

UNIVERSITY OF HERTFORDSHIRE

School of Physics, Engineering, and Computer Science

MSc Computer Science (Artificial Intelligence and Robotics)

MSc Advanced Computer Science Project Module (7COM1039-0509-2021)

Date: 23rd September 2022

FACIAL EMOTION RECOGNITION USING MACHINE LEARNING

Name: Manas Pramod Deshmukh

Student ID: 20001516

Supervisor: Dr. Julia Goncharenko

MSc Final Project Declaration

This report is submitted in partial fulfilment of the requirement for the degree of Master of Science in Artificial Intelligence and Robotics (Sandwich) at the University of Hertfordshire (UH). It is my own work except were indicated in the report. I did not use human participants in my MSc Project.

I hereby give permission for the report to be made available on the university

# Acknowledgments

I would like to express my special appreciation to my Project Supervisor, **Dr. Julia Goncharenko**, for guiding me throughout this project and has been highly encouraging and motivating in rectifying my mistakes, and making sure that I complete this project on time. It would be impracticable to complete this work without his extensive help.

I would also like to acknowledge my Module Leader, **Prof. Myasar Tabany**, to give the necessary guidance throughout this project.

I would also like to thank all the Module Professors and Tutors who have directly and indirectly assisted me in this project.

Abstract

In order to forecast and identify human emotions, facial emotion recognition technology uses the subject's face as an input and bases its predictions on that information. Worldwide, FER technology is widely employed in real time for a variety of applications. I chose this project for research because, first and foremost, facial recognition technology will fundamentally alter the way we live in the future. Second, I had the notion that I would support society moving forward in this evolving field. In this project, I'll apply machine learning algorithms to a dataset of face expressions, and the desired outcomes will be evident as a result. In order to determine which machine learning algorithm produces better results than the other, I employed two of them in this research to predict human emotions and evaluated their metrics against one another. The FER 2013 dataset, which is freely accessible online, is used as a source for the CNN and SVM algorithms that are being employed in this project. Both of these methods are often employed worldwide in classification-related applications.

Table of Contents: -

|  |  |  |  |
| --- | --- | --- | --- |
| Sr No | Title | | Page No |
| **1** | **CHAPTER 1: Introduction** | |  |
| 1.0 | Introduction........................................................................... | | 9 |
| 1.1 | Research Question and Hypothesis | | 11 |
| 1.2 | Aim and Objective........................................................................... | |  |
|  | 1.2.1 | Aim......................................................................... | 12 |
| 1.2.2 | Objective................................................................ | 12 |
| 1.3 | Software/Tool Requirements................................................ | | 12 |
| 1.4 | Project Plan........................................................................... | | 13 |
| 1.5 | Ethical,Legal,Social and Professional Issues | |  |
|  | 1.4.1 | Ethical Issues.......................................................... | 13 |
| 1.4.2 | Legal Issues............................................................ | 13 |
| 1.4.3 | Professional and Social Issues................................ | 14 |
| **2** | **CHAPTER 2: Background Research** | |  |
| 2.0 | Background Research.......................................................... | | 15 |
| 2.1 | Dataset Collection............................................................... | | 15 |
| 2.2 | Literature Review............................................................... | | 17 |
|  |  | |  |
| **3** | **CHAPTER 3: Classification Methods** | |  |
| 3.1 | Classification Methods.......................................................... | | 24 |
| 3.2 | Support Vector Machine Classification............................... | | 25 |
| 3.3 | Convolution Neural Network Classification........................ | | 27 |
|  |  | |  |
| **4** | **CHAPTER 4: Methodology and Implementation** | |  |
| 4.0 | Methodology...................................................................... | | 30 |
| 4.1 | Introduction......................................................................... | | 30 |
| 4.2 | Design Implementation | | 30 |
|  | 4.2.1 | Libraries used......................................................... | 31 |
| 4.2.2 | Splitting train data,validation data and test data..... | 35 |
| 4.3 | Implementation of CNN algorithm on dataset..................... | | 36 |
| 4.4 | Implementation of SVM algorithm on dataset | | 42 |
| 4.5 | Evaluation Metrices | | 44 |
|  |  | |  |
| **5** | **CHAPTER 5: Results and Discussion** | |  |
| 5.0 | Results and Discussion | | 45 |
| 5.1 | Result of CNN | | 46 |
| 5.2 | Result of SVM | | 47 |
|  | 5.2.1 | SVM with rbf Kernel | 47 |
| 5.2.2 | SVM with linear Kernel | 49 |
| 5.3 | Original Emotion Vs Predicted Emotion Result | | 50 |
| 5.4 | Comparison of SVM and CNN algorithm | | 51 |
| 5.5 | Discussion | | 52 |
|  |  | |  |
| **6** | **CHAPTER 6: Conclusion and Future work** | |  |
| 6.0 | Conclusion | | 54 |
| 6.1 | Future Work | | 54 |
|  |  | |  |
| **7** | **CHAPTER 7: References/Bibliography** | |  |
| 7.0 | References/Bibliography | | 55 |
|  |  | |  |
| **8** | **CHAPTER 8: Appendices** | |  |
| 8.0 | Appendices | |  |
| 8.1 | Appendix 1 | | 57 |
| 8.2 | Appendix 2 | | 59 |
| 8.3 | Appendix 3 | | 60 |
| 8.4 | Appendix 4 | | 61 |

List of Figures

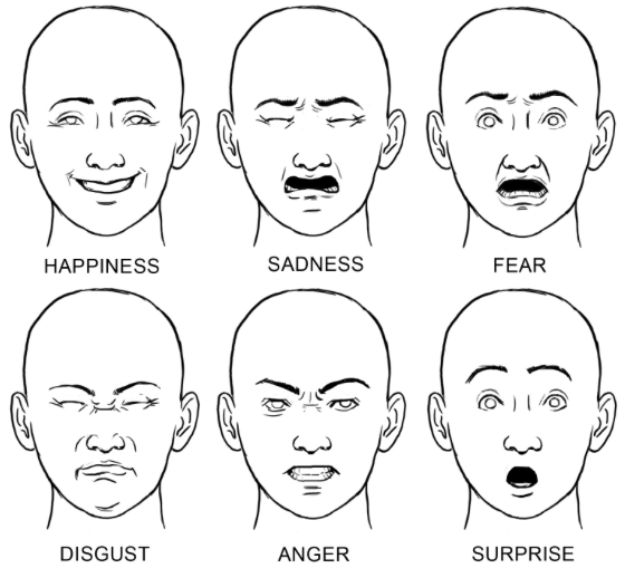
|  |  |  |
| --- | --- | --- |
| **Figure No** | **Figure Name** | **Page No** |
| Figure 1.1 | Universal Basic Types of Emotions | 10 |
| Figure 2.1.1 | Pie chart of train data images with emotions | 16 |
| Figure 2.1.2 | Pie chart of test data images with emotions | 17 |
| Figure 2.2.1 | Scheme in healthcare that introduces facial technology | 18 |
| Figure 3.1.1 | Support Vector Machine Classification | 25 |
| Figure 3.2.1 | Convolution Neural Network(CNN)architecture | 28 |
| Figure 4.2 | Design Implementation of Facial Emotion Recognition | 30 |
| Figure 4.2.1.1 | Libraries imported into the environment: | 32 |
| Figure 4.2.1.2 | Bar plot of Train Data: | 32 |
| Figure 4.2.1.3 | Bar plot of Test Data | 33 |
| Figure 4.2.1.4 | Images from train data with the emotion angry | 33 |
| Figure 4.2.1.5 | Images from train data with the emotion fear and happy | 34 |
| Figure 4.2.1.6 | Images from train data with the emotion sad and disgust | 34 |
| Figure 4.2.1.7 | Images from train data with the emotion neutral and surprise | 35 |
| Figure 4.2.2.1 | Splitting data into train, validation and test data | 35 |
| Figure 4.3.1 | CNN algorithm Implementation | 37 |
| Figure 4.3.2 | CNN Architecture | 38 |
| Figure 4.3.3 | Algorithm of Convolution Neural Network | 40 |
| Figure 4.3.4 | Summary of CNN algorithm model | 41 |
| Figure 4.3.5 | Epochs, training data, and the model fitting stage | 41 |
| Figure 4.3.6 | History of CNN algorithm model | 42 |
| Figure 4.4.1 | Design Implementation of SVM algorithm | 43 |
| Figure 4.4.2 | SVM algorithm with rbf kernel | 44 |
| Figure 4.4.3 | SVM algorithm with linear kernel | 44 |
| Figure 5.1.1 | Loss of training and validation data plotted against 20 epochs | 46 |
| Figure 5.1.2 | Accuracy of training and validation data plotted against 20 epochs | 47 |
| Figure 5.2.1.1 | Scatterplot of Train Data | 48 |
| Figure 5.2.1.2 | Scatterplot of Test data with rbf kernel | 48 |
| Figure 5.2.1.3 | Classification Report of SVM with rbf kernel: | 49 |
| Figure 5.2.2.1 | Scatterplot of Test data with linear Kernel | 49 |
| Figure 5.2.2.2 | Confusion Matrix of SVM with linear kernel | 50 |
| Figure 5.2.2.3 | Classification Report of SVM with linear kernel: | 50 |
| Figure 5.3.1 | Original Emotion Vs Predicted Emotion result | 51 |
| Figure 5.4.1 | SVM model accuracy test with rbf kernel | 51 |
| Figure 5.4.2 | CNN accuracy test | 52 |
| Figure 5.4.3 | SVM model accuracy test with linear kernel | 52 |

**CHAPTER 1**

**1.0 Introduction: -**

A technology called facial emotion recognition is used to analyse feelings from many sources, including images and videos. It is a part of the "affective computing" technology family, a multidisciplinary field of research on how well computers can identify and comprehend affective states and human expressions. Affective computing frequently builds on Artificial Intelligence technology. Facial expressions, a type of non-verbal communication, can be used to infer human emotions. The development of FER technology has been significantly influenced by the wide usage of cameras, as well as current developments in pattern recognition, machine learning, and analysis of biometrics or fingerprints. Many businesses invest in the technology, demonstrating its expanding importance. These businesses range from large tech corporations like NEC or Google to smaller ones like Affectiva or Eyeris. Based on the algorithm, basic emotions (such as happy, disgust, surprise, angry sad, and fear) can be identified from facial expressions. Surveillance cameras, cameras near billboards in businesses, social media, streaming services, and personal devices are some of the sources of the photos and videos used as input by FER algorithms.

A vital tool for describing human emotion is facial expression. Humans experience a wide range of emotions throughout the course of the day, which may be influenced by their mental or physical health. Although people experience a wide range of emotions, six fundamental facial expressions are recognised by modern psychology: Universal emotions include joy, sorrow, surprise, fear, disgust, and anger.



**Figure 1.1.** Universal Basic Types of Facial Emotions:The above figure shows six basic types of emotions that occur recurrently which are Happy, Sadness, Fear, Disgust, Anger and Surprise. The motion and description of these emotions are mentioned below: (Dubey and Singh, 2016)

**1)Anger-**

One of the deadliest emotions is anger. Since this emotion could be detrimental, people want to avoid it. Irritation, annoyance, exasperation, hate, and dislike are related emotions to anger. Anger is shown as upper and lower lids are pulled up, the eye is open, the teeth are closed, lips are tight and pulled down eyebrows.

**2)Fear-**

The feeling of danger is fear. It can be because to the possibility of suffering physical or mental injury. Additional feelings of fear include horror, jitters, panic, concern, and dread. Fear is shown as dropped jaw, down outer eyebrow, up inner eyebrow and open mouth.

**3)Happy-**

The most coveted human emotion is happiness. Cheerfulness, pride, relief, hope, pleasure, and thrill are examples of secondary emotions. Happiness is shown as mouth open, eyes open, raised cheeks, pulled up corner, around eyes wrinkles and edge up mouth.

**4)Sad-**

The opposing feeling of happiness is sadness. Suffering, hurt, despair, pity, and hopelessness are secondary emotions. Sadness is shown as down outer eyebrow, raised inner corner of eyebrows, edge down mouth, eye closed and pulled down lip corner.

**5)Surprise-**

This feeling manifests itself when unanticipated events occur. Astonishment and wonderment are related feelings to surprise. Surprise emotion is shown as eye open, up eyebrows, open mouth and jaw dropped.

**6)Disgust-**

A sense of dislike is disgust. Disgust can be caused by any taste, smell, sound, or texture. Disgust can be shown as depressed lip corner, wrinkle on nose, depressed lower lip and pulled down eyebrows. (Dubey and Singh, 2016)

**1.1 Research Question and Hypothesis: -**

The research questions that I created are

1) Is facial emotion recognition technology utilised to foretell facial emotions in the real world?

**Hypothesis: -**

**-Yes**, facial emotion recognition technology is utilised to foretell facial emotions in the real world.

**-No**, facial emotion recognition technology is not utilised to foretell facial emotions in the real world.

2) Does the SVM algorithm perform better than the CNN algorithm in terms of facial emotion recognition accuracy?

**Hypothesis: -**

**-Yes**, SVM algorithm perform better than CNN algorithm in terms of facial emotion recognition accuracy.

**-No**, SVM algorithm do not perform better than CNN algorithm in terms of facial emotion recognition accuracy.

**1.2 Aim and objectives: -**

**1.2.1 Aim-**

1)Implement specific algorithms to predict accuracy.

2)Compare two algorithms to check which algorithm can perform better another algorithm in terms of accuracy.

3)To predict whether original emotion of person matches predicted emotion or not.

**1.2.2 Objectives: -**

1)To find out what types of emotions does human beings have.

2)To see how an algorithm works and what result will be obtained after applying algorithm.

**1.3 Software/Tools Requirements:**

**1)Jupyter Notebook:**

The jupyter notebook is a notebook used for numerous types of programming and analysis. It works incredibly well at forecasting machine learning characteristics like accuracy, loss, etc. The operations are carried out very effectively and smoothly thanks to the structure, method, and user-friendly interface. This notebook's primary benefit is the ability to analyse and run programmes step-by-step.

**2)Python:**

Machine learning projects are the most common and frequently utilised applications of this language. It is quite simple to comprehend and includes all built-in libraries needed for programmes, depending on necessity.

**1.4 Project Plan: -**

The project plan of my project is mentioned in appendix 3 section.

**1.4 Ethical/Social/Legal/Professional Issues: -**

**1.4.1 Ethical Issues: -**

The ethical issues that need to be considered while performing projects on facial recognition are:

**1)Racial Bias: -**

The problem with this ethical issue is that there can be bias differentiation which will be an issue. All people have different skin tones so sometimes algorithms mismatch skin tones which can give inaccurate results or output that can be main problem.

