Emotion Recognition from Facial Expression Using Convolutional Neural Network

A Thesis Presented

By

Behaylu Yalew Tesfaye

To

The Faculty of Informatics

Of

St. Mary's University

In Partial Fulfillment of the Requirements

For the Degree of Master of Science

In

Computer Science

January, 2024

ACCEPTANCE

Emotion Recognition from Facial Expression Using Convolutional Neural Network

By

Behaylu Yalew Tessfaye

Accepted by the Faculty of Informatics, St. Mary's University, in partial fulfillment of the requirements for the degree of Master of Science in Computer Science

Thesis Examination Committee:

Internal Examiner

External Examiner

Dean, Faculty of Informatics

January 2024

DECLARATION

I, the undersigned, declare that this thesis work is my original work, has not been presented for a degree in this or any other universities, and all sources of materials used for the thesis work have been duly acknowledged.

Behaylu Yalew

____

Signature

Addis Ababa

Ethiopia

This thesis has been submitted for examination with my approval as advisor.

Dr. Million Meshesha

Advisor

Million Jan 24, 2024

Signature

Addis Ababa

Ethiopia

January 2024

Acknowledgements

First and foremost, I want to convey my heartfelt and sincere appreciation to my advisor, Dr. Million Meshesha, for his unwavering support throughout my M.Sc. thesis. I am grateful for his patience, motivation, enthusiasm, and vast knowledge, which played a crucial role in guiding me through every aspect of the thesis. Beyond my advisor, I extend my gratitude to everyone who has supported me over the years in any way possible. I also want to acknowledge the invaluable encouragement and support from my dear family, especially my mother, without whom nothing would be deemed impossible.

Table of Contents

List of Acronyms

AI	Artificial Intelligence
API	Application Programming Interface.
ANN	Artificial Neural Network
CSV	Comma Separated Value
CPU	Central Processing Unit
DCNN	Deep Convolutional Neural Networks
DL	Deep Learning
DNN	Deep Neural Network
FACS	Facial Action Coding System
FC	Fully Connected
FER	Face Emotion Recognition
GUI	Graphical User Interface
GPU	Graphics Processing Unit
ILSVRC	ImageNet Large Scale Visual Recognition Challenge
MB	Model Size
ML	Machine Learning
NLP	natural language processing
NN	Neural Network
OS	Operating system
RAM	Random access memory
RGB	Red, Green, and Blue
RNN	Normal recurrent network
ResNet	Resnet-50 Model architecture The Residual Neural Network
SVD	Singular Value Decomposition
SVM	Support Vector Machine

List of Figures

List of Tables

Abstract

Facial expressions are a fundamental component of human communication. Recognizing emotions conveyed through facial expressions helps us understand others' feelings, intentions, and social cues, facilitating effective interaction and empathy. This paper uses the FER 2013 dataset to provide an extensive analysis of facial emotion recognition. This study's primary objective was to select a suitable model for face emotion detection using transfer learning techniques. The evaluation process focuses on assessing the accuracy of the models employed. Specifically, to gauge whether interpolation yields improved outcomes, the researchers plan to conduct an experimental analysis of the interpolation technique's effectiveness in upscaling lower-resolution images. By systematically analyzing the impact of interpolation on image quality and model performance, the study aims to provide empirical evidence regarding the efficacy of this technique in enhancing the accuracy of the models employed in image processing tasks specaly for face emotion detection. By systematically analyzing the impact of interpolation on both image quality and model performance, this study seeks to offer empirical evidence concerning the effectiveness of this technique in improving the accuracy of models utilized in image processing tasks, particularly for facial emotion detection.

In order to classify seven distinct emotions this study tested with three alternative convolutional neural network architectures: VGG16, Resnet50, and Inception V3 the accuracy measurement like precision, recall and f-1 score metrics were used to illustrate the model's performance and results using interpolation to a 48x48 size, this study could obtain a maximum of 23 percent recall for all models examined. This study offers valuable insights into the efficacy of various pre-trained CNN architectures and interpolation methods in the domain of facial emotion detection. By assessing these models, it not only informs the selection of suitable architectures and interpolation sizes for emotion detection tasks but also serves as a catalyst for further research in this field. The findings not only guide practitioners in choosing optimal models for their applications but also inspire additional investigations aimed at refining and advancing the techniques used in facial emotion detection.

Keywords: Emotion Recognition; Facial Expression; Digital Image Processing; Convolutional Neural Network

Chapter 1

Introduction

1.1 Background

Every living creature relies greatly on communication and interaction in its daily existence. Emotion is a major component that enhances human communication. Body language and facial expressions are important forms of nonverbal communication in interpersonal relationships. Consequently, it takes comparatively little knowledge of one's own sentiments to establish connections. Understanding emotions requires facial expression detection since it can mimic human coding abilities. It may be possible to determine the emotions a user is experiencing at any one time thanks to its application in fields including computer vision, health, and image recognition. It now plays a big part in the contemporary world as well [1].

Nowadays human face and its characteristics have been one of the main issues in computer vision. It is one of the important components of communication where humans expires their emotions through facial expressions. Humans can communicate their emotions through speech or body language. The ability to focus on the sensitivity and emotional content of people's mental states, behaviors, intentions, and personalities is made simple by the structure of facial expressions [1]. The human behavior model also sheds light on automatic facial expression recognition systems. Facial emotion recognition is a field of study that focuses on methods and techniques for identifying and recognizing emotions from facial expressions. It is made easier by advancements in machine learning and artificial intelligence technologies. Additionally, expressions are predicted to be the next computer-mediated communication medium [1]. There are several areas where emotional recognition from facial expression has been applied, such as human-machine interaction as used in security surveillance, social service marketing, and computer games [1]. In behavioral science, emotion recognition from facial expressions is used to determine social facts like origin, gender, and age. It is also used in medicine to monitor pain, sadness, anxiety, and the treatment of mental retardation. Even though most facial expressions are easy for humans to perceive, reliable expression recognition by machines is still challenging

[2], particularly when input data is subject to a variety of conditions such as head pose, environmental disorder, and lighting, which are all potential sources of variation in the subject's face. Despite its capacity to learn features, deep learning still has issues when used to identify emotions from facial expressions. Although there are facial expression databases, they are not enough to train the commonly a deep neural network that was acknowledged for having the best potential accuracy rate when used for object identification tasks. Additionally, there is a great deal of inter-subject variability due to the many biographical details that are available, such as background, ages, gender, race, and degree of expression [2]. Since emotions can change depending on the surroundings, appearance, culture, and face reaction, emotion detection is a difficult process. However, surveys on emotion recognition are very helpful in examining facial emotion recognition [3].

By harnessing the benefits of deep learning methods, it is possible to develop a system for identifying emotions. Critical factors in enhancing the precision of an emotion detection model include the methodology employed for model creation, the scale of the training dataset, the strategies applied for parameter tuning, and similar considerations. [4]. For a decent accuracy model, hyper parameter tuning is an essential phase in the deep learning process. A model can truly be successful in reaching the necessary accuracy if it chooses its hyper parameter optimization carefully. By investigating, testing, and training the model using hyper parameter tweaking techniques like Bayesian Optimization, a respectable facial emotion recognition model may be developed. This also shows a significant breakthrough in the science of deep learning. The type of task and the properties of the data determine which deep learning (DL) architecture is best. Because Convolutional Neural Networks (CNNs) are so good at capturing spatial hierarchies of features, they are frequently used for image-related tasks. Because they can handle sequential data, recurrent neural networks (RNNs) and their variations, like long short-term memory networks (LSTMs), are widely used in time series analysis and natural language processing applications. Artificial Neural Networks (ANNs) are a flexible foundational architecture that can be used for a wide range of tasks, including regression and classification. The selection of a specific DL model is influenced by the unique requirements of the problem at hand, as each architecture is tailored to excel in particular types of data and tasks.

1.2 Motivation of the study

Humans frequently interact emotionally, which is a psychological phenomenon. Effective human communication, engagement, and decision-making depend on accurate emotion perception. The study of emotion detection systems has evolved into a common project in both academia and business because of the advancements in big data technologies and artificial intelligence. Textual data in emotional communication is frequently used to infer emotional states like pleasure, sadness, and rage since it is the simplest and most direct form of transmission. The fast growth of social media, mobile networks, and smart terminals in recent decades has also made online social networks and online marketing an unheard-of worldwide phenomenon.

With the use of tools like face recognition and computer vision, AI facial emotion detection is able to rapidly interpret expressions on the face to ascertain the current mood. It can help distinguish between happy and unhappy individuals. AI and ML are extensively used across numerous industries. They have been utilized in data mining to find instances of insurance fraud [4]. Data mining using clustering was utilized to find trends in stock market data. ML can be utilized to produce FER solutions that are dependable, affordable, and need little computation time [5]. However, as the lighting and the backdrop are always changing, it might be difficult to capture people's features clearly when they are moving.

In addition, individuals frequently change their appearances by tanning, growing facial hair, etc. Additionally, some people are less expressive than others, which makes it more difficult to detect any emotional cues. Convolutional Neural Networks (CNNs) try more than other learning methods and have been at the forefront of numerous innovations in a wide range of applications, including object identification and large-scale picture classification. CNN has also had a lot of success with face recognition. However, researchers' interest is in understanding how it is important to the senses, what it means for deep neural networks to learn facial expressions, and how this can improve performance [6]. It is therefore the aim of this study to apply deep learning algorithms for the recognition of face emotion.

1.3 Statement of the Problem

Understanding human behavior is crucial for improved human-computer interaction, and the computer may learn a lot from a person's expressions. Since daily interactions between humans and computers are common, robots must be able to recognize human facial expressions in order to study and comprehend human behavior. It is crucial to create a reliable face expression recognition system. For instance, a number of scholars have proposed a novel approach that makes use of machine learning and deep learning. According to Kumar's (2019) [7] proposal, facial muscle movements can be used to identify the mood on a person's face [7]. An end-to-end deep learning framework based on attentional convolutional networks has been suggested by Shervin Minaee and Amirali Abdolrashidi (2019) [8] to classify the underlying emotion in the human face [8]. A unique depth camera-based method for recognizing facial emotions has been employed by Uddin et al. (2017) [9]. A series of visual stimuli representing three tiers of emotional expression intensity, ranging from low to high, was created by Wingenbach et al. (2016) [10].

