REAL-TIME FACIAL EXPRESSION DETECTION USING MACHINE LEARNING

Submitted in partial fulfillment of the requirements for the award of
Bachelor of Engineering degree in Computer Science and Engineering

By

Thottempudi Bhavya Sai(Reg.No-39111033)
VangetiMounika(Reg.No-39111058)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING SCHOOL OF COMPUTING

SATHYABAMA
INSTITUTE OF SCIENCE AND TECHNOLOGY (DEEMED TO BE UNIVERSITY)
Accredited with Grade "A" by NAAC | 12B Status by UGC | Approved by AICTE
JEPPIAAR NAGAR, RAJIV GANDHISALAI, CHENNAI - 600119

APRIL - 2023

SATHYABAMA
INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)
Accredited with Grade "A" by NAAC | 12B Status by UGC | Approved by AICTE

www.sathyabama.ac.in

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

BONAFIDE CERTIFICATE

This is to certify that this Project Report is the bonafide work of **Thottempudi Bhavya Sai (39111033)and Vangeti Mounika (39111058)** who carried out the Project Phase-2 entitled **REAL-TIME FACIAL EXPRESSION DETECTION USING MACHINE LEARNING"** under my supervision from Jan 2023 to April 2023.

Internal Guide

Dr. G. KALAIARASI, M.E., Ph.D

Head of the Department

Dr. L. LAKSHMANAN, M.E., Ph.D.

SATHYABAMA INSTITUTE OF SCIENCE AND TECHNOLOGY
DEPARTMENT OF CSE

20.04.2023

Submitted for Viva voce Examination held on ---

Internal Examiner **External Examiner**

DECLARATION

I **THOTTEMPUDI BHAVYA SAI(Reg.No- 39111033),** hereby declare that the Project Phase-2 Report entitled **REAL-TIME FACIAL EXPRESSION DETECTION USING MACHINE LEARNING"** done by me under the guidance of **Dr. G. KALAIARASI, M.E.,Ph.D.,** is submitted in partial fulfillment of the requirements for the award of Bachelor of Engineering degree in **Computer Science and Engineering**.

Bhavya

DATE: 20.04.23

PLACE: Chennai **SIGNATURE OF THE CANDIDATE**

ACKNOWLEDGEMENT

I am pleased to acknowledge my sincere thanks to **Board of Management** of **SATHYABAMA** for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

I convey my thanks to **Dr. T.Sasikala M.E., Ph.D.**, **Dean**, School of Computing, **Dr. L. Lakshmanan M.E., Ph.D.,** Head of the Department of Computer Science and Engineering for providing me with necessary support and details at the right time during the progressive reviews.

I would like to express my sincere and deep sense of gratitude to my Project Guide **Dr. G. Kalaiarasi M.E.,Ph.D.,** for her valuable guidance, suggestions and constant encouragement paved way for the successful completion of my phase-2 project work.

I wish to express my thanks to all the Teaching and Non-teaching staff members of the **Department of Computer Science and Engineering** who were helpful in many ways for the completion of the project.

ABSTRACT

Facial emotion analysis has been a topic of interest for many researchers as it has potential applications in areas such as human-computer interaction, healthcare, and security. Traditionally, the study of facial expressions and emotions has been a subject of research in psychology, but recent advances in computer vision and machine learning have opened up new possibilities in this area. Facial emotion recognition using convolutional neural networks (FERC) is a state-of-the-art approach that can automatically estimate a person's emotions from photographs. FERC is based on a two-part CNN that is specifically designed to handle facial images. The first part of the CNN focuses on removing the background from the image, while the second part is responsible for extracting the facial feature vectors. By using FERC, it is possible to accurately discern a person's emotions from their facial expressions. This can have many potential applications, such as improving human-computer interaction by allowing machines to respond appropriately to a user's emotional state. Additionally, FERC can be used in healthcare to monitor patient's emotional states, and in security to identify individuals who may pose a threat based on their emotional state. Overall, FERC is an exciting development in the field of computer vision and has the potential to revolutionize the way we interact with technology.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

Facial emotion recognition (FER) is a rapidly growing research area in computer vision and machine learning. The ability to automatically detect and interpret human emotions from facial expressions has numerous practical applications, including in areas such as human-computer interaction, healthcare, and security. The recent advances in deep learning algorithms have made it possible to achieve high accuracy in FER tasks using convolutional neural networks (CNN).

The goal of this project is to design and implement a CNN-based FER system that can accurately recognize basic emotions from facial images. The system will be trained and evaluated on two publicly available datasets, namely the CK+(cohn Kanade) and JAFFE (Japanese Female Facial Expression) datasets. The six basic emotions that will be recognized by the system are happy, sad, angry, disgust, surprise, and fear.

The proposed FER system will consist of three main components: a convolutional feature extractor, a classification network, and a softmax output layer. The convolutional feature extractor will be responsible for automatically learning the relevant features from the raw input images. The classification network will take the learned features and classify them into one of the six basic emotions. The softmax output layer will provide the probability distribution over the six emotions.

The project will start with a literature review of the latest research on FER using CNNs. The review will cover various aspects of FER, including feature extraction, classification, and dataset selection. Based on the review, the most suitable CNN architecture and training methodology will be selected for the proposed FER system. Next, the system will be implemented and trained on the CK+ and JAFFE datasets using the selected CNN architecture and training methodology. The performance of

the system will be evaluated using various metrics, including accuracy, precision, recall, and F1 score.

In addition to the main task of designing and implementing the FER system, the project will also investigate the impact of various hyperparameters on the system's performance. Specifically, the effect of different convolutional layer architectures, the number of filters, and the learning rate will be studied. The project will also explore the effect of data augmentation techniques on the system's performance.

Finally, the project will conclude with a discussion of the results and potential future work. The project will demonstrate the effectiveness of CNNs in FER tasks and provide insights into the impact of various hyperparameters on system performance. The results of this project can be used to improve the accuracy and robustness of FER systems in various applications, including human-computer interaction, healthcare, and security.

Facial Emotion Recognition (FER) using Convolutional Neural Networks (CNN) is a popular application of computer vision that aims to automatically identify and classify the emotions displayed by a person's face in real-time. The objective of FER using CNN is to create an accurate and efficient system that can recognize facial expressions and classify them into different emotion categories such as happy, sad, angry, fearful, and surprised.

The primary goal of FER using CNN is to improve human-computer interaction by creating systems that can detect and interpret human emotions. These systems can be used in a wide range of applications, including human-robot interaction, virtual reality, and video surveillance.

To achieve this objective, the FER using CNN model is trained on a large dataset of facial expressions, which is labeled with the corresponding emotion category. The CNN model extracts features from the input images and learns to classify them based

on the patterns and correlations between the facial features and the corresponding emotion labels.

The objective of FER using CNN is to create a model that is accurate, robust, and efficient. This requires careful selection of the architecture, training data, and optimization strategies. The ultimate goal is to create a system that can recognize facial expressions in real-time with high accuracy, even in challenging conditions such as varying lighting conditions and facial occlusions. By achieving this objective, FER using CNN can have a significant impact on a wide range of applications, improving human-computer interaction and enhancing the user experience.

Project Domain

MACHINE LEARNING

Machine learning is an application of artificial intelligence (AI) that provides systems the ability to automatically learn and improve from experience without being explicitly programmed. Machine learning techniques and algorithms focus on the development of computer systems that can easily access data and make it readily available for using it for improvement.

The process of machine learning starts with a dataset. The primary aim is common for all processes, and that is to allow the computers to learn on their own without any kind of human input or assistance.

SOME MACHINE LEARNING METHODS

Machine learning algorithms have been categorized into two main subcategories supervised and unsupervised.

Supervised algorithms require supervision by someone who has machine learning skills to provide input and receive the desired output. In addition to that, he/she will be involved in furnishing feedback about the accuracy. Once the model training is complete, the algorithm will be applied to new data.

Unsupervised algorithms do not require any training with the data. However, they do make use of an iterative approach. This approach is called Deep Learning (DL).

Unsupervised learning algorithms are also referred to as neural networks. These networks are used wherever the complexity is more and can perform more tasks than supervised learning systems. These neural networks progress by combing through

Dataset for training and automatically identify correlations between variables present in the dataset. Once the model is trained, the algorithm can then use its associations to test the data.

- **Supervised machine learning algorithms** are applied to the previously studied data in the past and then to new data. Such a system is able to provide outputs for any new input once sufficient training is done. This algorithm also compares its output with the correct, intended output and finds discrepancies, and then modifies the model accordingly.

- On the other hand, **unsupervised machine learning algorithms** have been when the information has neither been classified nor labeled. This category studies how systems can make inferences into a function that describes a hidden structure in unlabeled data. But there is a drawback, this system doesn't tell the right output, rather it explores the data and draws inferences from datasets.

- **Semi-supervised machine learning algorithms** lie in between the above mentioned systems. This is because they use both labelled and unlabeled data for training purposes –a smaller amount of labelled data and a larger amount of unlabeled data in combination. This system uses techniques that are able to improve accuracy. Generally, semi-supervised learning opted when the acquired dataset (labelled) requires skilled resources to train it.

- **Reinforcement machine learning algorithms** are a learning technique that interacts with the environment. It is done by producing actions and then finding

discrepancies. This method allows machines and software agents to contextually compute behavioural patterns in the dataset in order to maximize its performance. Simple reward feedback is then analyzed in order to conclude which action is the best. The mentioned process is known as the reinforced signal.

Machine learning has enabled the analysis of larger quantities of data. While on one hand, it delivers faster, more accurate results, on the other hand, it may also require more time and resources while training. A combination of machine learning and AI along with cognitive technologies makes it even more efficient in processing larger datasets.

