FACIAL EMOTION RECOGNITION USING CNN

A DISSERTATION

SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS

FOR THE AWARD OF THE DEGREE

OF

MASTER OF TECHNOLOGY

IN

DATA SCIENCE

Submitted by:

Amulya Kaustubh

2K21/DSC/02

Under the supervision of

Dr. Abhilasha Sharma

DEPARTMENT OF SOFTWARE ENGINEERING

DELHI TECHNOLOGICAL UNIVERSITY

(Formerly Delhi College of Engineering)

Bawana Road, Delhi-110042

MAY, 2023

DEPARTMENT OF SOFTWARE ENGINEERING

DELHI TECHNOLOGICAL UNIVERSITY

(Formerly Delhi College of Engineering)

Bawana Road, Delhi-110042

DECLARATION

I, Amulya Kaustubh, 2K21/DSC/02 student of M.Tech (DSC), hereby declare that the project entitled "***Facial Emotion Recognition Using CNN***" which is submitted by me to Department of Software Engineering, Delhi Technological University, Shahbad Daulatpur, Delhi in partial fulfilment of requirement for the award of the degree of Master of Technology in Data Science, has not been previously formed the basis for any fulfilment of requirement in any degree or other similar title or recognition. This report is an authentic record of my work carried out during my degree under the guidance of Dr. Abhilasha Sharma.

Amulya

Place: Delhi

Date: 31st May, 2023

Amulya Kaustubh

(2K21/DSC/02)

DEPARTMENT OF SOFTWARE ENGINEERING

DELHI TECHNOLOGICAL UNIVERSITY

(Formerly Delhi College of Engineering)

Bawana Road, Delhi-110042

CERTIFICATE

I hereby certify that the project entitled "***Facial Emotion Recognition Using CNN***" which is submitted by Amulya Kaustubh (2K21/DSC/02) to the Department of Software Engineering, Delhi Technological University, Shahbad Daulatpur, Delhi in partial fulfillment of the requirement for the award of the degree of Master of Technology in Data Science is a record of the project work carried out by the student under my supervision.

Abhilasha Sharma

Place: Delhi

Date: 31st May, 2023

Dr. Abhilasha Sharma

SUPERVISOR

Assistant Professor

Department of Software Engineering

ABSTRACT

Facial emotion recognition is a fundamental task in the field of computer vision and human-computer interaction that aims to automatically detect and classify emotions expressed by individuals based on their facial expressions. It is an expanding field that finds application in various domains including mental health monitoring, marketing, social robotics, sentiment analysis, education, security, and gaming. In human-computer communication, nonverbal interaction methods such as facial expressions, eye ball movements, and body gestures are utilized, with facial emotion being particularly prevalent as it effectively communicates people's emotions and feelings. However, recognizing facial expressions poses challenges for machine learning methods due to the significant variations exhibited by individuals in expressing their emotions. Factors like differences in brightness, backdrop, position, and subject characteristics such as shape and ethnicity in the facial images contribute to the complexity of facial emotion recognition. As a result, it remains a difficult topic in the field of deep learning and computer vision. In this project, a simple approach for it is introduced that combines a CNN with certain image preprocessing procedures. The proposed model comprises of four convolutional layers, followed by max pooling. The FER2013 dataset has been used for training the network. The network uses a single-component architecture to detect and classify facial photos into one of the seven fundamental human facial expressions. The model was trained for a total of 50 epochs, achieving a training and validation accuracy of 86.13% and 62.39%, respectively. The corresponding training and validation losses are measured at 0.38 and 1.19, respectively

ACKNOWLEDGMENT

I am very thankful to **Dr. Abhilasha Sharma** (Assistant Professor, Department of Software Engineering) and all the faculty members of the Department of Software Engineering at DTU. They all provided us with immense support and guidance for the project. I would also like to express my gratitude to the University for providing us with the laboratories, infrastructure, testing facilities, and environment which allowed us to work without any obstructions. I would also like to appreciate the support provided to us by our lab assistants, seniors and our peer group who aided us with all the knowledge they had regarding various topics.

Amulya

AMULYA KAUSTUBH

(2K21/DSC/02)

TABLE OF CONTENTS

CONTENT **PAGE NUMBER**

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVATIONS

Abbreviation Name	Term Used
FER	Facial Emotion Recognition
CNN	Convolutional Neural Network
LSTM	Long-Short Term Memory
ReLU	Rectified Linear Activation Unit
FER2013	Facial Emotion Recognition 2013
FERA	Facial Expression Recognition and Analysis
SFEW	Static Facial Expressions in the Wild
CK+	Cohn Kanade +
DL	Deep Learning
SGD	Stochastic-Gradient Descent
DNN	Deep Neural Network
VGG	Visual Geometry Group
ResNet	Residual Network
SBNCNN	Sparse Batch Normalization CNN
OCA	Occlusion-aware Channel Attention
GFE	Global Feature Extractors
EDNN	Extended Deep Neural Network
DT	Decision Trees
SVM	Support Vector Machines
BiLSTM	Bidirectional Long Short-Term Memory
DISFA	Denver Intensity of Spontaneous Facial Action
JAFFE	Japanese Female Facial Expression
RAF-DB	Real-world Affective Faces Database
RGB	Red, Green and Blue
STCNN	Spatio-Temporal Convolutional Neural Network
NN	Neural Network

CHAPTER 1

INTRODUCTION

1.1 OVERVIEW

Facial Emotion Recognition (FER) has evolved to a significant area of study in computer vision and deep learning. Deep Learning (DL) models are trained to precisely detect and categorize facial expressions as human emotions. Due of its numerous applications in fields including human-computer interaction, psychology, and marketing, this technology has drawn a lot of interest. It begins with data collection, which involves the capturing of facial photos. These photos are used as the input for the CNN model, which has several layers and is made to extract significant information and forecast emotions. The convolutional layer is a CNN's initial important part. To extract local information, convolutional filters are applied to the input images, convolute them over various spatial locations. Important facial expression-related patterns and features, including wrinkles around the eyes or the shape of the mouth, are captured by these filters. Convolutional algorithms are intended to be translation-invariant, which allows them to recognize important characteristics regardless of their location in the image. The output of the convolutional and pooling layers is subsequently passed through several connected layers. These layers combine the previously learnt elements and record their deeper relationships. The model has the ability to learn intricate combinations of features as each neuron in the fully connected layer receives input from every neuron in the layer preceding it. Finally, the output layer assigns probabilities to each potential emotion class via softmax activation. The feeling that is expected to occur is the one with the highest likelihood. A labelled dataset is necessary to train a CNN model for facial emotion identification. This dataset includes facial images that have been annotated with the various emotion categories, such as neutral, happy, sad, and furious. The model is taught using a supervised learning strategy, where it learns to reduce the discrepancy between the ground truth labels in the training data and the predicted emotion labels. During training, the model’s parameters are adjusted through an optimization method, such as SGD, which updates the parameters based on the computed loss. The difference between the predicted emotion and the actual emotion label is measured by the loss function. The model's parameters are modified iteratively until the CNN becomes proficient at accurately predicting emotions. Utilizing CNNs for facial expression recognition has several benefits, including their capacity to automatically identify pertinent features from the input data. Traditional methods needed manual feature engineering, which took time and had a limited capacity to record intricate patterns and subtle variations in facial expressions. CNNs, on the other hand, have the ability to automatically learn hierarchical representations of the input data, which enables them to recognize minute differences and nuances in facial expressions. Facial emotion recognition using CNNs has revolutionized the industry by offering precise and automatic solutions for emotion categorization based on facial expressions. CNN-based facial expression identification is anticipated to advance and find use in a variety of fields due to continuous developments in deep learning.

1.2 MOTIVATION

FER using Convolutional Neural Network (CNN) is motivated by the aspiration to create cutting-edge technology that can precisely and automatically interpret human emotions from facial expressions. Human communication is greatly influenced by emotions, which also have an impact on our interactions, judgements, and general well-being. There are many real-world uses for being able to accurately identify and comprehend emotions from facial expressions in many different fields.

Human-computer interaction is one of the primary motivators behind CNN-based facial expression identification. By equipping robots with the capability to perceive and react to human emotions, we can create user experiences that are more personalized and intuitive. FER can improve immersion and customize the gaming experience based on the user's emotional state in virtual reality and gaming applications. Similar to this, robots can be taught to recognize and react properly to the emotions of their human counterparts, enhancing natural and empathic interactions. Applications for facial emotion recognition can be found in industries like marketing and advertising. Understanding consumer emotions can help marketers better understand their preferences, level of engagement, and response to commercials. Businesses may improve their marketing strategy, craft advertising to evoke particular emotional responses, and gauge customer satisfaction using real-time face expression analysis. Recognition of facial emotions can help in the diagnosis and treatment of a variety of psychiatric and neurological diseases in the medical sector.