**2)Privacy of Data: -**

This is a crucial issue that must be considered throughout the project's execution because data breaches may occur. Data that needs to be worked should be publicly approved. University approval is also necessary as without it there can be major issue that can arise which can create a problem. (Olga, 2021)

**3)Mass Surveillance: -**

Mass surveillance is an issue where cameras monitor the life of people. Without   
consent of people, it is not ethical to monitor their day-to-day schedules.

**1.4.2. Legal Issues: -**  
**1) Data Security: -**

Security is an issue when it comes to storing lot of data online. The data that is uploaded on internet should be strongly protected and safe to avoid misuse and improper utilization of data.

**2)Violation of Human Rights: -**

Human rights violation is also a one of the legal issues that arise when using this technology. The main problem is this technology is found discriminatory due to lack of distinguishing colour of people. For example, black people are at maximum risk of misidentification because of this technology which can give inaccurate results.

**3)Criminal Justice: -**

This is one of the legal issues that deals with identification of crime with the help of this technology. It may seem good thing but it also has flaws. For example, if there is fighting going on between good guy and bad guy so sometimes this technology can identify good guy as a suspect instead of bad guy due to identification problem which can be major issue.

**1.4.3. Professional and Social Issues:**

In my project there are no professional and social issues that can be addressed.  
(Academic Integrity and Academic Misco,2021)

**CHAPTER 2**

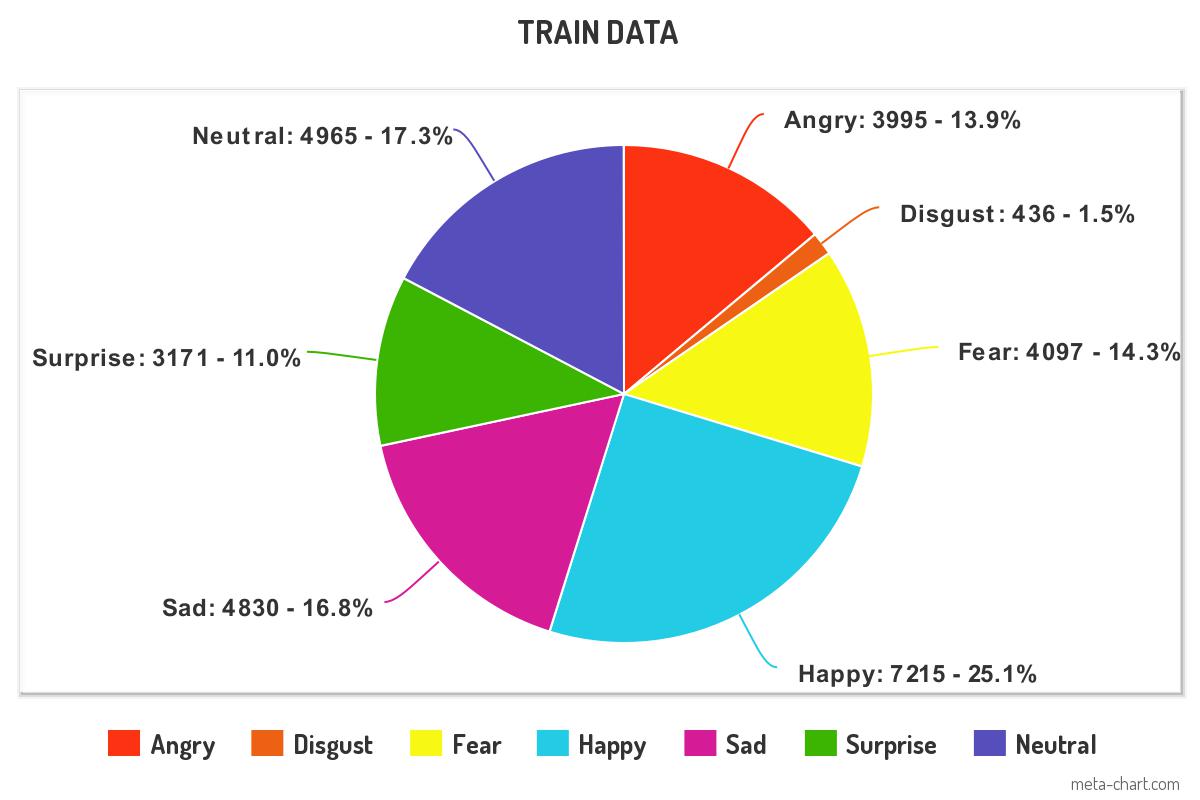
**2.0 Background Research: -**

Human life is tremendously dependent on emotions, and during the past few decades, basic research on emotions has yielded several major findings with significant practical implications. The scientific investigation and comprehension of emotion is credited to "The Expression of the Emotions in Man and Animals" by Charles Darwin and "The Mechanism of Human Facial Expression." by G.G.Duchene de Bologne.. These early studies emphasised the significance of facial expressions in emotional life and promoted the idea that expressions could be viewed as physiologically based reflex responses with adaptive purposes. The Darwinian idea was firmly rooted in the view that emotions work as catalysts for physiological action, that facial emotion and other physiological reactions serve to transmit intentions as well as assist in survival, and that emotions act as a stimulant for physiological action.

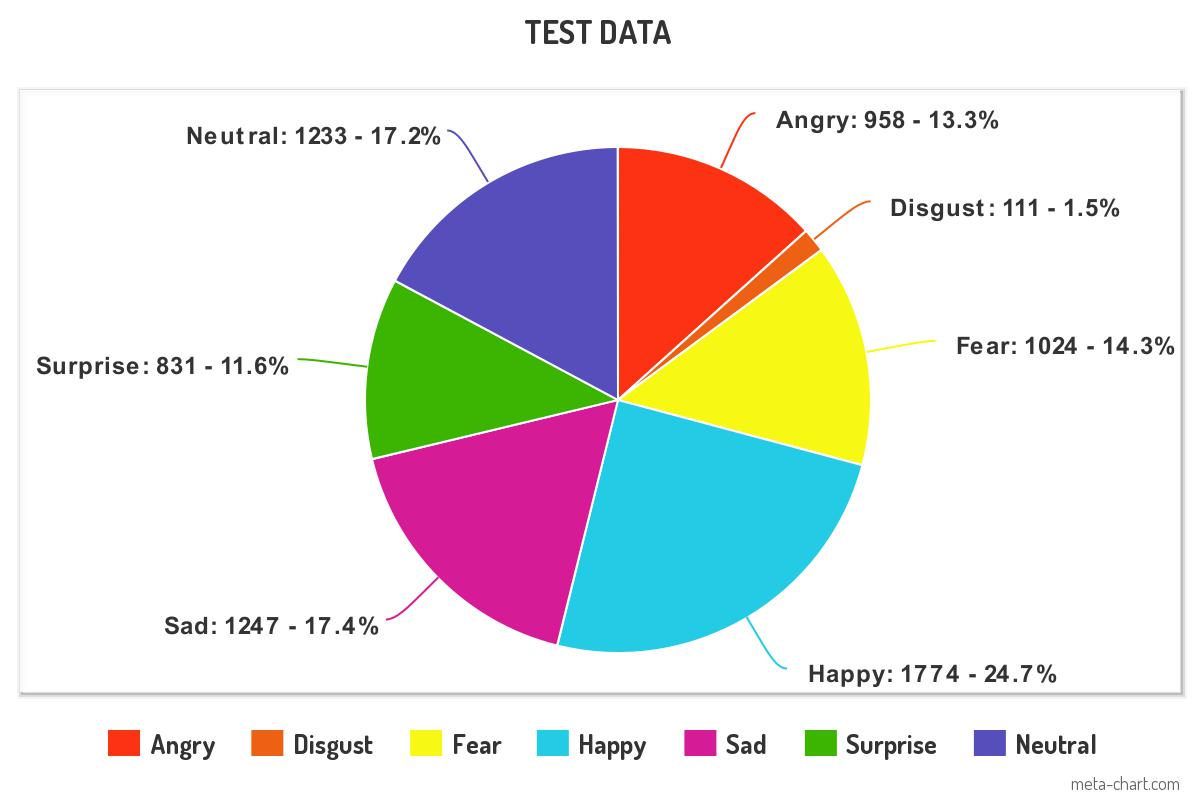
Ekman in 1992 and 1993, who has spent decades studying facial expressions and emotions, has argued that certain basic emotions—such as fear, joy, sorrow, anger, contempt, and surprise—do exist and can be identified by their facial expressions. The capacity to recognise facial expressions often develops along a developmental path, becoming more accurate with practise, exposure to others, and cognitive growth. Infants can recognise emotions by their facial expressions, and most kids can label simple emotions by the time they're 18 months old as said by Bretherton, McNew, & Beeghly-Smith in 1981. Izard & Harris in 1995 found from cross-sectional studies that, children as young as five years old can recognise specific emotions, such as happiness, sadness, and anger, at an adult-like degree. Even while the capacity to distinguish more nuanced expressions like disgust and surprise seems to take longer to develop, most toddlers are able to identify and categorise the basic emotions of happy and angry by the time they are approximately 3 years old. (Sheaffer, Golden and Averett, 2009)

**2.1 Dataset Collection: -**

The dataset that I have used is an FER 2013 dataset which is available on internet and on Kaggle. The FER2013 (Facial Expression Recognition 2013) dataset includes pictures of people and categories that describe their emotions. The 48x48 pixel grayscale images in the dataset represent seven different emotions, including anger, disgust, fear, happiness, sadness, surprise, and neutrality. The dataset consists of 2 sets of examples: a training set with 28709 cases, and a testing set with 7185 case examples. I have combined public test data and private test data together as a test data so that proper analysis can be performed and emotions can be classified or predicted accurately. The dependent variable in my dataset is emotion and the independent variable is images in pixel format.

**Figure 2.1.1.** Pie chart of train data images with emotions:In above figure seven emotions along with size of their respective images are plotted in above pie chart.(Source:Created by me on meta-chart)

It can be seen from above figure that happy type of emotion consists of more data than other emotions. The happy emotion can be seen highlighted in light blue colour showing 25% images. The disgust emotion contains very a smaller number of images. Only 1.5% of images are of this emotion.



**Figure 2.1.2.** Pie chart of test data images with emotions:In above figure it can be seen that seven emotions along with size of their respective images are plotted in above pie chart. (Source: Created by me on meta-chart)

Above figure of pie chart shows that there are seven different types of expressions (disgust,,angry,happy,sad,fear,neutral and surprise) divided in separate sections according to test data. The numbers next to these emotions denote number of images in test data. The percentage next to these number of images tells us the total percentage of images present in this test file as mentioned in above figure no 2.1.1 and figure 2.1.2.

**2.2 Literature Review**

Emotions constitute a psychological condition of the human mind and thinking processes and play a key role in interpersonal communication. Nonverbal information is conveyed through facial expressions and body language, which helps people communicate. A number of face characteristics must be retrieved from the expression of a certain individual in order to perform expression recognition. Classifying facial features into one of the many different categories of emotion is necessary for emotion recognition. Despite the representation of an emotion, the task of detecting the emotion is one of pattern recognition The face recognition approach is utilised in the healthcare industry to help medical professionals identify patients who are in pain or suffering by engaging with them and evaluating their facial expressions.

If a patient is sick or injured, his or her agony will manifest itself through their senses. This method can be helpful in diagnosing and treating mental health issues as well. In addition to analysing depression, this technology aids in the diagnosis of mental diseases. During a patient's stay in a hospital, it is possible to identify their emotions using technology, and this information can be utilised to determine how they are feeling. The findings of this investigation will aid in determining where patients require additional care if they are in pain or depressed. Technology that uses facial expressions is crucial in the diagnosis and evaluation of cognitive impairments. Before performing important procedures, this can also assist in detecting the physical aspects of a patient's face. When a face is burned in an accident, the structure of the face might change, making it difficult for medical professionals to recognise the patients.

Emotion/Expression Recognition

ASD Assessment

Pain Monitoring

Depression Monitoring

**Computer Vision Tasks**

**Healthcare Goal**

**Figure 2.2.1:** Scheme in healthcare that introduces facial technology to predict health problems:In figure face is detected using emotion/expression recognition technology section. Emotion is analysed and according to that various health problems are detected such as ASD assessment, pain monitoring along with depression monitoring. (Khan, 2022)

In this figure the faces of patients are detected using Emotion/Expression Recognition section. Specified computer vision task is explained in with reference to the problems in the healthcare industry. The section of emotion and expression recognition provides an overview of studies that make use of automatic analysis of facial expression in addition to emotion recognition. Additionally, it is also clear from the figure that the computer vision task of emotion/expression detection contributes to the identification of problems such ASD assessment, pain monitoring, and depression monitoring. (Khan, 2022)

Another application of facial emotion recognition technology is in smart devices like for example in smart cars to regulate the driver's emotions using mood lighting and show the identified emotions on an LCD display in order to control the speed of the cars based on the driver's emotions. Although one disadvantage to this technology in this application is that in low light conditions it is difficult to predict human emotions. But it can be developed further as per need. The intelligent automobile will be required to have emotional human-machine systems. Traffic accidents are directly correlated with driver emotion and driving skill. After reaching 1.35 million, the number of fatalities resulting from traffic accidents is still rising. The inability to control one's emotions has been identified as one of the major reasons lowering driving safety among these events. Determining and recognising driver emotions is thus a developing area for intelligent automotive human-machine systems.

One of the main contributing factors to accidents is the inability to control one's emotions while driving the car. One of the emotions that has a detrimental effect on a person's driving is anger. Due to the fact that anger alters perception and inhibits normal thinking and judgement, drivers who are furious tend to feel self-righteous about what happened. A person needs to be better aware of his emotions and have the capacity to manage them in order to drive more safely on the highways. Drivers can sometimes lack awareness, but if they become aware of their emotional states, responding to the situation safely becomes easier. Additionally, if a driver starts to feel sleepy or fatigued, this face emotion recognition system will identify this sensation and warn the driver, preventing an accident.