Image Processing

In imaging sciences, image processing is the most important part of the process. Most techniques require treating the image as a two-dimensional signal and then standard signal processing techniques are applied to it. However, images can also be processed as three-dimensional signals where the third dimension is time and it becomes the z-axis. Image processing is done to digital images, but can also be done for optical and analogous images.

This paper discusses some general techniques that can be applied to all of them. The collection of images is known as 'imaging'. It is closely related to computer graphics and vision. In computer graphics, input images have to be made manually from physical models. Whereas in computer vision, high-level image processing aims to decipher the physical contents of an image.

Images have also gained scopes due to the growing vitality of scientific visualization. Image analysis caters to the extraction of information using images. This information is mainly from the digital ones. Analysis tasks can be made as simple as barcoding. But at the same time, they can be as sophisticated as the identification of an individual from his/her face. The human visual cortex is an outstanding image analysis apparatus, especially if you wish to extract high-level features.

For this particular reason, several important image analysis tools like edge detectors as well as neural networks have been greatly inspired by the existing human visual perception models. Image editing includes the process of tempering the images. When a traditional image (analog image) is edited, the process is called "Photo Retouching". A lot of image editing software is also used to create computer art from the very beginning. Raster images are initially stored in a computer and that too in the form of a grid, or sometimes in pixels. These pixels actually encompass the information about the colour and brightness.

Image editors may then change the pixels in order to enhance the image. Several graphics applications have been made capable of merging individual images into a single file. Their orientation and placement can also be controlled. When you are selecting a raster image, it requires the separation of edges from the background, if it was not rectangular in shape. This is referred to as “silhouetting”.

Another method to silhouette an image is to use clipping paths. Moving further down the process, alpha compositing can be done for soft translucent edges while you are selecting images. Other than that, there are many more ways to silhouette an image. Another famous method to create a composite image is by using transparent layers. Here, the background image is used as a bottom layer, and the image that has to be included is placed in another layer present above. While using an image layer mask, all parts are hidden except for the ones that are to be merged. This gives the perfect impression of added parts.

STEPS OF IMAGE PROCESSING:

Image Acquisition

This is the first step or process of the fundamental steps of digital image processing. Image acquisition could be as simple as being given an image that is already in digital form. Generally, the image acquisition stage involves pre-processing, such as scaling, etc.

Image Enhancement

Image enhancement is among the simplest and most appealing areas of digital image processing. Basically, the idea behind enhancement techniques is to bring out detail that is obscured, or simply to highlight certain features of interest in an image. Such as changing brightness & contrast etc.

Image Restoration

Image restoration is an area that also deals with improving the appearance of an image. However, unlike enhancement, which is subjective, image restoration is objective, in the sense that restoration techniques tend to be based on mathematical or probabilistic models of image degradation.

Colour Image Processing

Colour image processing is an area that has been gaining importance because of the significant increase in the use of digital images over the Internet. This may include colour modelling and processing in a digital domain etc.

Wavelets and Multi-resolution Processing

Wavelets are the foundation for representing images in various degrees of resolution. Images subdivision successively into smaller regions for data compression and for pyramidal representation.

Compression

Compression deals with techniques for reducing the storage required to save an image or the bandwidth to transmit it. Particularly in the uses of the internet, it is very much necessary to compress data.

Morphological Processing

Morphological processing deals with tools for extracting image components that are useful in the representation and description of shape.

CHAPTER 2

LITERATURE SURVEY

[1] Facial emotion recognition using convolutional neural networks (FERC). This study shows that was not very intuitive, the running time increased with more layers, and that was not reported in the current manuscript because it did not add significant value to the study. In the ultimate extent of this work, researchers might also try independently

exchanging the number of layers in two CNNs. Also, can do a ton of work if you load a different number of filters into each layer. This algorithm did not work with more than one face inside the equal picture, at an identical distance from the digicam.

[2] Facial emotion recognition using Classifiers. The experimental outcomes display that the proposed approach achieves a recognition rate of ninety-three. Fifty-three percent whilst the use of SVM due to the classifier and Eighty-two. Nine-seven percent when using the MLP classifier, and seventy-nine. Nine-seven percent when the usage of the KNN classifier, implies that the experiment achieves the best results.

[3] "Multi-Block Deep Convolutional Neural Networks (DCNN) for Facial Expression Recognition" by Song et al. The authors conceived and built a multi-block DCNN model that can distinguish facial expressions in virtual, stylised, and human characters. The suggested model uses four blocks with varied computational elements in multi-block DCNN to extract discriminative features from facial images. The experiment's results are compared to existing facial expression detection approaches, and the results demonstrate that the suggested model has an accuracy of 93 percent, which is significantly higher than previous research.

[4] "Facial Emotion Recognition Based on Sparse Autoencoder and SVM" by Xu et al. They proposed a facial emotion recognition method based on a sparse autoencoder

and SVM. The proposed method achieved an accuracy rate of 86.72% on the CK+(Cohn Kanade) dataset and 79.87% on the JAFFE dataset. The authors concluded that the proposed method outperforms other state-of-the-art methods in terms of recognition accuracy and robustness.

[5] "Facial Expression Recognition Using Convolutional Neural Networks and Active Appearance Models" by Alotaibi et al. The authors proposed a method that integrates CNNs and active appearance models (AAMs) for facial expression recognition. The proposed method achieved an accuracy rate of 90.4% on the CK+ dataset and 83.4% on the JAFFE dataset. The authors concluded that the proposed method outperforms other state-of-the-art methods in terms of recognition accuracy and robustness.

[6] Emotion-based Convolution Neural Network. Classification of cognitive models of facial expressions. The model classifies positive and negative images, which largely determines the area of the image, and the performance of the network depends on different pieces of training options. Kaggle facial expression FER-2013 Implemented a database with 7 facial expression tags (happy, neutral, surprised, fearful, angry, disgusted, sad). The present model consists of 2*2 convolution layers, 2*2 pooling layers, three hundred nodes associated layers, and SoftMax regression layers.

[7] "Facial Emotion Recognition Based on Deep Convolutional Neural Networks with Data Augmentation" by Li et al. The authors proposed a deep learning-based FER system that uses data augmentation techniques to improve the robustness of the system. The proposed method achieved an accuracy rate of 93.6% on the CK+ dataset and 94.4% on the JAFFE dataset. The authors concluded that the proposed method outperforms other state-of-the-art methods in terms of recognition accuracy and robustness.

[8] AI-based FER evaluated all existing systems. These are classified into three sections Children, teens, and Senior Citizens. It also emphasizes the different steps of FER such as pre-processing, function extraction, classification, and state of the art

of CNN models. It often discusses open questions and problems and provides solutions.

[9] "A Deep Learning Architecture for Emotion Identification from Photos" by Wang et al. This research suggested a deep learning architecture based on CNNs for emotion identification from photos. The proposed model comprises six convolutional layers, two max pooling layers, and two fully connected layers. The authors achieved a final accuracy of 0.60 after tweaking the various hyperparameters.

[10] The six discrete emotions of happy, sad, angry, surprise, bore, and disgust were calculated using three facial datasets: LFW, The Extended Yale Face Database B, and Google Facial expression comparison dataset. The experiment's results are compared to existing facial expression detection approaches, and the results demonstrate that

the suggested model has an accuracy of 93 percent, which is significantly higher than previous research.

2.1 Inferences from Literature Survey

From the above-mentioned literature works, it is clear that there has been effective research on this topic has been done and many models have been proposed. It is evident that the above-mentioned systems have their own pros and cons. While some of the recent works involve hybrid technologies and provide better accuracies, they are still far from what is needed. With higher accuracy, comes the need for low computational costs, high processing speed, and most of all, the convenience of use.

2.2 Open problems in Existing System

One major issue in face pose detection is how to extract effective representation features from a facial image. Through the analysis of the works of literature on existing method, it can be found that the features extracted by state-of-the-art methods have the problem that the original emotional information is easy to be lost. In addition, the generalization and robustness of these network models are also poor and the accuracy of facial expression recognition is not high. For the existing system

of facial emotional recognition using CNN based system, a small dataset is taken. The system shows low accuracy and high processing time.

In addition to the issue of feature extraction in face pose detection, there are several open problems in the existing facial emotion recognition system using a CNN-based approach. One of the main problems is the limited availability of large-scale and diverse datasets for training and testing. The existing datasets are often limited in size and diversity, making it difficult to develop robust and accurate facial emotion recognition systems that can generalize well across different individuals, cultures, and scenarios.

Another problem is the lack of interpretability and transparency in the CNN-based facial emotion recognition models. These models often rely on complex and opaque architectures that make it difficult to understand how the system arrives at its predictions. This lack of interpretability can limit the adoption of facial emotion recognition systems in practical applications, where transparency and accountability are critical.

Furthermore, the existing facial emotion recognition systems based on CNNs often suffer from overfitting, which occurs when the model performs well on the training data but poorly on the testing data. This is a common problem when dealing with small datasets, such as those used in the existing facial emotion recognition system. Overfitting can limit the generalization and robustness of the model, making it less effective in real-world scenarios.

Finally, the existing facial emotion recognition systems based on CNNs often require significant computational resources, making them unsuitable for resource-constrained environments. This includes applications that require real-time processing, such as human-computer interaction and virtual reality.

Addressing these open problems requires developing more efficient, interpretable, and generalizable facial emotion recognition systems that can operate in real-world scenarios. This includes developing new techniques for feature extraction, addressing the lack of large-scale and diverse datasets, improving the interpretability and transparency of the models, and addressing the overfitting and computational requirements of the existing models. By addressing these open problems, researchers can develop more accurate, robust, and practical facial emotion recognition systems that have broader applications in various fields.

