Facial Expression Recognition using Convolutional Neural Networks

Kosisochukwu Andrew Ibe

Submitted to the
Institute of Graduate Studies and Research
in partial fulfillment of the requirements for the degree of

Master of Science
in
Computer Engineering

Eastern Mediterranean University
August 2023
Gazimağusa, North Cyprus

Approval of the Institute of Graduate Studies and Research

Prof. Dr. Ali Hakan Ulusoy
Director

I certify that this thesis satisfies all the requirements as a thesis for the degree of Master of Science in Computer Engineering.

Prof. Dr. Zeki Bayram
Chair, Department of Computer Engineering

We certify that we have read this thesis and that in our opinion it is fully adequate in scope and quality as a thesis for the degree of Master of Science in Computer Engineering.

Assoc. Prof. Dr. Adnan Acan
Supervisor

Examining Committee

1. Assoc. Prof. Dr. Adnan Acan
2. Assoc. Prof. Dr. Mehtap Köse Ulukök
3. Asst. Prof. Dr. Ahmet Ünveren

ABSTRACT

The ability to recognize facial expressions is extremely useful in a variety of fields, such as interaction between humans and computers, computational neuroscience, and robotics for social purposes. It is possible to enable a broad variety of applications by having the ability to reliably categorize and interpret facial expressions from photos or videos. Some examples of these applications are cognizant systems and smart user interfaces for computers. The purpose of this thesis is to study and create a model with enhanced efficiency and accuracy in recognizing facial expressions using Convolutional Neural Networks (CNNs).

Firstly, a comprehensive literature review will be done in order to analyze the cutting-edge CNN-based models and approaches that are currently being used for facial expression recognition identifying important hurdles, recent accomplishments, and possible chances for improvement in this field. This lays a sturdy groundwork for the future construction of the experimental model. The methodology for this study comprises a 10-layer CNN model incorporated with a set of FER2013 data which includes 7 classes of emotions. The dataset would be preprocessed and enhanced using data augmentation before using 5 K-fold cross validation to fit the data to the model.

The trained model will be evaluated utilizing a wide variety of performance metrics, among which are accuracy, precision, recall, and F1-score. The effect of many parameters on recognition performance is explored in this thesis. These aspects include dataset size, model architecture, and hyperparameter tweaking.

The outcomes of the experiments are meticulously analyzed, and evaluated in light of the most recent developments in the field as well as any relevant benchmarks. The findings contribute to a deeper knowledge of the possibilities as well as the limitations that are present in facial expression recognition using CNNs.

Keywords: Facial Expression Recognition, Deep Learning, Neural networks.

ÖZ

Yüz ifadelerini tanıma yeteneği, insanlar ve bilgisayarlar arasındaki etkileşim, hesaplamalı sinirbilim ve sosyal amaçlar için robotik gibi çeşitli alanlarda son derece yararlıdır. Fotoğraflardan veya videolardan yüz ifadelerini güvenilir bir şekilde kategorize etme ve yorumlama becerisine sahip olarak çok çeşitli uygulamalara olanak sağlamak mümkündür. Bilişsel sistemler ve bilgisayarlar için akıllı kullanıcı arayüzleri bu uygulamalara örnek olarak verilebilir. Bu tezin amacı, Konvolüsyonel Sinir Ağları (CNN'ler) kullanarak yüz ifadelerini tanımada etkinliği ve doğruluğu artırılmış bir modeli incelemek ve oluşturmaktır.

İlk olarak, yüz ifadesi tanıma için şu anda kullanılmakta olan CNN tabanlı en yeni modelleri ve yaklaşımları analiz etmek için kapsamlı bir literatür taraması yapılacak ve bu alanda önemli engelleri, son başarıları ve olası iyileştirme şanslarını belirleyeceğiz. Bu, deneysel modelin gelecekteki inşası için sağlam bir temel oluşturur. Bu çalışmanın metodolojisi, 7 duygu sınıfı içeren bir dizi FER2013 verisiyle birleştirilmiş 10 katmanlı bir CNN modelini içermektedir. Veri kümesi, verileri modele uydurmak için 5 K-katlı çapraz doğrulama kullanılmadan önce veri artırma kullanılarak önceden işlenir ve geliştirilir.

Eğitilen model, doğruluk, kesinlik, geri çağırma ve F1 puanı gibi çok çeşitli performans ölçütleri kullanılarak değerlendirilecektir. Bu tezde birçok parametrenin tanıma performansı üzerindeki etkisi araştırılmıştır. Bu yönler, veri kümesi boyutunu, model mimarisini ve hiperparametre ince ayarını içerir. Deneylerin sonuçları titizlikle analiz edilir ve alandaki en son gelişmelerin yanı sıra ilgili kıstaslar ışığında değerlendirilir. Bulgular, CNN'leri kullanarak yüz ifadesi tanımada mevcut olan sınırlamaların yanı sıra olasılıklar hakkında daha derin bir bilgiye katkıda bulunur.

Anahtar Kelimeler: Yüz İfadesi Tanıma, Derin Öğrenme, Sinir ağları.

DEDICATION

To my family

ACKNOWLEDGEMENT

I would like to thank my supervisor, Associate Professor Dr. Adnan Acan, for all the help, patience and knowledge he's given me over the years. His perceptive assessment and consistent support have had a major impact on the course and value of this research. I am grateful to him for inspiring me to explore new concepts.

My family, especially my mom, dad, and brother and sisters, have been tremendously supportive of me in this attempt. I am grateful for their confidence and understanding as I poured endless hours into research and writing. Because of their unconditional love and support, I was able to push through my limitations and achieve my academic goals.

A big thanks to everyone who has contributed to this study in any way, whether by helpful suggestions, technical advice, or just good vibes. Your help, in whatever form it took, has been tremendously appreciated.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

AAM	Active appearance models
ASM	Active shape models
ADAM	Adaptive moment estimation
CNN	Convolutional neural network
FER	Facial expression recognition
FER2013	Facial expression recognition 2013
GMM	Gaussian mixture models
HMM	Hidden markov models
HOG	Histogram of oriented gradients
kNN	K-nearest neighbors
LDA	Linear discriminant analysis
LBP	Local binary patterns
PCA	Principal component analysis
ReLU	Rectified linear activation function
RGB	Red, green and blue
SIFT	Scale-invariant feature transform
SVM	Support vector machines

Chapter 1

INTRODUCTION

In understanding and detecting emotions, facial expressions play a key role. This is well discussed in pattern recognition and image processing. The word “interface” indicates how significant the face is in communication between two entities [1]. Although humans are instinctively able to comprehend emotions, it has remained a difficult task for computers. Research has shown that the understanding of what is said in a discussion between two entities can dramatically affect the reading of facial expressions and influence the flow of a conversation [2]. Hence, recognizing facial expressions can help build a better user experience solving problems related to image processing, human computer interaction and security [3].

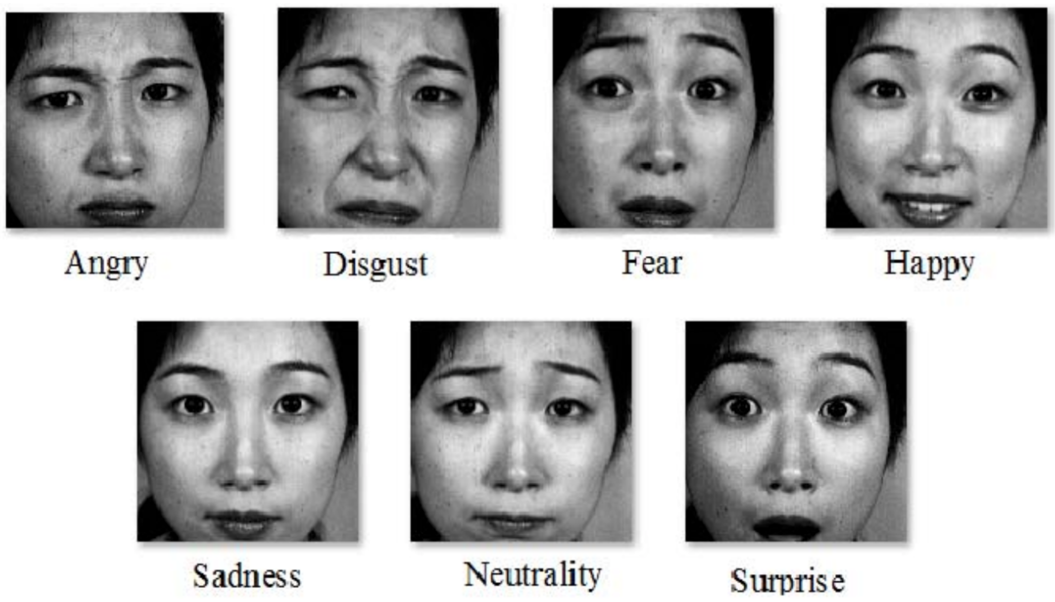


Figure 1: Facial expressions from FER2013 dataset. [4]

1.1 Background and Motivation

Early studies in the area of facial expression recognition relied on human observers to manually analyze facial features and emotions. However, new developments in computer vision and machine learning have resulted in widespread adoption of automated facial expression recognition systems. Facial expression recognition is the study and practice of identifying and analyzing emotional states from a person's facial expressions [5]. It reveals a great deal about a person's emotional state, intentions, and attitudes, making them an integral part of communication and social interactions. In recent years, facial expression recognition has become increasingly important in various uses in the real world, such as communicating with robots, keeping tabs on drivers, and diagnosing mental disorders.

