Minor Project

A Report On

“Facial Emotion Recognition”

Submitted By

Bikki Kumar(2K22/DSC/04)

MASTER OF TECHNOLOGY

IN

DATA SCIENCE

Under the guidance of

Dr. Ruchika Malhotra

(Professor, SE, DTU)

Department of Software Engineering

Delhi Technological University, Delhi

May-2023

DELHI TECHNOLOGICAL UNIVERSITY

SHAHBAD DAULATPUR, DELHI-110042

DECLARATION

I, **BIKKI KUMAR**, bearing **Roll No. 2K22/DSC/04,** student of second-semester MTech, Department of Software Engineering, Delhi Technological University, declare that the Project report entitled "Facial Emotion Recognition**"** has been carried out by me and submitted in partial fulfilment of the course requirements of a degree in **Mater of Technology** in **Data Science** from **Delhi Technological University** during the academic year **2022-2023**.

BIKKI KUMAR (2K22/DSC/04)

Date: 22th May 2023

Place: Delhi

CERTIFICATE

This is to confirm that **BIKKI KUMAR (2K22/DSC/04)** completed the Project **"Facial Emotion Recognition"** under my supervision as part of her MASTER OF TECHNOLOGY degree at DELHI TECHNOLOGICAL UNIVERSITY.

Place-Delhi

Date-

Dr. Ruchika Malhotra
(Project Supervisor)
(Professor, SE Department)

ABSTRACT

Facial expression detection is an emerging field with various applications in social robots, neuromarketing, and games. Nonverbal communication methods, including facial expressions, eye movements, and gestures, play a crucial role in human-computer interaction. Among these, facial emotion recognition is widely used as it provides insights into people's emotional states and feelings. However, accurately recognizing facial expressions poses challenges for machine learning methods due to the inherent variations in individuals' expression styles.

Even for the same person and facial expression, photos can differ in factors such as brightness, background, and position. These variations are further magnified when dealing with different subjects, taking into account differences in facial structure, ethnicity, and other factors. Consequently, facial expression recognition remains a complex topic in computer vision.

This project presents a straightforward approach to tackle facial expression recognition by combining a Convolutional Neural Network (CNN) with specific image pre-processing techniques. The CNN model is trained on the FER-13 dataset, which contains labeled facial images for seven fundamental human facial expressions. The proposed architecture consists of four convolutional layers followed by max pooling. By utilizing a single component architecture, the network efficiently classifies facial photos into their respective emotion categories.

The suggested architecture demonstrates comparable or superior accuracy and training time when compared to both state-of-the-art approaches and traditional CNNs. This highlights the effectiveness of the proposed approach in addressing facial expression recognition tasks.

By successfully tackling the complexities of facial expression recognition, our project contributes to advancing the field of computer vision and human-computer interaction. The ability to accurately interpret facial expressions in real-time has the potential to revolutionize various domains, including human-computer interfaces, emotion-driven applications, and virtual reality experiences.

ACKNOWLEDGMENT

I am grateful to Dr. Ruchika Malhotra (Professor, Department of Software Engineering) and all of the Department of Software Engineering faculty members at DTU. They all gave us a lot of help and advice for the minor project.

I'd also want to thank the University for providing us with the laboratories, infrastructure, testing facilities, and environment that allowed us to operate without interruption.

I'd also want to thank our lab assistants, seniors, and peer group for their assistance with all of their expertise on numerous issues.

Bikki Kumar

2K22/DSC/04

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVATIONS

Abbreviation Name	Term Used
FER	Facial Emotion Recognition
NN	Neural Network
LSTM	Long Short-Term Memory
ReLU	Rectified Linear Activation Unit
FER-2013	Facial Emotion Recognition-2013
DL	Deep Learning
CK+	Cohn Kanade +
I/P	Input
NLA	Non-Linear Activation
O/P	Output
ELU	Exponential Linear Unit
AF	Activation Function
TA	Training Accuracy
VA	Validation Accuracy
TL	Training Loss
VL	Validation Loss

CHAPTER 1

INTRODUCTION

Human social communication relies heavily on facial expressions. In face-to-face communication,facial expressions are more powerful than words. Body language accounts for 55%, words accountfor 7% and voice tone accounts for 38%, for effective message transmission. The emotion is a partof non-verbal communication and cannot be hidden by a person. Looking at someone's face can provide us insight into their emotions. In general, facial expression recognition seeks to distinguishand classify the meaningful motions of various facial muscles into distinct emotion categories. Some of the applications of facial expression recognition include computer vision, nonverbal human psychology, and human's inter-communication with computer. Despite the fact that it is sometimes more challenging due to the backgrounds and low-resolution faces, facial expression identification in photographs has recently gained popularity. Feelings, energy, and dispositional impacts are all aspects of emotion. Humans use facial expression to communicate their emotionalstates and intentions. It is one of the most powerful, natural, and widespread signals. In this project,we will explain how to identify facial expressions based on a convolutional neural network. The image is sent to the system and the CNN is used to predict facial expressions. It should be either sadness, anger, happiness, surprise, fear, disgust, or neutrality. The objective of work is to categorization of facial expressions in seven non-verbal expressions shown by human.The objective of this project is to develop a methodology for identifying facial expressions based on a convolutional neural network (CNN). The system receives an image as input, and the CNN predicts the corresponding facial expression, which can be categorized as sadness, anger, happiness, surprise, fear, disgust, or neutrality. By categorizing facial expressions into these seven non-verbal expressions, we aim to gain a deeper understanding of human emotional states and enhance communication between humans and computers.Emotions encompass a range of aspects including feelings, energy, and dispositional impacts. Humans naturally utilize facial expressions to communicate their emotional states and intentions.

CHAPTER 2

RELATED WORK

Table 1: Related Work

Author Name	**Model/Implementation**	**Dataset**	**Accuracy**
Mollahosseini et al.(2016)	CNN	MultiPie, MMI, DISFA, FERA, SFEW, CK+, FER2013	94.7%, 77.9%, 55%, 76.7%, 47.7%, 93,2%, 61.1%
Lopes et al. (2017)	CNN	CK+, JAFFE, BU-3DFE	96.76% for CK+
Mohammadpour etal. (2017)	CNN	CK+	97.01%
Cai et al. (2018)	SBN-CNN	JAFFE, CK+	95.24%, 96.87%
Li et al. (2018)	ACNN	RAF-DB, AffectNet	80.54%, 54.84%
Yolcu et al. (2018)	CNN	RafD	94.44%
Deepak jain et al. (2018)	CNN	JAFFE, CK+	95.23%, 93.24%
Yu et al. (2018)	STC-NLSTM	CK+, Oulu-CASIA,MMI, BP4D	99.8%, 93.45%, 84.53%
Agrawal et Mittal.(2019)	CNN	FER2013	65%
Kim et al. (2019)	CNN-LSTM	MMI, CASME II	78.61%, 60.98%
Liang et al.(2020)	DCBiLSTM	CK+, Oulu-CASIA, MMI	99.6%, 91.07%, 80.71%

CHAPTER 3

RESEARCH METHODOLOGY

3.1 DATASET COLLECTION

The dataset utilized in this project is specifically referred to as the Facial Expression Recognition 2013 Dataset (FER-2013). This dataset serves as a fundamental resource for training and evaluating facial expression recognition models, playing a crucial role in the successful implementation of our project. By leveraging the diverse and well-curated samples within the FER-2013 dataset, we aim to achieve robust and accurate facial expression classification capabilities. The availability of this dataset has significantly contributed to the advancements in the field of facial expression recognition, enabling researchers and practitioners to explore innovative approaches and push the boundaries of this domain.