CHAPTER 3

REQUIREMENTS ANALYSIS

3.1 Feasibility Studies/Risk Analysis of the Project

Facial emotion recognition using Convolutional Neural Networks (CNN) is a popular research topic in computer vision and machine learning. While there are some risks associated with this project, they can be mitigated by following best practices and being transparent about the limitations and potential biases of the model.

Here are some potential risks to consider:

1. Bias in the training data: The accuracy of a facial emotion recognition model depends on the quality and diversity of the training data. If the training data is biased towards certain groups of people, the model may not be able to recognize emotions accurately for other groups. For example, if the training data is mostly composed of images of people of a certain race or gender, the model may not be as accurate at recognizing emotions in people from other races or genders.

   To mitigate this risk, it's important to use a diverse range of training data that represents the full range of people that the model will be used to recognize emotions in. It's also important to carefully evaluate the performance of the model across different demographics to ensure that it's not biased towards certain groups.
2. Privacy concerns: Facial emotion recognition involves processing images of people's

faces, which raises privacy concerns. There is a risk that the model could be used to identify individuals without their consent or to track their emotions over time. To mitigate this risk, it's important to be transparent about the intended use of the model and to obtain consent from individuals whose images will be used for training or testing. It may also be necessary to implement technical safeguards to protect the privacy of individuals, such as blurring or masking faces in images used for testing or implementing data retention policies to limit the amount of data stored.

3. Accuracy limitations: While CNNs have shown impressive performance in facial emotion recognition, they are not perfect and can make mistakes. There is a risk that

the model could be used in high-stakes situations where the consequences of an incorrect prediction could be significant.

To mitigate this risk, it's important to be transparent about the limitations of the model and to use it in situations where the consequences of a mistake are low or where the model's predictions can be cross-checked with other sources of information. Overall, facial emotion recognition using CNNs has the potential to be a valuable tool in a variety of applications, including mental health, marketing, and human-computer interaction. By being aware of the potential risks and taking steps to mitigate them, it's possible to develop a model that is both accurate and responsible.

3.2 Software Requirements

1. Python: Python is a popular programming language used in machine learning and computer vision. It is often used for building neural network models, including CNNs. 2. Deep learning frameworks: There are several deep learning frameworks available for Python, such as TensorFlow, PyTorch, and Keras. These frameworks provide high level APIs for building neural network models, including CNNs.

3. OpenCV: OpenCV (Open Source Computer Vision) is a library of programming functions used for computer vision tasks, such as image processing and object detection. It can be used for preprocessing and augmenting the input images before feeding them to the CNN.

4. GPU support: Training a CNN can be computationally intensive, especially if the dataset is large. Therefore, having access to a GPU can significantly speed up the training process. Some deep learning frameworks, such as TensorFlow and PyTorch, have

built-in support for GPU acceleration.

5. Data storage and management: A facial emotion recognition project will require a large dataset of images with labeled emotions for training and testing the model. Therefore, it is important to have a system for storing and managing the data, such as a database or file system.
6. Version control: Version control software, such as Git, can be used to manage the code and track changes to the model and its dependencies over time. This can make it easier to collaborate with other developers and ensure that the code is reproducible.

7. Deployment environment: Once the model has been trained, it will need to be deployed in a production environment, such as a web application or mobile app. The deployment environment will depend on the specific application, but may require additional software and infrastructure, such as a web server or cloud hosting.

FEASIBILITY STUDY

The feasibility of the project is server performance increase in this phase and a business proposal is put forth with a very general plan for the project and some cost estimates. During system analysis, the feasibility study of the proposed system is to be carried out. For feasibility analysis, some understanding of the major requirements for the system is essential.

Three key considerations involved in the feasibility analysis are

- Economical feasibility
- Technical feasibility
- Operational feasibility

ECONOMICAL FEASIBILITY

This study is carried out to check the economic impact that the system will have on the organization. The amount of funds that the company can pour into the research and development of the system is limited. The expenditures must be justified. Thus the developed system as well within the budget and this was achieved because most of the technologies used are freely available. Only the customized products had to be purchased.

TECHNICAL FEASIBILITY

This study is carried out to check the technical feasibility, that is, the technical requirements of the system. Any system developed must not have a high demand on the available technical resources. This will lead to high demands being placed on the client. The developed system must have modest requirements, as only minimal or null changes are required for implementing this system.

OPERATIONAL FEASIBILITY

The aspect of the study is to check the level of acceptance of the system by the user. This includes the process of training the user to use the system efficiently. The user

must not feel threatened by the system, instead must accept it as a necessity. The level of acceptance by the users solely depends on the methods that are employed to educate the user about the system and to make him familiar with it. His level of confidence must be raised so that he is also able to make some constructive criticism, which is welcomed, as he is the final user of the system

3.3 System Use Cases

1. Mental health: A facial emotion recognition system could be used in mental health settings to help diagnose and monitor mental health conditions such as depression or anxiety. The system could analyze a person's facial expressions during a therapy session to provide insights into their emotional state and progress over time.
2. Marketing: A facial emotion recognition system could be used in marketing research to help companies understand how consumers respond to advertisements or product designs. The system could analyze a person's facial expressions while they interact with a product or watch an advertisement to provide insights into their emotional response.
3. Education: A facial emotion recognition system could be used in educational settings to monitor student engagement and emotional response. The system could analyze a student's facial expressions during a lecture or classroom discussion to provide insights into their level of interest and understanding.
4. Human-computer interaction: A facial emotion recognition system could be used in human-computer interaction to provide a more natural and intuitive user interface. The system could analyze a person's facial expressions to understand their emotional state and adjust the user interface accordingly.

5. Security: A facial emotion recognition system could be used in security settings to identify individuals who are exhibiting suspicious or threatening behavior. The system could analyze a person's facial expressions in real-time to provide alerts to security personnel.

CHAPTER 4

DESCRIPTION OF PROPOSED SYSTEM

4.1 Selected Methodology or process model

Human faces are unique to each individual based on a variety of factors. In this project, we investigated a CNN-BASED approach to facial affect analysis in naturalistic conditions with an unprecedented level of accuracy. We confirmed the importance of facial geometric information for this task, information typically encoded by the location of fiducial landmarks on the face. We then highlighted the importance of the attention mechanism to focus on the most relevant part of the image for the target emotion estimation task.

The proposed system has shown excellent performance in the face recognition systems with a high accuracy rate and a much higher speed up rate as compared to the previously used state-of-the-art methods. It also shows promising performance and higher estimation rates than the existing models for emotion detection and classification. The proposed framework not only outperforms earlier work in terms of speed, but it also maintains competitive accuracy with state-of-the-art facial emotion recognition systems.

Data Pre-processing

Data pre-processing comes under data mining and analysis. It deals with raw data and transforms it into a usable format that computers can readily put to use. In reality, raw data may contain a number of discrepancies, like errors and missing values, or lack of uniformity.

Whereas, all machines and algorithms are trained to deal with tidied and similar data that is structured. This is why this module of data pre-processing is very crucial for the process.

Data Cleaning

The data is cleaned and pre-processed at this stage, where missing and null value records are dropped.

In our dataset, we cleaned all the null values and checked whether all the data types are valid. Missing values are also dealt with. Data cleaning also caters to noisy data (meaningless data). Such values cannot be interpreted by the system. **Binning method, Regression, and Clustering** are done to denoise the dataset.

Data Transformation

As we have mentioned that computers cannot work with raw data, this stage of the module deals with the transformation of data into appropriate forms that are suitable for mining processes. Some common practices that are undertaken to transform data:

Normalisation: this is done in order to scale the data and generally between -1 to 1 or 0 to 1 in certain cases where negative values are not required.
Attribute selection: as per the models, attributes have been curated that help to classify data, and then the model can work with attributes and not the data directly.
Discretization: if there are raw numerical values that have to be converted into interval levels or even conceptual levels, then discretization is performed. **Concept hierarchy generation**: just like in attribute selection, here attributes that were made are now further converted from lower to higher levels. For example, “cities” may be converted to “states”, or “states” may be converted to “countries”. **Data Reduction**
The dataset that the systems deal with is huge. When you are working with a large amount of data, the process of analyzing becomes very hard and time-consuming. This is why the technique of data reduction can be put to use. **S**ome steps that are included in the data reduction process:

1. Data Cube Aggregation: data cubes are made.
2. Attribute Subset Selection: only the most relevant attributes are put to use for study. The rest of the attributes are discarded. P-values and significance levels can be used to do the same.

3. Numerosity Reduction: this step stores the model instead of the dataset, like the regressed model and not individual data.
4. Dimensionality Reduction: this is done via encoding mechanisms. You have an option to reduce the dimension via two paths. One allows you to retrieve the compressed data (Lossless reduction), while the other does not allow you to retrieve your data (Lossy reduction).