Traditional approaches to facial expression recognition usually depend on handcrafted features and low-depth classifiers, such as Support Vector Machines (SVMs) and Random Forests. However, these approaches often suffer from limitations such as poor performance and sensitivity to variations in lighting, pose, and occlusions. In light of the recent developments in deep learning methods, particularly Convolutional Neural Networks (CNNs), significant advancements have been made in facial expression recognition.

CNNs are a specific kind of neural network that can automatically learn hierarchical representations of data, such as images [6]. CNNs have excelled spectacularly in many computer vision applications, such as object recognition, image segmentation, and face recognition. In facial expression recognition, CNNs have exhibited superior performance relative to traditional approaches, achieving state-of-the-art results on benchmark datasets such as the FER2013 dataset.

This thesis aims to investigate the potential of CNNs for facial expression recognition and compare their performance to that of conventional methods. This thesis seeks to contribute to the development of more efficient and precise facial expression recognition systems by identifying the strengths and weaknesses of CNNs in facial expression recognition. The optimal hyperparameters for CNNs in facial expression recognition will aid in the design and implementation of future facial expression recognition systems.

1.2 Problem Statement

Facial expression recognition is difficult since people's faces change depending on their surroundings. Differences in lighting, expressions, positions, and occlusions are all examples of what might go wrong. Handcrafted features and shallow classifiers have been the backbone of previous attempts at facial expression recognition. Poor performance and sensitivity to changes in illumination, posture, and occlusion are common issues with these approaches. This is due to the fact that these techniques rely on fixed characteristics, which may not transfer well to new situations or people.

Convolutional Neural Networks (CNNs) and other deep learning approaches have made significant progress in facial expression recognition in recent years. CNNs can learn representations of facial expressions automatically from raw visual data, unlike prior methods. Nonetheless, despite their achievements, CNNs for facial expression recognition still face a number of obstacles.

Obtaining enough labeled training data is a major challenge when employing CNNs for facial expression recognition. In order for CNNs to learn correct representations of facial expressions, it is necessary to have a huge volume of labeled data., which can be difficult and expensive to collect. Furthermore, the diversity of the subjects' expressions and the demographic representation of the subjects in the training data can significantly affect the efficiency of CNNs.

The choice of optimal hyperparameters, such as the number of layers, filter size, and learning rate, is another challenge in CNN-based facial expression identification. Finding the optimal settings for a CNN is challenging and time-consuming because its effectiveness is very sensitive to these hyperparameters.

Moreover, there are no standard evaluation metrics or benchmark datasets, this complicates the task of comparing and contrasting the results of various facial expression recognition algorithms. In order to conduct valid comparisons amongst algorithms, it is essential that standardized evaluation measures and benchmark datasets be developed.

1.3 Objective

In light of the above mentioned difficulties, the objective of this thesis is to investigate the potential of convolutional neural networks (CNNs) for facial expression recognition and to address the problems that arise during their implementation. The following are some of the specific concerns that will be investigated:

1. How can convolutional neural networks be trained to recognize expressions from a small sample of faces?

2. When it comes to identifying emotional states, what hyperparameters should CNNs use?
3. How can the effectiveness of CNNs in facial expression recognition be measured, and how do they stack up against more conventional methods?
4. When it comes to identifying emotions, what limitations do CNNs have, and how might they be fixed?

This study aims to answer these questions by looking into various hyperparameter settings to improve facial expression recognition performance and novel approaches for training CNNs with minimal data.

1.4 Scope and Limitations

In this thesis, we explore the use of Convolutional Neural Networks (CNNs) to the problem of recognizing face expressions. Targeted areas include:

1. Using CNNs for facial expression recognition: a feasibility study.
2. Implementing and assessing the efficacy of a CNN-based system for recognizing facial expressions.
3. Seeing where CNNs fall short and suggesting improvements for facial expression recognition.

Theses have limitations and this one does too:

1. Only freely available FER datasets will be used in this research, which may not be representative of real-world conditions.

2. Grayscale facial expression photographs will be used for the investigation; it's possible that the system's performance would change if color images were used.
3. Pose, illumination, and occlusions are not addressed in this thesis but might affect the efficiency of the system..
4. The research will only compare the CNN-based system's results to those of more conventional methods if specified evaluation metrics are reached.

Chapter 2

LITERATURE REVIEW

Throughout computer vision and artificial intelligence, facial expression recognition has been a hotspot for research. Convolutional neural networks (CNNs) have recently emerged as a useful tool for facial expression recognition due to their ability to learn complicated visual data. In this section, we will take a look back at what has been written on CNNs and FER.

2.1 Facial Expression Recognition

The aim of facial expression recognition is to give machines the ability to read people's emotions through their facial expressions. The human face is capable of displaying a wide range of emotions such as happy, sad, anger, fear, surprise, neutral and disgust. They were unable to keep the recognition rates for the various classes constant due to the fluctuation [7]. FER has many possible applications in fields as diverse as psychology, robotics, and human computer interaction.

Local Binary Patterns (LBP), Histogram of Oriented Gradients (HOG), and the Scale-Invariant Feature Transform (SIFT) are examples of feature extraction techniques that were commonly used prior to the development of CNNs for FER. In order to categorize the facial expressions, these methods relied heavily on feature extraction from photographs and the use of machine learning techniques such as Support Vector Machines (SVM) and Random Forests.

Conventional methods had a fair amount of success in identifying facial expressions, but they could not identify relevant image attributes successfully. With the advent of convolutional neural networks (CNNs), which can learn high-level picture features directly, this obstacle was finally solved.

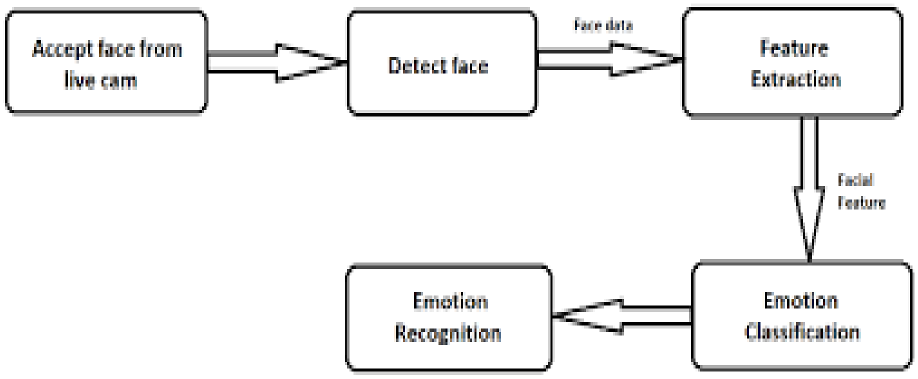


Figure 2: Facial expression recognition (FER) system architecture [8].

There are two types of approaches involved with facial expression recognition: appearance-based and model-based techniques.

2.1.1 Appearance-based Techniques

Appearance-based Techniques consist of examining the face in detail, based on features like shape, texture, and pigmentation. It is possible to further categorize these techniques by subdividing them into feature extraction and selection techniques.

Feature extraction techniques consist of capturing facial images and using extraction techniques to obtain useful data for classification. Local Binary Patterns, Histogram of Oriented Gradients, and Scale-Invariant Feature Transform are three common feature extraction techniques.

Feature selection techniques consist of picking out the most salient features to use in a classification. Principal Component Analysis (PCA) and Linear Discriminant Analysis (LDA) are two well-known feature selection techniques.

2.1.2 Model-based Techniques

Model-based techniques make use of mathematical models to represent the correlation between facial features and emotional states. These methods can be broken down even further into geometric and appearance models.

Geometric models entail representing the shape of the face using facial landmarks. Distances and angles are just two examples of the geometric properties that may be calculated from these landmark places and put to use in a classification system. Active Shape Models (ASM) and Active Appearance Models (AAM) are two widely used types of geometric models.

Appearance models use statistical models to characterize facial features. Emotional expressions can be captured by these models along with the physical changes to the face. Hidden Markov Model (HMM) and the Gaussian Mixture Model (GMM) are commonly used to describe external appearances.

