Calendar

Description automatically generated Text, logo

Description automatically generated

**VIDEO CONFERENCING MEET ANALYTICS USING FACIAL EMOTION RECOGNITION(FER)**

## A PROJECT REPORT

*Submitted by*

**AFRIDHA.M (170701007)**

**ANVITA ASHOKKUMAR (170701018)**

**CINDRELLA.S.K (170701027)**

***in partial fulfilment for the award of the degree***

***of***

**BACHELOR OF ENGINEERING**

**IN**

**ELECTRONICS AND COMMUNICATION ENGINEERING**

**SRI VENKATESWARA COLLEGE OF ENGINEERING**

**(An Autonomous Institution; Affiliated to Anna University, Chennai-600 025)**

# ANNA UNIVERSITY:CHENNAI 600 025

## MAY 2021

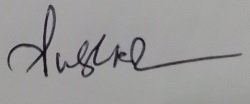
**SRI VENKATESWARA COLLEGE OF ENGINEERING**

**(An Autonomous Institution; Affiliated to Anna University, Chennai -600 025)**

**ANNA UNIVERSITY, CHENNAI – 600 025**

**BONAFIDE CERTIFICATE**

Certified that this project report **“VIDEO CONFERENCING MEET ANALYTICS USING FACIAL EXPRESSION RECOGNITION (FER)”** is the bonafide work of **“AFRIDHA.M (170701007), ANVITA ASHOKKUMAR (170701018), CINDRELLA.S.K (170701027)”** who carried out the project work under our supervision.

**SIGNATURE SIGNATURE**

Ms K.S.Subhashini M.E Mr.R.Prasanna

**INTERNAL SUPERVISOR EXTERNAL SUPERVISOR**

Asst Professor Project Manager

Department of Electronics and Mentormonks

Communication Engineering



**SIGNATURE**

Dr. S.Muthukumar , M.E., Ph.D

**HEAD OF DEPARTMENT**

Department of Electronics and Communication Engineering

Submitted for the project viva-voce examination held on 27/05/2021

**INTERNAL EXAMINER EXTERNAL EXAMINER**

**ACKNOWLEDGEMENT**

We express our sincere gratitude to **Dr. S. Ganesh Vaidyanathan, Ph.D.,** Principal, Sri Venkateswara College of Engineering, for being the source of inspiration throughout our study in the college.

We thank **Dr. S. Muthukumar, Ph.D.,** Professor, Head of the Department, Electronics and Communication Engineering, for his permission and encouragement to carry out this project.

We are extremely grateful to **Ms.K.S.Subhashini,** Professor, Department of Electronics and Communication Engineering, for her constant encouragement and support for the successful completion of this project.

We wish to thank our project coordinators **Mrs. K. S. Subhashini, M.E.**, Assistant Professor**, Mr. S.P. Sivagnana Subramanian, M.E.,** Assistant Professor, and **Dr.S.Sudha** for their valuable inputs and for encouraging us throughout this project.

This report would not have been possible without the contribution of many Industry resource persons. **Our sincere gratitude** to **Mr.R.Prasanna**, Project Manager, **MENTORMONKS,** our External Guide for his thorough Technical support and constant supervision which personally contributed immensely to identify our potential. We also thank him for his guidance which was a remarkable force that enabled us to complete the internship program.

**ABSTRACT**

Video conferencing and video calls are popular tools for conducting two-way video and audio communications over long distances. This technology has been developing rapidly due to the emergence of high-speed networking solutions, inexpensive hardware components, and the deployment of cellular networks. It is also important to know if the conference was a success, know if established goals and objectives were reached, and an important aspect is audience opinion. While evaluating audience opinion, facial emotions can bring relevant information. Analyzing the facial emotions of participants attending the conference requires one or more people who pay attention to the public and take notes of their reactions during the conference. If a conference is given online, it will be impossible to know the audience's emotions. We propose a system that will analyze the facial expressions of people who attend a meeting by analyzing the image from a camera. This system will be able to identify faces, following them even if they are moving, and generate a report of facial expressions at the end of the meeting.

**TABLE OF CONTENTS**

|  |  |  |
| --- | --- | --- |
| **CHAPTER** | **TITLE** | **PAGE NO** |
|  | **ACKNOWLEDGEMENT** | iii |
|  | **ABSTRACT** | iv |
|  | **LIST OF FIGURES** | vii |
|  | **LIST OF SYMBOLS** | ix |
| **1** | **INTRODUCTION** | 1 |
|  | 1.1 Motivation | 1 |
|  | 1.2 Facial Emotion Recognition | 2 |
|  | 1.3 Digital Image Processing | 3 |
|  | 1.4 Introduction to Machine Learning | 4 |
|  | 1.5 Machine learning Methods | 5 |
|  | 1.6 Introduction to Deep Learning | 6 |
|  | 1.7 CNN: An Overview | 7 |
|  | 1.8 Convolution Layer | 7 |
| **2** | **LITERATURE SURVEY** | 12 |
|  | 2.1 Background | 12 |
|  | 2.2 Facial Emotion Task | 12 |
|  | 2.3 Facial Emotion Analysis | 13 |
| **3** | **PROPOSED SYSTEM** | 16 |
|  | 3.1 Problem Definition | 16 |
|  | 3.2 Proposed Solution | 16 |
|  | 3.3 Emotion Recognition using Xception Model | 17 |
|  | 3.4 Block Diagram | 19 |
|  | 3.5 Objective of the Proposed Solution | 19 |
| **4** | **IMPLEMENTATION AND RESULT** | 20 |
|  | 4.1 Exploring the Dataset | 20 |
|  | 4.2 Methodology | 23 |
|  | 4.3 Training and Testing Phase | 26 |
|  | 4.4 Classification and Evaluation | 27 |
|  | 4.5 Experiment and Results | 32 |
| **5** | **REQUIREMENTS** | 37 |
|  | 5.1 Software Requirements | 37 |
| **6** | **CONCLUSION AND FUTURE WORK** | 41 |
|  | **REFERENCES** | 42 |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

**LIST OF FIGURES**

|  |  |  |
| --- | --- | --- |
| **FIG NO.** | **TITLE** | **PAGE NO.** |
| 1.1 | THE SOURCES OF FACIAL EXPRESSIONS | 2 |
| 1.2 | BASIC CNN | 8 |
| 1.3 | MAX POOLING | 9 |
| 1.4 | FULLY CONNECTED LAYER | 11 |
| 3.1 | XCEPTION MODEL | 17 |
| 3.2 | OVERALL ARCHITECTURE OF XCEPTION | 18 |
| 3.3 | BLOCK DIAGRAM | 19 |
| 4.1 | FER 2013 DATASET | 20 |
| 4.2 | HAAR CASCADE CLASSIFIER | 25 |
| 4.3 | HAAR FEATURES SELECTED | 26 |
| 4.4 | TRAINING PHASE | 27 |
| 4.5 | TESTING PHASE | 27 |
| 4.6 | CONFUSION MATRIX | 28 |
| 4.7 | ACCURACY GRAPH | 29 |
| 4.8 | LOSS GRAPH | 30 |
| 4.9 | NETWORK STRUCTURE | 33 |
| 4.10 | TRAINED MODEL | 33 |
| 4.11 | ACCURACY AND LOSS GRAPH | 34 |
| 4.12 | EMOTION RECOGNITION IMPLEMENTATION | 34 |
| 5.1 | ANACONDA AND PYTHON | 37 |
| 5.2 | TENSORFLOW | 38 |
| 5.3 | KERAS | 39 |
| 5.4 | OpenCV | 39 |
| 5.5 | NumPY | 40 |

**LIST OF SYMBOLS**

|  |  |
| --- | --- |
| 1D | One Dimensional |
| 2D | Two Dimensional |
| AI | Artificial Intelligence |
| AR | Augmented Reality |
| AU | Action Units |
| BN | Bayesian Networks |
| CNN | Convolutional Neural Network |
| CV | Computer Vision |
| DCNN | Deep Convolutional Neural Network |
| FACS | Facial Action Coding System |
| FER | Facial Emotion Recognition |
| FN | False Negative |
| FP | False Positive |
| FPR | False Positive Rate |
| GPU | Graphics Processing Unit |
| HCI | Human Computer Interaction |
| IDLE | Integrated Development and Learning  Environment |
| IEEE | The Institute of Electrical and Electronics Engineers |
| IOT | Internet of Things |
| LDA | Linear Discriminant Analysis |
| NN | Neural Network |
| RELU | Rectified Linear Unit |
| ROC | Receiver Operating Characteristics |
| ROI | Region of Interest |
| SVM | Support Vector Machine |
| TN | True Negative |
| TP | True Positive |
| TPR | True Positive Rate |
| VJD | Viola Jones Detector |
| VR | Virtual Reality |

**CHAPTER 1**

**INTRODUCTION**

With advances in computing and telecommunications technologies, digital images and videos are playing key roles in the present information era. The human face is an important biometric object in image and video databases of surveillance systems. Face recognition has a critical role in biometric systems and is attractive for numerous applications including visual surveillance and security. Because of the general public acceptance of face images on various documents, face recognition has a great potential to become the next-generation biometric technology of choice. Face images are also the only biometric information available in some legacy databases and international terrorist watch-lists and can be acquired even without subjects’ cooperation.

