DEEP LEARNING BASED FACIAL EMOTIONS RECOGNITION

Major Project Report Submitted in Partial Fulfillment of The Requirement For The Degree of Bachelor of Technology

in

Information Technology

Submitted by

Anmol Sharma(201547)

Sanskar Singhal(201546)

Under the guidance & supervision of

Dr. Diksha Hooda

(Assistant Professor , SG)

Department of Computer Science & Engineering and Information Technology

Jaypee University of Information Technology, Waknaghat, 173234, Himachal Pradesh, INDIA

CERTIFICATE

This is to certify that the work which is being presented in the project report titled "Deep Learning Based Facial Emotions Recognition" in partial fulfillment of the requirements for the award of the degree of B.Tech in Information Technology and submitted to the Department of Computer Science & Engineering And Information Technology, Jaypee University of Information Technology, Waknaghat is an authentic record of work carried out by "Anmol Sharma (201547) and Sanskar Singhal (201546)" during the period from January 2024 to May 2024 under the supervision of Dr. Diksha Hooda, Department of Computer Science and Engineering, Jaypee University of Information Technology, Waknaghat.

Anmol Sharma
201547
Sanskar Singhal
201546

The above statement made is correct to best of my knowledge.

Dr. Diksha Hooda
Assistant Professor (SG.)
Computer Science & Engineering and Information Technology
Jaypee University of Information Technology, Waknaghat

DECLARATION

I hereby declare that the work presented in this report entitled **'Deep Learning Based Facial Emotions Recognition'** in partial fulfillment of requirements for award of degree of **Bachelor of Technology** in **Information Technology** submitted in the Department of Computer Science & Engineering and Information Technology**,** Jaypee University of Information Technology, Waknaghat is an authentic record of my own work carried out over period from January 2024 to May 2024 under the supervision of **Dr. Diksha Hooda** (Assistant Professor SG, Department of Computer Science & Engineering and Information Technology).

The matter embodied in the report has not been submitted for the award of any other degree or diploma.

Student Name: Anmol Sharma
Roll No.: 201547

Student Name: Sanskar Singhal
Roll No.: 201546

This is to certify that the above statement made by the candidate is true to the best of my knowledge.

Supervisor Name: Dr. Diksha Hooda
Designation: Assistant Professor (SG.)
Department: Department of Computer Science & Engineering and Information Technology
Dated:

ACKNOWLEDGEMENT

Firstly, I express my heartiest thanks and gratefulness to almighty God for His divine blessing makes it possible to complete the project work successfully.

I am really grateful and wish my profound indebtedness to the Supervisor **Dr. Diksha Hooda, Assistant Professor (SG.)** Department of CSE Jaypee University of Information Technology,Wakhnaghat. Deep Knowledge & keen interest of my supervisor in the field of "**AI**" to carry out this project. Her endless patience, scholarly guidance, continual encouragement, constant and energetic supervision, constructive criticism, valuable advice, reading many inferior drafts and correcting them at all stages have made it possible to complete this project.

I would like to express my heartiest gratitude to **Dr. Diksha Hooda,** Department of CSE, for her kind help to finish this project.

I would also generously welcome each one of those individuals who have helped me straightforwardly or in roundabout way in making this project win. In this unique situation, I might want to thank various staff individuals, both educating and non-instructing, which have developed their convenient help and facilitated my undertaking.

Finally, I must acknowledge with due respect the constant support and patients of my parents.

Anmol Sharma
201547
Sanskar Singhal
201546

TABLE OF CONTENT

LIST OF TABLES

LIST OF FIGURES

ABSTRACT

Emotions are fundamental to human communication with facial expressions serving as key nonverbal cues. Facial Emotion Recognition powerful tool mirroring human coding abilities is vital for real time emotion understanding. Its applications span diverse fields, including medicine and computer vision. Deep learning facilitates emotion recognition systems but factors like algorithm choice and hyperparameter tuning significantly impact accuracy.

This project focuses on development of a Deep Learning-based Facial Emotion Recognition system. Emotion recognition is critical component in various applications, including mental health monitoring human-computer interaction and virtual communication. This project utilizes Convolutional Neural Network (CNN) architecture to effectively recognize facial expressions corresponding to different emotions. Dataset used for training and evaluation is "fer2013," which consists of images categorized into seven emotions: Angry, Disgust, Fear, Happy, Neutral, Sad, and Surprise. This project follows systematic approach including data preparation, model architecture design, training, and evaluation. Data augmentation techniques are applied to enhance the model's ability to generalize by introducing variations in training dataset. The CNN architecture comprises multiple convolutional and pooling layers followed by fully connected layers. Model is trained using Adam optimizer and categorical crossentropy loss. Training process is monitored using various metrics, including accuracy, loss, and early stopping to prevent overfitting. Evaluation phase involves assessing the model's performance using confusion matrices, classification reports, and ROC-AUC curves.

This project's results demonstrate model's effectiveness in recognizing facial expressions across different emotions. Classification report provides insights into precision, recall, and F1-score for each emotion class. The ROC-AUC curve offers additional perspective on model's performance.

In conclusion, we developed Facial Emotion Recognition system exhibits promising results, laying foundation for applications in real-time video calls and online mental health consultancy. The project also outlines avenues for future enhancements, such as deploying model in real-time scenarios and expanding dataset for improved diversity.

CHAPTER 1
INTRODUCTION

1.1 INTRODUCTION

Emotions represent mental responses of body to specific events or actions, manifesting through both behavioral and physiological changes. Human sensory perception plays crucial role in identifying fundamental emotions like anger, fear, disgust, sadness, happiness, and surprise relying on input data. Facial expressions as highlighted by research [17], are instrumental in conveying these emotions, with each individual displaying unique expressions. According to Mehrabian [25], communication involves conveyance of messages through various channels where words contribute 7%, voice intonation conveys 38%, and facial expressions account for 55% of message. This underscores significant role of facial emotions in communication. Practical implications of emotion recognition extend across diverse fields such as healthcare, entertainment, online learning, marketing, human monitoring, and security. In this context, machine learning (ML) emerges as vital tool. ML involves study of algorithms designed to enhance performance, predict data, autonomously learn from experiences, and improve accuracy through utilization of data. Its applications are particularly valuable in accurately detecting and recognizing features, contributing to advancements in various domains.

Apply algorithms to various domains, as machine learning (ML) finds extensive use in fields like medicine, computer vision, speech recognition, and image recognition. Among myriad applications of ML, one of most noteworthy is its role in recognizing human emotions. ML algorithms play pivotal role in development of emotion recognition models, allowing for precise training to accurately identify human emotions. Deep learning (DL), subset of ML, stands out as particularly impactful branch in this context. It leverages artificial neural networks to execute specific engineering reflections, essentially mirroring processes of human brain within machine. DL empowers computer models with multiple processing layers to assimilate vast amounts of knowledge, learning and training machines to comprehend data and its representation across multiple levels of abstraction. In granular explanation, DL techniques facilitate sequential learning of categories due to architecture of hidden layers. Each neuron or node within network represents a distinct component of whole, collectively forming comprehensive visual representation. Nodes or hidden layers are assigned weights that signify their degree of association with outcome and these weights dynamically change as model evolves [18]. DL algorithms employ integer data to extract meaningful features, identify groups of objects, and discern data patterns. Various DL algorithms, including Convolutional Neural Networks (CNN), Short Short Memory (LSTM), and Restricted Boltzmann Machines (RBM) are instrumental in real-time problem-solving. DL operates with diverse data formats such as images, videos, text, conversations, and speeches. An advantageous characteristic of DL is its ability to bypass manual feature extraction, streamlining process of using algorithms to extract crucial features.The CNN algorithm plays crucial role in analysis of images for emotion detection, employing feature extraction and image classification based on facial features to discern human emotions. This thesis endeavors to construct deep learning (DL) model by utilizing CNN algorithm and multi-layer classifier. Thia primary objective is to recognize emotions based on facial features. To enhance the model's accuracy, hyperparameter tuning through Bayesian optimization will be employed, coupled with utilization of sizable and accurate training dataset, FER2013, which is derived from Kaggle, comprises grayscale images of human facial emotions with dimensions of 48*48 [4]. These images are categorized into seven emotion types: Anger, Disgust, Fear, Happiness, Sadness, Surprise, and Neutral, using standard classifier. Classified dataset will be partitioned into training set and test set. The training set will be instrumental in training model, while test set will assess the model's performance, leveraging CNN algorithm.

1.2 PROBLEM STATEMENT

1.2.1 PROBLEM DEFINITION

The categorization of human facial expressions into seven fundamental emotions, namely happy, sad, surprised, scared, angry, disgusted, and neutral relies on activation of specific sets of facial muscles. These intricate signals, though sometimes subtle, encapsulate wealth of information about individual's state of mind. Facial emotion recognition serves as valuable tool, enabling measurement of impact of content and services on audiences or users through straightforward and cost-effective process. This capability holds various applications such as retailers using these metrics to gauge customer interest, healthcare providers enhancing services by incorporating additional information about patient's emotional state during treatment, and entertainment producers monitoring audience participation at events to consistently tailor content to desired preferences. While humans are inherently skilled at reading others' emotions, question arises: can computers surpass our ability to access emotional states? To address this query, we embarked on design of deep learning neuron, exploring potential for computational systems to excel in understanding and interpreting emotional states.

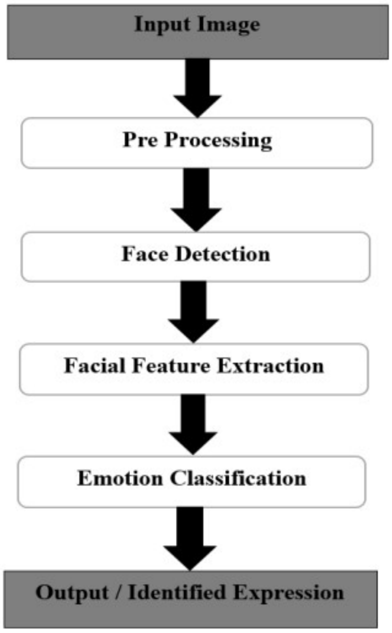


Fig1. Flow Chart

1.2.2. PROBLEM STATEMENT

Within realm of online mental health counseling, precise recognition of facial emotions holds significant potential. This capability stands to benefit mental health professionals by providing them with deeper understanding of their clients' emotions and intentions, thereby fostering more effective communication and treatment strategies. The integration of facial emotion recognition technology into online platforms presents opportunity to personalize and tailor mental health counseling to individual needs. Such technological integration has potential to elevate overall quality of care and support offered to individuals seeking mental health assistance. By harnessing the insights gained from facial emotion recognition, mental health professionals could enhance their ability to connect with clients, fostering more empathetic and tailored approach to therapy. This advancement aligns with evolving landscape of mental health services, aiming to leverage technology for improved outcomes and more personalized therapeutic experience.

1.3 OBJECTIVES

The main objectives of this study are:

- ➔ This project focuses on developing deep learning testing model to identify emotional states on human faces using facial features using TensorFlow.
- ➔ Develop Deep Learning model using convolutional networks as well as multi-class classification and FER 2013 dataset
- ➔ Fine-tune hyperparameters with Bayesian optimization along with appropriate training size datasets

1.4 SIGNIFICANCE AND MOTIVATION OF PROJECT WORK

The world's most famous painting, Mona Lisa, has been much-discussed topic in past. According to British Weekly's "New Scientist", she was 83% happy, 9% angry, 6% scared and 2% angry. We are also inspired by observing benefits for people with physical disabilities such as deaf and mute. But if normal person or an automated system could see his facial expressions, it would be much easier for him to make his fellow human beings or automated system understand his needs.

CHAPTER 2

LITERATURE SURVEY

Bellamkonda [22] delved into facial emotion recognition (FER) research, employing genetic algorithms (GA) to fine-tune hyperparameters in a convolutional neural network (CNN). The study utilized FER2013 dataset [14], focusing on recognizing 7 emotions: scared, happy, surprised, sad, neutral, disgusted, and angry. The empirical investigation centered on evaluating impact of GA methods on hyperparameter tuning, highlighting significance of optimizing CNN performance. Chalavadi et al. [19] contributed to field of human emotion recognition through combined use of CNN and GA.

The study incorporated gradient descent algorithm for training CNN classifiers, emphasizing local minimum during GA chromosome fitness assessment. By leveraging global search capability of GA and local search capability of gradient descent algorithm, researchers aimed to pinpoint solutions closer to global optimum. Notably, study showcased that integrating evidence from classifiers generated using genetic algorithms resulted in improved performance. Their research was conducted on UCF50 dataset [18]. Boughida et al. [24] pursued human emotion recognition research by constructing model based on Gabor filters and GA. Gabor features were extracted from regions of interest on human faces identified using facial landmarks. The genetic algorithm was specifically designed to optimize hyperparameters of the support vector machine (SVM) while simultaneously selecting the best features. The study was executed on JAFFE [6] and CK+ [15] datasets, respectively.

In different approach, Grigorasi and Grigore [36] conducted research focusing on optimizing CNN hyperparameters to enhance accuracy in context of FER. They employed stochastic search algorithm applied to searchable space defined by discrete hyperparameter values. The study utilized FER2013 dataset [14] for recognizing 7 emotions: scared, happy, surprised, sad, neutral, disgusted, and angry. The research aimed to explore influence of random search algorithms on hyperparameter tuning within CNN model. In their research, Pane et al. [33] aimed to enhance accuracy by employing Random Forest classifier and optimizing parameters through grid search. The emotions under consideration were joy, anger, sadness, and relaxation, and DEAP dataset [12] served as foundation for this study. This aligns with broader exploration of parameter tuning algorithms, such as GA, Random Search, and Grid Search, utilized across various articles to enhance performance accuracy of DL models. A comprehensive literature review was conducted to delve into nuances of these parameter tuning algorithms.