Different developers build different architectures for machine learning and face recognition, such as LeNet, AlexNet, GoogLeNet, VGGNet, ResNet50, MobileNets, etc., each architectures offering unique enhancements. They have pushed boundaries in image recognition and classification tasks, with widespread applications in various domains, but selecting which has achieved better accuracy in image recognition is another obstacle. The accuracy of CNN's facial expression identification is not always guaranteed because facial expression detection based on the face is influenced by lighting, position, and cosmetics. Also, different algorithms based on local features and deep learning extraction techniques are developed by different researchers for DL data. Most of the work done on Machine learning data has the following drawbacks:

- The recognition rate of facial expression recognition is not satisfactory or sufficient.
- Most researches are try to train a modal to categories facial expression.
- It needs high performance processor to train a model.

Therefore, the aim of this study is to investigate and compare the GoogleNet, VGG16 and ResNet50 architectures for face emotion recognition to see if they can solve the drawbacks observed in previous works.

1.4 Research questions

To explore the above-stated problem, the following research questions are formulated to investigate and find solution:

- Which deep learning method is more effective at identifying the emotions on a face?
- How well does the suggested model recognize facial expressions?

1.5 Objective of the study

1.5.1 General Objective

The general objective of this thesis is to select suitable model for emotion recognition from facial expression using deep learning convolutional neural network architecture.

1.5.2 Specific Objectives

The study's overall goal is attained by completing the following particular objectives:

- To review literature and select suitable methods and techniques for deep learning conventional neural network architectures.
- To investigate and understand how to implement deep learning FER.
- To collect and prepare data set for experimentation
- To select suitable deep-learning architecture and model.
- To conduct experiment using the pre-trained models such as GoogLeNet, VGG16 and ResNet50 of facial emotion detection.
- To evaluate the performance of the proposed model for facial emotion recognition

1.6 Methodology of the study

This section describes the proposed system's methods for detecting lumpy skin diseases. A topic's identification, analysis, processing, and selection of material are all done using specific methods or approaches known as methodology [11]. The methodology of a study outlines the methodologies and step-by-step processes utilized to comprehensive examination and accomplish the study's goal.

1.6.1 Research design

A research design refers to the organization of conditions for gathering and analyzing data with the goal of addressing a research problem. In this study, the objective is to conduct a comparison of models for recognizing facial emotions. To this end, the study follows experimental research. Experimental research is the process of implementing an algorithm and empirically evaluating the efficiency and effectiveness of those algorithms through solving real-world problems. Effective facial expression recognition is a critical issue to tackle in many fields, including psychology, human-computer interaction, and marketing research. In order to classify seven different emotions: surprise, sadness, anger, pleasure, sadness, fear, neutral, and disgust, this study experimented on two alternative convolutional neural network architectures on GoogleNet, ResNet50 and VGG16. In order to conduct an extensive experimentation, in this paper perform image data collection and preparation, implementation and evaluation undertaking experimental research involves dataset preparation, implementation, and performance evaluation [12].

1.6.2 Data collection and preparation

For this study, we obtained the necessary database using a secondary data collection strategy. This study makes use of the FER-2013 database, which contains lots of labelled photos. A collection of 35887 grayscale, 48x48-pixel face photos representing the seven emotions of anger, contempt, fear, happiness, neutrality, sadness, and surprise make up this collection [13]. This dataset includes both face-focused and non-focused face photos, which aids in the training of our model. This dataset was mostly compiled using photos from Google searches for various emotions and their varied synonyms [13]. Pre-processing of the acquired data must be done carefully to prevent problems like memory crashes, wide image size and color variations, and other similar problems. To solve problems, design acknowledgment and image pre-processing can be used in many different contexts.

1.6.3 Implementation tools

Developing AI and ML applications is difficult and takes time. Still, there are a lot of Python-compatible libraries. The main justification for developers' preference for it over other languages is this. Libraries are prewritten code segments that can be called and imported into an editor to reuse the function. In this study, the Python programming language is used for the implementation, and we also use the OpenCV, Tensorflow, Kreas, and Numpy packages.

OpenCV packages: The OpenCV (Open Source Computer Vision Library) package serves several purposes. OpenCV is a widely used library in the field of computer vision, and its functionalities are leveraged for various tasks related to face emotion detection. It plays a foundational role in face emotion detection by providing tools for face detection, image preprocessing, region extraction, and visualization. It is often used in conjunction with other libraries to build comprehensive solutions for analysing facial expressions and emotions.

Tensorflow: The Google Brain team created the open-source machine learning framework Tensorflow. It offers an extensive collection of tools, including face emotion detection tools, for developing and implementing machine learning models. TensorFlow plays a crucial role in face emotion detection by providing the necessary infrastructure for building, training, and deploying deep learning models. It offers a wide range of features that support the entire machine learning pipeline, from model development to deployment.

Keras package: face emotion detection lies in providing a high-level and user-friendly interface for building, training, and deploying neural network models. Specifically, in the context of face emotion detection, Keras simplifies the development and deployment of neural network models for face emotion detection by providing a user-friendly interface and integration with powerful deep learning frameworks like TensorFlow. Its modularity, flexibility, and support for pre-trained models make it a valuable tool for researchers and developers working on emotion recognition tasks.

Numpy packages: A basic package for scientific computing in Python is called NumPy while it may not be directly involved in the specific tasks of face emotion detection, NumPy serves as a foundational library that provides essential functionalities for numerical operations, array manipulation, and data handling. While NumPy itself may not be the primary tool for implementing face emotion detection algorithms, it plays a crucial role as a foundational library that complements other specialised libraries and frameworks used in the overall workflow. Its array manipulation and numerical computing capabilities contribute to efficient data handling and processing in the context of face emotion detection.

1.6.4 Evaluation Methods

In this thesis, we evaluated the performance of the classifier using a range of assessment objects. We prepared a significant number of confusion matrices based on the classification performance of the classifier using various facial expressions. We compute precision, recall, F-score, and accuracy to evaluate the classification performance of models in accordance with the confusion matrix. Additionally, we gathered the test timings for classifiers in various face states. The confusion matrix is a display tool for assessing the accuracy of the classification model in the field of object detection. More specifically, each row in the matrix shows the sample's actual result, and each column shows the model's predicted result. Precision is used to express the fraction of positive cases that are predicted to be positive. The term recall refers to the ability to recognize each real object in a dataset. A useful statistic in cases of data imbalance is the F-score. For recall and accuracy, the F-score is the harmonic average. The accuracy formula is computed as follows: total number of correct predictions divided by total number of instances.

1.7 Significance of the study

One of the computer vision technologies that can solve issues in several fields is emotion recognition. In contrast, emotion recognition identifies more explicit and precise emotional states, referring to a variety of mental states, including happiness, rage, sadness, and fear. Emotions are a crucial aspect of human nature and have been extensively researched in both this area and psychology. Depending on the underlying emotion, such as fear, anger, joy, pleasure, sadness, disgust, or surprise, emotion recognition attempts to classify into one or more emotion classes.

Systems for recognizing emotions are currently being used in a variety of areas of daily life, including as commercial communications, public opinion monitoring, and mental health monitoring. Emotion recognition can enhance customer preference-based commercial tactics. And in Understanding people's reactions to a specific circumstance through social networks and emotion detection in crisis or catastrophe scenarios is helpful for crisis management and essential decision-making. Based on their statements and online comments, public emotions may be monitored and forecasted during elections [14].

In systems involving human-computer interaction, like dialogue systems, automated question-answering systems, and companion robots, the integration of emotion recognition technology is beneficial for enhancing user experience. Additionally, in an efficient e-learning environment, real-time emotion tracking enhances students' motivation and overall effectiveness. [15]. The essence of emotion suggests the possible uses of emotion recognition across diverse problem domains. Furthermore, this thesis can serve as a valuable reference for researchers engaged in DL, particularly in the field of facial emotion recognition research.

1.8 Scope and Limitation of the study

There are a lot of different ways to investigate facial emotion recognition: Viola-Jones algorithm, knowledge- or rule-based approaches, feature-based or feature-invariant methods, appearance-based techniques, template matching, and convolutional neural network-based methods. However, due to time limitations, this thesis is forced to limit itself to only exploring a tiny portion of the CNN-filled GoogleNet, VGG16, and ResNet50 architectures. Since there are too many distinct emotion modals and architectures, this thesis does not cover all models and architecture; rather, a selection of emotion pre-trained models is the main focus, which is facial emotion. Using a different dataset, this study offers a thorough analysis of facial expression emotion detection effective facial expression recognition is a critical issue to tackle in many fields, including psychology, human-computer interaction, and marketing research. This study tested three alternative convolutional neural network architectures: GoogleNet, ResNet50, and VGG16, with the goal of classifying seven distinct emotions: fear, anger, pleasure, sadness, disgust, neutral, sadness, and surprise.

1.9 Organization Of the rest Thesis

The initial chapter of this thesis offers a concise introduction, presenting details about the assignment's context, goal definition, problem statement, scope of work, and the structure of the thesis report. It also outlines the overall framework incorporated in this study. The remaining four chapters are organized into three subheadings each. The background material is sufficient for the reader to comprehend the motivation for the investigation. A summary of the entire study is given in this chapter.

The second chapter of the study explains emotion recognition, facial expression analysis, and deep learning, which is used to demonstrate the experiment methodology, deep learning, and their types that are used in this thesis, and presents reviews of prior research that is relevant to the study topic with particular reference to the thesis objectives. In order to complete this task and represent the main findings and advice of others, it includes summaries from books, journals, and a collection of relevant works.

Chapter three discusses methodology in the study in detail, which includes literature review collection, dataset collection, classification, design, feature extraction, feature extraction methods, and implantation tools. Dimension reduction methods with their representations used in the study, programming tools used in coding, designing, editing, and algorithms used for compression are also discussed in this chapter.

The study findings and a thorough experiment conducted using the methods described in Chapter 3 are provided in Chapter 4. The final chapter, chapter five, summarizes the findings, makes inferences for readers of the research, and offers suggestions for further investigation.