2.1.3 Challenges in Facial Expression Recognition

Facial expression recognition is challenging because of issues like illumination, occlusion, and position changes. The scarcity of labeled data adds to the difficulty of training FER systems.

Techniques like data augmentation and transfer learning have been explored to overcome these challenges. Altering the existing training images through rotation, zooming and scaling can produce new data through a process called data augmentation. Using a convolutional neural network (CNN) that has been trained on a bigger dataset, like ImageNet, to initialize the weights of a CNN for FER is an example of transfer learning.

2.2 Convolutional Neural Networks

When it comes to identifying images, convolutional neural networks (CNNs) do exceptionally well. The ability to extract complex attributes from raw data and the scalability to handle big datasets have contributed to the rise in popularity of CNNs in recent years.

2.2.1 Architecture of CNNs

A convolutional neural network (CNN) has an architecture made up of many layers. There are four main categories for these layers: convolutional, pooling, fully connected and softmax.

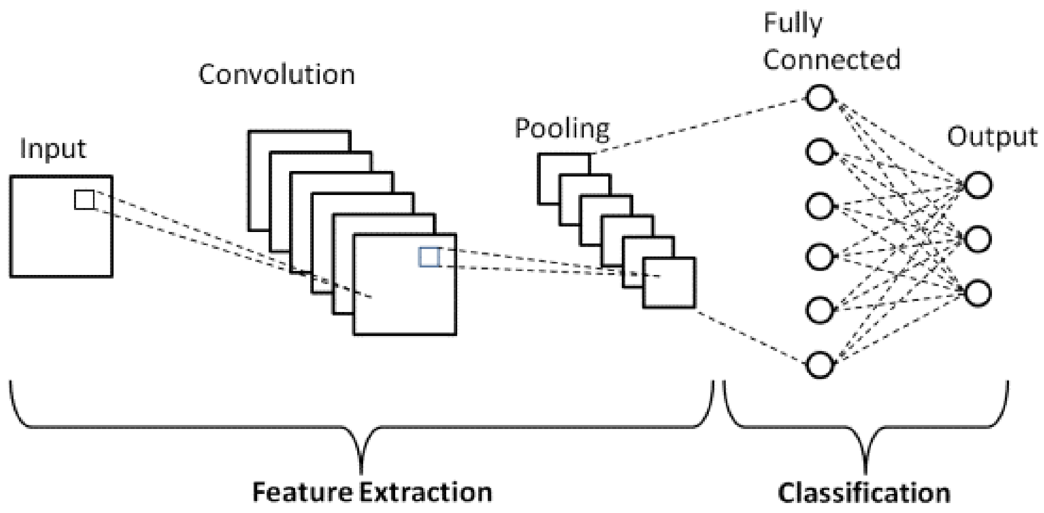


Figure 3: Basic convolutional neural network (CNN) architecture [9].

1. **Convolutional Layers:** CNNs rely on their convolutional layers to function. Each filter in this set convolves over the input image to generate a feature map. Each filter is trained to detect a certain attribute from the initial image.

2. **Pooling Layers:** Convolutional layer-produced feature maps are too large in space, thus the spatial dimensions are reduced using pooling layers. As a result, the network's computational complexity is lowered and its sensitivity to input image fluctuations is enhanced.
3. **Fully Connected Layers:** The resultant data from the last pooling layer is sent to a variety of fully connected layers. These layers, which function similarly to those in a classic neural network, are responsible for transferring the information learnt in the convolutional layers to the output classes.
4. **Softmax Layer:** This layer is specifically concerned with multi-class classification. It does this by taking the outputs generated from the fully connected layers and converting them into probability distribution of different classes. The class predicted will be as a result of the class having the highest probability.

2.2.2 Training of CNNs

CNNs are trained using the optimization algorithm backpropagation, which is gradient-based. The model is trained by showing it a series of classified images to train on and then adjusting its weights so that the discrepancy between the expected and true labels is as little as possible.

When it comes to training CNNs, the function that calculates cross-entropy loss is by far the most popular option. The network's weights are adjusted with the use of an optimization algorithm like stochastic gradient descent.

2.2.3 Challenges in Training CNNs

CNNs can be computationally intensive to train, particularly when dealing with large datasets. In addition, the limited availability of labeled data makes training CNNs difficult. One way to get around this problem is to use the weights of a network that has already been trained on a big dataset like ImageNet to perform the desired task.

2.2.4 Applications of CNNs

Applications for CNNs include image classification, object detection, and facial expression recognition. In medical image analysis, they have been utilized for duties such as tumor detection and segmentation.

There are many applications for CNNs in the field of machine learning for language processing, including sentiment analysis and language translation. A word or phrase sequence is used as data to the system, and a prediction of sentiment or a translated sentence is produced as result.

2.3 CNNs for Facial Expression Recognition

Convolutional Neural Networks (CNNs) are today a powerful tool in recognizing facial expression because of the way they can collect intricate attributes straight from images. CNNs find high-level image details through numerous layers of convolutional and pooling processes. In order to classify the data, the last layer's data is sent to a fully connected layer.

Multiple studies have demonstrated CNNs' efficacy for facial expression recognition. A CNN consisting of eight convolutional layers and triple fully connected layers, [10] attained an accuracy of 97.35 percent using the CK+ dataset.

2.4 Related Studies

There have been numerous CNN-based facial expression recognition studies. Here are some noteworthy studies:

"*Deep Facial Expression Recognition: A Survey*" [11];

In this paper, we provide an in-depth look into the various deep learning-related strategies currently in use for FER. The authors cover the typical datasets to be evaluated and criteria used in FER, as well as the many different architectures of deep learning that were recently utilized towards FER, such as convolutional neural networks (CNNs), recurrent neural networks (RNNs), and generative adversarial networks (GANs). They also discuss the difficulties and eventual advancements of deep learning within the area of FER.

"*Facial Expression Recognition via Deep Learning*" [12];

This research put forth a convolutional neural network (CNN) architecture for facial expression recognition, which achieved best-in-class results across multiple comparison datasets. The design that was suggested comprised a series of convolutional layers, subsequent to a pooling layer and finally a fully connected layer. The authors improved the network's efficiency by using a mixture of data augmentation and transfer learning.

"*Facial Expression Recognition Using Convolutional Neural Networks: State of the Art*" [13];

In this study, we compared and contrasted the latest approaches to facial expression recognition with convolutional neural networks. The authors reviewed several recent research and observed recurring problems, which included a high demand for classified data as well as the challenge of accounting for variations in illumination and position.

"*Facial Expression Recognition Using Deep Convolutional Neural Networks*" [14];

This research summed up the state of the art in deep convolutional neural networks for recognizing facial expressions. Several recent experiments were reviewed, and the authors' conclusions were drawn about the importance of pre-processing techniques like detection of faces and aligning in improving the network's accuracy. The authors address the challenges facing the area and offer various suggestions for future study.

A real-time and quick facial recognition model utilizing CNN was proposed by [15]. The process is divided into two phases by the writers. The network is trained on a computer and then deployed on a Field Programmable Gate Array (FPGA), during which it attains 99.25% accuracy, which is outstanding.

The cloud-based authentication services offered by Google, Amazon, as well as Microsoft were compared by [16] in their study. They used a dataset of 140 images. Conclusively, Microsoft Azure's Face API is considered to be the best overall [17]. Although Google does its finest work in terms of recognition accuracy while Amazon consistently performs above average.

"*An analysis of facial expression recognition under partial facial image occlusion*" [18];

This research surveyed the state of the art in facial expression recognition using deep learning methods like convolutional neural networks (CNNs). The authors reviewed several previous experiments and noted widespread problems, among them a dearth of standardized datasets and the challenge of dealing with position, illumination, and opacity differences. Future research objectives in the area of study were additionally looked into by the authors, and these included the incorporation of domain-specific knowledge and the utilization of multimodal data.

2.5 Comparison of CNN with Other Approaches

The use of convolutional neural networks (CNNs) has been demonstrated to perform better than more conventional methods for facial expression recognition on a number of benchmark datasets. Contrary to conventional methods, which necessitate features being manually created, CNN-based systems can learn complex characteristics independently from raw image data [19]. Given the intrinsic complexity of facial expressions and the fact that they can vary greatly depending on brightness and viewpoint, this can be extremely beneficial for facial expression recognition.

Local Binary Patterns (LBP) and Histogram of Oriented Gradients (HOG) are examples of the kinds of manually created characteristics that are often used by conventional approaches to depict facial expressions. Properties of facial expressions like texture and shape can be captured by these attributes [20]. However, these capabilities may not stand up to changes in lighting and poses without some fine-tuning [21].