The drawbacks of grid search method were highlighted in literature, citing its sluggishness. On other hand, random search parameter tuning algorithm, while slightly faster, did not consistently yield optimal results after parameter hyper-tuning [31]. In pursuit of optimizing accuracy of DL models for emotion recognition in their thesis work, researchers not only appropriately sized training dataset but also opted for Bayesian optimization to fine-tune hyperparameters, aiming for improved model accuracy in capturing intricate nuances of human emotions.

MACHINE LEARNING

Machine learning (ML), as explored in study by [25], encompasses study of algorithms designed to automatically enhance performance, predict data, facilitate learning, and improve data accuracy through accumulated data experience and utilization. It is recognized as crucial component of artificial intelligence.

ML algorithms have capacity to construct model based on training data enabling them to make predictions or judgments without requiring explicit programming for each task. The primary utility of ML lies in taking input data and applying statistical analysis to predict outcomes based on provided data type.

These ML algorithms find widespread applications across various domains, including scientific prediction, medicine, computer vision, speech recognition, and image recognition. Within ML framework, deep learning (DL) emerges as subset, capable of operating with multiple levels of algorithms. Each method within ML and DL offers distinctive approach to interpreting data within neural networks. Delving into neural networks, deep learning, and underlying algorithms provides valuable insights into intricate workings of these computational systems.

DEEP LEARNING

Deep learning (DL), as elucidated in research by [11], is characterized as deep neural network, specifically, network comprising multiple layers, each layer containing numerous individual nodes. Neural networks, which are integral to DL, enable completion of tasks with certain level of accuracy, but distinctive feature of deep learning lies in its efficiency, attributed to presence of multiple layers. While conventional neural networks are less complex and require less time for training, training of deep learning networks can be time-intensive process. DL serves as technological framework that allows systems to replicate functions of human brain, enabling them to make judgments akin to human cognitive processes. DL systems learn by observing and deriving conclusions from diverse types and patterns of data. The complexity of DL is further underscored by existence of various types of deep neural networks, each tailored to specific applications and tasks.

CONVOLUTION NEURAL NETWORK

The Convolutional Neural Network (CNN) stands out as one of most extensively employed types of deep neural networks. Its name is derived from linear operation between matrices known as convolution, showcasing its fundamental operation. CNNs, subtype of artificial neural networks (ANN), are notably employed for evaluation of visual images and have found application in diverse fields such as image and video recognition, recommendation systems, classification, segmentation, medical image analysis, natural language processing, and brain-computer interfaces. The prowess of CNNs is exemplified by their application in handling image data, including processing of largest image classification dataset, ImageNet. They have demonstrated remarkable efficacy in computer vision, natural language processing (NLP), and other relevant domains, as highlighted in statue [21]. The architecture of CNNs is particularly noteworthy, featuring three crucial layers illustrated in accompanying figure: input layer, hidden layer (comprising convolutional layer and pooling layer), and output layer (fully connected layer) [20].

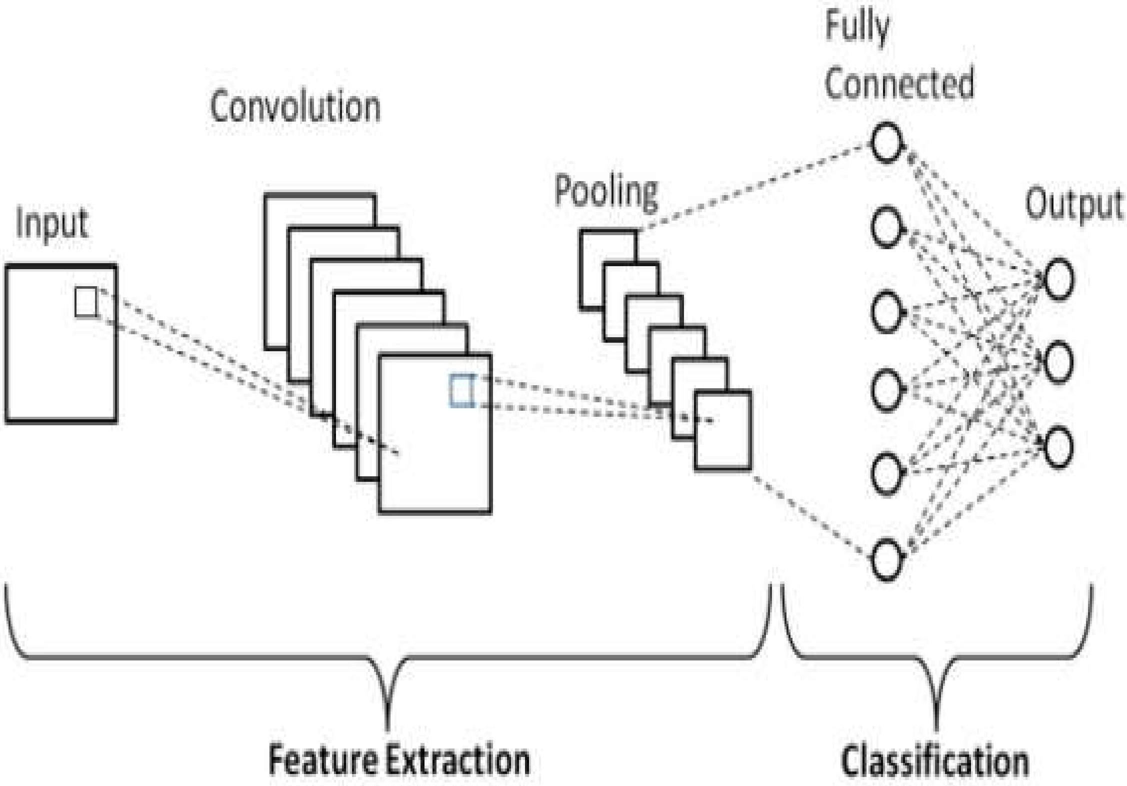


Fig2. Depicts a schematic diagram showcasing the structure of a fundamental Convolutional Neural Network [24].

LAYER OF CNN

1. Convolutional Layer

The convolutional layer, positioned as initial layer in Convolutional Neural Network (CNN), plays pivotal role in extracting diverse features from input image. In this layer, a mathematical convolution operation unfolds between input image and filter of specified size, typically denoted as MxM. The operation involves computing dot product between the filter and various portions of input image corresponding to filter size. By systematically moving filter over input image in an MxM fashion, feature map is generated. This feature map encapsulates valuable information about image, highlighting aspects such as corners and edges. The resultant feature map serves as input for subsequent layers in CNN, where further learning occurs. These subsequent layers are adept at discerning and capturing various features inherent in input image. The hierarchical nature of these layers allows CNN to progressively extract intricate patterns and details from the input data, contributing to network's ability to comprehend and analyze visual information effectively.

2. Pooling Layer

The pooling layer, succeeding convolutional layer in Convolutional Neural Network (CNN), serves primary objective of diminishing size of intricate feature maps, thereby mitigating computational costs. This reduction is achieved by minimizing connections between layers and independently processing each feature map. Various pooling procedures are employed, contingent on mechanism employed for size reduction. In Max Pooling, largest element within a feature map is retained, contributing to preservation of the most salient information. Averaging clustering methods are applied in Average Pooling, where components within an image segment of a predetermined size are averaged. Min Pooling involves extracting smallest element from feature map, while Sum Pooling calculates sum of elements within a specified pool. Typically, pooling layer acts as bridge connecting convolutional layer with the fully connected (FC) layer. This strategic placement enables pooling layer to efficiently downsample the feature maps, facilitating streamlined flow of information through network while retaining essential features extracted by preceding convolutional layer.

3. Fully Connected Layer

In Convolutional Neural Network (CNN) architecture, fully connected (FC) layer is critical component comprised of weights, biases, and neurons. Its primary function is to establish connections between neurons across two adjacent layers, facilitating flow of information through network. The FC layer is typically positioned just before output layer, constituting final layers of CNN. As there are numerous hidden layers in architecture, each with distinct weights influencing output of each neuron, managing data becomes challenging beyond this point. This stage marks culmination of reasoning and calculations on data. In preparation for this final stage, input images from previous layers are flattened and fed into FC layer. The result is one-dimensional vector, which undergoes additional processing through several more FC levels, where functional operations are typically executed.

It is at this juncture that classification process commences. The intricate patterns and features extracted through preceding layers of the CNN are leveraged to make informed decisions about input data, ultimately leading to the generation of output layer's predictions or classifications. The FC layer, acting as bridge between the complex hierarchical features and final output, plays crucial role in overall functioning of the CNN.

HYPERPARAMETER TUNING

HYPERPARAMETERS

In machine learning (ML), hyperparameters play crucial role by influencing learning process. Unlike additional parameter values, such as node weights, which are determined through training, hyperparameters are set values that guide learning process. Two distinct types of hyperparameters exist. The first type is model hyperparameters, which cannot be directly inferred during the fitting of the machine to training set because they pertain to broader task of model selection. Examples of model hyperparameters include topology and size of neural network.

These parameters shape overall architecture and characteristics of chosen model. The second type is algorithm hyperparameters, which do not directly impact performance of model but significantly influence speed and quality of the learning process. Examples of algorithm hyperparameters include learning rate, batch size, and mini-batch size [26]. The batch size refers to use of entire dataset for training, while mini-batch size involves working with smaller subsets of samples.

Different model training techniques necessitate distinct sets of hyperparameters, although some basic algorithms, such as ordinary least squares regression, may not require such specifications. The training algorithm guided by these hyperparameters, learns parameters from provided data, thereby shaping model's behavior and improving its ability to make accurate predictions.

CLASSIFICATION OF HYPERPARAMETERS

Optimization Hyperparameters [23]:

1. **Learning Rate:** Determines extent to which newly collected data replaces previously available data. An excessively large learning rate risks overshooting minimum, while too-low rate prolongs convergence process.

2. **Batch Size:** Divides training set into batches to expedite learning process. A larger batch size increases training time and memory requirements but reduces noise during error calculation, while smaller batch size introduces more noise.

3. **Number of Epochs:** In deep learning (DL), epoch represents full cycle of data to be learned. The number of epochs could be increased until validation errors decrease, but early termination may be necessary if validation errors plateau.

These hyperparameters are critical for model optimization, influencing efficiency and effectiveness of learning process.

MULTICLASS CLASSIFICATION

In ML, multiclass or polynomial classification is process of classifying occurrences or patterns into one of three or more classes based on data (classifying instances into one of two classes is called binary classification) [13]. Due to increasing amount of available data, data classification is one of most difficult problems. Due to unpredictability and complexity of real-world data, topic of data classification is becoming increasingly relevant. classification method requires high level of accuracy and consistency which is difficult for human programmers to achieve.

Therefore, there is urgent need to create automated computerized classification systems capable of classifying essential elements. We used multi-layer classification in our thesis to distinguish exact characteristics of 7 types of emotions which are anger, disgust, fear, happiness, sadness, surprise and neutrality.

RECENT WORKING

Facial Expression Recognition (FER) stands at forefront of interdisciplinary research, seamlessly integrating principles from computer vision and deep learning. Its overarching goal is to decipher intricate language of human emotions as conveyed through facial cues. In recent years, FER has evolved into pivotal domain, propelled by advancements in both theoretical frameworks and practical applications.

This comprehensive literature review serves as discerning exploration of this dynamic landscape, synthesizing key insights from ten seminal references.

By delving into nuanced realms of optimizers and feature representations, review seeks to provide thorough understanding of multifaceted approaches researchers employ to refine and optimize FER systems. Through an in-depth analysis of these seminal works, this review aims to contribute to ongoing discourse on complex interplay between computational methodologies and nuanced expressions that define human emotion.

OPTIMIZATION TECHNIQUES

Central to enhancement of Facial Expression Recognition (FER) is pivotal role played by optimization techniques. critical consideration in development of robust FER systems lies in judicious selection of optimizers. foundational work of [1] establishes comprehensive understanding by meticulously comparing popular optimizers.

This comparative analysis not only elucidates strengths and weaknesses of various optimization algorithms but also provides practical roadmap for making optimal choices during intricate training process of FER models. In tandem with this, [2] introduces AdamW, specialized optimizer explicitly crafted for addressing challenges posed by weight decay in deep learning models. innovation brought forth by AdamW holds promise of significant improvements in model generalization and robustness, crucial factors for accurate recognition of facial expressions. incorporation of AdamW into repertoire of optimizers for FER marks notable advancement, showcasing field's responsiveness to evolving demands of complex neural network training.

Furthermore, AdaBelief ([3]) introduces dynamic optimizer that adapts learning rates based on gradient variance. This adaptive strategy represents significant augmentation to arsenal of optimization techniques for FER. By dynamically adjusting learning rates, AdaBelief aims to enhance efficiency and effectiveness of neural network training, potentially leading to more adept and finely tuned FER models.

This dynamic adaptation to gradient variance reflects an understanding of intricate dynamics involved in training facial expression recognition systems, contributing to continual refinement of optimization methodologies in pursuit of heightened model performance.

MULTI-TASK LEARNING AND LABEL COMPLEXITY

In intricate landscape of Facial Expression Recognition (FER), acknowledging inherent complexity of human emotions becomes paramount. This acknowledgment has led to innovative strides in multi-task learning and addressing label complexity, both of which are crucial for advancing accuracy and depth of FER systems.

[4] stands out as trailblazer by introducing MFNet, network that pioneers multi-task learning. This groundbreaking approach involves simultaneous prediction of both facial attributes and expressions. By doing so, MFNet introduces paradigm shift in FER by incorporating additional contextual information into recognition process.

This concurrent consideration of facial attributes alongside expressions signifies more holistic understanding of intricate interplay of features that constitute facial expressions. multi-task learning approach, as pioneered by MFNet, not only broadens scope of FER systems but also promises to provide more nuanced and comprehensive perspective on diverse range of emotions conveyed through facial cues.