One of another application that use facial emotion recognition is in education sector. It's critical to understand human emotions in a variety of contexts, including classroom presentations. Teachers can tell whether students are paying attention, whether they appear to understand the course material, whether the lecture pace is too sluggish or too quick by looking at the students' faces. With the advancement of techniques for machine learning and appropriate technology, computer-aided FER became feasible. By utilising this technology, it is possible to learn how student emotions could be used to enhance passive teaching techniques, in which lecturers offer lectures and students listen and take notes. Teachers' actions and students' reactions are part of classroom communication. Analysis of student facial expressions has been the subject of extensive study, but little research has been done on how teachers' facial expressions affect students. (Dukić and Sovic Krzic, 2022)

Recognition of facial expressions has the potential to foretell how a teacher's emotions will affect the classroom. Intelligent evaluation of an instructor's conduct while delivering a lecture may enhance the learning environment while also reducing the time and resources needed for manual evaluation techniques. When a student complies with a directive or an educational goal is achieved, an instructor may feel happy. Disappointment arises when students demonstrate a lack of interest and a refusal to understand an idea. Similar to this, students who lack discipline exhibit anger. Teachers claim that managing these facial expressions typically aids them in accomplishing their objectives. They claim that these expressions frequently result from disciplinary classroom interactions. Using emotion recognition to automatically evaluate instructors could help them become better teachers. The time and resources currently spent filling out dozens of survey forms might be saved by such an assessment process. (Dukić and Sovic Krzic, 2022)

The majority of the aforementioned applications have integrated this facial expression recognition through the use of various classification and regression techniques. In my study, I used both the Convolution Neural Networks SVM and CNN classification methods or approaches, which i shall explain below.

The mathematical process known as convolution is substituted for generic matrix multiplication in at least one of the layers of an advanced type of artificial neural network known as a CNN. Since they were developed primarily to handle pixel data, they are used in image recognition and processing. An input layer, hidden layers, and an output layer make up a convolutional neural network. Hidden layers are any intermediate layers in a feed-forward neural network that have their inputs and outputs hidden by the activation function and final convolution. (Convolutional neural network - Wikipedia, 2022)

Convolutional layers are found in a convolutional neural network's hidden layers. This normally has a layer that performs a dot product of the convolution kernel's input matrix with the layer. ReLU is usually used as the activation mechanism for this product, which is frequently the Frobenius inner product. As the convolution kernel passes over the input matrix for the layer, adding to the input of the subsequent layer, the convolution technique creates a feature map. Additional layers like pooling, normalizing, and fully connected layers are added after this. (Convolutional neural network - Wikipedia, 2022)

**Convolution Layers:**

A tensor with the following dimensions serves as the input for a CNN: (number of inputs) x (input width) x (input height) (input channels). After travelling through a convolutional layer, the image is abstracted to a feature map, sometimes referred to as an activation map, that has the dimensions: (height of feature map) x (number of inputs) x (channels of feature map) x (width of feature map). Convolutional layers integrate the input before sending the combined data to the layer below. This is akin to the response of a cell in the visual cortex to a specific stimulus.

Data is exclusively processed for the receptive field of each convolutional neuron. Fully coupled feedforward neural networks are capable of learning features and data classification., however larger inputs like high-resolution photos typically aren't feasible with this architecture. A relatively large number of neurons would be required in a shallow architecture due to the vast input size of photos, where each pixel is a significant input feature. Additionally, because convolution and/or pooling account for the spatial relationships between various features, convolutional neural networks are ideal for data with a grid-like design (such as photos).

**Pooling layer:**

Convolutional networks may include local and/or global pooling layers in addition to conventional convolutional layers. Pooling layers reduce the dimensionality of data by combining the outputs of neuron clusters place at a single layer into a single neuron at the subsequent layer. Small clusters are combined using local pooling; 2 by 2 tile sizes are frequently employed. Global pooling has an impact on every neuron in the feature map. The maximum and average pooling methods are the two most used. Max pooling uses the highest value of every local cluster of neurons inside the feature map, whereas average pooling uses the average value of every local cluster of neurons.

**Fully connected layer:**

Fully connected layers enable communication in between each neuron in one layer and every other layer. Similar to a conventional multilayer perceptron neural network (MLP). To categorise the photos, the flattened matrix passes through a fully linked layer.

**Receptive field:**

A fixed number of places in the layer before every neuron inside a neural network provide input to that neuron. Only a small portion of the preceding layer, called as the neuron's receptive field, serves as an input source for every neuron in a convolution layers. Typically, the space is square (e.g., 5 by 5 neurons). Of contrast, the receptive field in a completely connected layer encompasses the entire prior layer. Every neuron inside a convolutional layer therefore receives information from a larger portion of the input as compared to earlier levels. The convolution, which takes into account both the value of a single pixel as well as the pixels around it, is what causes this when it is applied repeatedly. While utilising dilated layers, the receptive field's pixel count remains constant, but as the field's dimensions grow when adding the effect of many layers, the field's population gets sparser.

**Weights:**

Every neuron in a neural network calculates an output value by applying a specific function to the input values collected from the receptive field inside the previous layer. The function which is applied to the input data is determined by a bias and a vector of weights (typically real numbers). As learning advances, these biases and weights are changed iteratively. Filters are the weights and biases vectors that correspond to specific input attributes (e.g., a particular shape). The unique property of CNN is that several neurons can use the same filter. A lower memory footprint is obtained because, instead of each receptive field having its own bias and vector weighting, just one bias and one vector of weights are being used across all receptive fields that share that filter. (Convolutional neural network - Wikipedia, 2022)

Support-vector machines (SVM) are machine learning supervised learning models and learning algorithms that analyse data for regression and classification. One of the most effective prediction methods is the SVM, which is based on statistical learning frameworks or the VC theory proposed by Chervonenkis (1974) and Vapnik (1982, 1995). A non-probabilistic binary linear classifier is created using an SVM training method that divides new samples into one of two categories based on a set of training examples. SVM maximises the distance between the two categories by assigning training samples to spatial coordinates. New samples are then forecasted onto the same area and anticipated to fall into a category based on which end of the gap they fall. By implicitly transforming their inputs onto high-dimensional feature spaces, SVMs can successfully do non-linear classification in parallel to linear classification.

The kernel trick is the name of this tactic. When the data are unlabeled, unsupervised learning is required because supervised learning is not an option. By using clustering, this strategy searches for naturally occurring groupings of the data, and then maps additional data to these discovered groups. In the perspective of support-vector machines, a data point is considered to be a p-dimensional vector (p numbers list), and our goal is to determine whether we can partition such values using a (p-1)-dimensional hyperplane. For this, a linear classifier is employed. The data could be categorised by many hyperplanes. One workable choice for the ideal hyperplane is the one that shows the largest margin of difference among the two classes. As a result, we choose the hyperplane with the greatest possible distance on each side from it to the nearest data point. If such a hyperplane exists, it is known as the maximum-margin hyperplane, and the linear classifier it identifies is known as the maximum-margin classifier.

A support-vector machine produces a hyperplane or collection of hyperplanes in a high- or infinite-dimensional space that can be utilized for classification, regression, or other tasks including outliers' detection. The hyperplane with the greatest distance to the closest training data point of any class achieves a respectable separation since, generally speaking, the bigger the margin, the lower the classifier's generalisation error (so-called functional margin). The hyperplanes of the higher-dimensional space are described as a group of points whose dot product with a particular vector is constant, in which a set of vectors is an orthogonal (and hence minimal) set of vectors. (Support-vector machine - Wikipedia, 2022)

CNNs need comparatively little pre-processing compared to other image classification techniques. This implies that the network tends to optimise the filters (or kernels) by autonomous learning, as opposed to conventional techniques where these filters are hand-engineered. An important advantage of this feature extraction is its independence from prior knowledge and human interaction. The classification methods that will help in solving problems will be discussed below.

**CHAPTER 3**

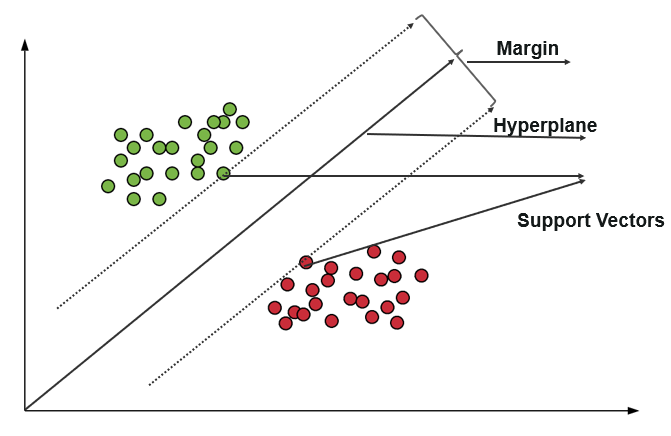
**3.0 Classification Methods: -**

A supervised type of machine learning is a classification method. Support Vector Machine (SVM) and Convolutional Neural Network (CNN), two well-liked classification methods, are contrasted in terms of how well they classify images. Based on feature extraction and feature selection, images are categorised. Linear (SVM) and non-linear (CNN) approaches are the two classifiers being studied. SVM is viewed as a high-end classifier, and CNN has a great feature extraction capability. While SVM complexity cannot be enhanced, CNN complexity may be increased by adding more layers to the feature extraction and selection process. CNN is referred to as filters or features that recognise particular qualities such as vertical edges, horizontal edges, etc. when processing images using weighted matrixes. Each layer of the image's processing enables the filters to distinguish an increasing number of complicated properties. The key challenge in computer vision and machine learning is image classification. Several methods and procedures have been put forth for effective and quick classification.

Neural networks hold more promise for text and image processing than other technologies. SVM is the demonstrated top performer for content-based image retrieval. However, classifying photos and recognising items is a difficult endeavour, particularly for complicated images with numerous elements. The most helpful information for raster images is colour, which is represented as an RGB matrix. Another important component is texture information, particularly for photos in grey scale. A piece of information pertinent to the resolution of a computational problem is referred to as a feature of an image. The technique of performing some alterations on the feature to make it more pertinent for a calculation task is known as feature extraction. Utilizing feature extraction, it is possible to simplify complex images by shrinking their dimensions. Due to the large memory and computing time requirements, processing high dimensional images can occasionally be challenging. The feature extraction process often employs a variety of techniques to obtain a representation of the data, which is subsequently classified using the classifier. For more information, about CNN please refer appendix 1 below.(Jawale and Magar, 2019)

**3.1 Support Vector Machines Classification: -**

A Support Vector Machine (SVM), which is a discriminative classifier, is technically defined as a separating hyper plane. If labelled training data is provided to the SVM classifier (supervised learning), the algorithm generates an ideal hyper plane that classifies fresh data items. Each class is placed on one side of this hyper plane, which divides the plane into two halves in two-dimensional space. It is a method of supervised learning. It is a classification and regression linear model. he points from both classes that seem to be closest to the line are located using the SVM approach. These points are referred to as support vectors. It is then calculated how far apart the line is from the support vectors. The margin refers to this separation. The goal of SVM is to maximise the margin. The best hyperplane is one for which the margin is at its greatest value.



**Figure 3.1.1.** Support Vector Machine Classification: The figure above shows a centre line which is hyperplane. Along with that it creates two margin lines shown by dotted lines having some distance from each other. The green dots and red dots are two class data points also called support vectors that are separated by hyperplane. (Waseem, 2022)

During training, the algorithm examines the input data to find patterns and features in a multidimensional feature space (hyper plane). All of the input data items are shown as points in this space, and they are mapped to the output classes in order to create as large and distinct a separation between the classes as feasible. The SVM algorithm maps new data items into that same space by classifying them into one of the two categories for prediction. Kernel concept can be utilised to find a clear class or hyper plane in randomly distributed data where explicit characterization of the hyper plane is not feasible. SVM employs the term "kernel" to describe a technique for utilising a linear classifier to solve nonlinear classification problems for randomly distributed data points.

kernel can be defined mathematically as

𝑘(𝑥, 𝑦)= 𝑓(𝑥). 𝑓(𝑦) (3,1)

k is a kernel function in this case.

The inputs x and y are n-dimensional.

f is a map from n-dimensional space to m-dimensional space if m>>n.

There are four different types of kernels: sigmoid, radial, linear, and polynomial. (Jawale and Magar, 2019)

1. **Linear kernel:**

the fundamental building block, which is one dimension in nature. When there are lot of features to implement this proves to be best than others. Mostly for text classification problems this type of kernel is used as it is linearly separable.

Formula for linear kernel:

F(x,xj)=sum(x,xj) (3,2)

Here x and xj denote the data that needs to be classified. (Awasti, 2020)

1. **Polynomial kernel:**

It is a more extensive illustration of the linear kernel.. It is less accurate and lvess efficient and it is not preferred as much over other types of kernel functions.

Formula for Polynomial kernel:

F(x,xj)= (x. xj+1) d (3,3)

Here ‘.’ denotes dot product of both values

d denotes the degree

To separate classes F(x,xj) represents decision boundary. (Awasti, 2020)

1. **Radial basis function (RBF):**

In svm, it is the most frequently used kernel function. It typically applies to non-linear data. It aids in correct separation when there is no data knowledge.

Formula for Radial basis function:

F(x,xj) =exp (-gamma \* ||x-xj|| 2)  (3,4)

Gamma varies from 0 to 1. Gamma value is manually provided. Preferred value for gamma is 0.1. (Awasti, 2020)

**4)Sigmoid Kernel:**

Most commonly used Kernel is a sigmoid kernel in an neural network. The kernel function is comparable to a neural network's two-layer perceptron model, which serves as an activation function for neurons.

Formula for Sigmoid Kernel:

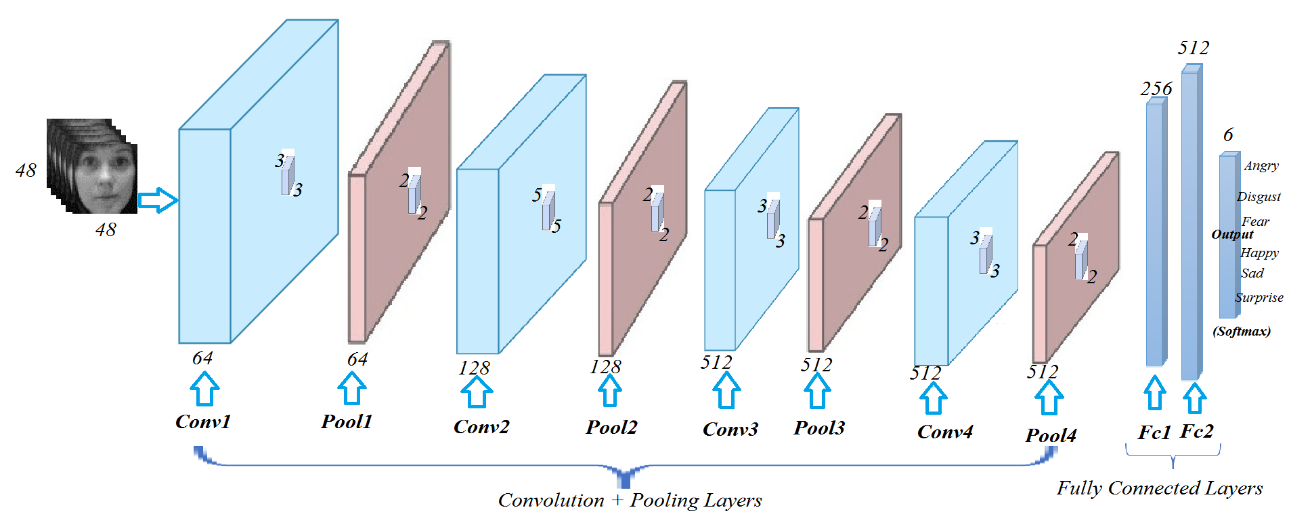
F(x,xj) = tanh (αxay + c) (3,5) (Awasti, 2020)

**Why I decided to implement SVM algorithm?**

The reason I decided to use this algorithm was that SVM is generally mostly used in image classification for predicting emotions accurately. It produces significant accuracy with requirement of less computation.