Chapter 2

Literature Review And Related Works

The second chapter of the study explains review about emotion recognition, deep learning layers and its architecture which will be used to demonstrate the experiment methodology, deep learning, and their types that are used in this thesis, and presents reviews of prior research that is relevant to the study topic with particular reference to the thesis objectives. In order to complete this task and represent the main findings and advice of others, it includes summaries from books, journals, and a collection of relevant works.

2.1 Overview

Emotion recognition, also known as affective computing, is a rapidly expanding area of artificial intelligence that enables computers to interpret and analyze nonverbal cues from people, including their face language, body language and voice tones in order to determine how they are feeling. As a result, computer vision technology combined with visual emotion artificial intelligence assesses how faces appear in pictures and videos to determine a person's emotional state [16], so for this deferent Models are frequently used as input for images in image processing. The name of this multi-layer approach, which uses many operations to extract information from photos and classify them, comes from a linear mathematical action between matrices.

By using image processing and pattern recognition algorithms to search this algorithm for facial expressions, facial expression identification is a method for identifying the emotions of people. This project has applications in entertainment, psychology, and human-computer interaction, among other fields. Convolutional neural networks (CNNs) are used for classification and feature extraction; with deep learning techniques, they have shown especially impressive performance in face emotion recognition in recent years. The fully connected layer with linear, dropout, ReLU, and softmax functions, data augmentation techniques, and the theoretical foundations of convolutional neural networks—specifically, GoogleNet, VGG16, and ResNet50—are all covered in this background section.

2.2 Convolutional Neural Networks

Convolutional Neural Network (CNN) is one of neural network architecture in Deep Learning, it is used to recognize the pattern from structured arrays [17] Nonetheless, CNN architectures have changed over a long period of time. Numerous variations of the core CNN Architecture have been created, resulting in incredible advancements in the rapidly expanding field of deep learning. It belongs to the class of artificial neural networks (ANN) methods, and while filtering, sorting, and classifying data from the source, it makes use of many layer blocks. According to Brownlee (2020) [17], these layers could be convolutional, pooling, and fully connected classification output layers. The number of filters that will be utilized in the learning process to create feature maps is first specified using a convolutional layer. Furthermore, filter attributes are defined, such as the size of the pixels used in each filter. There is a defined kernel form.

A well-liked deep neural network for image processing and recognition, CNN consists of one or more fully linked classification layers after several layers of pooling and convolutional operations. The convolutional layers take the input image and extract local characteristics from it using a sequence of learnable filters. By sampling the output of the convolutional layers, the pooling layers reduce the dimensionality of the feature maps.

2.3 CNN architectures

In recent decades, a number of CNN designs have been developed, and they have excelled in recognition tasks, particularly for ImageNet. According to the ImageNet project, there are more than 1.4 million images in the visual repository that may be used for research on visual recognition (Ima). The following are some of the most notable architectures employed by the top finalists in the ILSVRC (ImageNet Large Scale Visual Recognition Challenge), which attempts to evaluate algorithms for object detection and image classification at large scale.

2.3.1 VGG-16

The VGG-16 CNN architecture was developed by the Visual Geometry Group at the University of Oxford. Its sixteen layers are composed of three fully linked layers, thirteen convolutional layers, and max pooling layers with stride 2. The max pooling layers with stride 2 are arranged after the convolutional layers, which are composed of 3x3 filters with padding 1 and stride 1. The extracted features are classified by the fully linked layers. It has been demonstrated that VGG16 performs well on tasks requiring the recognition of emotions. In a comparison study, VGG16 achieved a top- five error rate of Seven and five tenths percent on the ImageNet classification test, outperforming alternative CNN designs [18].

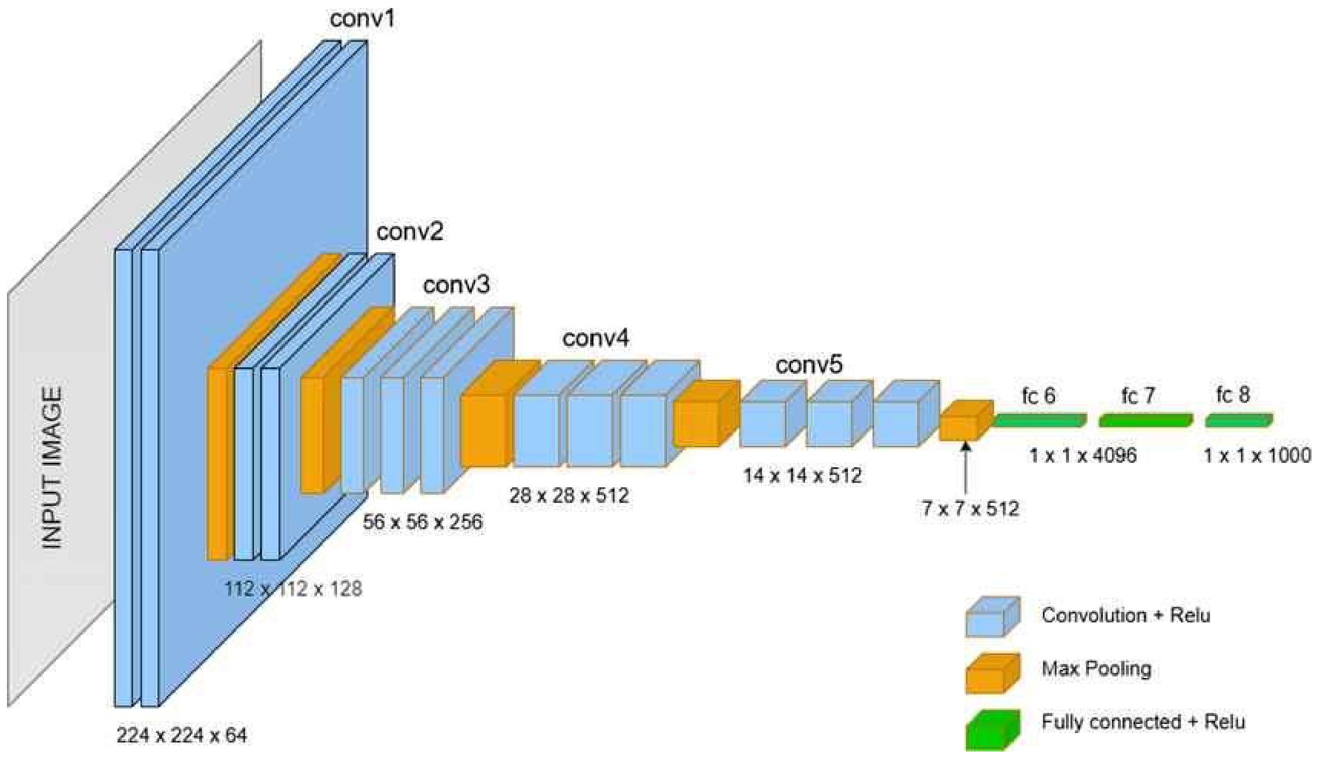


Figure 2.1: VGG16 architecture (by Fan Sun, 2022) [40]

The model's capacity to recognize things in images is complemented by the fact that model weights are freely available and can be applied to models and apps for object recognition. Their top network has an extremely homogeneous design and sixteen convolutional FC layers. Throughout the process, this network uses 2x2 pooling and 3x3 convolutions. In the experiments for this thesis, the VGG-16 model is also employed. The VGGNet's drawbacks include the higher cost of assessment and evaluation, as well as the need for 140 million parameters and a lot of memory.

2.3.2 ResNet50

The convolutional neural network known as ResNet-50, which has 50 layers, was developed by Microsoft Research. According to Dos Santos et al., CNNs, or convolutional neural networks, are an adaptation of the ResNet50 architecture. (2020). There are a total of 50 layers: 16 convolutional bottleneck layers, 1 convolutional layer, and 1 fully connected layer. The goal of the bottleneck layers is to improve the training efficiency of the network by reducing the amount of parameters in it. The ResNet50 architecture uses skip connections to address the potential vanishing gradient problem with deep networks.

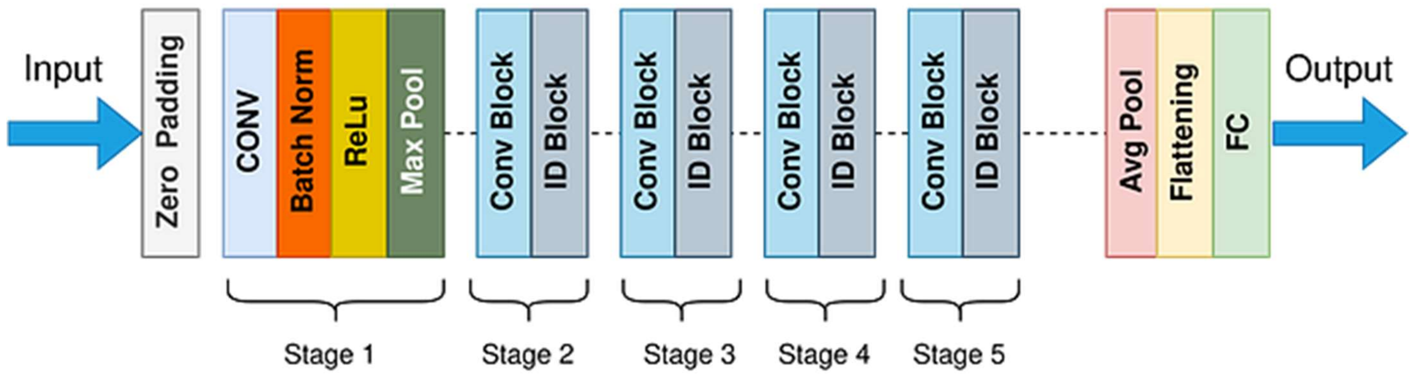


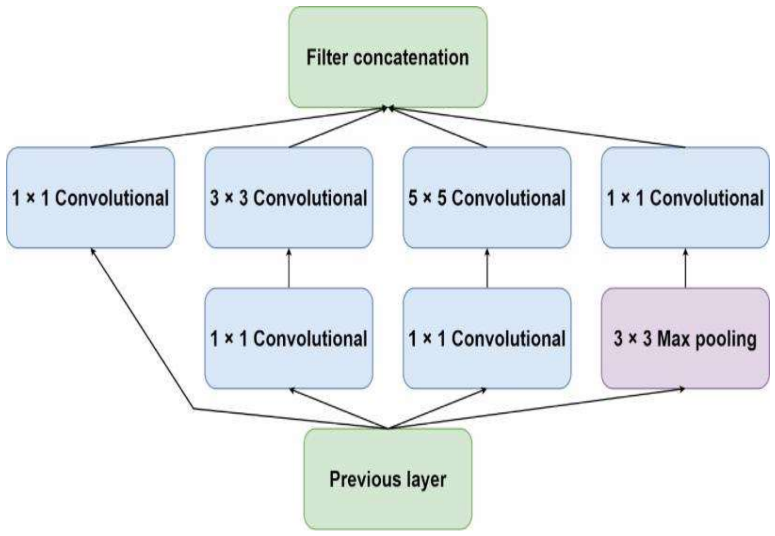
Figure 2.2: Resnet-50 Model architecture (ResNet-50, 2022) [19].

