (6) Images type (frontal), 10-fold cross validation
Proposed	**82.63**	Images used (980), proposed CNN for seven expression classes

Table 8. Performance comparison of the proposed FERS for the GENKI database.

Method	Accuracy (%)	Remarks
Vgg16 [67]	72.08	VGG16 CNN for seven expression classes
ResNet50 [68]	82.30	ResNet 50 CNN for seven expression classes
Inception-v3 [69]	85.38	Inception-v3 CNN for seven expression classes
An et al. [29]	88.50	Feature (HOG), classifier (ELM)
Zhang et al. [71]	94.21	Feature (CNN), classifier (Softmax)
Gao et al. [72]	94.33	Feature (ensemble), classifier (ensemble)
Proposed	**96.75**	Proposed CNN for seven expression classes

Table 9. Performance comparison of the proposed FERS for the CK+ database.

Method	Accuracy (%)	Remarks
ResNet50 [68]	91.87	Images used (981), expression class (7), train/test split
Inception-v3 [69]	94.07	Images used (981), expression class (7), train/test split
Sun et al. [73]	94.67	Images used (510), expression class (7), k-fold cross-validation
Proposed	**96.81**	Images used (981), proposed CNN for seven expression classes

Table 10. Performance comparison of the proposed FERS for the SFEW database.

Method	Accuracy (%)	Remarks
Vgg16 [67]	24.78	Images used (700), expression class (7), train (346)/test (354)
ResNet50 [68]	24.98	Images used (700), expression class (7), train (346)/test (354)
Inception-v3 [69]	29.52	Images used (700), expression class (7), train (346)/test (354)
Liu et al. [74]	26.14	Images used (700), expression class (7), train (346)/test (354)
Proposed	**36.79**	Images used (700), expression class (7), Train (346)/Test (354)

5. Conclusions

A novel method for facial expression recognition systems has been proposed in this work. The objective of the proposed system is to predict the seven basic types of expressions on the human face. The applications of this proposed system have been well described and demonstrated in the diversified fields of e-Healthcare, social IoT, emotion AI, and cognitive AI. The implementation of the proposed system has three components. In the first component, an image preprocessing task has been performed where a face region was extracted from a body silhouette image using the facial landmark points. Then, in the second component, from the extracted face region, the multi-resolution images were considered. The convolutional neural network architectures have been proposed for each resolution of the images. Here, the images undergo the CNN architectures and are classified into seven basic facial expression classes based on learning the parameters of CNN models. To enhance the performance of the recognition system and better handle the challenging issues of the facial expression recognition system, some advanced techniques such as image augmentation, progressive image resizing, transfer-learning, and fine-tuning of parameters were employed in the third component. Finally, fusion methods were applied to the best performance of the different CNN models to achieve a better performance than the existing state-of-the-art methods. Extensive experimentation has been performed using four benchmark databases, namely KDEF, GENKI-4k, CK+, and SFEW, and the performance of the proposed system has been compared with some existing methods concerning each database. The comparison of the performance of the proposed method with the competing methods shows the superiority of the proposed system.

Author Contributions: Original Draft Preparation, Methodology, S.U.; Investigation, Formal Analysis, S.H.; Supervision, Funding Acquisition, V.A.; Conceptualization, Validation, R.K.R. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by Dr. Vijayan Asari, Professor in Electrical and Computer Engineering, University of Dayton, 300 College Park, Dayton, OH 45469-0232.

Institutional Review Board Statement: The ethics committee or institutional review board approval is not required for this manuscript. This research respects all the sentiments, dignity, and intrinsic values of animals or humans.

Informed Consent Statement: Not applicable.