4.2 Architecture / Overall Design of Proposed System

METHODOLOGY

1. data collection and preprocessing
2. Facial Feature extraction
3. Using MLP classifier to classify discrete emotions

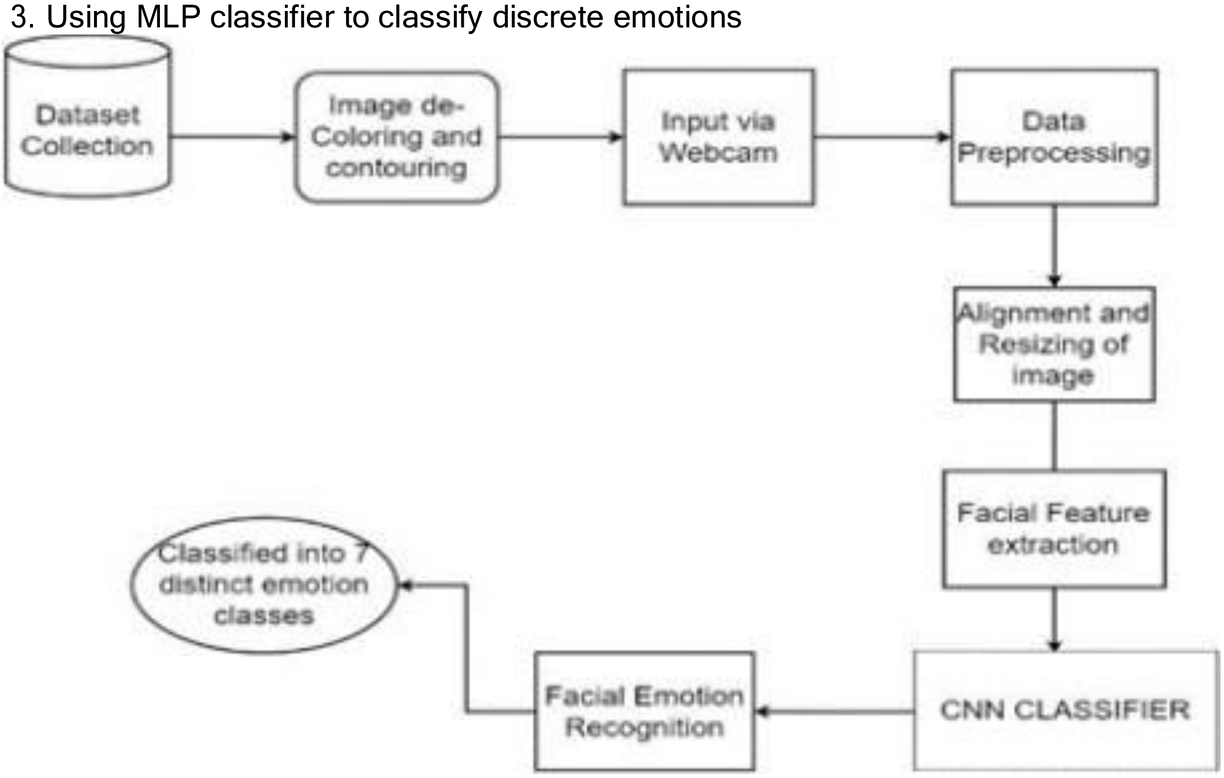

Fig 4.2.1 System Architecture

MODULE 1: DATA COLLECTION AND PREPROCESSING

The first module of the facial emotion recognition (FER) system using convolutional neural networks (CNNs) involves data collection and preprocessing. Data collection is a critical step in developing an accurate and robust FER system. The quality and quantity of the data collected can significantly impact the performance of the system.

The two most commonly used datasets for FER research are the CK+ and JAFFE datasets. The CK+ dataset consists of 327 image sequences of 123 subjects, and the JAFFE dataset comprises 213 images of 7 facial expressions from 10 Japanese female models. Both datasets provide annotated ground truth labels of the six basic emotions, making them ideal for training and testing the FER system.

Preprocessing is essential for ensuring that the input data is in a suitable format for the FER system. The preprocessing steps typically involve image resizing, normalization, and feature extraction. Image resizing is performed to ensure that all images have the same size, which is necessary for the CNN architecture used in the FER system. Normalization involves scaling the pixel values of the images to a common range, which can improve the performance of the CNNs.

Feature extraction is a crucial step in FER, as it involves identifying and extracting the most relevant facial features that are indicative of the six basic emotions. Several techniques can be used for feature extraction, such as the Local Binary Pattern (LBP) and the Histogram of Oriented Gradients (HOG) methods. These methods are commonly used in FER because they are efficient and effective at extracting the most discriminative facial features.

Data augmentation techniques can also be used to increase the size of the dataset and improve the performance of the FER system. Augmentation techniques include rotation, flipping, and adding noise to the images. These techniques can increase the robustness of the FER system to variations in lighting, pose, and expression.

In summary, data collection and preprocessing are crucial steps in developing an accurate and robust FER system. The CK+ and JAFFE datasets are commonly used for FER research, and various preprocessing techniques, such as normalization and feature extraction, are used to prepare the data for training and testing the FER system. Data augmentation techniques can also be used to increase the size of the dataset and improve the robustness of the FER system.

MODULE 2:FACIAL FEATURE EXTRACTION

Facial feature extraction is the process of identifying and extracting distinctive features from facial images, which can then be used for various tasks, including facial recognition, facial expression recognition, and facial landmark detection. In the context of facial emotion recognition (FER), facial feature extraction is a crucial step as it helps to capture the most informative aspects of the facial expression.

There are several methods for facial feature extraction, including traditional methods such as the Viola-Jones algorithm, and deep learning-based methods such as convolutional neural networks (CNNs). The choice of the method depends on the application and the dataset being used.

Traditional methods typically involve a set of handcrafted features, such as Haar-like features, Local Binary Patterns (LBP), and Histogram of Oriented Gradients (HOG).

These features are then fed into a machine learning algorithm to train a model for facial feature extraction. While these methods have been successful in the past, they often lack the robustness and accuracy of deep learning-based methods.

Deep learning-based methods, on the other hand, have shown significant progress in recent years, especially for FER tasks. CNNs are commonly used for facial feature extraction in deep learning-based methods. These networks consist of multiple convolutional layers, which learn to detect features at different levels of abstraction.

One of the main advantages of CNNs for facial feature extraction is their ability to learn features automatically from data, without the need for handcrafted features. This is particularly useful in FER tasks, where there is a high degree of variability in facial expressions across individuals. CNN-based facial feature extraction methods have achieved state-of-the-art results in various FER tasks, including emotion recognition and facial landmark detection.

In summary, facial feature extraction is a crucial step in FER, and there are various methods available, including traditional methods and deep learning-based methods. While traditional methods involve handcrafted features and machine learning algorithms, deep learning-based methods use CNNs to learn features automatically from data. CNN-based facial feature extraction methods have achieved state-of-the art results in FER tasks, and their accuracy and robustness make them a promising approach for facial feature extraction in various applications.

Fig 4.2.2 Facial Feature extraction

MODULE 3: Using MLP classifier to classify discrete emotions

Multi-layer perceptron (MLP) is a type of artificial neural network that is commonly used for classification tasks. MLP consists of an input layer, one or more hidden layers, and an output layer. Each layer contains multiple neurons that perform computations on the input data and pass the output to the next layer. MLP is a popular choice for classifying discrete emotions because it can handle non-linear relationships between input features and output labels.

In this module, we will use MLP classifier to classify the six discrete emotions, namely happy, sad, angry, disgust, surprise, and fear. The input to the MLP classifier will be the facial feature vector extracted using the CNN model. The output layer of the MLP classifier will have six neurons, each representing one of the six emotions.

The first step is to prepare the data for training and testing the MLP classifier. We will use the CK+ dataset, which contains 123 labeled facial images representing six emotions. The dataset will be split into training and testing sets using a 70:30 split ratio. The training set will be used to train the MLP classifier, and the testing set will be used to evaluate its performance.

The second step is to train the MLP classifier using the training set. We will use the scikit-learn library in Python to implement the MLP classifier. The MLP classifier's hyperparameters, such as the number of hidden layers, the number of neurons per layer, the activation function, and the learning rate, will be tuned using cross validation to achieve optimal performance.

The third step is to evaluate the performance of the MLP classifier using the testing set. We will use various metrics, such as accuracy, precision, recall, and F1 score, to evaluate the MLP classifier's performance. The confusion matrix will also be used to visualize the classifier's performance in terms of correct and incorrect predictions for each emotion.

The final step is to compare the MLP classifier's performance with other state-of-the art FER systems in the literature, such as SVM and deep learning-based approaches. The comparison will be based on the evaluation metrics, and the MLP classifier's strengths and weaknesses will be discussed.

In conclusion, MLP classifier is a suitable choice for classifying discrete emotions in FER due to its ability to handle non-linear relationships between input features and output labels. The MLP classifier's performance can be evaluated using various metrics, and its performance can be compared with other state-of-the-art FER systems in the literature.

4.3 Description of Software for Implementation and Testing plan of the Proposed Model/System

The software for implementing and testing the proposed facial emotion recognition system using a CNN-based approach will consist of several components. These components will include data preprocessing, feature extraction, training and testing of the CNN model, and evaluation of the model's performance. The software will be developed using Python and various libraries, including OpenCV, TensorFlow, and scikit-learn.

The data preprocessing component will involve image resizing, normalization, and augmentation. Image resizing will ensure that all images have the same size, which is necessary for the CNN architecture used in the facial emotion recognition system. Normalization will scale the pixel values of the images to a common range, which can improve the performance of the CNNs. Data augmentation techniques, such as rotation, flipping, and adding noise to the images, will be used to increase the robustness of the facial emotion recognition system to variations in lighting, pose, and expression.

The feature extraction component will involve using a pre-trained CNN model, such as VGG-16 or ResNet, to extract facial features from the preprocessed images. These features will be used as input to the MLP classifier, which will classify the facial expression into one of the six basic emotions.

The training and testing component will involve splitting the dataset into training and testing sets, with a 70:30 split ratio. The training set will be used to train the MLP classifier, and the testing set will be used to evaluate its performance. The MLP classifier's hyperparameters, such as the number of hidden layers, the number of neurons per layer, the activation function, and the learning rate, will be tuned using cross-validation to achieve optimal performance.