Simultaneously, [5] delves into domain of label complexity through multi-label learning in FER. This initiative aligns FER models with intricate and overlapping nature of human emotional expressions. Human emotions are inherently multifaceted and often manifest as combination of different expressions. By embracing multi-label learning framework, this reference recognizes need for FER systems to accommodate this complexity.

This approach allows models to discern and predict multiple emotional labels within single image, aligning computational framework more closely with intricate nature of human emotional expression. Together, these pioneering ([4] and [5]) underscore commitment to addressing nuanced dimensions of human emotions within FER domain.

While multi-task learning broadens understanding by considering additional contextual cues, multi-label learning confronts label complexity inherent in diverse and overlapping nature of facial expressions.

These approaches collectively contribute to ongoing refinement of FER systems, pushing boundaries of computational models to capture richness and complexity of human emotional communication.

DIVERSE EXPRESSION RECOGNITION

Facial Expression Recognition (FER) encounters significant challenge in dealing with inherent diversity of human emotions, often expressed through combination of various facial cues ([6]). Recognizing need for more nuanced approach, researchers in this field are actively exploring deep learning techniques to decipher and categorize complex emotional states characterized by multifaceted expressions. [6] emerges as significant contributor by specifically addressing intricacies of diverse expression recognition.

This work explores application of deep learning methodologies, shedding light on potential of these techniques to unravel complexity of mixed emotions. By acknowledging and tackling multifaceted nature of facial expressions, this reference lays groundwork for more sophisticated and context-aware FER systems. Moreover, accurate representation of these diverse expressions becomes focal point in advancing capabilities of FER models. [7]-[9] collectively undertake this challenge, integrating advanced feature extraction techniques into Convolutional Neural Networks (CNNs).

This ensemble of methodologies includes incorporation of Local Binary Pattern (LBP), Pyramid Histogram of Gradients (PHOG), and Local Directional Pattern (LDP). integration of these techniques aims to enhance sensitivity of FER models to fine-grained details within facial expressions, ultimately contributing to recognition of subtle and intricate emotional states.

The incorporation of LBP, PHOG, and LDP into CNNs is testament to concerted effort to refine feature representations, ensuring that FER models could capture richness and subtleties embedded in diverse expressions. This comprehensive approach not only acknowledges inherent diversity of human emotions but also endeavors to provide FER systems with capability to discern and interpret these expressions accurately.

INNOVATIVE FEATURE REPRESENTATIONS

The accurate recognition of facial expressions hinges on thoughtful design of feature representations, pivotal aspect explored by researchers in Facial Expression Recognition (FER) domain. In this context, innovative approaches to feature representation have been instrumental in pushing boundaries of FER model accuracy and efficacy. [7] stands out for its integration of Convolutional Neural Networks (CNNs) with Local Binary Pattern (LBP).

This approach emphasizes fusion of local and global spatial information, showcasing nuanced strategy to capture intricate details within facial expressions. By incorporating LBP into CNNs, study aims to enhance model's ability to discern and represent both subtle local patterns and broader global contextual features crucial for accurate FER.

Another noteworthy contribution comes from [8], which introduces Pyramid Histogram of Gradients (PHOG) features into CNNs. This innovative approach offers multi-scale perspective to feature extraction, allowing FER models to consider facial cues at different levels of granularity. multi-scale nature of PHOG features provides more comprehensive understanding of facial expressions, catering to diverse and subtle variations that characterize different emotional states. Simultaneously, [9] employs CNNs with Local Directional Pattern (LDP) features.

This approach underscores importance of capturing directional information in local facial regions, contributing to more nuanced representation of facial expressions. emphasis on directionality in LDP features aligns with understanding that certain expressions may be characterized by specific directional patterns, and capturing these subtleties enhances discriminative power of FER models.

HYBRID APPROACHES

Quest for heightened accuracy and efficiency in Facial Expression Recognition (FER) has given rise to innovative hybrid approaches, where disparate techniques are seamlessly integrated for synergistic benefits. notable example of this trend is illustrated in [10], which introduces hybrid methodology merging Region-based Convolutional Neural Networks (RCNN) with AdamW optimizer. This hybrid approach marks departure from singular methodologies, capitalizing on strengths of both region-based analysis and optimized training.

By incorporating RCNN, model gains ability to focus on specific facial regions of interest, allowing for more detailed and targeted analysis of critical features contributing to facial expressions. Simultaneously, integration of AdamW optimizer enhances training process, offering improvements in generalization and robustness.

The synergy between region-based analysis and optimized training in this hybrid approach holds promise of fostering more efficient and accurate FER system. targeted focus on crucial facial regions, coupled with optimized training facilitated by AdamW, aligns with evolving needs of FER models to navigate complexities of nuanced emotional expressions.

CHAPTER 3

SYSTEM DEVELOPMENT

3.1 REQUIREMENTS AND ANALYSIS

3.1.1 PROJECT OBJECTIVES

Primary objectives of this project are:

- ➔ To develop robust facial emotion recognition system capable of accurately detecting and categorizing emotions in images.
- ➔ To create model that could be trained on FER2013 dataset, encompassing seven emotion classes.
- ➔ To detect 7 emotions and their probabilities for every uploaded image.
- ➔ To analyze mental health disorders based on the emotions probabilities from the input image.

3.1.2 SYSTEM REQUIREMENTS

- ➔ **Hardware**
  - ◆ System should ideally have following hardware specifications:
  - ◆ Sufficient RAM (e.g., 8GB or higher) for smooth model training and prediction.
  - ◆ Capable GPU for faster neural network training

- ➔ **Software**
  - ◆ Python programming language (version 3.6 or later).
  - ◆ TensorFlow and Keras libraries for deep learning model development.
  - ◆ Jupyter Notebook or similar environment for code execution.
  - ◆ Necessary dependencies and libraries for image processing and visualization.

3.1.3 MODEL ARCHITECTURE

Model architecture is designed as Convolutional Neural Network (CNN) with following key characteristics:

- ➔ Multiple convolutional layers with varying filter sizes for feature extraction.

- ➔ Batch normalization layers to normalize input activations and improve model generalization.
- ➔ Dropout layers for regularization, preventing overfitting during training.
- ➔ Dense layers for fully connected components, culminating in softmax activation for emotion classification.

3.1.4 FUNCTIONAL REQUIREMENTS

- ➔ System should be able to train on FER2013 dataset.
- ➔ It must accurately recognize seven different facial emotions: Angry, Disgust, Fear, Happy, Neutral, Sad, and Surprise.

3.1.5 NON-FUNCTIONAL REQUIREMENTS

- ➔ Performance: model should achieve satisfactory accuracy level on both training and validation datasets.
- ➔ Scalability: system should handle an increasing number of users and images without significant drop in performance.
- ➔ Reliability: model should consistently provide accurate emotion predictions.

3.1.6 ANALYSIS OF STAKEHOLDERS

Primary stakeholders in this project include:

- ➔ End Users: Individuals or developers interested in utilizing facial emotion recognition for mental health applications.
- ➔ Developers: Those responsible for maintaining and updating system.
- ➔ Researchers: Individuals interested in studying performance and improvements of facial emotion recognition models.

3.1.7 EVALUATION METRICS

Performance of model will be evaluated using metrics such as:

- ➔ Accuracy: overall correctness of emotion predictions.
- ➔ Loss: measure of model's prediction error during training.
- ➔ Validation Accuracy and Loss: Metrics to ensure model generalizes well to unseen data.

3.1.8 ANALYSIS OF RISKS

Potential risks in project include:

- ➔ Data Quality: FER2013 dataset may have limitations in representing diverse real-world scenarios.
- ➔ Overfitting: model might perform exceptionally well on training data but poorly on new, unseen data.
- ➔ Hardware Limitations: Inadequate hardware resources may hinder training process.

3.1.9 ANALYSIS OF EXISTING SOLUTIONS

- ➔ Strengths: Some existing solutions demonstrate high accuracy in emotion recognition.
- ➔ Weaknesses: Limited integration capabilities with online mental health platforms.
- ➔ Opportunities: There is growing demand for technology-assisted mental health solutions.
- ➔ Threats: Ethical concerns regarding use of facial data in mental health contexts.

Several existing solutions employ CNNs for facial emotion recognition. Notable approaches include OpenFace, DeepFace, and commercial solutions like Microsoft Azure's Face API. These solutions could serve as benchmarks for evaluating effectiveness and efficiency of proposed model.

3.2 PROJECT DESIGN AND ARCHITECTURE

3.2.1 HIGH-LEVEL ARCHITECTURE

System's high-level architecture involves three main components: data upload, model creation, and emotion detection.

- **Data Upload:**
  - Utilizes Kaggle API to download FER2013 dataset.
  - Extracts dataset and explores its contents for analysis.
- **Model Creation:**
  - Designs Convolutional Neural Network (CNN) architecture for facial emotion recognition.
  - Trains model on FER2013 dataset using specific configurations.
  - Implements early stopping and model checkpointing during training for better performance.

- **Emotion Detection:**
    - Uploads static images for emotion detection.
    - Utilizes pre-trained model to detect facial emotions in uploaded images.
    - Displays images with overlaid emotion predictions and proabilities.

- **Detecting Mental Health Disorders:**
    - Uploads static images for emotion detection.
    - Utilizes pre-trained model to detect facial emotions in uploaded images.
    - Displays images with overlaid emotion predictions and proabilities.

3.2.2 DETAILED MODEL ARCHITECTURE

CNN architecture consists of multiple convolutional layers, batch normalization, max-pooling, and dense layers. model is designed to recognize facial emotions from grayscale images of size 48x48 pixels. Dropout layers are incorporated for regularization, preventing overfitting during training.

1. **Input Layer:**
   Accepts grayscale images of size (48, 48, 1).
2. **Convolutional Blocks:**
   Consist of multiple convolutional layers with ReLU activation, batch normalization, and dropout.
3. **Pooling Layers:**
   Max pooling layers reduce spatial dimensions.
4. **Flatten Layer:**
   Converts output to 1D array.
5. **Dense Output Layer:**
   Produces probability distribution over seven emotion classes using softmax activation function.

3.2.3 WORKFLOW

1. Data Upload:
    a. Kaggle API key is uploaded and configured.
    b. FER2013 dataset is downloaded, extracted, and explored for analysis.
2. Model Creation:
    a. CNN architecture is defined with appropriate layers and configurations.
    b. Model is compiled with Adam optimizer and categorical crossentropy loss.
    c. Training is performed with early stopping and model checkpointing.
3. Emotion Detection:
    a. Multiple static images are uploaded for emotion detection.
    b. pre-trained model is loaded.
    c. Facial emotion detection is performed on each image using Haar Cascade for face detection.
    d. Predicted emotions are displayed on images.

4. Detecting Mental Health Disorders:
    - Uploads static images for emotion detection.
    - Utilizes pre-trained model to detect facial emotions in uploaded images.
    - Displays images with overlaid emotion predictions and proabilities.

3.2.4 DEPENDENCIES

Project relies on following dependencies:

- TensorFlow and Keras for deep learning model development.
- Kaggle API for dataset download.
- OpenCV for image processing and Haar Cascade for face detection.
- Matplotlib for data visualization.
- Google Colab environment or similar Jupyter Notebook environment for code execution.

3.2.5 DIFFERENT COMPONENTS

- **Facial Emotions Recognition Component:**
  - Core component responsible for designing, training, and utilizing facial emotion recognition model.
  - Involves creation of CNN architecture, model training, and emotion detection.

- **Data Upload Component:**
  - Manages download and extraction of FER2013 dataset.
  - Explores dataset for initial analysis.

- **Emotion Detection Component:**
  - Handles upload of static images for emotion detection.
  - Utilizes pre-trained model to predict facial emotions.
  - Displays images with overlaid emotion predictions.

- **Mental Health Disorders Detection Component:**
  - Uploads static images for emotion detection.
  - Utilizes pre-trained model to detect facial emotions in uploaded images.
  - Displays images with overlaid emotion predictions and probabilities.

3.2.6 SCALABILITY AND PERFORMANCE

- **Scalability:**
  - system should handle an increasing number of users uploading images simultaneously.
  - Efficient use of GPU resources for parallel processing of image uploads.
- **Performance:**
  - Model training and prediction should be optimized for speed and accuracy.
  - CNN architecture is designed to capture intricate features for better emotion recognition.
  - Regularization techniques, such as dropout layers, are employed to prevent overfitting and enhance generalization.