The dataset encompasses a vast collection of approximately 30,000 RGB images capturing various human facial expressions. These images serve as valuable samples for training and evaluating models in the field of facial expression recognition. The dataset's substantial size allows for a comprehensive exploration of the diverse range of facial expressions exhibited by individuals. By leveraging this extensive collection of facial images, our project aims to enhance the accuracy and robustness of our facial expression recognition system.

The images belonging to the dataset can be divided into seven categories and each category isassigned a label which are as follows:

- Angry (Label =0)
- Disgust (Label =1)
- Fear (Label =2)
- Happy (Label =3)
- Sad (Label =4)
- Surprise (Label =5)
- Neutral (Label =6)

The faces have been automatically registered such that they are more or less centered in each image and take up around the same amount of area.

Table 2: Dataset Image Distribution

Label	**Emotion**	**Number of images for training**	**Number of images for testing**
0	Angry	3995	958
1	Disgust	436	111
2	Fear	4097	1024
3	Happy	7215	1774
4	Sad	4830	1247
5	Surprised	3171	831
6	Neutral	4965	1233
Total		**28709**	**7178**

3.1.1 Dataset Sample:

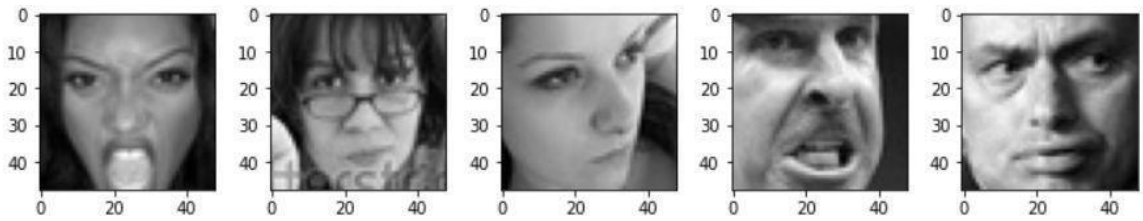

Fig 1: Sample Images for Angry Class

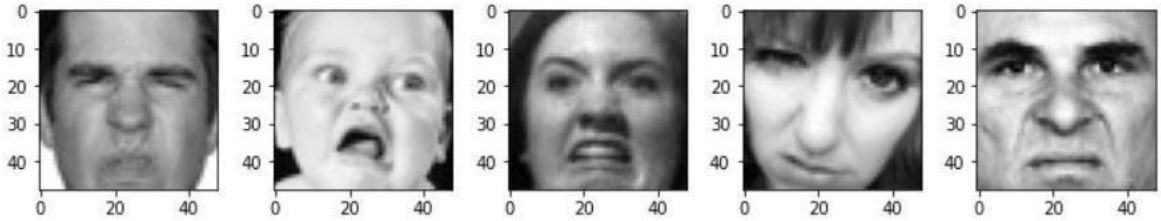

Fig 2: Sample Images for Disgust Class

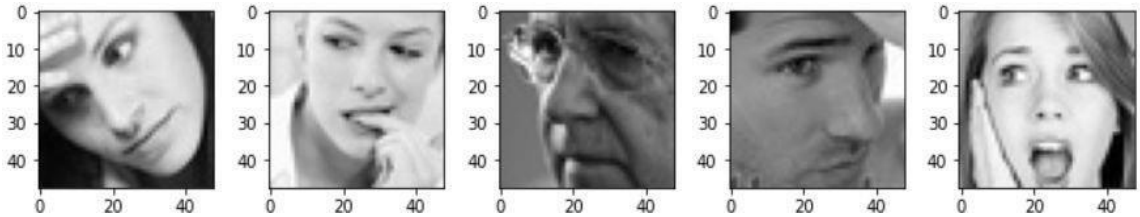

Fig 3: Sample Images for Fear Class

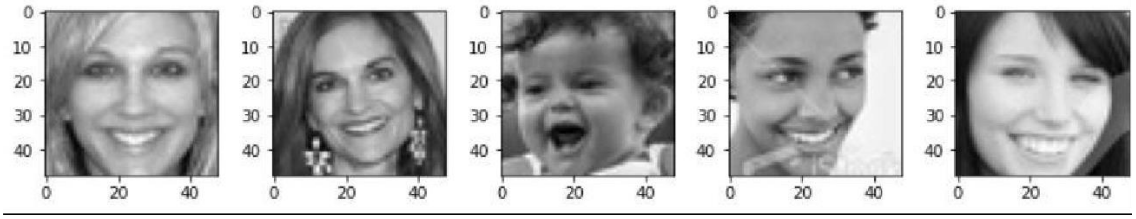

Fig 4: Sample Images for Happy Class

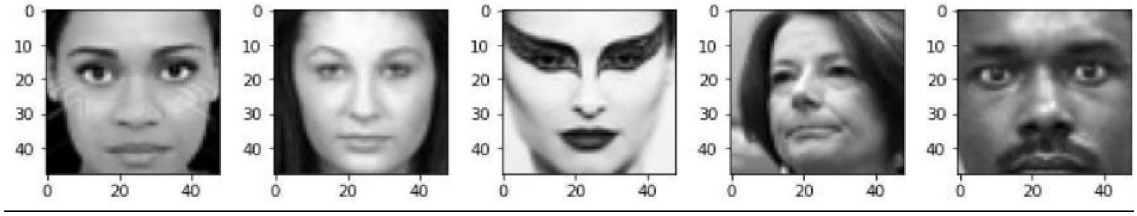

Fig 5: Sample Images for Sad Class

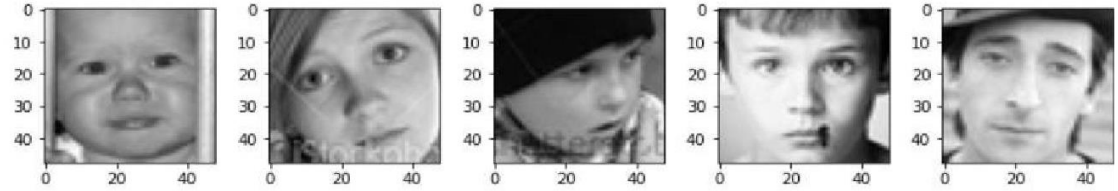

Fig 6: Sample Images for Neutral Class

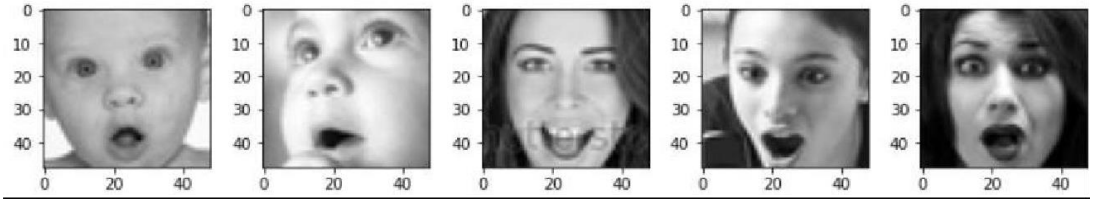

Fig 7: Sample Images for Surprised Class

3.2 DATA PREPROCESSING

Image Resizing

The suggested CNN model accepts images up to 48*48 pixels in size at the i/p layer. As a result, images with dimensions other than this will be rejected by the CNN model's i/p layer. Therefore, image scaling was used to resolve the problem.