Due to the strong correlation between emotions and mental health, accurate recognition of facial expressions can aid in the early identification and continuous monitoring of conditions such as anxiety, autism and depression. Healthcare workers can collect factual information and gain a better grasp of patients' emotional states by employing CNNs to analyze facial expressions. Facial expression recognition holds promising prospects for its application in security and surveillance systems. It can be useful to spot suspicious or possibly hazardous people in public places if facial expressions can be automatically detected and analyzed in real-time. It can increase the efficiency of surveillance systems, enhancing security and safety for everyone. The recognition of facial expressions of emotion can prove beneficial in educational environments as well. This technology enables teachers to monitor their students' emotional, intellectual, and engagement levels throughout courses, enabling more individualized and flexible teaching methods. In order to ensure that students have the best possible learning experience, virtual learning platforms can also use facial emotion detection to provide feedback and support.

1.3 PROBLEM STATEMENT

The problem at hand in utilizing CNN for FER involves developing a dependable and precise system capable of automatically detecting and classifying human emotions by analyzing facial expressions. The need for cutting-edge technologies that can comprehend and interpret human emotions, which are essential to communication, social interactions, and wellbeing, is what drives this issue.

The intricate and ever-changing nature of facial emotions presents a significant challenge in the identification of facial expressions. Facial muscle movements, small changes in facial features, and variations in intensity and length are all used to communicate emotions. A complex system that can adapt to changes in lighting, head attitude, facial occlusions, and individual differences is needed to capture and analyze these subtleties. By utilizing their capacity to recognize spatial correlations in facial images and learn hierarchical representations, CNNs present a possible approach. The availability of labelled training data presents another difficulty. A sizable and varied dataset of labelled facial photos with accompanying emotion labels is necessary for facial emotion recognition models. Such datasets can be time- and money-consuming to gather and annotate. Additionally, it is essential to guarantee the dataset's diversity and representativeness in order to build a solid model that generalizes well to various people, demographics, and cultural backgrounds. The challenge of utilizing CNNs to recognize facial expressions also involves tuning the architecture and parameters of the CNN model. Designing an optimal network architecture requires determining the number and configuration of convolutional layers, pooling layers, fully connected layers, and activation functions. For best performance and to prevent overfitting, hyperparameter optimization, including learning rate, batch size, and regularization methods, is crucial.

1.4 PROPOSED SOLUTION

The proposed solution for FER endeavors to develop a dependable and accurate system that can automatically detect and classify human emotions by analyzing facial expressions. To extract relevant features from facial images, capture spatial correlations, and learn hierarchical representations, this system utilizes CNN. The acquisition of a diversified and labelled dataset of facial picture datasets with matching emotion labels is the initial stage in the suggested solution. The CNN model will be trained and tested using this FER2013 dataset. Next, the proposed solution involves creating a powerful CNN architecture that is optimized for face expression recognition. Convolutional, pooling, and fully connected layers make up the standard design. By applying filters to the input photos, the convolutional layers capture local features and extract pertinent data about facial emotions. The pooling layers down sample the feature maps, preserving essential features while reducing the spatial dimensions. The classifier uses the fully connected layers to link the retrieved characteristics to various emotion groups. Hyperparameter adjustment is done to enhance the CNN model's performance. This entails picking the right activation functions, batch size, learning rate, and regularization strategies. In order to improve the model's generalization ability, techniques such as dropout have been implemented.

1.5 ORGANIZATION OF DISSERTATION

The necessity and significance of this project are addressed in Chapter 1. It presents the problem statement and its proposed solution. Chapter 2 provides a thorough overview of the existing literature relevant to the project's topic. It summarizes and synthesizes the key findings from previous studies. In Chapter 3, the research methodology employed in this project is discussed, encompassing the research design, data collection methods, and various approaches utilized to address the problem

statement. Chapter 4 presents the implementation results of the proposed model, including the accuracy and loss function metrics. It also examines the behavior of the model when exposed to new instances of human facial data.
In Chapter 5, a summary of the key findings and their implications is presented. Additionally, this chapter includes recommendations for future research directions and practical applications based on the obtained results. Finally, the reference section comprises a comprehensive list of all the sources that were consulted and utilized as references throughout this project.

CHAPTER 2

RELATED WORKS

2.1 PREVIOUS RESEARCH

The paper "Going Deeper in Facial Expression Recognition using Deep Neural Networks"[1] presents a Deep Neural Network (DNN) architecture for FER. The CNN that forms the foundation of the suggested architecture has three primary components: a feature extractor, a classifier, and a regressor. The feature extractor pulls out hierarchical information from facial photos using several stages of convolution and pooling layers. Fully-connected layers, the classifier and regressor, predict the valence-arousal values and the facial expression class, respectively. The authors implemented a number of changes to the CNN to enhance its performance, including batch normalization, dropout regularization, and the use of activation functions such as Rectified Linear Activation Unit (ReLU). To expand the training set and to avoid overfitting, they also used data augmentation procedures. In order to determine how various elements of the suggested architecture contribute to overall performance, the authors also carried out ablation research. The findings demonstrated the significance of each component in achieving high performance, and they also demonstrated that the employment of batch normalization and dropout regularization was very efficient in lowering overfitting and enhancing generalization.

In the paper "Facial Expression Recognition with Convolutional Neural Networks"[2], the authors propose a CNN based approach for FER. Five convolutional layers and two fully-connected layers make up the suggested CNN architecture. The input image is first subjected to a 5x5 filter in the first layer, which is followed by a max-pooling operation with a 2x2 stride. A filter of size 5x5 with a stride of 1x1 is likewise applied by the following two convolutional layers. The final convolutional layer has a filter size of 1x1, a stride of 1x1, and a filter size of 3x3 for the fourth convolutional layer. The final convolutional layer's output is flattened before being fed into fully-connected layers, which produce the probability distribution for each of the six facial expressions. The authors additionally undertake feature normalization and data augmentation to further enhance the performance of their model. Techniques for enhancing data include rotation, flipping the horizontal axis, and random cropping. The authors also evaluate various FER techniques, such as manually created feature-based techniques and DL techniques, in comparison to their suggested CNN architecture. The outcomes demonstrate that their suggested CNN performs better than these approaches in terms of accuracy.

In the paper "Facial Emotion Recognition using Deep Convolutional Networks"[3], a DL approach is proposed for FER using a CNN. Three convolution layers, each followed by a max-pooling layer, and two fully-linked layers make up the suggested CNN architecture. First, the input image is preprocessed by being made grayscale and having its intensity values normalized. The feature maps are down sampled by the max-pooling layers to simplify computation, the convolutional layers retrieve features from the input image. The classification task is drawn out by the fully-connected layers by mapping the learned features to the output classes. The authors also investigated data augmentation techniques to further enhance performance, and they discovered that these methods boosted accuracy by up to 3%. Additionally, they evaluated how well the proposed CNN performed in comparison to existing DL techniques like Visual Geometry Group (VGG) and Residual Network (ResNet) and found that their CNN surpassed others w.r.t accuracy and computational efficiency.

The paper "Facial Expression Recognition Method Based on Sparse Batch Normalization"[4] proposes a novel DL model that uses a Sparse Batch Normalization (SBNCNN) to improve the performance of traditional FER methods. The proposed method is based on two main components: sparse convolutional layers and batch normalization layers. First, by utilizing sparse convolutional layers, the SBNCNN architecture aims to minimize the number of parameters in the employed network. In order to choose crucial features while minimizing the amount of non-zero weights in the network, sparse convolution layers were used. When compared to conventional CNN architectures, the SBNCNN has up to 60% fewer parameters. Second, to increase the network's capacity for generalization, the SBNCNN employs batch normalization layers. The input data are normalized by batch normalization layers to have mean equal to zero and variance equal to one. Batch normalizations helps speed up network convergence and lessen overfitting by minimizing internal covariate shift. Traditional batch normalization layers, however, can also make the network less sparse by adding noise. The suggested solution makes use of SBN layers, which only normalize a portion of the batch's features rather than all of them, to address this problem. Additionally, the author performed ablation tests to examine the effects of different components of the suggested strategy. The outcomes demonstrate that the SBNCNN's performance is enhanced by both the SBN and sparse convolutional layers.