Though there has been a great deal of progress in face detection and recognition in the last few years, many problems remain unsolved. Research on face detection must confront many challenging problems, especially when dealing with outdoor illumination, pose variation with large rotation angles, low image quality, low resolution, occlusion, and background changes in complex real-life scenes.

* 1. **MOTIVATION**

The vision of a digital twin as stated by Prof. El Saddik is a digital replication of a living or non-living physical entity. By bridging the physical and the virtual worlds, data is transmitted seamlessly, allowing the virtual entity to exist simultaneously with the physical entity. A digital twin facilitates the means to monitor, understand, and optimize the functions of the physical entity and provides continuous feedback to improve quality of life and wellbeing. A digital twin is the convergence of several technologies such as AI, AR/VR, and Haptics, IoT, Cybersecurity, and Communication Networks.

Diagram

Description automatically generated

**FIGURE 1.1 THE SOURCES OF FACIAL EXPRESSIONS**

One component of the digital twin vision is so-called affective computing or the task of recognizing the emotion of a given subject in real-time. Affect computing has been an emerging trend in the last decade, as it meets the goal of intelligent human-computer interaction (HCI), which is to seek efficient communication between humans and machines. Fasel et al. noted that emotional voice, gestures, facial expressions, etc., constitute the factors of human emotions. Facial expressions, among these factors mentioned above, play the most critical role in affect analysis. Facial changes in facial expressions are responses to a person’s internal emotional states, intentions, or social communications. Darwin et al. established facial expression analysis as a research field in 1872. Since then, facial expression recognition (FER) has received a great deal of attention and has been an active research topic across a variety of disciplines, such as biology, neuroscience, psychology, and computer vision. Especially in computer vision, for its impact and prominent potentiality, automatic FER has been growing in an extensive range of applications, e.g., HCI, biometric identification, surveillance and security, intensive care monitoring, aerial image analysis, driver state surveillance, and human entertainment industry and virtual reality.

**1.2 FACIAL EMOTION RECOGNITION**

Facial emotion recognition is the process of detecting human emotions from facial expressions. The human brain recognizes emotions automatically, and software has now been developed that can recognize emotions as well. This technology is becoming more accurate all the time and will eventually be able to read emotions as well as our brains do. AI can detect emotions by learning what each facial expression means and applying that knowledge to the new information presented to it. Emotional artificial intelligence, or emotion AI, is a technology that is capable of reading, imitating, interpreting, and responding to human facial expressions and emotions.

**1.3 DIGITAL IMAGE PROCESSING**

The identification of objects in an image would probably start with image processing techniques such as noise removal, followed by (low-level) feature extraction to locate lines, regions, and possibly areas with certain textures.

The clever bit is to interpret collections of these shapes as single objects, e.g., cars on a road, boxes on a conveyor belt, or cancerous cells on a microscope slide. One reason this is an AI problem is that an object can appear differently when viewed from different angles or under different lighting. Another problem is deciding what features belong to what object and which are background or shadows etc. The human visual system performs these tasks mostly unconsciously, but a computer requires skillful programming and lots of processing power to approach human performance. Manipulating data in the form of an image through several possible techniques. An image is usually interpreted as a two-dimensional array of brightness values and is most familiarly represented by such patterns as those of a photographic print, slide, television screen, or movie screen. An image can be processed optically or digitally with a computer. To digitally process an image, it is first necessary to reduce the image to a series of numbers that can be manipulated by the computer. Each number representing the brightness value of the image at a particular location is called a picture element, or pixel. A typical digitized image may have 512 × 512 or roughly 250,000 pixels, although much larger images are becoming common. Once the image has been digitized, three basic operations can be performed on it in the computer. For a point operation, a pixel value in the output image depends on a single-pixel value in the input image. For local operations, several neighboring pixels in the input image determine the value of an output image pixel. In a global operation, all of the input image pixels contribute to an output image pixel value. These operations, taken singly or in combination, are how the image is enhanced, restored, or compressed. An image is enhanced when it is modified so that the information it contains is clearer, but enhancement can also include making the image more visually appealing.

**1.4 INTRODUCTION TO MACHINE LEARNING**

Machine learning is a subfield of artificial intelligence (AI). The goal of machine learning generally is to understand the structure of data and fit that data into models that can be understood and utilized by people. Although machine learning is a field within computer science, it differs from traditional computational approaches. In traditional computing, algorithms are sets of explicitly programmed instructions used by computers to calculate or problem solve. Machine learning algorithms instead allow for computers to train on data inputs and use statistical analysis to output values that fall within a specific range. Because of this, machine learning facilitates computers in building models from sample data to automate decision-making processes based on data inputs. Any technology user today has benefitted from machine learning. Facial recognition technology allows social media platforms to help users tag and share photos of friends. Optical character recognition technology converts images of text into movable types. Recommendation engines, powered by machine learning, suggest what movies or television shows to watch next based on user preferences. Self-driving cars that rely on machine learning to navigate may soon be available to consumers. Machine learning is a continuously developing field. Because of this, there are some considerations to keep in mind as you work with machine learning methodologies or analyze the impact of machine learning processes.

**1.5 MACHINE LEARNING METHODS**

In machine learning, tasks are generally classified into broad categories. These categories are based on how learning is received or how feedback on the learning is given to the system developed. Two of the most widely adopted machine learning methods are supervised learning which trains algorithms based on example input and output data that is labeled by humans, and unsupervised learning which provides the algorithm with no labeled data to allow it to find structure within its input data.

**1.5.1 SUPERVISED LEARNING**

In supervised learning, the computer is provided with example inputs that are labeled with their desired outputs. The purpose of this method is for the algorithm to be able to “learn” by comparing its actual output with the “taught” outputs to find errors and modify the model accordingly. Supervised learning, therefore, uses patterns to predict label values on additional unlabelled data. For example, with supervised learning, an algorithm may be fed data with images of sharks labeled as fish and images of oceans labeled as water. By being trained on this data, the supervised learning algorithm should be able to later identify unlabelled shark images as fish and unlabelled ocean images as water. A common use case of supervised learning is to use historical data to predict statistically likely future events. It may use historical stock market information to anticipate upcoming fluctuations or be employed to filter out spam emails. In supervised learning, tagged photos of dogs can be used as input data to classify untagged photos of dogs.

**1.5.2 UNSUPERVISED LEARNING**

In unsupervised learning, data is unlabelled, so the learning algorithm is left to find commonalities among its input data. As unlabelled data are more abundant than labeled data, machine learning methods that facilitate unsupervised learning are particularly valuable. The goal of unsupervised learning may be as straightforward as discovering hidden patterns within a dataset, but it may also have a goal of feature learning, which allows the computational machine to automatically discover the representations that are needed to classify raw data. Unsupervised learning is commonly used for transactional data. With this data fed into an unsupervised learning algorithm, it may be determined that women of a certain age range who buy unscented soaps are likely to be pregnant, and therefore a marketing campaign related to pregnancy and baby products can be targeted to this audience to increase their number of purchases. Unsupervised learning is often used for anomaly detection including for fraudulent credit card purchases, and recommender systems that recommend what products to buy next. In unsupervised learning, untagged photos of dogs can be used as input data for the algorithm to find likenesses and classify dog photos together**.**

**1.6 INTRODUCTION TO DEEP LEARNING**

Deep learning (also known as deep structured learning or hierarchical learning) is part of a broader family of machine learning methods based on learning data representations, as opposed to task-specific algorithms. Learning can be supervised, semi-supervised, or unsupervised. Deep learning models are loosely related to information processing and communication patterns in a biological nervous system, such as neural coding that attempts to define a relationship between various stimuli and associated neuronal responses in the brain. Deep learning architectures such as deep neural networks, deep belief networks, and recurrent neural networks have been applied to fields including computer vision, speech recognition, natural language processing, audio recognition, social network filtering, machine translation, bioinformatics, and drug design, where they have produced results comparable to and in some cases superior to human experts.