One of the most recent CNNs created for artificial neural networks (ANNs) is the Residual Neural Network (ResNet). The ResNet architecture is thought to be a particularly deep CNN because it has more than one hundred layers [16] ResNet is a learning architecture that is very well suited for emotion recognition, frequently outperforming other well-known convolutional neural networks. By using skip connections, the network can learn residual functions, or functions that convert input data into output data. The final output of the network is then obtained by adding the residual functions to the original input. This technique trains the network to distinguish between input and output data, which simplifies the process of deep network training. [12].

ResNet50 has shown state-of-the-art performance on a range of picture identification tasks. Showed that ResNet50 outperformed VGG16 with a top-5 error rate of 5.25% in the ImageNet classification task, a notable improvement. The previous best result, 5 points 6 percent, belonged to VGG16. Additionally, ResNet50 has proven effective in transfer learning—a process in which a pre-trained model is improved on a smaller dataset for a particular goal.

2.3.3 GoogLeNet

GoogLeNet architecture is a CNN that is twenty-two layers deep [20]. GoogLeNet was created with the idea of an inception module in mind. Its image input size is 224 × 224 × 3 pixels. Parallel processing is present in the inception module, which includes convolutions in sizes of 1x1, 2x2, 5x5, and Max Pooling 3x3 [21]; [22]. GoogLeNet was developed using nine inception modules, as shown in Figure 2.4. Once entering an inception module, data is split into four groups for processing in parallel, and once exiting the module, the groups are combined into a single set. Figure 2.3 Inception Module used in GoogLeNet [23].



The GoogLeNet architecture was first introduced as GoogLeNet (Inception V1), and it was later renamed Inception V2, and most recently Inception V3. Empirical evidence suggests that Inception modules are capable of producing richer representations with fewer parameters, even though they are conceptually convolutional feature extractors. A traditional convolutional layer seeks to learn filters in a three-dimensional space using two spatial dimensions (width and height) and one channel dimension. As a result, one convolution kernel is capable of mapping both spatial and cross-channel correlations simultaneously. Additionally, a pooling layer and an alternate convolutional layer are included in this architecture. With a patch size of 7x7, the first convolutional layer shrinks the image without sacrificing any of its spatial information. In each layer that appears, the input image size will be shrunk by a factor of four. The final architecture has several inception modules layered on top of one another [17].

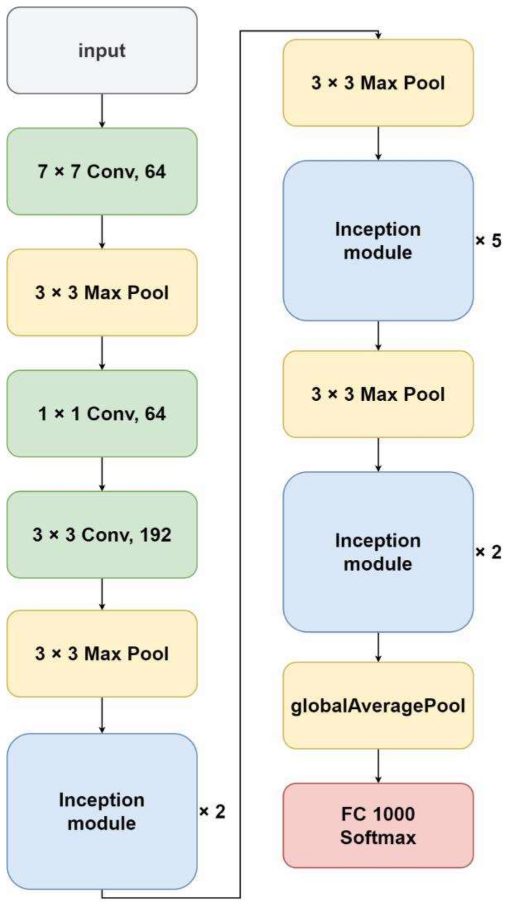


Figure 2.4 GoogLeNet architecture. (Ketwongsa et al., 2022) [23]

This 22-layer CNN architecture is another significant one. Szegedy et al. (2014) [24] introduced it in 2014. GoogleNet's concept is to minimize the dimension by using an inception module made up of kernels with varying granularities, such as 1x1 and 3x3 and 5x5 kernels. In the end, these are combined to create an inception module. The convolutional layers are separated by the introduction of these modules. It obtained a 6.7% top-5% test error [25].

2.4 CNN Layer

The CNN is composed of three different types of layers: convolutional, pooling, and fully-connected. The CNN architecture is produced by layering these layers. In addition to these three layers, there are two more important factors that are described below: the dropout layer and the activation function [26].

2.4.1 Conventional Layer

This layer uses an MxM-sized filter to perform a mathematical operation known as convolution on the input photos. The input image is systematically scanned with the filter applied, and at each position, the. Product between the filter and the local area of the image that the filter covers (MxM) is computed. A feature map that emphasizes the input image's patterns, like corners and edges, is produced by this process. The input picture's more intricate details are revealed by advancing layers of the neural network using the feature map, which acts as a representation of the features that have been extracted. The convolutional layer is essential for capturing hierarchical patterns and features, which lays the groundwork for the CNN's capacity to learn and identify visual data [26].

2.4.2 Pooling Layer

A typical practice following a convolutional layer involves the use of a pooling layer. This layer primarily aims to reduce the size of the convolved feature map, thereby conserving computational resources. This is achieved by eliminating connections between layers and individually modifying each feature map. Various pooling techniques exist, depending on the chosen method. Max Pooling, for instance, heavily relies on the feature map's most significant contribution. Average pooling computes the average of elements within a specified image segment, while Sum Pooling calculates the total of components in the designated section. The Pooling Layer commonly acts as a link connecting the Convolutional Layer and the Fully Connected Layer [26].

2.4.3 Fully Connected Layer

The Fully Connected (FC) layer, which also has weights and biases, connects the neurons between two layers. These layers are usually placed before the output layer and make up the last few tiers of a CNN architecture. This flattens and sends the input image to the FC layer from the layers below. The process of categorizing now begins as the flattened vector passes through a few more FC levels and then undergoes the usual operations on mathematical functions. [26].

2.5 Evaluation Metrics

Performance measurement plays a crucial role in evaluation. Approaches to performance evaluation typically rely on a graphical or numerical depiction. The computation of true positives (TPs), false positives (FPs), false negatives (FNs), and true negatives (TNs) is the basis of the numerical measurements. A confusion matrix is used to illustrate the relationships. The goal of the classification usually determines which numerical measure is best. Based on the values of the confusion matrix, the performance measures include: exact match ratio, f-score, recall, sensitivity, accuracy, and precision for binary, multi-class, multi-labelled, and hierarchical classifiers. The most popular performance metric for assessing classification algorithms is accuracy. A series of collaborative representation models for face recognition and item classification tasks are evaluated using the accuracy measure. Classifier performance is also assessed using classification accuracy. In order to remove the detrimental effect of noisy features on picture classification, collaborative linear coding techniques are evaluated using average precision (AP) and mean AP (mAP).

2.6 Related Works

A review of the literature shows that Deep Convolutional Neural Networks (DCNN) are widely used for image interpretation. A number of articles are specifically looked at with the intention of employing CNNs for face emotion recognition.

Deep neural network (DNN)-based facial emotion detection research has surpassed human accuracy on certain tasks and achieved state-of-the-art performance on many benchmark datasets, according to analyses of the current status of the. This has raised interest in and study in the field, with numerous studies concentrating on enhancing these systems' resilience and accuracy using a hybrid model for face expression recognition. For the purpose of extracting facial features, the DCNN has more convolutional layers and ReLU activation functions for real-time facial feature detection. FER-2013 grayscale photos and data enhancement methods are utilized for training and validation [27]; [24]; [28].

A deep learning approach was proposed by Ankan et al. (2022) [29] to classify human face emotions from infrared thermal pictures. In the absence of visible light, their approach provides a robust framework for the identification of exact expressions. The difficulties that facial expression recognition algorithms encounter in practical settings are covered by Lucie et al. (2022) [30]. Through a crowdsourcing experiment, the impact of different visual distortions on human performance in FER was investigated. Even after they had been modified, the models' performance was found to be more prone to distortions than that of humans when two open-source FER algorithms were compared with human performance. The paper also addresses bias and annotation issues and provides strategies for enhancing existing datasets.

A current limitation in the examination of facial emotion recognition through feature extraction and classification using deep neural networks (DNNs) is the potential susceptibility to overfitting with respect to the training data. Poor generalization performance on fresh data may arise from an inadequately diversified training dataset or overly complex model architecture [31]. The possibility of bias in the model or the dataset, which could result in unfair or erroneous predictions, is another disadvantage. Due to the possibility of biased or incomplete datasets that might not accurately represent the entire range of emotions or facial expressions across diverse cultures and demographics, this is especially important when it comes to facial emotion recognition [32].

The lack of interpretability in the DNN models utilized for facial emotion recognition is another drawback. Understanding the rationale behind the model's predictions can be difficult, diminishing the transparency and reliability of the system [33]. Moreover, the majority of current state-of-the-art DNN models are not designed to work with the 48x48 image size, which is a common image size used in facial emotion detection datasets like the FER 2013 dataset. Due to insufficient interpolation approaches, this constraint may result in subpar performance or extra processing demands for scaling the photos to a suitable size [34].

With this in mind, deep neural networks, or DNNs, have shown a lot of promise in the area of facial expression recognition. Still, there are issues with bias, interpretability, overfitting, and compatibility with non-standard image sizes. Further research and development are required to address the practical constraints of real-world applications and overcome these challenges in order to enhance the accuracy, robustness, and fairness of these systems.

Chapter 3

Design and Methods

3.1 Overview

This chapter details where the data was gathered, how it was prepared, and how it was analysed to create a classification model using the chosen modelling approaches. In this thesis, an experimental research approach is used where a set of variables is kept constant. In contrast, the other set of variables is being measured as the subject of an experiment. Different experiments are carried out using different dataset ratios. Finally, evaluation methods utilized to assess the effectiveness of the created model are provided and debated.

3.2 Architecture

The experimental research method is employed in the thesis. This section outlines the suggested steps and procedures that was applied in this study in order to meet the thesis's goal. It is divided into four primary stages: pre-processing of the data, detection of face, extraction of features, and classification for identifying facial emotions. The research major tasks are shown in Figure 3.1, where the subsequent process flow is noted. This work suggests a feature extraction and classification method based on the VGG16, GoogleNet, and ResNet50 models for face emotion recognition. Seven distinct emotions are labeled on images from the dataset used in this study: happy, sorrow, anger, surprise, fear, disgust, and neutral.

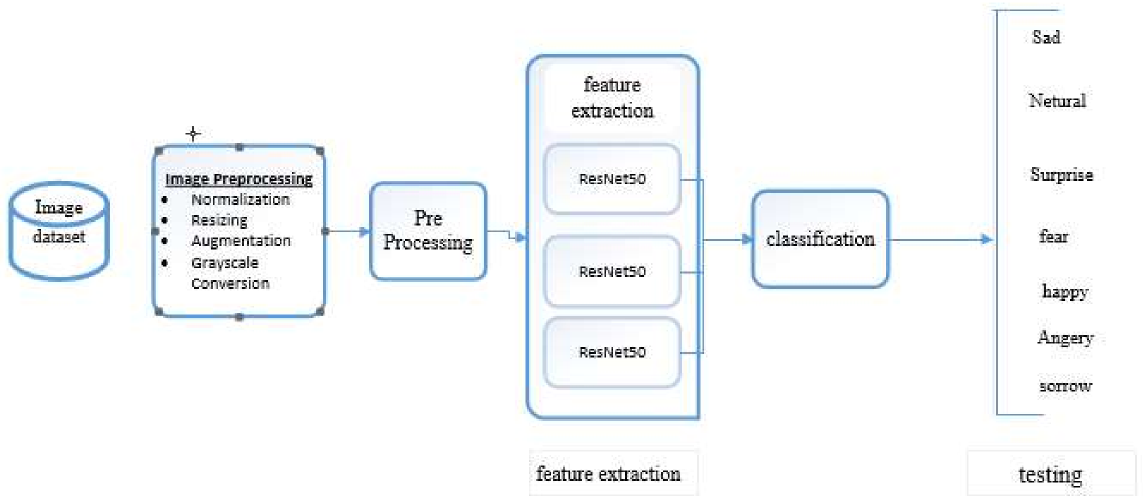


Figure 3.1 Research flow

3.3 Data Collection and Preparation

As machine learning algorithms heavily rely on data, the selection of an appropriate dataset was pivotal in the system development process. Choosing an incorrect dataset could have significantly complicated the project. This research will gather trend datasets from well-organized databases. Since the main objective of this study is not facial recognition, opting for face datasets, for instance, would have needlessly introduced additional complexity to the face detection algorithm.

Numerous datasets focusing on facial expressions exhibit diverse picture qualities, but accessing many high-quality datasets requires permission. Fortunately, obtaining permission is usually straightforward, as most dataset owners only require users to fill out a form as a procedural formality. Because they provided tagged photos with the necessary facial expressions and a backdrop free of noise, the datasets trend VGG16 and ResNet50 model are chosen for this research to be loaded on DCNN after thorough study.

3.4 Image preprocessing

Traditionally, preprocessing was a critical step, involving the transformation of raw data before inputting it into a neural network or deep learning algorithm. However, in the case of CNNs, the statement suggests that explicit preprocessing on the dataset is not as crucial. This is because CNNs have the capability to directly take in the raw pixel values of images and autonomously learn relevant features during training. The architecture of CNNs is designed to automatically extract hierarchical features, eliminating the need for extensive manual preprocessing, thus streamlining the workflow in image-related tasks. Image preprocessing for Deep Convolutional Neural Networks (DCNNs) like VGG16, ResNet50, and GoogLeNet (Inception) involves several common steps. These steps aim to standardise the input data and enhance the model's ability to learn useful features from images.

Here are some common image preprocessing steps: Resize input images to match the expected input size of the neural network architecture. Resize images to the specific dimensions required by each model (e.g., VGG16: 48x48, ResNet50: 224x224, GoogLeNet: 224x224). Normalize pixel values to a common scale to ensure consistent input to the model. Normalize pixel values to be in the range [0, 1] by dividing each pixel value by 255. Subtract the mean pixel value of the dataset to center the data on zero. This is often done to match the training conditions of the original, pre-trained models. Subtract the mean pixel value of the dataset from each pixel in the input images. Augment the training data with variations (rotations, flips, and shifts) to improve the model's generalization ability. Apply random transformations to the training images during training but not during validation or testing.

3.5 Face detection

The initial and crucial step in the processing pipeline was face detection. Despite our images exclusively featuring frontal facial expressions, detecting the face was a prerequisite before proceeding with further analysis. Once the face was successfully detected, identifying the region of interest and extracting relevant features became a more straightforward task. Various face detection algorithms, including Haar-cascades from OpenCV, were experimented with. Eventually, we opted for a face detector based on the histogram of oriented gradients (HoG) from the Dlib library. The combination of HoG descriptors and Support Vector Machines (SVM) proved effective in pinpointing faces within the images. To ensure uniformity, the images were converted to grayscale and resized during the preprocessing stage.

3.6 Pre trained models

Over time, several different picture classification models have been created, for example for the yearly ImageNet Large Scale Visual Recognition Challenge (ILSVRC). If these advancements in designs and training could be put to use, it would greatly benefit the creation of new deep convolutional neural network models. Models have previously been trained on millions of photos in a variety of categories, and they have discovered feature weights that perform very well for particular tasks.

Parts of these pre-trained models can be incorporated into new models because APIs often make them publicly available. Parts will function as a feature extraction procedure or will employ weights for discovered features as the basis for brand-new interpretation issues. Thankfully, Keras is accessible, provides access to pre-trained CNNs, and allows free model weight downloads from their libraries. We may choose any of the cutting-edge models and utilise them for our problem. Appropriate CNNs include those that use repeating structures (VGG16), inception modules (GoogLeNet), or residual modules (ResNet50) from the Keras apps [35]. Images from CNN models follow a set structure. For training or model weight calculations in your own model, images can be set up with a different pixel input shape or even in a grayscale format. This is useful when loading models from their database into a new baseline model. But, you have to convert your own image format to what these models need in order to benefit from the outcomes that these architectures have already attained in terms of feature recognition or model weight calculations.

Only in situations where we have pre-trained weights are models from the deep learning domain available in Keras applications. Applications for Keras models range widely, including feature extraction, prediction, and fine-tuning. We shall delve deeply into the subject of Keras applications in this essay. The subtopics of "what are keras applications?", "how to use models from Keras.applications?", "keras applications model," and "a conclusion about the same" will be used to try to grasp it.

Keras Available models

Model	Size (MB)	Top-1 Accuracy	Top-5 Accuracy	Parameters	Depth
Xception	88	79.0%	94.5%	22.9M	81
VGG16	528	71.3%	90.1%	138.4M	16
VGG19	549	71.3%	90.0%	143.7M	19
ResNet50	98	74.9%	92.1%	25.6M	107
ResNet50V2	98	76.0%	93.0%	25.6M	103
ResNet101	171	76.4%	92.8%	44.7M	209
ResNet101V2	171	77.2%	93.8%	44.7M	205
ResNet152	232	76.6%	93.1%	60.4M	311
ResNet152V2	232	78.0%	94.2%	60.4M	307
InceptionV3	92	77.9%	93.7%	23.9M	189
InceptionResNetV2	215	80.3%	95.3%	55.9M	449
MobileNet	16	70.4%	89.5%	4.3M	55
MobileNetV2	14	71.3%	90.1%	3.5M	105
DenseNet121	33	75.0%	92.3%	8.1M	242
DenseNet169	57	76.2%	93.2%	14.3M	338
DenseNet201	80	77.3%	93.6%	20.2M	402
NASNetMobile	23	74.4%	91.9%	5.3M	389
NASNetLarge	343	82.5%	96.0%	88.9M	533
EïcientNetV2B0	29	78.7%	94.3%	7.2M	-

Table 3.1 List of available models in keras [36].

Deep learning models are utilised in Keras applications for prediction, feature extraction, and fine-tuning. All required weights are downloaded and saved in the /.keras/models folder automatically when the model is instantiated. Upon instantiating the Keras apps, the models are constructed using the image data format and all configurations specified in the Keras file, which may be found at.keras/Keras.JSON. As an example, suppose that the image_data_format setting is set to the value of channels_last. In this case, the model loads automatically from this repository based on the TensorFlow convention data format, which is then followed by height, width, and depth.

Transfer learning is the process of using these pre-trained CNNs as a component for feature or weight computation in your own computer vision code, such as for picture and emotion identification.

3.7 Implementation Tools

Since there are currently more than 2000 high-level programming languages [16], choosing the right one to construct the system was a difficult process. Fortunately, the project requirements made this problem easier. The programming language used during the load trend model phase had to be easy to learn, practical, and compatible with the GPU libraries because the project used neural networks. Additionally, it was critical that it had adequate documentation and community support so that issues that arose during the development cycle could be resolved quickly.