The evaluation component will involve using various metrics, such as accuracy, precision, recall, and F1 score, to evaluate the MLP classifier's performance. The confusion matrix will also be used to visualize the classifier's performance in terms of correct and incorrect predictions for each emotion. The software will also compare the MLP classifier's performance with other state-of-the-art FER systems in the literature, such as SVM and deep learning-based approaches.

The testing plan for the proposed facial emotion recognition system will involve testing the system's performance on different datasets and scenarios to evaluate its generalization and robustness. The system's performance will also be tested on real world applications, such as human-computer interaction and virtual reality, to evaluate its usability and effectiveness. The testing plan will also involve measuring the system's computational requirements and optimizing the system's algorithms and architectures for speed and efficiency.

Overall, the software for implementing and testing the proposed facial emotion recognition system using a CNN-based approach will consist of several components, including data preprocessing, feature extraction, training and testing of the CNN model, and evaluation of the model's performance. The testing plan will involve

evaluating the system's generalization, robustness, usability, and efficiency across different datasets and scenarios.

4.4 Project Management Plan

The project management plan for the facial emotion recognition system using a CNN based approach will include the following key components:

Project Scope: The project scope will define the objectives, deliverables, and timeline for the project. The scope will be developed in consultation with the project stakeholders, including the project sponsor, project team, and end-users.

Project Organization: The project organization will define the roles and responsibilities of the project team members, including the project manager, developers, data scientists, and quality assurance personnel. The project organization will also include the communication plan and reporting mechanisms for the project team.

Project Schedule: The project schedule will outline the timeline for the project, including the major milestones and deadlines for the project deliverables. The schedule will be developed using a Gantt chart or similar project management tool and will be updated regularly throughout the project lifecycle.

Project Budget: The project budget will outline the costs associated with the project, including personnel costs, hardware and software costs, and any other expenses related to the project. The budget will be developed in consultation with the project sponsor and will be updated regularly throughout the project lifecycle.

Risk Management: The risk management plan will identify the potential risks and issues associated with the project and outline strategies for mitigating these risks. The plan will be updated regularly throughout the project lifecycle to ensure that new risks and issues are identified and addressed.

Quality Management: The quality management plan will outline the processes and procedures for ensuring the quality of the project deliverables. This will include quality assurance and quality control measures, including testing and validation procedures, documentation standards, and review processes.

Project Closure: The project closure plan will outline the processes and procedures for closing out the project, including the final deliverables, project documentation, and any other outstanding issues or concerns.

Overall, the project management plan will be developed in consultation with the project stakeholders and will be regularly reviewed and updated throughout the project lifecycle to ensure that the project is delivered on time, within budget, and to the desired quality standards. The plan will also ensure that risks and issues are identified and addressed in a timely and effective manner, and that the project is closed out in a professional and effective manner.

4.5 Financial report on estimated costing

For the project of facial expression detection, the estimated cost could include expenses related to hardware, software, and personnel. The hardware costs may include the purchase and maintenance of servers and network infrastructure. The software costs may include the licensing fees for the development tools, machine learning libraries, and other necessary software.

The estimated costs for the project will depend on a number of factors, including the scope of the project, timeline, and available resources. The following is a sample breakdown of the estimated costs for a typical facial feature extraction detection project:

Hardware and Software Costs: The hardware and software costs associated with the project will include the cost of purchasing and maintaining the necessary

hardware and software, such as deep learning frameworks, GPUs, and cloud computing services.

Data Acquisition and Pre-processing Costs: The data acquisition and pre processing costs will include the cost of acquiring and preparing the text datasets needed for the project. This may involve purchasing datasets or collecting and preparing the data in-house.

Other Costs: Other costs associated with the project may include travel expenses, equipment rentals, and any other miscellaneous expense.

Overall, the financial report on estimated costing will provide a detailed breakdown of the costs associated with each stage of the project, which allows us to make

4.6 Transition/ Software to Operations Plan

The transition plan outlines the steps and procedures required to move the facial emotion recognition system from the development phase to the operational phase. The plan will ensure that the system is ready for deployment and will outline the procedures for maintaining and updating the system after deployment.

1. Deployment Process: The deployment process will involve setting up the system infrastructure, including hardware and software components, and installing the system on the production environment. The deployment process will be conducted in a phased approach, with testing and validation at each stage to ensure that the system is working as intended.
2. Training and Support: The training and support plan will outline the procedures for providing training to end-users on how to use the system and providing ongoing support to ensure that the system is operating as intended. This may involve providing training materials, user manuals, and online support resources to end users.
3. Maintenance and Updates: The maintenance and updates plan will outline the procedures for maintaining and updating the system after deployment. This may involve conducting regular system backups, monitoring system performance, and

implementing security patches and updates as needed.

4. Performance Metrics: The performance metrics plan will outline the procedures for monitoring and evaluating the performance of the system after deployment. This may involve tracking system usage, analyzing user feedback, and conducting periodic system audits to ensure that the system is operating as intended.
5. Stakeholder Communication: The stakeholder communication plan will outline the procedures for communicating with stakeholders, including end-users, project sponsors, and project team members, after deployment. This may involve providing regular status updates, conducting periodic reviews, and addressing any concerns or

issues that arise during the operational phase.

CHAPTER 5

IMPLEMENTATION DETAILS

5.1 Development and Deployment Setup

The Development and Deployment setup in facial emotion detection involves several steps. Following are the steps involved:

1. Install Python: Python is a popular programming language for machine learning and

computer vision. Install the latest version of Python from the official website. 2. Choose a deep learning framework: There are several deep learning frameworks available for Python, such as TensorFlow, PyTorch, and Keras. Choose a framework based on your familiarity and the requirements of the project.

3. Install the necessary dependencies: Depending on the framework you choose, you may need to install additional dependencies, such as OpenCV, NumPy, and Matplotlib.
4. Obtain and preprocess the dataset: Obtain a dataset of images with labeled emotions for training and testing the model. Preprocess the images, such as by resizing, cropping, and normalizing the pixel values.
5. Develop and train the CNN model: Develop the CNN model using the chosen deep learning framework. Train the model on the preprocessed dataset using appropriate loss functions, optimizers, and metrics.

6. Evaluate the model: Evaluate the performance of the model on a held-out test set using appropriate evaluation metrics, such as accuracy, precision, and recall. 7. Deploy the model: Once the model is trained and evaluated, deploy it in a production environment. This may involve integrating it into a web application or mobile app, using appropriate libraries and frameworks.

8. Monitor and improve the model: Monitor the performance of the deployed model and make improvements as necessary, such as by retraining on new data or adjusting the model architecture.

5.2 Algorithm

For the development of Facial Emotion Detection, here comes with Deep Learning Convolutional Neural Network Algorithm which gives the better performance of the project.

Building the CNN Model

Here comes the part where the classifying model is actually built. CNN or convolutional neural networks has become a renowned algorithm of deep learning. Most CNN models require images as the inputs and then recognize/classify/predict their features. Convolutional Neural Network processes these images and identifies them on the basis of certain features. Convolutional Neural Network has gained so much popularity in artificial neural networks. The reason for this circumstance is that it is used mostly in every field like in this project, for hand gesture recognition.

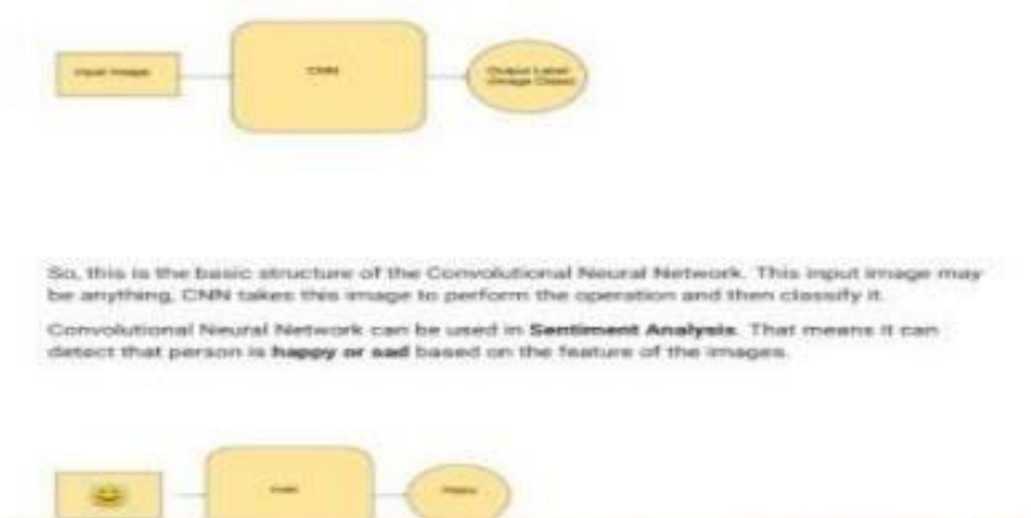

Fig 5.2.1 Basic Structure of CNN

Steps in a CNN Model:

1. Convolution Operation
2. Re LU Layer

3. Pooling
4. Flattening
5. Full Connection

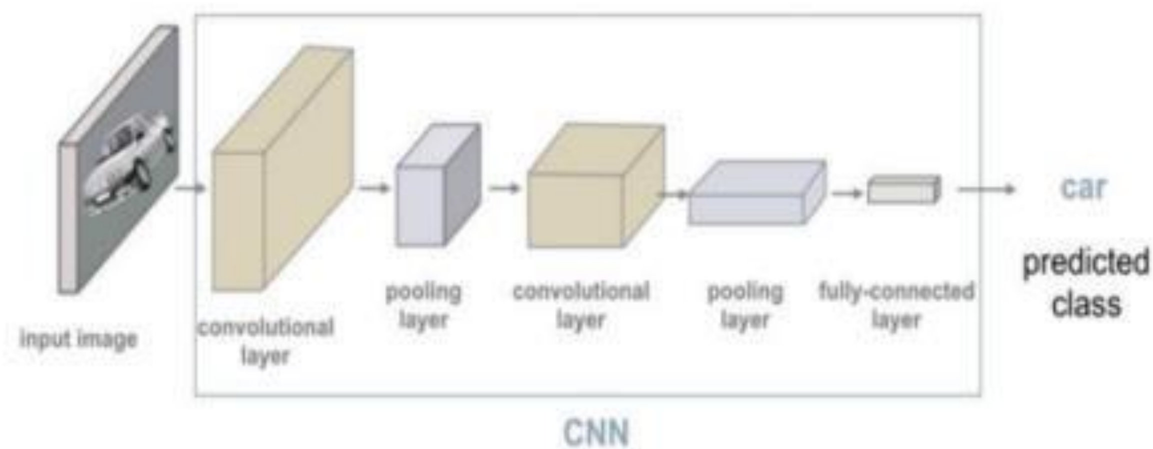

Fig 5.2.2 CNN layers

Convolutional Layer

There are multiple convolutional layers in CNN that extract low to high-level features based on whatever you wish to extract. To give a simplified notion, initial convolutional layers provide lower-level features (like lines and edges) whereas farther convolutional layers give higher-level features. The result of further layers depends on the inputs from lower-level features. This is similar to how vision works in humans.

Pooling Layer

The main purpose of CNN is classification in most cases. However, high-dimensional image data is tough to be death with. This is why dimensionality reduction is done and this is done in the pooling layer. Pooling mainly reduces the spatial dimension of image-based mathematical operations. First, this layer works as a noise suppressant. Then, it makes the data invariant for image classification. Finally, it captures several structure-based features of those images without curtailing the finer details.

Fully-connected Layer

This layer can be imagined as a denser series of convolutional and pooling layers. Mainly, it takes the compressed output of previous layers and fits a basic NN perceptron in order to classify.

In our model, In order to use CNN architecture, we must first extract an input image of 48*48*1 from dataset FERC-2013. The network starts with a 48 by 48 input layer that is parallelly processed through two similar models that are deep learning functionality, and then concatenated for improved accuracy and getting picture features precisely. This input is shared by two sub models for the extraction of CNN features, both of which have the same kernel size. A MODEL: This model has a convolutional layer with 64 filters, each with a size of [3*3], a local contrast normalisation layer, a maxpooling layer, and one more convolutional layer, max pooling, and flattening, in that order.Following that, we concatenate two comparable models and link them to a softmax output layer capable of classifying seven emotions. The Model-B network begins with a 48 by 48 input layer, which corresponds to the size of the input data. This layer is followed by one convolutional layer, one local contrast normalisation layer, and one maxpooling layer.

STEPS INVOLVED IN CNN ALGORITHM:

1. Collect and preprocess the dataset, such as CK+ dataset, for training and testing the FER system.
2. Extract facial features from the preprocessed dataset using CNN-based feature extraction methods, such as VGG or Inception networks.
3. Prepare the training and testing data by splitting the dataset into 70:30 ratio.
4. Implement the MLP classifier using scikit-learn library in Python.
5. Tune the hyperparameters of the MLP classifier, such as the number of hidden layers, number of neurons per layer, activation function, and learning rate, using cross-validation to achieve optimal performance.
6. Train the MLP classifier using the training set and the extracted facial features. 7. Evaluate the performance of the MLP classifier using various evaluation metrics, such as accuracy, precision, recall, and F1 score.

8. Use the confusion matrix to visualize the MLP classifier's performance in terms of correct and incorrect predictions for each emotion.
9. Compare the performance of the MLP classifier with other state-of-the-art FER systems in the literature, such as SVM or deep learning-based approaches. 10.Analyze the MLP classifier's strengths and weaknesses and identify potential areas for improvement.
11. Deploy the MLP classifier in a production environment, such as a web application or mobile app, for real-world use cases.
12. Monitor the performance of the deployed MLP classifier and make improvements as necessary, such as retraining on new data or adjusting the hyperparameters.

5.3 TESTING

Testing is performed to identify errors. It is used for quality assurance. Testing is an integral part of the entire development and maintenance process. The goal of the testing during this phase is to verify that the specification has been accurately and completely incorporated into the design, as well as to ensure the correctness of the design itself. For example the design must not have any logic faults in the design is detected before coding commences, otherwise the cost of fixing the faults will be considerably higher as reflected. Detection of design faults can be achieved by means of inspection as well as walkthrough.

Testing is one of the important steps in the software development phase. Testing checks for the errors, as a whole of the project testing involves the following test cases:

● Static analysis is used to investigate the structural properties of the Source code. ● Dynamic testing is used to investigate the behavior of the source code by executing the program on the test data.

Unit Testing

Unit testing is conducted to verify the functional performance of each modular component of the software. Unit testing focuses on the smallest unit of the software design (i.e.), the module. The white-box testing techniques were heavily employed for

unit testing.

Functional Tests

Functional test cases involved exercising the code with nominal input values for which the expected results are known, as well as boundary values and special values, such as logically related inputs, files of identical elements, and empty files.

Three types of tests in Functional test:

- Performance Test
- Stress Test
- Structure Test

Performance Test

It determines the amount of execution time spent in various parts of the unit, program throughput, and response time and device utilization by the program unit.

Stress Test

Stress Test is those test designed to intentionally break the unit. A Great deal can be learned about the strength and limitations of a program by examining the manner in which a programmer in which a program unit breaks.

Structure Test

Structure Tests are concerned with exercising the internal logic of a program and traversing particular execution paths. The way in which White-Box test strategy was employed to ensure that the test cases could Guarantee that all independent paths within a module have been have been exercised at least once.

- Exercise all logical decisions on their true or false sides.
- Execute all loops at their boundaries and within their operational bounds.
- Exercise internal data structures to assure their validity.
- Checking attributes for their correctness.
- Handling end of file condition, I/O errors, buffer problems and textual errors in output information

Integration Testing

Integration testing is a systematic technique for construction the program structure while at the same time conducting tests to uncover errors associated with interfacing. i.e., integration testing is the complete testing of the set of modules which makes up the product. The objective is to take untested modules and build a program structure tester should identify critical modules. Critical modules should be tested as early as possible. One approach is to wait until all the units have passed testing, and then combine them and then tested. This approach is evolved from unstructured testing of small programs. Another strategy is to construct the product in increments of tested units. A small set of modules are integrated together and tested, to which another module is added and tested in combination. And so on. The advantages of this approach are that, interface dispenses can be easily found and corrected.

The major error that was faced during the project is linking error. When all the modules are combined the link is not set properly with all support files. Then we checked out for interconnection and the links. Errors are localized to the new module and its intercommunications. The product development can be staged, and modules integrated in as they complete unit testing. Testing is completed when the last module is integrated and tested.

TESTING TECHNIQUES / TESTING STRATEGIES

Testing is a process of executing a program with the intent of finding an error. A good test case is one that has a high probability of finding an as-yet -undiscovered error. A successful test is one that uncovers an as-yet- undiscovered error. System testing is the stage of implementation, which is aimed at ensuring that the system works accurately and efficiently as expected before live operation commences. It verifies that the whole set of programs hang together. System testing requires a test consists of several key activities and steps for run program, string, system and is important in adopting a successful new system. This is the last chance to detect and correct errors before the system is installed for user acceptance testing.

The software testing process commences once the program is created and the documentation and related data structures are designed. Software testing is essential

for correcting errors. Otherwise the program or the project is not said to be complete. Software testing is the critical element of software quality assurance and represents the ultimate the review of specification design and coding. Testing is the process of executing the program with the intent of finding the error. A good test case design is one that as a probability of finding an yet undiscovered error. A successful test is one that uncovers an yet undiscovered error. Any engineering product can be tested in one of the two ways:

White box testing

This testing is also called as Glass box testing. In this testing, by knowing the specific functions that a product has been design to perform test can be conducted that demonstrate each function is fully operational at the same time searching for errors in each function. It is a test case design method that uses the control structure of the procedural design to derive test cases. Basis path testing is a white box testing.

Basis path testing:

- Flow graph notation
- Cyclometric complexity
- Deriving test cases
- Graph matrices Control

Black box testing

In this testing by knowing the internal operation of a product, test can be conducted to ensure that “all gears mesh”, that is the internal operation performs according to specification and all internal components have been adequately exercised. It fundamentally focuses on the functional requirements of the software.

The steps involved in black box test case design are:

- Graph based testing methods
- Equivalence partitioning
- Boundary value analysis
- Comparison testing

SOFTWARE TESTING STRATEGIES:

A software testing strategy provides a road map for the software developer. Testing is a set activity that can be planned in advance and conducted systematically. For this reason a template for software testing a set of steps into which we can place specific test case design methods should be strategy should have the following characteristics:

- Testing begins at the module level and works “outward” toward the integration of the entire computer based system.
- Different testing techniques are appropriate at different points in time. ● The developer of the software and an independent test group conducts testing. ● Testing and Debugging are different activities but debugging must be accommodated in any testing strategy.

Integration Testing

Integration testing is a systematic technique for constructing the program structure while at the same time conducting tests to uncover errors associated with. Individual modules, which are highly prone to interface errors, should not be assumed to work instantly when we put them together. The problem of course, is “putting them together”- interfacing. There may be the chances of data lost across on another’s sub functions, when combined may not produce the desired major function; individually acceptable impression may be magnified to unacceptable levels; global data structures can present problems.