**1.7 CNN: AN OVERVIEW**

Convolution is a mathematical operation that involves a combination of two functions to produce a third function. CNN is a type of deep learning model for processing data that has a grid pattern, such as images, which is inspired by the organization of the animal visual cortex and designed to automatically and adaptively learn spatial hierarchies of features, from low- to high-level patterns. In CNN, the convolution is performed on the input data with the use of a filter to produce a feature map. In the context of CNNs, a convolutional layer (called filter or kernel) is applied to the input data to then produce a feature map. CNN is a mathematical construct that is typically composed of three types of layers (or building blocks): convolution, pooling, and fully connected layers.

**1.8 CONVOLUTION LAYER**

A convolution layer is a fundamental component of the CNN architecture that performs feature extraction, which typically consists of a combination of linear and nonlinear operations, i.e., convolution operation and activation function.



**FIGURE 1.2 BASIC CNN**

**1.8.1 CONVOLUTION**

Convolution is a specialized type of linear operation used for feature extraction, where a small array of numbers, called a kernel, is applied across the input, which is an array of numbers, called a tensor. An element-wise product between each element of the kernel and the input tensor is calculated at each location of the tensor and summed to obtain the output value in the corresponding position of the output tensor, called a feature map. This procedure is repeated applying multiple kernels to form an arbitrary number of feature maps, which represent different characteristics of the input tensors; different kernels can, thus, be considered as different feature extractors.

**1.8.2 ACTIVATION FUNCTION**

The outputs of a linear operation such as convolution are then passed through a nonlinear activation function. Although smooth nonlinear functions, such as sigmoid or hyperbolic tangent (tanh) function, were used previously because they are mathematical representations of a biological neuron behaviour, the most common nonlinear activation function used presently is the rectified linear unit (ReLU), which simply computes the function: f(*x*) = max(0, *x*).

**1.8.3 POOLING LAYER**

A pooling layer provides a typical down-sampling operation which reduces the in-plane dimensionality of the feature maps to introduce a translation invariance to small shifts and distortions and decrease the number of subsequent learnable parameters. It is of note that there is no learnable parameter in any of the pooling layers, whereas filter size, stride, and padding are hyperparameters in pooling operations, like convolution operations.



**FIGURE 1.3 MAX POOLING**

**1.8.4 MAX POOLING**

The most popular form of pooling operation is max pooling, which extracts patches from the input feature maps, outputs the maximum value in each patch, and discards all the other values. A max pooling with a filter of size 2 × 2 with a stride of 2 is commonly used in practice. This down-samples the in-plane dimension of feature maps by a factor of 2. Unlike height and width, the depth dimension of feature maps remains unchanged.

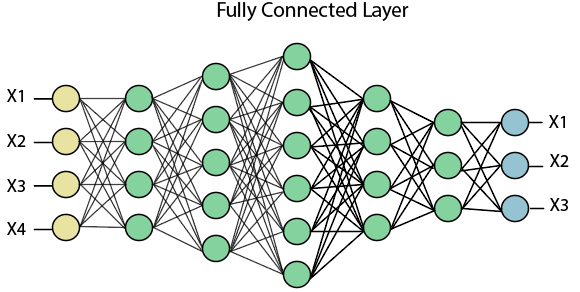
**1.8.5 GLOBAL AVERAGE POOLING**

A global average pooling performs an extreme type of down-sampling, where a feature map with the size of height × width is down-sampled into a 1 × 1 array by simply taking the average of all the elements in each feature map, whereas the depth of feature maps is retained. This operation is typically applied only once before the fully connected layers. The advantages of applying global average pooling are as follows:

* 1. Reduces the number of learnable parameters and
  2. Enables the CNN to accept inputs of variable size.

**1.8.6 FULLY CONNECTED LAYER**

The output feature maps of the final convolution or pooling layer are typically flattened, i.e., transformed into a one-dimensional (1D) array of numbers (or vector), and connected to one or more fully connected layers, also known as dense layers, in which every input is connected to every output by a learnable weight. Once the features extracted by the convolution layers and downs-sampled by the pooling layers are created, they are mapped by a subset of fully connected layers to the final outputs of the network, such as the probabilities for each class in classification tasks. The final fully connected layer typically has the same number of output nodes as the number of classes. Each fully connected layer is followed by a nonlinear function, such as ReLU.



**FIGURE 1.4 FULLY CONNECTED LAYER**

**CHAPTER 2**

**LITERATURE SURVEY**

* 1. **BACKGROUND**

A Facial expression is the visible manifestation of the affective state, cognitive activity, intention, personality, and psychopathology of a person and plays a communicative role in interpersonal relations. Human facial expressions can be easily classified into 7 basic emotions: happy, sad, surprise, scared, anger, disgust, and neutral. Our facial emotions are expressed through the activation of specific sets of facial muscles. Automatic recognition of facial expressions can be an important component of natural human-machine interfaces; it may also be used in behavioral science and clinical practice. It has been studied for a long period and obtaining progress in recent decades. Though much progress has been made, recognizing facial expressions with a high accuracy remains to be difficult due to the complexity and varieties of facial expressions.

* 1. **FACIAL EXPRESSION RECOGNITION TASK**

According to Lopes et al., the two main branches of facial expression recognition systems are as follows: those addressing static images and those addressing dynamic image sequences. Systems that work with static images consider only one still image at a time (frame-by-frame) and do not use temporal information. In contrast, those involving dynamic image sequences encoding a range of frames within a temporal window as an individual concentrate more on analyzing temporal variation. This work adopts the frame-based scheme. Automatic FER systems generally receive the two kinds of expected input and output one of the seven basic universal emotions (i.e., angry, disgust, scared, happiness, sadness, surprise, and neutral) that were classified by Ekman in 1975 in a cross-cultural study on the existence of “universal categories of emotional expressions”. In 1978, Ekman and Friesen developed the Facial Action Coding System (FACS) to describe observable facial muscle movements as action units (AU). This system taxonomizes these basic emotions by decomposing each facial expression into core AU and has been widely applied to FER tasks, providing a powerful tool for feature extraction.

* 1. **FACIAL EXPRESSION ANALYSIS**

The automatic facial expression analysis typically involves three steps:

1. Face acquisition,
2. Facial feature extraction and representation, and
3. Facial expression recognition (classification).
   * 1. **FACE ACQUISITION**

Face acquisition refers to detecting the face region in a frame (face detection). One of the most widely used face detectors was proposed by Viola and Jones and is termed the VJ detector. Instead of working only with image intensities (which consumes substantial computational power), Papageorgiou et al. developed a framework based on Haar wavelet representation in 1998. Later, in 2001, Viola and Jones further developed this idea by proposing the Haar-like features that represent the changes of texture or edges of special facial regions and can be operated much faster than pixels in systems. The features compute the differences between the sums of pixels within those rectangular areas. This kind of feature is then organized using summed-area tables (called integral images) and an algorithm (called a cascade of classifiers) to speed up computation. Based on these three key points, the VJ detector can perform robust and efficient face detection in real-time. However, it concentrates only on frontal faces.

* + 1. **FACIAL FEATURE EXTRACTION**

Facial features in facial expression analysis, as mentioned in, can be divided into two types: intransient and transient. Intransient facial features refer to those that are always present in the face (e.g., eyes, mouth, nose, and eyebrows) but may be deformed when people display facial expressions. Transient facial features, on the other hand, represent wrinkles or other texture changes, especially in regions surrounding the eyes and the mouth, that appear with facial expressions. One way to classify the approaches applied to facial feature extraction and representation is into geometric-based (shape-based) or appearance-based categories. Geometric-based (shape-based) representation considers the shape information (intransient features, e.g., facial points or locations of eyebrows, eyes, the mouth, the nose) explicitly and ignores the texture. The face geometry is represented in the feature vectors extracted from the shape information. The appearance-based representation uses the intensity values or the pixels that refer to the textural changes, such as facial wrinkles. The facial features can be extracted from the entire face or particular facial regions. Notably, shape-based representation is typically vulnerable to illumination variation. It is challenging to maintain high accuracy and reliable facial component detection in real-time systems due to the varied environments.

**2.3.3 EMPIRICAL CLASSIFIERS FOR FER**

After the facial features are available, the last step for facial expression analysis is the classification of those basic emotions, which is referred to as facial expression recognition. Three stages of the training process in FER consists of

* Feature learning
* Feature selection
* Classifier construction

Feature learning extracts features associated with facial expression. For feature selection, the ideal deeply learned features to be selected maximize interclass differences (to be separable) and minimize intraclass variations (to be discriminative). Ultimately, a classifier is constructed to infer the facial expression for each class. These kinds of classifiers leverage the extracted and selected hand-crafted or learned features related to facial expressions to execute the classification.