The paper "Occlusion aware facial expression recognition using CNN with attention mechanism"[5] proposes a novel DL model for FER in the presence of occlusions. The suggested model effectively handles occlusions while detecting facial expressions by combining the strength of CNN and attention mechanisms. The authors acknowledge right away that occlusions can be quite problematic for facial expression recognition systems since they might obscure crucial facial features. To address this problem, they suggest a CNN model incorporating attention mechanisms. The suggested model consists of three stages: feature extraction, feature attention, and classification. The feature extraction phase extracts feature from the input image using a number of convolution and pooling layers. The feature attention component makes advantage of the attention mechanism to draw attention to and suppress the less significant characteristics. The facial expression categories are mapped to the characteristics in the classification step using a fully linked layer. The authors suggest an innovative attention mechanism termed Occlusion-aware Channel Attention (OCA) to deal with occlusions. OCA learns to ignore the occluding areas of the face and concentrate on the non-occluding areas. This enables the model to still produce reliable predictions when there are occlusions.

In the paper "Facial expression recognition for monitoring neurological disorders based on convolutional neural network"[6], authors present a method for detecting facial expressions based on CNN to aid in the diagnosis of neurological disorders. The suggested approach comprises three steps: feature extraction, classification, and pre-processing. The facial image is aligned and then made grayscale during the pre-processing stage. The CNN is then fed the aligned grayscale image to extract features. The CNN architecture utilized in this study is comprised of three convolution layers, two fully-linked layers, and a max-pooling layer after each. A softmax layer receives the output of the final fully-linked layer and classifies it. The authors also used data augmentation methods to extend the size of the training dataset.

In the paper "Extended Deep Neural Network for Facial Emotion Recognition"[8], authors propose an Extended Deep Neural Network (EDNN) architecture for FER that aims to increase the accuracy of emotion classification by considering both local and global facial features. CNN for feature extraction, Global Feature Extractors (GFE), and softmax classifiers for emotion recognition make up the three primary parts of the EDNN architecture. While the GFE is used to extract global characteristics that capture the general structure of the face, the CNN is used for extracting local information from facial pictures. Five convolutional layers and two fully-linked layers builds the CNN portion of the EDNN. A 32-filter kernel is used in the first convolutional layer, while 64-filter kernels are used in successive layers. After each convolutional layer, the feature maps' spatial dimensionality is decreased using the max-pooling process. The GFE is then fed the CNN's output. The GFE component of the EDNN is a four-layer fully connected neural network that takes the CNN output and extracts global features by taking into account the relationships between the local features. The output of the GFE is combined with the output of the final fully-connected layer of the CNN and is fed into the softmax classifier for emotion detection and recognition.

In the paper "Spatio-Temporal Convolutional Features with Nested Long-Short Term Memory (LSTM) for Facial Expression Recognition"[10], authors propose a Spatio-Temporal Convolutional Features with Nested LSTM model for facial expression recognition. Three modules make up the suggested model: a feature selection module, a nested LSTM module, and a spatiotemporal convolutional feature extraction module. Using a 3D convolutional network, the spatio-temporal convolutional feature extraction module captures the spatial and temporal details of a sequence of face expressions. Using an attention mechanism, the feature selection module chooses the most discriminative characteristics among the retrieved features. A facial expression sequence's temporal dynamics are modelled using nested LSTM modules. Ablation studies were also carried out by the authors to assess the efficiency of various parts of the suggested model. The outcomes demonstrate that each of the three modules contributes to the effectiveness of the suggested paradigm. A face expression sequence's spatial and temporal details can be effectively extracted using the spatio-temporal convolutional feature extraction module. The most discriminative features from the retrieved characteristics can be chosen with good results by the feature selection module.

The paper "Using CNN for facial expression recognition"[7], authors propose a novel approach to recognize facial expressions using CNN. The suggested method aims to overcome the limitations of the traditional machine learning methods used in FER, such as feature extraction and selection, by using DL techniques. Four convolution layers and two fully-linked layers build the suggested CNN model. The model predicts the corresponding expression from the input of a facial image. The performance of the authors' suggested CNN model was compared to more established machine learning models like Support Vector Machines (SVM) and Decision Trees (DT). The results showed that the proposed CNN model outperformed the traditional machine learning models. The authors also performed experiments to examine the impact of various parameters on the effectiveness of the CNN model. They discovered that the accuracy of model was enhanced by reducing the kernel size and increasing the number of convolutional layers. The CNN model's resistance to numerous circumstances, including occlusions and lighting, was also put to the test by the authors. The outcomes demonstrated the CNN model's robustness by demonstrating its ability to cope with various illumination scenarios and occlusions.

In the paper "Multi-Objective based Spatio-Temporal Feature Representation Learning Robust to Expression Intensity Variations for Facial Expression Recognition"[9], authors propose a multi-objective approach for FER that is robust to variations in expression intensity. The authors discuss the significance of FER in a number of disciplines, including psychology, security, and human-computer interaction. They emphasize how paying attention to minute variations in facial expressions can reveal important clues about human emotions and behavior. The difficulties of facial expression identification are covered, including changes in lighting, position, and expression intensity. They provide a multi-objective strategy that takes on these difficulties by concurrently optimizing a number of objectives. With three goals in mind—classification accuracy, robustness to variations in expression intensity, and compactness of the feature representation—they specifically optimize a spatio-temporal convolutional neural network Spatio-Temporal Convolutional Neural Network (STCNN). The findings demonstrate that the suggested approach outperforms existing approaches, reaching a greater level of classification accuracy and robustness to changes in expression intensity. The authors also show how the suggested method learns more condensed feature representations, which is helpful for practical applications.

In the paper "Deep convolutional BiLSTM fusion network for facial expression recognition"[11], authors proposed a deep convolutional Bidirectional Long Short-Term Memory (BiLSTM) fusion network for FER. A CNN, a BiLSTM network, a fusion layer, and a softmax layer make up the proposed network's four components. The BiLSTM network extracts the temporal information from the retrieved characteristics while the CNN extracts the spatial elements of facial images. The CNN and BiLSTM outputs are combined at the fusion layer, and the softmax layer is employed for classification. The authors performed preprocessing on the datasets by resizing the pictures to 64 x 64 pixels and normalizing the pixel values to lie between 0 and 1. The authors further evaluated the proposed network's robustness to expression intensity variations by artificially generating images with different expression intensities. The results showed that the proposed network was more robust to expression intensity variations than other FER methods.

Table I: Summary of the Related Works

Title	Author	Year of Publish	Model	Dataset	Accuracy
“Going Deeper in Facial Expression Recognition using Deep Neural Networks”	Mollahosseini et al. [1]	2016	CNN	MultiPie, MMI, DISFA, FERA, SFEW, CK+, FER2013	94.7%, 77.9%, 55%, 76.7%, 47.7%, 93,2%, 61.1%
“Facial Expression Recognition with Convolutional Neural Networks”	Lopes et al. [2]	2017	CNN	CK+, JAFFE, BU-3DFE	96.76% for CK+
“Facial Emotion Recognition using Deep Convolutional Networks”	Mohammadpour et al. [3]	2017	CNN	CK+	97.01%
“Facial Expression Recognition Method Based	Cai et al. [4]	2018	SBN-CNN	JAFFE, CK+	95.24%, 96.87%

on Sparse Batch Normalization”					
“Occlusion aware facial expression recognition using CNN with attention mechanism”	Li et al. [5]	2018	ACNN	RAF-DB, AffectNet	80.54%, 54.84%
“Facial expression recognition for monitoring neurological disorders based on convolutional neural network”	Yolcu et al. [6]	2018	CNN	RafD	94.44%
“Extended Deep Neural Network for Facial Emotion Recognition”	Deepak jain et al. [8]	2018	CNN	JAFFE, CK+	95.23%, 93.24%
“Spatio-Temporal Convolutional Features with Nested LSTM for Facial Expression Recognition”	Yu et al. [10]	2018	STC-NLSTM	CK+, Oulu-CASIA, MMI, BP4D	99.8%, 93.45%, 84.53%
“Using CNN for facial expression recognition”	Agrawal et Mittal. [7]	2019	CNN	FER2013	65%
“Multi-Objective based Spatio-Temporal Feature Representation Learning Robust to Expression Intensity Variations for Facial Expression Recognition”	Kim et al. [9]	2019	CNN-LSTM	MMI, CASME II	78.61%, 60.98%
“Deep convolutional BiLSTM fusion network for facial expression recognition”	Liang et al. [11]	2020	DCBiLSTM	CK+, Oulu-CASIA, MMI	99.6%, 91.07%, 80.71%

CHAPTER 3

RESEARCH METHODOLOGY

3.1 DATASET

Following are the particulars regarding the dataset utilized for this project:

3.1.1 Dataset Description

- The FER2013 dataset has been used for this project.
- It comprises of approximately 35,887 RGB images of human faces depicting various facial expressions.
- The dataset contains images that can be categorized into seven distinct categories, each assigned a corresponding label as follows:
    - Angry (Label = 0)
    - Disgust (Label = 1)
    - Fear (Label = 2)
    - Happy (Label = 3)
    - Sad (Label = 4)
    - Surprise (Label = 5)
    - Neutral (Label = 6)
- The facial images have undergone automatic registration to ensure that they are relatively centered and occupy a similar amount of space within each image.