Furthermore, conventional approaches often use classifiers like Support Vector Machines (SVM) or k-Nearest Neighbors (kNN) to divide up facial expressions. Despite their ease of use and interpretability, these classifiers may struggle to accurately reflect the intricate nonlinear connections between attributes provided and classification results [22].

Instead, CNN-based methods use fully connected layers to find a nonlinear relationship between provided attributes and classification results. This allows the system to recognize complex nonlinear associations regarding the provided attributes and classification results that might be missed by more traditional approaches [23].

CNN-based methods aren't without their own drawbacks, though. It can be time-consuming and expensive to obtain a sufficient amount of labeled data that would be used for training purposes. Furthermore, CNN-based techniques might be sensitive to aspects like illumination and position, especially when the data used for training doesn't seem comparable to the one used in testing [24].

Although there are benefits and drawbacks to both CNN-based and conventional approaches to facial expression recognition, the former is expected to remain a hot topic of study for the foreseeable future.

Chapter 3

METHODOLOGY

3.1 Data Collection and Preprocessing

Approximately 35,000 grayscale pictures (48x48 pixels) from the FER2013 dataset [25] were utilized for this research. Researchers looked through various public image databases for datasets that included facial expressions that met their needs. The intended spectrum of emotions shown in the photos were broad: anger, disgust, fear, happiness, sadness, surprise, and even neutral. Images were culled from a variety of digital resources, including image archives, social media, and internet data stores. They made sure to get shots of people from all walks of life and of all cultures, ages, and sexes. To create a diverse and representative collection, they included pictures from across a wide range of ages and genders. Images were manually tagged with labels indicating whether the subject was angry, disgusted, afraid, happy, sad, surprised or neutral. The labeling procedure was checked for accuracy and consistency by expert annotators or trained persons. Multiple specialists examined and confirmed the tagged photos to ensure the quality of data. During this stage of quality control, any discrepancies or mislabeled items were found and fixed. Images that didn't meet their proposed facial expressions requirements were taken out of the dataset. Following the completion of the labeling and validation processes, the images were grouped into a structured dataset. A matched picture-label dataset was created by matching each image with its related label. In order to aid model building and evaluation, the dataset was split into training, validation, and testing subsets.

3.1.1 Preprocessing Steps

Here are the steps typically followed in the pre-processing of the FER2013 dataset:

1. **Image Resizing:** Each image in the FER2013 dataset is resized to a consistent resolution. A common choice is 48x48 pixels, which provides a balance between preserving facial details and reducing computational complexity.
2. **Augmentation Techniques:** To enhance the model's generalizability and expand the dataset's variety, augmentation approaches might be used. Common augmentation techniques for FER tasks include rotation, horizontal/vertical flipping, random cropping, and adding small perturbations or noise to the images. Augmentation should be applied randomly to generate variations of the original images, effectively increasing the dataset size. The following operations will be applied randomly to each image in the training set:
    - Random rotation = 15 degrees
    - Random horizontal flip
    - Random width shift range = 0.1
    - Random height shift range = 0.1
    - Random shear range = 0.1
    - Random zoom range = 0.1
3. **Data Split:** Create a training, validation, and testing set using the FER2013 dataset once it has been preprocessed. On average, an 80:20 split is employed, with 80% of the dataset being utilized during

training, a subset of the data used in training would be used for validation while the remaining 20% will be directed toward testing.

The pre-processing steps will be implemented using the Python programming language and the Keras deep learning library.

3.2 Model Architecture

Our proposed CNN architecture will have several layers that explore how to examine input photos for important aspects and label them with one of seven possible facial expressions.

- **Input layer:** accepts the preprocessed and feature-selected images. Each image is represented by a feature vector of reduced dimensionality. The input layer is responsible for passing input images to subsequent layers for further processing and feature extraction. It functions as the CNN model's entry point, receiving input data and forwarding it throughout the network. The input layer does not introduce non-linearity or learnable parameters; rather, it serves as a conduit for propagating input data throughout the network architecture. Its primary function is to provide the input data necessary for subsequent layers to perform convolution, activation, pooling, and other operations to extract meaningful features and make predictions.
- **Convolutional layers:** Relevant features are extracted using convolutions performed with learnable filters by these layers. In our CNN model, we will implement three convolutional layers, then, after each of those, there will be a rectified linear activation function (ReLU), and a max-pooling layer. The input images are convolved by the primary convolutional layer, which is made up of 32 filters of size 3x3. By applying element-wise rectification to the convolved outputs, the ReLU activation function generates non-linearity.

This facilitates the capture of complex and discriminative characteristics. Consequently, the max-pooling layer cuts down on the amount of space taken by the feature maps while preserving the most essential data. On the outputs of the previous layer, the second and third convolutional layer employs 64 filters of size 3x3 and 128 filters of size 3x3 and replicates the same process. These convolutional layers and their corresponding activation functions and max-pooling layers collaborate to extract progressively more complex image features. By employing multiple filters and aggregating operations, CNN is able to recognize complex data patterns and hierarchical representations. This enables the model to learn and distinguish between various visual features, enhancing its ability to accurately classify and recognize facial expressions.

- **Flatten layer:** plays a crucial part in transforming the results of the convolutional layers. Its job is to take the multi-dimensional feature maps produced by the convolutional layers and turn them into a feature vector with a single dimension. This transformation facilitates efficient processing of the retrieved characteristics by the subsequent fully linked layers. The flatten layer removes the spatial structure and organizes the data linearly by flattening the feature maps. In essence, it reduces the feature maps' spatial dimensions to a single long vector. The completely connected layers can then perform their operations on the entire collection of features as a whole, rather than treating them as discrete points in space. The flatten layer does not add any non-linearity or learnable parameters. Its sole function is to reformat the information. In order for the fully linked layers to learn and make predictions from the extracted data, the feature maps must first be flattened. The CNN model is able to pick up both local and global details in the input data thanks to the flatten layer, which acts as a bridge between the convolutional and fully connected layers. Because of this, the network can use the hierarchical representations it has learned in the convolutional layers to make better choices in the later layers.

- **Fully connected layers:** function as the model's classification part and be trained to assign one of seven facial expressions to the extracted features. Our design will employ a total of 256 neurons in one layer. A ReLU is applied at the end of the layer. The 256-neuron, fully connected layer receives the feature vector after it has been flattened and applies the ReLU activation function. High-level representations of the input data can be captured by the model thanks to the layer's learning of complicated combinations of characteristics retrieved at earlier levels. The information is used to refine the features being learned. Seven neurons represent the facial expressions we're trying to categorize in the final output layer. It takes the data from the fully connected layers before it and applies the Softmax activation function to it. For each facial expression class, the softmax algorithm calculates a probability distribution. This enables us to understand the model's output as the percentage of occurrence of each facial expression in the given image. Our CNN model is able to successfully train discriminative representations of the extracted features and make predictions because it uses fully connected layers. Accurate facial emotion identification is made possible by the network's ability to capture complex patterns and correlations in the data through the use of numerous layers and activation functions.
- **Dropout layers:** are incorporated into the design of the network to prevent overfitting by randomly turning off some of the nodes. Both dropout layers

will have a dropout rate of 0.25 and 0.5 respectively in our model. By lowering the model's reliance on isolated features and connections, regularization is facilitated by the dropout layers. The dropout layers prevent the network from becoming overly dependent on certain activations by randomly eliminating neurons throughout the training process. In this way, the network is motivated to acquire more generalized and stable representations of the data. After the convolutional layers comes the first dropout layer. It disables a quarter of the neurons at random, giving the network a wider variety of features. To prevent the model from being overly specific to the training data and to increase its capacity to generalize to novel data, this layer serves as a regularizer. Before the last output layer, a second dropout layer is positioned. Applying dropout now helps to improve the model's generalization capabilities and decrease overfitting by introducing additional randomness and reducing interdependencies among the neurons. Our CNN model's performance on unknown data is greatly enhanced by the addition of dropout layers, which we use to effectively counteract overfitting. To prevent the network from becoming overly dependent on the training data, these layers are critical for increasing robustness.

- **Optimization and loss functions:** In order to learn more quickly, we employ the Adam optimization method with a learning rate of 0.001. Adaptive Moment Estimation, or Adam, is an optimization approach that takes the best features of momentum and RMSprop and mixes them. In order to achieve faster convergence and improved handling of sparse gradients, it dynamically modifies the learning rate for each parameter. To properly update the weights of our CNN model during training and optimize them to minimize the loss and enhance the model's performance, we will use the Adam optimizer. We use a loss function based on categorical cross-entropy to aid in the task of facial expression recognition. For issues involving many classes, the categorical cross-entropy loss function is frequently employed. It measures the degree to which the model's output probability distribution deviates from the actual labels. Optimizing the model's parameters is made possible by the categorical cross-entropy loss function, which rewards the right classifications and punishes inaccurate ones. To improve its capacity to recognize and classify facial expressions, the model learns to give greater probability to the correct labels by minimizing this loss. To efficiently train our CNN model for face expression detection, we combine the Adam optimization approach with the categorical cross-entropy loss function. To maximize accuracy and generalization to new data, an optimization method can tweak the model's weights to minimize a loss function.