**3.2 Convolution Neural Network Classification: -**

Convolutional neural networks are examples of deep learning artificial neural networks which are used mostly to classify images or cluster them depending on how similar they are and recognise objects within scenes. Images are treated as tensors by convolutional neural networks during processing. Matrixes of numbers called tensors have extra dimensions. The order of the tensor is the number of dimensions (1, 2, 3, etc.) of a tensor. A single image's height and width decide two row-column dimensions, while its R, G, and B values specify an extra three layers. When using the CNN model to analyse images, the input image is initially evaluated before being filtered. Image feature extraction is a part of filtering. By multiplying, the two functions are related to one another. Convolution is the measure of how much two functions overlap as one passes over the other. The convolution layer generates a feature map.



**Figure 3.2.1.** Convolution Neural Network (CNN) architecture: The process of transmitting an image from the first convolutional layer to the final, fully connected layer that produces output is shown step-by-step in this graphic. The middle layers, which are invisible, aid in processing an image. (Atabansi, 2020)

As can be seen, in my project, each greyscale image of 48x48 pixels is sent through a cascade of multiple layers of convolution neural networks. A different-sized feature map is present for each layer. Each convolution layer's output is a predetermined feature map, and all of these matrices together constitute the new input layer that the following layer will employ. A convolution layer with a kernel size of 3x3 makes up the first layer of a CNN. The second layer is a pooling layer with tiles of size 2x2, which aids in reducing the dimensionality of the data. Just below the convolution layer is this pooling layer. The last layer is fully connected and has an activation function that primarily uses ReLU to provide output.

It starts with a few filters for picking up low-level features. As the CNN is examined more closely, more filters are used to identify high-level features. During the feature extraction process, convolutional nets apply a number of filters to a single image, with each filter recording a different signal. Convolutional networks take these filters, or slices of the image's feature space, and map each one separately by creating a map of every place where that feature appears. Convolutional networks learn several regions of a feature space, enabling scalable and trustworthy feature engineering. Convolutional neural networks for feature extraction and image analysis take in inputs, do a dot product, and opportunistically perform a non-linearity as a follow-up. These neurons have biases and weights that can be taught. From the raw image pixels on one end to the class scores on the other, the complete network continues to reflect a single differentiable score function.

**Why did I use CNN algorithm?**

Here, the subject of why I used a CNN algorithm for my facial emotion recognition research is raised. The CNN algorithm, which is mostly utilised for classification purposes, is the explanation for the response. The accuracy of image categorization is improved by the capability to extract features of an image in order to discern emotions.

**CHAPTER 4**

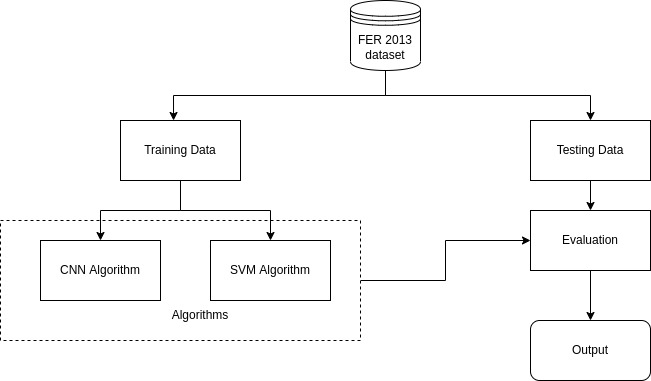
**4.0 Methodology: -**

**4.1 Introduction: -**

The project I'm working on uses the FER 2013 dataset, which is available. This dataset contains grayscale images of people from all over the world. Predictions and analyses are performed using this dataset. Different algorithms are employed to this dataset to calculate the accuracy, precision, and loss values. The algorithms used are compared to one another to see which algorithm works better than the other. This comparison provides the answer to the question 1 mentioned above in chapter 1 section 1. 1. After completing these steps, it is compared to other images to determine whether the anticipated feeling manifested itself in the original image. The overarching design framework is first stated and explained in this chapter. Then, an algorithm is run or applied one by one to this dataset. Each algorithm's step-by-step process and explanation are listed below.

**4.2 Design Implementation: -**

The design that I have implemented is on Facial Expression Recognition FER 2013 dataset.



**Figure 4.2:** Design Implementation of Facial Emotion Recognition:An overall implementation of FER is depicted in the figure. Training data and test data are separated from the dataset. The algorithms CNN and SVM operate on training data before sending the model and its results to the evaluation stage, where they are compared to testing data to produce the predicted image. (Source:Self)

The 38755 48x48 pixel greyscale images in the FER 2013 dataset are split into seven different emotional types. Both training and testing data for the operation are included in this dataset. To perform operations on the training data, the algorithms like CNN and SVM are used. Each of these algorithms completes its task and produces a result in the form of accuracy or loss. This accuracy and loss indicate how many images are correctly calculated and how many are incorrectly categorised by the machine, respectively. The result is then forwarded for evaluation, where the test and current results are compared to see whether there are any similarities in the accuracy or loss of the data. The evaluation stage uses an image from test data to determine whether the image's emotion matches that predicted by the machine or kernel. The detailed implementations of these algorithms are mentioned below in section 4.3.

**4.2.1. Libraries Used: -**

1)NumPy:

An open-source library of python which is used for working matrices and arrays.’np.array’ is an array object which is mostly used in NumPy. Arrays of numbers are the CNN inputs and to convert images to arrays this NumPy is used to perform operations related to CNN along with performing multiplication of matrices. NumPy, a library for the Python programming language, supports large, multi-dimensional arrays and matrices as well as a sizable number of high-level mathematical operations that may be done on these arrays. (NumPy - Wikipedia, 2022)

2)Pandas:

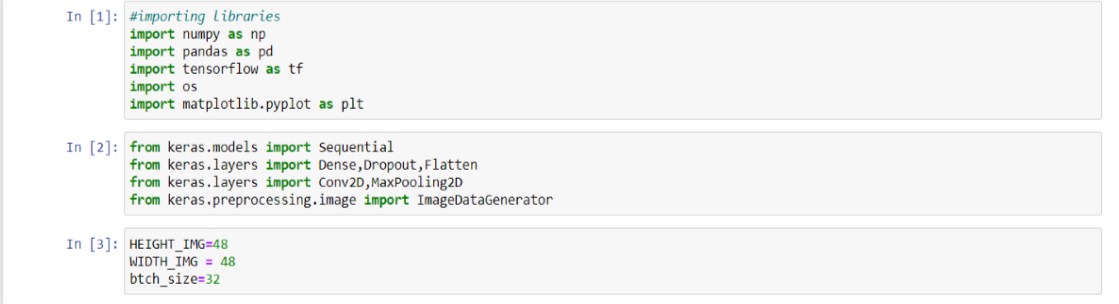
The pandas software library for the Python programming language is used for data manipulation and analysis. It offers data structures and procedures for utilising mathematical tables and time series. Pandas is mostly used in Data Frames for tabular data manipulation and analysis. (Pandas (software) - Wikipedia, 2022)

3)Matplotlib:

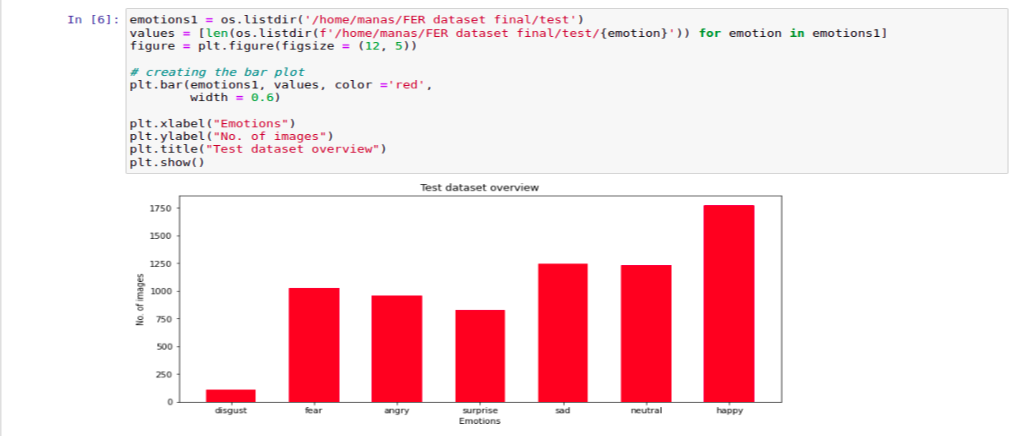
It is a python library which is used to create 2D plots and graphs using python scripts. It contains pyplot module which makes things simple and easy for plotting. It also provides control line style feature, properties of fonts along with axes which can be formatted.

4)Keras:

The Python programming language contains a software library called pandas that is used for data manipulation and analysis. It offers data structures and procedures for handling time series data and mathematical tables. Data Frame analysis and tabular data manipulation are the main uses of Pandas. (Keras - Wikipedia, 2022)

**Figure 4.2.1.1.** Libraries imported into the environment: As observed in the above image, several libraries that are necessary are being imported.(Source:Made by me on Jupyter notebook)  
The most popular libraries imported into the code are NumPy and pandas. In order to create graphs and bar plots that aid in visualisation, the matplot library is imported.

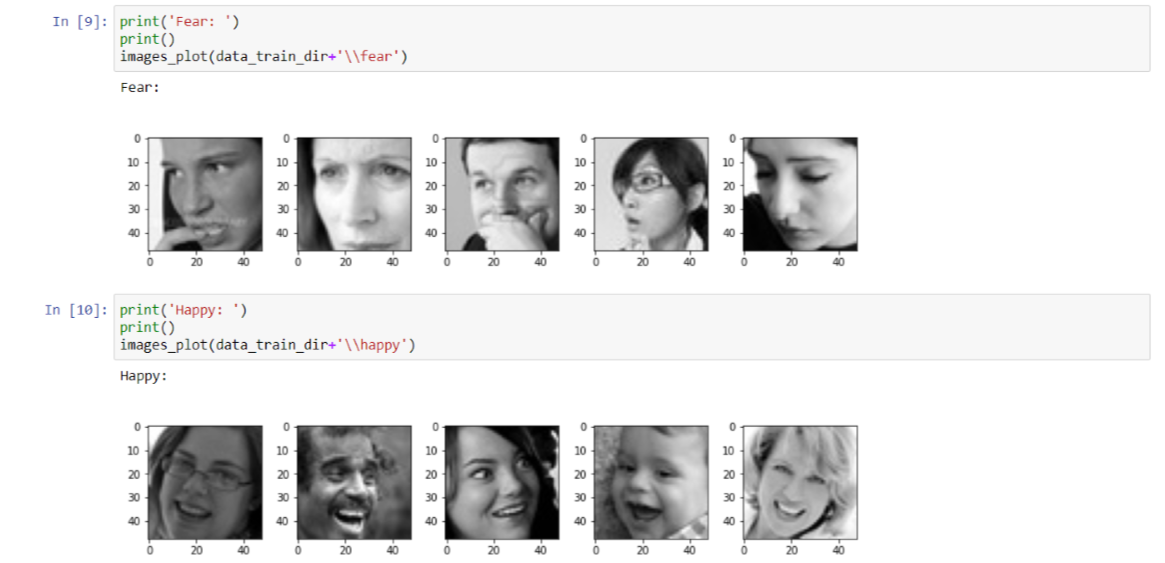
  
**Figure 4.2.1.2.** Bar plot of Train Data: Seven emotions are shown in this above-the-bar figure, each with an associated image. The number of images that each emotion has from the train data is represented on the Y-axis, while the X-axis represents a division of the emotions.(Source:Self)

  
**Figure 4.2.1.3.** Bar Plot of Testing data: Seven types of emotions are represented by their corresponding images in the above bar plot. The Y-axis represents the number of images from the test data that correspond to each emotion, and the X-axis is the number of emotions divided.(Source:Created by me on jupyter notebook)

The red bars in this bar graph indicate how many images each of these emotions has. The highest bar represents the happy emotion and contains more images, whereas the small bar at the start of the bar plot represents the disgust emotion and contains less test data.

  
**Figure 4.2.1.4.** Images from train data with the emotion Angry: Images of angry emotion taken from the test results are shown in this graphic. Each image is 6x6 inches in height and width and is in grayscale.(Created by me on jupyter notebook)

The "images plot" function is shown in the above figure. It is made up of an image directory and a top variable where the user can specify how many images should be displayed. The term "files\_image" is defined, and a for loop is used to select 5 images. The image from file is read and plotted using the function "plt.imread."

  
**Figure 4.2.1.5** Images from train data with the emotions Fear and Happy: Along with face expressions, this figure also depicts representations of the emotions of fear and happiness.(Source:made by me in jupyter notebook)

  
**Figure 4.2.1.6**. Images from train data with the emotions Sad and Disgust: This figure includes several depictions that evoke emotions of sadness and disgust. Also, it's only for understanding the precise emotions that some images evoke.(Source:Self)

**Figure 4.2.1.7.** Images from train data with the emotions Neutral and Surprise: This figure includes some neutral and surprise emotion images. This is merely to understand the specifics of each emotion and their corresponding images(Source:Created by me in jupyter notebook)

**4.2.2. Splitting Train data,Validation data and Test data:**

  
**Figure 4.2.2.1.** Splitting data into train, validation, and test data: In this figure, image augmentation is done prior to data separation. Data is then separated into test generator, validation generator, and train generator.(Source:Self)

Figure 4.2.2.1 above separates train data, validation data, and test data. Image augmentation is carried out in the first section using an image data generator. This image data generator produces data augmentation in batches. The primary application of ImageDataGenerator is to obtain original data input, transform that data in a subsequent step on a random basis, and output newly changed data. The rescale factor in the training data generator is used to rescale the figure to a specific scale. It also includes additional characteristics that aid in carrying out activities correctly.