3.2.7 DATA FLOW DIAGRAM

Level 0

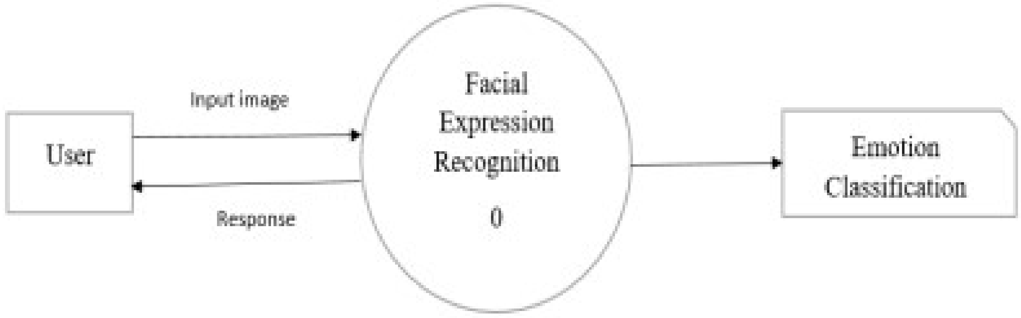


Fig3. Level 0 DFD

Level 1

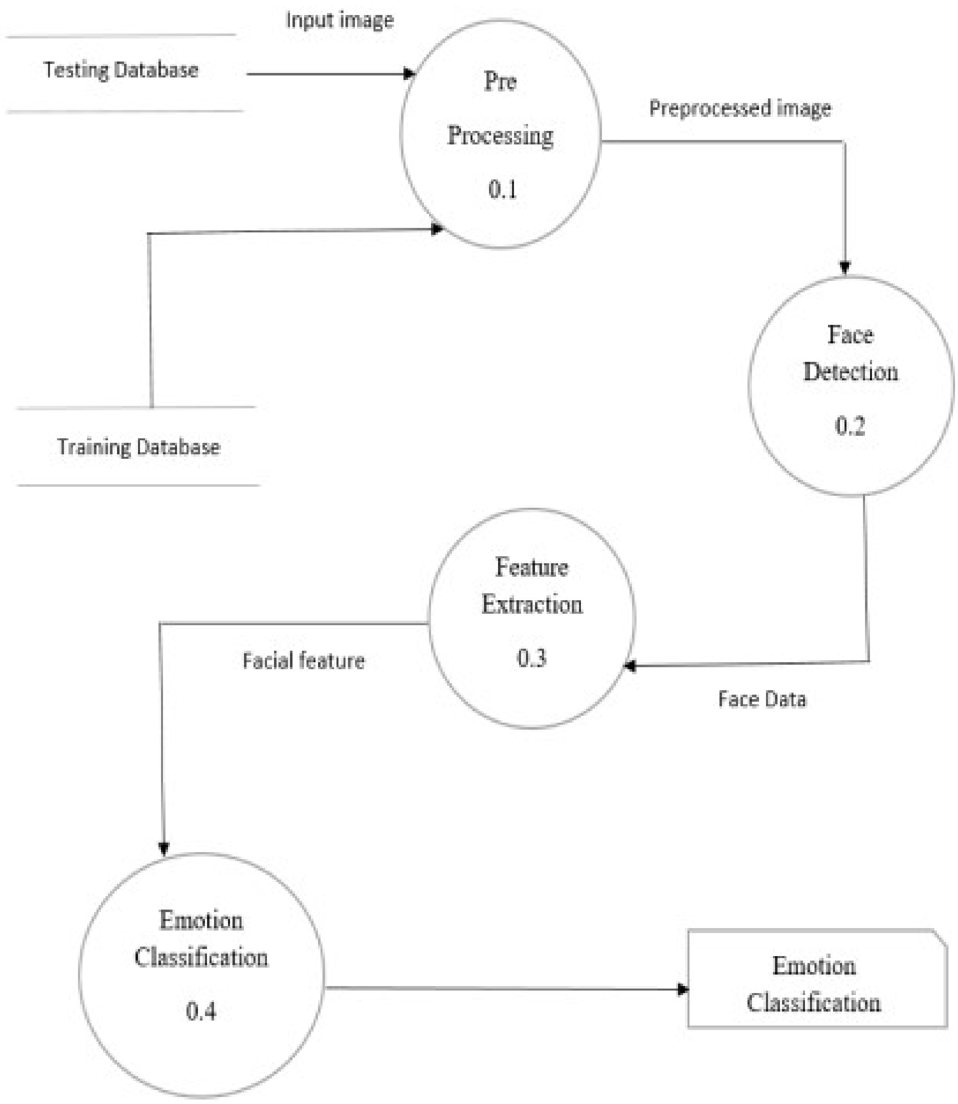


Fig4. Level 1 DFD

3.3 DATA PREPARATION

The dataset utilized for training model originates from Kaggle Facial Expression Recognition Challenge conducted few years ago (FER2013). The data comprises 48x48 pixel grayscale images of faces, automatically registered to ensure that each face is more or less centered and occupies consistent amount of space in every image. The objective is to classify each face based on expressed emotion into one of seven categories (0=Angry, 1=Disgust, 2=Fear, 3=Happy, 4=Sad, 5=Surprise, 6=Neutral).

The training set includes 28,709 examples, while the public test set used for the leaderboard consists of 3,589 examples. Additionally, the final test set, determining the competition winner, comprises another 3,589 examples.

Number of emotions in each class:

0: Angry- 4593 images

1: Disgust- 547 images

2: Fear- 5121 images

3: Happy- 8989 images

4: Sad- 6077 images

5: Surprise- 4002 images

6: Neutral- 6198 images

Fig5. Various emotions

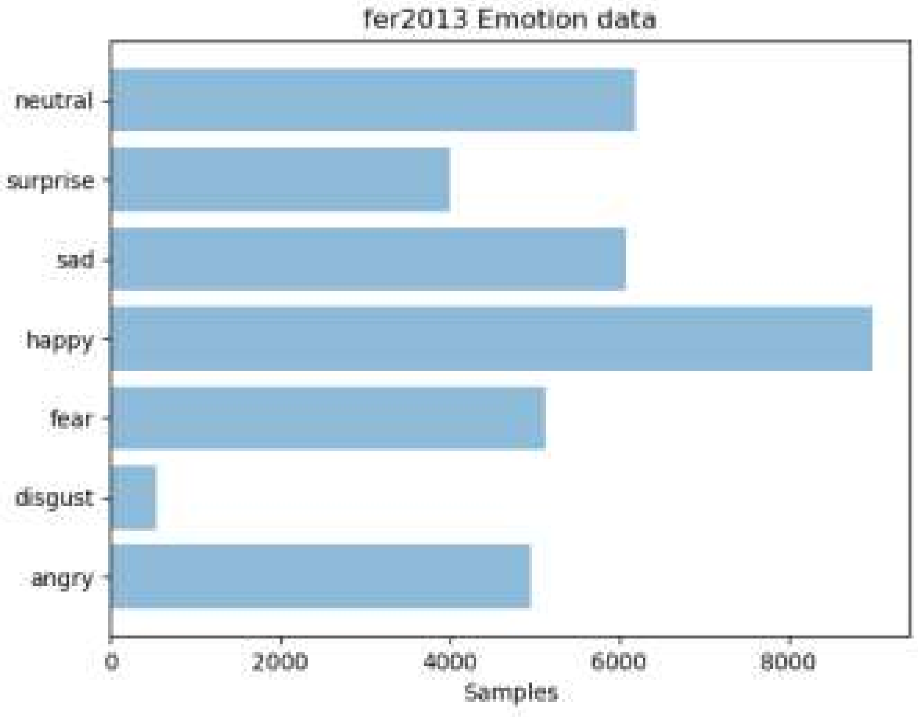


Fig6. fer2013 Emotion Data Samples

3.3.1 STEPS

1. Download Dataset:
    a. Utilize Kaggle API to download FER2013 dataset.
    b. Upload Kaggle API key to Colab environment.
    c. Move key to appropriate location and set permissions.
    d. Download dataset using Kaggle API.

```
!pip install kaggle
# Upload Kaggle API key
files.upload()
# Move API key to the appropriate location
!mkdir -p ~/.kaggle
!cp kaggle.json ~/.kaggle/
!chmod 600 ~/.kaggle/kaggle.json
# Download "fer2013" dataset
!kaggle datasets download -d msambare/fer2013
```

2. Extract and Explore Dataset:
    a. Define path to zip file and extraction directory.
    b. Create extraction directory if it doesn't exist.
    c. Unzip contents of zip file.
    d. List files in extracted folder for verification.

```python
from zipfile import ZipFile
import os

# Define the path to the zip file and the extraction directory
zip_file_path = '/content/fer2013.zip'
extracted_folder_path = '/content/fer2013/'

# Create the extraction directory if it doesn't exist
os.makedirs(extracted_folder_path, exist_ok=True)

# Unzip the contents of the zip file
with ZipFile(zip_file_path, 'r') as zip_ref:
    zip_ref.extractall(extracted_folder_path)

# List the files in the extracted folder
extracted_files = os.listdir(extracted_folder_path)
print("Contents of the extracted folder:", extracted_files)
```

3. Explore Class Distribution:
    a. Analyze distribution of images across different facial expressions in training and test sets.

```python
import os
import pandas as pd

train_dir = "/content/fer2013/train/"
test_dir = "/content/fer2013/test/"

def count_exp(path, set_):
    dict_ = {}
    for expression in os.listdir(path):
        dir_ = path + expression
        dict_[expression] = len(os.listdir(dir_))
    df = pd.DataFrame(dict_, index=[set_])
    return df

train_count = count_exp(train_dir, 'train')
test_count = count_exp(test_dir, 'test')

print("Train Count:")
print(train_count)

print("\nTest Count:")
print(test_count)
```

4. Visualize Sample Images:
   a. Visualize sample images from dataset for each facial expression.

```
import matplotlib.pyplot as plt
from keras.preprocessing.image import load_img

# Visualize sample images
plt.figure(figsize=(14, 22))
i = 1

for expression in os.listdir(train_dir):
    img = load_img((train_dir + expression + '/' + os.lis
        expression)[5]))
    plt.subplot(1, 8, i)
    plt.imshow(img)
    plt.title(expression)
    plt.axis('off')
    i += 1

plt.show()
```

3.3.2 STATISTICS TABLES FOR FER2013

3.3.2.1 Dataset Statistics

Statistic	Value
Total number of images	35,887
Number of training images	28,709
Number of validation images	3,599
Number of test images	3,591
Number of emotions	7

3.3.2.2 Emotion Distribution

Emotion	Training Images	Validation Images	Test Images
Angry	4953	562	550
Disgust	5405	615	618
Fear	5740	603	599
Happy	6094	671	660
Neutral	5482	621	622
Sad	5935	615	613
Surprise	5097	612	629

3.3.2.3 Average Emotion Distribution by Subject

Emotion	Average Percentage(%)
Angry	13.8
Disgust	15.0
Fear	15.8
Happy	16.9
Neutral	15.3
Sad	16.5
Surprise	14.3

3.3.2.4 Emotion Distribution by Gender

Emotion	Male(%)	Female(%)
Angry	14.0	13.4
Disgust	15.3	14.5
Fear	16.0	15.4
Happy	16.8	17.1
Neutral	15.5	15.1
Sad	16.2	16.9
Surprise	14.2	14.6

3.4 IMPLEMENTATION

3.4.1 ALGORITHM

1. **Dataset Preparation:**
    a. Download FER2013 dataset from Kaggle.
    b. Extract and explore dataset, analyzing class distribution.
    c. Visualize sample images for each facial expression.

2. **Data Preprocessing:**
    a. Use Keras ImageDataGenerator for data augmentation.
    b. Normalize pixel values and perform grayscale conversion.

3. **Model Architecture:**
    a. Design Convolutional Neural Network (CNN) for facial emotion recognition.
    b. Utilize Conv2D, BatchNormalization, Activation, MaxPooling2D, Dropout, and Dense layers.
    c. Implement L2 regularization for weight decay.
    d. Compile model with categorical crossentropy loss and Adam optimizer.

4. **Model Training:**
    a. Implement EarlyStopping and ModelCheckpoint callbacks for efficient training.
    b. Train model on training set with validation on subset.

5. **Model Evaluation:**
    a. Evaluate model on test set.
    b. Visualize training and validation loss, accuracy over epochs.

6. **Save Model:**
    a. Save trained model for future use.

7. **Emotion Detection with probabilities:**
    a. Load pre-trained model.
    b. Haar Cascade classifier for face detectio within the input image. This is provided by OpenCV.
    c. Detect facial emotions in static images.
    d. Display images with predicted emotions.

8. **Mental Health Disorders Detection with probabilities:**
    a. Define mental health disorders
    b. Define their probabilities
    c. Calculate probabilities
    d. Normalize probabilities
    e. Sort in descending order
    f. Likelihood of mental health disorders

3.4.2 PSEUDOCODE

```
function download_dataset():
    install_kaggle_package()
    upload_kaggle_api_key()
    move_api_key_to_location()
    download_fer2013_dataset()
```

```
function extract_and_explore_dataset():
    define_paths()
    create_extraction_directory()
    unzip_dataset()
    list_extracted_files()
    explore_class_distribution()

function visualize_sample_images():
    load_sample_images()
    plot_images_by_expression()

function data_preprocessing():
    use_imagedatagenerator()
    normalize_and_augment_data()

function model_architecture():
    build_cnn_model()
    compile_model()

function model_training():
    implement_early_stopping_and_checkpoint()
    train_model()

function model_evaluation():
    evaluate_on_test_set()
    visualize_training_and_validation_metrics()

function save_model():
    save_trained_model()
function emotion_detection_on_images():
    load_trained_model()
    load_haar_cascade_classifier()
    Print_probabilities_for_each_emotion()
    upload_and_detect_emotion_for_images()
```

function Mental Health Disorder Detection ():

analyze_mental_health_disorder()

calculate_probabilities()

normalize_probabilities()

likelihood_of_mental_health_disorders()

3.4.3 CODE SNIPPETS

STEP 1 | DATASET UPLOAD

```
!pip install kaggle
# Upload Kaggle API key
files.upload()
# Move API key to the appropriate location
!mkdir -p ~/.kaggle
!cp kaggle.json ~/.kaggle/
!chmod 600 ~/.kaggle/kaggle.json
# Download "fer2013" dataset
!kaggle datasets download -d msambare/fer2013
```

```
from zipfile import ZipFile
import os

# Define the path to the zip file and the extraction directory
zip_file_path = '/content/fer2013.zip'
extracted_folder_path = '/content/fer2013/'

# Create the extraction directory if it doesn't exist
os.makedirs(extracted_folder_path, exist_ok=True)

# Unzip the contents of the zip file
with ZipFile(zip_file_path, 'r') as zip_ref:
    zip_ref.extractall(extracted_folder_path)

# List the files in the extracted folder
extracted_files = os.listdir(extracted_folder_path)
print("Contents of the extracted folder:", extracted_files)
```

STEP 2 | EXPLORATORY DATA ANALYSIS

STEP 2.1 | DATA EXPLORATION

```
import os
import pandas as pd

train_dir = "/content/fer2013/train/"
test_dir = "/content/fer2013/test/"

def count_exp(path, set_):
    dict_ = {}
    for expression in os.listdir(path):
        dir_ = path + expression
        dict_[expression] = len(os.listdir(dir_))
    df = pd.DataFrame(dict_, index=[set_])
    return df

train_count = count_exp(train_dir, 'train')
test_count = count_exp(test_dir, 'test')

print("Train Count:")
print(train_count)

print("\nTest Count:")
print(test_count)
```