**CHAPTER 3**

**PROPOSED SOLUTION**

**3.1 PROBLEM DEFINITION**

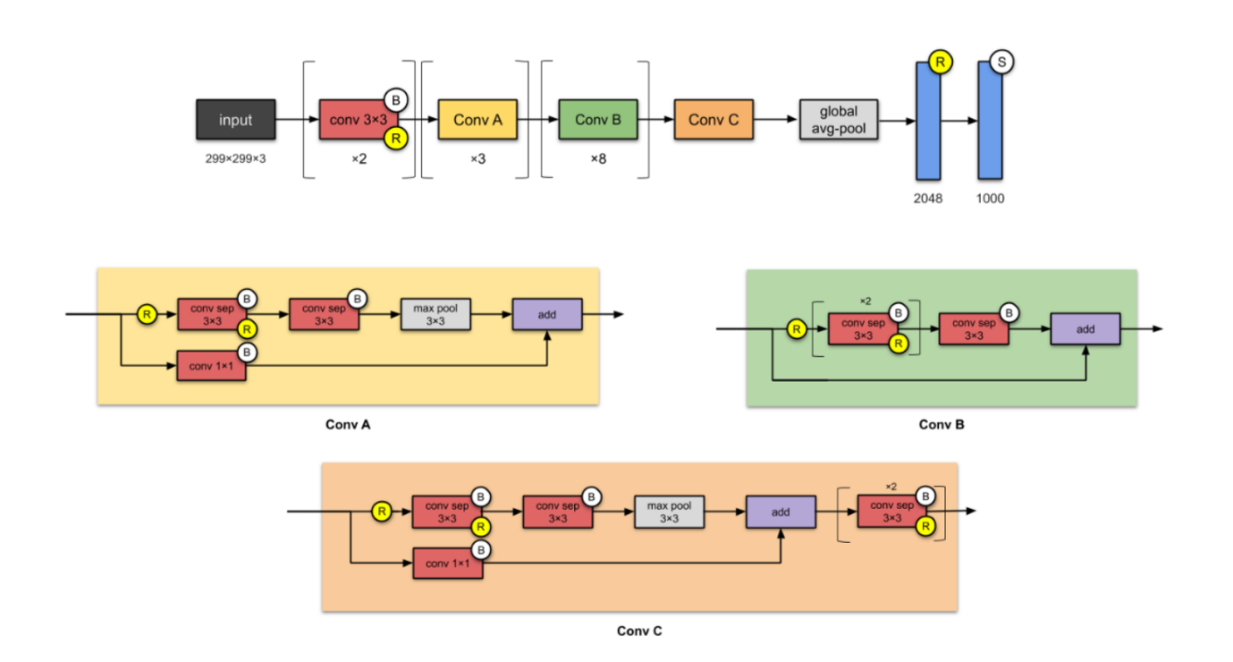
Human emotions and intentions are expressed through facial expressions and deriving an efficient and effective feature is the fundamental component of the facial expression system. Facial expressions convey non-verbal cues, which play an important role in interpersonal relations. Automatic recognition of facial expressions can be an important component of natural human-machine interfaces; it may also be used in behavioral science and clinical practice. An automatic Facial Expression Recognition system needs to solve the following problems: detection and location of faces in a cluttered scene, facial feature extraction, and facial expression classification.

**3.2 PROPOSED SOLUTION**

To develop a system that can identify the facial expressions during a video conference meeting, a “Haar-cascade Frontalface” filter has been implemented using OpenCV. This filter is based on the value of the pixels on the image in grayscale to find faces. This filter counts how many black and white pixels can be found in each region and decides by windowing whether there’s a face or not. A convolutional neural network is used to identify facial expressions. This model is based on the FER-2013 dataset, which contains images in a grayscale of faces classified as {angry, disgusted, happy, sad, surprised, neutral}.

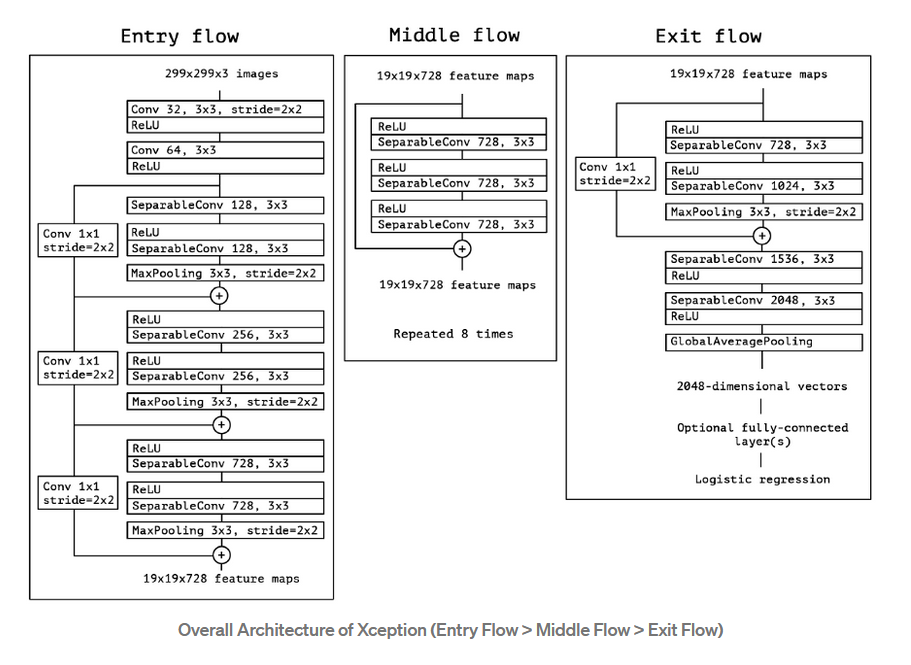
**3.3 EMOTION RECOGNITION USING XCEPTION MODEL**

Xception is a convolutional neural network that is 71 layers deep. Xception stands for Extreme version of Inception. Xception Model is proposed by Francois Chollet. Xception is an extension of the inception Architecture which replaces the standard Inception modules with depthwise separable convolutions.



**FIGURE 3.1 XCEPTION MODEL**

Xception slightly outperforms Inception v3 on the ImageNet dataset, and vastly outperforms it on a larger image classification dataset with 17,000 classes. Most importantly, it has the same number of model parameters as Inception, implying a greater computational efficiency.



**FIGURE 3.2 OVERALL ARCHITECTURE OF XCEPTION**

Xception is an adaptation from Inception, where the Inception modules have been replaced with depth-wise separable convolutions. It also has roughly the same number of parameters as Inception-v1. The Xception architecture has 36 convolutional layers forming the feature extraction base of the network. The 36 convolutional layers are structured into 14 modules, all of which have linear residual connections around them, except for the first and last modules. In short, the Xception architecture is a linear stack of depth-wise separable convolution layers with residual connections. This makes the architecture very easy to define and modify; it takes only 30 to 40 lines of code using a high-level library such as Keras or TensorFlow-Slim, not unlike an architecture such as VGG-16, but rather unlike architectures such as Inception V2 or V3 which are far more complex to define. An open-source implementation of Xception using Keras and TensorFlow is provided as part of the Keras Applications module.

**3.4 BLOCK DIAGRAM**

Diagram

Description automatically generated

**FIGURE 3.3 BLOCK DIAGRAM**

**3.5 OBJECTIVE OF THE PROPOSED SOLUTION**

* To propose a training method for CNN. The employment of this method is to obtain discriminative deep features that achieve accuracy superior to that of current state-of-the-art methods on two publicly available datasets.
* To design and implement a system that can realize facial expression recognition from a webcam, achieving real-time operation on images while combining efficiency and accuracy.
* To provide a comprehensive evaluation configuration for a specific dataset for better comparison with the state-of-the-art system that does not provide a consistent evaluation methodology.

**CHAPTER 4**

**IMPLEMENTATION AND RESULT**

**4.1 EXPLORING THE DATASET**

The FER-2013 dataset was presented in the sub-challenge/competition Facial Expression Recognition Challenge of Challenges in Representation Learning in the ICML 2013 workshop, which was hosted by Kaggle. The dataset itself was retrieved from the Internet using the Google image search API consisting of 36,887 images, where 28,709 are used as training data, 3589 are used for public validation, and another 3589 are used for the private test. The dataset contains 48 × 48 pixel low-resolution grayscale images across seven basic facial expression classes. Because of the label noise and the variety of real-world conditions collected from the Internet, the FER-2013 has become by far one of the more widely used and most challenging spontaneous datasets for FER, the human recognition rate of which is approximately 68%.