RGB to Grayscale Conversion

The dataset contains RGB human facial images. Converting the RGB image to grayscale will reduce the channels of the image from three to one. So, it will make the computation faster.

3.3 CLASSIFICATION

A neuron is the most basic component of a NN. When the i/p to neurons are weighted and then combined, information is propagated forward. This information is sent to a NLA function

thatcontains a bias component that shifts the o/p.

The o/p ranges from 0 to 1, making it suited for probability situations. Because most real-world data is nonlinear, the AF's goal is to inject nonlinearities into the network. NNs can also approximate complex functions by using a nonlinear function

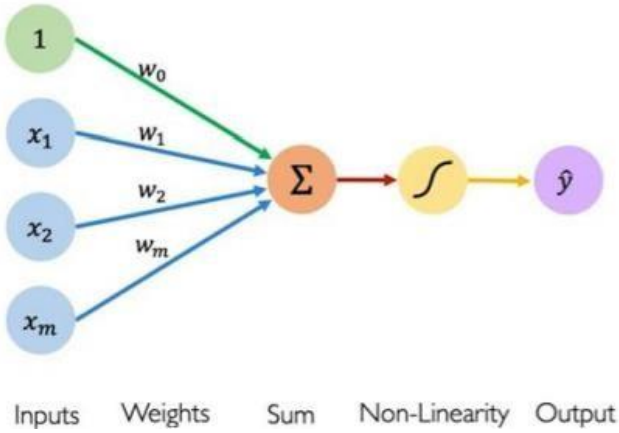

Fig 8: The Basic Structure of a Neuron

Individual neurons can be combined to create a multi-output neural network. Each input is connected to each neuron inside the layer in the case of dense or fully connected layers. Multiple hidden layers are stacked to create deep neural networks (DNNs), where each hidden layer's neurons are coupled to those in the layer before it. The network can learn complicated representations and hierarchical properties from the input data thanks to this layered connectivity architecture, which also makes it easier for information to move back and forth. DNNs are useful for a variety of tasks, including classification, regression, and multi-output prediction, since they can capture intricate relationships and extract high-level representations from the input by merging these interconnected neurons and utilizing the depth of the network.

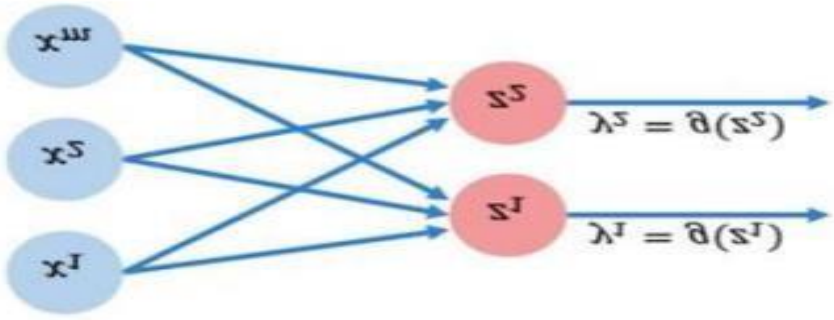

Fig 9: A Neural Network with multiple outputs.

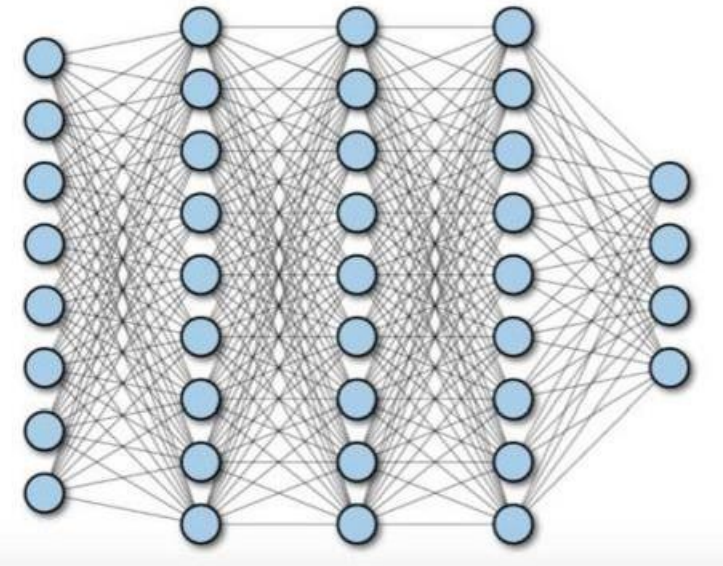

Fig 10: A Fully Connected NN

1) Convolution Neural Network

CNN is a type of more powerful DL algorithm that takes image as an i/p, assigns meanings to different aspects in theimage, and then distinguishes them. A CNN requires substantially less preparation than conventional classification techniques. CNN's design is inspired by the tissues of the visualcortex and resembles the human brain's connecting network of neurons. One of the features ofCNN is that it compresses the image into an easy-to-use format while preserving importantinformation for accurate judgement. This is important for building an architecture that can bescaled to large datasets while learning features.

The following are the primary twelve CNN operations:

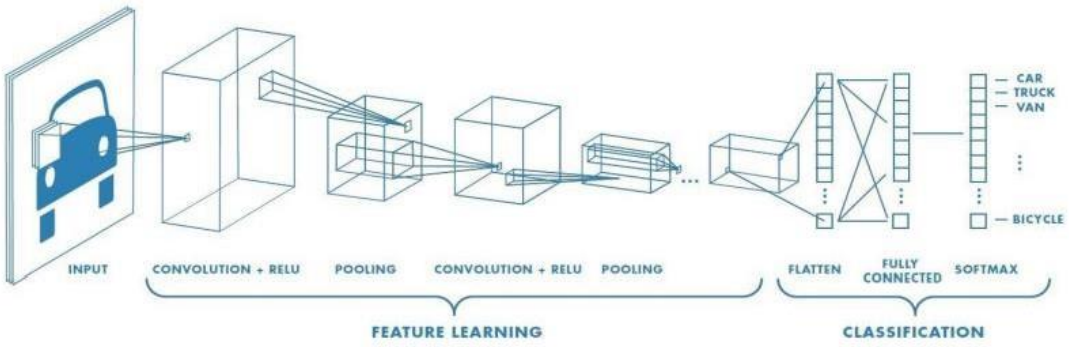

Fig 11: The CNN Operations

a) Convolution operation

Convolution is a technique used to extract and isolate fine information from an input image, like edges. The convolution layer performs the following tasks:

- Edges, color, gradient orientation, and rudimentary textures are all learned by the first convolutional layer.
- The following convolutional layers learns increasingly complicated texturesand patterns.
- Objects or portions of objects are learned in the final convolutional layer.

The kernel serves as the fundamental component responsible for executing the convolution operation. Its purpose is to selectively extract the essential information from the feature map by filtering out irrelevant data. Moving in specified stride lengths, the kernel scans the image from left to right, covering its entire width. Once complete, the process repeats from the leftmost side, progressively encompassing the entire image with the same stride length.

With a stride length of one, the kernel undergoes nine movements, executing matrix multiplication between itself and the corresponding portion of the underlying image during each step.

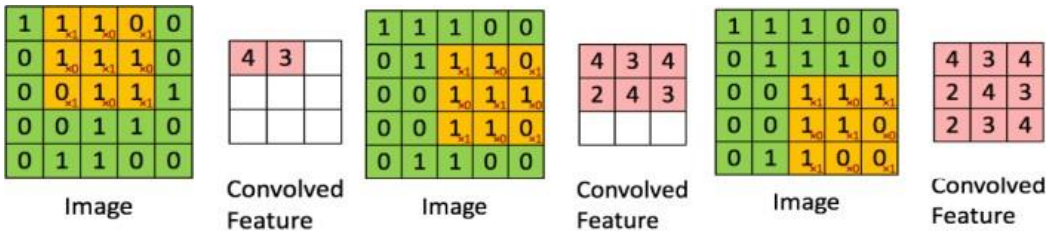

Fig 12: Convolution Operation.