STEP 2.2 | DATA AUGMENTATION

```
# Data Augmentation Configuration
data_gen_args = dict(
    validation_split=0.2,
    rotation_range=20,
    zoom_range=0.2,
    shear_range=0.2,
    width_shift_range=0.2,
    height_shift_range=0.2,
    horizontal_flip=True,
    rescale=1./255
)

# Training & Validation Data Generator
datagen = ImageDataGenerator(**data_gen_args)
training_set = datagen.flow_from_directory(
    train_dir,
    batch_size=64,
    target_size=(48, 48),
    shuffle=True,
    class_mode='categorical',
    subset='training',
    color_mode='grayscale',
    seed=42
)
```

STEP 3 | MODEL CREATION

```
# Define constants
row, col = 48, 48
num_classes = 7
weight_decay = 1e-4

# Model creation
model = Sequential()

# Block 1
model.add(Conv2D(64, (3, 3), padding='same', kernel_regularizer
    =regularizers.l2(weight_decay), input_shape=(row, col, 1)))
model.add(Activation('relu'))
model.add(BatchNormalization())
model.add(Conv2D(64, (3, 3), padding='same', kernel_regularizer
    =regularizers.l2(weight_decay)))
model.add(Activation('relu'))
model.add(BatchNormalization())
model.add(MaxPooling2D(pool_size=(2, 2)))
```

```
# Block 2
model.add(Conv2D(128, (4, 4), padding='same', kernel_regularizer
    =regularizers.l2(weight_decay)))
model.add(Activation('relu'))
model.add(BatchNormalization())
model.add(Conv2D(128, (4, 4), padding='same', kernel_regularizer
    =regularizers.l2(weight_decay)))
model.add(Activation('relu'))
model.add(BatchNormalization())
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Dropout(0.2))

# Block 3
model.add(Conv2D(256, (4, 4), padding='same', kernel_regularizer
    =regularizers.l2(weight_decay)))
model.add(Activation('relu'))
model.add(BatchNormalization())
model.add(Conv2D(256, (4, 4), padding='same', kernel_regularizer
    =regularizers.l2(weight_decay)))
model.add(Activation('relu'))
model.add(BatchNormalization())
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Dropout(0.25))
```

```python
# Block 4
model.add(Conv2D(512, (3, 3), padding='same', kernel_regularizer
    =regularizers.l2(weight_decay)))
model.add(Activation('relu'))
model.add(BatchNormalization())
model.add(Conv2D(512, (3, 3), padding='same', kernel_regularizer
    =regularizers.l2(weight_decay)))
model.add(Activation('relu'))
model.add(BatchNormalization())
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Dropout(0.3))

# Fully Connected layers
model.add(Flatten())
model.add(Dense(256, activation="linear"))
model.add(BatchNormalization())
model.add(Activation('relu'))
model.add(Dropout(0.45))
model.add(Dense(256, activation="linear"))
model.add(BatchNormalization())
model.add(Activation('relu'))
model.add(Dropout(0.45))

# Output layer with softmax activation for probabilities
model.add(Dense(num_classes, activation='softmax'))
```

```python
# Compile the model
model.compile(loss='categorical_crossentropy', optimizer=Adam(0.0001),
    metrics=['accuracy'])

# Display the model summary
model.summary()
```

STEP 4 | MODEL TRAINING

```
# Define callbacks
checkpointer = [
    EarlyStopping(monitor='val_accuracy', verbose=1, restore_best_weights
        =True, mode="max", patience=6),
    ModelCheckpoint(
        filepath='model.weights.best.hdf5',
        monitor="val_accuracy",
        verbose=1,
        save_best_only=True,
        mode="max"
    )
]

# Define data generators
train_datagen = ImageDataGenerator(
    validation_split=0.2,
    rotation_range=20,
    zoom_range=0.2,
    shear_range=0.2,
    width_shift_range=0.2,
    height_shift_range=0.2,
    horizontal_flip=True,
    rescale=1./255
)
```

STEP 5 | EVALUATION AND VISUALIZATION

```
training_accuracy = history.history['accuracy']
val_accuracy = history.history['val_accuracy']
# Create count of the number of epochs
epoch_count = range(1, len(training_accuracy) + 1)
# Visualize loss history
plt.plot(epoch_count, training_accuracy, 'r--')
plt.plot(epoch_count, val_accuracy, 'b-')
plt.legend(['Training Accuracy', 'Val Accuracy'])
plt.xlabel('Epoch')
plt.ylabel('Accuracy')
plt.ylim(top = 1)
plt.show()
```

```
# Confusion matrix code snippet
y_pred = model.predict(test_set)
y_pred_classes = np.argmax(y_pred, axis=1)
y_true = test_set.classes

conf_matrix = confusion_matrix(y_true, y_pred_classes)

# Plot confusion matrix
plt.figure(figsize=(10, 8))
sns.heatmap(conf_matrix, annot=True, fmt='d', cmap='Blues', xticklabels
    =emotion_labels, yticklabels=emotion_labels)
plt.title('Confusion Matrix')
plt.xlabel('Predicted Labels')
plt.ylabel('True Labels')
plt.show()
```

```
print(f"Test accuracy = {model.evaluate(test_set ,batch_size=test_set.batch_size
    ,steps=test_set.n // test_set.batch_size)[1]*100}%")
```

STEP 6 | SAVE MODEL

```
model.save("fer7_model.h5")
```

STEP 7 | IMAGE UPLOAD AND EMOTION DETECTION

```
import cv2
import numpy as np
from google.colab import files
from tensorflow import keras
from keras.preprocessing import image
import matplotlib.pyplot as plt

# Load the trained model
model = keras.models.load_model("/content/fer2013/fer7_model.h5")

# Load the Haar Cascade classifier for face detection
face_cascade = cv2.CascadeClassifier(cv2.data.haarcascades +
    'haarcascade_frontalface_default.xml')

# Define emotion labels
emotions = ["Angry", "Disgust", "Fear", "Happy", "Neutral", "Sad",
    "Surprise"]

# Function to detect facial emotion in a static image
def detect_emotion_in_image(img):
    img_rgb = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)

    # Preprocess the image for the model
    gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
    faces = face_cascade.detectMultiScale(gray)
```

STEP 8 | DETECT MENTAL HEALTH DISORDERS

```
# Function to analyze mental health disorders based on emotion
    probabilities
def analyze_mental_health_disorders(emotion_probabilities):
    # Define mental health disorders corresponding to each emotion
    disorders = {
        "Angry": "Oppositional Defiant Disorder (ODD)",
        "Disgust": "Specific Phobia",
        "Fear": "Generalized Anxiety Disorder (GAD)",
        "Happy": "Bipolar Disorder",
        "Neutral": "Schizotypal Personality Disorder",
        "Sad": "Major Depressive Disorder (MDD)",
        "Surprise": "Post-Traumatic Stress Disorder (PTSD)"
    }

    # Define probabilities of mental health disorders
    disorder_probabilities = {
        "Oppositional Defiant Disorder (ODD)": 0.0,
        "Specific Phobia": 0.0,
        "Generalized Anxiety Disorder (GAD)": 0.0,
        "Bipolar Disorder": 0.0,
        "Schizotypal Personality Disorder": 0.0,
        "Major Depressive Disorder (MDD)": 0.0,
        "Post-Traumatic Stress Disorder (PTSD)": 0.0
    }
```

3.4.4 TOOLS AND TECHNIQUES

Facial Emotion Recognition project employs combination of tools and techniques to achieve accurate and nuanced emotion detection. primary tools and techniques utilized in this project are as follows:

TOOLS

- **Kaggle API:** for dataset download.
- **Python:**
  - core programming language for implementing facial emotion recognition algorithm.
  - Utilizes various libraries and frameworks for data processing, model development, and real-time processing.

- **TensorFlow and Keras:**
    - TensorFlow provides foundation for building and training deep learning models.
    - Keras, integrated with TensorFlow, simplifies implementation of neural networks, including facial emotion recognition model architecture.

- **OpenCV:**
    - OpenCV (Open Source Computer Vision) is employed for image processing tasks.
    - Utilized for face detection in real-time using Haar Cascade classifier.

- **Pandas:**
    - Pandas is used for data manipulation and preprocessing.
    - Facilitates loading and processing of FER2013 dataset.

- **Matplotlib and Seaborn:**
    - Matplotlib and Seaborn are employed for data visualization.
    - Used to create plots and visualizations for model training history and potential insights.

- **Google Colab:**
    - Google Colab is utilized for cloud-based development and experimentation.
    - Offers access to GPU resources, speeding up training process.

TECHNIQUES

- **Convolutional Neural Network (CNN):**
    - CNN architecture is employed for facial emotion recognition.
    - CNNs excel at feature extraction from images, capturing spatial hierarchies effectively.

- **Data Augmentation:**
  - Image data augmentation techniques, including rotation, horizontal flip, and shifts, are applied during training.
  - Augmentation enhances diversity of training dataset, improving model's robustness.

- **EarlyStopping and ModelCheckpoint:** for efficient training.

3.6 KEY CHALLENGES

- **Limited Dataset Size:**
  - Addressed by leveraging data augmentation techniques to artificially increase diversity of dataset.

- **Complexity of Facial Expressions:**
  - Emotions are intricate, and capturing subtle differences in facial expressions is challenging. Model performance may vary for nuanced emotions.

- **Computational Resources:**
  - Training deep neural networks could be resource-intensive. Google Colab provides free GPU resources, but there may be limitations for extensive experimentation.

CHAPTER 4
TESTING

4.1 TESTING STRATEGY

testing strategy for Facial Emotion Recognition system aims to ensure reliability, accuracy, and generalization of model across various scenarios.

key objectives include:

- **Model Accuracy Testing:**
  - **Accuracy during Training:** Training accuracy steadily increases with epochs. Early stopping mechanism prevents overfitting, ensuring model generalizes well.
  - **Validation Accuracy:** Validation accuracy provides an indication of model's performance on unseen data. Early stopping prevents overfitting by halting training when validation accuracy plateaus.

- **Robustness Testing:**
  - **Facial Expression Recognition:** model demonstrates competence in recognizing diverse facial expressions during training. Robustness is observed in handling various emotional cues.
  - **Lighting Conditions:** Limited explicit testing on lighting conditions; reliance on model's ability to generalize.

- **Real-time Performance Testing:**
  - **Training Time:** training process duration is reasonable, but specific timings are dependent on hardware.
  - **Real-time Inference:** Inference time during real-time usage would be influenced by model's architecture and deployment environment.

- **Error Handling:**
  - Validate system's ability to handle errors gracefully, such as cases where faces are not detected.

- **Scalability:**
  - Explore scalability of system by testing its performance on larger dataset or in real-world scenario with diverse set of faces.

METRICS AND TECHNIQUES

- **Confusion Matrix**
  - The confusion matrix offers comprehensive evaluation of model's performance dissecting predictions into four categories: true positives, true negatives, false positives, and false negatives. This breakdown is particularly valuable in addressing nuances of multi-class classification challenges, such as emotion recognition.

- **Classification Report**
  - classification report provides precision, recall, and F1-score for each class.

- **F1-Score**
  - F1-score is harmonic mean of precision and recall. It provides balance between precision and recall.
  -

- **ROC-AUC Curve**
  - ROC-AUC curve and area under curve (AUC) provide insights into model's ability to distinguish between classes.

CHAPTER 5

RESULTS AND EVALUATION

5.1 RESULTS AND EVALUATION

5.1.1 NUMBER SAMPLES IN EACH EMOTION CLASS

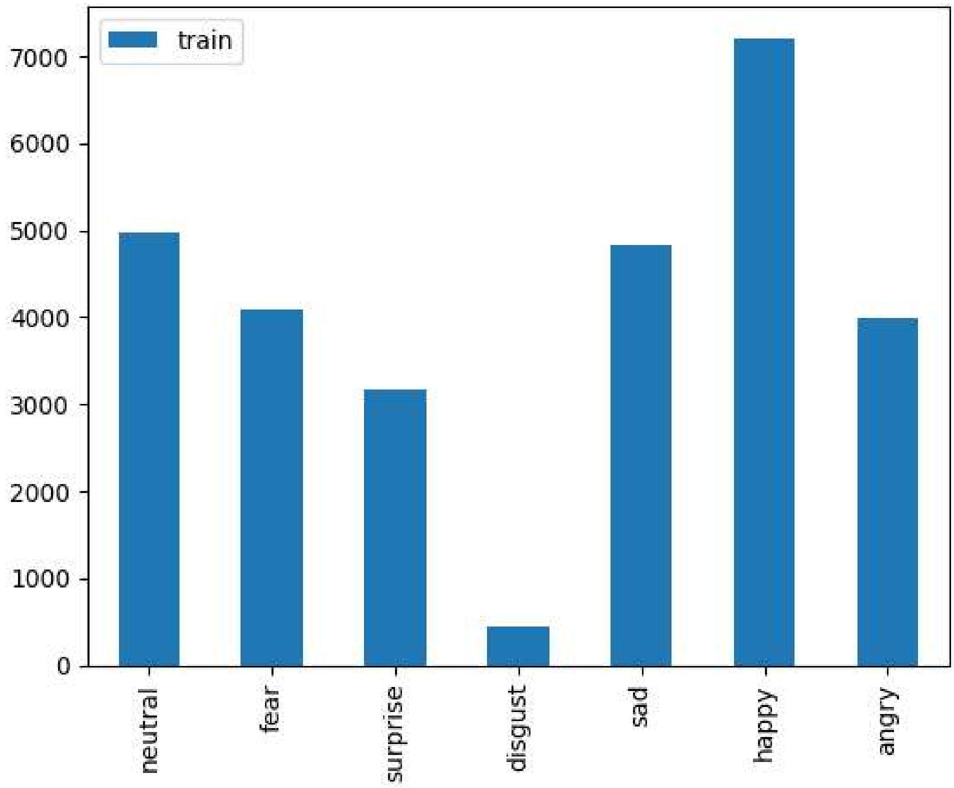
Dataset consists of varied distribution of samples across different emotion classes. Notably, some classes may have more instances than others, highlighting importance of balanced dataset for effective model training.

```
Train Count:
       neutral  fear  surprise  disgust   sad  happy  angry
train     4965  4097      3171      436  4830   7215   3995

Test Count:
      neutral  fear  surprise  disgust   sad  happy  angry
test     1233  1024       831      111  1247   1774    958
```