A picture containing text, person, posing, dressed

Description automatically generated

**FIGURE 4.1 FER 2013 DATASET**

**4.1.1 IMBALANCED CLASSIFICATION**

An imbalanced classification problem is an example of a classification problem where the distribution of examples across the known classes is biased or skewed. The distribution can vary from a slight bias to a severe imbalance where there is one example in the minority class for hundreds, thousands, or millions of examples in the majority class or classes. Imbalanced classifications pose a challenge for predictive modeling as most of the machine learning algorithms used for classification were designed around the assumption of an equal number of examples for each class. This results in models that have poor predictive performance, specifically for the minority class. This is a problem because typically, the minority class is more important, and therefore the problem is more sensitive to classification errors for the minority class than the majority class.

**4.1.2 PROBLEMS DUE TO CLASS IMBALANCE**

The number of examples that belong to each class may be referred to as the class distribution. Imbalanced classification refers to a classification predictive modeling problem where the number of examples in the training dataset for each class label is not balanced. That is, where the class distribution is not equal or close to equal and is instead biased or skewed. For example, we may collect measurements of flowers and have 80 examples of one flower species and 20 examples of a second flower species, and only these examples comprise our training dataset. This represents an example of an imbalanced classification problem.

**4.1.3 CAUSES OF CLASS IMBALANCE**

The imbalance to the class distribution in an imbalanced classification

predictive modeling problems may have many causes. There are perhaps two main groups of causes for the imbalance we may want to consider; they are data sampling and properties of the domain. The Class imbalance may be due to the way the examples were collected or sampled from the problem domain. This might involve biases introduced during data collection, and errors made during data collection.

• Biased Sampling.

• Measurement Errors.

For example, examples were collected from a narrow geographical region, or slice of time, and the distribution of classes may be quite different or perhaps even collected differently. Errors may have been made when collecting the observations. One type of error might have been applying the wrong class labels to many examples. Alternately, the processes or systems from which examples were collected may have been damaged or impaired to cause the imbalance. Often in cases where the imbalance is caused by a sampling bias or measurement error, the imbalance can be corrected by improved sampling methods, and/or correcting the measurement error. This is because the training dataset is not a fair representation of the problem domain that is being addressed. The imbalance might be a property of the problem domain. For example, the natural occurrence or presence of one class may dominate other classes. This may be because the process that generates observations in one class is more expensive in time, cost, computation, or other resources. As such, it is often infeasible or intractable to simply collect more samples from the domain to improve the class distribution. Instead, a model is required to learn the difference between the classes.

**4.1.4 DATA AUGMENTATION**

Data augmentation is often employed during the training of the CNN since the process itself incorporates a large quantity of data. In the training scheme of this work, the cropped faces are first distorted with a lightweight library in TensorFlow before feeding them into the CNN. Each cropped face is randomly sampled by one of the distorted bounding boxes. The area of the sampled patch is [0.85, 1] of the original supplied image, and the number of generated images is as high as 100. After the sampling step, the sampled patches are rescaled to 48 × 48 pixels since the CNN training input must be square. The reason for the selection of this rescale parameter 48 × 48 is to remain consistent with the resolution of the FER-2013 dataset. Furthermore, after rescaling, the images are also randomly flipped horizontally with a probability of 0.5 to have two times more data. Finally, normalization is performed for faster convergence. The data are normalized into the range of [−1, 1] instead of the typical [0, 1] because the activation function ReLU, which is max(0, x), works better when negative values are also provided.

**4.2 METHODOLOGY**

**4.2.1 PREPROCESSING**

In the processing of FER, some interference may arise that influences the effectiveness of feature extraction due to the complexity of the environment and the various performance of the shooting equipment during data acquisition.

Pre-processing of expression images can vary depending on factors such as the performance of the acquisition equipment or changes in illustration conditions. This issue typically manifests in the images containing different levels of noise and the average pixel intensities of different images showing various brightness contrast. Thus, it is necessary to perform data pre-processing, the general purpose of which is to eliminate noise and to normalize and centralize the gray value of the image to provide a solid foundation for subsequent classification and identification. However, extensive image pre-processing may require a large run-time cost, which threatens real-time capability. In our work, we perform a minimal amount of pre-processing while maintaining accuracy. Image restoration is the operation of taking a corrupted/noisy image and estimating the clean original image. Corruption may come in many forms such as motion blur, noise, and camera misfocus. Image restoration is different from image enhancement in that the latter is designed to emphasize features of the image that make the image more pleasing to the observer, but not necessarily to produce realistic data from a scientific point of view. Image enhancement techniques (like contrast stretching or de-blurring by the nearest neighbor procedure) provided by "Imaging packages" use no prior model of the process that created the image. With images, enhancement noise can be effectively removed by sacrificing some resolution, but this is not acceptable in many applications. More advanced image processing techniques must be applied to recover the object. De-Convolution is an example of an image restoration method. It is capable of increasing resolution, especially in the axial direction removing noise increasing contrast.

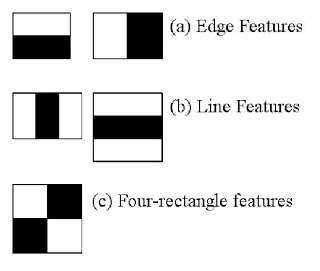
**4.2.2 GRAYSCALE CONVERSION**

Grayscale is a range of monochromatic shades from black to white. Many image editing programs allow converting a color image to black and white, or grayscale. This process removes all color information, leaving only the luminance of each pixel. The reason for differentiating such images from any other sort of color image is that less information needs to be provided for each pixel. In addition, grayscale images are entirely sufficient for many tasks and so there is no need to use more complicated and harder-to-process color images.

**4.2.3 FACE DETECTION USING HAAR CASCADES**

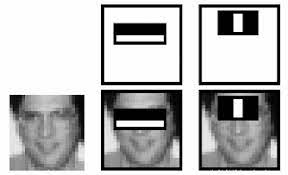
Object Detection using Haar feature-based cascade classifiers is an

effective object detection method proposed by Paul Viola and Michael Jones in their paper, “Rapid Object Detection using a Boosted Cascade of Simple Features” in 2001. It is a machine learning-based approach where a cascade function is trained from a lot of positive and negative images. It is then used to detect objects in other images. Initially, the algorithm needs a lot of positive images (images of faces) and negative images (images without faces) to train the classifier. Then we need to extract features from it. They are just like our convolutional kernel. Each feature is a single value obtained by subtracting the sum of pixels under the white rectangle from the sum of pixels under the black rectangle.



**FIGURE 4.2 HAAR CASCADE CLASSIFIER**

All possible sizes and locations of each kernel are used to calculate plenty of features. For each feature calculation, we need to find the sum of pixels under white and black rectangles. To solve this, they introduced integral images. It simplifies the calculation of the sum of pixels, how large may be the number of pixels, to an operation involving just four pixels.

****

**FIGURE 4.3 HAAR FEATURES SELECTED FOR FACE DETECTION.**

For each feature, it finds the best threshold which will classify the faces to positive and negative. We select the features with a minimum error rate, which means they are the features that best classifies the face and non-face images. Each image is given equal weight in the beginning. After each classification, the weights of misclassified images are increased. Then the again same process is done. New error rates are calculated. Also new weights. The process is continued until the required accuracy or error rate is achieved or the required number of features is found. The final classifier is a weighted sum of these weak classifiers. It is called weak because it alone cannot classify the image, but together with others forms a strong classifier. The final setup had around 6000 features, each 24x24 window. In an image, most of the image region is a non-face region. So, it is a better idea to have a simple method to check if a window is not a face region. For this, they introduced the concept of Cascade of Classifiers.

**4.3 TRAINING AND TESTING PHASE**

The facial expression recognition system is implemented using a convolutional neural network. The block diagram of the system is shown in the following figures:

Diagram

Description automatically generated

**FIGURE 4.4 TRAINING PHASE**

Diagram

Description automatically generated

**FIGURE 4.5 TESTING PHASE**

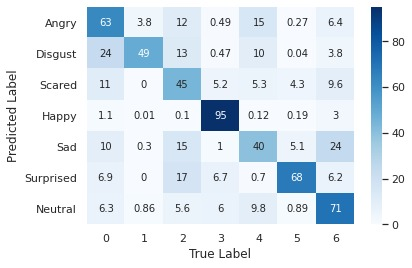
During training, the system received training data comprising grayscale images of faces with their respective expression label and learned a set of weights for the network. The training step took an input as an image with a face. Thereafter, an intensity normalization is applied to the image. The normalized images are used to train the Convolutional Network. To ensure that the training performance is not affected by the order of presentation of the examples, the validation dataset is used to choose the final best set of weights out of a set of training performed with samples presented in different orders. The output of the training step is a set of weights that achieve the best result with the training data. During the test, the system received a grayscale image of a face from the test dataset and gives an output of the predicted expression by using the final network weights learned during training

**4.4 CLASSIFICATION AND EVALUATION METRICS**

Classification is one of the most widely used problems in machine learning with various industrial applications, from face recognition, content moderation, medical diagnosis, to text classification, hate speech detection on Twitter. Models such as support vector machine (SVM), logistic regression, decision trees, random forest, XGboost, convolutional neural network, recurrent neural network are some of the most popular classification models. There are various ways to evaluate a classification model.