Table II: 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**

3.1.2 Dataset Sample

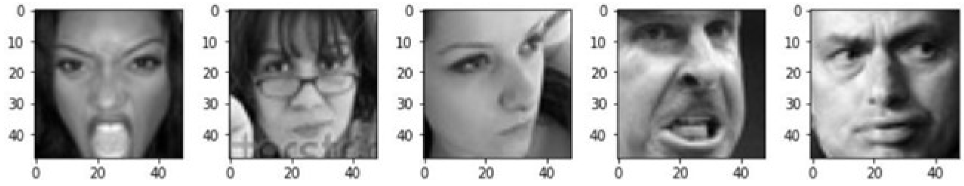


Fig. 3.1: Sample Images for Angry Class

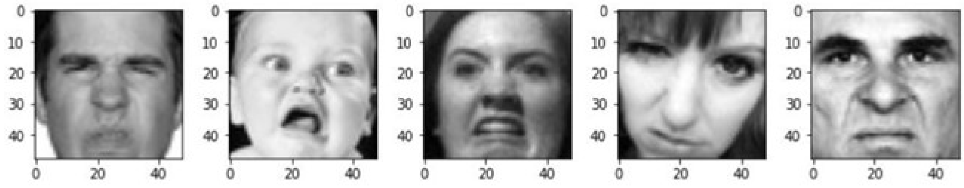


Fig. 3.2: Sample Images for Disgust Class

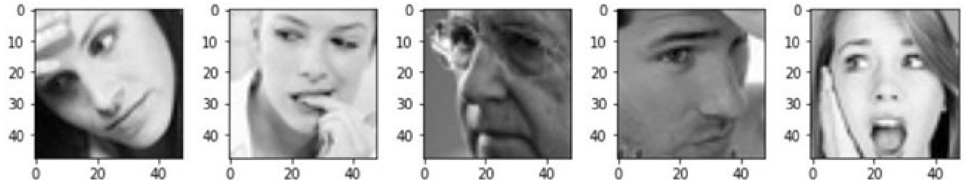


Fig. 3.3: Sample Images for Fear Class

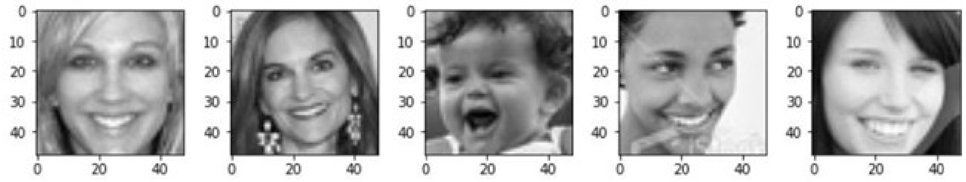


Fig. 3.4: Sample Images for Happy Class

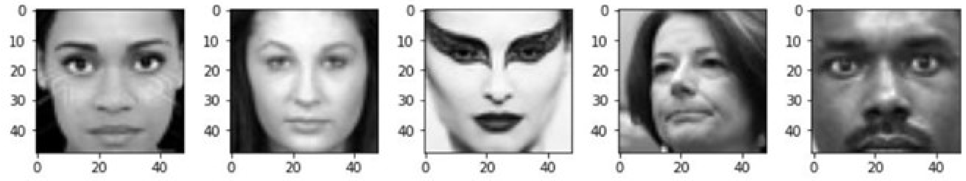


Fig. 3.5: Sample Images for Sad Class

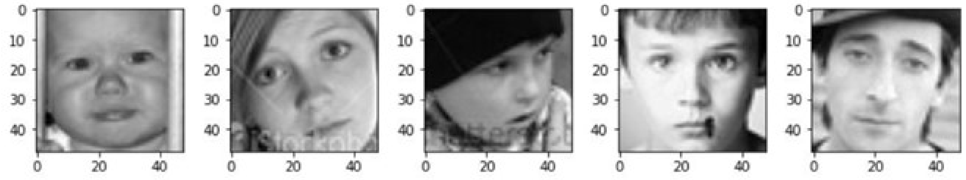


Fig. 3.6: Sample Images for Neutral Class

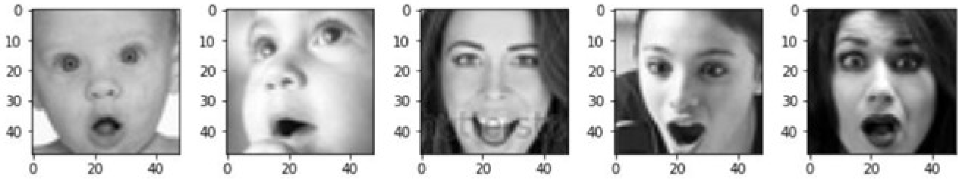


Fig. 3.7: Sample Images for Surprised Class

3.2 DATA PREPROCESSING

The following operations have been applied on the dataset to perform data preprocessing:

3.2.1 Data Cleaning

The dataset includes some incorrect or ambiguous labelled images, which may reduce the effectiveness of face expression detection systems. Before training the model, it is crucial to recognize and eliminate these images.

3.2.2 Image Resizing

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

3.2.3 RGB to Grayscale Conversion

The dataset contains RGB human facial images. The images are converted to grayscale from RGB which decreases the number of image’s channels from three to one. So, it will make the computation faster.

3.2.4 Normalization

The process of normalization is used to uniformly scale the image's pixel values. This enhances the algorithm's precision and lessens the impact of changing lighting. To normalize the dataset's images, the pixel’s value is adjusted to fall in the range of 0 to 1.

3.3 Classification

Classification is the process of labeling an object with a class or category based on its traits or attributes. In the context of machine learning, it is the process of instructing a model to predict the class or category label associated with a given input data point [11].

DL models may automatically develop sophisticated representations of the input data, making them ideal for categorization tasks. The model achieves this by extracting intricate and complex features from the input data using multiple layers of non-linear transformations. In DL, classification entails training a Neural Network (NN) to forecast the class labels of the input data. This is accomplished by mapping the input data to a set of output probabilities, where each probability corresponds to a specific class label [7]. The predicted class label is determined by selecting the one with the highest likelihood. A neuron is the most basic component of a NN. During the process of forward propagation, a neuron multiplies its inputs by their respective weights and combines them through summation. The resulting information is subsequently fed into a nonlinear activation function, which includes a bias term for output adjustment. [7]. The resulting output typically ranges between 0 and 1, making it suitable for probability-related scenarios. Since real-world data frequently demonstrates nonlinearity, the activation function serves the purpose of introducing nonlinear components into the network. By employing nonlinear functions, NN are able to effectively approximate complex functions.

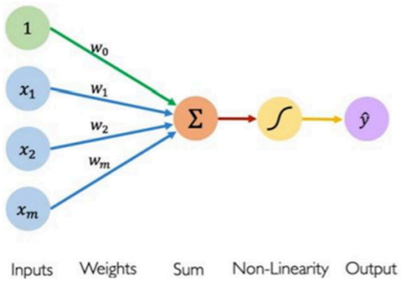


Fig. 3.8: The Basic Structure of a Neuron.

An architecture for a neural network that produces numerous outputs from a single input is known as a multi-output neural network. Multi-output neural networks can be used for tasks that call for the simultaneous creation of numerous predictions or classifications, in contrast to conventional neural networks that only produce a single output [8]. Several approaches exist for creating multi-output neural networks, such as

- Multiple output layers: One way to create a multi-output NN is by using multiple output layers, each of which corresponds to a different prediction or classification task. The network processes the input data, and each output layer produces its own respective output value.

- Single output layer with multiple outputs: Another way to create a multi-output NN is by using a single output layer with many output nodes, each of which represents a separate prediction or classification task. The output nodes are trained jointly to produce accurate predictions for all tasks.

A NN with multiple outputs can be created by integrating multiple neurons. A dense or fully connected multi-output NN implies that every input is connected to every neuron, which has been depicted in Figure 9. In a DNN, numerous hidden layers are organized hierarchically, where each neuron in a hidden layer establishes connections with neurons in the preceding layer. A visual representation of a fully connected DNN with five layers is illustrated in Figure 10.

