Scope
Volume 14 Number 01 March 2024

Facial Image Emotion Recognition and Detection Using Conv. Relu Features Extraction and ANN Classification Deep Learning Model

D. O. Njoku[1], J. N. Odii[2], E. C. Nwokorie[3], C.G. Onukwugha[4], J. E. Jibiri[5] & Mark P. Mcwilliams[6]

[1, 2, 3,4, 6]Department of Computer Science, Federal University of Technology, Owerri
[5]Department of Information Technology, Federal University of Technology, Owerri

Corresponding Author: **D.O. Njoku**

Abstract

Facial image emotion recognition detection is an increasingly important area of research in computer vision and artificial intelligence. The development of deep learning models has provided significant improvement accuracy and reliability of the system. Model was trained with a total 2,489,095 parameters, with 35,887 images with 100 epoches. The deep learning model applied Conv.Relu features extraction and Artificial Neural Network (ANN) techniques Convolutional Neural Network (CN) classification, the model generated an accuracy of 76.83% on the training datatset and 65.38% on the validation dataset, the model was performs well but it still overfifed as at training accuracy was at 65%.Seven emotions such happy, sad, neutral, disgust, surprise and fear where evaluated using precision, recall and f-score of 0.26 while the disgust emotion has the worst precision, recall and f-score of 0.01.

Keywords: Artificial Neural Network, Emotion, Facial Image, Cov.Relu, Convolution Neural Network, Deep learning, , recognition, detection, parameters, training

Introduction

Facial emotion recognition and detection have been a topic of interest in the field of artificial intelligence and computer vision for many years. Accurately identifying human emotions through facial expressions can have numerous applications, including improving mental health care, enhancing human-robot interaction, and improving security measures (Minaee, Minaeri and Abdolrashidi, 2021). To execute a given job such as object recognition or speech recognition, deep learning includes training artificial neural networks with several layers. Deep learning

models are trained on big datasets of photos including various facial expressions and emotions for facial emotion recognition and detection. To forecast the emotion in a new image, the models learn to recognize patterns in the images that correspond to various emotions. One of the earliest attempts to use deep learning for facial emotion recognition was the work of (Shan, Gong, and McOwan, (2022). They classified facial expressions using a convolutional neural network (CNN) into six fundamental emotions: anger, contempt, fear, happiness, sorrow, and surprise. On the CK+ dataset, which contains 327 labeled images of facial expressions, the model achieved an accuracy of 91.4%.

Since then, a lot of research has been done to raise the reliability and accuracy of deep learning models for identifying and detecting facial emotions. The creation of more sophisticated neural network topologies and training methods has been the focus of many of these investigations. In recent years, there has been significant progress in the development of deep-learning models for facial emotion recognition and detection. One of the earliest and most well-known datasets used for this task is the Cohn-Kanade AU-Coded Expression Database (Kanade, Cohn., and Yingli,, 2000), which contains over 5000 images of human faces displaying different emotions. Researchers have used this dataset to train deep neural networks to recognize and classify facial expressions of emotions such as happiness, sadness, anger, surprise, disgust, and fear. The Deep Emotion Recognition (DeepER) model, which reached cutting-edge accuracy on the Cohn-Kanade database, uses a deep convolutional neural network (CNN) to extract information from facial photos. Since then, many additional deep learning models have been put out, such as Deep Boltzmann Machines (DBMs), Deep Belief Networks (DBNs), and Recurrent Convolutional Neural Networks (RCNNs)(Deng, Guo, and Liu , 2021)

More recent trend in deep learning is the availability of frameworks/libraries that have aided in the advancement of face emotion identification. These frameworks/libraries include a variety of tools, functions, and pre-trained models that make deep learning model building and deployment easier. Popular libraries such as TensorFlow, Keras, PyTorch, OpenCV, Caffe, etc each has unique benefits and characteristics that make them suited for certain elements of face emotion identification applications. Comparing and assessing different deep learning libraries for face emotion identification is critical for researchers and practitioners. Performance, simplicity of use, flexibility, documentation, and integration capabilities vary among these libraries, determining their applicability to various project requirements. Deep learning models for recognizing and detecting facial emotions have recently been implemented in practical applications including video surveillance and mental health treatment. Several issues still need to be resolved, such as strengthening the models' robustness to manage fluctuations in facial expressions and creating more models to better understand how they generate their predictions.

Despite substantial advances in face emotion identification using convolutional neural networks (CNNs), face recognition has several difficulties when trying to reliably identify emotions interpretable from facial expressions, particularly in complex and dynamic situations (Mehendale, 2022). This issue needs to be addressed to increase these models' accuracy and resilience. An existing CNN models for face emotion identification need big, diversified, and well-labeled datasets, which can be challenging to acquire and annotate. Furthermore, these models frequently fail to generalize across varied populations, cultural backgrounds, and lighting situations, resulting in possible biases and poor performance in real-world applications Arora,. et. al (2022). In light of the aforementioned difficulties, the issue statement is to create and assess deep learning models that can precisely identify and detect facial emotions in a practical context.

This paper tends to use Conv.Relu features extraction and ANN classification deep learning model for facial recognition and detection. The objectives of this paper is to detect, identify and verify human facial emotion via webcam and images; to utilize a large dataset with a varied range of facial expressions, demographic background, and cultural contexts to ensure the model performance across different population; the research also presents a deep learning algorithm approach of CNN to recognized different sorts of emotions based on facial. The relevant of this paper in the ability to recognize and interpret facial expressions as an essential component of human communication and plays a significant role in social interactions, affective computing, and human-robot interactions. By developing accurate and effective facial emotion recognition systems, researchers can improve the ability of machines to understand and respond to human emotions, which can have practical applications in various fields such as education, healthcare, and entertainment.

Facial Emotion Recognition (FER) and Detection is a technology that analyzes emotions from many sources, such as images and movies. It is a member of the “affective computing” family of technologies, a multidisciplinary field of study on computers' capacity to recognize and understand human emotions and affective states. It frequently builds on Artificial Intelligence technology (Kumar et al., 2021). Facial expressions are types of nonverbal communication that reveal human emotions. The analysis of facial landmark placements (e.g., end of the nose, brows) is used to determine emotions. Recently, the widespread use of cameras, as well as technical improvements in biometrics analysis, machine learning, and pattern recognition, have aided in the advancement of FER technology. FER analysis comprises three steps:

a) Face Detection: it is the pre-processing step for recognizing facial expressions. It involves converting an image to a normalized pure facial picture for features extraction involves identying feature point rotating to line up, and finding and cropping the face region using a rectangle based on the face model (Pandey, Gupta and Shyam, 2022). Methods for face detection include Knowledge-based, feature-invariant, template-matching, and appearance based.

b) Feature extraction: This transforms pixel data into a higher-level representation of the face's or its components' shape, motion, color, texture, and spatial configuration (Pandey et. al, 2022). This can be done using various techniques: Gabor filters, discrete cosine transform (DCT), Principle component analysis (PCA), Independent component analysis (ICA), and Linear discriminate analysis (LDS)
c) Expression classification: A classifier, which is frequently composed of pattern distribution models coupled with a decision mechanism, performs expression classification. Action units and prototypic facial expressions are the two primary types of classes utilized in facial expression recognition that Ekman identified To extract expressions, multiple categorization approaches are employed, they are Hidden Markov model (HMM), Neural network (NN), Support vector machine (SVM), Ada boost, and sparse representation (SRC).