Python's Keras deep learning library shall be used to implement the CNN architecture. We will evaluate the model's efficacy by first training it on the data set aside for training, then validating it on the subset of data used in training, and then testing it on the remaining data.

3.3 Training, Validation and Testing

Firstly, we assemble the dataset, ensuring that the images and their labels are paired correctly. This requires meticulously organizing the dataset so that each image is associated with its corresponding label, thereby forming a coherent training set.

The subsequent crucial stage is optimizing the CNN model's hyperparameters with the subset partitioned for validation. Among these hyperparameters are the optimal number of convolutional layers, number of filters in each convolutional layer, number of fully connected layers, number of neurons in each fully connected layer, learning rate, and dropout rate. By adjusting these hyperparameters, we can optimize the architecture and efficacy of the model. With the hyperparameters tailored, the CNN model can be trained using the Adam optimizer on the training set. With a learning rate of 0.001, the Adam optimizer assists in iteratively modifying the model's weights and preferences to reduce the discrepancy across the label distribution and expected probability distribution. During training, the sparse cross-entropy loss function will be fit to the model to efficiently learn the patterns and representations required for accurate facial expression recognition.

The training process entails performing a fixed number of iterations over the training set. Typically, this ranges from 50 to 100 epochs, though the precise number can differ based on the dataset and the model's convergence. Alternately, training may continue until the validation accuracy plateaus, which indicates that the model has reached its peak performance. During each epoch, the model refines its ability to discern facial expressions by updating its weights and biases based on the gradients computed during backpropagation.

Once training is complete, the testing set is used to evaluate the efficacy of the trained CNN model. This evaluation computes various performance metrics, including accuracy, precision, recall, and F1-score, to determine how well the model generalizes to unseen data and classifies facial expressions accurately. In addition, a confusion matrix is created to visually represent the distribution of predicted and true labels, revealing the model's strengths and shortcomings in recognizing specific facial expressions.

3.4 K-Fold Cross Validation

Cross-validation is widely used for data resampling in order to discover the true prediction error of a model and to tweak its settings. Five-fold stratified cross-validation is often utilized in practice. One parameter, k, determines the number of folds to form from a given data sample. Consequently, K-fold cross-validation is a prevalent name for the procedure. k=5 for 5-fold cross-validation, for instance, can be substituted for k in the model's reference when a specified value for k is provided. When performing this type of cross-validation, the dataset is split into k sections of equal size spontaneously. Cross-validation is performed on a single partition whereas the other k partitions are used for actual training. This procedure is carried out n rounds. Then, the algorithm's overall performance is calculated to show how well the strategy works. Cross-validation has an advantage over other methods because each data item is utilized for training and validation. Each observation is utilized n times during the training phase and once during the validation phase. This thesis employs a 5-fold cross validation, as depicted in the image below. K Fold divides all samples into equal-sized pleats (same as Leave One Out method). Folds are utilized to learn the prediction function, while the omitted fold is used to verify it.

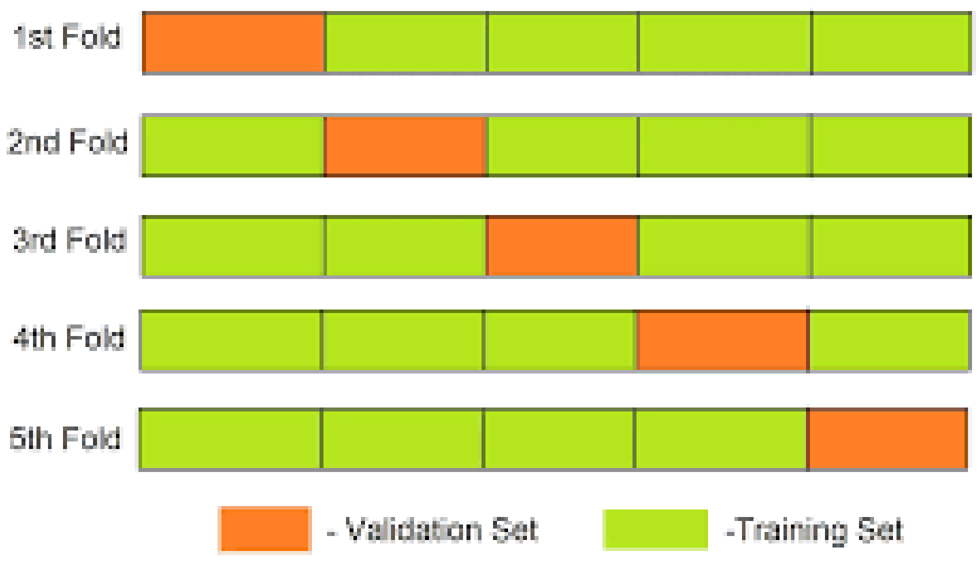


Figure 4: K-fold cross-validation in python using sklearn [26].

To determine the superiority of the proposed CNN method, traditional facial expression recognition techniques are compared. This comparative analysis compares the CNN model's performance to that of extant methods and examines the differences in accuracy and efficiency. By conducting this comparison, we can determine the advancements and benefits of the CNN-based approach, emphasizing its accuracy in facial expression recognition.

3.5 Design & Code Implementation

# Import the necessary libraries for building and training a CNN model for image classification using Keras and TensorFlow including utilities for evaluation, data preprocessing, and cross-validation:

```
import numpy as np

import matplotlib.pyplot as plt

from sklearn.metrics import classification_report, confusion_matrix

from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense, Flatten, Conv2D, MaxPooling2D
from tensorflow.keras.losses import sparse_categorical_crossentropy
from tensorflow.keras.optimizers import Adam
from sklearn.model_selection import StratifiedKFold
from tensorflow.keras.preprocessing.image import ImageDataGenerator
# Loading the data that would be used for training and testing:
train_data_dir = '/Users/engrk/Desktop/ferCNN/train'
test_data_dir = '/Users/engrk/Desktop/ferCNN/test'
# Define the emotion class names:
emotion_names = ['Angry', 'Disgust', 'Fear', 'Happy', 'Sad', 'Surprise', 'Neutral']
# Designate score containers for each fold:
accuracy_per_fold = []
loss_per_fold = []
y_true_total = []
y_pred_total = []
# Create an ImageDataGenerator for data augmentation:
data_generator = ImageDataGenerator(
    rescale=1. / 255,
    rotation_range=15,
    width_shift_range=0.1,
    height_shift_range=0.1,
    shear_range=0.1,
    zoom_range=0.1,
    horizontal_flip=True
)
```

```
# Preprocessing using the ImageDataGenerator:
train_data = data_generator.flow_from_directory(
    train_data_dir,
    target_size=(48, 48),
    color_mode='grayscale',
    class_mode='sparse'
)
test_data = data_generator.flow_from_directory(
    test_data_dir,
    target_size=(48, 48),
    color_mode='grayscale',
    class_mode='sparse'
)
# Get the data and labels from the train_data and test_data generators:
train_data_array, train_labels_array = train_data.next()
test_data_array, test_labels_array = test_data.next()
# Concatenate the train and test data arrays along with their labels:
merged_data_array = np.concatenate((train_data_array, test_data_array))
merged_labels_array = np.concatenate((train_labels_array, test_labels_array))
# Carrying out K-fold Cross Validation with merged data:
skf = StratifiedKFold(n_splits=5, shuffle=True, random_state=3)
fold_no = 1
for train_idx, test_idx in skf.split(merged_data_array, merged_labels_array):
    # Define the model architecture
    model = Sequential()
    # Convolutional Layers
   model.add(Conv2D(32,kernel_size=(3,3),activation='relu',input_shape=(48, 48,1)))
    model.add(MaxPooling2D(pool_size=(2, 2)))
    model.add(Conv2D(64, kernel_size=(3, 3), activation='relu'))
    model.add(MaxPooling2D(pool_size=(2, 2)))
    model.add(Conv2D(128, kernel_size=(3, 3), activation='relu'))
    model.add(MaxPooling2D(pool_size=(2, 2)))
    model.add(Dropout(0.25))
    # Flatten layer
    model.add(Flatten())
    # Fully Connected (Dense) layers
    model.add(Dense(256, activation='relu'))
    model.add(Dropout(0.5))
    model.add(Dense(7, activation='softmax'))
    # Compile the model
            model.compile(loss=sparse_categorical_crossentropy,   optimizer=Adam(),
metrics=['accuracy'])
    print(f'Training for fold {fold_no} ...')
 # Get the training subset based on indices
    train_data_subset = merged_data_array[train_idx]
# Get the validation subset based on indices
   val_data = merged_data_array[test_idx]
# Get the training labels subset based on indices
 train_labels_subset = merged_labels_array[train_idx]
# Get the validation labels subset based on indices
```

```
   val_labels = merged_labels_array[test_idx]
    # Training the model with data
    history = model.fit(
        train_data_subset,
        train_labels_subset,
        epochs=50
    )
    #Evaluating the trained model
    values = model.evaluate(test_data_array, test_labels_array)
    print(f'Value for fold {fold_no}: {model.metrics_names[0]} of {values[0]:.2f};'
          f' {model.metrics_names[1]} of {values[1] * 100:.2f}%')
    accuracy_per_fold.append(values[1] * 100)
    loss_per_fold.append(values[0])
    fold_no += 1
# Test Prediction result
    y_pred = np.argmax(model.predict(test_data_array))
    y_true = test_labels_array
    # Total labels for all folds
    y_true_total.extend(y_true)
    y_pred_total.extend(y_pred)
    # Save the model architecture
    model_json = model.to_json()
    with open("emotion_model.json", "w") as json_file:
        json_file.write(model_json)
    # Storing the best weight from the model
    model.save_weights('emotion_model.h5')
# Display all average
print(Value per fold')
for i in range(len(accuracy_per_fold)):
        print(f'> Fold {i + 1} - Loss: {loss_per_fold[i]:.2f} - Accuracy:
{accuracy_per_fold[i]:.2f}%')
print('All folds average values:')
print(f'> Accuracy: {np.mean(accuracy_per_fold):.2f} (+-
{np.std(accuracy_per_fold):.2f})')
print(f'> Loss: {np.mean(loss_per_fold):.2f}')
# Map the class labels to their emotion names
y_true_total_emotions = [emotion_names[int(label)] for label in y_true_total]
y_pred_total_emotions = [emotion_names[int(label)] for label in y_pred_total]
# Computation of Performance Metrics
print('Classification Report:')
print(classification_report(y_true_total_emotions, y_pred_total_emotions))
# Confusion matrix process
plt.figure()
plt.imshow(confusion_matrix(y_true_total_emotions, y_pred_total_emotions,
labels=emotion_names))
plt.title('Confusion Matrix')
plt.xlabel('Predicted Label')
plt.ylabel('True Label')
plt.colorbar()
plt.xticks(np.arange(len(emotion_names)), emotion_names)

plt.yticks(np.arange(len(emotion_names)), emotion_names)

plt.show()
```