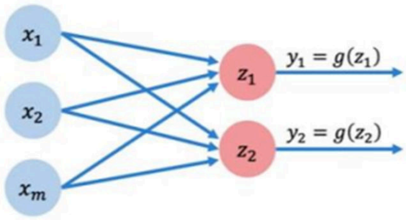


Fig. 3.9: A Neural Network with Multiple Outputs

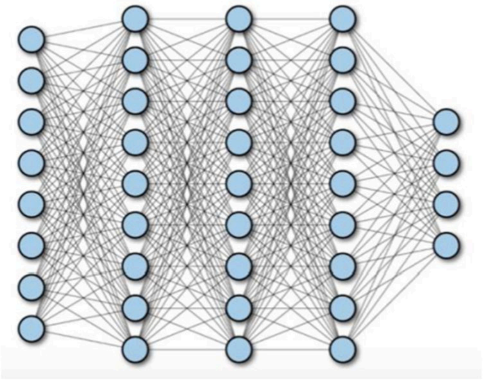


Fig. 3.10: A Fully Connected NN

3.4 WORKING OF CNN

CNNs are a type of deep learning algorithm that is specifically designed for image classification tasks that takes an image as input, assigns importance (learnable weights and biases) to various aspects/objects in the image, and can distinguish between them. CNNs have a number of benefits over other machine learning classification methods, including:

- Automatic feature extraction: Unlike standard machine learning algorithms that need manual feature engineering, CNNs can automatically identify meaningful features from unprocessed pixel data. This can result in significant time and effort savings and improve performance on difficult tasks.
- Hierarchical representation: In order to learn features at various levels of abstraction and capture increasingly complicated correlations in the input data, CNNs use a hierarchical architecture. These more complex traits could be difficult to capture using conventional machine learning algorithms.
- Invariance to translation and rotation: CNNs utilize convolutions to extract features that are invariant to the translation and rotation of the input image. In picture classification tasks where objects may appear in several places or orientations inside the image, this can be advantageous.
- Adaptability to new data: CNNs have the ability to incrementally adjust to incoming data, allowing them to keep learning and developing over time. To include new data, traditional machine learning algorithms may need to be retrained on the entire dataset.

The architecture of CNN is motivated by the arrangement of the visual cortex and bears resemblance to the interconnected network of neurons found in the human brain. In the visual cortex, neurons are organized in a hierarchical manner, with simple neurons in lower layers responding to simple visual patterns, and more complex neurons in higher layers responding to more complex patterns and objects. Similar to these hierarchical networks, CNNs learn to extract increasingly complicated and abstract information from the input image at each layer [3]. The concept of receptive fields, where neurons in the visual cortex respond to patterns within a small localized area of the visual field, serves as a guiding principle for the design of convolutional layers in CNNs. One of the key functions of a CNN is to compress images into a more manageable format without compromising crucial information necessary for accurate prediction [11]. This is vital for constructing an architecture that can effectively learn features and scale well with large datasets. The following are the principal operations employed in CNN:

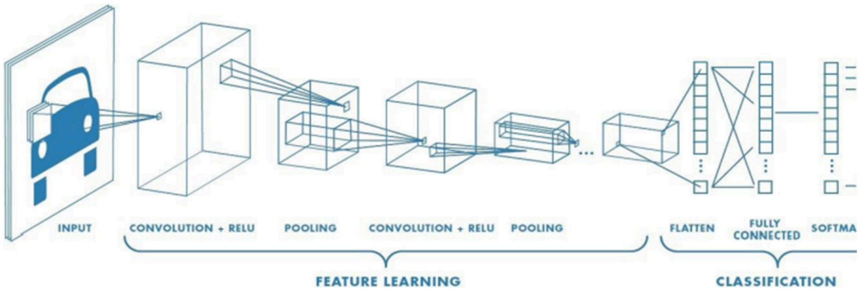


Fig. 3.11: The CNN Operations

3.4.1 Convolution Operation

In CNN, convolution is the core operation that is used to extract features from an input image. Convolution involves calculating the dot product between a small square-shaped filter and a localized portion of the input image, iteratively moving across the image. This procedure creates a feature map that shows whether specific characteristics are present or absent in the input image. The convolution filter, alternatively referred to as a kernel or weight matrix, is responsible for capturing and learning patterns during the training phase [6]. Typically, the kernel size is much smaller than the input image, and it is systematically moved across the entire input image to produce a new feature map. Convolution plays a vital role in CNNs as it enables the network to learn local patterns such as edges, curves, and textures, irrespective of their spatial location within the image [5]. Furthermore, CNNs can learn translational invariance—the ability to recognize patterns wherever they appear in the input image—by distributing the kernel's weights across various areas of the input image.

The convolution operation's goal is to extract high-level features from an input image, such as edges [5]. The convolution layer functions in the following manner:

- The initial convolution layer acquires knowledge of features related to edges, color, gradient orientation, and basic textures.
- Subsequent convolution layers progressively learn more intricate textures and patterns.
- The final convolution layer focuses on capturing objects or specific parts of objects.

The kernel plays a crucial role in executing the convolution operation. It acts as a filter, extracting only the relevant information for the feature map while disregarding irrelevant data. By utilizing a specified stride length, the kernel moves horizontally across the image, processing the entire width. Upon reaching the far-right edge of the image utilizing a designated stride length, the kernel loops back to the left side of the image, maintaining the same stride length, and iterates the process until the entirety of the image has been covered [8].

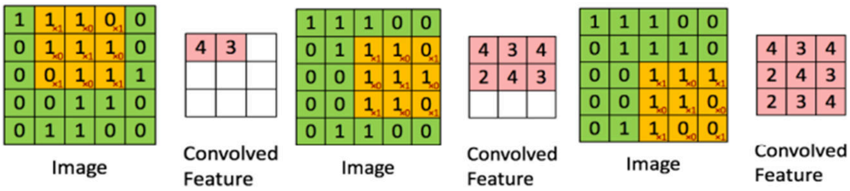


Fig. 3.12: Convolution Operation

To maintain identical dimensions between the kernel and the convolved feature, either same or valid padding techniques are utilized. Same padding is used when the convolved feature matches the dimensions of the input image, whereas valid padding is employed when the feature matches the dimensions of the kernel [7].

3.4.2 Pooling Operation

Pooling is a technique that is frequently used in CNN to down sample the feature maps and lower their dimensionality. Pooling aids in reducing the number of parameters in the network, preventing overfitting, and improving computational efficiency [3]. It is commonly applied after one or more convolutional layers. Max pooling, which selects the maximum value within a small region of the feature map and discards the remaining values, is the most frequently used pooling operation. Pooling provides benefits by enhancing the network's ability to detect features even if the images are slightly shifted or distorted. Pooling aids in the reduction of feature map dimensionality, allowing the network to capture more intricate and abstract information in subsequent layers.

Pooling can be categorized into two types: maximum pooling and average pooling. In maximum pooling, the highest value within the region covered by the kernel is selected, whereas in average pooling, the average value is calculated and returned [8]. Figure 13 illustrates the process of average pooling and max pooling.

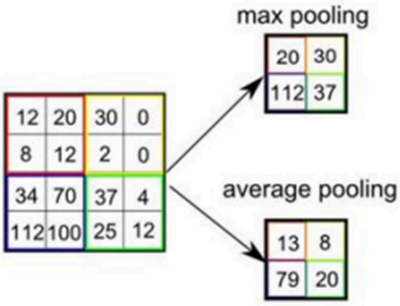


Fig. 3.13: Pooling Operation

3.4.3 Fully Connected Layer

In Convolutional Neural Networks (CNNs), a Fully Connected Layer (also known as a Dense Layer) is a type of layer where each neuron is connected to every neuron in the previous layer. The fully connected layer is commonly positioned at the end of the CNN architecture to transform the extracted features into class scores or probability distributions. In this layer, each neuron receives input from every neuron in the preceding layer, establishing a comprehensive interconnection among the neurons. The neuron performs a calculation by multiplying each input with its corresponding learnable weight and adding a bias term [9]. The weighted inputs are then combined to produce a sum. To introduce non-linearity to the network, this sum is passed through a non-linear activation function, such as ReLU. The output of a fully connected layer is typically a vector of class scores or probabilities, obtained by applying a softmax function to the activations of the last layer. This vector represents the probabilities of the input image belonging to each class in the classification task. Fully connected layers are beneficial as they enable the network to learn intricate, non-linear relationships between the extracted features and the target labels. However, the number of parameters in these layers can become large, potentially leading to overfitting, as every neuron in a fully connected layer is connected to every neuron in the layer below it [6]. Techniques like dropout and regularization are frequently employed to address this issue by preventing the network from memorizing the training data.