**4.4.1 CONFUSION MATRIX**

One of the key concepts in classification performance is the confusion matrix (also known as error matrix), which is a tabular visualization of the model predictions versus the ground-truth labels. Each row of the confusion matrix represents the instances in a predicted class and each column represents the instances in an actual class. The diagonal elements of the matrix denote the correct prediction for different classes, while the off-diagonal elements denote the mis-classified samples.



**FIGURE 4.6 CONFUSION MATRIX**

**4.4.2 CLASSIFICATION ACCURACY**

Accuracy is a metric that generally describes how the model performs across all classes. It is useful when all classes are of equal importance. It is calculated as the ratio between the number of correct predictions to the total number of predictions.

https://blog.paperspace.com/content/images/2020/09/Fig06.jpg

Classification accuracy is perhaps the simplest metrics one can imagine and is defined as the number of correct predictions divided by the total number of predictions, multiplied by 100. The accuracy may be deceptive. One case is when the data is imbalanced. The most used curve to understand the progress of Neural Networks is accuracy.

Diagram

Description automatically generated

**FIGURE 4.7 ACCURACY GRAPH**

**4.4.3 LOSS**

One of the most used plots to debug a neural network is a Loss curve during training. It gives us a snapshot of the training process and the direction in which the network learns. We can log loss in two periods:

* After every Epoch
* After every Iteration

During an epoch, the loss function is calculated across every data item and it is guaranteed to give the quantitative loss measure at the given epoch. But plotting curve across iterations only gives the loss on a subset of the entire dataset. An underfit model can be identified from the learning curve of the training loss only. It may show a flat line or noisy values of relatively high loss, indicating that the model was unable to learn the training dataset at all.

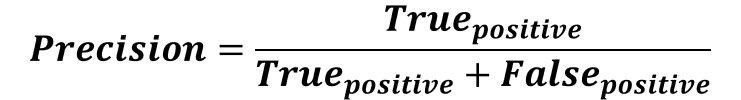
Diagram

Description automatically generated

**FIGURE 4.8 LOSS GRAPH**

**4.4.4 PRECISION**

The precision is calculated as the ratio between the number of Positive samples correctly classified to the total number of samples classified as Positive (either correctly or incorrectly). The precision measures the model's accuracy in classifying a sample as positive.

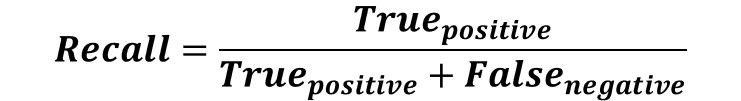


When the model makes many incorrect Positive classifications or few correct Positive classifications, this increases the denominator and makes the precision small. On the other hand, the precision is high when:

1. The model makes many correct Positive classifications (maximize True Positive).
2. The model makes fewer incorrect Positive classifications (minimize False Positive).

**4.4.5 RECALL**

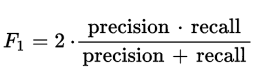
The recall is calculated as the ratio between the number of Positive samples correctly classified as Positive to the total number of Positive samples. The recall measures the model's ability to detect positive samples. The higher the recall, the more positive samples detected.



The recall cares only about how the positive samples are classified. This is independent of how the negative samples are classified, e.g., for the precision. When the model classifies all the positive samples as Positive, then the recall will be 100% even if all the negative samples were incorrectly classified as Positive.

**4.4.5 F1 SCORE**

One popular metric which combines precision and recall is called F1-score, which is the harmonic mean of precision and recall. The F-score, also called the F1-score, is a measure of a model's accuracy on a dataset.

****

F1 Score might be a better measure to use if we need to seek a balance between Precision and Recall and there is an uneven class distribution (a large number of Actual Negatives).

**4.4.6 ROC**

The receiver operating characteristic curve is a plot that shows the performance of a binary classifier as a function of its cut-off threshold. It essentially shows the true positive rate (TPR) against the false positive rate (FPR) for various threshold values. Many of the classification models are probabilistic, i.e., they predict the probability of a sample being a cat. They then compare that output probability with some cut-off threshold and if it is larger than the threshold, they predict its label as a cat, otherwise as non-cat. As an example, your model may predict the below probabilities for 4 sample images: [0.45, 0.6, 0.7, 0.3]. Then depending on the threshold values as below, different labels may be obtained:

cut-off= 0.5: predicted labels = [0,1,1,0] (default threshold)

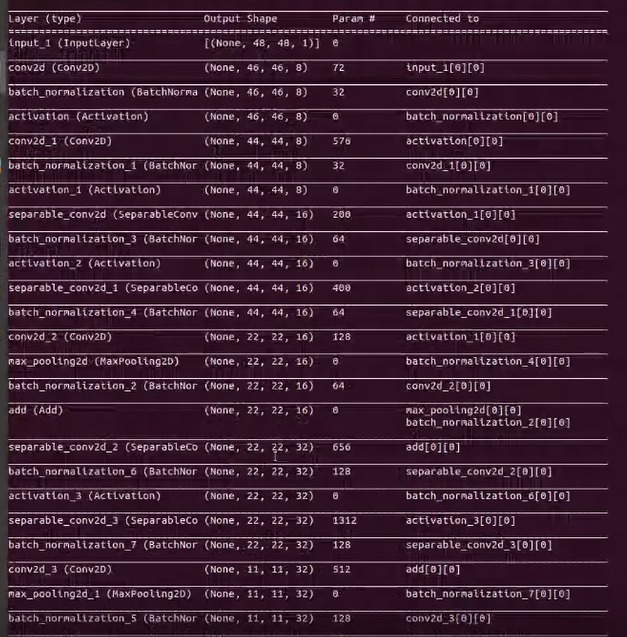
cut-off= 0.2: predicted labels= [1,1,1,1]

cut-off= 0.8: predicted labels= [0,0,0,0]

Each of these scenarios would result in different precision and recall (as well as TPR, FPR) rates. ROC curve essentially finds out the TPR and FPR for various threshold values.