The kernel can have the same dimensions as the convolved feature. Same or valid padding is used to accomplish this.

b) Pooling Operation:

The pooling layer is responsible for reducing the spatial size of convolved features in a neural network. This down sampling process aims to improve computational efficiency and capture essential features that are invariant to rotation and position. Pooling can be categorized into two types: maximum pooling and average pooling. In maximum pooling, the highest value

within a specific region of the image, defined by the pooling kernel, is selected and returned. On the other hand, average pooling computes the average value of the pixels within the pooling region and returns it as the output. By applying pooling operations, the network can condense the information while preserving the most relevant features, facilitating faster processing and enhancing the network's ability to learn abstract representations from the data.

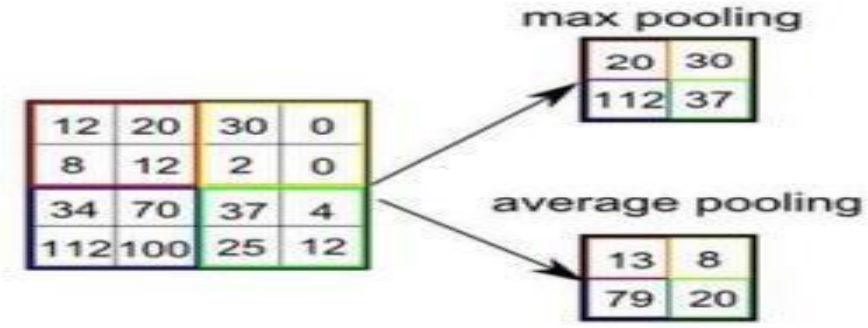

Fig 13: Pooling Operation.

c) Fully Connected Layer

All neurons in the previous layer are connected to neurons in a completely connected layer. A CNN's last layer is this layer. The o/p of this layer is obtained by flattening thei/p from the previous layer into a one-dimensional vector and applying an AF.

d) Dropout

The dropout rate is the probability of training a given node in a layer, with 1.0 indicating nodropout and 0.0 indicating that all layer o/p are disregarded.

Dropout is a regularization technique employed to mitigate the risk of overfitting in machine learning models. Overfitting arises when the training accuracy (TA) surpasses the accuracy achieved on the testing data. Dropout involves randomly excluding neurons from the network during specific forward or backward passes, effectively reducing the network's size.

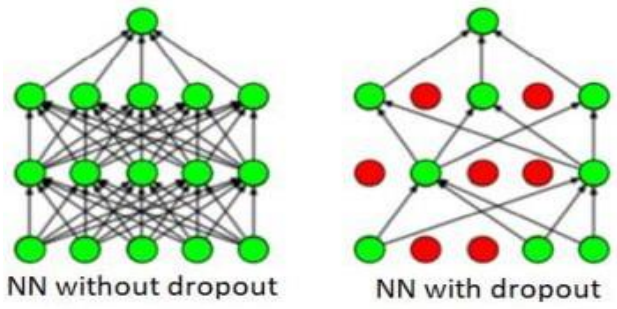

Fig 14: Dropout Operation.

e) Batch Normalization

When the distributions of the layer i/ps are the same, training a network is more efficient. A model can be biased by variations in these distributions. The i/p to the layers are normalized using batch normalization

f) Activation Function

SoftMax and ELU are frequently utilized activation functions in Convolutional Neural Networks (CNNs). The SoftMax function, defined below, is employed to compute the probabilities of multiple classes:

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

The fully connected layer of the suggested models uses the SoftMax function, where "zi" stands for the input values and "K" stands for the number of input values. This function converts real numbers into probabilities and ensures that the results add to one and are between zero and one. The outcomes can be understood as a probability distribution that represents the seven emotions by using SoftMax.

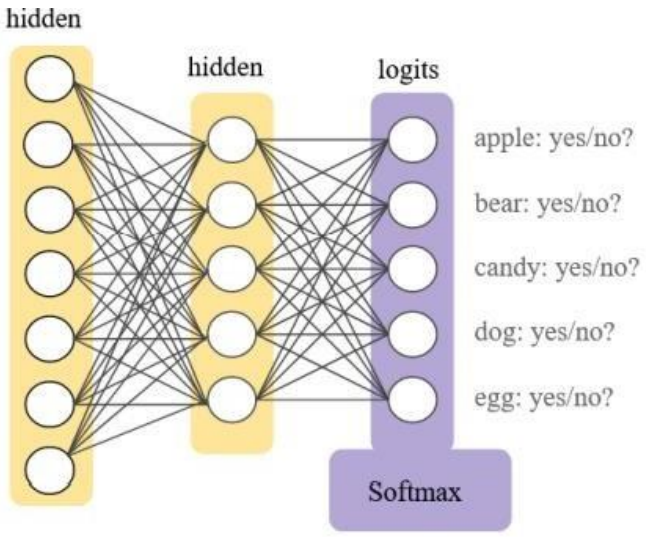

Fig 15: SoftMax Function

2) CNN Architecture

The layout of the CNN model implemented in this project is as follows:

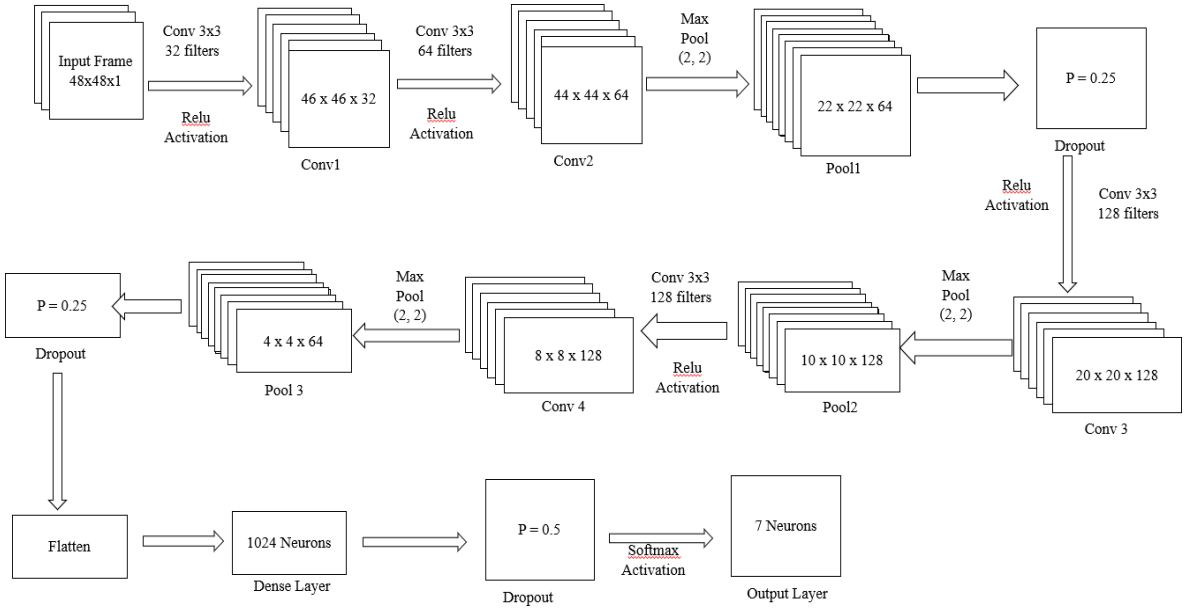

Fig 16: CNN Model Architecture

Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 48, 48, 64)	640
batch_normalization (BatchNo	(None, 48, 48, 64)	256
activation (Activation)	(None, 48, 48, 64)	0
max_pooling2d (MaxPooling2D)	(None, 24, 24, 64)	0
dropout (Dropout)	(None, 24, 24, 64)	0
conv2d_1 (Conv2D)	(None, 24, 24, 128)	204928
batch_normalization_1 (Batch	(None, 24, 24, 128)	512
activation_1 (Activation)	(None, 24, 24, 128)	0
max_pooling2d_1 (MaxPooling2	(None, 12, 12, 128)	0
dropout_1 (Dropout)	(None, 12, 12, 128)	0
conv2d_2 (Conv2D)	(None, 12, 12, 512)	590336
batch_normalization_2 (Batch	(None, 12, 12, 512)	2048
activation_2 (Activation)	(None, 12, 12, 512)	0
max_pooling2d_2 (MaxPooling2	(None, 6, 6, 512)	0

dropout_2 (Dropout)	(None, 6, 6, 512)	0
conv2d_3 (Conv2D)	(None, 6, 6, 512)	2359808
batch_normalization_3 (Batch	(None, 6, 6, 512)	2048
activation_3 (Activation)	(None, 6, 6, 512)	0
max_pooling2d_3 (MaxPooling2	(None, 3, 3, 512)	0
dropout_3 (Dropout)	(None, 3, 3, 512)	0
flatten (Flatten)	(None, 4608)	0
dense (Dense)	(None, 256)	1179904
batch_normalization_4 (Batch	(None, 256)	1024
activation_4 (Activation)	(None, 256)	0
dropout_4 (Dropout)	(None, 256)	0
dense_1 (Dense)	(None, 512)	131584
batch_normalization_5 (Batch	(None, 512)	2048
activation_5 (Activation)	(None, 512)	0
dropout_5 (Dropout)	(None, 512)	0

Fig 17: Convolution Parameters

Convolutional features refer to the patterns or features extracted from input data using convolutional neural networks (CNNs). CNNs are widely used in computer vision tasks and excel at capturing local patterns or structures in images.

Convolutional features are obtained by applying convolution operations to the input data. These operations involve sliding a small filter or kernel over the input, performing element-wise multiplication between the filter and the corresponding input patch, and then summing up the results. This process is repeated across the entire input, producing feature maps that highlight relevant patterns or features.

The filters or kernels used in convolutional layers of CNNs are typically learned through training. These filters are designed to detect different types of visual patterns such as edges, corners, textures, or more complex structures. As the network learns, it adjusts the weights of the filters to enhance their ability to capture useful features.

CNNs excel in extracting convolutional features that can represent the input data in a

hierarchical manner, progressively capturing more intricate and sophisticated representations. The lower-level layers of the network are adept at detecting fundamental features such as edges or textures, while the higher-level layers specialize in capturing abstract and specific characteristics. These extracted features find utility in a multitude of applications, including object recognition, image classification, segmentation, and other computer vision tasks.

In a convolutional neural network (CNN), there are several parameters associated with the convolution operation. Here are the main parameters:

1. **Kernel/Filter**: A kernel or filter is a small matrix that is convolved with the input data. The size of the kernel determines the receptive field of the convolution and the type of patterns it can capture. The kernel is typically learned during the training process.

2. **Stride**: The stride parameter determines the number of steps taken by the kernel as it traverses the input data. A stride value of 1 signifies that the kernel moves one step at a time, whereas a stride of 2 indicates a movement of two steps at a time. By using a larger stride, the spatial dimensions of the output feature map are reduced.

3. **Padding**: Padding is the process of adding extra pixels to the input data to preserve spatial dimensions after the convolution operation. Padding can be used to control the size of the output feature map. Common padding options include 'valid' (no padding) and 'same' (padding to ensure the output size is the same as the input size).

4. **Dilation**: Dilation is an optional parameter that controls the spacing between the kernel elements. A dilation rate of 1 means no spacing, and a higher dilation rate introduces gaps between the kernel elements. Dilation can help increase the receptive field without increasing the number of parameters.

Number of Filters: In CNNs, multiple filters are employed simultaneously to capture diverse features. The number of filters determines the depth or the count of output feature maps generated. Each filter conducts a separate convolution operation, resulting in a distinct feature map.

CHAPTER 4

PERFORMANCE EVALUATION MEASURE

The measures that are evaluating the performance of the CNN model implemented are as follows:

4.1 ACCURACY

A common metric for assessing a classification model's performance is accuracy, which is often expressed as a percentage. It reflects the percentage of times the model correctly predicted a value when the predicted value matched the actual value.

It is given by:

$$\text{Accuracy} = \frac{\text{Number of correct predictions}}{\text{Total number of predictions}}$$

4.2 LOSS

The difference between a predicted value and the true value, taking into account the related probability or uncertainty, is taken into account by a loss function, also known as a cost function. This offers a more thorough assessment of the model's performance.

$$\text{Loss} = -\sum_{c=1}^{m}\left(y_{o,c}\log(p_{o,c})\right)$$

Unlike accuracy, which measures the percentage of correct predictions, loss quantifies the cumulative errors made for each sample in the training or validation sets. During the training process, the loss is frequently employed to guide the search for optimal parameter values, such as weights in a neural network. The objective of the training procedure is to minimize this loss value.

Several commonly used loss functions include log loss, cross-entropy loss, mean squared error, and probability loss. These functions serve different purposes and are selected based on the specific requirements of the problem at hand.

The specific mathematical formulation of the loss function varies depending on the chosen approach and problem domain. It encapsulates the mathematical relationship between

predicted values and true values, incorporating factors such as error magnitudes and probabilities.

In summary, the loss function plays a crucial role in evaluating and refining machine learning models. By considering the discrepancies between predicted and true values, it provides valuable insights into the model's performance and guides the training process to minimize errors and improve accuracy.

where y is a bin Considering a scenario with m classes (happy, sad, neutral, fear, angry, disgust, surprised), the predicted probability p is compared with a binary indicator y, which takes the value of 0 or 1.

In classification tasks, common loss functions include cross entropy or categorical loss, which are often used with SoftMax. Activation for multiclass classification. For regression tasks, mean squared error (MSE) loss is commonly used. During the training process, the loss is continuously updated and minimized through optimization algorithms such as gradient descent. As the model gets trained, it aims to reduce the loss, which ideally leads to improved accuracy on the validation or test data. It’s important to note that accuracy and loss are complementary metrics.

4.3 PRECISION

Precision is a metric that measures the accuracy of positive predictions made by a classification model. It evaluates how well the model identifies true positive instances while minimizing false positives. In other words, precision quantifies the proportion of correctly predicted positive instances out of all instances that were predicted as positive.

Precision can be calculated using the formula: Precision = TP / (TP + FP)

Here, TP refers to the number of true positive predictions, which are instances correctly identified as positive by the model. FP represents the number of false positive predictions, which are instances incorrectly classified as positive when they are actually negative.

A high precision value indicates a low rate of false positives, meaning that the model is more precise in identifying positive instances. However, a high precision value does not guarantee a comprehensive identification of all positive instances in the dataset.

4.4 RECALL

Recall, also known as sensitivity or true positive rate, is a metric that assesses the model's ability to capture all positive instances in a dataset. It quantifies the proportion of correctly predicted positive instances out of all actual positive instances in the dataset.

Recall can be calculated using the formula: Recall = TP / (TP + FN)

Here, TP represents the number of true positive predictions, and FN denotes the number of false negative predictions. False negatives occur when the model incorrectly predicts negative for instances that are actually positive.

A high recall value indicates a low rate of false negatives, indicating that the model is effective at identifying positive instances in the dataset. However, a high recall value does not necessarily imply a low rate of false positives.

In summary, precision emphasizes the accuracy of positive predictions, while recall focuses on the model's ability to capture all positive instances. These metrics provide complementary insights into the performance of a classification model and are often used together to evaluate its effectiveness.

CHAPTER 5

RESULT

5.1 EVALUATING MODEL's ACCURACY

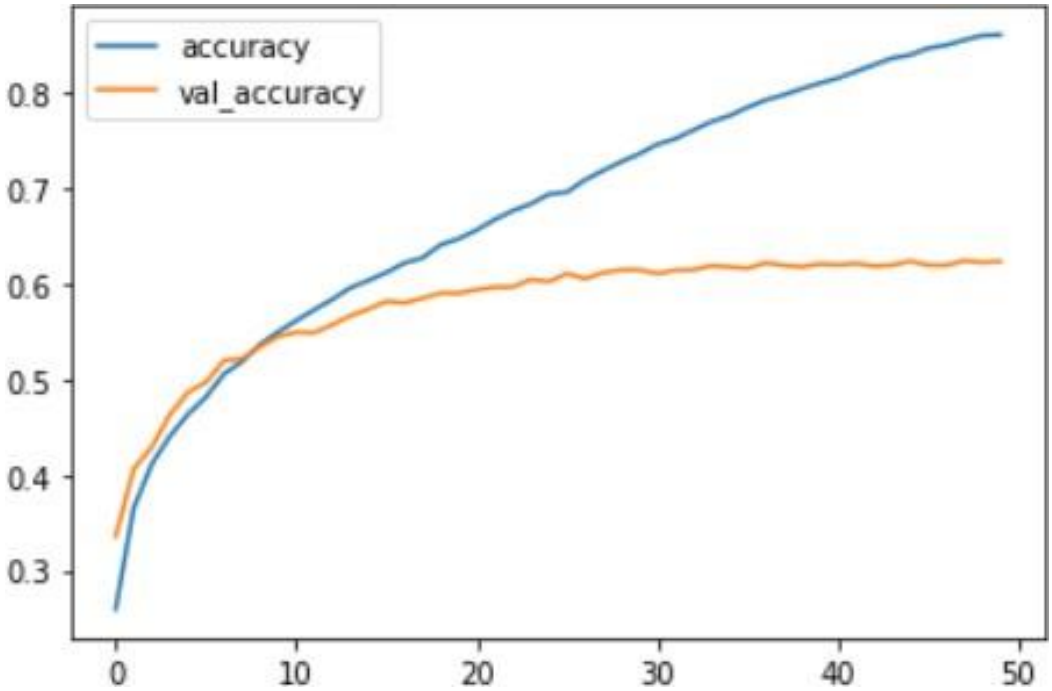

Fig 18: Implemented model's accuracy

The Convolutional Neural Network (CNN) model achieved a Training Accuracy (TA) of 86.13% and a Validation Accuracy (VA) of 62.39%. The Training Accuracy (TA) refers to the percentage of correctly predicted samples within the training dataset. It indicates how well the model performs on the data it was trained on. In this case, the CNN model achieved a TA of 86.13%, implying that it accurately predicted the correct output for 86.13% of the samples in the training dataset.

On the other side, the Validation Accuracy (VA) shows the proportion of samples in the validation dataset that were properly predicted. Typically, the model's performance on new data during training is assessed using the validation dataset. The CNN model's VA of 62.39% showed that it was less accurate than the training dataset. This could be a sign of overfitting, where the model may be having trouble generalizing to new, untried data because it is memorizing the training data too well.

It's critical to remember that these accuracy scores shed light on how well the model performed on the particular datasets used for both training and validation. However, the model's actual performance should be tested on a different, unknown test dataset and further evaluated using other assessment criteria. It is also worth considering potential strategies for improving the model's performance, such as adjusting hyperparameters, increasing the dataset size, implementing regularization techniques, or exploring different architectures.

5.2 EVALUATING MODEL's LOSS

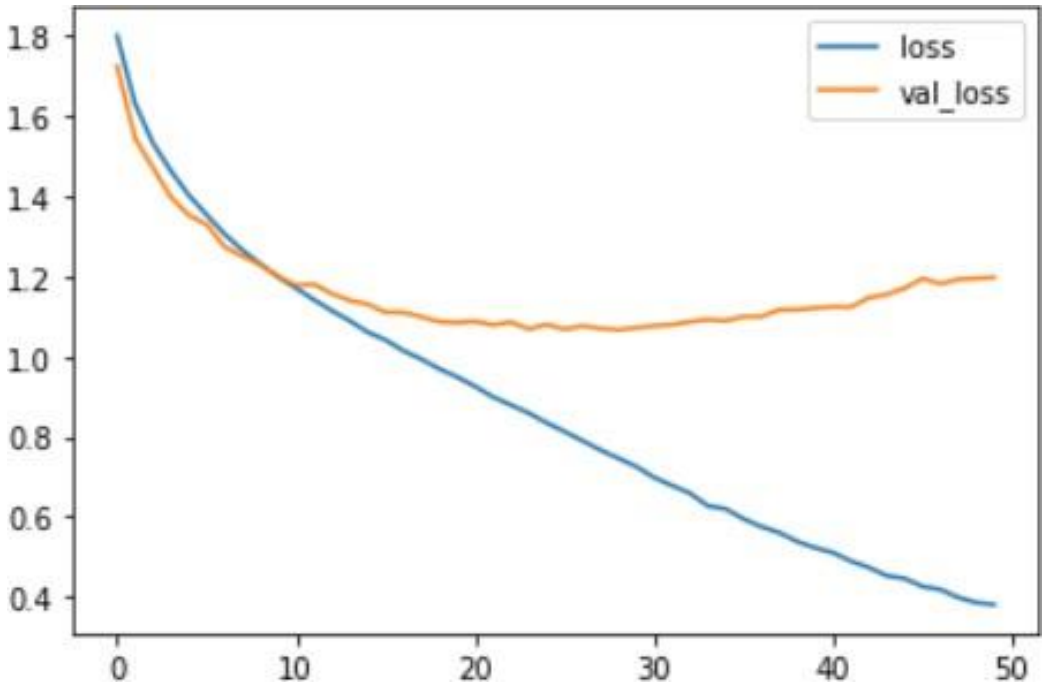

Fig 19: Implemented model's loss

The CNN model achieved a Training Loss (TL) of 0.38 and a Validation Loss (VL) of 1.19. The Training Loss (TL) is a measure of the error or discrepancy between the model's predicted output and the true output on the training dataset. A lower TL indicates that the model's predictions are closer to the true values for the training data. In this case, the CNN model achieved a TL of 0.38, suggesting that the model's predictions have relatively low error on the training dataset.

The Validation Loss (VL) measures the error between the model's predictions and the true values on the validation dataset. The validation dataset is used to assess how well the model generalizes to unseen data. Ideally, the model should have a low VL, indicating that it can

accurately predict outputs on new, unseen samples. In this instance, the CNN model achieved a VL of 1.19, suggesting that there is more error in the model's predictions on the validation dataset compared to the training dataset.

This is very important to note that TL and VL values are specific to the datasets used for training and validation, respectively. Lower TL and VL values are generally desirable, as they indicate better performance and a closer alignment between the model's predictions and the true values.

However, it's crucial to consider additional evaluation metrics and potentially investigate the model's performance on an independent test dataset to obtain a more comprehensive understanding of its generalization capabilities. Additionally, further optimization techniques, hyperparameter tuning, or model adjustments may be explored to improve the model's performance and reduce the VL.

5.3 TESTING MODEL'S PERFORMANCE ON LIVE HUMAN FACIAL INSTANCES

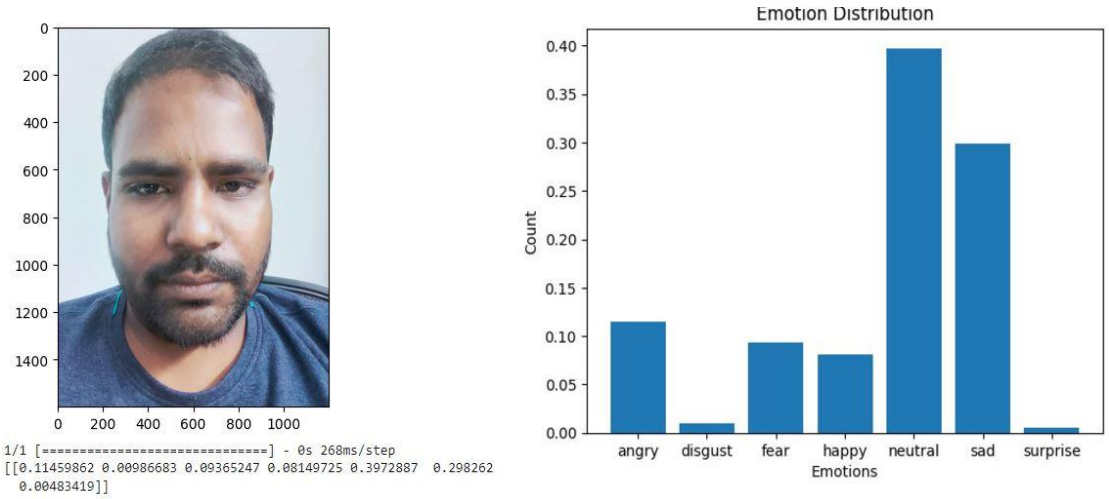

Fig 20: Live Model Testing-1

The trained model is fed with a live image of happy human face. The model gave probabilities of different emotions, but the highest probability was given for the happy emotion. Therefore, the model predicted the output is neutral.

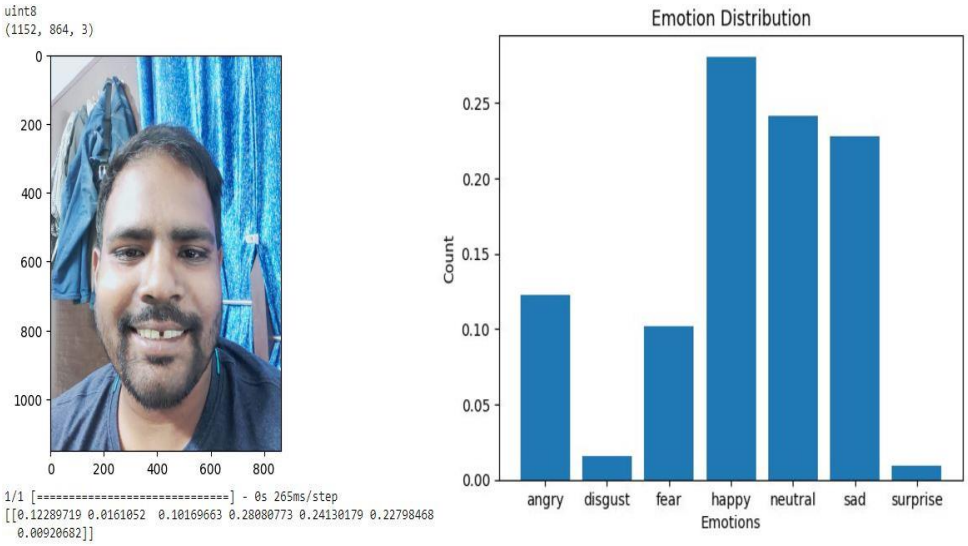

Fig 21: Live Model Testing-2

The trained model is fed with a live image of sad human face. The model gave probabilitiesof different emotions, but the highest probability was given for the sad emotion. Therefore,the model predicted the correct emotion.

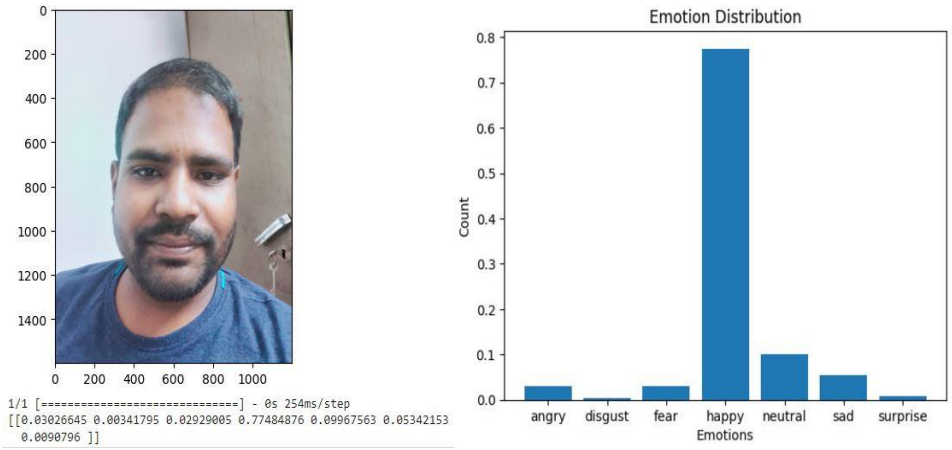

Fig 22: Live Model Testing -3

The trained model is presented with a real-time image featuring a human face expressing sadness. After analyzing the image, the model assigns probabilities to different emotions. Surprisingly, the highest probability is attributed to the emotion of happiness. Despite the

initial expectation of sadness, the model's prediction aligns with the visible expression of happiness in the image.

Furthermore, alongside the prediction, there is an accompanying performance array consisting of different values. This array represents the probabilities assigned to various emotions by the model. By identifying the maximum value within this array, we can determine the dominant emotion predicted by the model.

To visually represent this outcome, a bar graph is generated. The x-axis of the graph corresponds to the emotions, including categories such as angry, disgust, fear, happy, neutral, sad, and surprise. The y-axis represents the probabilities or values associated with each emotion. By observing the bar graph, we can identify the emotion with the highest value, which aligns with the model's correct prediction of happiness for the given image.

CONCLUSION AND FUTURE WORK

The goal of this project was to implement the CNN model on the FER-2013 dataset to classify human facial expressions into one of seven emotions (happy, sad, surprise, fear, anger, disgust, and neutral). We developed a facial expression identification system that combined established methods such as Convolutional Neural Networks with image pre-processing procedures. The model was trained for a total of 50 epochs. The TA and VA obtained were both 86.13 percent and

62.39 percent, respectively. The TL and VL were 0.38 and 1.19, respectively. Regularization canbe used to counteract the overfitting problem. Because our dataset is unbalanced, we can use unbalanced data treatment methods like data augmentation. Instead of static graphics, we can makeour system work on a live video feed.