3.4.4 Dropout Layer

Dropout is a regularization method employed in CNN to counteract the issue of overfitting. Overfitting occurs when a model becomes overly complex and starts to memorize the training data instead of effectively generalizing to new and unseen data. Dropout is a technique used to mitigate overfitting by introducing randomness into the network during training. With dropout, a fraction of the neurons in a layer is randomly eliminated (turned to zero) during each training iteration. By incorporating dropout in the network, each neuron is encouraged to collaborate with a subset of other neurons in the layer, instead of relying on a small group of highly correlated neurons. This fosters the learning of more robust and adaptable properties within the network. During training, each neuron is randomly deactivated with a specific probability, further enhancing the network's ability to generalize and prevent overreliance on individual neurons. Accordingly, each neuron will have a probability of 1 - p of being present in the forward pass and a probability of p of being absent, where p is the dropout rate. Each training sample receives a unique dropout operation that varies depending on the network layer. As a result, the architecture used during each training iteration varies significantly, and the network must have the ability to adapt to these changes [8]. During the forward pass, the output of each neuron is scaled by 1 / (1 - p) to compensate for the neurons that were dropped out. This makes sure that the mean and variance of the overall input to the following layer are the same as they would be in the absence of any dropout. The dropout is disabled during testing, and all neurons are utilized during the forward pass. To make sure that the overall input to the following layer has the same mean and variance as during training, the outputs of the neurons are scaled down by the dropout rate. Dropout might lengthen the network's training time because each iteration requires a different design [2]. Nevertheless, dropout is a useful tool in the deep learning toolbox because the advantages frequently exceed the disadvantages. Figure 14 depicts the scenario of dropout in the network.

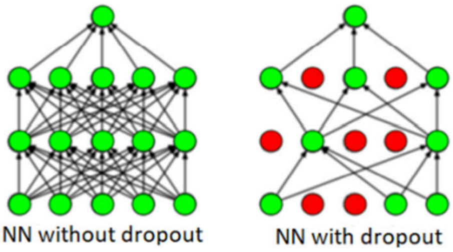


Fig. 3.14: Dropout Operation

3.4.5 Batch Normalization

CNN employ batch normalization as a method to enhance the network's performance and stability during training. Normalization is applied to the activations of the previous layer for each sub-batch of training samples. The fundamental goal of batch normalization is to transform the inputs into a layer so that they have a mean of zero and a variance of one. By doing this, the issue of bursting gradients and vanishing gradients, which can happen when the scale of the input values to a layer is too large or tiny, is less likely to occur [7]. The range of values that the activations can take on is constrained by normalizing the inputs, which aids in keeping them within a manageable range. It is typically applied to the output of a convolutional or fully connected layer before applying the activation function. This ensures that each channel of the output feature map undergoes a separate normalization process. Batch normalization offers several benefits, including increased network accuracy by reducing the impact of internal covariate shift, improved training stability, and faster convergence by reducing the dependence of gradients on the input value scale. The network can learn a different scale and offset for each channel by rescaling and shifting the normalized output using learned parameters. In order to enhance the effectiveness of network training, it is advantageous to maintain consistent distributions of layer inputs. Discrepancies in these distributions can introduce bias to the model. To mitigate this issue, batch normalization is utilized, which normalizes layer’s input [1].

3.4.6 Activation Function

In a CNN, an activation function is a mathematical function that is utilized to process the output of each neuron in a layer. Its purpose is to introduce nonlinearity into the network, enabling it to learn intricate and abstract representations of the input data. CNN uses ReLU as the activation function more frequently, which returns the input if it is positive and 0 otherwise. ReLU outperforms other activation functions in several aspects, such as computational efficiency and the elimination of the vanishing gradient problem. The softmax activation function is frequently used in the output layer of neural networks, particularly in classification tasks where the goal is to assign input data to one of multiple categories. It generates a probability distribution over the possible categories based on a vector of real-valued inputs, ensuring that the sum of the probabilities equals 1 [6]. To obtain the output of the softmax function for each category, the exponential of the input value is divided by the sum of the exponentials of all the input values. The softmax function offers a mechanism to interpret the network's output as probabilities, which may be used to generate a final prediction about which category the input data belongs to. This makes it valuable for classification tasks. Typically, the category with the highest probability is considered as the expected output of the network [11]. One of the main advantages of the softmax function is that it ensures that the output probabilities are always non-negative and sum to 1, which makes them interpretable as probabilities and ensures that they can be used in downstream applications that require probabilistic outputs. The formula for evaluating softmax function is defined in the equation 3.1.

$$\sigma(\vec{z})_i = \frac{e^{z_i}}{\Sigma_{j=1}^{K} e^{z_j}} \qquad (3.1)$$

where zi represents the of each input and K represents the count of inputs.

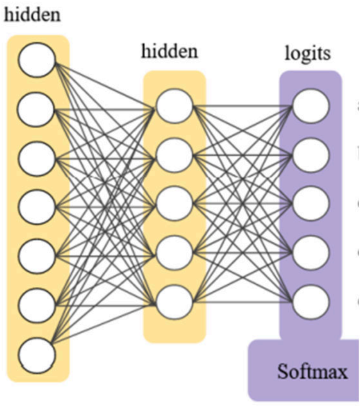


Fig. 3.15: Softmax Function

3.5 PERFORMANCE EVALUATION MEASURES

3.5.1Accuracy

A popular statistic for assessing the effectiveness of a CNN classification model is accuracy. The accuracy of a model measures the its capability to correctly predict the true class or category against an input. It is measured as the proportion of the model's accurate predictions to all of its other predictions [14]. Validation accuracy is a metric used to assess how well a model generalizes to new, unseen data. It is typically computed by reserving a portion of the dataset, called the validation set, during the training process. Following every training epoch, the performance of model undergoes evaluation using the validation set, ultimately determining the validation accuracy. Monitoring the validation accuracy is crucial for tracking the model's performance and detecting overfitting, which refers to a situation where the model performs well on the training data but poorly on new data. If the validation accuracy significantly lags behind the training accuracy, it indicates potential overfitting, and regularization techniques such as weight decay or dropout can be applied to mitigate this issue. The formula for calculating the accuracy has been mentioned in the equation 3.2.

$$\text{Accuracy} = \frac{\text{Number of correct predictions}}{\text{Total number of predictions}} \tag{3.2}$$

3.5.2 Loss

In CNN classification, loss refers to the measure of dissimilarity between the actual class labels of the training examples and the predicted class probabilities. Minimizing loss is the objective of training a CNN classification model, and it can be done using optimization techniques like gradient descent. In CNN classification, the model's performance during training is assessed using both loss and validation loss [14]. During each iteration of the training process, the model's weights and biases are updated using loss. A decrease in loss indicates that the model is better at capturing the patterns and relationships in the training data, as the objective of training is to minimize the loss. Conversely, validation loss refers to the loss computed using a separate validation dataset that is not used for training. It serves as an indicator of how well the model generalizes to new, unseen data. By monitoring the validation loss, we can assess the model's performance and determine if it is overfitting or underfitting the training data. Lower validation loss suggests that the model is more effective at generalizing to new data and is a desirable outcome during the training process. Unlike accuracy, which measures the correctness of predictions, loss is computed as the summation of errors made for each training sample and validation sample. During the training process, loss is commonly used to optimize the model's parameters (e.g., weights in a neural network) and find the best parameter values [12]. Some commonly used loss functions include log loss, mean squared error, and probability loss, each with its own characteristics and suitability for different types of problems. The formula for calculating loss is described in the equation 3.3.

$$Loss = -\sum_{c=1}^{m}(y_{o,c}\log(p_{o,c})) \tag{3.3}$$

where y takes value of either 0 or 1, p denotes the predictive probability, and m indicate the total instances of classes.