The GUI design and neural network implementation must a simple and user-friendly programming language. The language must be quick enough to handle this kind of input and the classification process since the programmer classifies photos in real-time. These factors led to a significant reduction in the list. R, C++, Python, and Matlab are some of the most widely used programming languages for machine learning, among others. R is thought to be a powerful language despite having a high learning curve. In contrast, C++ is thought to be the best choice when speed is a concern, but it is challenging to learn and even more so to optimize. Mat Lab is a well-rounded substitute that operates satisfactorily and is quick and easy to use. Python is an easy-to-use programming language that has a number of libraries that make it possible to use GPUs transparently; however, the development cycle is longer than with other programming languages, and using the main product and some of its libraries requires paying a license [37],

[38]. The elements utilized to construct the system are described in the next part in accordance with the Library and Tools chosen.

1. Python: Python stands out as a potent programming language widely embraced in the machine learning community. Data scientists and machine learning professionals prefer it for its simplicity, readability, and extensive library ecosystem. Python boasts robust libraries such as TensorFlow, PyTorch, and scikit-learn, providing efficient tools for constructing and enhancing machine learning models. Its straightforward syntax enables rapid development, allowing for swift prototyping and experimentation. With substantial community support and ongoing development, Python has become the language of choice for developing and deploying machine learning solutions, fostering a vibrant ecosystem of tools, tutorials, and frameworks.
2. Keras: is a high-level neural network framework that operates on top of TensorFlow and gives deep learning model designers and trainers an intuitive interface. By removing low-level details and providing pre-built layers and models, it streamlines the process of creating neural networks and makes it usable by both novice and seasoned practitioners.
3. OpenCV: It is an open source library that provides more than 2500 optimized computer vision and machine learning algorithms through C++, Python, and Java interfaces. Tasks like face detection, object recognition, object tracking, etc. are made easier by these techniques [39].
4. TensorFlow: is a well-known deep learning framework open-source that is used to create and train machine learning models. It gives developers access to a wide range of tools and frameworks that enable them to build neural networks and do calculations quickly on CPUs or GPUs.
5. Microsoft Visio: This tool is used to design the research methodology and suggested experiment system's architecture. A collection of programming functions with a focus on real-time computer vision is called Open CV (Open Source Computer Vision). OpenCV is used because it provides simple capabilities for carrying out many tasks without requiring implementations.

3.8 Development

Another crucial stage of this process is development, where risks included underestimating development time or writing bad, difficult code. Consequently, a modular design-based technique was employed to prevent these hazards. Software development is made easier by modular design, which divides a system into smaller units. The main concept behind this system is that each component carries out a defined function, and when one module is changed, the others are either not impacted at all or just slightly.

The training portion and the application portion of this phase were separated. The following modules were developed for the training section:

- The input module is responsible for extracting the labels and features.
- The DCNN module constructs the DCNN model after receiving the features and labels.

The following modules were created for the application part:

- The GUI module, which is the key element. The input data, the categorization of each facial expression, and the probability of each expression are all displayed in this module.
- The input processing module processes the picture using the same procedures as the input module in the training section. It reads the input data based on the source image, and it processes the image accordingly.
- The DCNN application module: utilizing the previously trained DCNNs, it receives the pre-process vector and returns the classification and probability for each expression.

3.9 Evaluation Methods

Examining a model is essential to determining its utility and efficacy. In computational problems like detection and classification, evaluation metrics like accuracy, recall, precision, and F1-score are used to determine the instance's class membership. These measurements were computed using the detection metrics. The detection metrics provide information about the efficacy of a model for every class. Each of the following metrics was created using the confusion matrix value:

3.9.1 Confusion Matrix

Multiple categorical outputs are produced by classification models. While the majority of error measurements can determine the overall error in our model, we are unable to identify specific flaws within our model. It is possible that the model misclassifies some categories more than others, but a conventional accuracy metric does not show this.

In assessing the effectiveness of machine learning classification models, the confusion matrix is frequently employed. This matrix facilitates the comparison of predicted outcomes for the target attribute with the actual values, providing a comprehensive evaluation of the model's performance.

Confusion Matrix		**Actual**	
		Emotion Detected Positive (1)	**Emotion Not Detected Negative (0)**
Predicted	**Emotion Exist Positive (1)**	TP[1,1] True Positive	FP[1,0] False Positive
	Emotion Not Exist Negative (0)	FN[0,1] False Negative	TN[0,0] True Negative

Table 3.2 Binary Classification Problem using Confusion Matrix (2x2 matrix)

Table 3.2 the provided matrix is a confusion matrix designed for binary classification model outputs. Within this matrix, the terms "Positive" and "Negative" denote the two classes intended to be distinguished by the model.

- **True Positive**

  These are instances where the actual value is 1, and the predicted value is also 1.

- **True Negative**

  Instances classified as True Negatives are those where the true value is 0, and the predicted value is also 0.

- **False Positive**

  False Positives occur when the true value is 0, but the predicted value is 1.

- **False Negative**

  False Negatives are instances where the true value is 1, but the predicted value is 0.

3.9.2 Accuracy Rate

The full correctness of the classification examples to their sense of belonging is known as accuracy. The following description provides an answer to the issue of how often the model properly predicts the class, i.e., a normal face image and a correct emotion picture. This formula is used to determine it. Generally, it is a measure of how often the classifier guesses correctly. The accuracy value is a value between 0 and 1.

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}$$

3.9.3 Precision

It is a measure of how accurately predicted from all classes and shows the likelihood that a prediction of a positive value will come true; For example, calculating the likelihood that a picture has an illness. The following formula is used to calculate precision:

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}$$

3.9.4 Recall

The recall is calculated by determining the ratio of accurately classified positive samples to the total count of positive samples. This metric measures the model's ability to identify positive instances effectively. A higher recall indicates that positive samples are being identified more frequently. In the context of predicting transactions, the recall metric signifies the proportion of predicted transactions that are correctly identified as positive.

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}$$

3.9.5 F1-Score

One of the measures has a value of 0 when the lowest F1-score number is 0. The harmonic mean of recall and precision is known as the F1-score. It denotes extreme memory or precision. This is the appearance of the computation. In general, the F1 score value shows the harmonic mean of precision and recall values.

$$\text{F1 Score} = 2 * \frac{\text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}}$$

Furthermore, suppose that the given data indicates a significant difference in the makeup of the classes. A model may be able to forecast the majority class in every scenario and have a high accuracy score when that class has more data instances than the other classes, but it will not be able to anticipate the minority classes. In this case, confusion matrices are useful.

Chapter 4

Experimental Results and Discussions

4.1 Overview

The training module's primary function is to train the DCNNs, and this chapter discusses how to implement both the training and application modules in detail. The other half of the chapter gives information on the suggested modules for the main system. Finally, the decision made and the experiments carried out during the training stage to assess the optimal hyper parameters and DCNN model are included in the other part, along with the experiments carried out during the test stage to examine the correctness of the models.

4.2 Dataset outlook

Researchers and practitioners use the FER2013 dataset to train and benchmark the performance of various machine learning and deep learning models for facial emotion recognition. In this research, the FER 2013 dataset underwent preprocessing, involving the generation of a CSV file. This file included the following: emotion labels, image pixel values, and a usage column indicating which images were from the training or validation set. 35,887 48x48 pixel grayscale photos that are divided into seven different classes to symbolize different emotions make up the dataset. In this paper, we commonly use this dataset for training and evaluating models for our facial expression analysis. The CV looks like this:

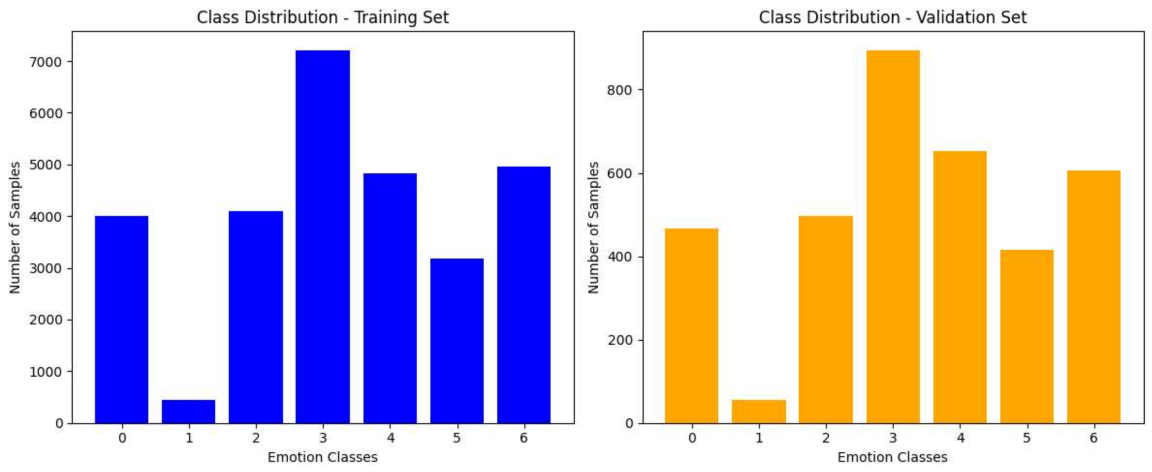


Figure 4.1 Distribution of classes for the training and validation sets, respectively.

Label mapping with seven distinct emotions can be found in the columns of the Fer2013 dataset. The following mapping is used: 0-anger, 1-disgust, 2-fear, 3-happiness, 4-sadness, 5-surprise, and 6-neutral. 22968 is used for training, and this study took validation into consideration.