Data Availability Statement: In this manuscript, the employed datasets have been taken with license agreements from the corresponding institutions with proper channels.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Mollahosseini, A.; Chan, D.; Mahoor, M.H. Going deeper in facial expression recognition using deep neural networks. In Proceedings of the 2016 IEEE (WACV), Lake Placid, NY, USA, 7–10 March 2016; pp. 1–10.
2. Jaimes, A.; Sebe, N. Multimodal human–computer interaction: A survey. *Comput. Vis. Image Underst.* **2007**, *108*, 116–134. [CrossRef]
3. Kaukomaa, T. *Facial Expressions as an Interactional Resource in Everyday Face-to-Face Conversation*; Helsingin yliopisto: Helsinki, Finland, 2015.
4. Tsai, H.H.; Chang, Y.C. Facial expression recognition using a combination of multiple facial features and support vector machine. *Soft Comput.* **2018**, *22*, 4389–4405. [CrossRef]
5. Ekman, P.; Friesen, W.V. Constants across cultures in the face and emotion. *J. Personal. Soc. Psychol.* **1971**, *17*, 124. [CrossRef]
6. Sun, X.; Lv, M. Facial expression recognition based on a hybrid model combining deep and shallow features. *Cogn. Comput.* **2019**, *11*, 587–597. [CrossRef]
7. Jack, R.E.; Sun, W.; Delis, I.; Garrod, O.G.; Schyns, P.G. Four not six: Revealing culturally common facial expressions of emotion. *J. Exp. Psychol. Gen.* **2016**, *145*, 708. [CrossRef] [PubMed]
8. Muhammad, G.; Alsulaiman, M.; Amin, S.U.; Ghoneim, A.; Alhamid, M.F. A facial-expression monitoring system for improved healthcare in smart cities. *IEEE Access* **2017**, *5*, 10871–10881. [CrossRef]
9. Jarwar, M.A.; Chong, I. Exploiting IoT services by integrating emotion recognition in Web of Objects. In Proceedings of the 2017 International Conference on Information Networking (ICOIN), Da Nang, Vietnam, 11–13 January 2017; pp. 54–56.
10. Barrett, L.F. AI Weighs in On Debate about Universal Facial Expressions. *Nature* **2021**, *589*, 202–203. [CrossRef] [PubMed]
11. Lisetti, C.L.; Schiano, D.J. Automatic facial expression interpretation: Where human-computer interaction, artificial intelligence and cognitive science intersect. *Pragmat. Cogn.* **2000**, *8*, 185–235. [CrossRef]
12. Fernández-Dols, J.M.; Russell, J.A. *The Psychology of Facial Expression*; Cambridge University Press: Cambridge, UK, 1997.
13. Ekman, P.; Rosenberg, E.L. *What the Face Reveals: Basic and Applied Studies of Spontaneous Expression Using the Facial Action Coding System (FACS)*; Oxford University Press: Oxford, UK, 1997.
14. Umer, S.; Dhara, B.C.; Chanda, B. Face recognition using fusion of feature learning techniques. *Measurement* **2019**, *146*, 43–54. [CrossRef]
15. Umer, S.; Rout, R.K.; Pero, C.; Nappi, M. Facial expression recognition with trade-offs between data augmentation and deep learning features. *J. Ambient. Intell. Humaniz. Comput.* **2021**, 1–15. [CrossRef]