3.6 PROPOSED ARCHITECTURE

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

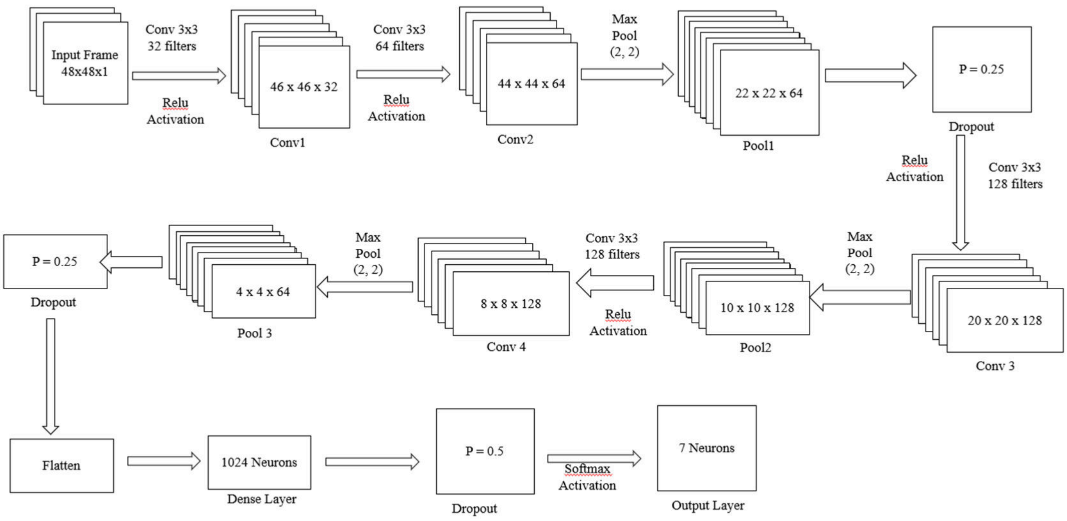
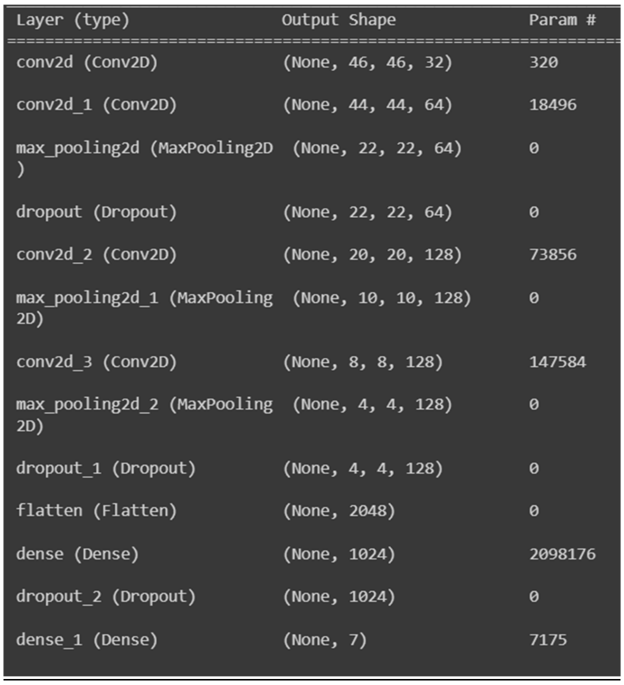


Fig. 3.16: Proposed Architecture



```
Layer (type)                 Output Shape              Param #
=================================================================
conv2d (Conv2D)              (None, 46, 46, 32)        320

conv2d_1 (Conv2D)            (None, 44, 44, 64)        18496

max_pooling2d (MaxPooling2D  (None, 22, 22, 64)        0
)

dropout (Dropout)            (None, 22, 22, 64)        0

conv2d_2 (Conv2D)            (None, 20, 20, 128)       73856

max_pooling2d_1 (MaxPooling  (None, 10, 10, 128)       0
2D)

conv2d_3 (Conv2D)            (None, 8, 8, 128)         147584

max_pooling2d_2 (MaxPooling  (None, 4, 4, 128)         0
2D)

dropout_1 (Dropout)          (None, 4, 4, 128)         0

flatten (Flatten)            (None, 2048)              0

dense (Dense)                (None, 1024)              2098176

dropout_2 (Dropout)          (None, 1024)              0

dense_1 (Dense)              (None, 7)                 7175
```

Fig. 3.17: Convolution Parameters

CHAPTER 4

RESULT

4.1 EVALUATING MODEL’S ACCURACY

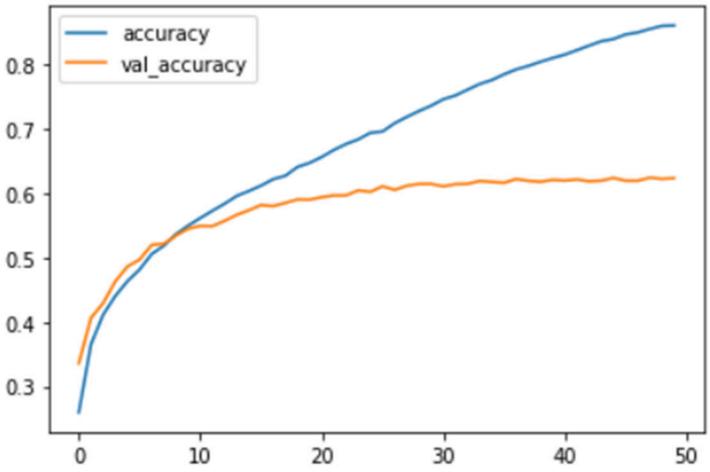


Fig. 4.18: Implemented Model’s Accuracy

The CNN model gave the Training Accuracy and Validation Accuracy equal to 86.13% and 62.39% respectively.

4.2 EVALUATING MODEL’S LOSS

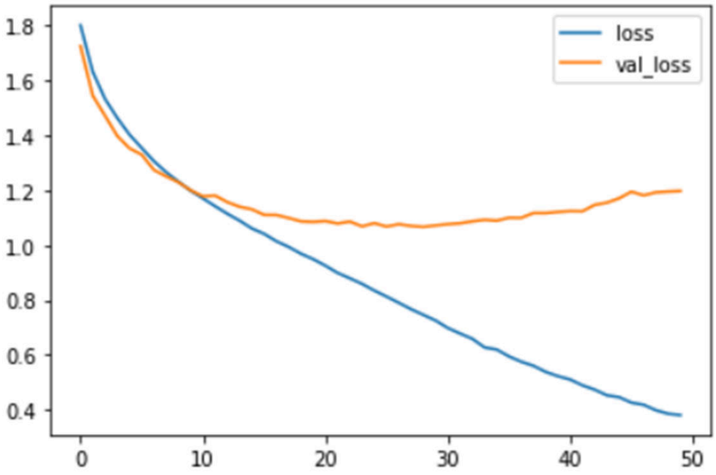


Fig. 4.19: Implemented Model’s Loss

The CNN model gave the Training Loss and Validation Loss equal to 0.38 and 1.19 respectively.

4.3 EVALUATING MODEL'S PERFORMANCE ON HUMAN FACIAL INSTANCES

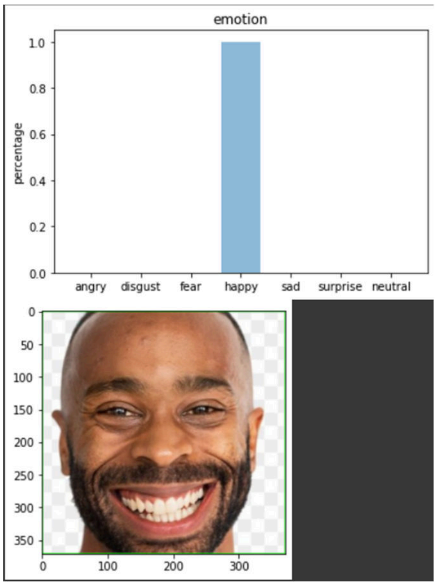


Fig. 4.20: Live Model Testing-1

The trained model is fed with an image of a 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 correct emotion.

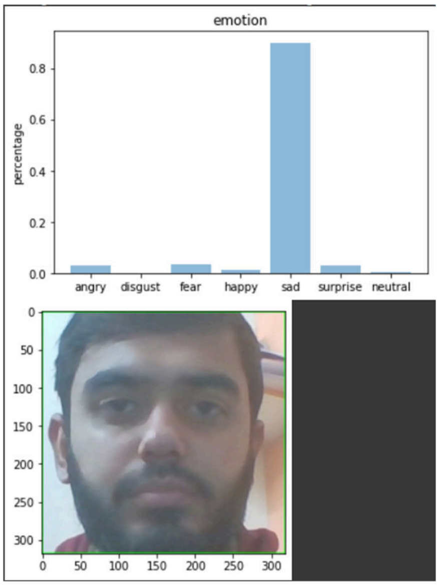


Fig. 4.21: Live Model Testing-2

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

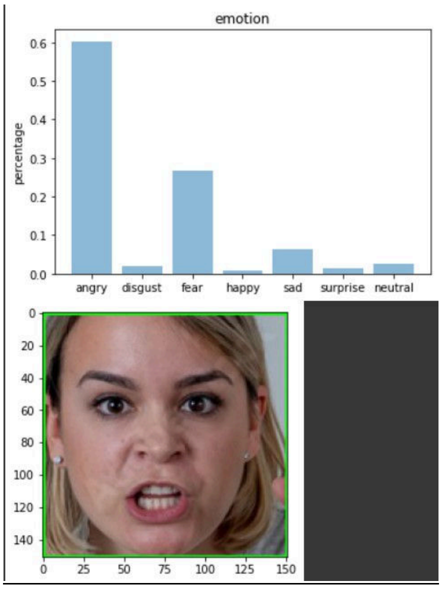


Fig. 4.22: Live Model Testing-3

The trained model is fed with an image of an angry human face. The model gave probabilities of different emotions, but the highest probability was given for the angry emotion. Therefore, the model predicted the correct emotion.