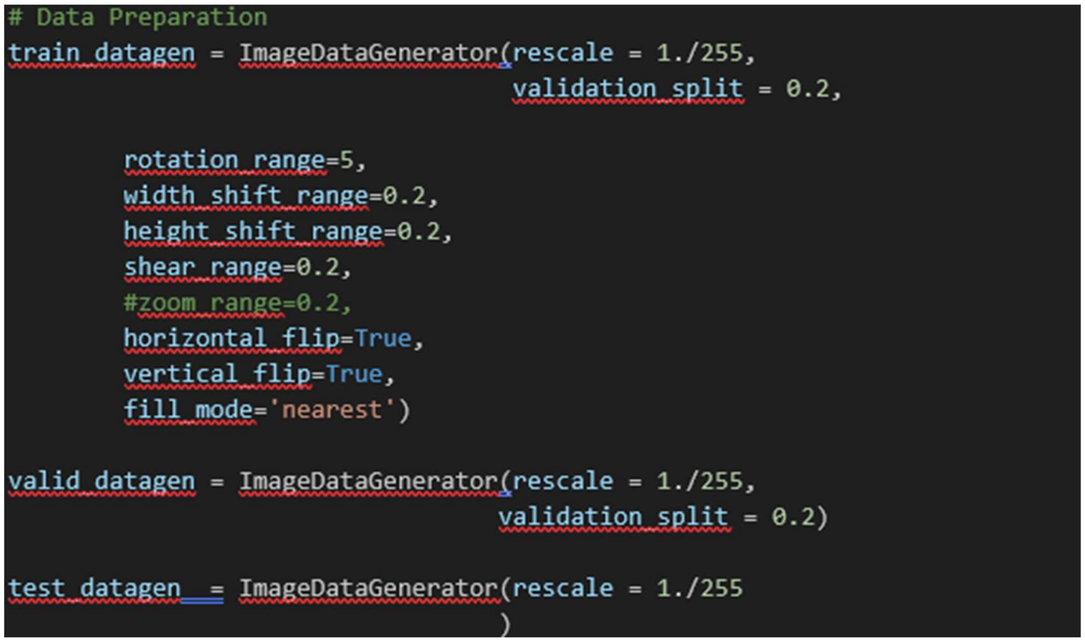
4.3 Experiment Setting

To develop and train a Convolutional Neural Network (CNN) model for emotion recognition, I initially preprocessed the dataset by standardizing all images to a consistent size of 48 x 48 pixels. For training the transfer learning model focused on recognizing facial emotions, I began by loading the pre-trained GoogleNet(V3), ResNet50, and VGG-16 models, subsequently freezing all of their pre-existing layers. Following this, I introduced a new fully connected layer with 48 units for VGG-16 and a final softmax layer for classification purposes. The Hall model's pre-trained layers ought to be frozen as well, meaning that their weights shouldn't change while the model is being trained. In addition to preventing overfitting on the training set, freezing the pre-trained layers helps expedite the training process.
Next, I used the identical training and validation sets as the DCNN models to train the model. The pre-trained layers are kept frozen throughout training, while the weights of the final softmax layer and the newly connected layer are modified. This technique helps the model learn new representations specific to the emotion recognition task by utilizing the previously trained

representations from ImageNet. Metrics like accuracy and precision can be used to assess the performance of the GoogLeNet(V3), ResNet50, and VGG16 transfer learning models on the validation set after training. The outcomes can then be compared.

4.3.1 Data Preparation

This is a common practice to artificially increase the size of our training dataset and improve the generalization ability of our model so we use ‘ImageDataGenerator’ it is a powerful Keras tool for performing real-time data augmentation during training of neural networks. Data augmentation encompasses the application of diverse transformations to the training images, enhancing dataset diversity and refining the model's ability to generalize. In this study, ImageDataGenerator objects were employed for data augmentation, facilitating the augmentation and preprocessing of image data for both the training and validation sets..



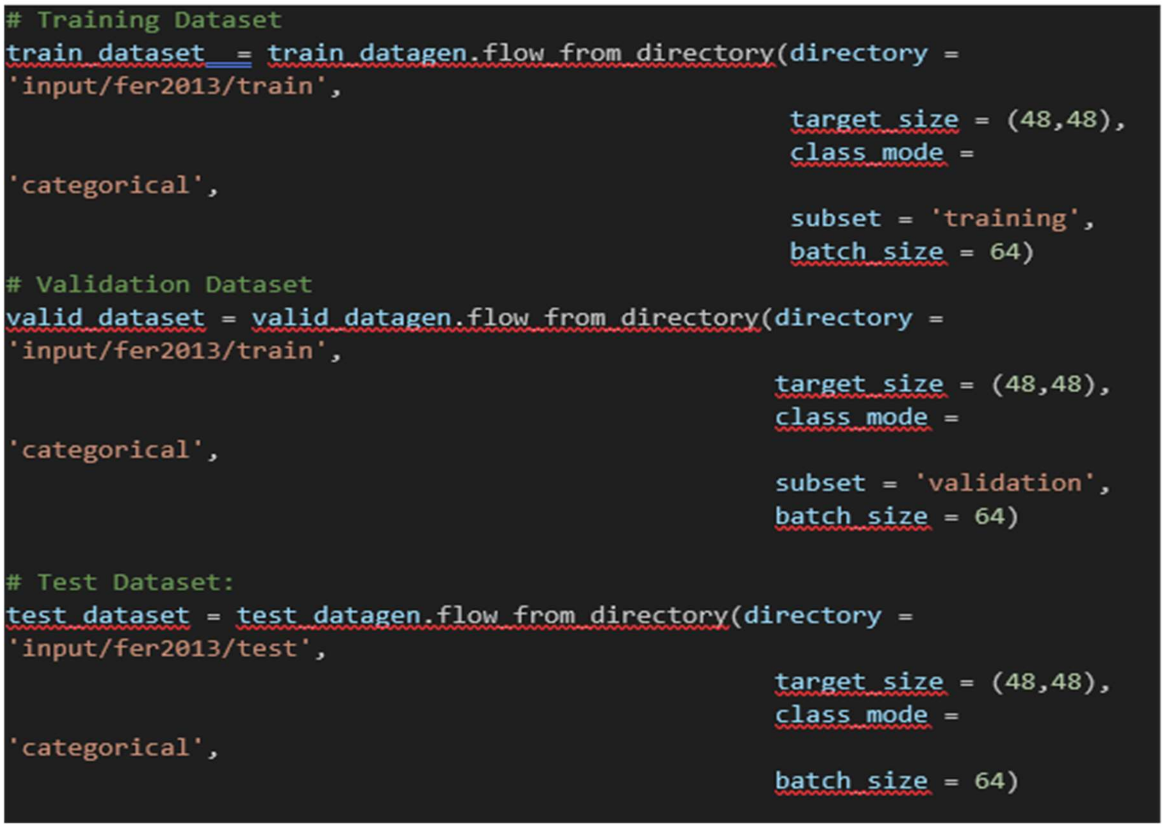
```
# Data Preparation
train_datagen = ImageDataGenerator(rescale = 1./255,
                                   validation_split = 0.2,

        rotation_range=5,
        width_shift_range=0.2,
        height_shift_range=0.2,
        shear_range=0.2,
        #zoom_range=0.2,
        horizontal_flip=True,
        vertical_flip=True,
        fill_mode='nearest')

valid_datagen = ImageDataGenerator(rescale = 1./255,
                                  validation_split = 0.2)

test_datagen  = ImageDataGenerator(rescale = 1./255
                                  )
```

Figure 4.2 data preparation

The train_datagen object played a pivotal role in augmenting the training set. This object offers a range of transformation choices, including image rotation by degrees, width and height shifting, and horizontal image flipping. Moreover, the fill_mode parameter was configured to 'nearest,' ensuring that vacant pixels are filled with the nearest pixel value. Throughout the training and validation stages, these ImageDataGenerator objects were crucial in producing batches of augmented and preprocessed images.

4.3.2 Load the Training Dataset and Validation Dataset

Setting up data generators for training, validation, and testing using the Keras ‘ImageDataGenerator’ class for a convolutional neural network (CNN). This is a common approach when working with image data in deep learning.

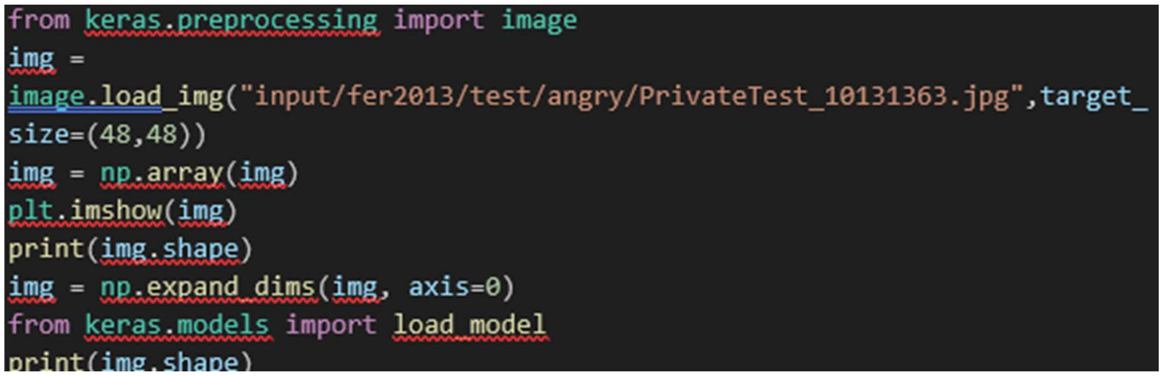


```
# Training Dataset
train_dataset  = train_datagen.flow_from_directory(directory =
'input/fer2013/train',
                                                   target_size = (48,48),
                                                   class_mode =
'categorical',
                                                   subset = 'training',
                                                   batch_size = 64)
# Validation Dataset
valid_dataset = valid_datagen.flow_from_directory(directory =
'input/fer2013/train',
                                                  target_size = (48,48),
                                                  class_mode =
'categorical',
                                                  subset = 'validation',
                                                  batch_size = 64)

# Test Dataset:
test_dataset = test_datagen.flow_from_directory(directory =
'input/fer2013/test',
                                                  target_size = (48,48),
                                                  class_mode =
'categorical',
                                                  batch_size = 64)
```

Figure 4.3 Class distribution for the validation as well as training sets, respectively

Train_datagen is likely an instance of ImageDataGenerator configured for data augmentation. Images are loaded from the 'input/fer2013/train' directory and then resized to a target size of (48, 48). The class mode is set to 'categorical' indicating that the labels are one-hot encoded. The subset is set to 'training' to indicate this is for training. The batch size is set to 64. 'Validation' indicating this is for validation, Similar to the training dataset, but for validation, it finally goes on the test dataset. After setting up these data generators, we can use them to train your CNN model using the fit_generator function in Keras.