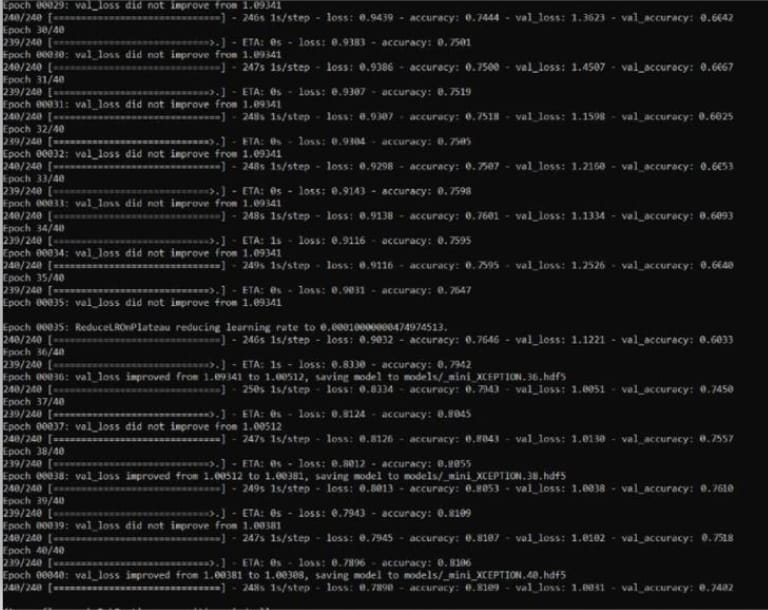
**4.5 EXPERIMENT AND RESULTS**

**4.5.1 MODEL SUMMARY**

****

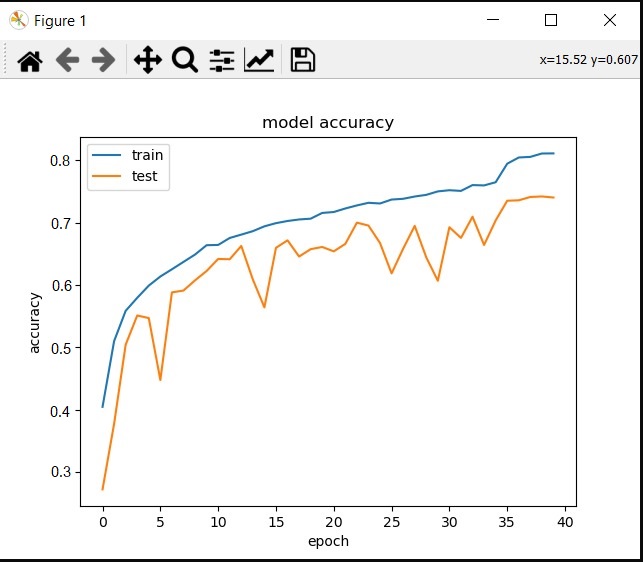
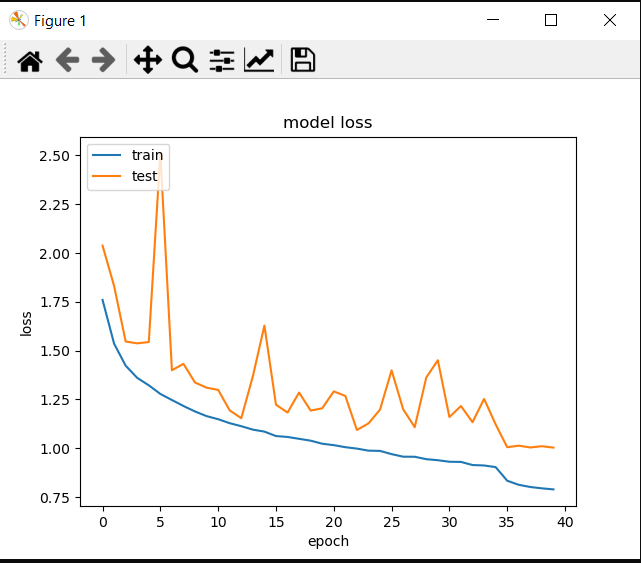
**FIGURE 4.9 NETWORK STRUCTURE**

**4.5.1 TRAINED MODEL**

****The network was trained for 40 epochs, and an accuracy of 81% was obtained on the training data.

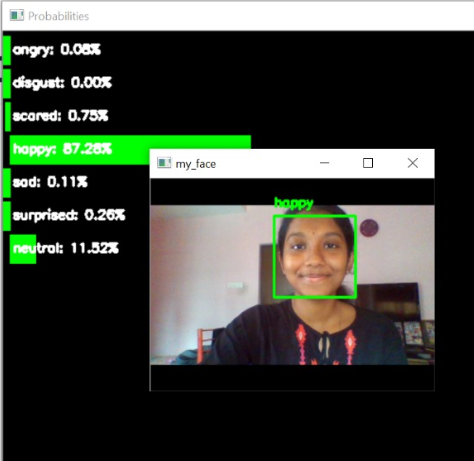
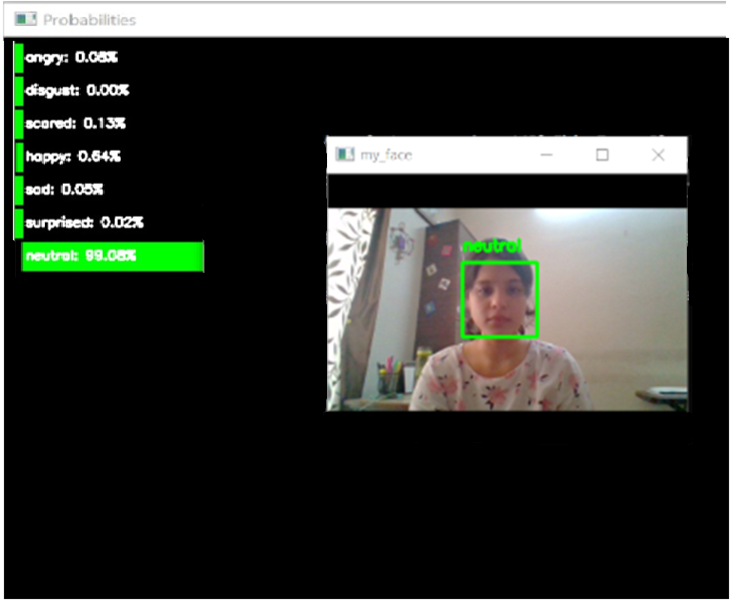
**FIGURE 4.10 TRAINED MODEL**

**4.5.2 MODEL PERFORMANCE**

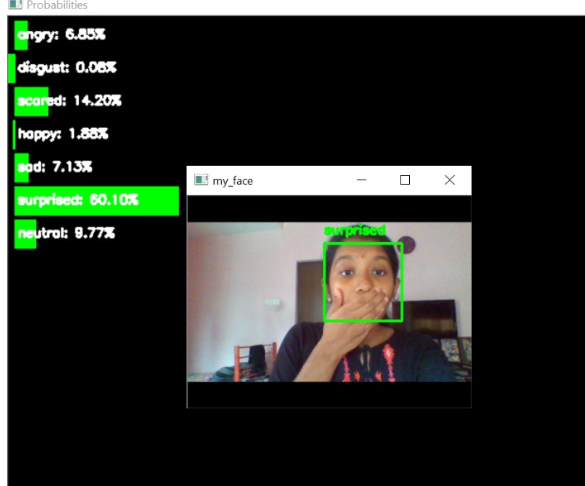
** **

**FIGURE 4.11 ACCURACY AND LOSS GRAPH**

**4.5.3 IMPLEMENTATION- EMOTION RECOGNITION**

** **

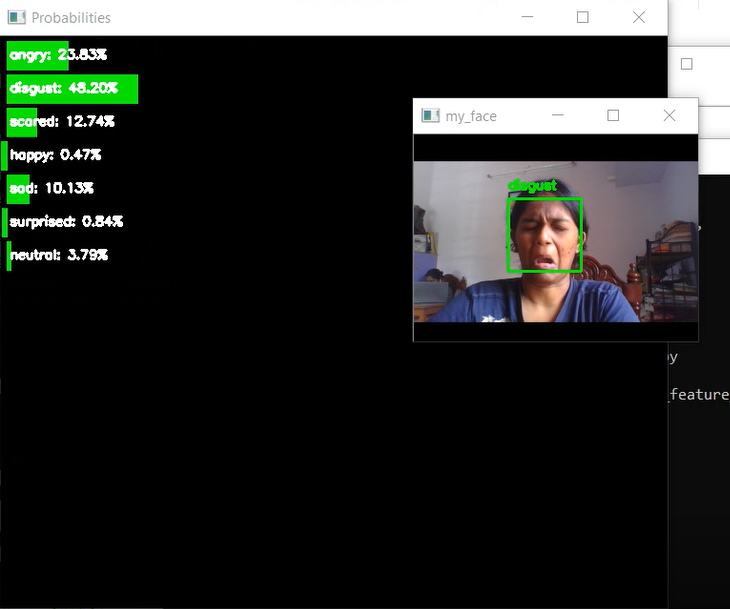
**A screenshot of a computer

Description automatically generated with low confidence **

**A screenshot of a person

Description automatically generated with medium confidence Graphical user interface, application

Description automatically generated**

****

**FIGURE 4.12 EMOTION RECOGNITION IMPLEMENTATION**

The experimental analysis is evaluated using the Python IDLE tool. Mini-Xception algorithm is applied to the emotion detection dataset. Dataset contains 35,887 images. Accuracy is calculated from the confusion matrix. To train our models we used dataset: FER-2013. The FER-2013 dataset had been created by acquiring and combining the result of Google image search for every particular emotion. Every image in the FER dataset is labeled image and it consists of seven emotional images i.e., anger, disgust, scared, happy, sad, surprise, and neutral. This dataset consists of 35,887 images in the training set, 3,500 labeled images in the test set, and 3,500 images in the development set. It consists of pair of posed and un-posed identification images; these images are in grayscale and the pixel value is 48x48. Even though this dataset consists of labeled emotions, it was discovered that the proposed model achieved better accuracy. Accuracy is calculated from the confusion matrix. For quantitative analysis of the experimental results, the following performance metrics are considered, including accuracy (AC). To do this we also use the variables True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN).