16. Mavadati, S.M.; Mahoor, M.H.; Bartlett, K.; Trinh, P.; Cohn, J.F. Disfa: A spontaneous facial action intensity database. *IEEE Trans. Affect. Comput.* **2013**, *4*, 151–160. [CrossRef]
17. Pantic, M.; Rothkrantz, L.J.M. Automatic analysis of facial expressions: The state of the art. *IEEE Trans. Pattern Anal. Mach. Intell.* **2000**, *22*, 1424–1445. [CrossRef]
18. Dalal, N.; Triggs, B. Histograms of Oriented Gradients for Human Detection. In Proceedings of the IEEE Computer Society Conference on (CVPR), San Diego, CA, USA, 20–25 June 2005; Volume 1, pp. 886–893. [CrossRef]
19. Jindal, R.; Vatta, S. Sift: Scale invariant feature transform. *IJARIIT* **2010**, *1*, 1–5.
20. Ojala, T.; Pietikäinen, M.; Harwood, D. A comparative study of texture measures with classification based on featured distributions. *Pattern Recognit.* **1996**, *29*, 51–59. [CrossRef]
21. Liu, M.; Shan, S.; Wang, R.; Chen, X. Learning expressionlets on spatio-temporal manifold for dynamic facial expression recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, 23–28 June 2014; pp. 1749–1756.
22. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. Imagenet classification with deep convolutional neural networks. *Commun. ACM* **2017**, *60*, 84–90. [CrossRef]
23. Krumhuber, E.G.; Küster, D.; Namba, S.; Shah, D.; Calvo, M.G. Emotion recognition from posed and spontaneous dynamic expressions: Human observers versus machine analysis. *Emotion* **2019**, *21*, 447–451. [CrossRef] [PubMed]
24. Le Cun, Y.; Bottou, L.; Bengio, Y. Reading checks with multilayer graph transformer networks. In Proceedings of the 1997 IEEE International Conference on Acoustics, Speech, and Signal Processing, Munich, Germany, 21–24 April 1997; Volume 1, pp. 151–154.
25. Shojaeilangari, S.; Yun, Y.W.; Khwang, T.E. Person independent facial expression analysis using Gabor features and Genetic Algorithm. In Proceedings of the 2011 8th International Conference on Information, Communications & Signal Processing, Singapore, 13–16 December 2011; pp. 1–5.
26. Dhall, A.; Goecke, R.; Lucey, S.; Gedeon, T. Static facial expression analysis in tough conditions: Data, evaluation protocol and benchmark. In Proceedings of the 2011 IEEE International Conference on Computer Vision Workshops (ICCV Workshops), Barcelona, Spain, 6–13 November 2011; pp. 2106–2112.
27. Zhao, X.; Liang, X.; Liu, L.; Li, T.; Han, Y.; Vasconcelos, N.; Yan, S. Peak-piloted deep network for facial expression recognition. In Proceedings of the European Conference on Computer Vision, Amsterdam, The Netherland, 8–16 October 2016; pp. 425–442.
28. Tian, Y.I.; Kanade, T.; Cohn, J.F. Recognizing action units for facial expression analysis. *IEEE Trans. Pattern Anal. Mach. Intell.* **2001**, *23*, 97–115. [CrossRef] [PubMed]
29. An, L.; Yang, S.; Bhanu, B. Efficient smile detection by extreme learning machine. *Neurocomputing* **2015**, *149*, 354–363. [CrossRef]
30. Jain, A.K.; Li, S.Z. *Handbook of Face Recognition*; Springer: London, UK, 2011; Volume 1.