Chapter 4

RESULTS AND DISCUSSION

4.1 Evaluation Metrics

I used a 5-fold cross validation technique to evaluate my CNN model. In each fold, segmented samples were partitioned using 80:20 ratio, 80% was used for training while 20% for testing. However, 20% of the segmented train data was used to validate the model during training so as to avoid overfitting the model. The evaluation and performance metrics are further discussed below.

4.1.1 Confusion Matrix

The effectiveness of a classification model on a set of test data when the true values are known can be depicted using a table called a confusion matrix. The rows of the matrix indicate situations that were correctly anticipated, whereas the columns represent those that were incorrectly forecasted. The "confusion matrix" gets its name from the fact that it reveals how frequently the network mislabels one class as another. The introductory layout of a confusion matrix is as follows:

- The matrix is square, with as many rows as there are classes in the model and as many columns as there are rows.
- Predicted classes are represented in row while the columns represent the factual class.
- The diagonal elements describe the number of points for which the predicted marker is equal to the true marker, while the off- slant elements are those that are mislabeled by the classifier.

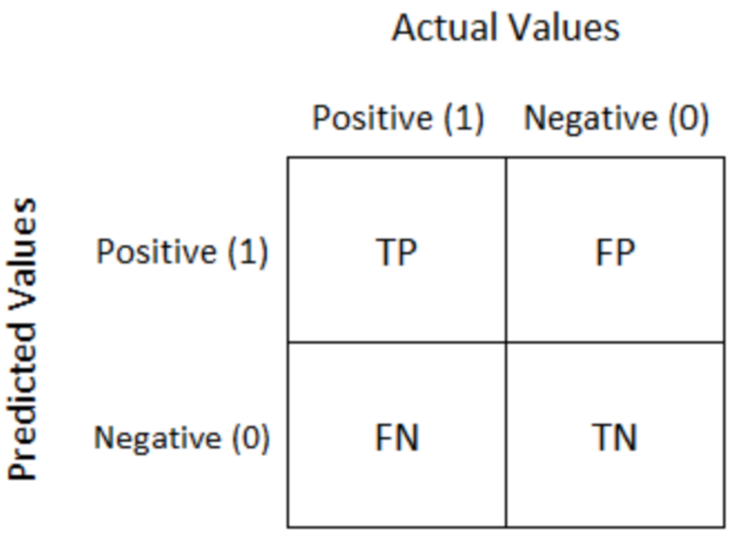


Figure 5: Understanding confusion matrix [27].

4.1.2 Precision, Recall, F1-Score and Support

- **Precision:** represents the classifier's success in making correct predictions of the positive class. It represents the ratio of accurate predictions to total accurate predictions. If the classifier has a high precision, it is less likely to produce false positive predictions.
- **Recall:** is a metric for how well the classifier can identify true positives. How often correct forecasts are made relative to the total number of verified successes. When a classifier has a high recall, it doesn't overlook many true positives.
- **F1-Score:** is optimally balanced between specificity and memorability. One number that maximizes both accuracy and memorability. The classifier's precision and recall are well-balanced if the F1-score is high.
- **Support:** refers to the number of compliances of the authentic response that lie in that specific class. It’s harnessed to offer an idea of the number of compliances that are hooked up with each class.

4.1.3 Performance Metrics

In the table below, TP, FP, FN, TN implies True Positive, False Positive, False Negative and True Negative respectively. This performance metrics is the optimum method for analyzing the dataset used when their values tend closer to 1.

Table 1: Performance metrics

Performance metrics	**Formula**
Precision	TP / (TP + FP)
Recall	TP / (TP + FN)
F1-Score	2 * (Precision * Recall) / (Precision + Recall)

In terms of the FER experiments using the custom CNN, the TP, FP, FN and TN are explained as follows:

- **True Positive (TP):** if the model correctly predicts an image as "Happy" and the ground truth label for that image is also "Happy," it would be considered a true positive.
- **False Positive (FP):** if the model predicts an image as "Happy," but the ground truth label for that image is "Sad," it would be considered a false positive.
- **False Negative (FN):** if the model predicts an image as "Neutral," but the ground truth label for that image is "Happy," it would be considered a false negative.
- **True Negative (TN):** A true negative outcome would be achieved when the model accurately predicted an image as "Neutral" and the ground truth label for that image was likewise "Neutral".

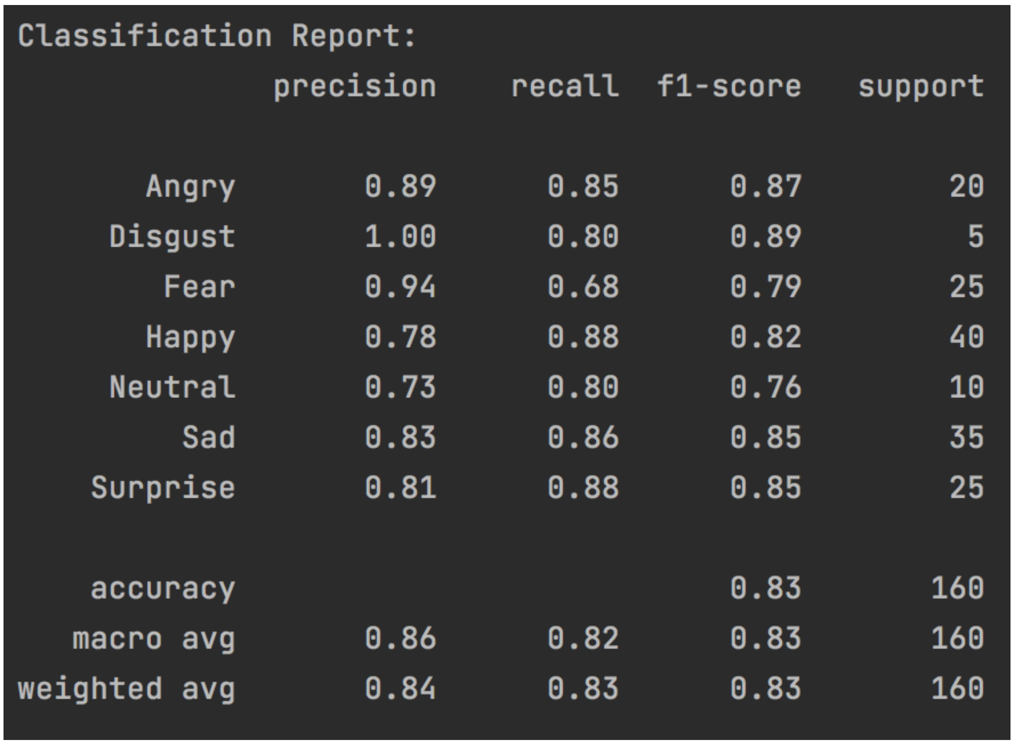
4.2 Analysis of Results

In under 50 epochs of training, the model converged after it attained its peak degree of accuracy for each fold and reached its saturation point, while encountering difficulties in correctly categorizing specific facial expressions like the Disgust class. The total number of successes in each fold was added up at the conclusion of training to arrive at an overall success rate.