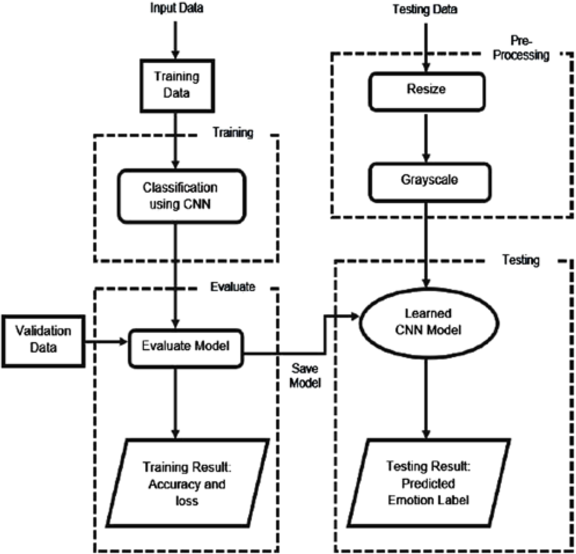
The train generator part includes a train data directory and a colour, grayscale image. Image width and image height make up the required target size. A variable called "batch size" stores the number of images needed for a certain batch of activities.

he parameters for the validation generator section are the same as those for the train generator previously discussed. A subset is the only parameter that differs from the train generator. Since this is validation data that we obtain from training data, validation is used as a subset in this instance.

The test generator is made up of a test data directory with the same parameters as those previously described. It is a test data set that will be applied to additional data testing.

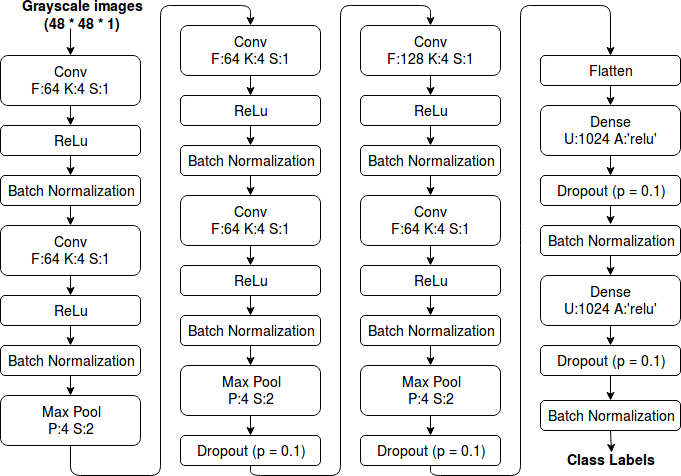
**4.3 Implementation of CNN algorithm on dataset: -**

This algorithm categorises every single image into universal seven emotions that are described as sad,happy,fear,surprise,neutral,disgust and angry as mentioned in dataset. This CNN algorithm of neural networks exhibited success in processing of an image.Firsty dataset was separated into two parts training data and test data and then trained on set of this training data. Before being provided to CNN, the data was not subjected to a feature extraction process. Multiple CNN architectures were tested in order to increase accuracy with the validation set while reducing overfitting. The CNN architecture is mentioned below:



**Figure 4.3.1.** CNN algorithm implementation:Figure above depicts the design involved in implementing the CNN algorithm, from dataset division to training and evaluation of data. When it reaches testing stage, it is compared along with test data to determine predicted emotion from image. (Zahara, Musa and Karim, 2020)

The dataset must first be split into train data and test data before being sent to the training stage of CNN classification. The entire CNN categorization of an image data process occurs at this stage. The evaluation step follows the classification stage, and training data from that stage is given along with validation data to determine training accuracy and training loss. This model is reviewed, saved, and then sent to the testing stage. The tested model is given to the learned CNN model for evaluation using test data that has first undergone pre-processing and resizing before being put through testing. We receive testing loss and accuracy as output.



**Figure 4.3.2** Convolution Neural Network (CNN) architecture:The above figure shows flow of algorithm from starting stage which is convolution layer where greyscale images through series of layers to the last stage which is dense layer that result in output as class labels. (Talegaonkar et al., 2019)

The steps that make up the CNN network are as follows:

1. **Convolution Layer: -**

It is the first stage of CNN algorithm that contain feature size of 64 with stride length of 1. In this layer 48 x 48 x 1 greyscale image is passed to this layer.A filter which is learnable and represented randomly in convolution layer is slid over the input or convolved. The operation computes the dot product between each local input region and the filter. The output, also known as the feature map, is a 3D volume comprising several filters. (Talegaonkar et al., 2019)

1. **Batch Normalization: -**

The batch normalizer speeds up the training process by applying a transformation that maintains the activation mean close to 0 and the activation standard deviation close to 1, as shown in figure 4.3.1. (Talegaonkar et al., 2019)

1. **Max Pooling: -**

In order to decrease the spatial size of the input layer and the cost of calculation, the pooling layer is used. (Talegaonkar et al., 2019)

1. **Fully Connected Layer: -**

In a completely connected layer, each and every neuron from previous layer is connected to neurons that serve as an output. The size of the final output layer depends on how many classes the input image needs to be divided into.. (Talegaonkar et al., 2019)

1. **Activation Function: -**

Utilizing activation functions helps to cut down on overfitting. The ReLu activation function has been employed in the CNN architecture. Benefit: The gradient of the ReLu activation function is always equal to 1, showing that most of the mistake is transmitted backwards.

The equation of ReLu activation function is given by:

f(x) = max (0, x) (4,1) (Talegaonkar et al., 2019)

1. **Softmax:-**

N real numbers are passed as a vector to the softmax function. Then it normalizes that vector's values into a range between (0, 1).

**Evaluating a model: -**

The model created during the training stage was then assessed using 7185 images from the validation set. (Talegaonkar et al., 2019)

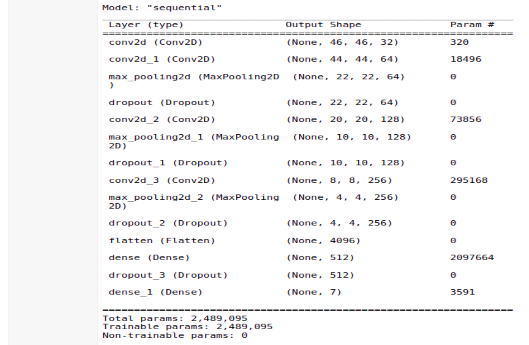
The evaluated model is then put to the test using test data. After the model has been loaded, several images created or extracted from the test data are tested for accuracy. The results are then tested on a selection of photos that contain both the original image's original emotion and the model's predicted emotion for the same image.

**Compiling a model: -**

The primary concept that needs to be understood while building a model is that of optimizers. These machine learning methods use several optimizers that are accessible. In this approach, I used a categorical cross-entropy loss function and an Adam optimizer with a learning rate of 0.001. Because the dataset is categorical in nature and uses Softmax as the activation function for multiclass classification. Accuracy, precision, loss, and recall metrics are used. Section 4.5, "Evaluation Metrics," provides a discussion of these evaluation metrics.

  
  
**Figure 4.3.3.** Algorithm of Convolution Neural Network: The CNN algorithm's steps, including the convolution layer that is employed, the amount of max pooling, and the dropout layer, are shown in the above figure.(Source:Jupyter Notebook by me)

The first step is to add a convolution layer with a 32 by 3 by 3 kernels using ReLu as the activation function. The maximum pooling layer, which has a pool of size 2x2, is then added. Use is made of the 0.1 dropout layer. The first stage of processing is finished at this point. A second convolution layer with a kernel size equal to 3x3 and a size of 128 is added in the following step, along with the ReLu activation function. The image is then flattened and the dense layer of 512 is added along with ReLu activation function. A third convolution layer with a kernel size equal to 3x3 and a size of 256 is added further in the following step, along with the ReLu activation function. The output layer which is dense layer of size 7 and activation function softmax is added further.

  
**Figure 4.3.4.** Summary of CNN algorithm model: The top diagram summarises the steps taken by the CNN algorithm to process an image, from the convolutional layer at the beginning to the dense layer at the end.(Source:Self)

This report is divided into sections that include the type of layer, the output shape, and the estimated parameters. This breakdown of an image's size and processing steps from beginning to end provides a quick understanding of how an image is processed.

  
**Figure 4.3.5**. Epochs, training data, and the model fitting stage: The model in this figure is prepared for training and fitting. The parameters of the model, which include steps per epoch, validation data, and validation steps, are detailed below.(Source:Jupyter Notebook by me)

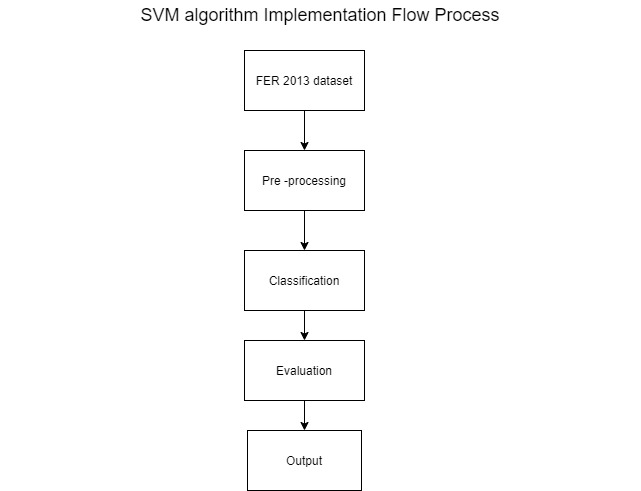
20 epochs, training data, and validation data are utilised to train the model in figure 4.3.3 above. The train data length divided by the train data batch size is contained in the steps per epoch variable. The number of steps necessary for training validation data is indicated by the validation steps. Given by the length of the validation data divided by the size of the validation data batch.

  
**Figure 4.3.6.** History of CNN algorithm model: The metrics and parameters used and calculated, including as accuracy, loss, precision, and recall, are shown in detail in this figure.(Source:Self)

The number of epochs that are supplied as an input parameter is visible in picture 4.3.4 above. This figure illustrates the findings of samples or images step-by-step. Additionally, it includes the batch size indicated under epoch variable. The calculations produce a number of metrics, including training accuracy, loss, precision, and recall scores as well as validation accuracy, precision, loss, and recall scores. The outcomes of this approach are described in the results and discussion section in chapter 5 below.

**4.4 Implementation of SVM algorithm on dataset: -**

Finding the optimum hyperplane to partition binary classes at the greatest feasible distance with the fewest amount of support vectors is the basic objective of SVM. SVM functions primarily as a binary classifier. Using one-vs-one (OVO) or one-vs-all (OVA) approaches, SVM is enhanced for multi-class classification.



**Figure 4.4.1.** Design Implementation of SVM algorithm: An illustration of the design flow for the SVM algorithm is shown above. There are several stages in it, starting with loading the dataset and continuing through pre-processing, classification, evaluation, and ending with the output stage.(Source:Self)

As seen in figure 4.4.1 above, the design starts with importing the dataset into a pre-processing stage. This design's second stage of implementation deals with scaling down the data into an accessible format so that it may be provided to the classifier for better understanding and classification, preventing distorted results. The parameters kernel type, gamma value, and C value are supplied to the classifier SVC, which is employed in this classification stage. In an evaluation stage, the trained output is compared to test data to forecast the metrics after passing these tests. After moving on to the evaluation stage, a product is produced called an evaluation metric, which aids in determining the accuracy or loss of the data.

Unbalanced training datasets are frequently a result of OVA. It also creates a significant imbalance for a few classes, including disgusted facial expression, in the Fer2013 dataset. There will be just 436 positive occurrences and 28273 negative occurrences if a binary classifier for disgust emotion is trained using OVA, only 1.5% of the training occurrences will be positive ones.

**Figure 4.4.2.** SVM algorithm with rbf kernel: First SVM library is implemented in this figure using Sklearn. Then, the parameters for the SVC classifier are set, including the kernel type='rbf', the auto-generated gamma value, and the model variable-stored C value. After that, the model is fitted to train data, and the fit's outcome is then compared to test data results.

**Figure 4.4.3.** SVM algorithm with linear kernel: First SVM library is included in this figure using Sklearn. Then, the parameters for the SVC classifier are set, including the kernel type='linear', the auto-generated gamma value, and the model variable-stored C value. After that, the model is fitted to train data, and the fit's outcome is then compared to test data results.(Source:Jupyter notebook by me)

**4.5. Evaluation Metrices:**

The primary evaluation measures used to assess a model's efficacy are accuracy, loss, precision, recall, and f1-score. Each of these indicators is explained in more detail below.

**Accuracy:**

It measures the proportion of accurate predictions to all predictions. The formula of accuracy is given by

Accuracy = (4,1)

**Loss:**

Categorical cross-entropy is the loss function employed, and it is provided by

Loss= - (4,2)

binary indicator (0 or 1) is y.

predicted probability is y.

number of classes(sad,neutral,angry,fear) is m.

**Precision:**

When true positive results are divided by the sum of true positives and false positives, precision is calculated as follows:

Precision = (4,3)

where TP denotes True Positives and FP denotes false positives.

**Recall:**

Recall metrics are calculated by dividing the total of all true positive findings by the sum of all true positives and all false negative results.

Recall = (4,4)

where TP=True Positives and FN=False Negative

**Confusion matrix:**

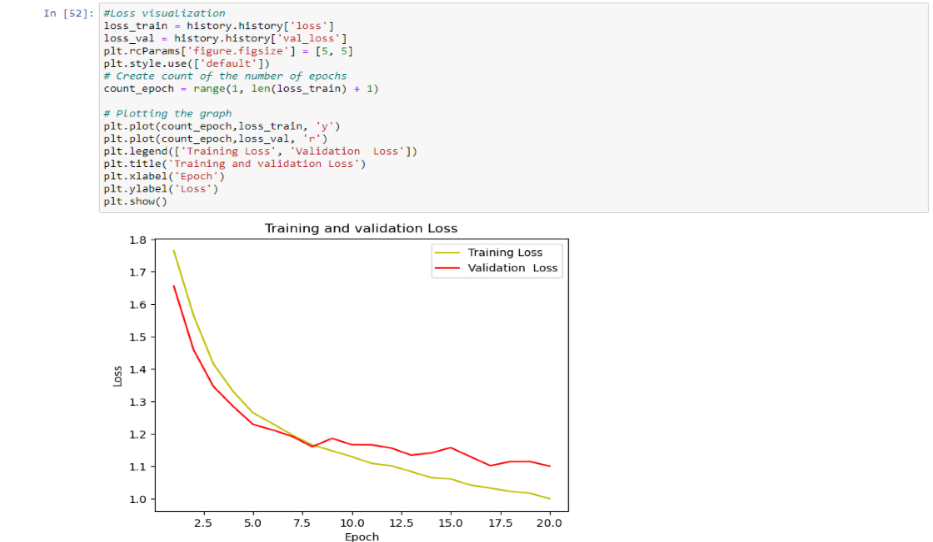
The True Positives (TP), True Negatives (TN), False Positives (FP), and False Negatives (FN) values are provided for four combinations of true and predicted values. The numbers in this combination are used to compute metrics like loss, accuracy, and f-score. Correct prediction of emotion is True positives (TP), Incorrect prediction of emotion is false positives (FP), Correct prediction of incorrect emotion is True Negatives (TN) and incorrect prediction of incorrect emotion is a False Negatives (FN). (Adil, 2022)

**CHAPTER 5**

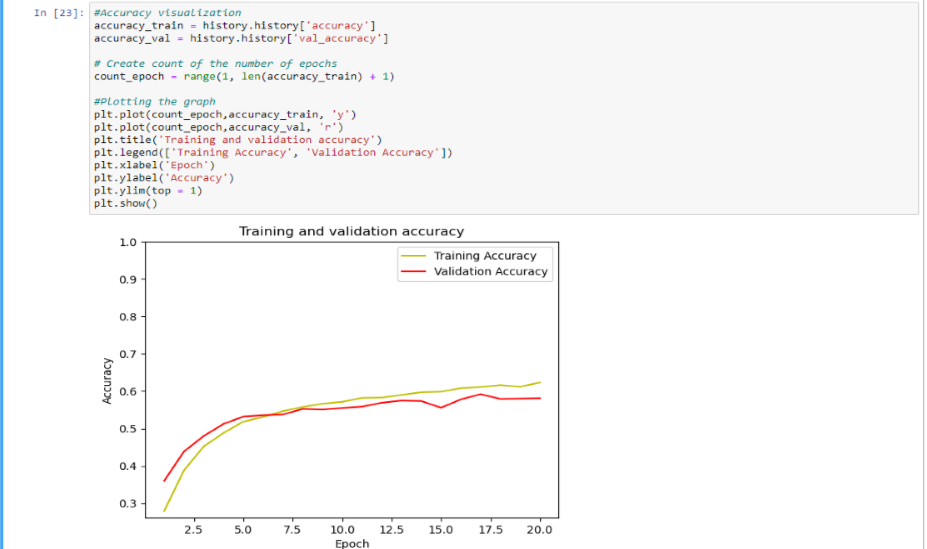
**5.0 Results and Discussion: -**

The outcomes of the implementation of the algorithms CNN and SVM are covered in this chapter. Below is a comparison of these algorithms' results that will assist in answering my research questions.

**5.1 Result of CNN: -**

  
**Figure 5.1.1**. Loss of training and validation data plotted against 20 epochs: This graph displays a comparison between training data and validation data. The validation loss is shown by the red line, while the training loss is shown by the yellow line.(Source:Self)

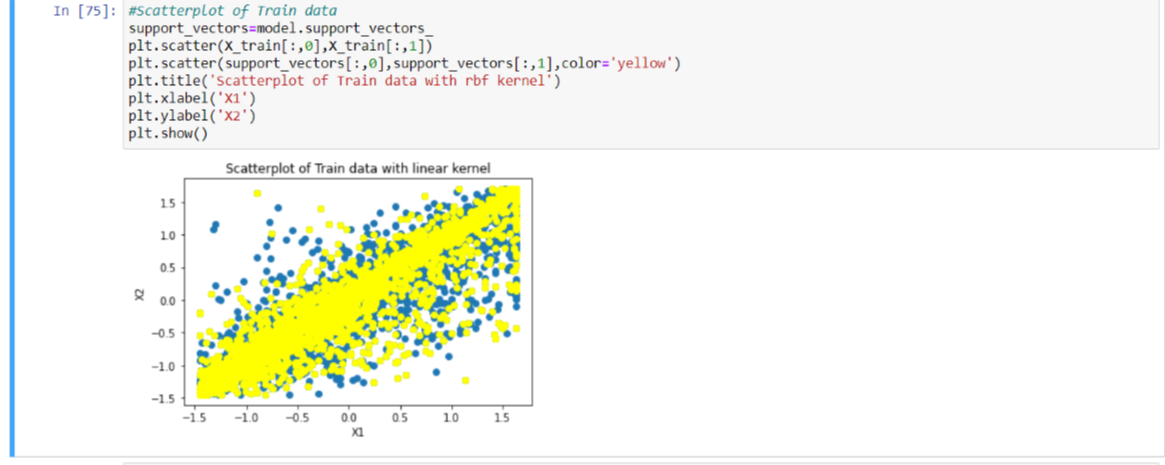
As observed in figure 5.1.1 above, training loss and validation loss are both more than 1.5 at the beginning. However, when the model is trained, the train loss begins to decline and the rate of data validation is matched. The distance between training loss and validation loss begins to grow once around 8 epochs are complete. The training loss decreases linearly when additional epochs are conducted, while the validation loss struggles to attain its minimum value more.

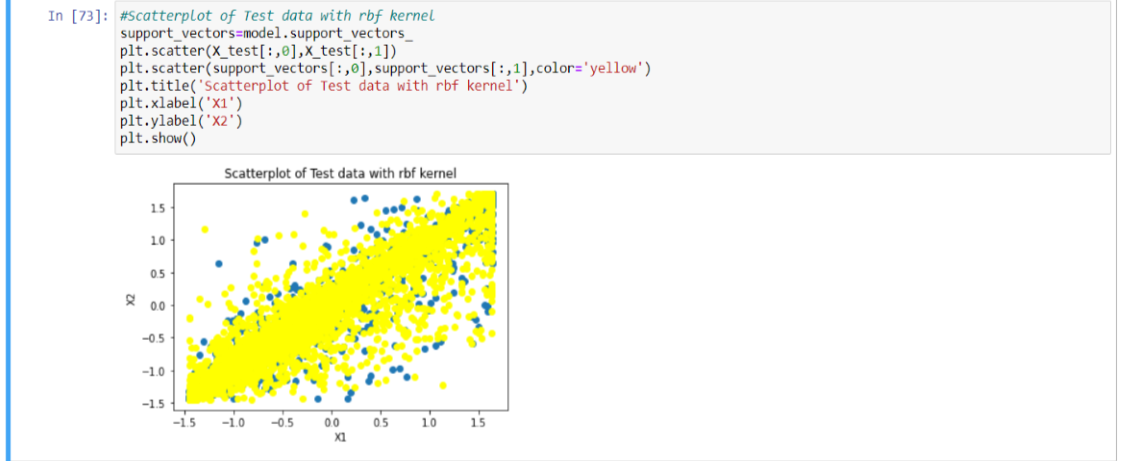
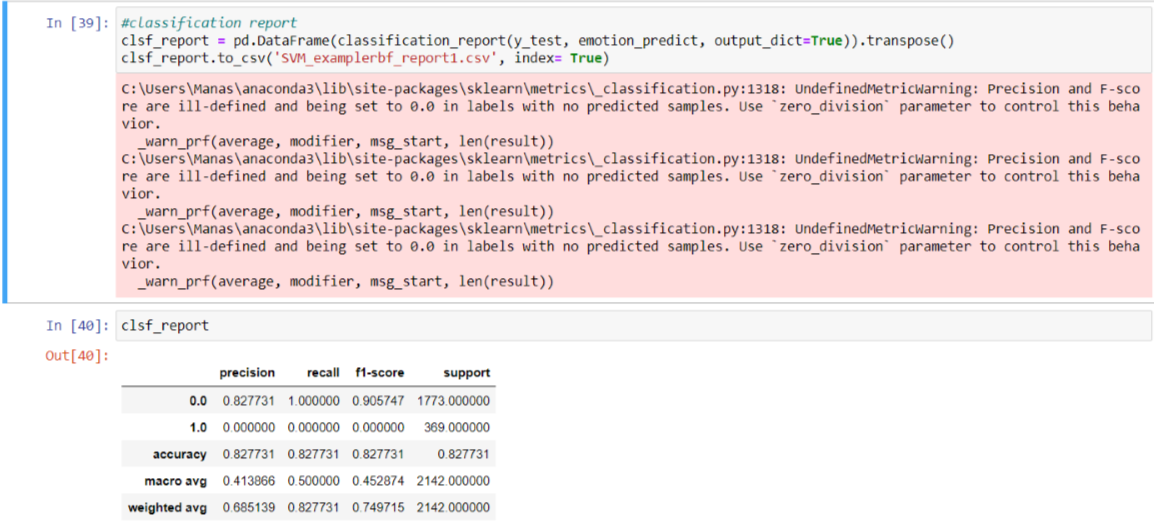
  
**Figure 5.1.2.** Accuracy of training and validation data plotted against 20 epochs: The accuracy of the training and validation sets as a function of the number of epochs is shown in the graph above. Number of epochs is represented on the X-axis, and accuracy is plotted on the Y-axis. The red line represents data validation accuracy, and the yellow line represents data training accuracy.(Source:Self)

The correctness of the data used for training and validation is shown in figure 5.1.2 above. It is clear that when training progresses through the epochs, accuracy of valid data begins to train and increase quickly. Both lines start to become constant as epochs approach a value of 8, retaining an accuracy of more than 60%.

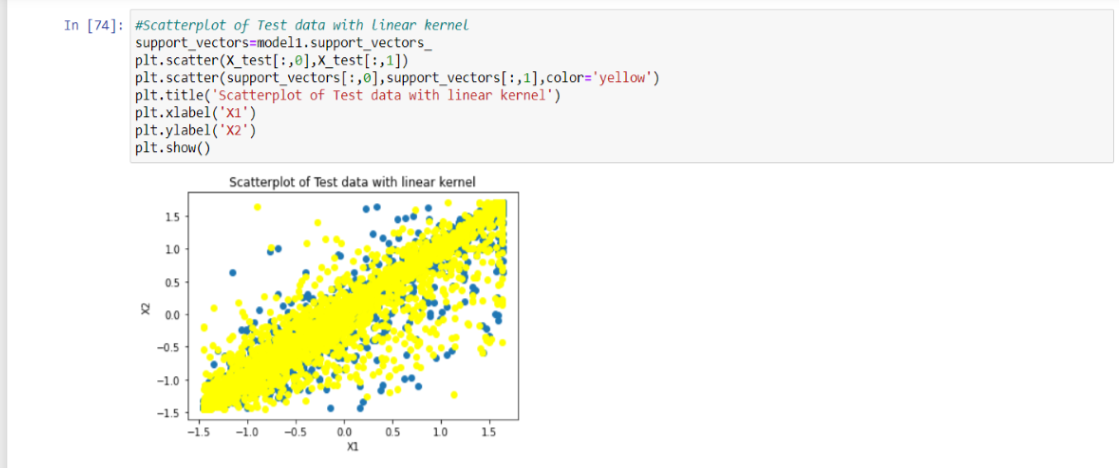
**5.2 Result of SVM: -**

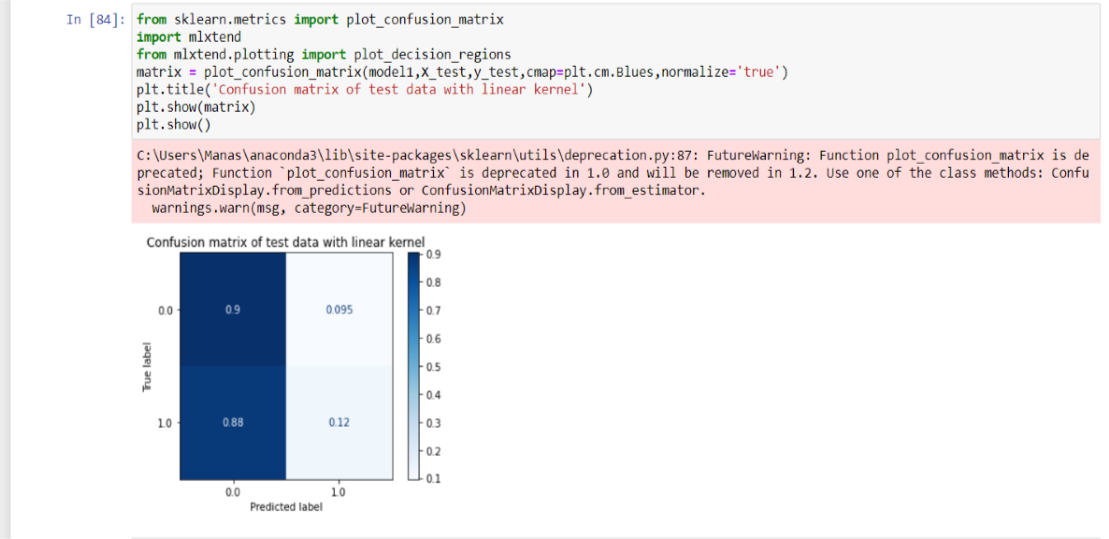
**5.2.1 SVM result with rbf Kernel**

**Figure 5.2.1.1.** Scatterplot of Train data: Above figure depicts scatterplot of training data with linear kernel. The blue dot s and yellow dots are the support vectors that are places in a hyperplane.(Source:Jupyter Notebook by me)

**Figure 5.2.1.2.** Scatterplot of Test data with rbf kernel: In above figure support vectors are plotted on a hyperplane and it can be seen from figure that there is large number of one type of support vector.(Source:Jupyter Notebook by me)  
  
**Figure 5.2.1.3** Classification Report of SVM with rbf kernel: This figure is of a classification report of output data that contain precison,recall and f1-score parameters along with the accuracy. (Source: Created by me in Jupiter notebook)

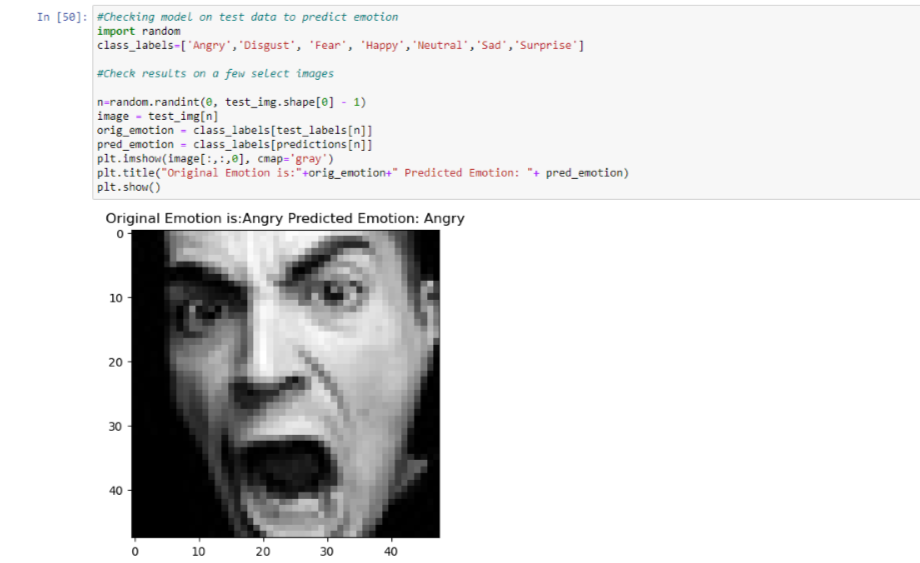
**5.2.2 SVM result with linear kernel:**

**Figure 5.2.2.1** Scatterplot of test data with linear kernel: This figure talks about scatterplot of test data by using linear kernel. The yellow dot and blue dot are the support vectors that are scattered on a hyperplane.(Source:Self)

  
**Figure 5.2.2.2.** Confusion Matrix of SVM with linear kernel: In above figure the confusion matrix of a test data is plotted using linear kernel.(Source: Self)

The prediction results on the categorization task are summarised in the confusion matrix above. Each class's share of correct and incorrect predictions is tallied up and broken-down using count values.

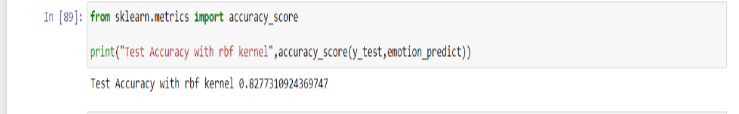
  
**Figure 5.2.2.3.** Classification report of SVM with linear kernel: The image above shows a classification report with different metrics, including recall, precision, and f1 score with accuracy score.  
**5.3 Original Emotion Vs Predicted Emotion Result: -**

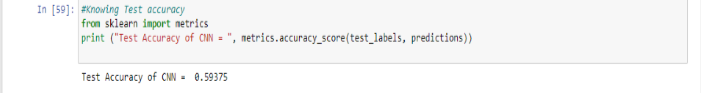
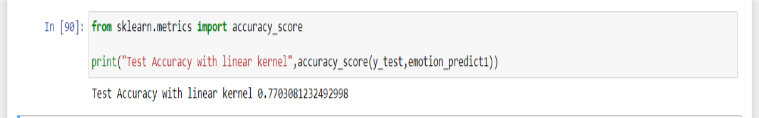
  
**Figure 5.3.1** Original Emotion Vs Predicted Emotion result: This result is obtained using CNN algorithm to know how machine learning classifies original emotion from the predicted emotion from image.(Source: Created by me in Jupiter Notebook)

The training and validation set of data were being used to implement the CNN algorithm. The model begins training with train data after applying the algorithm, and the results are saved in the model. Then, using sample test data, this model is loaded onto a machine and used to forecast emotion based on the original emotion.

**5.4 Comparing CNN and SVM algorithm: -**

Both these algorithms are performed on training and test data of FER dataset. The result after training data is evaluated with test data and the result in the form of accuracy is generated. The accuracies of both these algorithms are compared to know which can perform better than another algorithm.

**Figure 5.4.1**. SVM model accuracy test with rbf kernel: This test data accuracy, which is 82%, was obtained using the rbf kernel. (Source:Jupyter Notebook by me)

**Figure 5.4.2.**CNN model accuracy test: With 20 epochs and the Adam optimizer, test data accuracy is obtained, and it is 59%..(Source: In Jupyter notebook by me)  
**Figure 5.4.3.** SVM model accuracy test with linear kernel: This test data's accuracy with the linear kernel is 77%.(Source: Created by me in Jupyter Notebook)

Figures 5.4.1 and 5.4.2 show that an SVM algorithm has a higher accuracy score or rate than the CNN algorithm. The SVM algorithm's accuracy rate of 82% is far higher than that of the CNN algorithm. The accuracy achieved by the CNN model, even though just 20 epochs were used to train it, is about 60%, which is less than SVM. Both of these algorithms run on training data, with the results subsequently being tested on test data.

**5.5 Discussion: -**

Finding answers to the research questions I came across in this study will aid with real-world problem solving and reveal the research's limitations. The first question that I have encountered in my research was ‘Is facial emotion recognition technology utilised to foretell facial emotions in the real world or not’. The answer to this question is yes, this technology is utilised in real world to foretell facial emotions. More information on this given subject is provided in the literature review section 2.1 above, where I also mentioned some of the real-world applications for this technology. It is clear from this part that facial emotion recognition technology is employed in the majority of these applications.

Does the SVM method outperform the CNN algorithm, which is my second research question? The findings of the comparison between these two algorithms where I compared the accuracies of two algorithms where I got SVM accuracy of 82% with rbf kernel and 59% accuracy in CNN with adam optimizer which is mentioned in results section 5.4 allows me to confirm my second research question hypothesis, which is that the SVM algorithm outperforms the CNN algorithm in terms of assessment metrics like accuracy. I used the SVM algorithm on training data with a linear kernel to verify that this is correct, and it also provided me with 77% accuracy, which is higher than the CNN algorithm. This further demonstrates that the SVM method outperforms CNN, as seen in figure 5.3.3 above.

It can also be argued that this face expression recognition technology has some drawbacks, such as poor image quality, which reduces the efficacy of facial recognition. When compared to digital cameras, the image quality of scanning videos is very bad. Small image sizes make facial recognition particularly challenging. For instance, a small image size combined with the target's distance from the camera results in an identified face that is just 100 to 200 pixels on a side. (Edgell and Trimpe, 2022)

When high-definition cameras are available in the future, these limitations can be improved. The ability of facial recognition algorithms to recognise faces in a database of registered individuals will improve.

**CHAPTER 6**

**6.0 Conclusion & Future work: -**

**6.1 Conclusion: -**

From the aforementioned sections, it is clear that facial emotion recognition technology is crucial for accurately predicting and analysing facial expressions in the actual world. The research focused on using several machine learning algorithms to this technology and provided a brief overview of how some of these algorithm's aid in accurately predicting emotion. The research questions that raised are also solved here and all the aims and objectives have been fulfilled. I have just focused on mainly accuracy parameter of both algorithms but other parameters can also be used to get result.

**6.2 Future Work: -**

I have used two machine learning algorithms in this research to forecast face expressions. This technique can be expanded to include additional sorts of algorithms that will improve the ability to recognise facial expressions of emotion. A new dataset of various facial emotions or expressions can be created in addition to the numerous datasets of facial emotion recognition that have already been developed. This dataset will assist in identifying age, gender, and other characteristics that define individuals in addition to predicting emotions. Future developments in face emotion technology will be benefited by the findings of this experiment.

**7.0 References: -**

Sheaffer, B., Golden, J. and Averett, P., 2009. Facial expression recognition deficits and faulty learning: Implications for theoretical models and clinical applications. *International Journal of Behavioral Consultation and Therapy*, 5(1), pp.31-55.

Khan, A., 2022. Facial Emotion Recognition Using Conventional Machine Learning and Deep Learning Methods: Current Achievements, Analysis and Remaining Challenges. *Information*, 13(6), p.268.

Dukić, D. and Sovic Krzic, A., 2022. Real-Time Facial Expression Recognition Using Deep Learning with Application in the Active Classroom Environment. *Electronics*, 11(8), p.1240.

En.wikipedia.org. 2022. *Convolutional neural network - Wikipedia*. [online] Available at: <https://en.wikipedia.org/wiki/Convolutional\_neural\_network> [Accessed 22 September 2022].

En.wikipedia.org. 2022. *Support-vector machine - Wikipedia*. [online] Available at: <https://en.wikipedia.org/wiki/Support-vector\_machine> [Accessed 22 September 2022].

Jawale, A. and Magar, G., 2019. Comparison of Image Classification Techniques: Binary and Multiclass using Convolutional Neural Network and Support Vector Machines. *ResearchGate*, [online] Available at: <https://www.researchgate.net/publication/344718153\_Comparison\_of\_Image\_Classification\_Techniques\_Binary\_and\_Multiclass\_using\_Convolutional\_Neural\_Network\_and\_Support\_Vector\_Machines> [Accessed 22 September 2022].

Awasti, S., 2020. *Seven Most Popular SVM Kernels*. [online] Dataaspirant. Available at: <https://dataaspirant.com/svm-kernels/> [Accessed 22 September 2022].

Talegaonkar, I., Joshi, K., Valunj, S., Kohok, R. and Kulkarni, A., 2019. *Real Time Facial Expression Recognition using Deep Learning*. [online] SSRN. Available at: <https://papers.ssrn.com/sol3/papers.cfm?abstract\_id=3421486> [Accessed 22 September 2022].

Adil, M., 2022. *Facial Emotion Detection Using Convolutional Neural Networks*. [online] Dspace.library.uvic.ca. Available at: <https://dspace.library.uvic.ca/handle/1828/13388> [Accessed 22 September 2022].

Edgell, J. and Trimpe, A., 2022. *4 Limitations of Facial Recognition Technology*. [online] Technology Solutions That Drive Government. Available at: <https://fedtechmagazine.com/article/2013/11/4-limitations-facial-recognition-technology> [Accessed 22 September 2022].

En.wikipedia.org. 2022. *Keras - Wikipedia*. [online] Available at: <https://en.wikipedia.org/wiki/Keras> [Accessed 22 September 2022].

En.wikipedia.org. 2022. *pandas (software) - Wikipedia*. [online] Available at: <https://en.wikipedia.org/wiki/Pandas\_(software)> [Accessed 22 September 2022].

En.wikipedia.org. 2022. *NumPy - Wikipedia*. [online] Available at: <https://en.wikipedia.org/wiki/NumPy> [Accessed 22 September 2022].

Dr Olga, A. (2021). MSc Project Handbook. Internal Project Handbook Unpublished.

[Herts.ac.uk](http://herts.ac.uk/). 2021. Academic Integrity and Academic Misconduct. [online] Available at: <<https://www.herts.ac.uk/__data/assets/pdf_file/0007/237625/AS14-Apx3-Academic-Misconduct.pdf>> [Accessed 22 September 2022]

Atabansi, C., 2020. Application of Convolutional Neural Network for Facial Expression Recognition. *ICRAI*,.

Zahara, L., Musa, P. and Karim, I., 2020. The Facial Emotion Recognition (FER-2013) Dataset for Prediction System of Micro-Expressions Face Using the Convolutional Neural Network (CNN) Algorithm based Raspberry Pi. *IEEE Eplore*,.

Dubey, M. and Singh, P., 2016. Automatic Emotion Recognition Using Facial Expression:A Review. *IRJET*, 3(2).

Waseem, M., 2022. *Support Vector Machine In Python | Classification Algorithms | Edureka*. [online] Edureka. Available at: <https://www.edureka.co/blog/support-vector-machine-in-python/> [Accessed 23 September 2022].

**8.0 Appendices: -**

**8.1. Appendix 1. Support Vector Machines (SVM): -**

The first SVM algorithm was developed in 1963 by Vladimir N. Vapnik and Alexey Ya Chervonenkis. Bernhard Boser, Isabelle Guyon, and Vladimir Vapnik suggested a method to create nonlinear classifiers in 1992 by utilising the kernel trick on maximum-margin hyperplanes.

**Multi-class SVM:**

Support-vector machines are used in Multiclass SVM to assign labels to instances, and the labels are selected from a limited collection of various elements. Reducing a single multiclass problem to a number of binary classification questions is the most popular method for doing this. Common methods contain:

* constructing binary classifiers that can differentiate between each pair of classes or between each label and the rest (one-versus-all) (one-versus-one). In the one-versus-all example, new instances are classified using a winner-takes-all technique, where the classifier with highest output function designates the class. To classify instances in the one-versus-one technique, a max-wins voting approach is used. In this method, each classifier assigns an instance to either of the two classes, increases the vote total for the allocated class by one vote, and then chooses the class that has received the most votes overall.
* Directed acyclic SVM (DAGSVM)
* Output code which are error-correcting.

Instead of breaking the multiclass classification problem down into numerous binary classification problems, Crammer and Singer devised a multiclass SVM method that transforms the problem into a single optimization problem.

**SVM Kernel:**

The SVM kernel is a function that converts non separable problems into separable problems by taking low-dimensional input space and transforming it into higher-dimensional space. It works best in non-linear separation issues. Simply explained, the kernel determines how to split the data depending on the labels or outputs defined after performing some incredibly sophisticated data transformations.SVM’s are of two types:

**1)Linear SVM: -**

Simply explained, the kernel determines how to split the data depending on the labels or outputs defined after performing some incredibly sophisticated data transformations.

**2)Non-linear SVM: -**

When a dataset cannot be identified using a straight line, it is said to be non-linear, and the classification algorithm utilised is known as a non-linear SVM classifier.

Support Vectors and Hyperplane in SVM: -

**Hyperplane: -**

In n-dimensional space, there may be several lines or decision boundaries used to separate the classes, but we must identify the optimum decision boundary that best aids in classifying the data points. The hyperplane of SVM is a name for this ideal boundary.

The dataset's features determine the hyperplane's dimensions, therefore if there are just two features (as in the example image), the hyperplane will be a straight line. Additionally, if there are three features, the hyperplane will only have two dimensions. We always build a hyperplane with a maximum margin, or the greatest possible separation between the data points.

**Support vectors: -**

Support vectors are the data points or vectors that are closest to the hyperplane and have the greatest influence on where the hyperplane is located. These vectors are called support vectors because they support the hyperplane.

**8.2. Appendix 2. Convolution Neural Networks (CNN)**

An advanced form of artificial neural network known as a convolutional neural network substitutes the mathematical operation known as convolution for generic matrix multiplication in at least one of its layers. They are employed in image processing and recognition since they were created primarily to process pixel data.

**Hyperparameters: -**

**Kernel Size: -**

The total amount of pixels processed makes up the kernel. Usually, it is stated as the kernel's dimensions, such as 2x2 or 3x3.