31. Turk, M.; Pentland, A. Eigenfaces for recognition. *J. Cogn. Neurosci.* **1991**, *3*, 71–86. [CrossRef]
32. Sonka, M.; Hlavac, V.; Boyle, R. *Image Processing, Analysis, and Machine Vision*; Cengage Learning: Stamford, CT, USA, 2014.
33. He, X.; Yan, S.; Hu, Y.; Niyogi, P.; Zhang, H.J. Face recognition using laplacianfaces. *IEEE Trans. Pattern Anal. Mach. Intell.* **2005**, *27*, 328–340.
34. Delac, K.; Grgic, M.; Grgic, S. Independent comparative study of PCA, ICA, and LDA on the FERET data set. *Int. J. Imaging Syst. Technol.* **2005**, *15*, 252–260. [CrossRef]
35. Yang, M.; Zhang, L. Gabor feature based sparse representation for face recognition with gabor occlusion dictionary. In Proceedings of the European Conference on Computer Vision, Heraklion, Greece, 5–11 September 2010; pp. 448–461.
36. Gutta, S.; Wechsler, H.; Phillips, P.J. Gender and ethnic classification of face images. In Proceedings of the Third IEEE International Conference on Automatic Face and Gesture Recognition, Nara, Japan, 14–16 April 1998; pp. 194–199.
37. Zhang, G.; Wang, Y. Multimodal 2D and 3D facial ethnicity classification. In Proceedings of the 2009 Fifth International Conference on Image and Graphics, Xi'an, China, 20–23 September 2009; pp. 928–932.
38. Zhang, Z.; Lyons, M.; Schuster, M.; Akamatsu, S. Comparison between geometry-based and gabor-wavelets-based facial expression recognition using multi-layer perceptron. In Proceedings of the Third IEEE International Conference on Automatic Face and Gesture Recognition, Nara, Japan, 14–16 April 1998; pp. 454–459.
39. Bartlett, M.S.; Littlewort, G.; Frank, M.; Lainscsek, C.; Fasel, I.; Movellan, J. Recognizing facial expression: Machine learning and application to spontaneous behavior. In Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05), San Diego, CA, USA, 20–25 June 2005; Volume 2, pp. 568–573.
40. Rose, N. Facial expression classification using gabor and log-gabor filters. In Proceedings of the 7th International Conference on Automatic Face and Gesture Recognition (FGR06), Southampton, UK, 10–12 April 2006; pp. 346–350.
41. Wu, T.; Bartlett, M.S.; Movellan, J.R. Facial expression recognition using gabor motion energy filters. In Proceedings of the 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition-Workshops, San Francisco, CA, USA, 13–18 June 2010; pp. 42–47.
42. Zavarez, M.V.; Berriel, R.F.; Oliveira-Santos, T. Cross-database facial expression recognition based on fine-tuned deep convolutional network. In Proceedings of the 2017 30th SIBGRAPI Conference on Graphics, Patterns and Images (SIBGRAPI), Niteroi, Brazil, 17–20 October 2017; pp. 405–412.
43. Gu, W.; Xiang, C.; Venkatesh, Y.; Huang, D.; Lin, H. Facial expression recognition using radial encoding of local Gabor features and classifier synthesis. *Pattern Recognit.* **2012**, *45*, 80–91. [CrossRef]
44. Almaev, T.R.; Valstar, M.F. Local gabor binary patterns from three orthogonal planes for automatic facial expression recognition. In Proceedings of the 2013 Humaine Association Conference on Affective Computing and Intelligent Interaction, Geneva, Switzerland, 2–5 September 2013; pp. 356–361.
45. Zhao, G.; Pietikainen, M. Dynamic texture recognition using local binary patterns with an application to facial expressions. *IEEE Trans. Pattern Anal. Mach. Intell.* **2007**, *29*, 915–928. [CrossRef] [PubMed]
46. Zhu, X.; Ramanan, D. Face detection, pose estimation, and landmark localization in the wild. In Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition, Providence, RI, USA, 16–21 June 2012; pp. 2879–2886.
47. Mita, T.; Kaneko, T.; Hori, O. Joint haar-like features for face detection. In Proceedings of the Tenth IEEE International Conference on Computer Vision (ICCV'05) Volume 1, Beijing, China, 17–21 October 2005; Volume 2, pp. 1619–1626.
48. Liu, Y.; Cao, Y.; Li, Y.; Liu, M.; Song, R.; Wang, Y.; Xu, Z.; Ma, X. Facial expression recognition with PCA and LBP features extracting from active facial patches. In Proceedings of the 2016 IEEE International Conference on Real-time Computing and Robotics (RCAR), Angkor Wat, Cambodia, 6–10 June 2016; pp. 368–373.
49. Tuceryan, M.; Jain, A.K. Texture Analysis. In *Handbook of Pattern Recognition & Computer Vision*; World Scientific Publishing Company: Singapore, 1993.
50. Dumoulin, V.; Visin, F. A guide to convolution arithmetic for deep learning. *arXiv* **2016**, arXiv:1603.07285.
51. Thomas, H.; Qi, C.R.; Deschaud, J.E.; Marcotegui, B.; Goulette, F.; Guibas, L.J. Kpconv: Flexible and deformable convolution for point clouds. In Proceedings of the IEEE International Conference on Computer Vision, Seoul, Korea, 27 October–2 November 2019; pp. 6411–6420.
52. Irani, M.; Peleg, S. Motion analysis for image enhancement: Resolution, occlusion, and transparency. *J. Vis. Commun. Image Represent.* **1993**, *4*, 324–335. [CrossRef]
53. Zhou, S.; Wu, Y.; Ni, Z.; Zhou, X.; Wen, H.; Zou, Y. Dorefa-net: Training low bitwidth convolutional neural networks with low bitwidth gradients. *arXiv* **2016**, arXiv:1606.06160.
54. Ma, W.; Lu, J. An equivalence of fully connected layer and convolutional layer. *arXiv* **2017**, arXiv:1712.01252.
55. Ferro-Pérez, R.; Mitre-Hernandez, H. ResMoNet: A Residual Mobile-based Network for Facial Emotion Recognition in Resource-Limited Systems. *arXiv* **2020**, arXiv:2005.07649.
56. Wang, C.Y.; Mark Liao, H.Y.; Wu, Y.H.; Chen, P.Y.; Hsieh, J.W.; Yeh, I.H. CSPNet: A new backbone that can enhance learning capability of cnn. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, Seattle, WA, USA, 14–19 June 2020; pp. 390–391.
57. Ioffe, S.; Szegedy, C. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv* **2015**, arXiv:1502.03167.