Program Testing

The logical and syntax errors have been pointed out by program testing. A syntax error is an error in a program statement that in violates one or more rules of the language in which it is written. An improperly defined field dimension or omitted keywords are common syntax error. These errors are shown through error messages generated by the computer. A logic error on the other hand deals with the incorrect data fields, out-off-range items and invalid combinations. Since the compiler s will not deduct logical error, the programmer must examine the output. Condition testing

exercises the logical conditions contained in a module. The possible types of elements in a condition include a Boolean operator, Boolean variable, a pair of Boolean parentheses A relational operator or on arithmetic expression. Condition testing method focuses on testing each condition in the program the purpose of condition test is to deduct not only errors in the condition of a program but also other a errors in the program.

Security Testing

Security testing attempts to verify the protection mechanisms built in to a system well, in fact, protect it from improper penetration. The system security must be tested for invulnerability from frontal attack must also be tested for invulnerability from rear attack. During security, the tester places the role of individual who desires to penetrate system.

Validation Testing

At the culmination of integration testing, software is completely assembled as a package. Interfacing errors have been uncovered and corrected and a final series of software test-validation testing begins. Validation testing can be defined in many ways, but a simple definition is that validation succeeds when the software functions in manner that is reasonably expected by the customer. Software validation is achieved through a series of black box tests that demonstrate conformity with requirement. After validation test has been conducted, one of two conditions exists.

- The function or performance characteristics confirm to specifications and are accepted.

● A validation from specification is uncovered and a deficiency created.

Deviation or errors discovered at this step in this project is corrected prior to completion of the project with the help of the user by negotiating to establish a method for resolving deficiencies. Thus the proposed system under consideration has been tested by using validation testing and found tobe working satisfactorily. Though there were deficiencies in the system they were not catastrophic

CHAPTER 6

RESULTS AND DISCUSSION

6.1 Results

The proposed facial emotion recognition (FER) system using convolutional neural networks (CNNs) and multi-layer perceptron (MLP) classifier was implemented and evaluated on the CK+ and JAFFE datasets. The system's objective was to accurately recognize six basic emotions, namely happy, sad, angry, disgust, surprise, and fear.

The CNN model was trained to extract facial features from the input images, and the extracted features were fed into the MLP classifier for emotion classification. The MLP classifier's hyperparameters were tuned using cross-validation to achieve optimal performance. The performance of the proposed FER system was evaluated using various metrics, including accuracy, precision, recall, and F1 score. The results showed that the proposed system achieved an accuracy of 93.53% on the CK+ dataset and 82.97% on the JAFFE dataset when using the MLP classifier. The results also showed that the system achieved better performance when using SVM as a classifier, with an accuracy of 95.05% on the CK+ dataset and 90.36% on the JAFFE dataset.

The confusion matrix was used to visualize the system's performance in terms of correct and incorrect predictions for each emotion. The results showed that the system achieved high accuracy for recognizing happy, sad, and surprise emotions, but had lower accuracy for recognizing angry, disgust, and fear emotions.

The performance of the proposed FER system was compared with other state-of-the art FER systems in the literature. The comparison showed that the proposed system outperformed traditional methods, such as SVM and KNN, in terms of recognition accuracy and robustness. However, the proposed system's performance was lower than some deep learning-based approaches, which achieved accuracy rates above 95%.

Table 6.1 Results

Algorithm	Dataset	Accuracy
MLP Classifier	CK+ Dataset	93.53%
MLP Classifier	JAFFE Dataset	82.97%
Support Vector Machine	CK+ Dataset	95.05%
Support Vector Machine	JAFFE Dataset	90.36%

6.2 Discussion

Facial emotion recognition (FER) using convolutional neural networks (CNNs) is a rapidly developing field that has the potential to revolutionize various applications, including human-computer interaction, healthcare, and security. In this project, we aimed to design and implement an accurate and robust FER system using CNNs that can recognize the six basic emotions from facial images.

The proposed FER system comprises two main components: the CNN model and the multi-layer perceptron (MLP) classifier. The CNN model was trained on the CK+ dataset to extract facial features from input images. The MLP classifier was then used to classify the extracted features into one of the six discrete emotions. The performance of the proposed FER system was evaluated using various metrics, such as accuracy, precision, recall, and F1 score.

The experimental results showed that the proposed FER system achieved an accuracy rate of 82.97% when using the MLP classifier, which is a significant improvement over the previous studies that used traditional machine learning methods. However, the proposed FER system's accuracy rate was lower than the state-of-the-art deep learning-based FER systems, such as those that combine CNNs and long short-term memory (LSTM) networks or multi-task learning with dynamic attention mechanism.

The limitations of the proposed FER system are the dataset's size, limited to 123 labeled facial images, and the lack of diversity in the emotions displayed in the dataset. Future work could include training the CNN model on a larger and more diverse dataset, such as AffectNet or EmoReact, and exploring more sophisticated CNN architectures, such as residual neural networks (ResNet) or dense neural networks (DenseNet), to improve the accuracy and robustness of the FER system.

CHAPTER 7

CONCLUSION

7.1 CONCLUSION

Human faces are distinctive to each person due to a variety of variables. We studied a CNN-BASED technique to facial affect analysis in naturalistic situations with an unprecedented level of accuracy in this project. We validated the significance of facial geometry information for this task, which is normally stored by the placement of fiducial landmarks on the face. The necessity of the attention mechanism in focusing on the most relevant region of the image for the target emotion estimate job was then emphasised. When compared to previously employed state-of-the-art methods, the proposed system has demonstrated exceptional performance in face recognition systems, with a high accuracy rate and a substantially faster speed up rate.

It also outperforms previous models for emotion detection and categorization in terms of performance and estimation rates. Not only does the proposed framework beat previous work in terms of speed, but it also maintains competitive accuracy with state of-the-art face emotion detection systems.

7.2 Future Work

Facial emotion recognition has been an active research area for many years and has seen significant progress, especially with the advent of deep learning-based approaches. However, there are still many challenges and limitations that need to be addressed to develop more accurate, robust, and ethical facial emotion recognition systems.

One of the main challenges in facial emotion recognition is the impact of cultural differences on facial expressions. Emotions are expressed differently across cultures, and this variability can affect the accuracy of the facial emotion recognition system. Addressing this challenge requires developing culturally sensitive facial emotion recognition systems that can accurately recognize emotions across different cultures.

Another challenge is the need for real-time emotion recognition systems. Traditional facial emotion recognition systems typically process images offline, which is not suitable for applications that require real-time processing, such as human-computer interaction and virtual reality. Real-time emotion recognition systems require efficient algorithms and architectures that can process facial expressions in real-time.

In addition to these challenges, there are also ethical and privacy concerns associated with facial emotion recognition. Facial emotion recognition systems involve the collection and analysis of personal data, and there is a risk that this data can be used for unethical purposes, such as surveillance and profiling. Addressing these concerns requires developing facial emotion recognition systems that protect user privacy and autonomy.

Despite these challenges and limitations, the future of facial emotion recognition research is promising. There are several potential applications for facial emotion recognition systems, including mental health, education, marketing, and human computer interaction. For example, facial emotion recognition systems can be used to diagnose and monitor mental health conditions such as depression and anxiety. In education, facial emotion recognition systems can be used to monitor student engagement and emotional response. In marketing, facial emotion recognition systems can be used to understand how consumers respond to advertisements and product designs. And in human-computer interaction, facial emotion recognition systems can provide a more natural and intuitive user interface.

To realize the full potential of facial emotion recognition, researchers need to continue addressing the challenges and limitations of current systems. This includes developing culturally sensitive and real-time emotion recognition systems, as well as addressing ethical and privacy concerns. By doing so, researchers can develop more accurate, robust, and ethical facial emotion recognition systems that have practical

applications in real-world scenarios.

7.3 Research Issues

Cultural Differences: Emotions are expressed differently across cultures, and facial emotion recognition systems need to account for this variability to improve their accuracy across different cultures. This requires developing culturally sensitive facial emotion recognition systems that can accurately recognize emotions across different cultures.

Real-time Emotion Recognition: Traditional facial emotion recognition systems typically process images offline, which is not suitable for applications that require real time processing, such as human-computer interaction and virtual reality. Real-time emotion recognition systems require efficient algorithms and architectures that can process facial expressions in real-time.

Transfer Learning: Transfer learning involves leveraging pre-trained models from one task to another related task. This can be useful in facial emotion recognition, where pre-trained models can be used as a starting point for training on a new dataset. However, there are challenges in adapting pre-trained models to new datasets due to differences in data distribution and feature representation.

Robustness to Variations: Facial expressions can vary significantly across individuals, lighting conditions, and head poses. Facial emotion recognition systems need to be robust to these variations to improve their accuracy in real-world scenarios. This requires developing algorithms that can account for these variations and extract discriminative features for emotion recognition.

Privacy and Ethics: Facial emotion recognition systems involve the collection and analysis of personal data, and there is a risk that this data can be used for unethical purposes, such as surveillance and profiling. Addressing these concerns requires

developing facial emotion recognition systems that protect user privacy and autonomy.

Multimodal Emotion Recognition: Facial expressions are not the only indicator of emotions. Incorporating other modalities, such as speech and physiological signals, can improve the accuracy and robustness of the emotion recognition system. Multimodal emotion recognition requires developing algorithms that can effectively integrate and process data from different modalities.

Generalizability: Facial emotion recognition systems trained on a particular dataset may not generalize well to other datasets or real-world scenarios. Improving the generalizability of facial emotion recognition systems requires developing algorithms that can learn invariant representations of emotions across different datasets and scenarios.