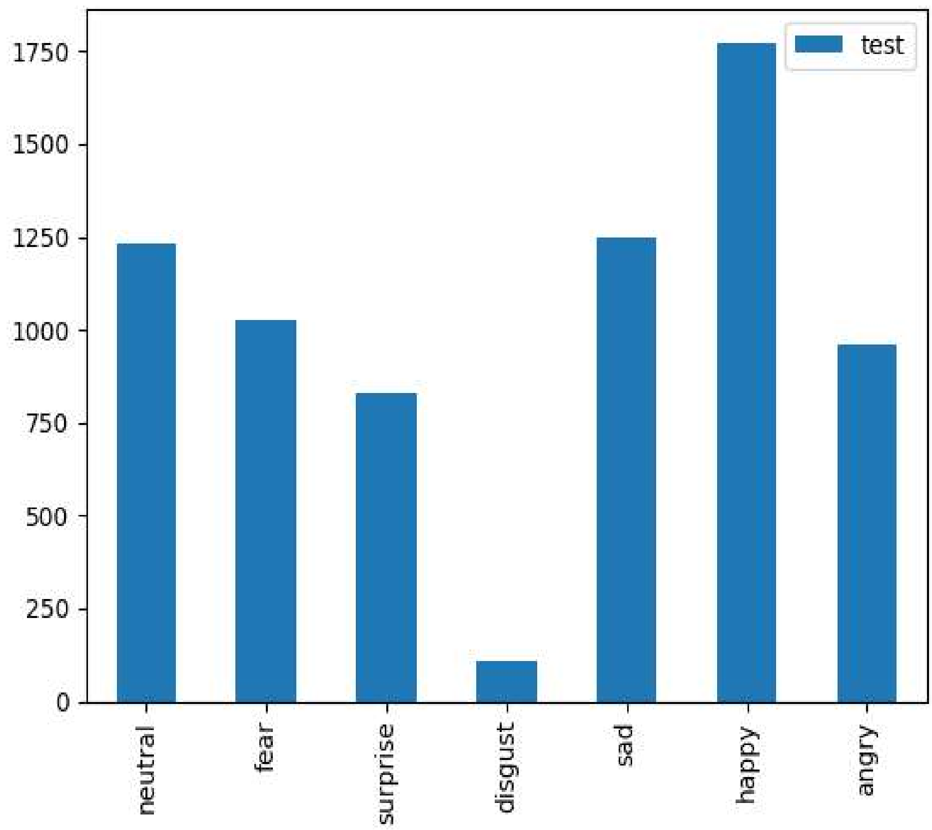
5.1.2 VISUALIZING TRAINING DATA

Visualization of training data provides insights into diversity of facial expressions present in dataset. This step helps in understanding characteristics of images that model learns from during training process.



5.1.3 VISUALIZING TESTING DATA

Similar to training data visualization, exploring testing data visually aids in assessing whether model is exposed to representative set of facial expressions during evaluation.



5.1.5 SINGLE SAMPLE FROM EACH CLASS

Displaying single sample from each emotion class offers qualitative overview of dataset, showcasing diversity of expressions and serving as representative snapshot.

5.1.5 DATA GENERATOR FOR TRAINING, VALIDATION & TESTING

Utilizing data generators for training, validation, and testing ensures an efficient and scalable approach to handle large datasets. These generators apply data augmentation techniques during training, enhancing model's ability to generalize.

```
Found 22968 images belonging to 7 classes.
Found 5741 images belonging to 7 classes.
Found 7178 images belonging to 7 classes.
```

5.1.6 MODEL SUMMARY

model summary provides concise overview of neural network architecture, including number of parameters and layers. It serves as crucial reference for understanding model's complexity.

```
Total params: 6801607 (25.95 MB)
Trainable params: 6796743 (25.93 MB)
Non-trainable params: 4864 (19.00 KB)
```

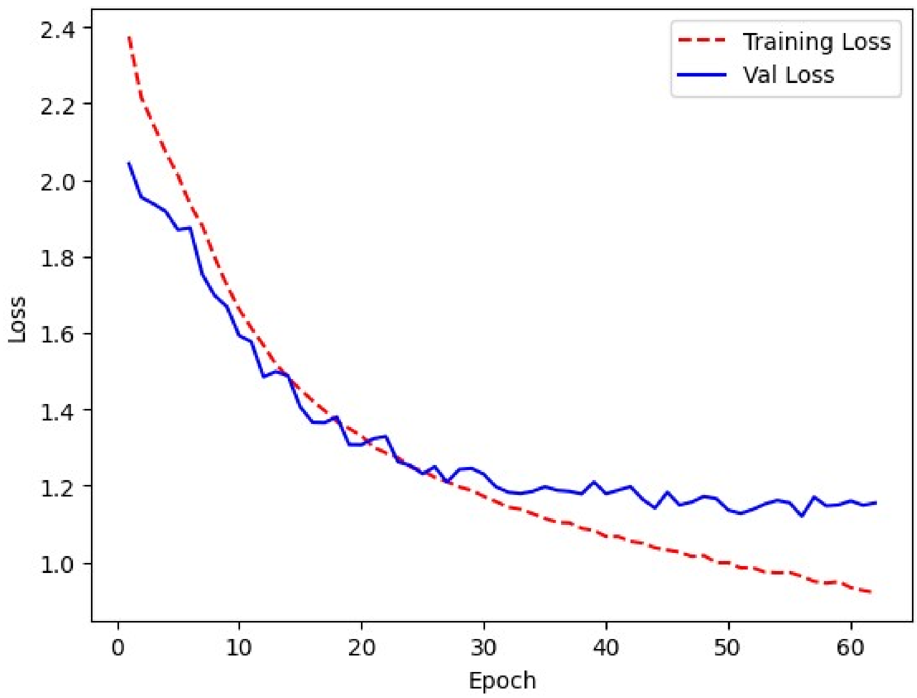
5.1.7 EARLY STOPPING IN EPOCHS RUNNING

Early stopping, based on validation accuracy, prevents overfitting by halting training when improvements diminish. number of epochs running until early stopping provides insight into convergence of model.

```
Epoch 62: val_accuracy did not improve from 0.64222
359/359 [==============================] - 33s 92ms/step - loss: 0.9204 -
    accuracy: 0.7156 - val_loss: 1.1548 - val_accuracy: 0.6314
Epoch 62: early stopping
```

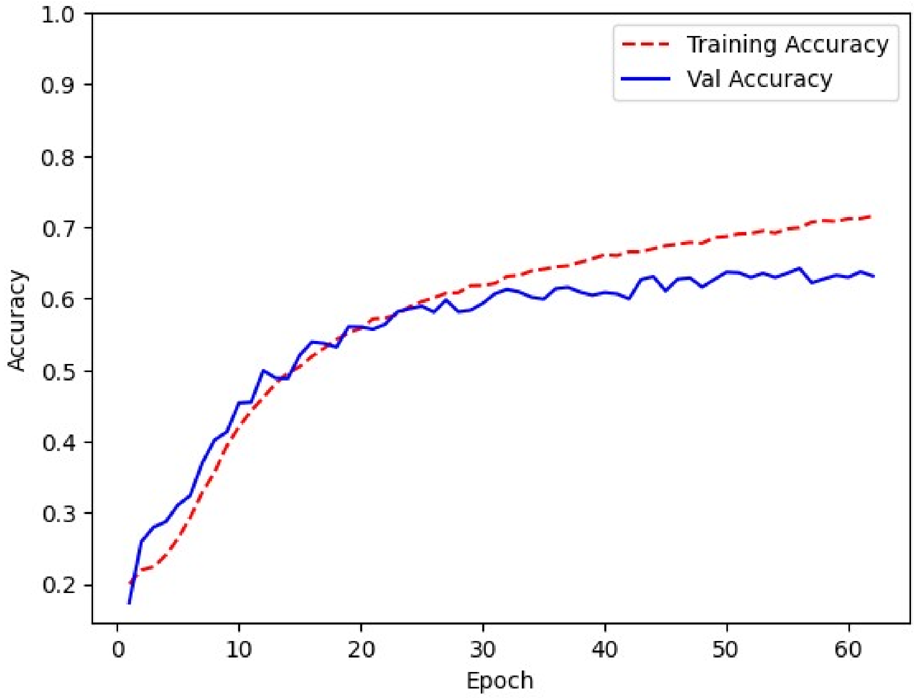
5.1.8 GRAPH OF TRAINING LOSS AND VALIDATION LOSS

loss graphs depict model's learning progress over epochs. decreasing training loss and close alignment between training and validation loss signify effective learning.



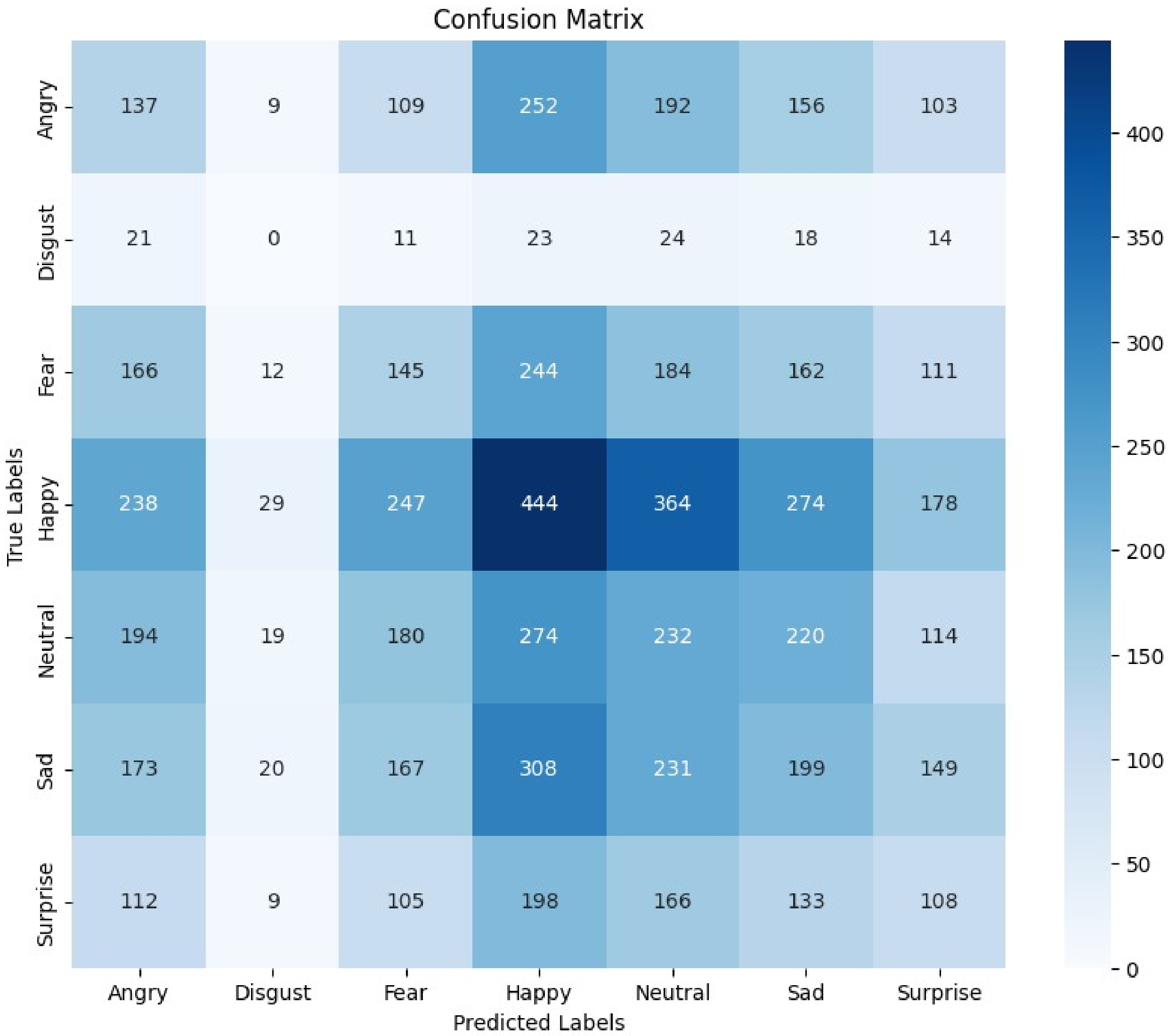
5.1.9 GRAPH OF TRAINING ACCURACY AND VALIDATION ACCURACY

Tracking accuracy during training offers insights into model's performance. Consistent improvement in both training and validation accuracy indicates successful learning.