58. Srivastava, N.; Hinton, G.; Krizhevsky, A.; Sutskever, I.; Salakhutdinov, R. Dropout: A simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res.* **2014**, *15*, 1929–1958.
59. Mescheder, L.; Geiger, A.; Nowozin, S. Which training methods for GANs do actually converge? *arXiv* **2018**, arXiv:1801.04406.
60. Hinton, G.E.; Srivastava, N.; Krizhevsky, A.; Sutskever, I.; Salakhutdinov, R.R. Improving neural networks by preventing co-adaptation of feature detectors. *arXiv* **2012**, arXiv:1207.0580.
61. Kingma, D.P. Adam: A Method for Stochastic Optimization. *arXiv* **2015**, arXiv:1412.6980.
62. Scherer, D.; Müller, A.; Behnke, S. Evaluation of pooling operations in convolutional architectures for object recognition. In Proceedings of the International Conference on Artificial Neural Networks, Thessaloniki, Greece, 15–18 September 2010; pp. 92–101.
63. Hernández-García, A.; König, P. Further advantages of data augmentation on convolutional neural networks. In Proceedings of the International Conference on Artificial Neural Networks, Rhodes, Greece, 4–7 October 2018; pp. 95–103.
64. Lim, B.; Son, S.; Kim, H.; Nah, S.; Mu Lee, K. Enhanced deep residual networks for single image super-resolution. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, Honolulu, HI, USA, 21–26 July 2017; pp. 136–144.
65. Jain, V.; Crowley, J.L. Smile detection using multi-scale gaussian derivatives. In Proceedings of the 12th WSEAS International Conference on Signal Processing, Robotics and Automation, Cambridge, UK, 20–22 February 2013.
66. Lucey, P.; Cohn, J.F.; Kanade, T.; Saragih, J.; Ambadar, Z.; Matthews, I. The extended cohn-kanade dataset (ck+): A complete dataset for action unit and emotion-specified expression. In Proceedings of the 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition-Workshops, San Francisco, CA, USA, 13–18 June 2010; pp. 94–101.
67. Simonyan, K.; Zisserman, A. Very deep convolutional networks for large-scale image recognition. *arXiv* **2014**, arXiv:1409.1556.
68. Szegedy, C.; Vanhoucke, V.; Ioffe, S.; Shlens, J.; Wojna, Z. Rethinking the inception architecture for computer vision. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 2818–2826.
69. Szegedy, C.; Ioffe, S.; Vanhoucke, V.; Alemi, A.A. Inception-v4, inception-resnet and the impact of residual connections on learning. In Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, San Francisco, CA, USA, 4–9 February 2017.
70. Rao, Q.; Qu, X.; Mao, Q.; Zhan, Y. Multi-pose facial expression recognition based on SURF boosting. In Proceedings of the 2015 International Conference on Affective Computing and Intelligent Interaction (ACII), Xi'an, China, 21–24 September 2015; pp. 630–635.
71. Zhang, X.; Mahoor, M.H.; Mavadati, S.M. Facial expression recognition using *lp-norm MKL* multiclass-SVM. *Mach. Vis. Appl.* **2015**, *26*, 467–483. [CrossRef]
72. Gao, Y.; Liu, H.; Wu, P.; Wang, C. A new descriptor of gradients self-similarity for smile detection in unconstrained scenarios. *Neurocomputing* **2016**, *174*, 1077–1086. [CrossRef]
73. Sun, X.; Xia, P.; Zhang, L.; Shao, L. A ROI-guided deep architecture for robust facial expressions recognition. *Inf. Sci.* **2020**, *522*, 35–48. [CrossRef]
74. Liu, M.; Li, S.; Shan, S.; Chen, X. Au-aware deep networks for facial expression recognition. In Proceedings of the 2013 10th IEEE International Conference and Workshops on Automatic Face and Gesture Recognition (FG), Shanghai, China, 22–26 April 2013; pp. 1–6.