**Padding: -**

Padding is the addition of extra pixels to an image's borders that are normally 0-valued. This is done to prevent the boundary pixels from being devalued (lost) from the output since they typically only take part in one instance of the receptive field. Typically, the padding used is one less than the appropriate kernel dimension. Using 3x3 kernels as an example, a convolutional layer would get a 2-pixel pad, or one pixel on every side of the image.

**Stride: -**

The analysis window advances a certain number of pixels per cycle, which is known as the stride. With a stride of 2, each kernel is 2 pixels farther from the one before it.

**No. Of filters: -**

Layers close to the input layer typically contain fewer filters, whereas higher layers can include more because feature map size is reduced with depth. The multiplication of component values va with pixel position is roughly constant across levels to normalise computation at each layer. The quantity of feature maps directly affects the capacity and is influenced by the quantity of accessible examples and the difficulty of the task.

**Filter size: -**

The most typical filter sizes mentioned in the literature come in a wide range and are often determined by the data collection. Finding the ideal level of granularity is difficult since it is necessary to produce abstractions at the suitable scale, given a certain data collection, without overfitting.

**Pooling type and size: -**

The most common method is max pooling, frequently with a 2x2 dimension. This suggests that the input has been considerably downscaled, cutting the cost of processing. 4x4 pooling in the bottom levels may be necessary for large input volumes. Greater pooling diminishes the signal's size and could cause unacceptably high information loss. The most effective pooling windows frequently don't coincide.

**8.3. Appendix 3 Project Plan: -**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Task No | Task | Start Date | End Date | Duration | Completion Status before | Completion Status now |
| 1 | Working on DPP | 3-06-2022 | 25-06-2022 | 22 days | Completed | Completed |
| 2 | DPP Submission | 26-06-2022 | 26-06-2022 | 1 day | Completed | Completed |
| 3 | Working on IPR | 27-06-2022 | 11-07-2022 | 15 days | Completed | Completed |
| 4 | IPR Submission | 15-07-2022 | 15-07-2022 | 1 day | Completed | Completed |
| 5 | Working on FPR | 17-07-2022 | 22-09-2022 | 39 days | Pending | Completed |
| 6 | FPR Submission | 23-09-2022 | 23-09-2022 | 1 day | Pending | Completed |

**8.4. Appendix 4**

**Project Code**

**#Programming of algorithms**

**#Importing libraries**

import numpy as np #Importing numpy library from NumPy

import pandas as pd #Importing pandas' library

import tensorflow as tf

import os

import matplotlib.pyplot as plt #Importing matplotlib library for plotting graphs

**#Importing keras CNN model libraries**

from keras.models import Sequential

from keras.layers import Dense,Dropout,Flatten

from keras.layers import Conv2D, MaxPooling2D

from keras.preprocessing.image import ImageDataGenerator

HEIGHT\_IMAGE=48

WIDTH\_IMAGE = 48

btch\_size=32

emotions = os.listdir('D:\\train')

value = [len(os.listdir(f'D:\\train\\{emotion}')) for emotion in emotions]

figure = plt.figure(figsize = (12, 5)) #Matplot library to plot graph

**# Plotting the bar plot**

plt.bar(emotions, value, color =’blue',

width = 0.6)

plt.xlabel("emotions")

plt.ylabel("Number of images")

plt.title("Train dataset overview")

plt.show()

emotions1 = os.listdir('D:\\test')

values = [len(os.listdir(f'D:\\test\\{emotion}')) for emotion in emotions1]

figure = plt.figure(figsize = (12, 5))

train\_dir='D:\\train'

test\_dir='D:\\test'

def images\_plot(dir\_image, top=5):

all\_dires\_image = os.listdir(dir\_image)

files\_image = [os.path.join(dir\_image, file) for file in all\_dires\_image][:5]

plt.figure(figsize=(12, 12))

for idx, path\_image in enumerate(files\_image):

plt.subplot(7, 7, idx+1)

img = plt.imread(path\_image)

plt.tight\_layout()

plt.imshow(img, cmap='gray')

**#Displaying various emotions with thie pictures**

print ('Angry: ')

print ()

images\_plot(test\_dir+'\\angry')

print ('Fear: ')

print ()

images\_plot(train\_dir+'\\fear')

print ('Happy: ')

print ()

images\_plot(train\_dir+'\\happy')

print ('sad: ')

print ()

images\_plot(train\_dir+'\\sad')

print ('Disgust: ')

print ()

images\_plot(train\_dir+'\\disgust')

print ('Neutral: ')

print ()

images\_plot(train\_dir+'\\neutral')

print ('Surprise: ')

print ()

images\_plot(train\_dir+'\\surprise')

**#Image Augmentation**

datagen\_train = ImagedataGenerator(

rescale=1. /255,

rotation\_range=20,

shear\_range=0.1,

zoom\_range=0.1,

horizontal\_flip=True,

fill\_mode='nearest',validation\_split=0.2) #Splitting training data 80% train,20% validation

**#Training Data**

generate\_train = datagen\_train.flow\_from\_directory(

train\_dir,

colr\_mode='grayscale',

targt\_size=(HEIGHT\_IMAGE, WIDTH\_IMAGE),

batch\_size=btch\_size,

class\_Mode='categorical',subset='training',

shuffle=True)

**#Valid Data**

generate\_validation = datagen\_train.flow\_from\_directory(

train\_dir,

colr\_mode='grayscale',

targt\_size=(HEIGHT\_IMAGE, WIDTH\_IMAGE),

batch\_size=btch\_size,

class\_Mode='categorical',subset='validation',

shuffle=False)

**#Testing Data**

generate\_test = ImagedataGenerator(rescale=1./255).flow\_from\_directory(

test\_dir,

colr\_mode='grayscale',

targt\_size=(HEIGHT\_IMAGE, WIDTH\_IMAGE),

batch\_size=btch\_size,

class\_Mode='categorical',

shuffle=False)

**# CNN Algorithm Implementation starts**

**#Implementing CNN algorithm**

import keras

model1 = Sequential()

model1.add (Conv2D (32, kernel\_size=(3, 3), input\_shpe=(48,48,1), activation='relu'))

model1.add (Conv2D (64, kernel\_size=(4, 4), activation='relu'))

model1.add (MaxPooling2D(pool\_size=(3, 3)))

model1.add (Dropout (0.2))

model1.add (Conv2D (128, kernel\_size=(4, 4), activation='relu'))

model1.add (MaxPooling2D(pool\_size=(3, 3)))

model1.add (Dropout (0.2))

model1.add (Conv2D (256, kernel\_size=(4, 4), activation='relu'))

model1.add (MaxPooling2D(pool\_size=(3, 3)))

model1.add (Dropout (0.2))

model1.add (Flatten ())

model1.add (Dense (512, activation='relu'))

model1.add (Dropout (0.2))

model1.add (Dense (7, activation='softmax'))

model1.compile(optimizer = 'Adam', loss='categorical\_crossentropy', metrics=['accuracy',keras.metrics.Preicision(), keras.metrics.Reecall()])

print (model1.summary())

Epochs=20

history=model1.fit(generate\_train, #Training Data

steps\_per\_epoch=generate\_train.n//generate\_train.batch\_size,

epochs=Epochs, #Epochs used 20

valid\_data=generate\_validation, #Validation Data

valid\_steps=generate\_validation.n//generate\_validation.batch\_size)

**#Loss metric visualization**

loss\_training = history.history['loss']

loss\_valid = history.history['val\_loss']

plt.rcParams['figure.figsize'] = [6, 6]

plt.style.use(['default'])

# Create Count of the number of epochs

count\_Epoch = range (1, len(loss\_training) + 1)

**# Plotting the graph**

plt.plot(count\_Epoch,loss\_train, 'y')

plt.plot(count\_Epoch,loss\_val, 'r')

plt.legend(['Training Loss', 'Valid Loss'])

plt.title('Training Loss and valid Loss')

plt.xlabel('Epoch')

plt.ylabel('trainLoss')

plt.show()

**#Accuracy visualization**

accuracy\_training = history.history['accuracy']

accuracy\_valid = history.history['val\_accuracy']

**# Create Count of the number of epochs**

count\_Epoch = range(1, len(accuracy\_training) + 1)

**#Plotting the graph**

plt.plot(count\_Epoch,accuracy\_training, 'y')

plt.plot(count\_Epoch,accuracy\_valid, 'r')

plt.title('Training accuracy and validation accuracy')

plt.legend(['Train Accuracy', 'Valid Accuracy'])

plt.xlabel('Epoch')

plt.ylabel('accuracy')

plt.ylim(top = 2)

plt.show()

model1.save('cnnf with 20 epochs.h5')

**#Loading model**

from keras.models import load\_model

**#Testing the model**

model1\_cnn = load\_model('cnnf with 20 epochs.h5', Compile=False)

**#Generate some batch of images**

Test\_img, Test\_lbl = generate\_test.\_\_next\_\_ ()

predict=model1\_cnn. predict(Test\_img)

predict = np.argmax(predict, axis=1)

Test\_labels = np.argmax(Test\_lbl, axis=1)

**#Knowing Test accuracy**

from sklearn import metrics

print ("Test accuracy of CNN = ", metrics.accuracy\_score(Test\_labels, predict))

from sklearn.metrics import classification\_report

#Classification report

report\_clsf = pd.dataFrame(classification\_report(Test\_labels, predict, output\_dict=True)).transpose()

clsf\_report.to\_csv('CNN\_exampleadam\_report1.csv', index= True)

clsf\_report

**#Checking model on test data to predict emotion**

import random

class\_labls=['angry','disgust', 'fear', 'happy','neutral','sad','surprise']

n=random.randint(0, Test\_img.shape[0] - 1)

image = Test\_img[n]

origin\_emotion = class\_labels[Test\_labels[n]]

predic\_emotion = class\_labels[predict[n]]

plt.imshow(image[:,:,0], cmap='gray')

plt.title("origi Emotion is:"+origin\_emotion+" predi Emotion: "+ predic\_emotion)

plt.show()

**#### SVM Algorithm Implementation starts**

**#Importing libraries from sklearn**

from keras.utils import to\_categorical

from sklearn.metrics import accuracy\_score

from sklearn.metrics import classification\_report, confusion\_matrix

import seaborn as sns

**#Reading dataset**

df=pd.read\_csv(r"C:\Users\Manas\Downloads\fer2013.csv")

df.head(5)

**#Trimming Data so that SVM algorithm can be implemented without any interruption and can take less space**

df\_data=df.drop(labels=range(10708,35887), axis=0)

df\_data.head(5)

df\_data['emotion'].value\_counts()

df\_data['Usage']. drop

df\_data1=df\_data.drop(columns='Usage')

df\_data1.head(5)

width, height=48,48

number\_classes=7

df\_data1['pixels'] =df\_data1['pixels']. apply(lambda pixel\_sequence: [int(pixel) for pixel in pixel\_sequence.split()])

X\_data=np.array(df\_data1['pixels'].tolist(),dtype='float32')

Y\_data=to\_categorical(df\_data1['emotion'],number\_classes)

X\_data.shape,Y\_data.shape

from sklearn.model\_selection import train\_test\_split

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X\_data,Y\_data,test\_size=0.2, random\_state=50)

X\_train.shape,X\_test.shape,y\_train.shape,y\_test.shape

**#Scaling data so that speed can be increased of SVM algorithm**

from sklearn import preprocessing

Train\_X=preprocessing.scale(X\_train)

Test\_X=preprocessing.scale(X\_test)

Train\_y=y\_train[:,-1]

y\_train.shape

Test\_y=y\_test[:,-1]

y\_test.shape

**#SVM algorithm Implementation with rbf kernel**

from sklearn.svm import SVC #Importing SVC classifier from sklearn library

model2=SVC (kernel='rbf',gamma='auto',C=0.1) #rbf kernel with C=0.1

model2.fit(Train\_X,Train\_y)

emotion\_prediction=model.predict(Test\_X)

from sklearn.metrics import accuracy\_score #Importing accuracy\_Score

print ("Test accuracy and rbf kernel", accuracy score(Test\_y,emotion\_prediction))

**#Plotting cm**

from sklearn.metrics import plot\_confusion\_matrix #Importing confusion matrix

import mlxtend

from mlxtend.plotting import plot\_decision\_regions

matrix =plot\_confusion\_matrix(model2,Test\_X,Test\_y,cmap=plt.cm.Blues,normalize='true')

plt.title('cm of test data with linear kernel') #Giving title

plt.show(matrix) #Plotting the matrix

plt.show() #Showing matrix

**#Scatterplot of Train data**

supporting\_vectors=model2.support\_vectors\_

plt.scatter(Train\_X[:,0],Train\_y[:,1])

plt.scatter(supporting\_vectors[:,0],supporting\_vectors[:,1],color='yellow')

plt.title('Scatterplot of Train data with linear kernel')

plt.xlabel('x1')

plt.ylabel('x2')

plt.show()

**#Scatterplot of Test data with rbf kernel**

supporting\_vectors=model2.support\_vectors\_

plt.scatter(Test\_X[:,0],Test\_y[:,1])

plt.scatter(supporting\_vectors[:,0],supporting\_vectors[:,1],color='yellow')

plt.title('Scatterplot of Test data with rbf kernel')

plt.xlabel('x1')

plt.ylabel('x2')

plt.show()

**y**

**#Classif report**

report\_clsf = pd.DataFrame(classification\_report(Test\_y, emotion\_prediction, output\_dict=True)). transpose ()

report\_clsf.to\_csv('SVM\_examplerbf\_report1.csv', index= True)

clsf\_report

**#SVM algorithm implementation with linear kernel**

from sklearn.svm import SVC

model1=SVC (kernel='linear', gamma='auto',C=0.1)

model1.fit(Train\_X,Train\_y)

emotion\_predict1=model1.predict(Test\_X)

from sklearn.metrics import accuracy\_score

print ("Test Accuracy with linear kernel",accuracy\_score(y\_test, emotion\_predict1))

**#Scatterplot of Test data with linear kernel**

supporting\_vectors=model2.support\_vectors\_

plt.scatter(Test\_X[:,0],Test\_y[:,1])

plt.scatter(supporting\_vectors[:,0],supporting\_vectors[:,1],color='yellow')

plt.title('Scatterplot of Test data with linear kernel')

plt.xlabel('x1')

plt.ylabel('x2')

plt.show()

**#Classificaion report**

clsf\_report = pd.DataFrame(classification\_report(Test\_y, emotion\_predict1, output\_dict=True)). transpose ()

clsf\_report.to\_csv('SVM\_examplelinear\_report1.csv', index= True)

clsf\_report