Table 2: The average model accuracies for training and testing

State	Accuracy
Training	99%
Testing	83%

To examine the potential CNN model for Facial Expression Recognition, the performance metrics such as Precision, Recall, F1-Score were generated and recorded respectively. The obtained results are presented in the image that follows:



```
Classification Report:
              precision    recall  f1-score   support

       Angry       0.89      0.85      0.87        20
     Disgust       1.00      0.80      0.89         5
        Fear       0.94      0.68      0.79        25
       Happy       0.78      0.88      0.82        40
     Neutral       0.73      0.80      0.76        10
         Sad       0.83      0.86      0.85        35
    Surprise       0.81      0.88      0.85        25

    accuracy                           0.83       160
   macro avg       0.86      0.82      0.83       160
weighted avg       0.84      0.83      0.83       160
```

Figure 6: Performance metrics of the model.

Loss and accuracy values were also computed at various training epochs across all folds during model validation in the table given below.

Table 3: K-folds' loss and accuracy values

Fold number #	**Loss value**	**Accuracy**
1	1.03	78.12%
2	0.61	87.50%
3	0.83	81.25%
4	0.88	81.25%
5	1.07	87.50%

This model has been trained and tested using 5 K-FOLD Cross Validation which was carried out duly following the aforementioned processes. The result achieved from testing the dataset are represented in the confusion matrix below:

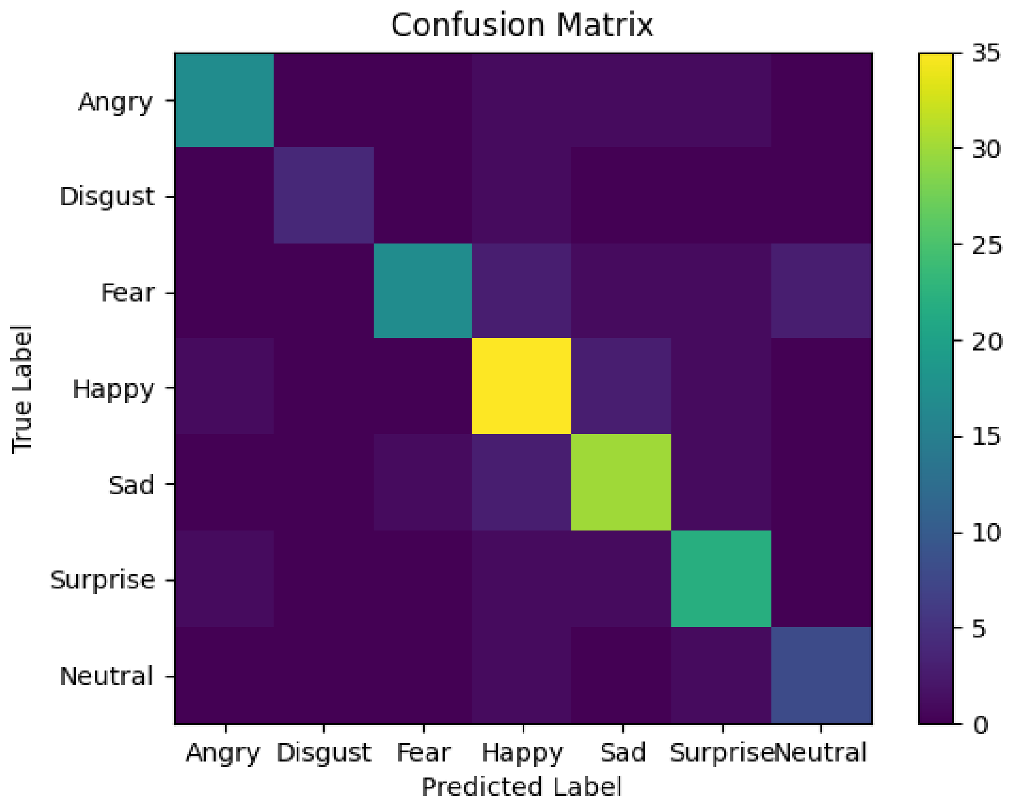


Figure 7: Confusion matrix of the model.

4.3 Discussion of Findings

It's worth noting that the dataset seems to have class imbalance, with varying levels of support (number of samples) for each emotion category. This can affect the evaluation metrics, especially for emotions with fewer samples, as reflected in lower precision, recall, and F1-scores for those categories, especially the Disgust class.

To further enhance the model's performance, adjustments to the training process may be necessary, such as increasing the training data. However, working with an accuracy of 83%, I noticed my model finds it easiest to classify the Happy emotion, but severely struggles with the Disgust emotion class due to low data for its class.

All the results derived were compared with similar studies conducted in recent years.

The following comparison table represents the dataset used alongside the accuracies.

Table 4: Comparison with other studies that used FER2013 dataset without extra training data.

Study	**Model**	**Year**	**Accuracy**
This study	CNN	2023	83.12
Yuan [28]	ResNet18 with tricks	2021	73.70
Khaireddin and Chen [29]	VGGNet	2021	73.28
Pramerdorfer and Kampel [30]	VGG	2016	72.70
Pramerdorfer and Kampel [31]	Res - Net	2016	72.40
Vulpe-Grigorasi and Grigore [32]	CNN optimization	2021	72.16
Fard and Mahoor [33]	Ad-Corre	2022	72.03
Pramerdorfer and Kampel [34]	Inception	2016	71.60
Minaee et al. [35]	DeepEmotion	2019	70.02
Goodfellow et al. [36]	Local learning BOW	2013	67.48

Chapter 5

CONCLUSION AND RECOMMENDATION

5.1 Conclusion

The studies included in this thesis explore the application of convolutional neural networks (CNNs) for facial expression recognition (FER) on the FER2013 dataset. The goal of this research was to build a functional CNN model for FER and evaluate its results using several evaluation metrics, this holds significant importance in various fields, including human-computer interaction, affective computing, and psychological research.

The first several chapters of this thesis offered a comprehensive review of the background ideas, methods, and previous studies on FER and CNNs. We talked about the problems and restrictions of conventional FER methods, and why more advanced methods like deep learning are required. The capacity of the convolutional neural network (CNN) architecture to instinctively acquire distinctive traits off data that is not processed has made it a useful tool for FER.

We utilized this as the basis for developing and deploying a CNN model for FER. We chose the right network design by thinking about things like how deep it would go, what kinds of layers it would have, and what sorts of settings we need to use. The FER2013 dataset annotated with different face expressions was used to train our model. The hyperparameters of the model were fine-tuned and its performance was optimized by extensive experimentation including the use of K-Fold cross validation. By conducting a thorough analysis and comparison with preexisting FER methods, we demonstrated how our own CNN model delivered innovative outcomes with respect to both efficiency and accuracy.

Additionally, we studied how various variables affected our FER model's output, such as image scaling, cropping, and rotation, which improved the model's capacity for fitting to previously unknown data. We also compared how the convergence and overall performance of the CNN changed when we used various optimization algorithms and learning rate schedules. These studies aided in our understanding of the model's inner workings and allowed us to devise methods for its enhancement.

It is vital to note that the FER2013 dataset is not a huge dataset and it has only about 35000 images with class imbalance and these images are not of great quality, so the outcomes may not generalize well to other datasets or to real-world scenarios. However, the results of this research are promising and demonstrate the potential of CNNs for FER by surpassing the performance of previous methods.

Overall, this thesis gives a comprehensive evaluation of the use of CNNs for FER on the FER2013 dataset. We believe that our research makes a substantial contribution to the field of FER by providing a robust and adaptable solution to the difficulties in facial expression analysis and could be used as a springboard for further research on FER taking advantage of advanced architectures and larger datasets.