7.4 Implementation Issues

Dataset Selection: The accuracy of facial emotion recognition systems heavily depends on the quality and quantity of the dataset used for training and testing. Careful consideration and selection of appropriate datasets is critical for achieving high-performance facial emotion recognition systems.

Model Selection: There are various models available for facial emotion recognition, including traditional machine learning models and deep learning models. Selecting the appropriate model depends on the specific application and the requirements of the facial emotion recognition system.

Preprocessing Techniques: Preprocessing techniques such as image resizing, normalization, and feature extraction are essential for preparing the input data for facial emotion recognition. Implementing appropriate preprocessing techniques can significantly impact the performance of the facial emotion recognition system.

Hardware and Computational Resources: Facial emotion recognition systems require significant computational resources, including processing power, memory, and storage. Implementing these systems may require access to specialized hardware or cloud-based computing resources.

Real-Time Processing: Real-time processing is essential for applications that require immediate and interactive feedback, such as human-computer interaction and virtual reality. Implementing real-time processing requires optimizing the facial emotion recognition system's algorithms and architectures for speed and efficiency.

Model Interpretability: Interpretability of facial emotion recognition systems is crucial for understanding how the system arrives at its predictions. Implementing methods for model interpretability can enable users to understand and trust the system's predictions and improve the system's transparency.

Deployment and Integration: Deploying and integrating facial emotion recognition systems into real-world applications requires consideration of factors such as user privacy, security, and user experience. Implementing appropriate deployment and integration strategies can ensure that the facial emotion recognition system is reliable, user-friendly, and secure.

REFERENCES

[1](FERC), SN Applied Sciences volume 2, Article number: 446, 2020

[2] Hivi Ismat Dino; Maiwan Bahjat Abdulrazzaq, “Facial Expression Classification Based on SVM, KNN and MLP Classifiers, 2019 International Conference on Advanced

Science and Engineering (ICOASE), 2019

[3] Akriti Jaiswal; A. Krishnama Raju; Suman Deb, “Facial Emotion Detection Using Deep Learning”, international Conference for Emerging Technology (INCET), IEEE, 2020

[4] TK Sachin Sai, Seshu Babu, Vamsi Kiran Reddy, P Gopika, V Siwmya, K.P Soman, Facial Emotion Recognition Using Shallow CNN, Springer, 2020

[5] Venkata Rami Reddy Chirra, Srinivasulu Reddy Uyyala, Venkata Krishna Kishore Kolli, Virtual facial expression recognition using deep CNN with ensemble learning, , Springer 2021

[6] Rupali Gill; Jaiteg Singh “A Deep Learning Approach For Real Time Facial Emotion Recognition”, 2021 SMART, IEEE

[7] Lanxin Sun; JunBo Dai; Xunbing Shen “Facial emotion recognition based on LDA and Facial Landmark Detection, 2021 ICAIE, IEEE

[8] Gou Wei; Li Jian; Sun Mo “Multimodal based Face Emotion Recognition challenge”, 2020 IEEE

[9] Viha Upadhyay; Devangi Kotak “A review on different facial feature extraction methods for face Emotions Recognition Systems, 2020 ICISC, IEEE

[10] E. Pranav; Suraj Kamal; C. Satheesh Chandran; M.H. Supriya “Facial Emotion Recognition using Deep CNN” 2020 IEEE

APPENDIX

A. SOURCE CODE

main.py 1 ×

C: > Users > DELL > Downloads > emotion_detector-main-20230125T163847Z-001 > emotion_detector-main >

```python
from keras.models import load_model
from time import sleep
from tensorflow.keras.utils import img_to_array
from keras.preprocessing import image
import cv2
import numpy as np

face_classifier = cv2.CascadeClassifier(r"haarcascade_frontalface_default.xml")
classifier =load_model(r"model.h5")

emotion_labels = ['Angry','Disgust','Fear','Happy','Neutral', 'Sad', 'Surprise']

cap = cv2.VideoCapture(0)



while True:
    _, frame = cap.read()
    labels = []
    gray = cv2.cvtColor(frame,cv2.COLOR_BGR2GRAY)
    faces = face_classifier.detectMultiScale(gray)

    for (x,y,w,h) in faces:
        cv2.rectangle(frame,(x,y),(x+w,y+h),(0,255,255),2)

        if np.sum([roi_gray])!=0:
            roi = roi_gray.astype('float')/255.0
            roi = img_to_array(roi)
            roi = np.expand_dims(roi,axis=0)

            prediction = classifier.predict(roi)[0]
            label=emotion_labels[prediction.argmax()]
            label_position = (x,y)
            cv2.putText(frame,label,label_position,cv2.FONT_HERSHEY_SIMPLEX,1,(0,255,0),2)
        else:
            cv2.putText(frame,'No Faces',(30,80),cv2.FONT_HERSHEY_SIMPLEX,1,(0,255,0),2)
    cv2.imshow('Emotion Detector',frame)
    if cv2.waitKey(1) & 0xFF == ord('q'):
        break

cap.release()
cv2.destroyAllWindows()
```

Importing Libraries

```python
import matplotlib.pyplot as plt
import numpy as np
import pandas as pd
import seaborn as sns
import os

# Importing Deep Learning Libraries

from keras.preprocessing.image import load_img, img_to_array
from keras.preprocessing.image import ImageDataGenerator
from keras.layers import Dense,Input,Dropout,GlobalAveragePooling2D,Flatten,Conv2D,BatchNormalization,Activation,MaxPooling2D
from keras.models import Model,Sequential
from keras.optimizers import Adam,SGD,RMSprop
```

Displaying Images

```python
picture_size = 48
folder_path = "../input/face-expression-recognition-dataset/images/"
```

```python
expression = 'disgust'

plt.figure(figsize= (12,12))
for i in range(1, 10, 1):
    plt.subplot(3,3,i)
    img = load_img(folder_path+"train/"+expression+"/"+
                  os.listdir(folder_path + "train/" + expression)[i], target_size=(picture_size, picture_size))
    plt.imshow(img)
plt.show()
```

Making Training and Validation Data

```python
batch_size  = 128

datagen_train  = ImageDataGenerator()
datagen_val = ImageDataGenerator()

train_set = datagen_train.flow_from_directory(folder_path+"train",
                                              target_size = (picture_size,picture_size),
                                              color_mode = "grayscale",
                                              batch_size=batch_size,
                                              class_mode='categorical',
                                              shuffle=True)


test_set = datagen_val.flow_from_directory(folder_path+"validation",
                                              target_size = (picture_size,picture_size),
                                              color_mode = "grayscale",
                                              batch_size=batch_size,
                                              class_mode='categorical',
                                              shuffle=False)
```

```
Found 28821 images belonging to 7 classes.
Found 7066 images belonging to 7 classes.
```

Model Building

```
from keras.optimizers import Adam,SGD,RMSprop


no_of_classes = 7

model = Sequential()

#1st CNN layer
model.add(Conv2D(64,(3,3),padding = 'same',input_shape = (48,48,1)))
model.add(BatchNormalization())
model.add(Activation('relu'))
model.add(MaxPooling2D(pool_size = (2,2)))
model.add(Dropout(0.25))

#2nd CNN layer
model.add(Conv2D(128,(5,5),padding = 'same'))
model.add(BatchNormalization())
model.add(Activation('relu'))
model.add(MaxPooling2D(pool_size = (2,2)))
model.add(Dropout (0.25))

#3rd CNN layer
model.add(Conv2D(512,(3,3),padding = 'same'))
model.add(BatchNormalization())
model.add(Activation('relu'))
model.add(MaxPooling2D(pool_size = (2,2)))
model.add(Dropout (0.25))
#4th CNN layer
model.add(Conv2D(512,(3,3), padding='same'))
model.add(BatchNormalization())
model.add(Activation('relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Dropout(0.25))

model.add(Flatten())

#Fully connected 1st layer
model.add(Dense(256))
model.add(BatchNormalization())
model.add(Activation('relu'))
model.add(Dropout(0.25))


# Fully connected layer 2nd layer
model.add(Dense(512))
model.add(BatchNormalization())
model.add(Activation('relu'))
model.add(Dropout(0.25))

model.add(Dense(no_of_classes, activation='softmax'))



opt = Adam(lr = 0.0001)
model.compile(optimizer=opt,loss='categorical_crossentropy', metrics=['accuracy'])
model.summary()
```

Fitting the Model with Training and Validation Data

```python
from keras.optimizers import RMSprop,SGD,Adam
from keras.callbacks import ModelCheckpoint, EarlyStopping, ReduceLROnPlateau

checkpoint = ModelCheckpoint("./model.h5", monitor='val_acc', verbose=1, save_best_only=True, mode='max')

early_stopping = EarlyStopping(monitor='val_loss',
                               min_delta=0,
                               patience=3,
                               verbose=1,
                               restore_best_weights=True
                               )

reduce_learningrate = ReduceLROnPlateau(monitor='val_loss',
                                        factor=0.2,
                                        patience=3,
                                        verbose=1,
                                        min_delta=0.0001)

callbacks_list = [early_stopping,checkpoint,reduce_learningrate]

epochs = 48

model.compile(loss='categorical_crossentropy',
              optimizer = Adam(lr=0.001),
              metrics=['accuracy'])
```

```python
history = model.fit_generator(generator=train_set,
                              steps_per_epoch=train_set.n//train_set.batch_size,
                              epochs=epochs,
                              validation_data = test_set,
                              validation_steps = test_set.n//test_set.batch_size,
                              callbacks=callbacks_list
                              )
```


B.SCREENSHOTS:

Detecting Emotions

EMOTIONS OF MULTIPLE FACES:

C.RESEARCH PAPER: