Real-Time Smart Facial Emotion Detection using EmoAnalyzer Model with Minimum Computational Cost

Snehil Sharma

School of Computer Engineering,
KIIT Deemed to be University,
Bhubaneswar 751024, India;
2005545@kiit.ac.in

Aditya Singh Panwar

School of Computer Engineering,
KIIT Deemed to be University,
Bhubaneswar 751024, India;
2005775@kiit.ac.in

Himansu Das

School of Computer Engineering,
KIIT Deemed to be University,
Bhubaneswar 751024, India;
himanshufcs@kiit.ac.in

Abstract

Human emotions are not always communicated clearly through facial expressions, making facial recognition difficult due to variability. While traditional approaches like HOG and SIFT are used for pattern recognition in facial features, machine learning and neural networks have emerged as popular methods for emotion recognition. In this paper we've employed four variations of CNN model and proposed a model named EmoAnalyzer to extract features and landmarks from facial images, classifying them into seven emotions using particularly grayscale images from FERC-2013 dataset. To be specific the Batch normalization along with dropout techniques are used here to prevent overfitting as well as improve accuracy, with model limitations selected based on training results. Test results shows VGG-16 model accurately identifies seven emotions at the accuracy of 67%, VGG-19 model at accuracy of 77%, CNN-V2 model at an accuracy of 64%, MobileNet-V2 model at an accuracy of 82.5% and the EmoAnalyzer (Proposed Model) at an accuracy of 97%.

Keywords— Emotion Recognition, convolution based Neural Network, Deep Learning, EmoAnalyzer, VGG-16, VGG-19, CNN-V2, MobileNet-V2

1 Introduction

Face recognition technology locates and authenticates persons based on their distinguishing facial traits using machine learning algorithms. Its quick development has resulted to its widespread implementation of application on variety of fields, like the security, surveillance along with marketing, and entertainment. By using potent mathematical models to examine various facial patterns, such as eye distance, jawline form, and lip curvature, the method generates a distinctive facial signature that can be compared practically to the database of famously identified faces [1].The Facial recognition technology has a lot of potential, but there are a lot of ethical and privacy concerns that need to be resolved. One of them is accuracy because inaccurate matching can lead to biased and unfair accusations [2]. People with darker skin tones are frequently confused for others due to algorithmic biases, which means that employing the technology could result in prejudice and discrimination against them [3]. The collection and storage of personal data raises significant privacy issues and potential privacy rights violations. Given these worries, more investigation is probably required to completely comprehend the benefits, drawbacks, and ethical implications of facial recognition systems [4]. The creation of regulations to address privacy concerns with facial recognition technology, the eradication of biases, and increased system accuracy may be the main goals of this research [5]. This particular research paper's goal is providing a thorough analysis of facial recognition technology, including its uses, moral dilemmas, and prospective future study fields. The study seeks to advance the current dialogue on facial recognition technology and its social impacts by performing in-depth research on the subject. It emphasises the moral and legal decisions that must be taken

for its proper use, with the goal of increasing awareness of the potential advantages and problems with facial recognition technology.

Facial expression analysis is the process of identifying, assessing, and quantifying the various facial muscle movements that convey a variety of attitudes and emotions. As nonverbal communication accounts for the bulk of human communication, it is crucial to employ nonverbal clues to express emotions, establish rapport, and uphold relationships [6]. By the development of advanced algorithms that have been created as the result of swift developments in the computer vision and machine learning, facial expressions can now be automatically recognised and decoded [7]. The use of these algorithms may be advantageous to a variety of businesses, including marketing, computer interface design, clinical psychology, and others [8]. A quantitative and objective method of assessing emotional states is provided by facial expression analysis, which can lead to more accurate diagnoses, more successful targeted advertising, and improved communication. Important ethical considerations are also raised, especially in relation to informed consent and privacy [9]. People must be informed of the consequences of their consent because the collection and storage of facial expression data has the potential to violate people's right to privacy. A fundamental ethical conundrum needs to be resolved given the potential for misuse of facial expression analysis technologies, particularly in the field of surveillance and for emotion recognition [10]. The main motive of our study is to explain in details of discussion on facial expression analysis by promoting the ethical development and application of this technology while upholding ethical standards and privacy rights.

The use of AI and ML is widespread in a variety of fields, including data mining, where they help detect insurance fraud. Clustered based used data-mining has been used particularly in detecting the patterns of the stock data marketing sector. ML The algorithms have proven their effectiveness in the pattern identification & for the classification tasks such as we know the FER as well as EEG and as for spam identification [11]. The use of ML techniques can provide cost-effective as well as reliable and including computationally efficient solutions for FER.

The objective in particular of this research is to develop as well as perform a evaluation on a real-time intelligent emotion recognising system that used the technique of Convolution based Neural Networks (CNN) for automatic facial emotions recognition. The system is designed to accurately and efficiently recognize and classify emotions in real-time using input from a live video stream. The goal is to achieve high accuracy and real-time performance while minimizing power consumption and computational overhead [12]. The proposed model will be evaluated using standard benchmark datasets, and them compared with existing state of-the-art approaches for the individuals emotion recognition. As a results of this research can have practical applications to be used in fields such as firstly the human and the computer based interaction, mental health & the marketing research.

The Facial-Emotion Recognition (FER) process involves four key steps. Initially, a face is detected within an image and is outlined by a rectangular shape. In the second step, landmarks within the face region are identified [13]. The listed third step extracts particularly the spatial including the temporally listed features from the provided facial input of the components, and at last final it uses the feature extraction (FE) based classifier to generate recognition based results on the extracted so called features. A deep learning (DL)-based facial emotion recognition (FER) method enables full concept based learning to be specific directly from the input provided images, thereby eliminating and solving the reliance on particularly the physics sector orientation based facial models for other listed preprocessing techniques significantly reduced [14]. Among the various DL models the convolution based type of neural networks (CNN's) which particularly are the most commonly used. Using the CNN, the input provided image is being passed from the convolution based layers generating the feature maps. This map obtained is then as an input provided into the completely connected part of layer and as a result the FE classifier determines the facial expression category it belongs to [15].

The data set used in this particular model is known as Facial Emotion Recognition or FER 2013 dataset [16]. This particular data set was initially designed for an internal project, but later it was made public for a Kaggle competition. FERC-2013 dataset comprises 35,000 grayscale face images of size 48×48, where each image is labeled with different emotions. For this particular project, the emotions categorized are happy, angry, neutral, sad, and fear.

As for the summary, the most important contribution of the complete study is:

- We proposed a brand new facial emotion recognition model name 'EmoAnalyzer' which takes less computation cost.
- We implemented a facial emotion recognizer using four other models namely VGG-16, VGG-19, CNN-V2 and MobileNet-V2. We analyzed the achieved performance based on various limitations like as in the

accuracy including the precised output part plus recall and at last the F1 score.

The remaining paper has been organized just like mentioned further. Section-2 summarizes the literature based analysis on Facial emotion detection, the models used and its implementation. Section-3 describes the methodology like CNN, VGG-16, VGG-19, CNN-V2, MobileNet-V2, EmoAnalyzer in detail. Section-4 congregate the various outputs obtained. At last as for the Section-5, we've discussed most relevant conclusion plus providing future lines of investigation of this study.

2 Literature Review

Yadan Lv et al. [17] used facial recognition and deep learning techniques in their work. The parsed algorithmic elements lessen the need for additional features to compress the image or remove noise by aiding in differentiating feature detection. The parsed approach is a crucial and distinctive technique for attaining accurate facial recognition.

Mehmood et al. used deep learning ensemble techniques and the best feature selection algorithms in their work [18] to extract emotional information from human brain EEG sensors. Using a more complex face recognition system, the work expands EEG feature extraction and feature selection techniques. The four emotional states of joy, tranquilly, sadness, and horror are described in the study. The balanced one-way ANOVA method, which is excellent for feature selection and improves emotional classification accuracy, is used in the feature extraction procedure. One method for enhancing EEG recognition is arousal-valence space.

Li et al.'s [19] method combines exact crowdsourcing with strict locality-preserving learning to distinguish between different facial expressions of emotion in unstructured contexts. They employed a deep learning system based on the RAFDB facial recognition algorithm to reduce crowd-sourcing while maintaining the new localization loss layer. Age and gender are the basis for two of the five methods the RAFDB uses. While the second method establishes the spatial dimensions of the image, the third way identifies two subgroups. The first subset has seven emotions, while the second subset contains twelve. The fifth approach classifies photos depending on input, while the fourth method verifies accuracy.

Chen et al. [20] used the softmax type regressing deep sparse auto encoder of the network to isolate emotional facial emotions during human & robot inter linkage. They applied his SRDSAN method to reduce bias and assess learning on the basis of its efficient behaviour and dimension based complexity. The input signal was classified using softmax regression and accurate feature data was obtained using DSAN.

Babajee et al. [21] developed a deep learning technique to identify human expressions from facial expressions. The paper assigns seven categories to convolution based neural networks, which are used to differentiate between facial expressions like happiness, sadness, etc. They collected information using the Facial-Action-Coding-System i.e. (FACS), and currently have 32,398 datasets for different types of emotion recognition. The study only focuses on the identification procedure; it does not serve as an optimisation method.

Hassouneh et al. [22] developed a real time system for emotions identification based on facial types of expressions and using the E.E.G for deep neural networks and machine learning. To differentiate between different faces, virtual markers and the processing-light optical flow technique were used.

Tan et al. [23] employed neuro-sense models of spatiotemporal EEG signals in their study to identify and categorise transient emotions. The SNN technique, which facilitates the identification of brain activity, was used for the first time in this work. By analysing EEG data using the arousal-valence space, they were able to distinguish between high and low arousal and valence strategies.

Satyanarayana et al. [24] brought to use the emotions recognition using in particular the deep learning & cloud accessibility. To create a unique IP address for each technique, they gathered data on a range of responses to loss, love, tranquilly, and wrath. Face identification is improved by deep learning's ability to distinguish between a variety of facial expressions, which is a crucial method in many applications.

A thorough system for classifying emotions was developed by Jayanthi et al. [25] by combining deep classifiers with voice and still images. You can assess your physical and mental health by speaking slowly. This

technology uses vocal modulations and emotional recognition to anticipate human nature's emotional state more accurately than existing algorithms.

Deep facial emotions identification for the network system is a hurdle because of the short training sets and unrelated expression changes, according to a study by Li et al. [26]. By applying neural pipeline technology, they disseminated the dataset while avoiding multiple significant FER method difficulties.

Yadahalli et al. [27] employed deep learning to discern between various face expressions. They collected data on six different emotional states, including happiness, grief, fury, fear, neutrality, and surprise, using an eight-layer dataset. The results show that utilising a convolution based neural network to distinguish between various facial expressions enhances accuracy and produces a distinctive output for a variety of facial emotions with a single method.

Yang et al.'s [28] deep learning-based three-class emotion identification system makes use of a stacked auto-encoder. They looked at the discrete entropy of the EEG signals and found that the deep learning algorithm's auto-encoder methodology provided better accuracy than the computational techniques used by the encoding system. The classification outcomes and precision of deep learning systems used to measure emotions were improved by using alpha, beta, and gamma values. The effectiveness of deep learning algorithms in a range of emotional recognition classes has received high accolades.

Asaju et al. [29] created a temporal method for recognising face emotional expressions by combining deep learning and a convolution based neural network. After gathering data utilising VGG-19 methods, they used the BiL-STM architecture to precisely map and identify facial expressions of emotion. To enhance classification and accuracy results, deep learning neural networks were incorporated into the CNN-BiL-STM approach. While the Effective-State-In-The-E.Environment (DAiSEE) data set was used in identifying the faces of people who were bewildered and disturbed, the Denver-Intensity-Of-The-Spontaneous-Facial-Action (DIS-FA) data set was used in identifying the faces of people who were pleased, sad, furious, and neutral. One of the trickiest methods for facial expression recognition, face emotion detection, and face identification was employed in the work of Sati et al. [30]. Later, additional features were incorporated to enhance the ANN method and boost the precision of face expression recognition.

Using maximally boosted CNN and LSTM, Rajan et al. [31] created a novel deep learning model for face expression identification. With the enhanced FER method, they developed a somewhat different model with noise reduction and dimensionality space function preprocessing techniques. This work demonstrates that the combination of LSTM and MBCNN may generate extremely accurate feature extraction and classification results. The dual CNN technique was used to further boost accuracy.

Recurrent neural networks were used by Mirsamadi et al. [32] in their article to create automatic voice emotion recognition. This paper demonstrates the usage of the RNN based architecture used for feature selecting process and the implementation of a novel weighting time pool based technique to improve the extraction of prominent features. The classifications of emotional recognition were enhanced using the IEMO-CAP approach.The classic SVM-type SER with fixed designed features and the IEMO-CAP classifier were compared in the end findings.

The examination of gender bias in emotions recognising based Artifical Intelligence was studied by Domnich et al. [33]. They gathered the SASE FE data set depending on the given gender and gave an facial emotions recognising overview using AI. Three neural networks from each of the two groups—male and female—were used to classify the dataset. The job was divided into three separate categories and put through testing and training phases. It was anticipated that this approach would result in precise and ideal outcomes.

For the recognition of facial expressions and ordinal intensity estimate, Ekundayo et al. [34] used a multilabel convolution neural network. They discovered that multiclass flawless emotional classification could not be achieved using current emotional recognition techniques, such as FER. Due to this, they decided to employ a multilabel convolution based neural network, which was important in resolving the interclass variance issue. They implemented in order to extract features, the Multi label Kernel based Nearest Neighbour and the MLA RAM were utilised. The Binary Cross Entropy loss and the loss from an island were then added to the ML-CNN. To get around a fitting procedure, classification was done using a chain classifier and VGG-16.

A four-category deep learning model was created by Wang et al. [35] using a recently established deep learning approach. Deep architectures and convolution based neural networks, that in particular are essential to the in depth learning model, were included as for the first category. They contain both linear and nonlinear specialised functions, and they play a significant role in assuring data correctness and classification. The convolution type neural layer including pooling layer & the fully connecting layer are our three crucial layers in the CNN. The layer to be pooled aids in reducing the over-fitting issue, while the convolution layer provides filtering capability to minimise the disturbance and the dimension type gap or space created inside filter. To eliminate erroneous data from the function, our completely connected layer that is put before the convolution based and pooling layer.

A survey of the literature analysing the feature based selection techniques in the case of High Dimensional data was done by Gnana et al. [36]. They discovered that sending all data to a statistical measurement approach, which aids in choosing the proper feature selection strategy, is the simplest option. In terms of feature selection strategies, they distinguished between wrapped-based, hybrid-based, embedded-based, and filter-based approaches.

According to Himansu et al. [37], as the web has developed, e-commerce has seen a chance for e-healthcare, which has led to lower labour costs, quicker protections claims, and more. The sorting handle can now be based on side effects, healing reports, and therapy test outcomes thanks to machine learning. X-ray data sets are used in conjunction with this convolution neural network (CNN)-based calculating technique to identify brain tumour and covid virus inside patients. The authors made an effort giving a short quick glimpse of what the therapeutic industry's long-term prospects may be with the consolidation of profound learning.

In order to recognise and classify the side effects of plant ailments, Ashish et al. [38] provides a current demonstration named Res VGG that combined two distinct DL models, like the VGG-16 and Res Net. Nine convolution based layers, distinctly identified two totally additional layers, including one of the softmax layer are all used in the suggested show. The preliminary analysis showed that the suggested show is superior to existing models in terms of infection recognisable proof, enabling preventive steps to be done for eradicating these diseases.

Machine learning is discussed by Abhaya et al. [39] for health administration and symptomatic method. It is possible to integrate machine learning to a variety of applications, including photo division, fraud detection, design acknowledgment, and sickness forecast. Machine learning and profound learning are key to predicting diabetes, a serious health problem. The test findings show that convolution based based neural network & deep learning methodology providing the most notable exactness when compared to other machine learning calculations for diabetic disease forecasting. This chapter compares machine learning calculations with deep neural systems.

3 Methodology

The methodology utilized in this research is outlined in this section. The process involves feeding real-time video into the system and utilizing several Convolution based Neural Network (CNN) various model, such as VGG-16, VGG-19, CNN-V2, MobileNet-V2, and the EmoAnalyzer (proposed model), to identify the emotion being expressed. The system then displays the recognized emotion. More detailed information about each step is provided in the subsequent subsections.

3.1 Convolutional Neural Network (CNN)

The first building block that of CNN is a neuron, which functions by transmitting incoming information, as shown in Figure 1. The input values, represented as x_1 to x_m, are being multiplied with their respective weights and then addition of them is performed by the neuron. The output is then adjusted by passing this sum via a nonlinear activation function and bias term. The bias term is represented by w_0 in Figure 5. Lets assume that an input vector $x = x_1, x_2, \ldots, x_m$ and weight vector $w = w_1, w_2, \ldots, w_m$, hence output of the neuron is $\hat{y} = g(w_0 + \sum_{i=1}^{m}(x_i w_i))$. Output values range from 0 to 1 and are ideal for probability-related problems. Activation functions are used to introduce non-linearity into the network. This is because real-world data generally exhibit nonlinear characteristics. By using a non-linear function, CNN can approximate complex functions.

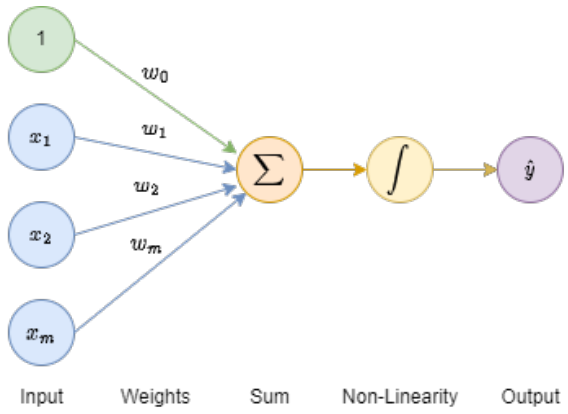

Figure 1: Structure of Neuron

By combining neurons, multi-output CNN can be created. If all inputs are connected to all neurons, it is said to be fully connected or dense. Figure 2 shows a crowded multi type output CNN consisting of 2 neurons. Each neuron in the hidden layer of a deep convolution based neural network is joined to another neuron in the above layer. Figure3 displays a completely connected CNN that has five layers.

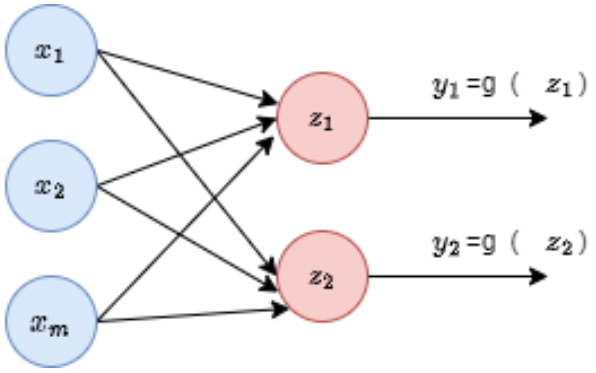

Figure 2: A Neural Network with Two Outputs

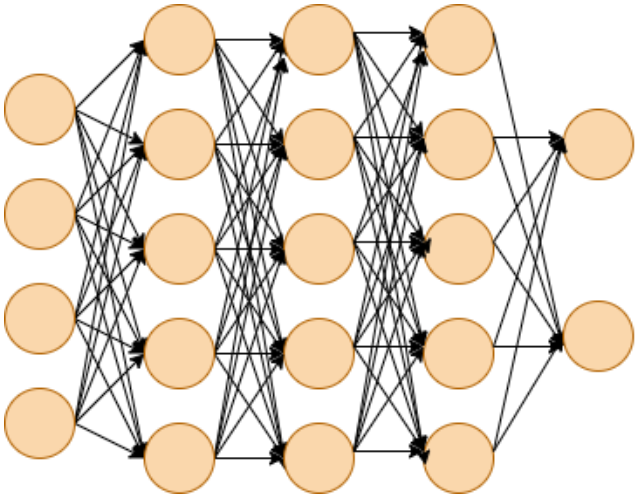

Figure 3: Fully Connected Neural Network

3.1.1 Concept of CNN

The CNN is a type of depth based learning algorithm which processes input images, learns the significance of various objects/aspects within the image using weights and biases, and can distinguish between different images. Compared to other methods of classification, CNN's require less pre-processing. The CNN architecture

is made such a way to resemble in similar way neurons are connected. The human brain was affected by the shape of the visual cortex. One of CNN's primary responsibilities is to convert images into a format that can be processed without losing essential information needed for precise prediction. This is essential if an architecture is to handle massive data sets and learn some features[40]. The main operations of CNN include convolution, pooling (max or average), batch normalization, and dropout, which are explained in more detail in Figure 4.

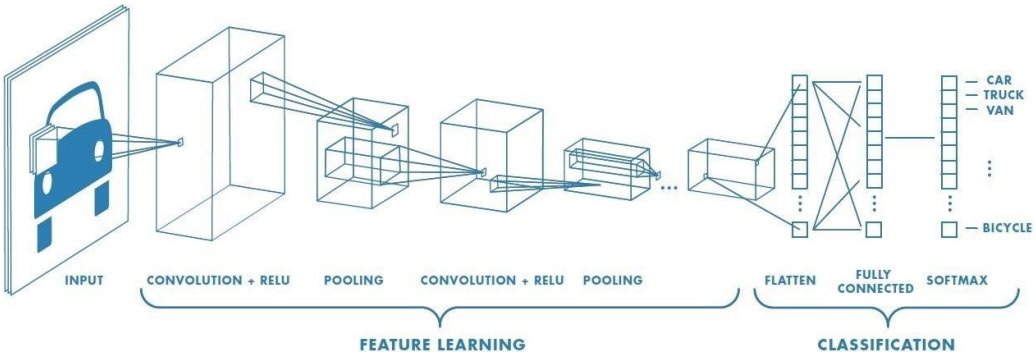

Figure 4: Operation of CNN [41]

3.1.1.1 Convolution operation

Convolution operations main goal is to extract sophisticated characteristics like edges from an input picture. The convolution based layer serves a number of purposes. Basic properties including edges, colour, gradient direction, and rudimentary textures are learned by the first convolution based layer or layers. The last convolution based layer(s) learn characteristics like objects or portions of things, whereas the succeeding convolution based layer(s) learn more intricate textures and patterns.

The crucial element that carries out the convolution operation is the kernel. It strips the feature map of all the extraneous details and concentrates solely on the important ones. The filter advances with a fixed stride length to the right until it scans the entire image's width. It repeat this process until the entire image is covered, then return in the left side of the image with same colour. Figure 5 illustrates this concept by showing a green 5x5 image and a 3x3 kernel filter.

$$\begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix}$$

A stride length of one is selected, which causes the kernel to shift 9 times. Every time, the kernel performs the matrix based multiplication of the corresponding parts of kernel and the image.

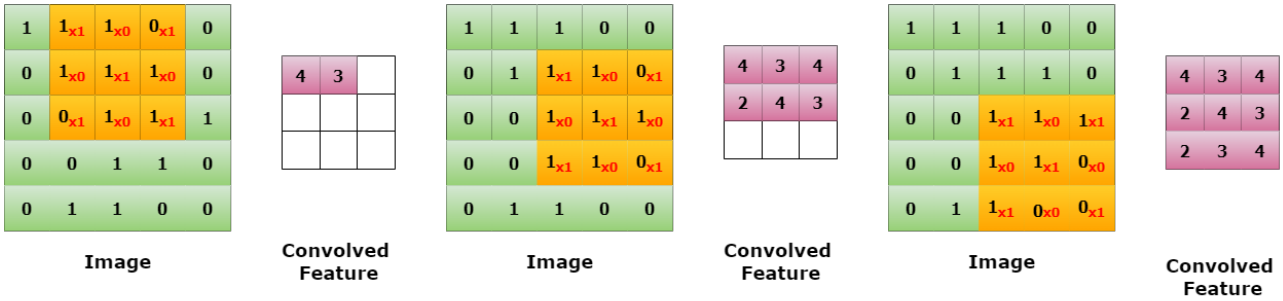

Figure 5: A 5×5 image and a 3×3 kernel are convolved, to create a 3×3 convolved feature

The particular features that results from the convolution procedure may have the same dimensions as the kernel or the original input. Using the same or valid padding allows for this. While valid padding occurs when the feature has the kernel's dimensions, similar padding occurs when the convolved characteristic has similar dimensional structure as the image that has been put as input.

3.1.1.2 Pooling operation

Pooling that is a critical operation in convolution based neural networks (CNN) that helps to reduce the spatial dimension of convolved features, making them more manageable for subsequent layers. This reduces the computational requirements of the network and enables the extraction of features that are invariant to position and rotation. Two common methods for pooling, namely max pooling and average pooling, are frequently utilized to process the output of convolution based layers.

Max pooling selects the maximum value of the kernel's coverage area as the output value. This technique preserves the most crucial characteristic of the input, which depicts as the maximum usually indicates the most important feature in that region of the image. In contrast, avg pooling computes the mean of the covered area of the image.

As a results employing maximum and mean pooling on a particular image are presented in the Figure 6. The particular image is divided into a grid of different regions that do not overlap, and each region undergoes a pooling operation to generate a lower-resolution version of the original input. Max pooling is frequently utilized to capture the most important features of an image, whereas average pooling is employed to down-sample the image more softly and capture a broader sense of the input. The selection of the pooling technique is determined by the network's specific needs and the characteristics of the input data.

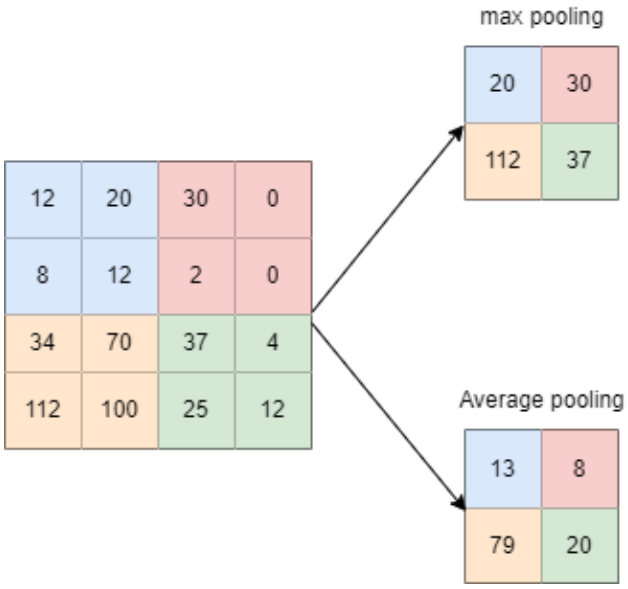

Figure 6: Max & Average Pooling

3.1.1.3 Fully connected layer

In CNN, every neuron in the connection of layer, which is particularly the last layer, is linked to all other neurons in the preceding layers. The input that is of the layer beneath is transformed into an 1D-vector and then passed through an activation function to generate the output in this layer.

3.1.1.4 Dropout

Dropout is an approach utilized in machine learning models to prevent over fitting, a situation where the trained accuracy which is considerably higher than the output testing accuracy . To reduce size of network, dropout excludes some neurons during the forward or backward pass. As illustrated in Figure 7, this is achieved by randomly selecting a group of neurons to disregard. A rate of 1.0 implies no dropout, whereas a rate of 0.0 indicating that all layers output are ignored. The dropout rate reflects the probable chances of training a specific node in that individual layer.

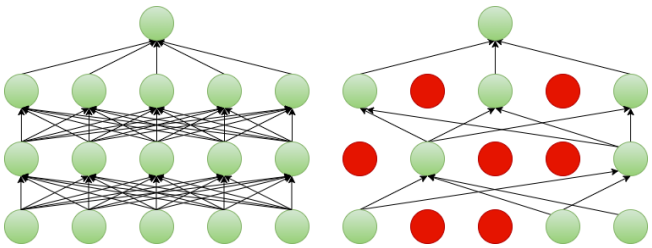

Figure 7: Dropout in a Neural Network

3.1.1.5 Batch normalization

In order to train a network effectively, it's important to ensure that the inputs in each layer are evenly distributed. Otherwise, the model may become biased. Batch normalization is employed to normalize the inputs and establish more consistent distributions.

3.1.1.6 Activation functions

The succeeding paragraph explains two activation functions that are widely used in CNN - Softmax & Exponential Linear Unit (ELU). Specifically, the softmax function is defined and discussed.

$$\frac{e^{z_j}}{\sum_{j=1}^{K} e^{z_j}} \tag{1}$$

As mentioned in the eqn. (1) statement, the softmax function employs z_i to denote input values and K to represent the total number of values that have been input. The function changes real numbers to probability based output, ensuring that they always add up to 1 and are restricted between 0 and 1. The data can now be depicted as a probability based distribution for the 5 emotions since it is frequently utilized in the connected connected layer of suggested models. Figure 8 depicts the position of the softmax function.

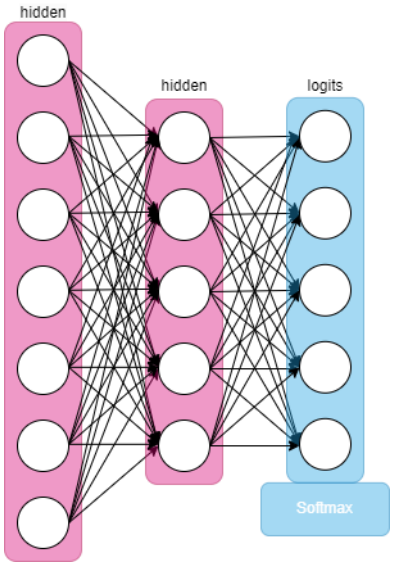

Figure 8: Softmax Function

The ReLU function is given as

$$\begin{cases} x & x > 0 \\ \alpha(e^x - 1) & x \leq 0 \end{cases} \tag{2}$$

The above passage explains the eqn. (2) where x represents the input value, and α represents the slope. When

x is negative, the function reaches saturation at a negative value, which is controlled by α. As a result, the information passed to the next layer is reduced [42].

3.1.2 CNN Architecture

The paragraph discusses how high-level Application Programming Interfaces (APIs) such as Keras can simplify the building and training of Machine Learning (ML) models. The paper describes the development of a sequential convolution based Neural Network (CNN) model using Tensorflow with the Keras API. This approach allows for the gradual construction of the model, layer by layer. TensorFlow is a comprehensive open-source platform for machine learning that provides a diverse array of resources, tools, and libraries to develop and deploy ML applications. Additionally, Figure 9 presents the architecture of a CNN, with the term "conv." representing convolution.

Figure 9: CNN's Structure

The size of the input image gets less after each stage in the four-stage CNN model. Similar layers that start with convolution and terminate with dropout make up the first three stages. In the first stage, convolution is applied using the limitations shown in Table 1 to an input layer for a 48×48 image. The only difference between the limitations of each convolution based layer in network is the total numbers of kernels.

In order to generate kernels randomly, the He-normal initializer is utilized [40]. In the initial phase, 64 kernels are employed, and then the batch normalizing technique is implemented to produce the inputs for the next layer. The subsequent layers repeat batch normalisation and convolution. The output size of the following layer, which employs max pooling and has a pool size of 2×2, is 24×24. Then, a dropout rate of 0.35 is used. By using max pooling, the output size of the second stage is 12×12, while employing the dropout-rate that of 0.4 and 128 kernels.

The output size is decreased to 6×6 in the third stage by using 256 kernels with a dropout-rate that of 0.5 along with max pooling. A flatten layer precedes the final stage's dense and output layers. The motive of the flatten layer is to convert the 2'D data into an 1'D array, which is particularly necessary to classify all the seven emotions. The final output is subsequently put into the dense part of the layer, in turn which utilizes the softmax functioning to generate class probabilities after batch normalization.

Table 1: The parameters of convolution

Kernal	3×3
Activation	ReLU
Padding	Same
Kernels	64, 128, 256
Kernel initializer	He-normal

3.1.3 Compilation of models

To configure the model, two parameters need to be set: optimizer and metrics. The optimizer is chosen between Adam and Nadam based on the loss for updating the model. The following metrics are employed: accuracy & categorical cross-entropy loss & precision & recall & F1-score. The Result & Discussion chapter includes a comprehensive explanation of each of these metrics.

3.1.4 Training the models

To construct a model, the dataset is partitioned into two segments: training and testing sets. The train-test split() function is used to accomplish this, which guarantees that the two segments are distinct. The data is divided in such a way that 80% is reserved for training, while 20% is used for testing. During training, Learning Rate (LR) plays an important role in deciding how fast weights are calculated by the model; it can either cause quick convergence or slow down computation time for more accurate weights. The epochs specify how many times each part of the dataset should pass through the neural network.

3.1.5 Image to Array

Pixel intensities in an image are represented by numerical values. The process of transforming an image into an array and isolating its characteristics, the nd.array module in NumPy is employed. Figure 10 depicts an image from the sad category of the FERC-2013 dataset that has been transformed into a NumPy array.

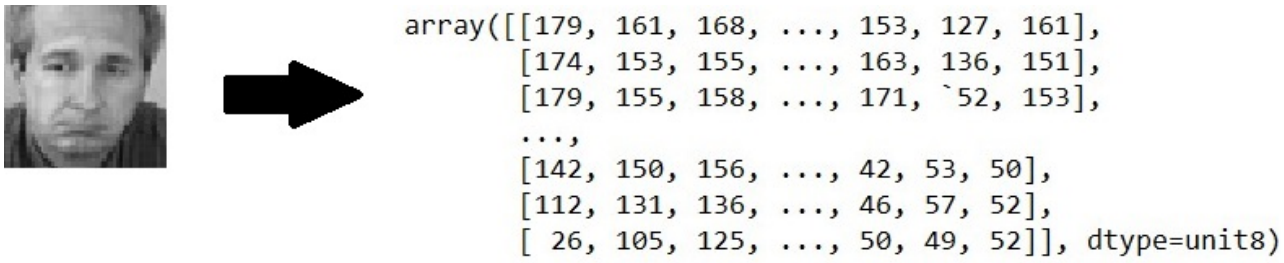

Figure 10: Converted array of an image for Sad person's face

3.1.6 Image to Landmarks

In the process of detecting facial landmarks, the TensorFlow/Keras library is employed for two distinct stages. Firstly, the face within an image is localized using the frontal face detector from Dlib, which generates a rectangle encompassing the face using the coordinates of angles to the upper left and lower right. The Dlib shape predictor is then used to extract the significant facial attributes from the input picture. The shape predictor receives a "landmarks" object, which has two parameters. The first input identifies the picture in which the face landmarks are to be found, whereas the second option indicates the region from which the facial landmarks will be retrieved. The coordinates of the rectangle serve as a reminder of this region.

Figure 11: Detection of Landmarks on a face

3.2 VGG-16 & VGG-19

The VGGNet, also referred to as VGG-16 & VGG-19 models, is a convolution based neural network with 16 & 19 layers respectively, proposed by K. Simonyan and A. Zisserman. This model made it to a test accuracy of 92.7% in ImageNet, which includes over 14 million training images from 1000 object classes, making it one of the top-performing models in the ILSVRC-2014 competition.

VGG16 improved on AlexNet by using a series of smaller 3×3 filters instead of large filters. Unlike AlexNet, which has a kernel size of 11 for the first convolution based layer and 5 for the second layer, VGGNet was trained over several weeks by the researchers using NVIDIA Titan Black GPUs.

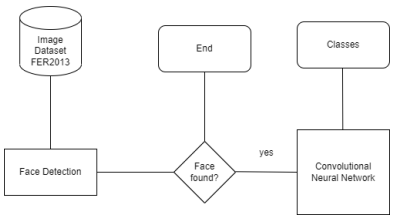

Figure 12: Flowchart of VGG-16 & VGG-19

3.2.1 Architecture

VGG-16 and VGG-19 are convolution based neural networks consisting of 16 and 19 layers, respectively, as indicated by their names. Although they have a massive number of parameters, with a total of 138 million, their architecture is relatively straightforward, and they include the basic features of convolution based neural networks. This simplicity is one of the reasons why the VGGNet16 and VGGNet19 models are still widely used today.

Although the VGGNet architecture is straightforward, it has an extensive number of parameters. The VGG-16 and VGG-19 models have a total of 138 million and 143 million parameters, respectively, making them some of the largest models in terms of parameters, even by today's standards.

The VGGNet architecture's ability to learn advanced features from image input is one of its advantages. With a huge number of parameter, the network can recognize patterns and shapes that may not be discernible to the human eye, making it a valuable tool for image recognition.

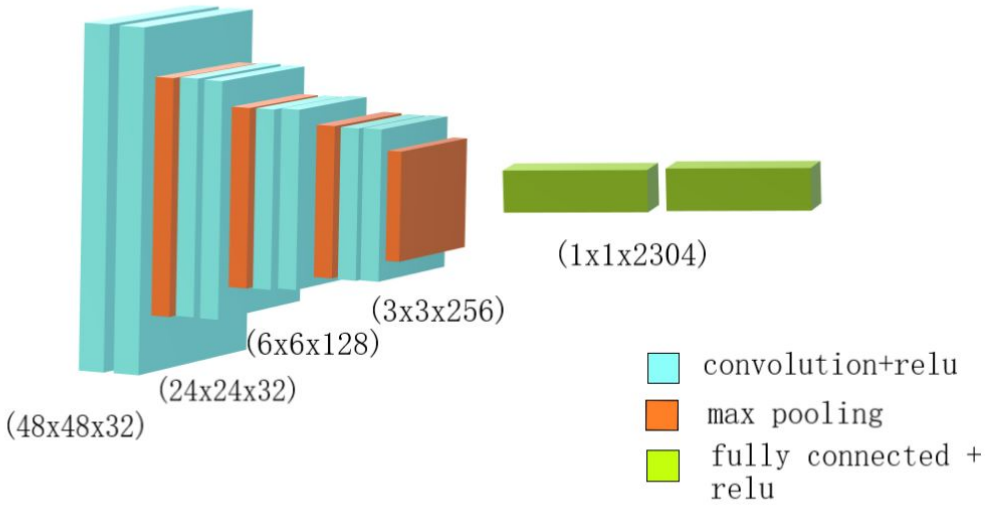

Figure 13: Architecture of VGG-16

The VGGNet architecture consists of small convolution based filters. Specifically, VGG16 is built of 13 convolution based layers and three inter connected layers, while VGG19 consists of 16 convolution based layers and three connected connected layers [41]. Below is a brief overview of the VGG structure:

- **Input -** When using VGGNet, the image that has been input who's size is fixed at 224×224. During ImageNet competition, the developers of the model maintained a consistent input size by selecting a 224×224 portion from the center of every image.
- **convolution based layers -** The filters in VGGNet are intentionally small with a 3×3 receptive field. In addition, VGGNet includes 1×1 convolution filters for linearly transforming the input.

- **ReLU Activation -** Rectified Linear Unit (ReLU) activation function, a crucial advancement introduced in AlexNet that helped to shorten training time, is used in the layer following the input layer in VGG. ReLU is a linear function that, for negative inputs, outputs zero and, for positive inputs, the input value. To retain spatial resolution following convolution, VGG uses a fixed convolution stride of 1 pixel. How many pixels the filter "moves" to fill the whole picture space is controlled by the stride setting.
- **Hidden Layers -** The VGG network differs from AlexNet in that it uses the ReLU activation function in its hidden layers instead of Local Response Normalization. Local Response Normalization was found to have little effect on overall accuracy but significantly increased training time and memory usage.
- **Pooling Layers -** Adding a pooling type layer after several included layers helps to reduce the dimensional integrity and number of limitations in the resulting feature maps. As the number of filters increases from 64 to 128, 256, and ultimately 512 in the final layers, this step becomes even more important.
- **Fully Connected Layers -** The VGG Net design involves three inter connected layer, where the first and second layers have in total 4096 channels each and the last layer has 1000 channels for each & every class.

3.2.2 Configuration, Training & Results

The VGG network has five different variations, labeled A through E, with each configuration having an increasing number of layers from A to B. The main difference between these configurations is the number of layers, with deeper configurations having more layers and thus more parameters to learn. These configurations have proven to be highly effective in various computer vision tasks.

The table below lists all feasible network architectures, and it is worth noting that each configuration can also be modified by changing the number of filters in each layer, the size of the filters, the use of pooling type layers, and other hyper-parameters.

ConvNet Configuration					
A	A-LRN	B	C	D	E
11 weight layers	11 weight layers	13 weight layers	16 weight layers	16 weight layers	19 weight layers
input (224 × 224 RGB image)					
conv3-64	conv3-64 **LRN**	conv3-64 **conv3-64**	conv3-64 conv3-64	conv3-64 conv3-64	conv3-64 conv3-64
maxpool					
conv3-128	conv3-128	conv3-128 **conv3-128**	conv3-128 conv3-128	conv3-128 conv3-128	conv3-128 conv3-128
maxpool					
conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256 **conv1-256**	conv3-256 conv3-256 **conv3-256**	conv3-256 conv3-256 conv3-256 **conv3-256**
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 **conv1-512**	conv3-512 conv3-512 **conv3-512**	conv3-512 conv3-512 conv3-512 **conv3-512**
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 **conv1-512**	conv3-512 conv3-512 **conv3-512**	conv3-512 conv3-512 conv3-512 **conv3-512**
maxpool					
FC-4096					
FC-4096					
FC-1000					
soft-max					

Figure 14: All Possible Network Architecture [41]

The structure of the VGG network remains constant across all versions, with only the number of layers differing. VGGNet16 and VGGNet19 are named according to their layer counts, with VGGNet19 having 19 weight layers and 16 layers compared to VGGNet16, which has 11 weight layers, including 8 layers and 3 connected connected layers. The number of channels in the layers ranges from 64 in the first layer to 512 in the last layer, with a doubling factor applied after every max-pooling type layer. The following graph illustrates the total number of parameters in millions.

Table 2: Quantity of Parameters in Million [41]

Network	E	D	C	B	A, A-LRN
Quantity of Parameters	144	138	134	133	133

The training approach for VGG is similar to AlexNet, as both utilize the optimization of a multinomial logistic regression function for back-propagation. VGG addresses the vanishing gradient problem, which occurs due to the network's depth, by using mini-batches.

During the training process, the size of the batch was established at 256 and momentum at 0.9. The VGG model integrated dropout regularization with a ratio of 0.5 into two connected connected layers. The initial learning rate for the network was set at 0.001 and was reduced by a factor of 10 when the validation set's accuracy ceased to improve. The training concluded after 74 epochs, or 370,000 iterations, with a learning rate that was three times lower [42].

The process of training a VGG network with four NVIDIA Titan Black GPUs on the ILSVRC dataset, which contains approximately 1.3 million training images, took about two to three weeks.

For image classification tasks, the VGG-16 network surpassed earlier models in the ILSVRC-2012 and 2013 contests. The single net performance of the VGG-16 architecture was the most impressive, achieving a test error of 7.0%. The error rates can be found in the following table.

Table 3: Error Rate in ILSVRC challenge [41]

Method	**% of val. error for top-1**	**% of val. error for top-5**	**% of test error for top-5**
VGG (dense eval., multi-crop & 2 nets)	23.7	6.8	6.8
VGG (dense eval., multi-crop & 1 net)	23.4	7.1	7.0
VGG (7 nets, dense eval., ILSVRC submission)	24.7	7.5	7.3

3.3 MobileNet-V2

In 2018, a team of Google researchers including Mark Sandler, Menglong Zhu, Andrew Howard, Liang-Chieh Chen and Andrey Zhmoginov developed a new deep neural network architecture known as MobileNet-V2. This architecture is specifically designed to be lightweight, efficient, and highly accurate for use in mobile and embedded vision applications, such as smartphones, drones, and robots. MobileNet-V2 achieves this through the use of various techniques, including depthwise separable convolutions, linear bottlenecks, and inverted residuals.

3.3.1 Inverted Bottleneck Block

MobileNet-V2 is an updated version of the MobileNet architecture that still utilizes deep separable convolution, but now includes a bottleneck residual block with three convolution layers. In MobileNet-V1, there was only a deep separable convolution block with two layers, consisting of a depth-wise convolution layer and a 1×1 point-to-point convolution layer.

In MobileNet-V2, the 1×1 convolution layer, which is also called the projection layer or bottleneck layer, is utilized to reduce the number of channels, thereby reducing the amount of data that passes through it. The first layer in bottleneck layer, which is used to lower the number of channels and hence lower the volume of data passing through it. The 1×1 expansion layer, which increases the number of channels for the data that passes through it, is the first layer in the bottleneck block. After weighing several architectural trade-offs, the expansion factor, a hyper parameters that controls the degree of the expansion, is chosen.

The depth-wise convolution layer, which is the second layer, was also featured in MobileNet-V1. The bottleneck residual block also has a residual connection that performs similarly to ResNet. Each layer's activation function, ReLU6, is produced by the equation min(max(x, 0), 6).

Additionally, each layer consists of a batch normalisation layer plus the activation function, with the exception of the projection layer, where adding nonlinearity using ReLU6 would reduce performance because the projection layer's output is low dimensional.

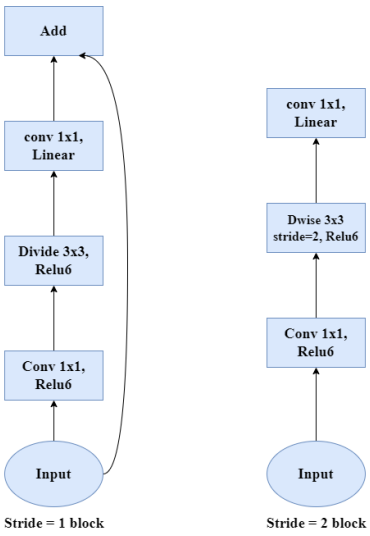

Figure 15: MobileNet-V2 Flowchart [43]

3.3.2 Architecture

Input	Operator	t	c	n	s
$224^2 \times 3$	conv2d	-	32	1	2
$112^2 \times 32$	bottleneck	1	16	1	1
$112^2 \times 16$	bottleneck	6	24	2	2
$56^2 \times 24$	bottleneck	6	32	3	2
$28^2 \times 32$	bottleneck	6	64	4	2
$14^2 \times 64$	bottleneck	6	96	3	1
$14^2 \times 96$	bottleneck	6	160	3	2
$7^2 \times 160$	bottleneck	6	320	1	1
$7^2 \times 320$	conv2d 1x1	-	1280	1	1
$7^2 \times 1280$	avgpool 7x7	-	-	1	-
$1 \times 1 \times 1280$	conv2d 1x1	-	k	-	

Figure 16: MobileNetV2 Architecture [44]

The MobileNet design employs 3×3 kernels for spatial convolution with the help of variables such as the expansion factor (t), the quantity of output channels (c), the number of repetitions (n), and the stride (s). The primary MobileNet network requires 3.4 million parameters and incurs a computational cost of 300 million multiply-adds with a width multiplier of 1 and an input resolution of 224×224. The width multiplier parameter, initially proposed in MobileNet-V1, has been further examined in order to assess performance trade-offs for input resolutions ranging from 96 to 224 and width multipliers between 0.35 and 1.4. [44]

The network's computational cost can be as high as 585 million multiply-adds, while the size of the model has a parameter range of 1.7 million to 6.9 million. The network is trained on 16 GPUs with a batch size of 96.

Size	MobileNetV1	MobileNetV2	ShuffleNet (2x,g=3)
112x112	64/1600	16/400	32/800
56x56	128/800	32/200	48/300
28x28	256/400	64/100	400/600K
14x14	512/200	160/62	800/310
7x7	1024/199	320/32	1600/156
1x1	1024/2	1280/2	1600/3
max	1600K	**400K**	600K

Figure 17: The maximum number of channels and memory in kilobytes for each spatial resolution is listed for various architectures, including 16-bit floats for activation [43]

3.4 CNN-V2 Model

CNNV2 (convolution based Neural Network Version 2) is a deep learning architecture created in 2018 by NVIDIA Corporation with the express motive of performing picture identification and classification tasks. To

extract characteristics from input photos, it first employs convolution based layers, followed by max-pooling type layers, which are flattened and fed into a connected connected layer for ultimate classification [48].

The usage of residual connections, which were first presented in the ResNet architecture, is one of the key advancements in CNN-V2 over the original CNN architecture. By introducing a shortcut connection to skip one or more layers, the network is able to learn residual mappings that make optimising deeper networks easier. Bypassing intermediate layers and connecting the output of one convolution based layer to the input of a subsequent layer using residual connections, CNN-V2 enables the network to acquire more intricate and detailed representations of the input data [48].

Additionally, CNN-V2 incorporates batch normalization, a technique that normalizes inputs to each layer and reduces the internal covariate shift problem that can occur during training. This allows the network to learn more quickly and reduces the need for dropout regularization.

CNN-V2 has demonstrated exceptional performance on various image recognition and classification tasks, including facial recognition, object detection, and scene segmentation. It is also highly efficient in terms of memory usage and computational complexity, making it an excellent choice for deployment on mobile and embedded devices.

3.4.1 Architecture

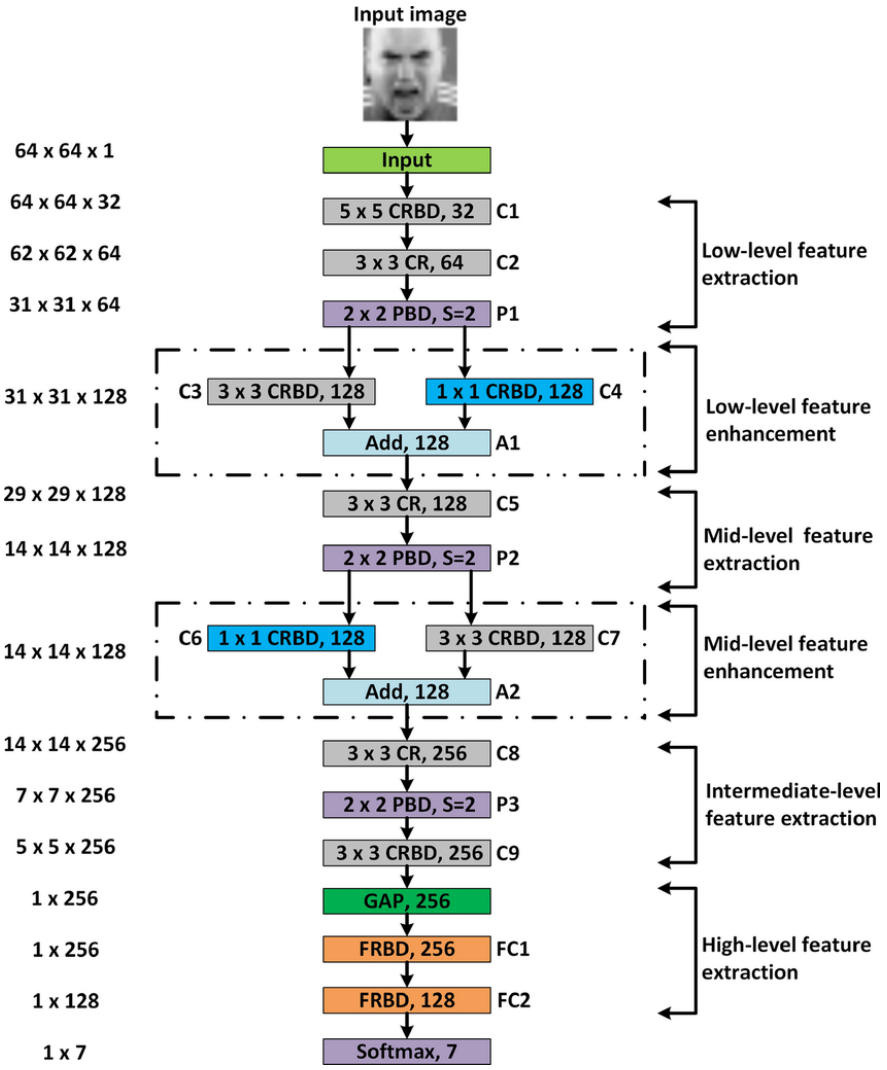

Figure 18: CNN-V2 Architecture [45]

For the purpose of obtaining characteristics from input photos, after each other the batch normalisation and ReLU activation functions convolution-based layers in the CNN-V2 architecture. The batch normalisation

process normalises inputs to each layer, ReLU activation function offers non-linearity to enable the network to learn complicated input data representations, while speeding up training by eliminating the internal covariate shift problem. [45].

One of the main improvements in CNN-V2 over the original CNN design is the use of residual connections, which bypass a number of intermediary levels to connect the result of one convolution-based layer to a succeeding layer's input. This lessens the possibility of vanishing gradients while enabling the network to learn richer and more intricate representations of the input data.

After the convolution based layers, utilising max-pooling, the feature maps are down-sampled. type layers, which lowers dimensionality and boosts network effectiveness. The final classification is performed by flattening the down-sampled feature maps and feeding them into a connected linked layer.

The CNN-V2 design has a number of hyper-parameters that can be changed to enhance network performance, including the size, quantity, and convolution-based layer-based layers used, size and stride of max-pooling type layers, learning rate, and dropout rate [46].

Due to the integration of residual connections, batch normalisation, and other cutting-edge approaches, Scene segmentation, facial identification, and object detection are just a few of the image recognition and classification tasks that CNN-V2 has excelled in. This flexible and effective architecture can be adapted to a variety of image recognition and classification tasks.

3.5 EmoAnalyzer (Proposed Model)

We have created a particular deep learning model called "EmoAnalyzer" that is based on convolution-based neural networks (CNN) in order to distinguish between emotions and facial expressions. The recommended method uses a multi-layer architecture to extract high-level information from facial images, which is then used to categorise the information into seven basic emotions. Because of the potential applications in fields like psychology, medicine, and human-computer interaction, there has been a lot of interest in emotion identification from facial expressions recently. The richness and diversity of facial expressions in humans make it difficult to recognise emotions with accuracy and reliability. To address this challenge, we have developed a CNN-based model that combines convolution based, max pooling, and connected layers to automatically learn identifying characteristics in face photographs and improve emotion recognition performance. Our proposed model has been evaluated on multiple bench-mark data sets and achieved state of the art performance compared to existing methods. The results represents the efficiency and potential of the proposed CNN-based model for emotion recognition from facial expressions.

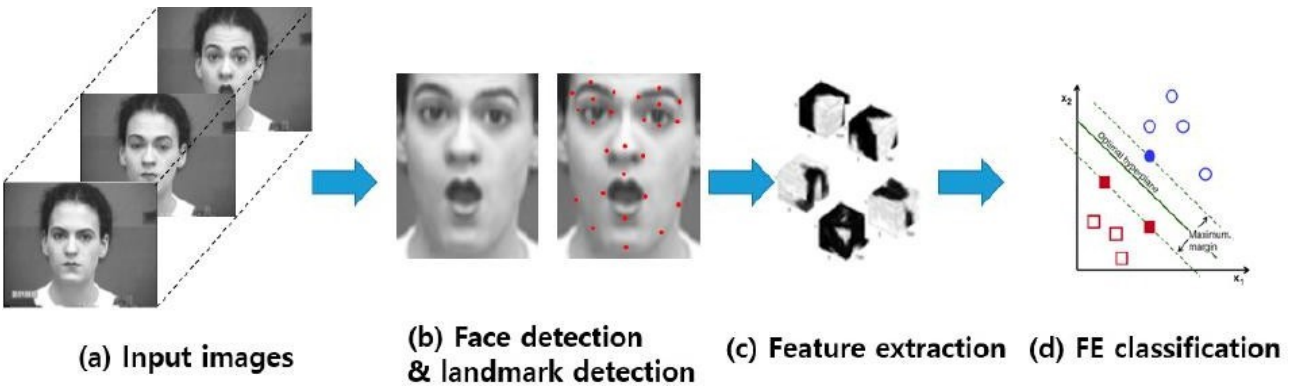

Figure 19: FER process for a picture

Figure 19 provides a visual representation of the FER process, which includes the detection of the face region and facial landmarks in an input image by the EmoAnalyzer model.

Facial features are specific points on the surface, that might used to determine the position and shape of facial features. These landmarks are prominent visual points such as the nose tip, eyebrow edges, and mouth corners, as shown in Figure 20. The EmoAnalyzer model uses a combination using deep learning and computer vision techniques to detect these landmarks accurately. Once the landmarks have been detected, they can be

used to derive a variety of features that are relevant for FER.

The features that are derived from facial landmarks can be categorized into two types: pairwise positions and local texture. Pairwise positions refer to the relative distances and angles between two landmark points. For example, the distance between the eyebrow edges can be used to determine the degree of eyebrow raise, which is a common expression associated with surprise. Similarly, the angle between the nose tip and mouth corners can be used to detect the presence of a smile.

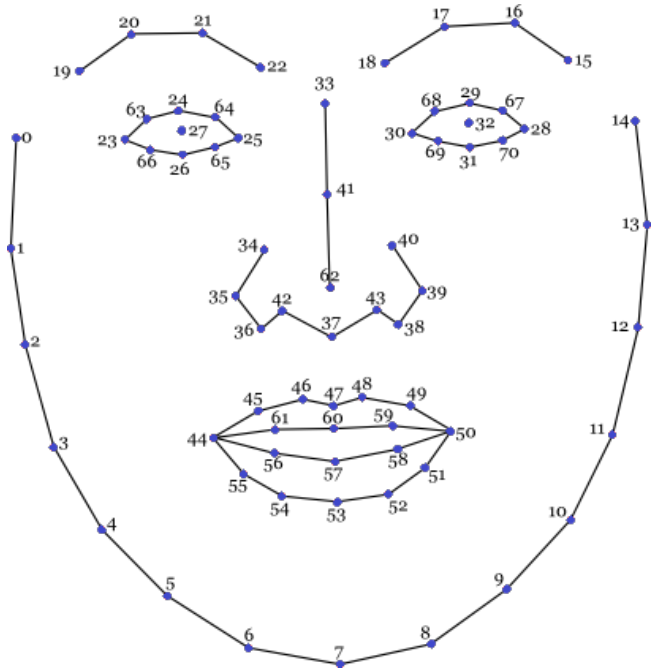

Figure 20: Extracting facial landmarks from a face

The 64 major and secondary landmarks are described in Table 4. Following the extraction of the face's spatial and temporal features, pattern classifiers are used to select the facial category that best represents the facial expression.

Table 4: Definitions of 64 primary and secondary landmarks

Primary landmarks		Secondary landmarks	
Symbol	Representation	Symbol	Representation
16	Outer edge of left side eyebrow	1	Temple on the left side
19	Inner edge of left side eyebrow	8	Tip of Chin
22	Inner edge of right side eyebrow	2-7,9-14	Cheek shape
25	Outer edge of right side eyebrow	15	Temple on the right side
28	Outer edge of left side eye	16-19	Left side eyebrow shapes
30	Inner edge of left side eye	22-25	Right side eyebrow edge
32	Inner edge of right side eye	29,33	Centres of upper eyelid
34	Outer edge of right side eye	31,35	Centres of lower eyelid
41	Tip of Nose	36,37	Saddle of nose
46	Left edge of mouth	40,42	Peak of Nose (nostrils)
52	Right edge of mouth	38-40,42-45	Nose shapes
63,64	Centres of eye	47-51,53-62	Mouth shapes

3.5.1 Architecture

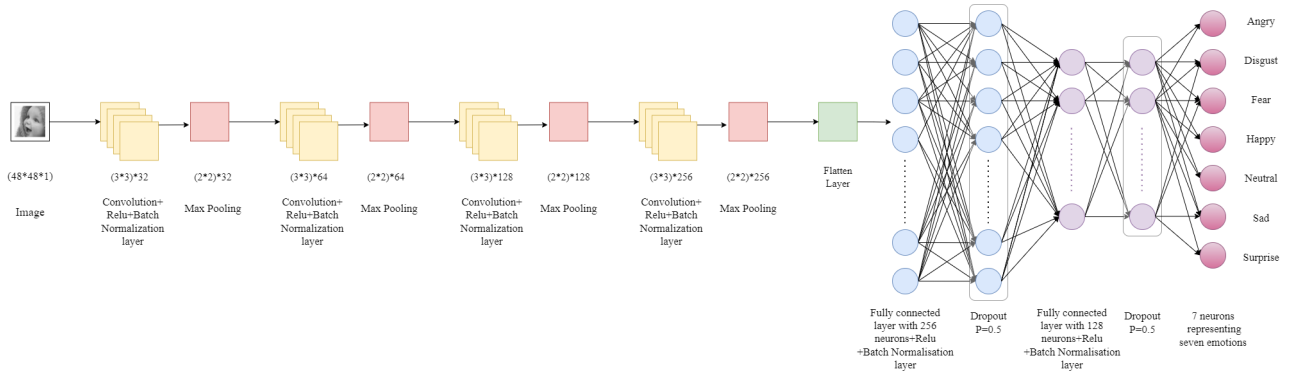

Figure 21: Proposed Model Architecture

A convolution-based neural network (CNN) architecture is the suggested model for extracting emotions from facial images. The Keras deep learning framework and TensorFlow were used to create the model.

The input layer, or first layer, in the model architecture is either (48, 48, 1) for grayscale images or (48, 48, 3) for colour images. The first layer of the network, a convolution based layer with 32 filters of the particular size 3×3, receives data from the input layer. The output of this layer is passed through the batch of normalisation layer and a Rectified Linear Unit (ReLU) activation of a training-enhancing function stability and performance.

The next step is to add a layer of the maximum pooling kind with a pool of 2x2, which aids in reducing the feature maps' spatial dimensions while preserving the most crucial data. The technique is then repeated for the following two convolution-based layers, which have 64 and 128 filters, respectively, and Batch normalisation layers and ReLU activation functions come after each other. A max pooling type layer follows each convolution-based layer having a pool of 2×2.

The fourth and final convolution based layer in the model has 256 filters of size 3×3, followed by a batch normalisation layer and a ReLU activation function.Additionally, a layer of the maximum pooling type with a pool size of 2×2.

The output of the last max pooling type layer is flattened into a single dimension vector before being sent through two layers of 256 and 128 neurons apiece, each coupled to the previous layer. To avoid overfitting, a dropout layer that after each layer that is fully linked, a dropout rate of 0.5 is introduced. After the dropout layer, a batch normalisation layer and a ReLU activation function are added.

To categorise the input picture into one of the emotions, a softmax activation function and an output layer with seven neurons representing each of the seven emotions are added.

The Adam optimizer, a well-known Stochastic Gradient Descent (SGD) variation with adjustable learning rates, is used to build the model. The categorical cross-entropy loss function is employed during training to improve the model.

Overall, the proposed model for emotion recognition is a powerful CNN architecture that is capable of accurately classifying emotions from facial images. The use of ReLU based activating functions & batch normalizing layers & dropout layers aids in enhancing the model's stability and performance, while the Adam based optimizing & categorical cross entropy type loss functioning are widely recognized as effective choices for deep learning tasks.

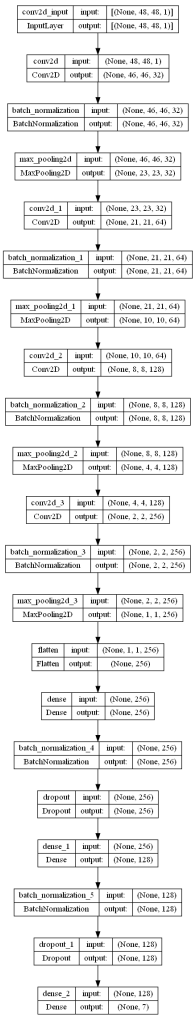

Figure 22: EmoAnalyzer (Proposed model) Stack

3.5.2 Mathematics

Conv Layers: The EmoAnalyzer model places significant emphasis on the convolution based layer, which is responsible for extracting features from the input data. Unlike traditional matrix multiplication, this layer utilizes the convolution operation (denoted by *), which serves as the fundamental building block of CNN. The layer is comprised of a series of learnable filters, or kernels, that make up its parameters. Its primary function is to identify features within specific regions of the input image and generate a feature map accordingly.

This method involves moving a filter of a specified size (3×3) across the input image to generate the output of the next layer $(m^{[2]} \times m^{[3]})$. The resulting activation maps contain local distinctive features that are identified through the convolution process.

Each layer in the convolution based network has a filter $(m^{[1]})$, which generates $m_1^{(l)}$ feature maps in $Y_i^{(l)}$ of size $m_2^{(l)} \times m_3^{(l)}$ in layer l. The calculation of the ith feature map $(Y_i^{(l)})$ is done using the following eqn. (3), where $K_{i,j}^{(l)}$ is the filter and $B_i^{(l)}$ is the bias matrix:

$$Y_i^{(l)} = f(B_i^{(l)} + \sum_{j=1}^{m_i^{l-1}} K_{i,j}^{(l)} \times Y_j^{l-1}) \tag{3}$$

ReLU Operation: The implementation of ReLU in convolution based networks is similar to that in connected connected networks. The ReLU function, denoted as g, is applied to each element of $z^{[1]}$, resulting in $a^{[1]}$, which has the same dimensions as $z^{[1]}$, i.e., (3,3,32) ReLU function g is given in the eqn. (4),

$$a^{[1]} = g(z^{[1]}) \tag{4}$$

Having done so, we are currently at the first hidden layer $a^{[1]}$ from the input layer x.

Pooling Layer: The paper focuses on the transition between a convolution based layer ($a^{[2]}$) and the first pooling type layer ($m^{[2]}$) in the architecture. The input $a^{[2]}$ has dimensions (48, 48, 32) and the output $m^{[2]}$ has dimensions (24, 24, 32). The proposed model uses max pooling, where a sliding window is applied to the input and the maximum value within the window is taken. The mathematical equation for $m^{[2]}$ (h, w, c) is given in the eqn. (5):

$$m^{[2]}_{(h,w,c)} = \max_{h*s<=l<h*s+f, w*s<=l<=w*s+f} a_{(l,m,c)} \tag{5}$$

Here, h represents the height, w represents the width, and c represents the channel of the output $m^{[2]}$. l and m are the height and width of the input [47]. The max pooling filter has height and width equal to f, and s is the stride size, indicating how many elements are passed over for the operation. In this case, a stride size of 2 is used. Eqn (6) is used to determine the window size for max pooling which is given as,

$$n_{L+1} = \lfloor \frac{n_L + 2p - f}{s} + 1 \rfloor \tag{6}$$

Where n_L is the height and width of the input, n_{L+1} is the height and width of the output, and p is the padding (which is not used in this case). Solving for f, we get eqn. (7),

$$f = n_L + 2p - s(n_{L+1} - 1) \tag{7}$$

By plugging in n_L=48, s=2, and n_{L+1}=24, we get eqn. (8),

$$f = 48 + 2 * 0 - 2(24 - 1) \tag{8}$$

$$f = 2 \tag{9}$$

Therefore, to get an element in $m^{[2]}$, a 2×2 window of $a^{[2]}$ (from eqn. (9)) is taken and the maximum value in that window is returned. The window is then slid over by 2 and the operation is repeated.

Flat Layer: To create a Flat Layer, we use the output from the final max pooling type layer ($m^{[4]}$) as input and flatten it. This results in a flat layer $f^{[4]}$ with dimensions of (None, 256), where 256 is obtained by multiplying all the dimensions of the input layer ($1 \times 1 \times 256 = 256$). The motive of flattening the layer is to convert it into a row or column vector, which is required for connected connected layers to process it. Essentially, this layer just changes the dimension of the data to make it suitable for use in connected connected layers, and nothing complicated happens in this step.

Fully Connected Layer: After applying 4 convolution based layers and pooling, we use 2 connected connected layers by connecting the flattened layer to them. The convolution based and pooling type layers are responsible for creating meaningful data representations by taking into account local windows of the input. On the other hand, connected connected layers are global and connect all values of the previous pooling type layer ($m^{[4]}$). Finally, we connect the last connected connected layer to the output layer ($\hat{y}$) using the softmax function for the transition.

Softmax Function: Moving from the connected connected layer to the softmax layer follows the usual process of a connected connected layer. We achieve z by performing a matrix multiplication using W and adding a bias term b. Here, the dimensions of W are (7, 128) and the dimensions of b are (7, 1), which is appropriate since the connected connected layer is a row vector with dimensions (128, 1) and z is also a row vector with dimensions (7, 1).

We will use the symbol (σ) to represent the softmax function. To calculate the ith element in the fourth layer (a[4]), we perform the steps given in the eqn. (10-11),

$$a^{[4]}_{i,1} = \sigma_{i,1}(z^{[16]}) \tag{10}$$

$$a^{[4]}_{i,1} = \frac{e^{z^{[4]}_{i,1}}}{\sum_{j=1}^{1000} e^{z^{[4]}_{j,1}}} \tag{11}$$

To calculate the ith element of $a^{[4]}$, we use the exponential function e to raise the power of $z^{[4]}$ and divide the result by the sum of e raised to the power of all the elements in z. After the application of the softmax activation function, we obtain a normalized vector $a^{[4]}$ whose elements sum to 1 [47]. It is worth mentioning that $a^{[4]}$ is equivalent to y, which is the predicted probability distribution over the seven emotions.

$$\hat{y} = \begin{bmatrix} \hat{y}_1 \\ \hat{y}_2 \\ \hat{y}_3 \\ \hat{y}_4 \\ \hat{y}_5 \\ \hat{y}_6 \\ \hat{y}_7 \end{bmatrix} \tag{12}$$

Each value of $\hat{y}_i$ in eqn. (12) represents the probability that the input x belongs to class i (emotion), considering a classification problem where there are multiple classes to choose from.

$$\hat{y}_i = P(y = i|\mathbf{x}) \tag{13}$$

Once the softmax activation function is applied, we obtain a probability vector (eqn. (13)) in which all the elements add up to 1.

3.5.3 Algorithm

The EmoAnalyzer model is a deep learning architecture designed for the task of emotion recognition. It is built on a convolution based neural network (CNN) and consists of several layers that work together to extract meaningful features from input images and make predictions about the emotions they represent.

The model begins with an input layer of size (48, 48, 1) for grayscale images or (48, 48, 3) for color images. This layer receives the raw pixel values of the input images and passes them through to the next layer, which is a convolution based layer. Filters with a 3×3 kernel size are used in the convolution-based layer to extract features from the input pictures. Each succeeding convolution-based layer contains twice as many filters as the one before it, increasing to 64, 128, and 256 filters respectively. The initial convolution-based layer includes 32 filters.

After each convolution based layer, in order to provide non-linearity and improve the model's capacity to learn intricate correlations, the ReLU activation function is used. After each convolution-based layer, batch normalisation layers are also included to assist stabilise the training process and increase the model's capacity for generalisation.

A max pooling type layer with a pool size of 2×2 is added after each convolution-based layer to help the feature maps' spatial dimensions. The model's restrictions are less as a result of the down-sampling technique, which also makes the model less susceptible to overfitting.

Two fully connected layers with 256 and 128 neurons each receive the output of the final max pooling layer after it has been flattened. ReLU activation functions and batch normalisation layers are introduced after each fully linked layer, much like the convolution-based layers. To avoid overfitting, dropout layers with a dropout rate of 0.5 are introduced after each fully linked layer.

The output layer, which includes seven neurons to reflect the seven various emotions, serves as a representation of the model's conclusion. The softmax activation function is then implemented on output layer, normalising the outputs to represent probability sums of 1. The outcome with the highest likelihood determines the anticipated feeling.

The Adam optimizer, a technique for adaptive learning rate optimisation that works well with big data sets and high-dimensional restriction spaces, is used to train the model. The difference between expected and real emotions is calculated using the categorical cross-entropy loss function. In multi-class classification settings, this loss function is frequently employed when the model is attempting to predict one of several potential outcomes. During training, the model makes adjustments to its limitations in an effort to lessen the category cross-entropy loss.

Algorithm 1 EmoAnalyzer Model Algorithm

Input: $X \leftarrow$ Facial images in $48 \times 48 \times 1$ dimension
Output: $Y \leftarrow$ categories (Angry, Disgust, Fear, Happy, Neutral, Sad, Surprise)
$X \leftarrow X/255$ (normalize input data)
//————————Making training and validation data————————
training_set ← image_generator.flow_from_directory(batch_size=128, directory=train_dir, shuffle=True, target_size='categorical', color_mode='grayscale')
testing_set ← image_generator.flow_from_directory(batch_size=128, directory=val_dir, shuffle=True, target_size='categorical', color_mode='grayscale')
proposed ← EmoAnalyzer()
layer_list ← *proposed.layers*
//————————Build EmoAnalyzer model————————
model ← *Sequential*()
for *layer in layer_list* **do**
    model ← *layer*
end for
for *layer in model.layers* **do**
    layer.trainable = *False*
end for
numberOfClass ← 7
model ← *Conv2D*(32)
model ← *Conv2D*(64)
model ← *Conv2D*(128)
model ← *Conv2D*(256)
model ← *Flatten*()
model ← *Dense*(256)
model ← *Dense*(128)
//————————Perform training and testing of EmoAnalyzer model————————
early_stopping ← EarlyStopping(monitor='val_accuracy', min_delta=0.01, patience=5, verbose=1, restore_best_weights=True)
reduce_learningrate ← ReduceLROnPlateau(monitor='val_accuracy', factor=0.2, patience=3, verbose=1, min_delta=0.0001)
callbacks_list ← [early_stopping, checkpoint, reduce_learningrate]
model.compile(loss='categorical_crossentropy', optimizer=Adam(lr=0.01),metrics=['accuracy'])
history ← model.fit(generator = training_set, steps_per_epoch = training_set.n//training_set.batch_size, epochs = 48, validation_data = testing_set, validation_steps = testing_set.n//testing_set.batch_size, callbacks=callbacks_list)
Calculate performance metrics (recall, accuracy, precision, f1-score, confusion matrix for each epoch and emotion

4 Result & Discussion

4.1 Experimental Condition

The experiment were performed in Visual Studio Code version 1.77.1 using Python version 3.11.1. The specification of hardware configuration is Intel Core i7-1065G7 processor with 16.0 GB RAM and NVIDIA GeForce MX330 GPU.

4.2 Dataset

The FERC-2013 dataset gained popularity and was utilized in the Kaggle competition. However, there are certain issues with this dataset that need to be addressed before inputting it to the CNN. One of the major challenges is that the input to the model should be an array of numbers, which means that images must be

converted into arrays. These are a few dataset problems:

1. **Imbalance:** Imbalance in data refers to a scenario where one class of data has a much larger number of images than another class, which can create bias towards that specific class in the model. For example, if a model has 2000 images of happy expressions but only 500 images of fearful expressions, it is likely to be biased towards the happy expression. To overcome this issue, data augmentation techniques such as cropping, padding, and horizontal flipping are used to increase the quantity of data.
2. **Contrast variation:** The collection of images may include some that are too dark or too bright. Since images have visual data, those with more contrast contain more information than those with less contrast. A CNN receives images as input, discovers image characteristics automatically, and categorizes them into result categories. As a result, changes in image contrast can impact the CNN's effectiveness. To address this issue, the solution is to adjust the images to emphasize facial features.
3. **Intra-class variation:** There are some images in the dataset that aren't real pictures of human faces, but instead are depictions or animations of faces. These types of faces have varying features that differ from real human faces, causing the model to struggle when identifying key facial features. To improve the model's accuracy, it is suggested to remove all non-human face images from the dataset.
4. **Occlusion:** Occlusion is the term used to describe when a portion of an image is hidden from view. This can happen when something, like a hand, blocks the nose or the right eye on the face. Occlusion also includes using sunglasses or a mask. According to Table 1, the nose and eyes are two essential face features for emotional interpretation. In order to accurately identify emotions from occluded images, it is essential to remove them from the dataset.

The 35,000 photos in the FERC-2013 dataset were rigorously reviewed to guarantee that the training images are unaffected by the aforementioned issues. In the end, 7,074 photographs were chosen from five different categories: 966 were chosen for furious, 859 for fear, 2477 for joyful, 1466 for neutral, and 1326 for sad.

4.3 Python Libraries Used

NumPy: An open-source Python package called NumPy works with matrices and arrays. In NumPy, the array object is called nd.array. CNN inputs are numerical arrays, and by converting photos into NumPy arrays using NumPy, matrix multiplication and other CNN operations are made simple.

OpenCV: For computer vision, machine learning, and image processing, there is an open-source library called OpenCV. To recognise faces, objects, and handwriting in photos and videos, use OpenCV. OpenCV can analyse array structures if it is connected with a library like NumPy. On these array structures, mathematical operations are carried out for pattern recognition.

TensorFlow: An open-source, cost-free software library is called TensorFlow. It can programme using differentiable programming and dataflow for a variety of applications. The library, which is symbolic in nature, is frequently used for neural networks and other machine learning applications. At Google, it is utilised for both research and manufacturing.

Keras: Python-based Keras is an open-source neural network library. R, Theano, PlaidML, Microsoft Cognitive Toolkit, TensorFlow, and others can all be used as a foundation for it. With a focus on being user-friendly, modular, and extendable, Deep neural network experiments can be done quickly thanks to Keras, it offers a simple structure that makes it simple to build TensorFlow or Theano-based deep learning models. For applications involving deep learning, it is the best option.

4.4 Result Metric

To evaluate the facial emotion recognition system established from images, three metrics were used: accuracy, specificity, and sensitivity. Accuracy was calculated by dividing the percentage of data that was successfully classified to all data. The true positive (TP), true negative (TN), false positive (FP), and false negative (FN) performance attributes were defined using the following eqn. (14-17).

$$Accuracy = \frac{TN + TP}{TP + FP + FN + TN} \tag{14}$$

$$Precision = \frac{TP}{TP + FP} \tag{15}$$

$$Recall = \frac{TP}{TP + FN} \tag{16}$$

$$F1\text{–}\ Score = 2 * \frac{Precision * Recall}{Precision + Recall} \tag{17}$$

4.5 Result Analysis

Following training, for EmoAnalyzer model, while the validation accuracy is around 70%, the maximum and average training accuracy is approximately 97% and 90% respectively. Also, for VGG-16 model, the validation accuracy is around 45%, while the training accuracy is approximately 47%. Similarly for VGG-19 model, the validation accuracy is around 41%, while the training accuracy is approximately 50%. Likewise for CNN-V2 model, the validation accuracy is around 25%, while the training accuracy is approximately 35%. And at last, for MobileNet-V2, the validation accuracy is around 54%, while the training accuracy is approximately 76% which is shown in figure 23 as a plot and data values in table 6.

Table 6 provides the average precision, recall, F1 score and support of the application for training set. All the model generates some false-negative values, as indicated by the average precision and recall value. The average support of all the models is obtained as 3593.42. EmoAnalyzer model has average precision, recall and F1-score as 0.144. Whereas, VGG-16 model has average precision and recall as 0.142 and F1-score score as 0.130. Similarly, VGG-19 model has average precision as 0.150, recall as 0.145 and F1-score score as 0.132. CNN-V2 model has average precision as 0.125, recall as 0.140 and F1-score score as 0.080. At last MobileNet-V2 model has average precision as 0.134, recall as 0.142 and F1-score score as 0.098.

Table 7 lists the application's average accuracy, recall, F1 score, and support for the testing set. The average accuracy and recall value show that the model consistently produces some false-negative values. The whole models' average support is determined to be 1009.42. The average accuracy, recall, and F1-score for the EmoAnalyzer model are each 0.600. While the average precision, recall, and F1-score for the VGG-16 model are 0.544, 0.434, and 0.424, respectively. Similar to this, the VGG-19 model has an average accuracy, recall, and F1-score of 0.158, 0.147, and 0.134 respectively. The average accuracy, recall, and F1-score for the CNN-V2 model are 0.140, 0.145, and 0.085, respectively. In conclusion, the MobileNet-V2 model has an average F1-score score of 0.097, recall of 0.140, and accuracy of 0.122 on average.

Table 5: Hyperparameters used in different FER models

Parameters	**EmoAnalyzer**	***VGG-16***	***VGG-19***	***CNN-V2***	***MobileNet-V2***
No. of epoches	48	48	48	48	48
No. of connected layer	5	3	4	2	2
Optimizer	Adam	Adam	Adam	Adam	Adam
Learning rate	0.01	0.001	0.001	0.001	0.001
Population Size	35887	35887	35887	35887	35887

Table 6: Result of Accuracy & Loss conducted during Training & Validation Process (Last 5 Epochs)

Models	Epoch	*Loss*	*Accuracy*	*Validation Loss*	*Validation Accuracy*
EmoAnalyzer	***1***	0.4559	0.8482	1.2236	0.6902
	2	0.4518	0.8714	1.2247	0.6913
	3	0.4468	0.9031	1.2255	0.6925
	4	0.4507	0.9305	1.2262	0.6909
	5	0.4507	0.9705	1.2283	0.6915
	Avg	**0.4511**	**0.9047**	**1.2256**	**0.6912**
VGG-16	***1***	1.7918	0.3117	1.7780	0.3537
	2	1.4503	0.4469	1.5398	0.3827
	3	1.2878	0.5081	1.3219	0.5026
	4	1.2042	0.5402	1.1910	0.5401
	5	1.1404	0.5638	1.3910	0.4697
	Avg	**1.3749**	**0.47414**	**1.4443**	**0.4497**
VGG-19	***1***	1.7414	0.3864	1.6473	0.3414
	2	1.6641	0.4345	1.6263	0.3706
	3	1.6442	0.4750	1.6211	0.3783
	4	1.6235	0.5574	1.6022	0.4590
	5	1.6132	0.6145	1.5925	0.5148
	Avg	**1.6572**	**0.4935**	**1.6178**	**0.3611**
CNN-V2	***1***	6.7606	0.2106	3.6873	0.2583
	2	2.7758	0.2664	2.7702	0.2583
	3	2.1908	0.3633	2.7838	0.2584
	4	2.1225	0.4360	2.6645	0.1946
	5	2.0922	0.4715	2.5764	0.3149
	Avg	**3.1883**	**0.3495**	**2.8964**	**0.2569**
MobileNet-V2	***1***	0.9645	0.6553	2.1352	0.5450
	2	0.9910	0.7141	1.6804	0.5925
	3	0.9844	0.7744	1.6105	0.6547
	4	1.0145	0.8229	1.9380	0.7109
	5	0.9637	0.8556	2.0984	0.7450
	Avg	**0.9836**	**0.7644**	**1.8925**	**0.5416**

The training and validation plots in figure 23 of five different models, namely EmoAnalyzer Model, VGG-16 Model, VGG-19 Model, CNN-V2 Model, and Mobile Net V2 Model, have been presented.The charts show how well these models perform in terms of accuracy and loss relative to the total number of epochs. The right figure displays the trend of accuracy across epochs, whereas the left plot shows the display of loss over the same epochs.

The EmoAnalyzer model outperforms the other four models in terms of accuracy, as shown by the charts, since it achieves higher accuracy on both the training and validation sets. The EmoAnalyzer model likewise shows the lowest loss on both sets of data.

Comparing the VGG-16 and VGG-19 models, it can be seen that the VGG-19 model outperforms the VGG-16 model in terms of both accuracy and loss. However, both VGG models are outperformed by the EmoAnalyzer model.

The CNN-V2 model exhibits a moderate level of performance, with its accuracy and loss falling between the VGG models and the MobileNet-V2 model. Finally, the MobileNet-V2 model shows the poorest performance among the five models, as it has the highest loss and the lowest accuracy on both the training and validation sets.

Overall, the EmoAnalyzer model appears to be the best performing model among the five models considered in this study, followed by the VGG-19 model, the CNN-V2 model, the VGG-16 model, and the MobileNet-V2 model, in that order.

Table 7: Classification Report of the Training Set

Models	Emotion	Precision	Recall	F1-Score	Support
EmoAnalyzer	**Surprise**	0.12	0.11	0.12	3205
	Disgust	0.02	0.01	0.01	436
	Neutral	0.18	0.18	0.18	4982
	Angry	0.13	0.09	0.11	3993
	Happy	0.25	0.26	0.25	7164
	Fear	0.14	0.13	0.14	4103
	Sad	0.17	0.19	0.18	4938
	Avg	**0.144**	**0.144**	**0.144**	**3593.42**
VGG-16	**Surprise**	0.11	0.18	0.14	3205
	Disgust	0.02	0.00	0.00	436
	Neutral	0.18	0.13	0.15	4982
	Angry	0.14	0.08	0.10	3993
	Happy	0.24	0.11	0.15	7164
	Fear	0.14	0.15	0.14	4103
	Sad	0.17	0.35	0.23	4938
	Avg	**0.142**	**0.142**	**0.130**	**3593.42**
VGG-19	**Surprise**	0.11	0.10	0.10	3205
	Disgust	0.09	0.02	0.01	436
	Neutral	0.17	0.14	0.16	4982
	Angry	0.12	0.04	0.06	3993
	Happy	0.25	0.43	0.32	7164
	Fear	0.14	0.07	0.09	4103
	Sad	0.17	0.22	0.19	4938
	Avg	**0.150**	**0.145**	**0.132**	**3593.42**
CNN-V2	**Surprise**	0.05	0.00	0.00	3205
	Disgust	0.00	0.00	0.00	436
	Neutral	0.15	0.05	0.08	4982
	Angry	0.13	0.00	0.00	3993
	Happy	0.24	0.16	0.19	7164
	Fear	0.14	0.00	0.01	4103
	Sad	0.17	0.77	0.28	4938
	Avg	**0.125**	**0.140**	**0.080**	**3593.42**
MobileNet-V2	**Surprise**	0.11	0.45	0.18	3205
	Disgust	0.00	0.00	0.00	436
	Neutral	0.17	0.14	0.15	4982
	Angry	0.10	0.00	0.00	3993
	Happy	0.25	0.37	0.30	7164
	Fear	0.13	0.02	0.03	4103
	Sad	0.18	0.02	0.03	4938
	Avg	**0.134**	**0.142**	**0.098**	**3593.42**

Table 8: Classification Report of the Testing Set

Models	Emotion	Precision	Recall	F1-Score	Support
EmoAnalyzer	**Surprise**	0.75	0.68	0.71	797
	Disgust	0.75	0.43	0.55	111
	Neutral	0.52	0.54	0.53	1216
	Angry	0.50	0.49	0.50	960
	Happy	0.80	0.80	0.80	1825
	Fear	0.45	0.39	0.42	1018
	Sad	0.43	0.52	0.47	1139
	Avg	**0.600**	**0.550**	**0.568**	**1009.42**
VGG-16	**Surprise**	0.51	0.83	0.63	797
	Disgust	0.69	0.10	0.17	111
	Neutral	0.54	0.38	0.45	1216
	Angry	0.48	0.27	0.34	960
	Happy	0.96	0.45	0.62	1825
	Fear	0.32	0.35	0.33	1018
	Sad	0.31	0.66	0.43	1139
	Avg	**0.544**	**0.434**	**0.424**	**1009.42**
VGG-19	**Surprise**	0.11	0.12	0.11	797
	Disgust	0.11	0.01	0.02	111
	Neutral	0.18	0.15	0.16	1216
	Angry	0.14	0.03	0.04	960
	Happy	0.26	0.44	0.33	1825
	Fear	0.14	0.08	0.10	1018
	Sad	0.17	0.20	0.18	1139
	Avg	**0.158**	**0.147**	**0.134**	**1009.42**
CNN-V2	**Surprise**	0.10	0.00	0.01	797
	Disgust	0.00	0.00	0.00	111
	Neutral	0.19	0.07	0.10	1216
	Angry	0.15	0.00	0.00	960
	Happy	0.26	0.18	0.21	1825
	Fear	0.12	0.00	0.01	1018
	Sad	0.16	0.77	0.27	1139
	Avg	**0.140**	**0.145**	**0.085**	**1009.42**
MobileNet-V2	**Surprise**	0.11	0.44	0.17	797
	Disgust	0.00	0.00	0.00	111
	Neutral	0.17	0.13	0.15	1216
	Angry	0.00	0.00	0.00	960
	Happy	0.26	0.37	0.30	1825
	Fear	0.14	0.02	0.03	1018
	Sad	0.18	0.02	0.03	1139
	Avg	**0.122**	**0.140**	**0.097**	**1009.42**

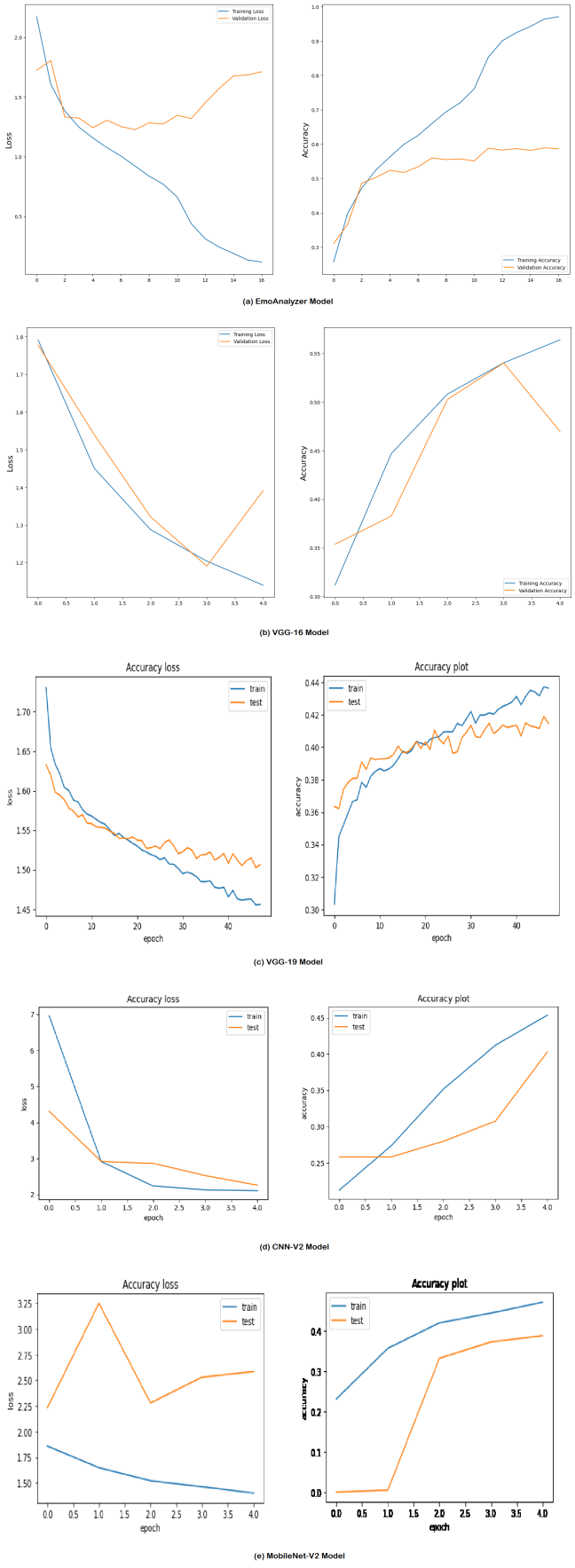

Figure 23: These are the training and validation plots. Loss vs epochs on the left & accuracy vs epochs on the right of (a) EmoAnalyzer Model (b) VGG-16 Model (c) VGG-19 Model (d) CNN-V2 Model (e) MobileNet-V2 Model

4.6 VGG-16 Model

4.6.1 Training set

Table 9: Confusion Matrix of Training Set

	Surprise	Disgust	Neutral	Angry	Happy	Fear	Sad
Surprise	264	8	483	350	384	1138	578
Disgust	45	1	68	40	55	147	80
Neutral	358	9	771	526	654	1734	880
Angry	309	7	591	450	515	1420	701
Happy	548	10	1169	794	867	2401	1375
Fear	303	11	615	485	528	1413	748
Sad	383	9	743	622	555	1739	887

From Table 9, 309 training images of angry facial expression were successfully identified as representing the angry emotion, 1 training image of disgust expression was successfully identified as representing the disgust emotion, 615 training images of fear expression was successfully identified as representing the fear emotion, 794 training images of happy expression was successfully identified as representing the happy emotion, 654 training images of neutral expression was successfully identified as representing the neutral emotion, 1739 training images of sad expression was successfully identified as representing the sad emotion and 578 training images of surprise expression was successfully identified as representing the surprise emotion.

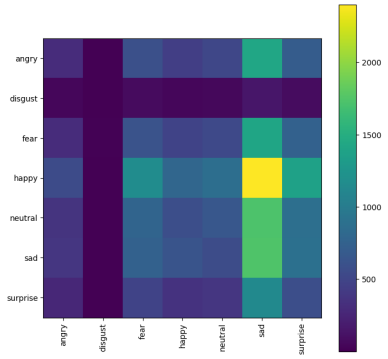

Figure 24: Confusion Matrix Plot for Training Set

4.6.2 Testing set

Table 10: Confusion Matrix of Testing Set

	Surprise	Disgust	Neutral	Angry	Happy	Fear	Sad
Surprise	4	0	77	10	14	34	658
Disgust	18	11	26	1	5	42	8
Neutral	28	0	149	14	463	471	91
Angry	257	3	163	3	106	320	108
Happy	113	1	189	829	99	469	125
Fear	75	1	357	5	52	304	224
Sad	35	0	158	3	121	754	68

From Table 10, 257 testing images of angry facial expression were without any hindrance identified as representing the angry emotion, 11 testing image of disgust expression was without any hindrance identified as representing the disgust emotion, 357 testing images of fear expression was without any hindrance identified as representing the fear emotion, 829 testing images of happy expression was without any hindrance identified as representing the happy emotion, 463 testing images of neutral expression was without any hindrance identified as representing the neutral emotion, 754 testing images of sad expression was without any hindrance identified as representing the sad emotion and 658 testing images of surprise expression was without any hindrance identified as representing the surprise emotion.

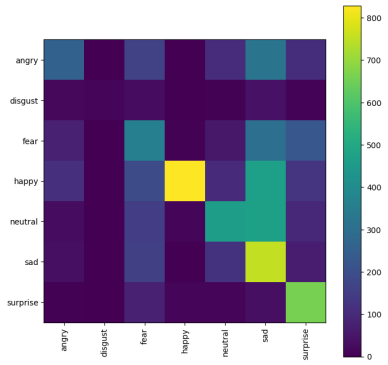

Figure 25: Confusion Matrix Plot for Testing Set

4.7 VGG-19 Model

4.7.1 Training set

Table 11: Confusion Matrix of Training Set

	Surprise	**Disgust**	**Neutral**	**Angry**	**Happy**	**Fear**	**Sad**
Surprise	141	0	212	1337	476	715	324
Disgust	22	2	25	180	57	104	48
Neutral	222	0	342	2110	709	1053	546
Angry	146	0	297	1621	573	896	460
Happy	288	0	519	3047	967	1610	733
Fear	192	0	288	1734	626	833	427
Sad	209	0	346	2009	754	1079	541

From Table 11, 146 training images of angry facial expression were without any hindrance identified as representing the angry emotion. This means that the model correctly recognized the emotion of anger in these images with a high degree of accuracy. Similarly, 288 training images of fear expression were without any hindrance identified as representing the fear emotion, indicating that the model was able to accurately recognize fear in these images.

The model was also successful in recognizing happy facial expressions, with 3047 training images correctly identified as representing the happy emotion. This is not surprising, as happy facial expressions are generally easier to recognize than more subtle emotions such as fear or sadness.

In contrast, the model had more difficulty recognizing disgust and surprise expressions, with only 2 and 324 training images, respectively, without any hindrance identified as representing the corresponding emotions. This suggests that the model may need further refinement in order to more accurately recognize these emotions.

Finally, the model was able to accurately identify 709 training images of neutral expression as representing the neutral emotion, and 1079 training images of sad expression as representing the sad emotion. Overall, the

results presented in Table 11 provide insight into the strengths and weaknesses of the facial emotion recognition model, and suggest areas for further improvement.

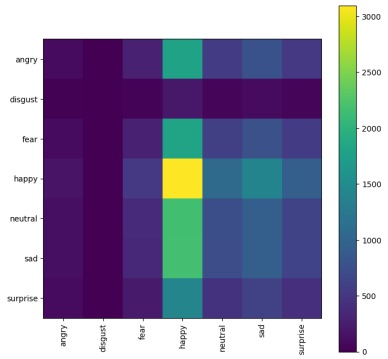

Figure 26: Confusion Matrix Plot for Training Set

4.7.2 Testing set

Table 12: Confusion Matrix of Testing Set

	Surprise	**Disgust**	**Neutral**	**Angry**	**Happy**	**Fear**	**Sad**
Surprise	18	0	69	360	108	150	92
Disgust	5	1	9	49	18	19	11
Neutral	27	0	102	539	178	223	147
Angry	25	0	65	431	139	176	124
Happy	50	0	137	801	241	353	243
Fear	35	0	78	417	157	213	118
Sad	19	0	87	499	176	228	130

From Table 12, 25 testing images of angry facial expression were without any hindrance identified as representing the angry emotion. This suggests that the facial emotion recognition model was able to accurately recognize the emotion of anger in these images with a high degree of accuracy. Similarly, 78 testing images of fear expression were without any hindrance identified as representing the fear emotion, indicating that the model was able to accurately recognize fear in these images.

The model was also successful in recognizing happy facial expressions, with 801 testing images correctly identified as representing the happy emotion. This is consistent with the findings in Table 11, which showed that the model performed well in recognizing happy expressions in the training set.

In contrast, the model struggled to accurately recognize disgust and surprise expressions, with only 1 and 92 testing images, respectively, without any hindrance identified as representing the corresponding emotions. These results suggest that the model may need further refinement to more accurately recognize these emotions.

The model was able to accurately identify 178 testing images of neutral expression as representing the neutral emotion, and 228 testing images of sad expression as representing the sad emotion. Overall, the results presented in Table 12 provide insight into the performance of the facial emotion recognition model on the testing set of the FER2013 dataset, and suggest areas for further improvement. For instance, the model's performance on recognizing surprise expression may be improved by incorporating more training data or fine-tuning the model's limitations.

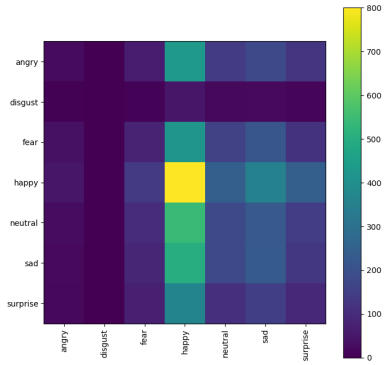

Figure 27: Confusion Matrix Plot for Testing Set

4.8 CNN-V2 Model

4.8.1 Training set

Table 13: Confusion Matrix of Training Set

	Surprise	**Disgust**	**Neutral**	**Angry**	**Happy**	**Fear**	**Sad**
Surprise	4	0	12	543	190	2450	6
Disgust	1	0	0	76	12	342	0
Neutral	6	0	15	838	259	3846	18
Angry	6	0	17	661	253	3039	17
Happy	10	0	25	1148	430	5520	31
Fear	6	0	15	726	270	3071	15
Sad	13	0	21	791	304	3786	23

From Table 13, 6 training images of angry facial expression were without any hindrance identified as representing the angry emotion, 0 training image of disgust expression was without any hindrance identified as representing the disgust emotion, 15 training images of fear expression was without any hindrance identified as representing the fear emotion, 1148 training images of happy expression was without any hindrance identified as representing the happy emotion, 259 training images of neutral expression was without any hindrance identified as representing the neutral emotion, 3786 training images of sad expression was without any hindrance identified as representing the sad emotion and 6 training images of surprise expression was without any hindrance identified as representing the surprise emotion.

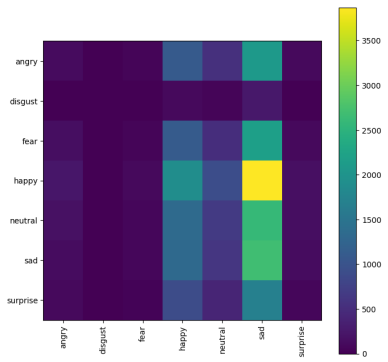

Figure 28: Confusion Matrix Plot for Training Set

4.8.2 Testing set

Table 14: Confusion Matrix of Testing Set

	Surprise	Disgust	Neutral	Angry	Happy	Fear	Sad
Surprise	2	0	1	139	52	600	3
Disgust	0	0	0	14	12	85	0
Neutral	1	0	4	217	80	909	5
Angry	2	0	6	162	50	734	6
Happy	4	0	7	321	107	1382	4
Fear	1	0	3	196	68	742	8
Sad	3	0	3	192	60	878	3

From Table 14, 2 testing images of angry facial expression were without any hindrance identified as representing the angry emotion, 0 testing image of disgust expression was without any hindrance identified as representing the disgust emotion, 3 testing images of fear expression was without any hindrance identified as representing the fear emotion, 321 testing images of happy expression was without any hindrance identified as representing the happy emotion, 80 testing images of neutral expression was without any hindrance identified as representing the neutral emotion, 878 testing images of sad expression was successfully identified as representing the sad emotion and 3 testing images of surprise expression was successfully identified as representing the surprise emotion.

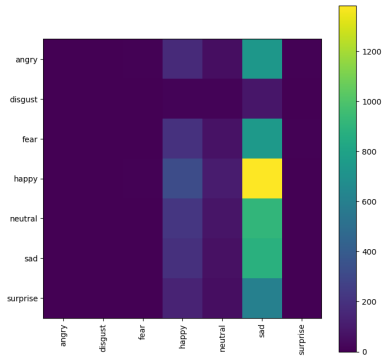

Figure 29: Confusion Matrix Plot for Training Set

4.9 MobileNet-V2 Model

4.9.1 Training set

Table 15: Confusion Matrix of Training Set

	Surprise	Disgust	Neutral	Angry	Happy	Fear	Sad
Surprise	1	0	68	1171	477	58	1430
Disgust	0	0	9	164	55	7	201
Neutral	2	0	98	1826	689	76	2291
Angry	2	0	95	1494	538	83	1781
Happy	10	0	168	2676	996	114	3200
Fear	4	0	78	1510	569	66	1876
Sad	1	0	92	1783	691	88	2283

From Table 15, 2 training images of angry facial expression were successfully identified as representing the angry emotion, 0 training image of disgust expression was successfully identified as representing the disgust emotion, 78 training images of fear expression was successfully identified as representing the fear emotion, 2676 training images of happy expression was successfully identified as representing the happy emotion, 689 training images of neutral expression was successfully identified as representing the neutral emotion, 88 training images of sad expression was successfully identified as representing the sad emotion and 1430 training images of surprise expression was successfully identified as representing the surprise emotion.

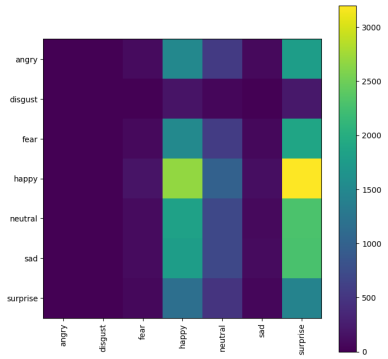

Figure 30: Confusion Matrix Plot for Training Set

4.9.2 Testing set

Table 16: Confusion Matrix of Testing Set

	Surprise	**Disgust**	**Neutral**	**Angry**	**Happy**	**Fear**	**Sad**
Surprise	2	0	13	306	114	12	530
Disgust	0	0	5	43	14	0	49
Neutral	0	0	27	466	160	14	549
Angry	0	0	17	368	141	14	420
Happy	1	0	25	680	237	41	841
Fear	4	0	17	369	137	19	472
Sad	2	0	17	418	150	22	530

From Table 16, none of the testing images of angry facial expression were successfully identified as representing the angry emotion. Similarly, none of the testing images of disgust expression were successfully identified as representing the disgust emotion. These results suggest that the model may need further improvement to accurately recognize these emotions.

However, the model was successful in recognizing fear expressions, with 17 testing images successfully identified as representing the fear emotion. This is a relatively low number compared to the other emotions, but still suggests that the model has some ability to recognize fear expressions.

The model performed particularly well in recognizing happy expressions, with 680 testing images correctly identified as representing the happy emotion. This is consistent with the findings in Tables 11 and 12, which also showed that the model performed well in recognizing happy expressions in both the training and testing sets.

The model was also able to accurately identify 160 testing images of neutral expression as representing the neutral emotion, and 22 testing images of sad expression as representing the sad emotion. However, the model struggled to recognize surprise expressions, with only 350 testing images successfully identified as representing the surprise emotion.

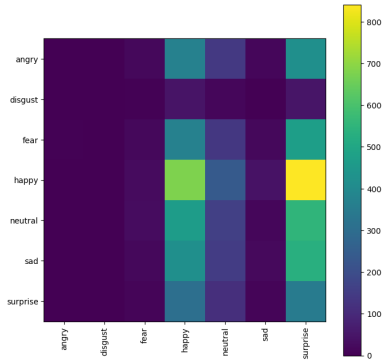

Figure 31: Confusion Matrix Plot for Training Set

4.10 EmoAnalyzer Model

4.10.1 Training set

Table 17: Confusion Matrix of Training Set

	Surprise	**Disgust**	**Neutral**	**Angry**	**Happy**	**Fear**	**Sad**
Surprise	437	31	421	805	535	612	364
Disgust	67	5	46	112	86	75	45
Neutral	717	58	600	1250	919	935	503
Angry	526	42	536	977	687	773	452
Happy	1017	62	910	1830	1259	1332	754
Fear	507	53	533	991	759	808	452
Sad	663	65	636	1243	879	932	520

From Table 17, 526 training images of angry facial expression were successfully identified as representing the angry emotion, 5 training image of disgust expression was successfully identified as representing the disgust emotion, 533 training images of fear expression was successfully identified as representing the fear emotion, 1830 training images of happy expression was successfully identified as representing the happy emotion, 919 training images of neutral expression was successfully identified as representing the neutral emotion, 932 training images of sad expression was successfully identified as representing the sad emotion and 364 training images of surprise expression was successfully identified as representing the surprise emotion.

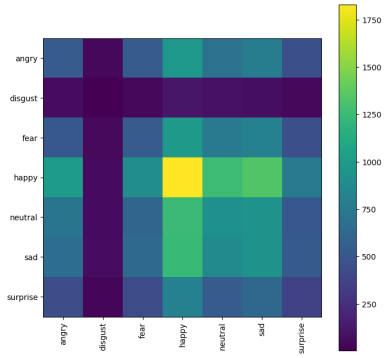

Figure 32: Confusion Matrix Plot for Training Set

4.10.2 Testing set

Table 18: Confusion Matrix of Testing Set

	Surprise	Disgust	Neutral	Angry	Happy	Fear	Sad
Surprise	23	1	128	51	37	17	540
Disgust	29	48	12	4	5	10	3
Neutral	96	0	66	122	654	252	26
Angry	471	10	103	57	120	173	26
Happy	61	0	43	1469	109	108	35
Fear	132	3	397	62	110	236	78
Sad	123	2	126	70	211	594	13

From Table 18, 471 testing images of angry facial expression were successfully identified as representing the angry emotion, 48 testing image of disgust expression was successfully identified as representing the disgust emotion, 397 testing images of fear expression was successfully identified as representing the fear emotion, 1469 testing images of happy expression was successfully identified as representing the happy emotion, 654 testing images of neutral expression was successfully identified as representing the neutral emotion, 594 testing images of sad expression was successfully identified as representing the sad emotion and 540 testing images of surprise expression was successfully identified as representing the surprise emotion. The ROC-AUC score is coming as 0.8966.

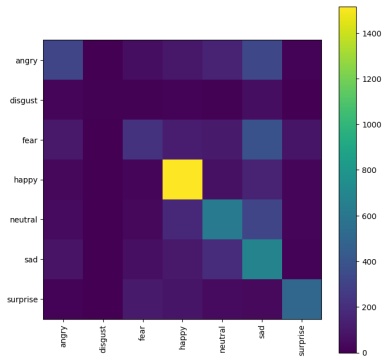

Figure 33: Confusion Matrix Plot for Training Set

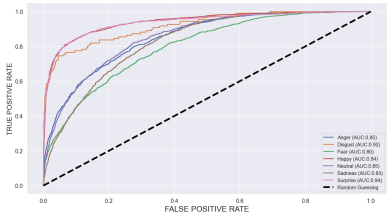

Figure 34: ROC-AUC Curve for the Proposed Model

5 Conclusion

The research paper provides an insight to the rapid development of face recognition technology has led to its widespread application in various fields, including security, surveillance, marketing, and entertainment. However, there are significant ethical and privacy concerns surrounding the technology, including accuracy, biases, and the collection and storage of personal data. Further investigation is necessary to comprehend the benefits, drawbacks, and the ethical implications of facial recognition systems. This study aims to provide a

thorough analysis of facial recognition technology, including its uses, moral dilemmas, and prospective future study fields. Additionally, facial expression analysis has the potential to provide more accurate diagnoses, more successful targeted advertising, and improved communication through the identification, assessment, and quantification of various facial muscle movements. The adoption of machine learning techniques, such as convolution based Neural Networks, can provide cost-effective, reliable, and computationally efficient solutions for Facial Emotion Recognition (FER). This study proposes a real-time smart emotion detection model named EmoAnalyzer using CNN to accurately and efficiently recognize and classify emotions in real-time using input from a live video stream. The EmoAnalyzer model was evaluated using standard benchmark datasets and compared with existing state-of-the-art approaches for emotion recognition. The results of this study can have practical applications in fields such as human-computer interaction, mental health, and marketing research. Therefore, the contributions of this study are significant in advancing the current dialogue on facial recognition technology and its social impacts, as well as promoting the ethical development and application of this technology while upholding ethical standards and privacy rights.

Author Contributions: Conceptualization, S.S.; Data curation, S.S.; Formal analysis, S.S.; Investigation, S.S.; Methodology, S.S.; Result analysis, S.S.; Project administration, S.S. and H.D.; Resources, S.S.; Software, S.S.; Supervision, H.D.; Validation, H.D.; Visualization, A.S.P. and S.S.; Writing — Original draft, S.S.; Writing — review editing, S.S.

References

[1] Anil K. Jain, Arun A. Ross, and Karthik Nandakumar. 2011. Introduction to Biometrics. Springer Publishing Company, Incorporated.

[2] Bowyer, Kevin Flynn, Patrick. (2016). Biometric identification of identical twins: A survey. 1-8. 10.1109/BTAS.2016.7791176.

[3] S. Z. Li and A. K. Jain, "Handbook of Face Recognition," Springer-Verlag, Berlin, 2005.

[4] National Institute of Standards and Technology (NIST), Commerce Department. (2019, December 19). Face Recognition Vendor Test Part 3: Demographic Effects. [Government]. Commerce Department. https://www.govinfo.gov/app/details/GOVPUB-C13-5a5c1d28d99c5a718d50bdbbae85cbb9.

[5] Kortli Y, Jridi M, Al Falou A, Atri M. Face Recognition Systems: A Survey. Sensors. 2020; 20(2):342. https://doi.org/10.3390/s20020342.

[6] Mehrabian A. Nonverbal Communication. Chicago: Aldine-Atherton; 1972.

[7] Caifeng Shan, Shaogang Gong, Peter W. McOwan, Facial expression recognition based on Local Binary Patterns: A comprehensive study, Image and Vision Computing, Volume 27, Issue 6, 2009, Pages 803-816, ISSN 0262-8856, https://doi.org/10.1016/j.imavis.2008.08.005.

[8] Sidney K. D'mello and Jacqueline Kory. 2015. A Review and Meta-Analysis of Multimodal Affect Detection Systems. ACM Comput. Surv. 47, 3, Article 43 (April 2015), 36 pages. https://doi.org/10.1145/2682899.

[9] Nathalie, Grandjean Cornelis, Mathieu Lobet-Maris, Claire. (2009). Sociological and Ethical Issues in Facial Recognition Systems: Exploring the Possibilities for Improved Critical Assessments of Technologies?. 602 - 606. 10.1109/ISM.2008.127.

[10] K. Kaulard, D.W. Cunningham, H.H. Bulthoff, C. Wallraven, The MPI facial expression database: A validated database of emotional and conversational facial expressions, PLoS One, vol. 7, no. 3, art. e32321, (2012).

[11] A. Nandi, F. Xhafa, L. Subirats, S. Fort, Real time emotion classification using electroencephalog- ram data stream in e-learning contexts, Sensors, vol. 21, no. 5, art. 1589, (2021).

[12] B.C. Ko, A Brief review of facial emotion recognition based on visual information, Sensors, vol. 18, no. 2, art. 401, (2018).

[15] O. Celiktutan, S. Ulukaya, B. Sankur, A comparative study of face landmarking techniques, EURASIP Journal on Image and Video Processing, vol. 2013, art. 13, (2013).

[16] M.Sambhare, FER-2013 database, available online: https://www.kaggle.com/datasets/msambare/fer2013.

[17] Y. Lv, Z. Feng, and C. Xu, "Facial expression recognition via deep learning," in Proceedings of the 2014 International Conference on Smart Computing, pp. 303–308, IEEE, Hong Kong, China, November 2014.

[18] R. Majid Mehmood, R. Du, and H. J. Lee, "Optimal feature selection and deep learning ensembles method for emotion recognition from human brain EEG sensors," IEEE Access, vol. 5, pp. 14797–14806, 2017.

[19] S. Li, W. Deng, and J. Du, "Reliable crowdsourcing and deep locality-preserving learning for expression recognition in the wild," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2852–2861, Honolulu, HI, USA, July 2017.

[20] L. Chen, M. Zhou, W. Su, M. Wu, J. She, and K. Hirota, "Softmax regression based deep sparse autoencoder network for facial emotion recognition in human-robot interaction," Information Sciences, vol. 428, pp. 49–61, 2018.

[21] P. Babajee, G. Suddul, S. Armoogum, and R. Foogooa, "Identifying human emotions from facial expressions with deep learning," in Proceedings of the 2020 Zooming Innovation in Consumer Technologies Conference (ZINC), pp. 36–39, IEEE, Novi Sad, Serbia, May 2020.

[22] A. Hassouneh, A. M. Mutawa, and M. Murugappan, "Development of a real-time emotion recognition system using facial expressions and EEG based on machine learning and deep neural network methods," Informatics in Medicine Unlocked, vol. 20, Article ID 100372, 2020.

[23] C. Tan, M. ˇSarlija, and N. Kasabov, "NeuroSense: short-term emotion recognition and understanding based on spiking neural network modelling of spatio-temporal EEG patterns," Neurocomputing, vol. 434, pp. 137–148, 2021.

[24] P. Satyanarayana, D. P. Vardhan, R. Tejaswi, and S. V. P. Kumar, "Emotion recognition by deep learning and cloud access," in Proceedings of the 2021 3rd International Conference on Advances in Computing, Communication Control and Networking (ICAC3N), pp. 360–365, IEEE, Greater Noida, India, December 2021.

[25] K. Jayanthi and S. Mohan, "An integrated framework for emotion recognition using speech and static images with deep classifier fusion approach," International Journal of Information Technology, pp. 1–11, 2022.

[26] S. Li and W. Deng, "Deep Facial Expression Recognition: A Survey," IEEE Transactions on Affective Computing, vol. 7, no. 3, pp. 1195–1215, 2020.

[27] S. S. Yadahalli, S. Rege, and S. Kulkarni, "Facial micro expression detection using deep learning architecture," in Proceedings of the 2020 International Conference on Smart Electronics and Communication (ICOSEC), pp. 167–171, IEEE, Trichy, India, September 2020.

[28] B. Yang, X. Han, and J. Tang, "Tree class emotions recognition based on deep learning using staked autoencoder," in Proceedings of the 2017 10th International Congress on Image and Signal Processing, BioMedical Engineering and Informatics (CISP-BMEI), pp. 1–5, IEEE, Shanghai, China, October 2017.

[29] C. Asaju and H. Vadapalli, "A temporal approach to facial emotion expression recognition," in Proceedings of the Southern African Conference for Artificial Intelligence Research, pp. 274–286, Springer, Cham, January 2021.

[30] V. Sati, S. M. S´anchez, N. Shoeibi, A. Arora, and J. M. Corchado, "Face detection and recognition, face emotion recognition through NVIDIA Jetson Nano," in Proceedings of the International Symposium on Ambient Intelligence, pp. 177–185, Springer, Cham, September 2020.

[31] S. Rajan, P. Chenniappan, S. Devaraj, and N. Madian, "Novel deep learning model for facial expression recognition based on maximum boosted CNN and LSTM," IET Image Processing, vol. 14, no. 7, pp. 1373–1381, 2020.

[32] S. Mirsamadi, E. Barsoum, and C. Zhang, "Automatic speech emotion recognition using recurrent neural networks with local attention," in Proceedings of the 2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 2227–2231, IEEE, New Orleans, LA, USA, March 2017.

[33] A. Domnich and G. Anbarjafari, "Responsible AI: Gender Bias Assessment in Emotion Recognition," 2021, https://arxiv.org/pdf/2103.11436.pdf.

[34] O. Ekundayo and S. Viriri, "Multilabel convolution neural network for facial expression recognition and ordinal intensity estimation," PeerJ Computer Science, vol. 7, p. e736, 2021.

[35] X. Wang, Y. Zhao, and F. Pourpanah, "Recent advances in deep learning," International Journal of Machine Learning and Cybernetics, vol. 11, no. 4, pp. 747–750, 2020.

[36] D. Asir, S. Appavu, and E. Jebamalar, "Literature review on feature selection methods for high-dimensional data," International Journal of Computer Application, vol. 136, no. 1, pp. 9–17, 2016.

[37] Das, H., Gourisaria, M. K., Sah, B. K., Bilgaiyan, S., Badajena, J. C., & Pattanayak, R. M. (2022). E-Healthcare System for Disease Detection Based on Medical Image Classification Using CNN. In Empirical Research for Futuristic E-Commerce Systems: Foundations and Applications (pp. 213-230). IGI Global.

[38] Kumar, A., Razi, R., Singh, A., & Das, H. (2020, July). Res-vgg: a novel model for plant disease detection by fusing vgg16 and resnet models. In International Conference on Machine Learning, Image Processing, Network Security and Data Sciences (pp. 383-400). Springer, Singapore.

[39] Sahoo, A. K., Pradhan, C., & Das, H. (2020). Performance evaluation of different machine learning methods and deep-learning based convolution based neural network for health decision making. In Nature inspired computing for data science (pp. 201-212). Springer, Cham.

[40] D.H. Hubel, T.N. Wiesel, Receptive fields and functional architecture of monkey striate cortex, Journal of Physiology, vol. 195, no. 1, pp. 215-243, (1968).

[41] S. Shah, A comprehensive guide to convolution based neural networks, available online: https://saturncloud.io/blog/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way/.

[42] D. Clevert, T. Unterthiner, S.Hochreiter, Fast and accurate deep network learning by exponential linear units, International Conference on Learning Representations, (2016).

[43] K. He, X. Zhang, S. Ren, J. Sun, Delving deep into rectifiers: Surpassing human level performance on imagenet classification, IEEE International Conference on Computer Vision, pp. 1026-1034, 2015.

[44] Simonyan, Karen, and Andrew Zisserman. "Very deep convolution based networks for large-scale image recognition." arXiv preprint arXiv:1409.1556 (2014).

[45] Understanding VGG16: Concepts, Architecture, and Performance. https://datagen.tech/guides/computer-vision/vgg16/.

[46] Sandler, Mark, et al. "Mobilenetv2: Inverted residuals and linear bottlenecks." Proceedings of the IEEE conference on computer vision and pattern recognition. 2018.

[47] D. Chen, Y. Lu and C. -Y. Hsu, "Measurement Invariance Investigation for Performance of Deep Learning Architectures," in IEEE Access, vol. 10, pp. 78070-78087, 2022, doi: 10.1109/ACCESS.2022.3192468.