In this project, the objective was to implement a Convolutional Neural Network (CNN) model on the FER-2013 dataset for the purpose of classifying human facial expressions into seven different emotions: happy, sad, surprise, fear, anger, disgust, and neutral.

The implemented system combined well-established techniques such as CNNs with image pre-processing procedures to develop a facial expression identification system. The model underwent training for a total of 50 epochs, which indicates that it went through 50 iterations of the training process to optimize its performance.

The obtained results from the training and validation phases were as follows: the Training Accuracy (TA) and Validation Accuracy (VA) both achieved a percentage of 86.13 percent, indicating that the model correctly classified the emotions in the dataset for the training and validation sets with a relatively high accuracy. However, the Validation Accuracy (VA) of 62.39 percent suggests that the model's performance may have been affected by overfitting, as it achieved a lower accuracy on unseen data compared to the training data.

The Training Loss (TL) and Validation Loss (VL) values were 0.38 and 1.19, respectively. The lower TL value indicates that the model had a small discrepancy between its predictions and the true values on the training dataset. However, the higher VL value suggests that the model's predictions had a larger error on the validation dataset, indicating potential difficulties in generalizing to unseen data.

To address the issue of overfitting, regularization techniques can be applied. Regularization helps prevent the model from memorizing the training data too precisely and promotes better generalization. Additionally, since the dataset is unbalanced, data augmentation techniques can be employed to artificially increase the diversity of the data and improve the model's ability to handle various facial expressions.

Furthermore, instead of solely working with static graphics, the system can be extended to operate on live video feeds. This would allow real-time facial expression classification, providing a more dynamic and interactive user experience.

Overall, the project aimed to implement a CNN model for facial expression classification, achieved relatively high accuracy on the training data, and identified areas for improvement, such as addressing overfitting and unbalanced data, as well as incorporating live video feed functionality.

REFERENCES

[1] Agrawal, A., & Mittal, N. (2020). Using CNN for facial expression recognition: a study of the effects of kernel size and number of filters on accuracy. Visual Computer, 36(2), 405–412. https://doi.org/10.1007/s00371-019-01630-9

[2] Cai, J., Chang, O., Tang, X. L., Xu, C., & Wei, C.(2018). Facial Expression Recognition Method Based on Sparse Batch Normalization CNN. Chinese Control Conference, CCC, 2018-July, 9608–9613. https://doi.org/10.23919/ChiCC.2018.8483567

[3] Cai, J., Chang, O., Tang, X. L., Xu, C., & Wei, C. (2018). Facial Expression Recognition Method Based on Sparse Batch Normalization CNN. Chinese Control Conference, CCC, 2018-July, 9608–9613. https://doi.org/10.23919/ChiCC.2018.8483567

[4] Fathallah, A., Abdi, L., & Dodik, A. (2018). Facial expression recognition via deep learning. Proceedings of IEEE/ACS International Conference on Computer Systems and Applications, AICCSA, 2017-October, 745–750. https://doi.org/10.1109/AICCSA.2017.12

[5] Jain, D. K., Shamsolmoali, P., & Sehdev, P. (2019). Extended deep neural network for facial Emotion recognition. Pattern Recognition Letters, 120, 69–74. https://doi.org/10.1

[6] Jaiswal, A., Krishnama Raju, A., & Deb, S. (2020). Facial emotion detection using deep learning. 2020 International Conference for Emerging Technology, INCET 2020, 1–5. https://doi.org/10.1109/INCET49848.2020.9154121

[7] Kim, D. H., Baddar, W. J., Jang, J., & Ro, Y. M. (2019). Multi-objective based spatio- temporal feature representation learning robust to expression intensity variations for facial expression recognition. IEEE Transactions on Affective Computing, 10(2), 223–236. https://doi.org/10.1109/TAFFC.2017.2695999

[8] Kim, G., & Lee, C. (2016). Convolutional Neural Network Using Convolutional NeuralNetwork.Springer, 2644(2), 747–749. https://link.springer.com/chapter/10.1007/978-14842- 2845-6_6

[9] Li, Y., Zeng, J., Shan, S., & Chen, X. (2019). Occlusion Aware Facial Expression Recognition Using CNN With Attention Mechanism. IEEE Transactions on Image Processing, 28(5), 2439– 2450. https://doi.org/10.1109/TIP.2018.2886767

[10] Liang, D., Liang, H., Yu, Z., & Zhang, Y. (2020). Deep convolutional BiLSTM fusion network for facial expression recognition. Visual Computer, 36(3), 499–508. https://doi.org/10.1007/s00371-019-01636-3

[11] Lopes, A. T., de Aguiar, E., De Souza, A. F., & Oliveira-Santos, T. (2017). Facial expression recognition with Convolutional Neural Networks: Coping with few data and the training sample order. Pattern Recognition, 61, 610–628. https://doi.org/10.1016/j.patcog.2016.07.026

[12] Mellouk, W., & Handouzi, W. (2020). Facial emotion recognition using deep learning: Review and insights. Procedia Computer Science, 175, 689–694 .2020.07.101

[13] Mollahosseini, A., Chan, D., & Mahoor, M. H. (2016). Going deeper in facial expression recognition using deep neural networks. 2016 IEEE Winter Conference on Applications of Computer Vision, WACV 2016. https://doi.org/10.1109/WACV.2016.7477450

[14] Ng, H. W., Nguyen, V. D., Vonikakis, V., & Winkler, S. (2015). Deep learning for emotion recognition on small datasets using transfer learning. ICMI 2015 - Proceedings of the 2015 ACM International Conference on Multimodal,443–449. https://doi.org/10.1145/2818346.2830593

[15] Singh, A., Srivastav, A. P., Choudhary, P., & Raj, S. (2021). Facial emotion recognition using convolutional neural network. Proceedings of 2021 2nd International Conference on Intelligent Engineering and Management, ICIEM 2021, 486–490. https://doi.org/10.1109/ICIEM51511.2021.9445346

[16] Taniguchi, T., Furusawa, K., Liu, H., Tanaka, Y., Takenaka, K., & Bando, T. (2016). Determining Utterance Timing of a Driving Agent with Double Articulation Analyzer.IEEE Transactionson Intelligent Transportation Systems, 17(3), 810–821. htttps,ps.link https://doi.org/10.1109/TITS.2015.2484421

[17] Vimal, K. U., Sandij, S. K., Yogesh, M., & Soundarya, S. (2021). Retraction: Facial Emotion Recognition Using Deep Learning. Journal of Physics: Conference Series, 1916(1), 17–21. https://doi.org/10.1088/1742-6596/1916/1/012118

[18] Yolcu, G., Oztel, I., Kazan, S., Oz, C., Palaniappan, K., Lever, T. E., & Bunyak, F. (2019). Facial expression recognition for monitoring neurological disorders based on convolutional neural network. Multimedia Tools and Applications, 78(22), 31581–31603. s11042-019- https://doi.org/10.1007/s11042-019-07959-6.

[19] Yu, Z., Liu, G., Liu, Q., & Deng, J. (2018). Spatio-temporal convolutional features with nested LSTM for facial expression recognition. Neurocomputing, 317, 50–57. https://doi.org/10.1016/j.neucom.2018.07.028

[20] Zhao, X., Shi, X., & Zhang, S. (2015). Facial expression recognition via deep learning. IETE Technical Review (Institution of Electronics and Telecommunication Engineers, India), 32(5), 347–355. https://doi.org/10.1080/02564602.2015.1017542