Emotion detection is based on the analysis of facial landmark positions (e.g., end of nose, eyebrows). Depending on the algorithm, facial expressions can be classified as basic emotions (e.g., anger, disgust, fear, joy, sadness, and surprise) or compound emotions (e.g., happily sad, happily surprised, happily disgusted, sadly fearful, sadly angry, sadly surprised). In other cases, facial expressions could be linked to physiological or mental state of mind (e.g., tiredness or boredom).

Deep learning techniques are a subset of machine learning approaches that rely on deep architectures and outperform other machine learning approaches in terms of accuracy and efficiency. Deep architectures are comprised of numerous levels of non-linear operational components, such as neural nets with many hidden layers. Deep learning approaches, which employ machine learning models with several hidden layers that are trained on vast amounts of data, might discover more valuable features and hence increase classification and prediction accuracy as stated by Lierler et al., (2012) Some widely used methods among others are the Convolutional Neural Networks, the Deep Belief Networks, and the Deep Boltzmann Machines.

Convolutional Neural Networks (CNNs)

Convolutional neural networks (CNNs) are a category of neural networks specialized in areas such as image recognition and classification. A convolutional neural network, for most of it, consists of three-layer types: the convolutional layers, the max-pooling layers, and the fully-connected layer. Convolutional layers are at the core of CNNs, allowing the network to learn and extract meaningful features from incoming data automatically. These layers are made up of filters (also known as kernels) that are convolved with the input to create feature maps. The filters capture many local patterns and spatial interactions, such as edges, textures, and forms. Convolutional layers are frequently followed by activation functions that inject

nonlinearity into the network, such as ReLU (Rectified Linear Unit) (Phung et. al, 2018). Pooling layers are widely used to minimize feature map spatial dimensions while keeping the most critical information. Max pooling and average pooling are two common pooling approaches. The output is generally flattened and sent through one or more fully connected layers after multiple convolutional and pooling layers. Depending on the application, these layers function as a typical neural network, doing classification or regression tasks

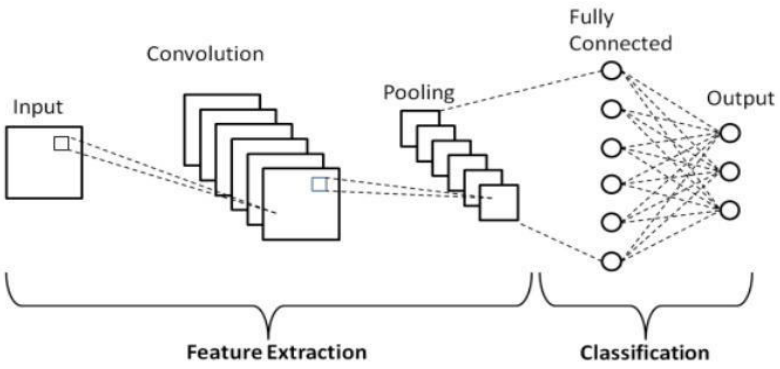

Figure1. A simple architecture of a CNN (Phung et. al. 2018)

Convolution Operation

Convolution is a mathematical process that describes a rule for combining two functions to create a third function. This third function is an integral that expresses the degree of overlap between two functions as they are shifted over one other. In other words, an input data set and a convolution kernel are subjected to a mathematical operation to produce a changed feature map Coskun et al. (2017) Convolution is described formally as follows:

$$h(t) = \int_{-\infty}^{\infty} f(\tau)\, g(t-\tau)\, d\tau \qquad (1)$$

The feature map is calculated by sliding the kernel over the entire input matrix as shown in Fig. 2.10

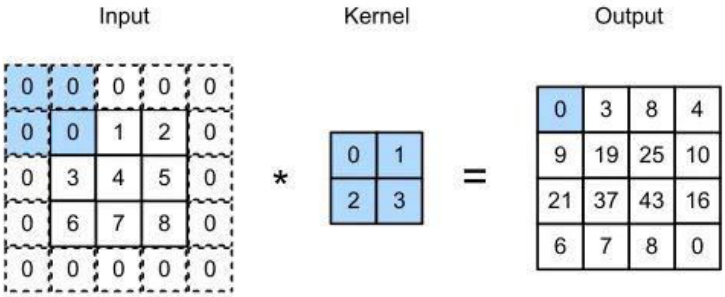

Figure 2. Filter: Coskun et al. (2017)

After generating a feature map, an activation function is performed to replace all negative pixel values to zero and therefore solves the cancellation problem as well as results in a much more sparse activation volume at its output. The sparsity is useful for multiple reasons but mainly provides robustness to small changes in input such as noise (Xavier et al. 2011). The pooling layer then performs Max-pooling, Min-pooling

or Average-pooling to reduce the spatial size or dimensionality of the feature map as in Figure 3

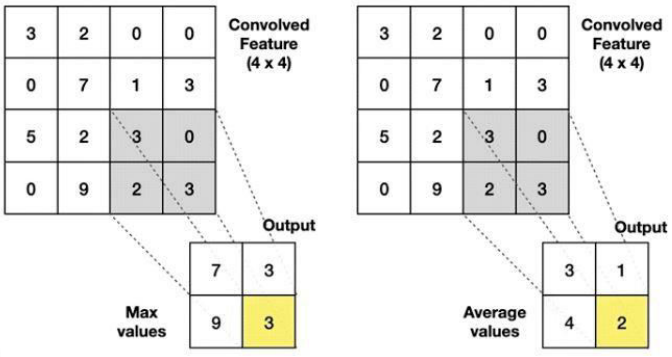

Figure 3. Pooling (Qayyum, 2022)

After Pooling, it is then Flattened to convert all the resultant 2-Dimensional arrays from pooled feature maps into a single long continuous linear vector. The flattened matrix is fed as input to the fully connected layer to classify the image.

CNN Architecture

Model architecture is a critical factor in improving the performance of different applications. Various modifications have been achieved in CNN architecture from 1989 until today. Such modifications include structural reformulation, regularization, parameter optimizations, etc. Studying these architectures features (such as input size, depth, and robustness) is the key to help researchers to choose the suitable architecture for their target task. Some of the architectures are as follows

a) AlexNet: AlexNet is well-known in deep CNN architecture (Krizhevsky et. al, 2017) as it achieved innovative results in the fields of image recognition and classification. Krizhevesky et al proposed AlexNet initially and then enhanced the CNN learning capabilities by increasing its depth and adopting different parameter optimization procedures. To overcome hardware limitations, AlexNet was trained using two GPUs (NVIDIA GTX 580) in parallel. Furthermore, in order to improve the CNN's applicability to multiple picture categories, the number of feature extraction steps was raised from five in LeNet to seven in AlexNet.