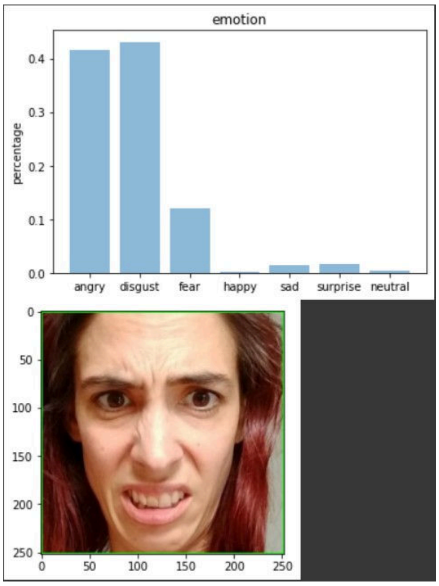


Fig. 4.23: Live Model Testing-4

The trained model is fed with an image of disgusted human face. The model gave probabilities of different emotions, but the highest probability was given for the disgust emotion. Therefore, the model predicted the correct emotion.

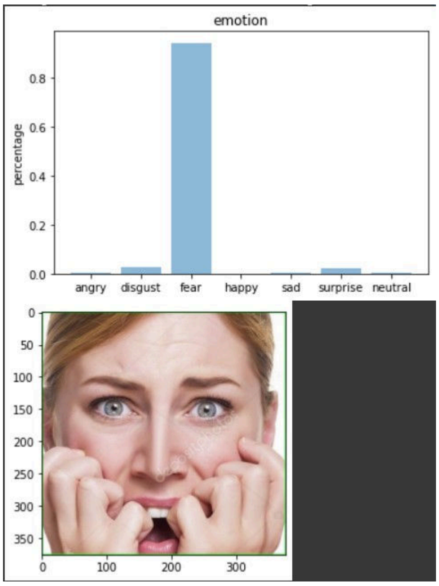


Fig. 4.24: Live Model Testing-5

The trained model is fed with an image of feared human face. The model gave probabilities of different emotions, but the highest probability was given for the fear emotion. Therefore, the model predicted the correct emotion.

CHAPTER 5

CONCLUSION AND FUTURE WORK

5.1 CONCLUSION

FER has emerged as a promising approach for automatic emotion detection and classification from facial expressions. The employment of CNN in the field has demonstrated notable improvements in precision, robustness, and practicality. CNN can be used to fetch useful features from facial images, record spatial correlations, and learn hierarchical representations, all of which are necessary for precise emotion recognition. CNN models can learn detailed patterns and distinctive features that are indicative of various emotional states by utilizing deep learning techniques. The convolution, pooling, and fully-connected layers make up the CNN architecture. Local characteristics, spatial data, and higher-level representations can each be extracted using these layers. The model's capacity for resilience and its generalizability are also enhanced by methods like dropout, batch normalization, and activation functions like ReLU. The purpose of this project was to build a CNN model on the FER2013 dataset for recognition and classification of human facial expressions into one of the seven human emotions that includes happy, sad, surprise, fear, anger, disgust, and neutral emotions. A facial expression identification system was developed that combined established methods such as CNN with image pre-processing procedures. It required a total of 50 epochs to train the proposed model. The training accuracy and validation accuracy obtained were both 86.13 percent and 62.39 percent, respectively. The training losses and validation losses were 0.38 and 1.19, respectively. The results proves that the deep CNNs are capable to learn characteristics of facial expression and increase FER accuracy. The proposed CNN model that was implemented was able to recognize and classify the seven basic facial emotions.

5.2 FUTURE WORK

We can concentrate our efforts in the future on gathering bigger and more varied datasets to train facial emotion identification models. To assure the system's generalization abilities, this includes gathering data from people of various demographics, cultures, and age ranges. The precision of facial emotion identification can be impacted by face occlusions, such as glasses, facial hair, or accessories. It can be investigated to find methods to efficiently handle occlusions and fluctuations in facial appearance, enabling the model to more accurately record and decipher facial expressions even under difficult circumstances. Different modalities, such as body language, voice intonation, and facial expressions are used to convey emotions. In the future, we can investigate the integration of many modalities to boost the reliability and accuracy of emotion identification systems. Combining face cues with auditory or written cues can offer more information, resulting in a more thorough understanding of emotions. By utilizing pre-trained models from comparable tasks or datasets and fine-tuning them for facial emotion identification, transfer learning techniques might be investigated. To improve the model's performance while working with data from various sources or domains, such as various cultures or age groups, domain adaptation methods can also be used.

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., Xue, C., & Wei, C. (2018). Facial Expression Recognition Method Based on Sparse Batch Normalization CNN. *Chinese Control Conference, CCC*, *2018July*, 9608–9613. https://doi.org/10.23919/ChiCC.2018.8483567

[3] Fathallah, A., Abdi, L., & Douik, A. (2018). Facial expression recognition via deep learning. *Proceedings of IEEE/ACS International Conference on Computer Systems and Applications,AICCSA,2017October*,745–750. https://doi.org/10.1109/AICCSA.2017.124

[4] 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/10.1016/j.patrec.2019.01.008

[5] 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

[6] 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

[7] Kim, G., & Lee, C. (2016). Convolutional Neural Network Using Convolutional Neural Network. *Springer*, *2644*(2), 747–749. https://link.springer.com/chapter/10.1007/978-1-4842-2845-6_6

[8] 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

[9] 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

[10] 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

[11] Mellouk, W., & Handouzi, W. (2020). Facial emotion recognition using deep learning: Review and insights. *Procedia Computer Science*, *175*, 689–694. https://doi.org/10.1016/j.procs.2020.07.101

[12] 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

[13] 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 Interaction*, 443–449. https://doi.org/10.1145/2818346.2830593

[14] 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

[15] 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 Transactions on Intelligent Transportation Systems*, *17*(3), 810–821. https://doi.org/10.1109/TITS.2015.2484421

[16] 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

[17] 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. https://doi.org/10.1007/s11042-019-07959-6

[18] 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

[19] 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

[20] L. O. Chua and T. Roska, "The CNN paradigm," *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*, vol. 40, no. 3, pp. 147–156, Mar. 1993, doi: https://doi.org/10.1109/81.222795.

[21] L. O. Chua, "CNN: A Vision of Complexity," *International Journal of Bifurcation and Chaos*, vol. 07, no. 10, pp. 2219–2425, Oct. 1997, doi: https://doi.org/10.1142/s0218127497001618.

[22] N. Raut, "Facial Emotion Recognition Using Machine Learning," *Master's Projects*, Apr. 2018, doi: https://doi.org/10.31979/etd.w5fs-s8wd.

[23] E. Pranav, S. Kamal, C. Satheesh Chandran, and M. H. Supriya, "Facial Emotion Recognition Using Deep Convolutional Neural Network," *2020 6th International Conference on Advanced Computing and Communication Systems (ICACCS)*, Mar. 2020, doi: https://doi.org/10.1109/icaccs48705.2020.9074302.

[24] M. A. H. Akhand, S. Roy, N. Siddique, M. A. S. Kamal, and T. Shimamura, "Facial Emotion Recognition Using Transfer Learning in the Deep CNN," *Electronics*, vol. 10, no. 9, p. 1036, Apr. 2021, doi: https://doi.org/10.3390/electronics10091036.

[25] S. Modi and Mohammed Husain Bohara, "Facial Emotion Recognition using Convolution Neural Network," *International Conference Intelligent Computing and Control Systems*, May 2021, doi: https://doi.org/10.1109/iciccs51141.2021.9432156.

turnitin

Similarity Report ID: oid:27535:36522442

PAPER NAME

thesis_amulya_kaustubh_2k21dsc02.docx

AUTHOR

Amulya Kaustubh

WORD COUNT

9873 Words

CHARACTER COUNT

57319 Characters

PAGE COUNT

45 Pages

FILE SIZE

1.6MB

SUBMISSION DATE

May 30, 2023 8:38 PM GMT+5:30

REPORT DATE

May 30, 2023 8:39 PM GMT+5:30

● 12% Overall Similarity

The combined total of all matches, including overlapping sources, for each database.

- 7% Internet database
- 3% Publications database
- Crossref database
- Crossref Posted Content database
- 10% Submitted Works database

● Excluded from Similarity Report

- Bibliographic material
- Quoted material
- Cited material
- Small Matches (Less then 10 words)