5.1 Recommendation for Future Work

This master's thesis investigates how Convolutional Neural Networks (CNNs) can be used for Facial Expression Recognition (FER), providing useful information for creating and assessing a high-quality FER system. While our findings have shown remarkable progress, there are still many uncharted territories to be discovered and improved upon. In this part, we present a number of suggestions for future research that could be used to build on and deepen the insights presented here:

1. Balanced Dataset Expansion: The FER2013 set of data utilized for this research is compact and imbalanced, so using a larger and more balanced dataset would allow for more robust and generalizable results.
2. Investigating other architectures: There are several other CNN architectures that have been developed, such as ResNet, VGGNet, and LHC-Net that could be trained on this dataset to see if they would perform better than the architectures used in this thesis.
3. Multimodal Fusion: Incorporating other modalities such as audio or text data could provide additional information which can help enhance the model's capabilities.
4. Testing the model on real-world scenarios: The model trained and tested in this thesis was based on a controlled dataset. To observe how well the model performs in an uncontrolled environment, testing it on everyday information would be fascinating.
5. Combining multiple models: Combining multiple models such as CNN, RNN, LSTM or any other architectures could contribute to the model's improved efficiency.
6. Exploring more advanced evaluation metrics: This thesis used accuracy, precision, recall, F1 score and support as evaluation metrics,

although other advanced evaluation metrics such as ROC, AUC, or Cohen Kappa could be used to evaluate the model.

7. Ethical Considerations: As FER technology develops further, it becomes more crucial to take ethical considerations into account. Privacy worries, potential bias, and fairness issues are just some of the ethical implications of FER applications that need to be considered in future research. Responsible data collection, transparency, and accountability in FER systems are essential for ensuring their ethical and equitable deployment, and researchers should participate actively in these discussions.

REFERENCES

[1] Katona, J., & Kovari, A. (2016). A brain–computer interface project applied in computer engineering. *IEEE Transactions on Education*, *59*(4), 319-326.

[2] Lajevardi, S. M., & Hussain, Z. M. (2009). Feature extraction for facial expression recognition based on hybrid face regions. *Advances in Electrical and Computer Engineering*, *9*(3), 63-67.

[3] Valstar, M. F., Jiang, B., Mehu, M., Pantic, M., & Scherer, K. (2011, March). The first facial expression recognition and analysis challenge. In *2011 IEEE International Conference on Automatic Face & Gesture Recognition (FG)* (pp. 921-926). IEEE.

[4] Uçar, A. (2017, July). Deep Convolutional Neural Networks for facial expression recognition. In *2017 IEEE International Conference on INnovations in Intelligent SysTems and Applications (INISTA)* (pp. 371-375). IEEE.

[5] Ekman, P., & Friesen, W. V. (1971). Constants across cultures in the face and emotion. *Journal of personality and social psychology*, *17*(2), 124.

[6] LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *nature*, *521*(7553), 436-444.

[7] Zhuang, H., Yang, M., Cui, Z., & Zheng, Q. (2017). A method for static hand gesture recognition based on non-negative matrix factorization and compressive sensing. *IAENG International Journal of Computer Science*, *44*(1), 52-59.

[8] Gopal, J. (2020). An Approach for Facial Recognition Using Deep Learning. *Journal of Advanced Research in Dynamical and Control Systems*, *12*(SP3), 137–143.

[9] Balaji, S. (2020). Binary image classifier cnn using tensorflow. *Medium. Aug*, *28*.

[10] Parkhi, O., Vedaldi, A., & Zisserman, A. (2015). Deep face recognition. In *BMVC 2015-Proceedings of the British Machine Vision Conference 2015*. British Machine Vision Association.

[11] Li, S., & Deng, W. (2020). Deep facial expression recognition: A survey. *IEEE transactions on affective computing*, *13*(3), 1195-1215.

[12] Lv, Y., Feng, Z., & Xu, C. (2014). Facial expression recognition via deep learning. In *2014 international conference on smart computing* (pp. 303-308). IEEE.

[13] Pramerdorfer, C., & Kampel, M. (2016). Facial expression recognition using convolutional neural networks: state of the art. *arXiv preprint arXiv:1612.02903*.

[14] Sang, D. V., & Van Dat, N. (2017, October). Facial expression recognition using deep convolutional neural networks. In *2017 9th International Conference on Knowledge and Systems Engineering (KSE)* (pp. 130-135). IEEE.

[15] Qu, X., Peng, C., Wei, T., Du, P., & Chen, C. (2018, December). A novel face attribute segmentation algorithm. In *2018 11th International Symposium on Computational Intelligence and Design (ISCID)* (Vol. 1, pp. 132-135). IEEE.

[16] Al-Omair, O. M. (2022). *Incorporating Emotion Recognition in Co-Adaptive Systems* (Doctoral dissertation, Florida Atlantic University).

[17] Verma, A., Malla, D., Choudhary, A. K., & Arora, V. (2019, February). A detailed study of azure platform & its cognitive services. In *2019 International conference on machine learning, big data, cloud and parallel computing (COMITCon)*(pp. 129-134). IEEE.

[18] Kotsia, I., Buciu, I., & Pitas, I. (2008). An analysis of facial expression recognition under partial facial image occlusion. *Image and Vision Computing*, *26*(7), 1052-1067.

[19] Bodapati, J. D., Naik, D. B., Suvarna, B., & Naralasetti, V. (2022). A deep learning framework with cross pooled soft attention for facial expression recognition. *Journal of The Institution of Engineers (India): Series B*, *103*(5), 1395-1405.

[20] Minaee, S., Minaei, M., & Abdolrashidi, A. (2021). Deep-emotion: Facial expression recognition using attentional convolutional network. *Sensors*, *21*(9), 3046.

[21] Li, S., & Deng, W. (2020). Deep facial expression recognition: A survey. *IEEE transactions on affective computing*, *13*(3), 1195-1215.

[22] Nayef Al-Dabagh, M. Z., Alhabib, M. H., & Al-Mukhtar, F. H. (2018). Face recognition system based on kernel discriminant analysis, k-nearest neighbor and support vector machine. *International Journal of Research and Engineering*, *5*(3), 335-338.

[23] Yamashita, R., Nishio, M., Do, R. K. G., & Togashi, K. (2018). Convolutional neural networks: an overview and application in radiology. *Insights into imaging*, *9*, 611-629.

[24] Georgakis, G., Mousavian, A., Berg, A. C., & Kosecka, J. (2017). Synthesizing training data for object detection in indoor scenes. *arXiv preprint arXiv:1702.07836*.

[25] Sambare, M. (2020). *Fer-2013*. Kaggle. https://www.kaggle.com/datasets/msambare/fer2013

[26] Wasnik, A. (2020, November 12). K-fold cross-validation in python using SKLearn. http://www.askpython.com/python/ examples/k-fold-cross-validation.

[27] Narkhede, S. (2021). *Understanding confusion matrix*. Medium. https://towardsdatascience.com/understanding-confusion- matrix-a9ad42dcfd62

[28] Yuan, Xiaojian. (2021). *Fer2013 Recognition - ResNet18 With Tricks*. https://paperswithcode.com/paper/fer2013-recognition-resnet18-with-tricks.

[29] Khaireddin, Y., & Chen, Z. (2021). Facial emotion recognition: State of the art performance on FER2013. *arXiv preprint arXiv:2105.03588*.

[30] Pramerdorfer, C., & Kampel, M. (2016). Facial expression recognition using convolutional neural networks: state of the art. *arXiv preprint arXiv:1612.02903*.

[31] Pramerdorfer, C., & Kampel, M. (2016). Facial expression recognition using convolutional neural networks: state of the art. *arXiv preprint arXiv:1612.02903*.

[32] Vulpe-Grigoraşi, A., & Grigore, O. (2021, March). Convolutional neural network hyperparameters optimization for facial emotion recognition. In *2021 12th International Symposium on Advanced Topics in Electrical Engineering (ATEE)* (pp. 1-5). IEEE.

[33] Fard, A. P., & Mahoor, M. H. (2022). Ad-corre: Adaptive correlation-based loss for facial expression recognition in the wild. *IEEE Access*, *10*, 26756-26768.

[34] Pramerdorfer, C., & Kampel, M. (2016). Facial expression recognition using convolutional neural networks: state of the art. *arXiv preprint arXiv:1612.02903*.

[35] Minaee, S., Minaei, M., & Abdolrashidi, A. (2021). Deep-emotion: Facial expression recognition using attentional convolutional network. *Sensors*, *21*(9), 3046.

[36] Goodfellow, I. J., Erhan, D., Carrier, P. L., Courville, A., Mirza, M., Hamner, B., ... & Bengio, Y. (2013). Challenges in representation learning: A report on three machine learning contests. In *Neural Information Processing: 20th International Conference, ICONIP 2013, Daegu, Korea, November 3-7, 2013. Proceedings, Part III 20* (pp. 117-124). Springer Berlin heidelberg.