**CHAPTER 5**

**REQUIREMENTS**

**5.1 SOFTWARE REQUIREMENTS**

**5.1.1 ANACONDA PYTHON**

Logo, company name

Description automatically generated

**FIGURE 5.1 ANACONDA AND PYTHON**

Anaconda is a free and open-source distribution of the Python and R programming languages for scientific computing (data science, machine learning applications, largescale data processing, predictive analytics, etc.), that aims to simplify package management and deployment. It is developed and maintained by Anaconda, Inc. The distribution includes data-science packages suitable for Windows, Linux, and macOS. Package versions are managed by the package management system Conda. This package manager was spun out as a separate open-source package as it ended up being useful on its own and for other things than Python. There is also a small, bootstrap version of Anaconda called Miniconda, which includes only Conda, Python, the packages they depend on, and a small number of other packages. Python is a widely used high-level, general-purpose, interpreted, dynamic programming language. Python supports multiple programming paradigms, including object-oriented, imperative, and functional programming or procedural styles. It features a dynamic type system and automatic memory management and has a large and comprehensive standard library. Python interpreters are available for installation on many operating systems, allowing Python code execution on a wide variety of systems. Using third-party tools, such as Py2exe or Pyinstaller, Python code can be packaged into standalone executable programs for some of the most popular operating systems, allowing the distribution of Python-based software for use in those environments without requiring the installation of a Python interpreter. CPython, the reference implementation of Python, is free and open-source software and has a community-based development model, as do nearly all its alternative implementations. CPython is managed by the non-profit Python Software Foundation.

**5.1.2 TENSORFLOW**

Logo, company name

Description automatically generated

**FIGURE 5.2 TENSOR FLOW**

TensorFlow is a Python-friendly open-source library for numerical computation that makes machine learning faster and easier. TensorFlow is an open-source library for numerical computation and large-scale machine learning. TensorFlow bundles together a slew of machine learning and deep learning (aka neural networking) models and algorithms and makes them useful by way of a common metaphor. It uses Python to provide a convenient front-end API for building applications with the framework while executing those applications in high-performance C++.

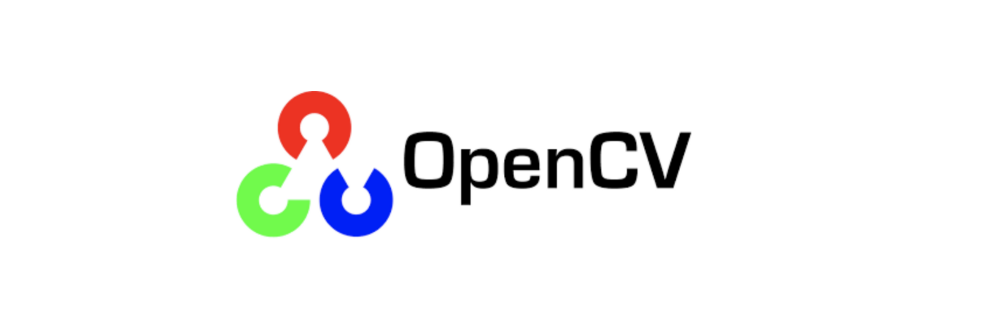
**5.1.3 KERAS**



**FIGURE 5.3 KERAS**

Keras is a minimalist Python library for deep learning that can run on top of Theano or TensorFlow. It was developed to make implementing deep learning models as fast and as easy as possible for research and development. It runs on Python 2.7 or 3.5 and can seamlessly execute on GPUs and CPUs that have the underlying frameworks.

**5.1.4 OPEN CV**



**FIGURE 5.4 OpenCV**

OpenCV (Open Source Computer Vision Library) is an open-source computer vision and machine learning software library. OpenCV was built to provide a common infrastructure for computer vision applications and to accelerate the use of machine perception in commercial products. Being a BSD-licensed product, OpenCV makes it easy for businesses to utilize and modify the code. The library has more than 2500 optimized algorithms, which includes a comprehensive set of both classic and state-of-the-art computer vision and machine learning algorithms. These algorithms can be used to detect and recognize faces, identify objects, classify human actions in videos, track camera movements, track moving objects, extract 3D models of objects, produce 3D point clouds from stereo cameras, stitch images together to produce a high-resolution image of an entire scene, find similar images from an image database, remove red eyes from images taken using flash, follow eye movements, recognize scenery and establish markers to overlay it with augmented reality.

**5.1.5 NUMPY**

Icon

Description automatically generated

**FIGURE 5.5 NumPY**

NumPy is the fundamental package for scientific computing in Python. It is a Python library that provides a multidimensional array object, various derived objects (such as masked arrays and matrices), and an assortment of routines for fast operations on arrays, including mathematical, logical, shape manipulation, sorting, selecting, I/O, discrete Fourier transforms , basic linear algebra, basic statistical operations, random simulation and much more.

**CHAPTER 6**

**CONCLUSION AND FUTURE WORK**

In this study, by combining the online course platforms and a compact deep learning model based on the architecture of CNN, a framework was proposed to analyze students’ emotions according to their facial expressions. The emotions were classified into anger, disgust, scared, happiness, sadness, surprise, and neutral respectively. Improvised algorithms with better performance and shorter operation time, including pre-processing and deep learning models, may be developed over time. For instance, the image pre-processing contains face detection, alignment, rotation, and resize, but with problems, such as backlight, shadows, and facial incompleteness, caused by complex environments, the current methods have their shortcomings that may be solved in the future. Although the CNN model in the proposed framework currently performs well, it may be replaced by models with higher learning capabilities and higher classification accuracy in the future. To ensure the competitiveness of the framework in a longer period, it needs to be adjusted and maintained regularly, and more advanced algorithms and technologies should be adopted to update it.

**REFERENCES**

1. Agrawal and N. Mittal, “Using CNN for facial expression recognition: a study of the effects of kernel size and number of filters on accuracy,” The Visual Computer, vol. 36, no. 2, pp. 405–412, 2020.
2. H. Hasan, B. Huang, and G. Tian, “Facial expression recognition based on deep convolution long short-term memory networks of double-channel weighted mixture,” Pattern Recognition Letters, vol. 131, pp. 128–134, 2020.
3. G. Tonguç and B. O. Ozkara, “Automatic recognition of student emotions from facial expressions during a lecture,” Computers & Education, vol. 148, Article ID 103797, 2020.
4. Jahandad, S. M. Sam, K. Kamardin, N. N. Amir Sjarif, and N. Mohamed, “Offline signature verification using deep learning convolutional neural network (CNN) architectures GoogLeNet inception-v1 and inception-v3,” Procedia Computer Science, vol. 161, pp. 475–483, 2019.
5. T. Liu, H. Liu, Z. Chen et al., “FBRDLR: fast blind reconstruction approach with dictionary learning regularization for infrared microscopy spectra,” Infrared Physics & Technology, vol. 90, pp. 101–109, 2018.
6. Z. Huang, H. Fang, Q. Li et al., “Optical remote sensing image enhancement with weak structure preservation via spatially adaptive gamma correction,” Infrared Physics & Technology, vol. 94, pp. 38–47, 2018.
7. R. Donoso, C. San Martín, and G. Hermosilla, “Reduced isothermal feature set for long wave infrared (LWIR) face recognition*,” Infrared Physics & Technology*, vol. 83, pp. 114–123, 2017.
8. Y. Bi, M. Lv, Y. Wei, N. Guan, and W. Yi, “Multi-feature fusion for thermal face recognition,” Infrared Physics & Technology, vol. 77, pp. 366–374, 2016.
9. H. Wu, Y. Liu, L. Qiu, and Y. Liu, “Online judge system and its applications in C language teaching,” in Proceedings of the International Symposium on Educational Technology (ISET), pp. 57–60, Beijing, China, July 2016.
10. H. Liu, Z. Zhang, S. Liu, J. Shu, T. Liu, and T. Zhang, “Blind spectrum reconstruction algorithm with L0-sparse representation,” Measurement Science and Technology, vol. 26, no. 8, pp. 085501–085507, 2015.
11. P. Liu, S. Han, Z. Meng, and Y. Tong, “Facial expression recognition via a boosted deep belief network,” in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1805–1812, Columbus, OH, USA, June 2014.
12. H. Gunes and B. Schuller, “Categorical and dimensional affect analysis in continuous input: current trends and future directions,” Image and Vision Computing, vol. 31, no. 2, pp. 120–136, 2013.
13. R. Zhi, M. Flierl, Q. Ruan, and W. B. Kleijn, “Graph-preserving sparse nonnegative matrix factorization with application to facial expression recognition,” IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics), vol. 41, no. 1, pp. 38–52, 2011.
14. P. Lucey, J. F. Cohn, T. Kanade, J. Saragih, Z. Ambadar, and I. Matthews, “The extended Cohn-Kanade dataset (CK+): a complete dataset for action unit and emotion-specified expression,” in Proceedings of the 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition-Workshops, pp. 94–101, San Francisco, CA, USA, July 2010.