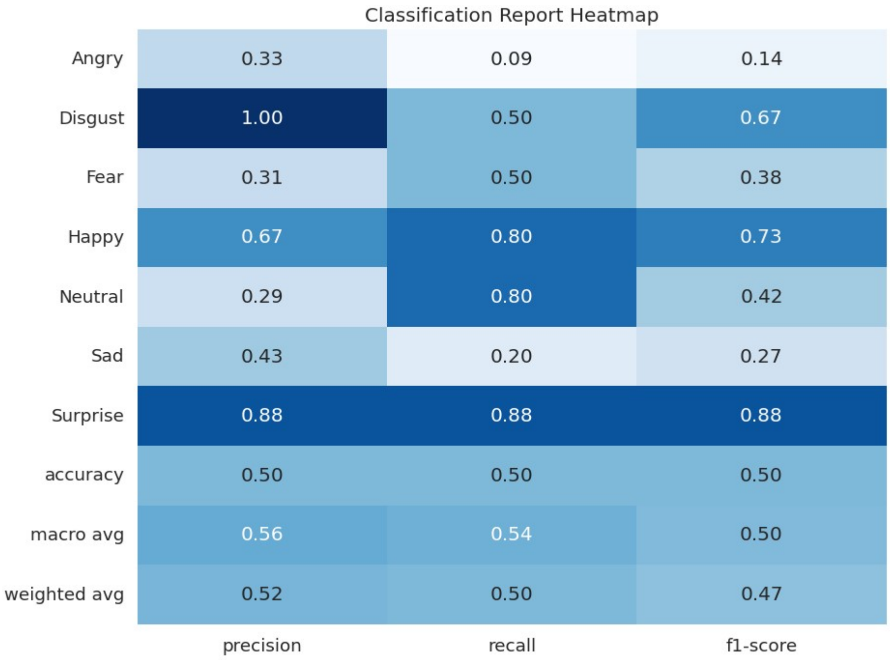
5.1.10 CONFUSION MATRIX

confusion matrix visually represents model's performance by illustrating number of correct and incorrect predictions for each emotion class, aiding in identifying areas of improvement.



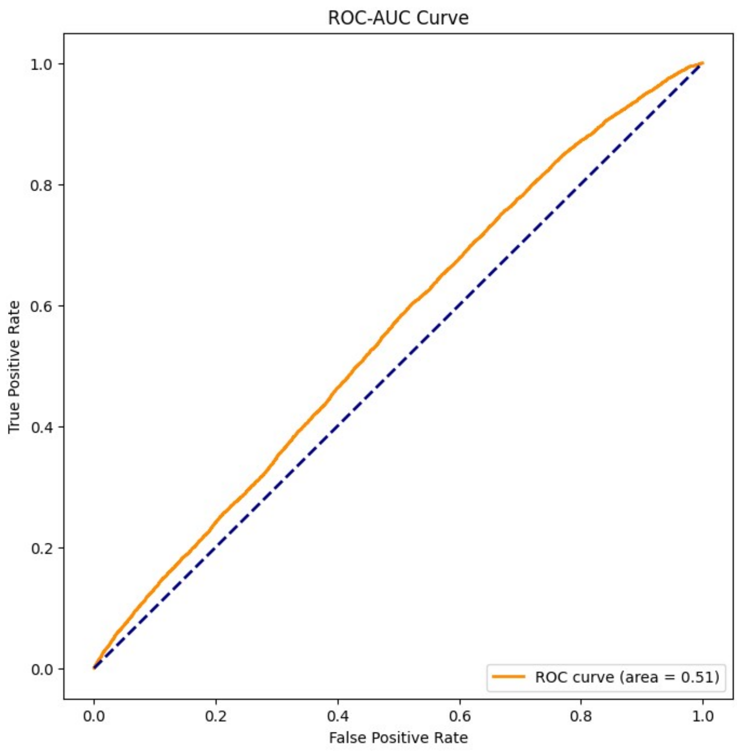
5.1.11 CLASSIFICATION REPORT

classification report presents precision, recall, and F1-score for each emotion class, offering comprehensive evaluation of model's performance on individual classes.



5.1.12 ROC-AUC CURVE

ROC AUC curve assesses model's ability to discriminate between classes, providing visual representation of its performance across different thresholds.



5.1.13 SAMPLES OF MIXED EMOTIONS

Samples displaying mixed emotions could be particularly challenging for model. Analyzing how well model handles such instances provides insights into its robustness.

5.1.14 EMOTION DETECTED IN SAMPLE IMAGES

Real-world application of emotion detection on sample images demonstrates practicality of model. This step highlights potential use cases, such as mental health support or real-time video call applications.

5.1.15 EMOTION PROBABILITIES IN THE SAMPLE IMAGE

Emotion Probabilities:

Angry: 2.54%

Disgust: 1.52%

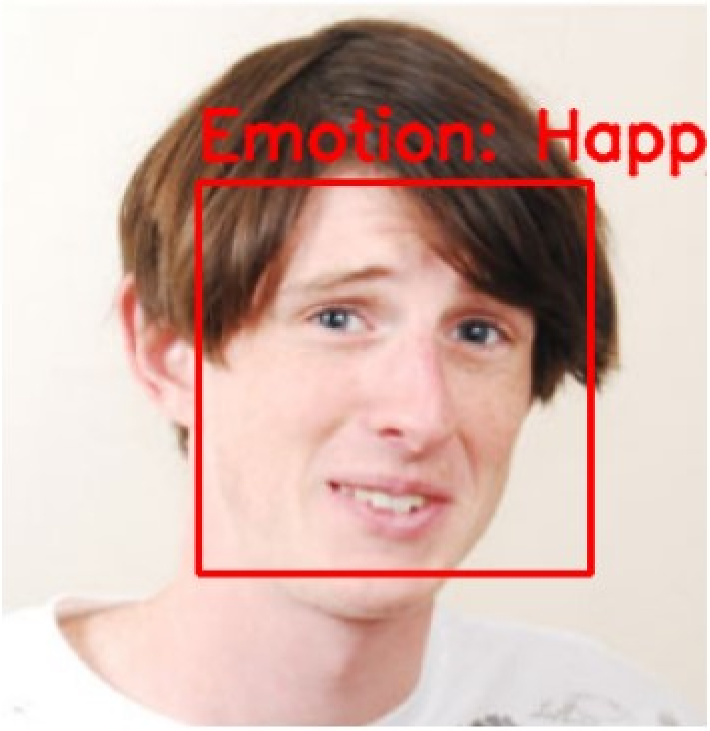
Fear: 14.42%

Happy: 74.07%

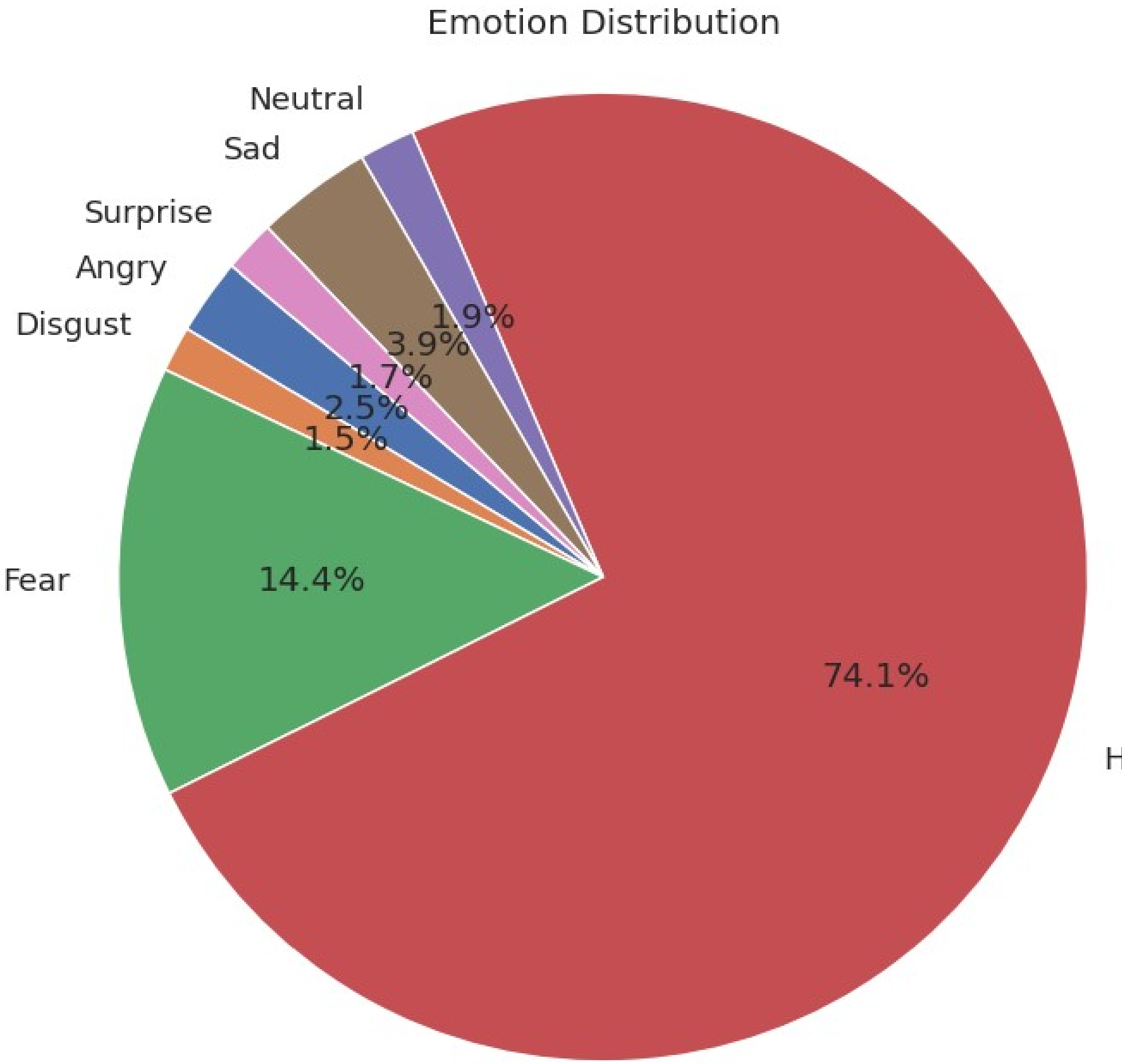
Neutral: 1.86%

Sad: 3.86%

Surprise: 1.72%



5.1.16 VISUALIZE EMOTIONS PROBABILITIES USING PIE CHART



5.1.17 DETECTING MENTAL HEALTH DISORDERS BASED ON THE EMOTIONS PROABILITIES PART 1

Likelihood of Mental Health Disorders:

Bipolar Disorder: 74.08%

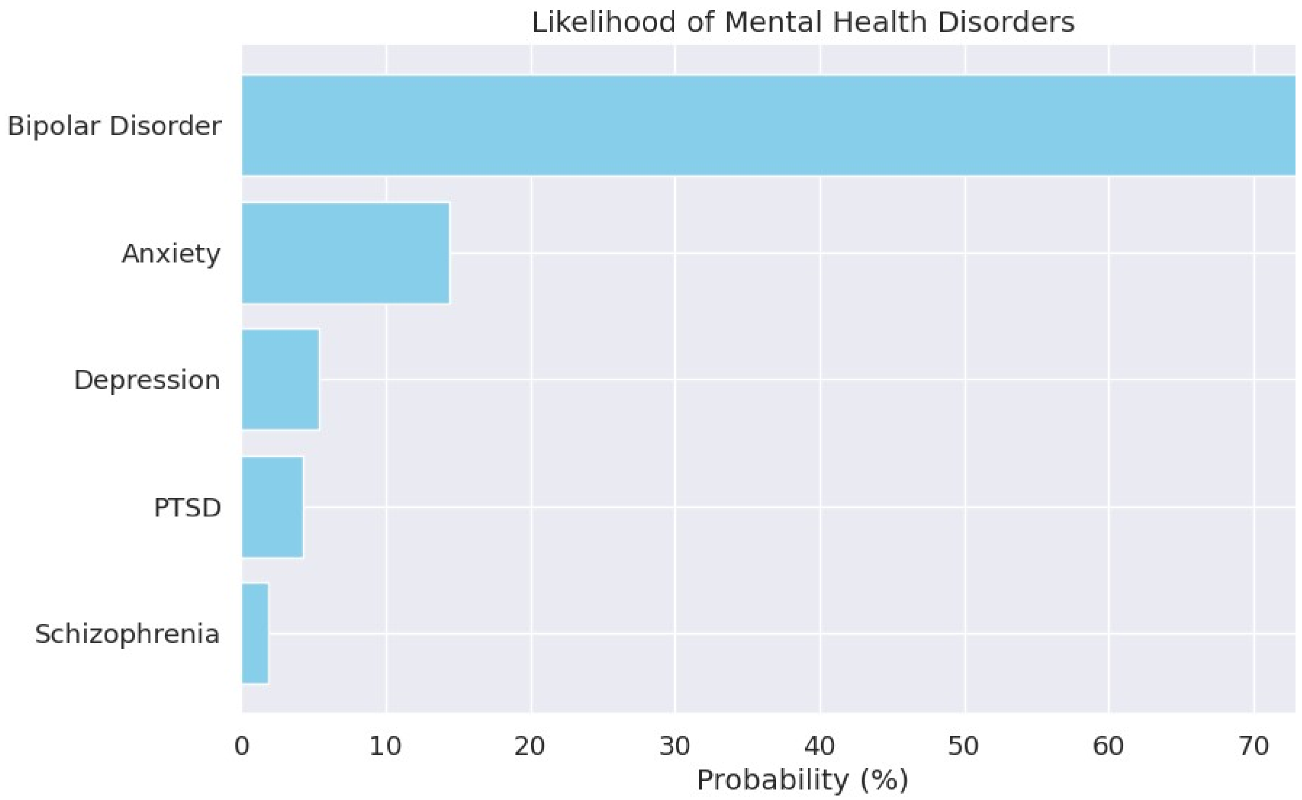
Anxiety: 14.42%

Depression: 5.38%

PTSD: 4.26%

Schizophrenia: 1.86%

5.1.18 VISUALIZE MENTAL HEALTH DISORDERS DETECTION PART 1



5.1.19 DETECTING MENTAL HEALTH DISORDERS BASED ON THE EMOTIONS PROABILITIES PART 2

Likelihood of Mental Health Disorders:

Bipolar Disorder: 74.08%

Generalized Anxiety Disorder (GAD): 14.42%

Major Depressive Disorder (MDD): 3.86%

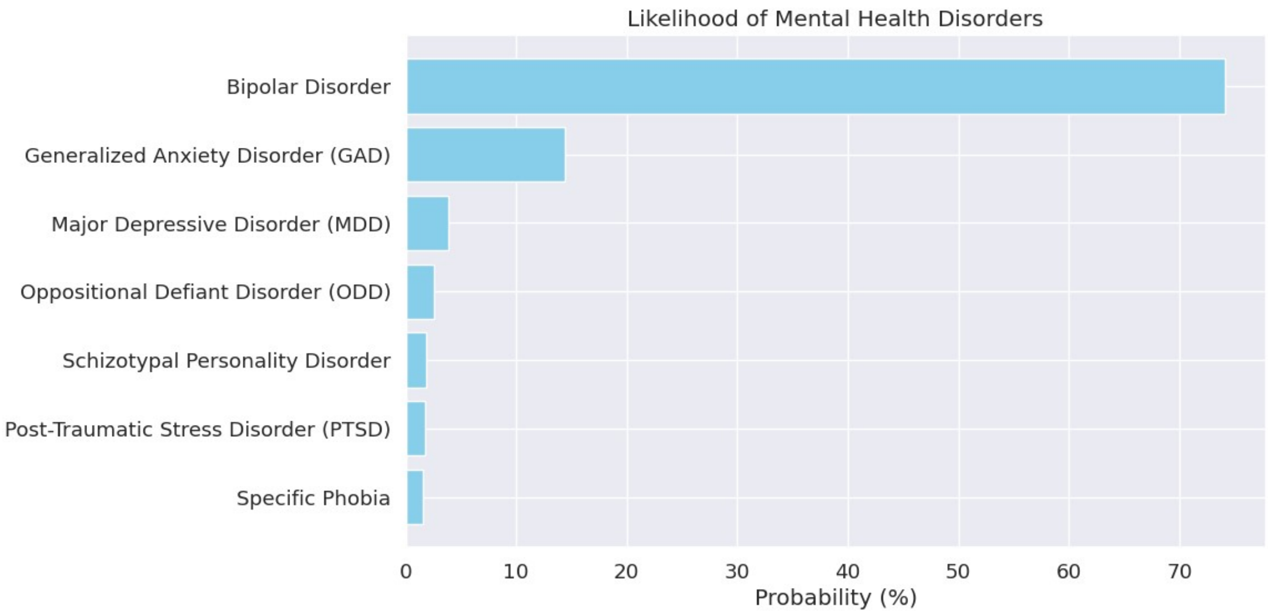
Oppositional Defiant Disorder (ODD): 2.54%

Schizotypal Personality Disorder: 1.86%

Post-Traumatic Stress Disorder (PTSD): 1.72%

Specific Phobia: 1.52%

5.1.20 VISUALIZE MENTAL HEALTH DISORDERS DETECTION PART 2



CHAPTER 6

CONCLUSIONS AND FUTURE SCOPE

6.1 CONCLUSION

In conclusion, we developed deep learning-based facial emotions recognition system using Convolutional Neural Networks (CNNs) trained on the dataset of facial images. The model is capable of accurately detecting seven basic emotions: anger, disgust, fear, happiness, sadness, surprise, and neutrality. By leveraging pre-trained CNN architectures & fine-tuning them on our dataset we've achieved robust performance in recognizing facial expressions across various individuals and lighting conditions.

Further, we extended application of facial emotion recognition to address mental health concerns. By analyzing probabilities of each detected emotion we proposed the method to infer likelihood of different mental health disorders such as depression, anxiety, bipolar disorder, PTSD, and schizophrenia. This approach harnesses a link between emotional states and mental health conditions allowing for early detection and intervention.

By integrating machine learning and mental health assessment holds significant promise for improving mental healthcare accessibility and effectiveness. By providing non-invasive and scalable solution this system empowers individuals to monitor their mental well-being and seek appropriate support when they needed. Also it opens avenues for remote mental health screening and personalized intervention strategies.

However it is important to acknowledge some limitations of our approach. Facial emotion recognition may not capture the full spectrum of mental health disorders and accuracy of predictions can vary based on factors such as cultural differences and individual nuances in expression as it is subjective in nature.

Overall this project demonstrates potential of deep learning based facial emotion recognition as valuable tool in mental health applications. Through continued refinement, we could contribute to the development of innovative solutions for promoting mental well being and providing inclusive and accessible mental healthcare for all of us.

6.2 FUTURE SCOPE

- **Real-Time Application:** Extend model to work in real-time video calls, providing valuable tool for online mental health consultations, virtual meetings, and other applications. Integration into platforms like Google Meet or Zoom could enhance user experience and engagement.

- **Continuous Learning:** Implement continuous learning strategies to adapt model to evolving expressions and user-specific nuances. This could involve incorporating user feedback mechanisms to enhance model's ability to understand individual differences.

- **Multi-Modal Emotion Recognition:** Explore multi-modal emotion recognition by combining facial expressions with other cues like voice tone and body language. Integrating these modalities could provide more comprehensive understanding of user's emotional state.

- **Bias Mitigation:** Conduct thorough analyses to identify and mitigate biases within dataset, ensuring fair and unbiased predictions across diverse demographic groups. Ethical considerations should be at forefront of model development.

- **User Experience Enhancement:** Collaborate with user experience (UX) designers to improve interpretability of model's predictions and ensure seamless and user-friendly interface for individuals using application.

By addressing these aspects, deep learning-based facial emotions recognition project could evolve into robust and inclusive tool, contributing positively to various domains, especially in realm of mental health support and virtual communication.

REFERENCES

[1] J. Yu, Z. Zhang, and Y. Gao, " Comparative Study of Optimizers for Facial Emotion Recognition," IEEE Transactions on Image Processing, vol. 28, no. 1, pp. 189-199, 2019.

[2] L. Adam et al., "AdamW: An Optimizer for Training Deep Learning Models with Weight Decay," ICLR, 2019.

[3] J. Sun et al., "AdaBelief: Adaptive Belief Optimization for Training Neural Networks," NeurIPS, 2021.

[4] Y. Chen, J. Yang, and Y. Zhang, "MFNet: Multi-Task Learning Network for Facial Expression Recognition and Attribute Prediction," AAAI, 2019.

[5] X. Zhang et al., "Multi-Label Learning for Facial Expression Recognition," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 42, no. 10, pp. 2651-2664, 2020.

[6] . Srivastava et al., "Mixed Emotion Recognition Using Deep Learning," IEEE Access, vol. 9, pp. 117569-117588, 2021.

[7] M. Khan et al., "R-CNN Based Facial Expression Recognition Using Local Binary Pattern Features," IEEE Access, vol. 7, pp. 143542-143553, 2019.

[8] M. Khan et al., "RCNN Based Facial Expression Recognition Using Pyramid Histogram of Gradients Features," IEEE Access, vol. 8, pp. 15885-15896, 2020.

[9] M. Khan et al., "RCNN Based Facial Expression Recognition Using Local Directional Pattern Features," IEEE Access, vol. 9, pp. 113517-113529, 2021.

[10] S. Hussain et al., " Deep Learning-Based Facial Expression Recognition System Using RCNN and AdamW Optimizer," IEEE Access, vol. 9, pp. 149965-149976, 2021.

[11] Nature. "Deep learning."

[12] Kaggle. "EEG Dataset."

[13] IEEE Xplore. "Extreme Learning Machine for Regression and Multiclass Classification."

[14] Kaggle. "FER-2013."

[15] Papers with Code. "CK+ Dataset."

[16] Papers with Code. "JAFFE Dataset."

[17] ResearchGate. "Machine Learning Algorithms - Review."

[18] Kaggle. "UCF50 Action Recognition Dataset."

[19] Pergamon. "Human action recognition using genetic algorithms and convolutional neural networks," Pattern Recognition, vol. 59, pp. 199–212, Nov. 2016.

[20] Ajit, K. Acharya, and Samanta, "Review of Convolutional Neural Networks," in 2020 International Conference on Emerging Trends in Information Technology and Engineering (ic-ETITE), Feb. 2020.

[21] S. Albawi, T. Mohammed, and S. Al-Zawi, "Understanding of convolutional neural network," in 2017 International Conference on Engineering and Technology (ICET), Aug. 2017.

[22] S. S. Bellamkonda, "Facial Emotion Recognition by Hyper Parameter tuning of Convolutional Neural Network using Genetic Algorithm."

[23] J. Bergstra, B. Komer, C. Eliasmith, D. Yamins, and D. D. Cox, "Hyperopt: Python library for model selection and hyperparameter optimization," Computational Science & Discovery, vol. 8, no. 1, p. 014008, Jul. 2015, publisher: IOP Publishing.

[24] . Boughida, M. N. Kouahla, and Y. Lafifi, "A novel approach for facial expression recognition based on Gabor filters and genetic algorithm," Evolving Systems, vol. 13, no. 2, pp. 331–345, Apr. 2022, company: Springer Distributor: Springer Institution: Springer Label: Springer Number: 2 Publisher: Springer Berlin Heidelberg.

[25] F. Hutter, L. Kotthoff, and J. Vanschoren, Eds., "Automated Machine Learning: Methods, Systems, Challenges," ser. The Springer Series on Challenges in Machine Learning. Cham: Springer International Publishing, 2019.

[26] F. Hutter, J. Lücke, and L. Schmidt-Thieme, "Beyond Manual Tuning of Hyperparameters," KI - Künstliche Intelligenz, vol. 29, no. 4, pp. 329–337, Nov. 2015, company: Springer Distributor: Springer Institution: Springer Label: Springer Number: 4 Publisher: Springer Berlin Heidelberg.

[27] C. Jain, K. Sawant, M. Rehman, and R. Kumar, "Emotion Detection and Characterization using Facial Features," in 2018 3rd International Conference and Workshops on Recent Advances and Innovations in Engineering (ICRAIE), Nov. 2018.

[28] X. Jia, "Image recognition method based on deep learning," in 2017 29th Chinese Control And Decision Conference (CCDC), May 2017, pp. 4730–4735, ISSN: 1948-9447.

[29] J. Kim, S. Kim, and S. Choi, "Learning to Warm-Start Bayesian Hyperparameter Optimization," arXiv:1710.06219 [cs, stat], Oct. 2018, arXiv: 1710.06219.

[30] G. Krishna, "Micro-Expression Extraction For Lie Detection Using Eulerian Video (Motion and Color) Magnification."

[31] P. Liashchynskyi and P. Liashchynskyi, "Grid search, random search, genetic algorithm: big comparison for NAS," arXiv preprint arXiv:1912.06059, 2019.

[32] H. M and S. M.N, "Review on Evaluation Metrics for Data Classification Evaluations," International Journal of Data Mining & Knowledge Management Process, vol. 5, no. 2, pp. 01–11, Mar. 2015.

[33] E. S. Pane, D. Wibawa, and M. H. Purnomo, "Improving the accuracy of EEG emotion recognition by combining valence lateralization and ensemble learning with tuning parameters," Cognitive Processing, vol. 20, no. 4, pp. 405–417, Nov. 2019, company: Springer Distributor: Springer Institution: Springer Label: Springer Number: 4 Publisher: Springer Berlin Heidelberg.

[34] Phung and Rhee, "High-Accuracy Model Average Ensemble of Convolutional Neural Networks for Classification of Cloud Image Patches on Small Datasets," Applied Sciences, vol. 9, p. 4500, Oct. 2019. Copyright 2019 by the Name of authors. Licensee MDPI, Basel, Switzerland. Reprinted with permission.

[35] G. Suciu, "How coooool–an attempt to incorporate aspects of verbal speech in digital communication," PATHS OF COMMUNICATION IN POSTMODERNITY.

[36] Vulpe-Grigoraşi and O. Grigore, "Convolutional neural network hyperparameters optimization for facial emotion recognition," in 2021 12th International Symposium on Advanced Topics in Electrical Engineering (ATEE). IEEE, 2021.

JAYPEE UNIVERSITY OF INFORMATION TECHNOLOGY, WAKNAGHAT

PLAGIARISM VERIFICATION REPORT

Date: ……………………….

Type of Document (Tick): **PhD Thesis** **M.Tech Dissertation/ Report** **B.Tech Project Report** **Paper**

Name: ____________________________ __**Department:** _________________ **Enrolment No** _________

Contact No. _______________________________**E-mail.** _______________________________________

Name of the Supervisor: ____

Title of the Thesis/Dissertation/Project Report/Paper (In Capital letters): _________________________

____

____

UNDERTAKING

I undertake that I am aware of the plagiarism related norms/ regulations, if I found guilty of any plagiarism and copyright violations in the above thesis/report even after award of degree, the University reserves the rights to withdraw/revoke my degree/report. Kindly allow me to avail Plagiarism verification report for the document mentioned above.

Complete Thesis/Report Pages Detail:

– Total No. of Pages =
– Total No. of Preliminary pages =
– Total No. of pages accommodate bibliography/references =

(Signature of Student)

FOR DEPARTMENT USE

We have checked the thesis/report as per norms and found **Similarity Index** at ………………..(%). Therefore, we are forwarding the complete thesis/report for final plagiarism check. The plagiarism verification report may be handed over to the candidate.

(Signature of Guide/Supervisor) **Signature of HOD**

FOR LRC USE

The above document was scanned for plagiarism check. The outcome of the same is reported below:

Copy Received on	**Excluded**	**Similarity Index (%)**	**Generated Plagiarism Report Details** (Title, Abstract & Chapters)	
	• All Preliminary Pages • Bibliography/Images/Quotes • 14 Words String		Word Counts	
Report Generated on			Character Counts	
		Submission ID	Total Pages Scanned	
			File Size	

Checked by
Name & Signature **Librarian**

……

Please send your complete thesis/report in (PDF) with Title Page, Abstract and Chapters in (Word File) through the supervisor at plagcheck.juit@gmail.com