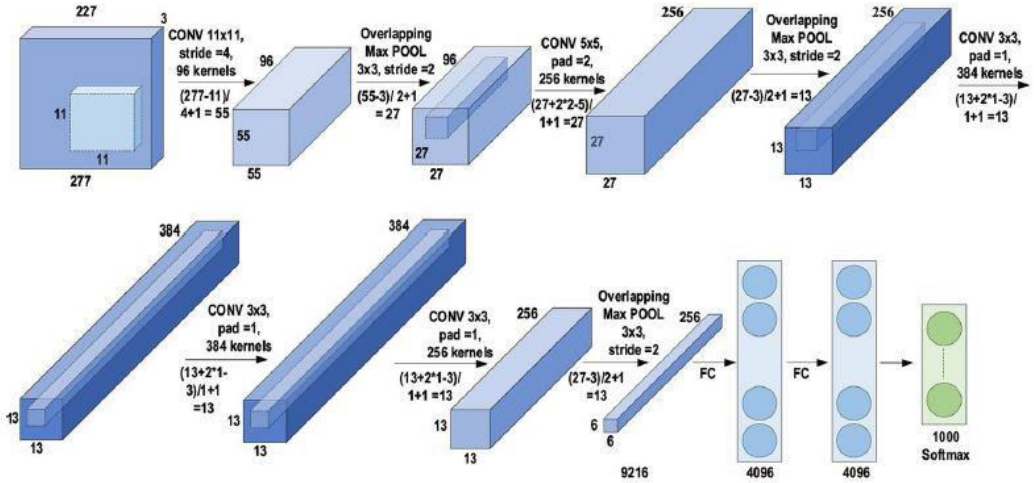

Figure 4. Architecture of AlexNet (Alzubaidi et. al, 2022)

b) VGGNet: VGGNet from the VGG group at Oxford finished second in the ILSVRC 2014 Simonyan and Zisserman. It improves on AlexNet and has 19 layers in all. Its key contribution was to demonstrate that the depth of the network, or the number of layers, is an important factor in achieving high performance. Although VGGNet achieves fantastic accuracy on ImageNet datasets, its implementation on even the most modest sized Graphics Processing Units (GPUs) is a difficulty due to massive computing needs in terms of memory and time. It becomes inefficient because of the wide breadth of the convolutional layers.

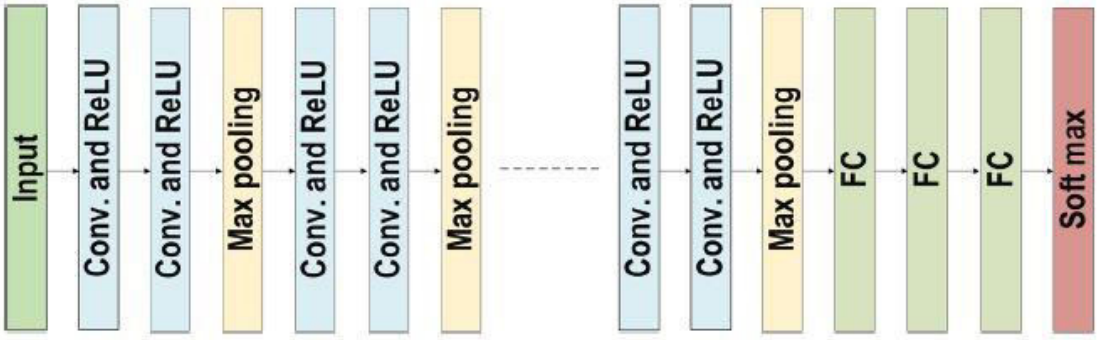

Figure 5. Architecture of VGG (Alzubaidi et. al, 2022)

d) ResNet: (Residual Network) developed by Kaiming He et al. was the winner of ILSVRC 2015, Kaiming, Zhang, Shaoqing , and Sun. ResNet is a 152 layer network, which was ten times deeper than what was usually seen during the time when it was invented It features special skip connections and a heavy use of batch normalization. It uses a global average pooling followed by the classification layer. It achieves better accuracy than VGGNet and GoogLeNet while being computationally more efficient than VGGNet. ResNet-152 achieves 95.51% top-5 accuracies.

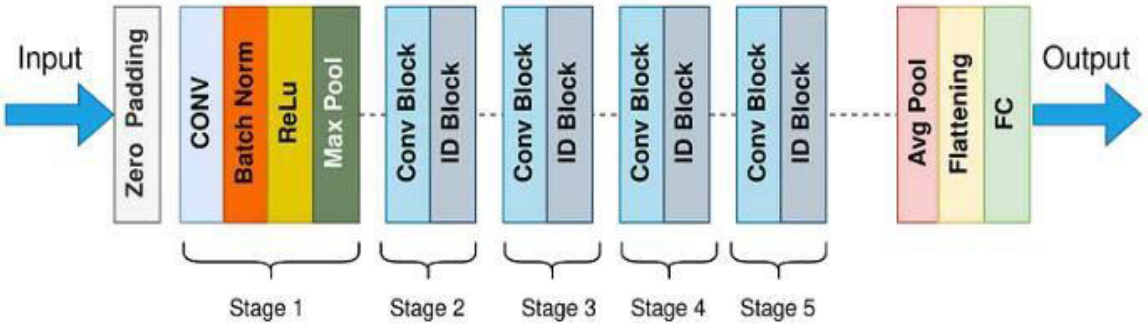

Figure 6. Architecture of ResNet(Kaiming et al. 2015)

Non-Linear Activation Function

Non-linear activation functions are employed in current neural networks. They let the model create complex mappings between the network's inputs and outputs, which is essential for learning and modeling complex data such as pictures, video, audio, and non-linear or high-dimensional data sets (Bhatt et .al, 2021. Some of the non-linear AF is as follows:

a) ReLU: In the CNN context, this is the most widely utilized function. It converts the input's whole values to positive numbers. The key advantage of ReLU over the others is its lower computational load. It's mathematically expressed by; $f(x)_{ReLU} = max(0,x)$ (2)

b) Sigmoid: This activation function takes real numbers as input and outputs only values between zero and one. The sigmoid function curve is S-shaped and mathematically expressed by;

$$f(x)_{sigm} = \frac{1}{1+e^{-x}} \quad (3)$$

c) Tanh: It is similar to the sigmoid function, as its input is real numbers, but the output is restricted to between –1 and 1. Its mathematical representation as:

$$f(x)_{tanh} = \frac{e^{x}-e^{-x}}{e^{x}+e^{-x}} \quad (4)$$

d) LeakyReLU: Rather of ReLU downscaling negative inputs, this activation function ensures they are never ignored. It is used to solve the Dying ReLU issue. It is mathematically represented as

$$f(x)_{LeakyReLU} = \begin{cases} x, & if\ x > 0 \\ mx, & x \leq 0 \end{cases} \quad (5)$$

e) SoftMax: It is commonly used in the output layer of a neural network for multi-class classification problems. It takes a vector of inputs and normalizes them to represent a probability distribution over the classes. It is mathematically represented as $\sigma(\vec{z})i = \frac{e^{z_i}}{\sum_{j=1}^{k} e^{z_i}}$ (6)

Facial Dataset

One of the success factors of deep learning is using samples to train neural networks. To accomplish this task, researchers now have at their disposal multiple FER

databases, each unique in terms of number and size of images and videos, lighting variations, populations, and facial poses. Some of them are in Table 1

Table 1. Facial dataset

Dataset	Descriptions	Emotions
CK+	593 videos for posed and non-posed expressions	Six basic emotions, contempt and neutral
Fer2013	35,887 grayscale images collect from google image search	Six basic emotions and neutral
AffectNET (Mollahosseini, Hasani, Mahoor, 2019)	More than 440.000 images collected from the internet	Six basic emotions and neutral
RAFD-DB	30000 images from real world	Six basic emotions and neutral
MultiPie	More than 750,000 images captured by 15 view and 19 illumination conditions	Anger, Disgust, Neutral, Happy, Squint, Scream, Surprise

Hyperparameters

During learning process, some parameters are considered in order to achieve the best possible performance

a) Batch size: Since the entire dataset cannot be propagated into the neural network at once for memory limitations, it is divided into batches, which makes the overall training procedure require less memory and become faster.

b) Epoch: The number of epochs indicates how many times all of the dataset has been sent forward and backward through the neural network, that is, one epoch is when every image has been seen once during training.

c) Iteration: the number of batches needed to completed one epoch and it is calculated as:

$$\neq iterations = \frac{\neq epochs * \neq training\ images}{batch\ size} \quad (7)$$

d) Learning rate: The learning rate parameter controls the step size for which the weights of a model are updated regarding the loss gradient. The lower its value is, the

slower the convergence is but it is ensured that it is not missed any local minimum. Usually number, like 0.001 that are multiplied to scale the gradient and ensure that any changes made to the weigh are quite small.

Research Method

The CNN model will be designed and implemented following the principles of the Structured Systems Analysis and Design Method (SSADM). The project will involve various stages, including feasibility study, requirements analysis, data collection and analysis, system design, coding and implementation, testing, documentation, deployment, and maintenance. The CNN model will be trained and evaluated on a dataset of images to accurately classify facial emotion and achieve high performance. The main concept of SSADM is as follow:

a) Feasibility Study: In this stage, we will analyze the feasibility of building the CNN-based image classification model. We will assess the technical capabilities of the deep learning frameworks, the availability of suitable hardware for training, and the economic viability of the project based on the expected benefits. Requirements Analysis: Gathering requirements is a crucial step to define the scope and functionality of the CNN model. Functional and non-functional requirements will be documented to guide the development process.
b) Data Collection and Analysis: A diverse and representative dataset is essential for training a robust image classification model. We will identify and collect an appropriate dataset for our problem domain. Data analysis will be performed to understand the distribution of classes, detect any class imbalances or biases, and preprocess the data accordingly.
c) System Design: Based on the requirements and data analysis, we will design the architecture and specifications of the CNN model. We will select an appropriate CNN architecture and define hyperparameters such as learning rate, batch size, and activation functions. The model components, including layers, filters, and pooling, will be specified.
d) Coding and Implementation: Using a deep learning framework such as TensorFlow or PyTorch, we will implement the designed CNN model. The code will define the chosen architecture and include data loading and preprocessing pipelines to prepare the dataset for training.
e) Testing: The testing stage will involve evaluating the performance of the trained CNN model. We will use various evaluation metrics like accuracy, precision, recall, and F1-score to assess the model's performance on both the training and testing datasets. Robustness testing will also be conducted to ensure the model's ability to generalize well to unseen data.
f) Documentation: Comprehensive documentation will be prepared for the CNN model. This includes documentation of the model architecture,

hyperparameters, training process, and evaluation results. Code documentation will also be provided for ease of understanding and future modifications.

Proposed System Model

In view of improving the process of face sentiment analysis systems, a classification mechanism is proposed using a custom CNN architecture. Due to the vast amount of data necessary for deep network training, the FER2013 dataset, which is publicly available on Kaggle website is used here. The proposed system comprised of 5 layers, high number of filters for feature map which generates more parameter to be trained.

Analysis of the Proposed System

The architecture implemented on this study is based on Convolutional Neural Network (CNN). It consists of the following:

i. Convolutional Layer: Convolutional layer is the most important component of any CNN architecture. It contains a set of convolutional kernels (also called filters), which gets convolved with the input image (N-dimensional metrics) to generate an output feature map. In proposed system, the layer comprises of three convolutional layers with the filter of 64, 128, 256 and kernel_size of 3x3 which is then passed through an activation function (ReLu) and then pooling and dropout layers are carried out.
ii. Pooling Layer: This layer is used to sub-sample the feature maps (produced after convolution operations), i.e. it takes the larger size feature maps and shrinks them to lower sized feature maps. While shrinking the feature maps it always preserves the most dominant features (or information) in each pool steps. The pooling operation is performed by specifying the pooled region size and the stride of the operation, similar to convolution operation. In the proposed system, each convolutional layer utilizes a max-pooling type with the size of 2x2.
iii. Dropout Layer: This is a regularization technique used to prevent overfitting in the neural network. It randomly sets a fraction of input units to 0 during training, effectively dropping them out. In the proposed system, dropout is set at 0.1 and 0.2, meaning 10% and 20% of the input units will be randomly set to 0 during each training batch. The dropout rate is a hyperparameter that can be tuned based on the specific problem and dataset.
iv. Fully Connected Layer: This is the last part of the CNN architecture (used for classification). The FC layers take input from the final convolutional or pooling layer, which is in the form of a set of metrics (feature maps) and those metrics are flattened to create a vector and this vector is then fed into the FC layer to generate the final output of CNN. In the proposed model, it is implemented using the Dense layer with 512 with a ReLu function and lastly 7 neurons which

represent the 7 classes that the model can classify. The SoftMax activation function is then applied to the output of the layer which converts the output of the layer into a probability distribution over the 7 classes

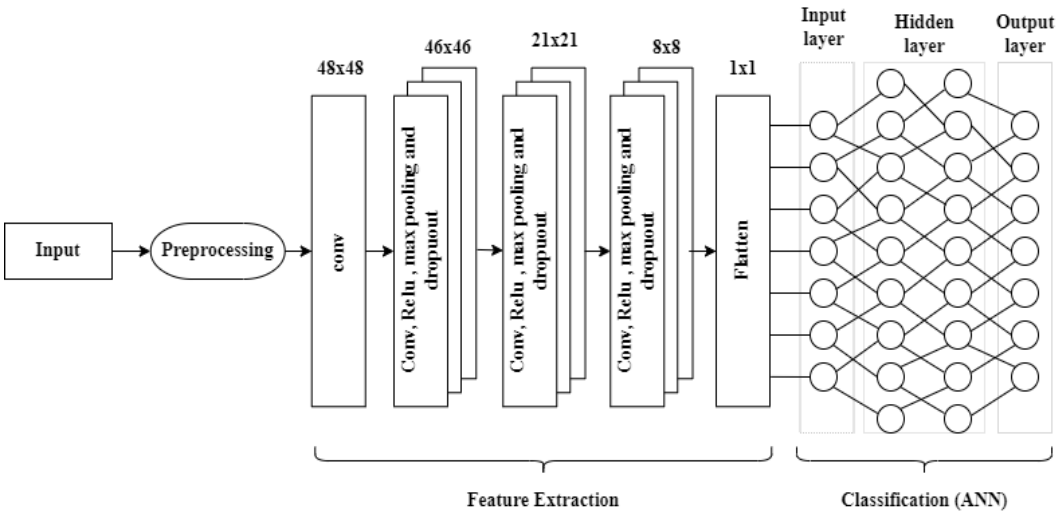

Figure 7. Block diagram of the Proposed System

Design

Firstly, during dataset training, the images are preprocessed and feed to the CNN architecture model to extract feature and classified images based on the Seven emotion classes available, this generates a model file (with the extension .h5) and this file contains already learnt features.

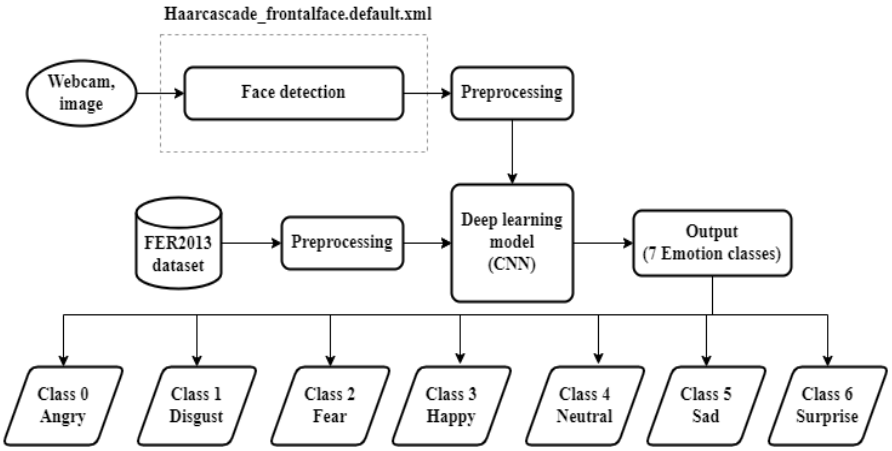

Figure 8. Proposed System design

During prediction (i.e. after model is trained), a webcam captures the face or an image frontal face is been detected and recognized using Haarcascade frontal face dataset, this uses a square frame to obtain the region of interest which is then preprocessed to grayscale image with resize to 48 pixels. the preprocessed image is the sent into the model and with the help of the .h5 file generated during training, the model then extracts featured from the new image, compares it with what it learnt during training and makes accurate prediction of the emotion expressed, the design of the system is as shown in Figure 8

Dataset Utilized

There are two databases utilized in order to assess the facial emotion expression algorithms.

i. Frontal face dataset from Haarcascade: HaarCascade Classifier is used to recognize faces in pictures utilizing characteristics. The frontal face is detected using the haarcascade frontal face default.xml from OpenCV library.

ii. FER2013 Dataset: It is an open-source dataset generated by Pierre-Luc Carrier and Aaron Courville and made publicly available on Kaggle website. It comprises of 35,887 grayscale, 48x48-pixel pictures of faces displaying a range of emotions -7 emotions, all labeled.

Emotion label in the dataset:

0:- 4593 images- Angry
1:- 547 images- Disgust
2:- 5121 images- Fear
3:- 8989 images- Happy
4:- 6077 images- Sad
5:- 4002 images- Surprise
6:- 6198 images- Neutral

Figure 9. Images of FER2013 dataset

The proposed system was implemented using the following: Installation of Python and its dependencies (i.e. numpy, cv2, tensorflow etc.); Installation of IDE (i.e. Jupyter notebook, VS Code); Writing Source code for training model using Jupyter notebook as shown Figure 10 shows th IDE source code and various facial emotion detection.

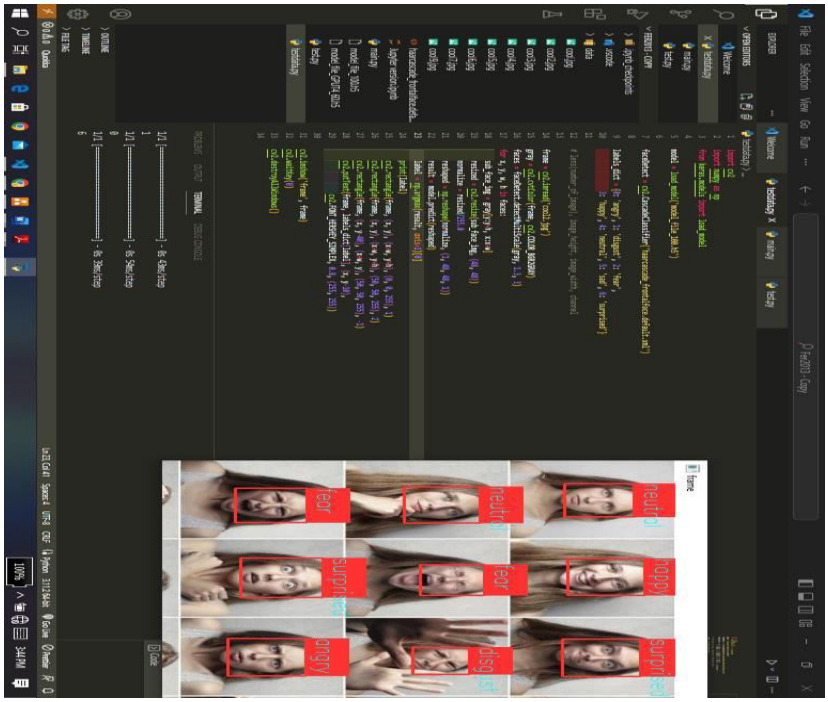

Figure 10. IDE for Jupter Sample Code

Data Preprocessing

Preprocessing of input image appears in two ways, during CNN model training of the dataset and during detection and recognition (after training) ad showed in the Figure

a) Preprocessing during CNN model training: The FER2013 dataset comes with some preprocessing applied to the images, making it ready for training a Convolutional Neural Network (CNN) model for facial expression recognition. The preprocessing steps in FER2013 include the following:

b) Resizing: All images in the FER2013 dataset are preprocessed to a fixed size of 48x48 pixels. Resizing the images to a consistent size ensures that they can be efficiently fed into the CNN model, which typically expects inputs of uniform dimensions.

c) Data splitting: FER2013 dataset splits its data between training dataset and validation dataset. This ensures that the deep learning model is trained on one subset, validated on another, and tested on a separate subset, thereby enabling the evaluation of the model's performance in real-world scenarios.

d) Grayscale Conversion: This data set came with grayscale image (one channel) rather than RGB image (three channels). Grayscale images are faster to process.

e) Label Encoding: The emotion labels in the original dataset are represented as integer values from 0 to 6, corresponding to the seven emotion categories. To prepare the labels for training in a multi-class classification setting, they are often converted to one-hot encoded vectors.

f) Data Augmentation: Applying random transformations (e.g., rotations, flips, zooms) to increase the training dataset's diversity and improve the model's ability to generalize.

g) Preprocessing during facial detection and recognition: In facial detection and recognition tasks, preprocessing is applied to the input images to prepare them for accurate and efficient processing by the detection and recognition algorithms. Preprocessing steps may vary depending on the specific algorithms used, but common steps include:

h) Grayscale Conversion: Converting RGB images to grayscale, which simplifies the processing while retaining facial features' important information.

i) Face Detection: Applying face detection algorithms (e.g., Haar Cascade) to locate and extract the facial regions of interest (ROI) from the input images.

j) Face Alignment: Aligning the detected face regions to a standard pose, which improves recognition accuracy by reducing variations due to head rotations and angles.

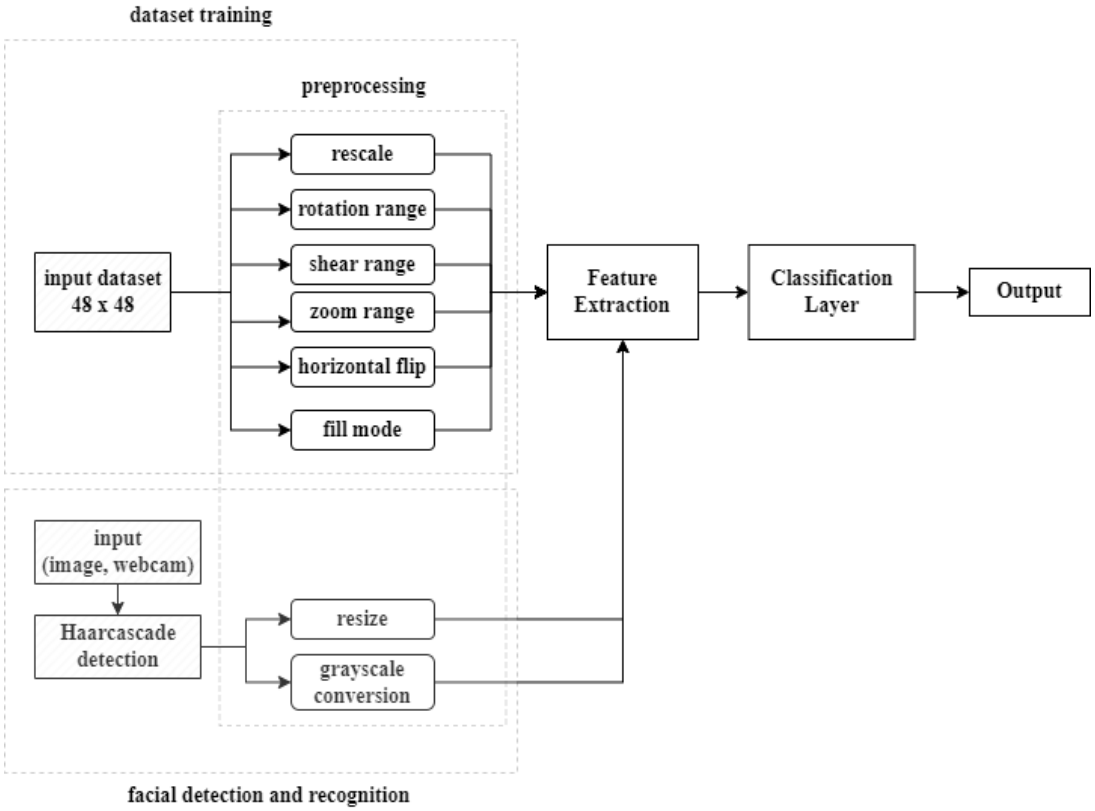

Figure 11. Block diagram of preprocessing

Results and Discussion

i. In FER2013 dataset contains two folders i.e. Training and testing dataset which contains seven class of emotion.

- Training dataset: This contains a total of 28709 images belonging to the seven classes

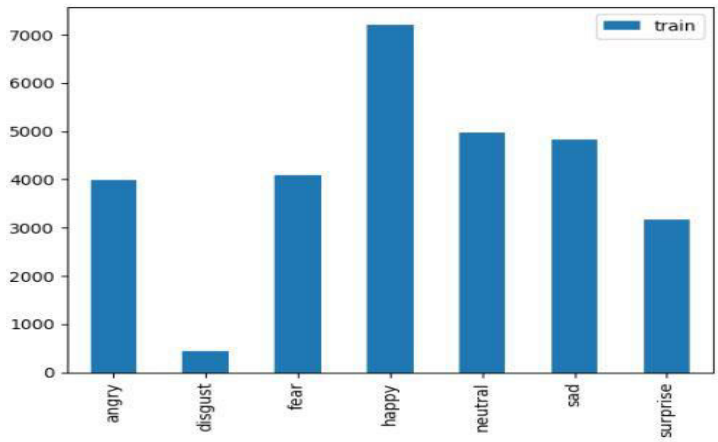

Figure 12. Training dataset

- Test dataset: This contains 7178 images belonging to the seven classes

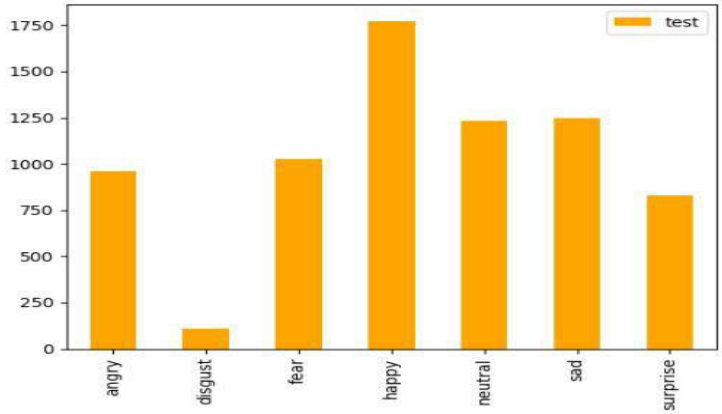

Figure 13. Testing dataset

ii. Model Summary: a concise overview of the deep learning model is as shown in Figure

Table 2. Model: "Sequential"

Layer (type)	**Outpu hape**	**Param #**
Conv2d (Conv2D)	(None, 48, 48, 32)	320
Conv2d_1 (Conv2D)	(None, 46, 46, 64)	18496
Max_pooling2d (MaxPooling2D)	(None, 23, 23, 64)	0
Dropout (Dropout)	(None, 23, 23, 64)	0
Conv2d_2 (Conv2D)	(None, 21, 21, 128)	73856
Max_pooling2d_1	(None, 10, 10,	0

(MaxPooling2D)	128)	
Dropout_1 (Dropout)	(None, 10, 10, 128)	0
Conv2d_3 (Conv2D)	(None, 8, 8, 256)	295168
Max_pooling2d_2 (MaxPooling2D)	(None, 4, 4, 256)	0
Dropout_2 (Dropout)	(None, 4, 4, 256)	0
Flatten (Flatten)	(None, 4096)	0
Dense (Dense)	(None, 512)	2097664
Dropout_3 (Dropout)	(None, 512)	0
Dense_1 (Dense)	(none, 7)	3591

Total params: 2,489,095

Trainable params: 2,489,095

Non-trainable params: 0

iii. Model History: The Model is trained at 100 epoches as shown in Table 2

Table 3. Model History

	Loss	**Accuracy**	**Val_loss**	**val_Accuracy**
0	1.796232	0.260906	1.709540	0.317522
1	1.702466	0.317118	1.570815	0.400946
2	1.614819	0.367542	1.498884	0.427595
3	1.552526	0.399100	1.443179	0.447405
4	1.487143	0.425742	1.386566	0.475307
...	.....	.....	.....	.....
95	0.770499	0.715138	0.998365	0.654157
96	0.764864	0.716184	0.990512	0.657227
97	0.760478	0.718346	0.990424	0.655971
98	0.755445	0.719601	1.004973	0.652204
99	0.761355	0.718660	0.997931	0.653599

ii. Model Performance and Loss Graphical Report

- Model performance : The training dataset show performance of 76.83% while Validation dataset shows performance of 65.38% as shown below

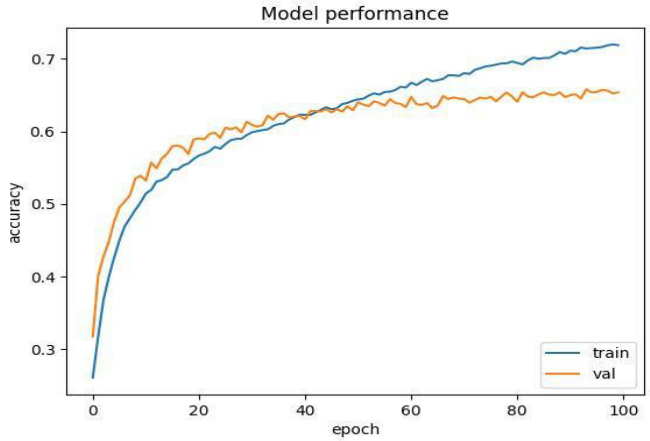

Figure 14. Graphical representation of Model performance

iii. Model loss: Fig 4.11 shows the validation loss does not decrease as much as the training loss. This suggests that the model is overfitting to the training dataset.

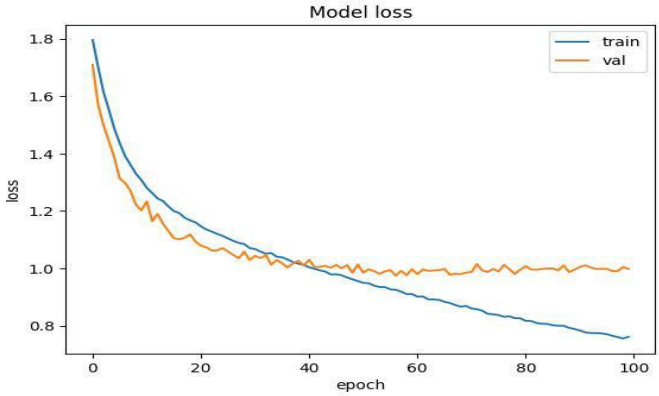

Figure 15. Graphical representation of model loss

v. Classification Report

- Training report: happy emotion has the highest precision while the disgust has the lowest precision.

Table 4. Training classification report

Emotion	Precisoin	Recall	F1-Score	Support
Angry	0.14	0.14	0.14	3995
Disgust	0.01	0.01	0.01	436
Fear	0.15	0.12	0.14	4097
Happy	0.25	0.27	0.26	7215
Neutral	0.17	0.20	0.19	4965
Sad	0.16	0.15	0.16	4830
Surprise	0.12	0.12	0.12	3171
Accuracy			0.18	28709
Maro avg	0.14	0.14	0.14	28709
Weight avg	0.18	0.18	0.18	28709

- Validation report: happy emotion has the highest precision while the disgust has the lowest precision.

Table 5. Validation classification report

Emotion	Precisoin	Recall	F1-Score	Support
Angry	0.14	0.15	0.15	958
Disgust	0.01	0.01	0.01	111
Fear	0.15	0.13	0.14	1024
Happy	0.26	0.26	0.26	1774
Neutral	0.19	0.21	0.20	1233
Sad	0.19	0.17	0.18	1247
Surprise	0.11	0.12	0.11	831
Accuracy			0.18	7178
Maro avg	0.15	0.15	0.15	7178
Weight avg	0.18	0.18	0.18	7178

Output shape and params computation

During training, the convolutional layers and dense layers generate learnable parameters while pooling layer reduces the dimensionality of the image at each layers of feature extraction. The param calculation is given as

Param = [shape of filter width * shape of filter height * no of filters in the previous layer + 1] * no of filters

a. **Table 6**. Conv layer_0

Filter	**Kernel size**	**Padding**	**Input shape**
32	(3, 3)	"same"	(48, 48, 1)

Output shape = (none, 48, 48, 32),

Param = [3 * 3 + 1] * 32 = 320 params

b. **Table 7. Conv layer_1**

Filter	**Kernel size**	**Pool Size**	**Input shape**
64	(3, 3)	(2, 2)	(48, 48, 1)

Output shape = (none, 46, 46, 64) with respect to eqn.(22),

Param = [3 * 3 * 32 + 1] * 64 = 18496 params,

Max pooling = 46 (input shape) / 2 (pool size) = 23

c. Table 8. Conv layer_2

Filter	Kernel size	Pool Size	Input shape
128	(3, 3)	(2, 2)	(23, 23, 1)

Output shape = (none, 21, 21, 128) with respect to eqn.(22),
Param = [3 * 3 * 64 + 1] * 128 = 73856 params,
Max pooling = 21 (input shape) / 2 (pool size) = 10.5 = 10.

d**. Table 9**. Conv layer_3

Filter	Kernel size	Pool Size	Input shape
256	(3, 3)	(2, 2)	(10, 10, 1)

Output shape = (none, 8, 8, 256) with respect to eqn.(22),
Param = [3 * 3 * 128 + 1] * 256 = 295168 params,
Max pooling = 8 (input shape) / 2 (pool size) = 4.

e. Flatten layer : Max pool of Conv layer_3 is (4, 4, 256), when flattened to a one-dimensional array, it given as [4 * 4 * 256] = 4096 neurons.

f. Dense layer: this is calculated by [current layer neuron * previous layer neuron] + current layer neuron.

- Dense layer_1: [512 * 4096] + 512 = 2097664 neurons.
- Dense layer_2: [7 * 512] + 7 = 3591 neurons.

Table 10. Sum of the parameter

Total params	Trainable params	Non-trainable param
2,489,095	2,489,095	0

Figure 16 shows the trainable Params for the sum parameter

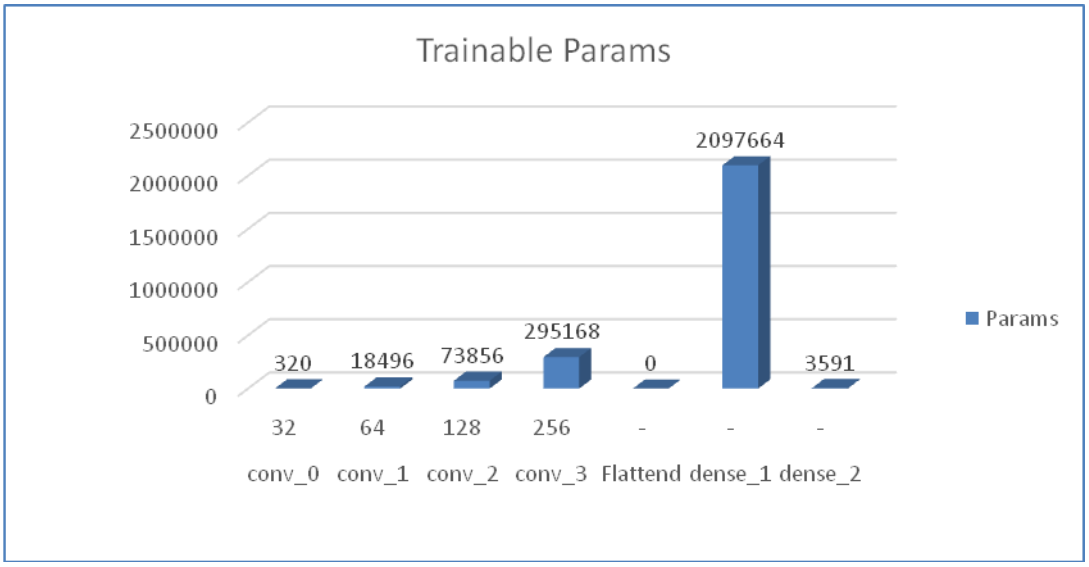

Figure 16. Parameter graphical representation

Deployment of Model

Once the model training is completed, it generates .h5 file which can then be deployed locally on VScode using OpenCV Library. It is loaded using the "import load_model" from keras

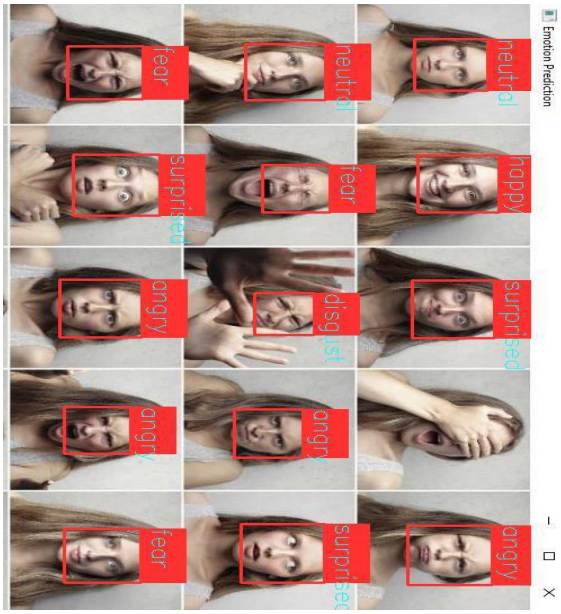

Figure 17a. Predicted Output Image

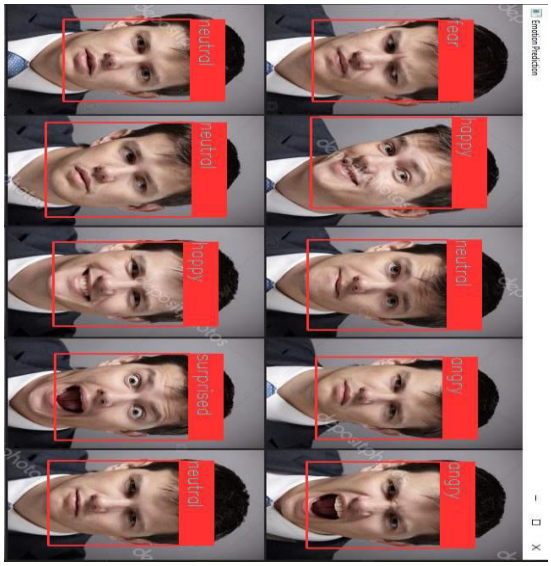

Figure17b. Predicted Output 2 Image

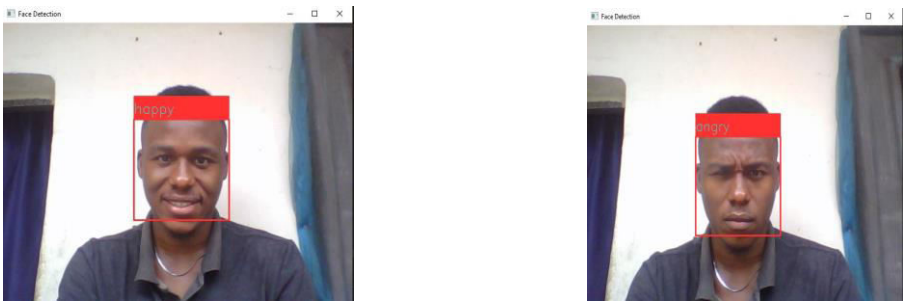

Figure. 18. Webcam Predicted Output Emotion for one of the researcher

Conclusion

CNN was developed to extract facial features to be able to detect and recognize emotion. It was trained on FER2013 dataset which consist a total of 35,887 images. The emotions considered are happy, sad, angry, neutral, disgust, surprise and fear.The CNN model on this report consist of 5 layers, three convolutional layer and two fully connected layers. The model was able to achieve a total 2,489,095 trainable parameter. The model was trained on a CPU processor at 100 epochs in 32 batches and it took 17 hours plus to complete this process. Fig 4.10 shows that the model generated an accuracy of 76.83% on the training dataset and 65.38% on the validation dataset which show it performs well but it still overfitted as at when training accuracy was at 65%, this means that the model is able to learn the training data very well, but it is not able to generalize to new data. The seven emotion where evaluated using precision, recall and f-score. The happy emotion has the best precision, recall and f-score of 0.26 while the disgust emotion has the worst precision, recall and f-score of 0.01. When thoroughly tune all hyperparameters towards an optimized model for facial emotion recognition. Adams optimizers and learning rate of 0.0001 was explored and the best classification accuracy achieved is 76.83%. We also carry out various data augmentation technique to increase the volume of data. For future work, we plan to explore other large dataset (e.g. imageNet) and investigate ensembles of different deep learning architectures to further improve our performance in facial emotion recognition.

References

1. Minaee, S.; Minaei, M. and Abdolrashidi, A (2021) "Deep-Emotion: Facial Expression Recognition Using Attentional Convolutional Networ*k*". Sensors 21, 3046
2. Shan, C., Gong, S. and McOwan, P.W (2022)"Facial expression recognition based on local binary patterns": *A comprehensive study Image and Vision Computing*,27(6),803-816, .
3. Kanade, T., Cohn., J. F. and Yingli ., (2000), "Comprehensive database for facial expression analysis," *Proceedings Fourth IEEE International Conference on Automatic Face and Gesture Recognition (Cat. No. PR00580)*, Grenoble, France, pp. 46-53,
4. Deng, G., Guo, J and Liu, X., (2021) "Deep Learning and Facial Expression Recognition," *in Handbook of Face Recognition*, 2nd ed., S. Z. Li and A. K. Jain, Eds. Springer, pp. 545-570
5. Arora, T. K. Chaubey, P. K. Raman, M. S. Kumar, B., Nagesh, P.K.Y., Anjani, H. M.S., Hamed, A., Ahmed, A., Balamuralitharan, S., and Debtera B., (2022)"Optimal Facial Feature Based Emotional Recognition Using Deep

Learning Algorithm"; *Computational Intelligence and Neuroscience* Volume, Article ID 8379202, 10

6. Kumar, B. K. Swaroopa, K. and. Balaga , T. R (2021)"Facial Emotion Recognition and Detection Using CNN". *Turkish Journal of Computer and Mathematics Education* Vol.12 No. 14,5960-5968, 2021
7. Pandey, A., Gupta,A., and Shyam, R.., (2022) "Facial Emotion Detection and Recognition". *International Journal of Engineering Applied Sciences and Technology*, Vol. 7, Issue 1, ISSN No. 2455-2143, Pages 176-179
8. Lierler, Y.Smith, S., Truszczynski, M., and Westlund, A. (2012)"Weighted-sequence problem": ASP vs CASP and declarative vs problem-oriented solving. In: Proceedings of the 14th International Conference on Practical Aspects of Declarative Languages, PADL'12, pp. 63–77.
9. Phung, V.H., and Rhee, E.J. (2018) "A Deep Learning Approach for Classification of Cloud Image Patches on Small Datasets". *J. Inf. Commun. Converg. Eng*. 16, 173–178.
10. Xavier, G. Bordes, A. and Bengio,, Y. (2011)"Deep Sparse Rectifier Neural Networks", *Proceedings of the 14th International Conference on Artificial Intelligence and Statistics*, vol. 15 of JMLR. Pp. 315-323
11. Qayyum, R.(2022) "Introduction To Pooling Layers In CNN"; 04, (3) 34
12. Krizhevsky, A., Sutskever, I., and Hinton, G.E., (2017) "Imagenet classification with deep convolutional neural networks". *Commun ACM*.;60(6):84–90
13. Alzubaidi, L. I. Amjad, J.Z., Humaidi, J. Ye Duan, A. Al. . Al-S., O.,. Santamaría, J., Mohammed, A., Muthana, F. and Farhan, Al-A. L. (2021) "Review of deep learning: concepts, CNN architectures, challenges, applications, future directions" 8,(12), 3465
14. Simonyan, K. and Zisserman, A. (2014) "Very Deep Convolutional Networks for Large-Scale Image Recognition",XIV:14, 09.1556
15. Kaiming , H., Zhang, X. Z., Shaoqing , R. and Sun, J., (2015) . "Deep Residual Learning for Image Recognition", XIV:15, 12.03385
16. Mehendale, N.(2022) "Facial emotion recognition using convolutional neural networks (FERC*),"SN Applied Sciences*, vol. 2, no. 3
17. Coskun, M., Yildirm, O., Ucair, A.,and Demir; Y., (2017) An Overview of Popular Deep Learning Method
18. Bhatt, D., Patel, C., Talsania, H., Patel, J., Vaghela, R. Pandya, S., Modi, K., and Ghayvat, H. (2021), "CNN Variants for Computer Vision: History, Architecture, Application, Challenges and Future Scope". *Electronics 202*, 10, 2470.