Loading an image using Keras's ‘image.load_img’ function, converting it to a NumPy array, and then expanding its dimensions. This is a common preprocessing step before feeding an image to a neural network here’s a breakdown of your code:



```
from keras.preprocessing import image
img =
image.load_img("input/fer2013/test/angry/PrivateTest_10131363.jpg",target_
size=(48,48))
img = np.array(img)
plt.imshow(img)
print(img.shape)
img = np.expand_dims(img, axis=0)
from keras.models import load_model
print(img.shape)
```

Figure 4.4 Loading an image using Keras's

The 'image.load_img' function is used to load the image from the specified file path and resize it to the target size of (48, 48), then the loaded image is converted to a NumPy array using 'np.array'. After that, the 'plt.imshow' function is used to display the image using Matplotlib. The shape of the image is printed using ‘print (img.shape), and then the 'np.expand_dims' function adds an extra dimension to the array to make it compatible with the expected input shape of your model. A snippet is appropriate for preparing a single image for input to a model. If we are working with batches of images, we may need to adapt the code accordingly. Additionally, ensure that our model is loaded and ready to accept the input shape specified by img.shape.

4.3.3 Base Model

By introducing a fresh classification head atop a pre-trained model, it becomes possible to fine-tune the model for image classification in a new domain, even with a smaller dataset. By utilizing the features that were obtained from the ImageNet dataset, this method helps the model adjust to the subtleties of the new domain. The adjusted model can then be trained on a smaller dataset unique to the new domain, which will result in high accuracy.

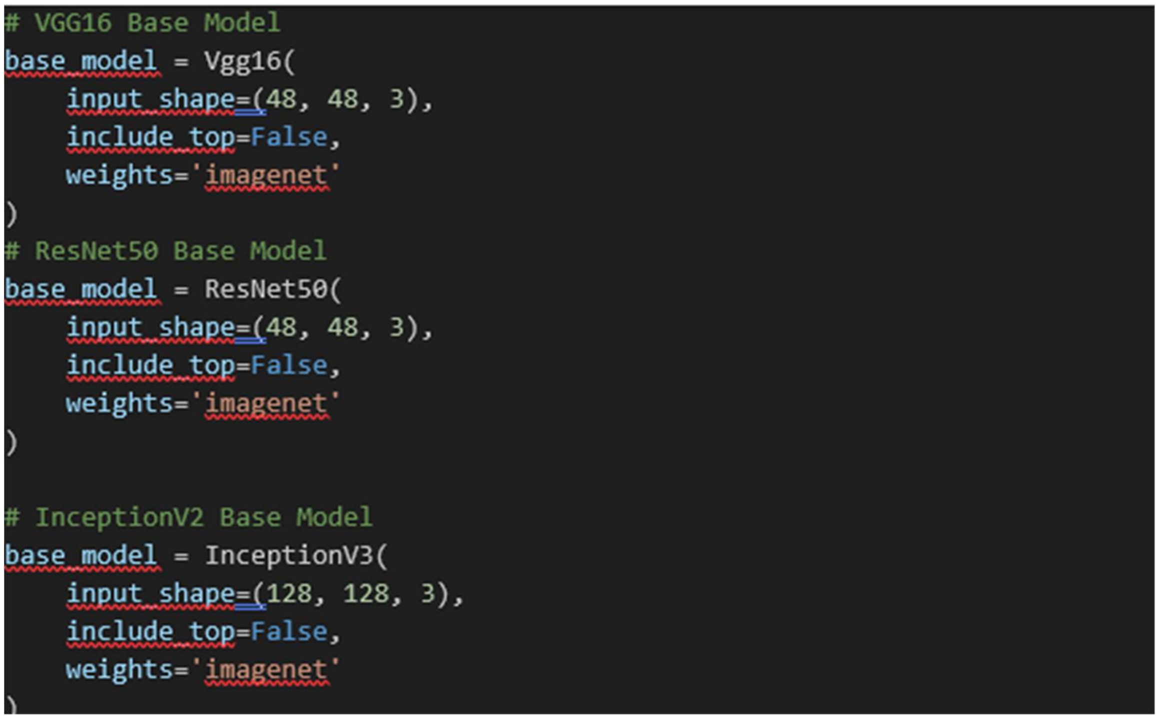

```
# VGG16 Base Model
base_model = Vgg16(
    input_shape=(48, 48, 3),
    include_top=False,
    weights='imagenet'
)
# ResNet50 Base Model
base_model = ResNet50(
    input_shape=(48, 48, 3),
    include_top=False,
    weights='imagenet'
)

# InceptionV2 Base Model
base_model = InceptionV3(
    input_shape=(128, 128, 3),
    include_top=False,
    weights='imagenet'
)
```

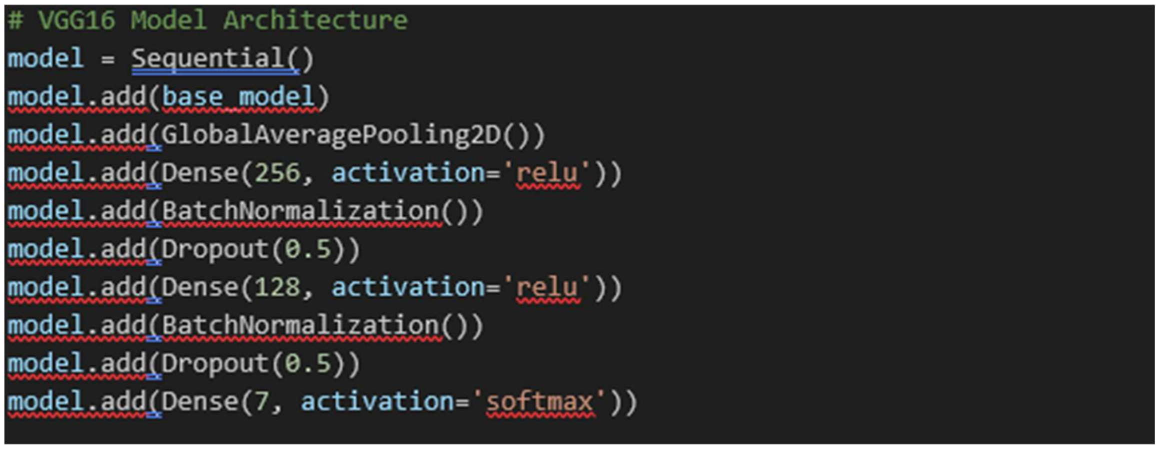
Figure 4.5 Base Model

This code snippet outlines a function, base_model, responsible for constructing the classification head over pre-trained convolutional neural networks, namely InceptionV3, VGG16, and ResNet50, imported from Keras. These models have been pre-trained on the ImageNet dataset. Ultimately, the code establishes a new instance of the Keras Model, incorporating a pre-trained model. The primary objective of these lines of code is to formulate a deep learning model tailored for image classification. The utilization of pre-trained models such as VGG16, GoogLeNet (InceptionV3), and ResNet50 capitalizes on their ability to already comprehend and extract features from images.

It is advisable to freeze the pre-trained layers of the entire model, implying that the weights of these layers remain static and are not modified during the training phase. This strategy of freezing pre-trained layers serves to expedite the training process and mitigate the risk of overfitting to the training data.

4.3.4 Define the Model Architecture

There are differences in the model architecture between the models that were used for emotion recognition. An illustration is provided with examples of existing model architectures, namely the VGG16, InceptionV3, and ResNet50 models utilized in pre-training. The weight "imagenet" is incorporated into these models. It is crucial to note that the architecture of the model cannot be altered as the weights have been trained for a specific input configuration. Hence, when modifying a property of the image, such as changing it from a 1-channel grayscale to a 3-channel RGB image, adjustments must be made to accommodate these variations. Let's examine one of our model architectures, for instance.



```
# VGG16 Model Architecture
model = Sequential()
model.add(base_model)
model.add(GlobalAveragePooling2D())
model.add(Dense(256, activation='relu'))
model.add(BatchNormalization())
model.add(Dropout(0.5))
model.add(Dense(128, activation='relu'))
model.add(BatchNormalization())
model.add(Dropout(0.5))
model.add(Dense(7, activation='softmax'))
```

Figure 4.6 VGG16 Model Architecture

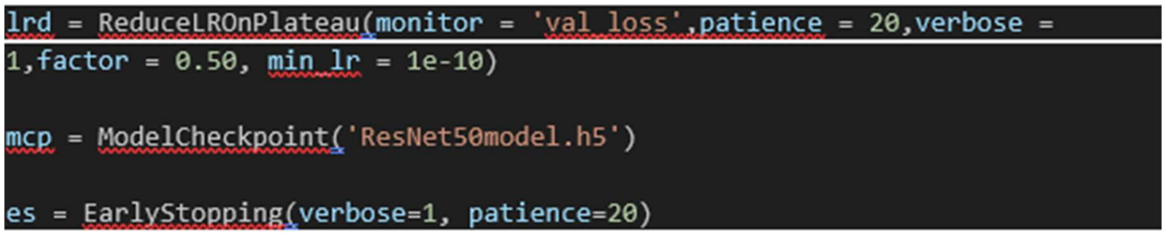
Those architecture is a common pattern in transfer learning, where a pre-trained base model is followed by additional layers to adapt the model to a specific task.

Loading the pre-trained ResNet50 model without the top layer is the initial step in the ResNet50 transfer learning model, just like it did with the VGG16 model. The final classification, which is unique to the ImageNet dataset, is handled by the top layer of the ResNet50 model. The pre-trained layers of the ResNet50 model should also be frozen so that we can use the model for a different classification job, such face detection and identification, by removing this layer.

A new fully connected layer with 128 units can be added on top of the frozen layers after loading the pre-trained ResNet50 model. The pre-trained layers and the final classification layer are joined by this new completely connected layer. Depending on the particular job and dataset, experimenting may be used to determine the number of units in the fully linked layer.

4.3.5 Callback Functions

This function ensures the learning rate is adjusted, the model is saved, and training is stopped early if certain conditions are met. Adjust the parameters of these callbacks based on our specific needs and the behavior we want during training.



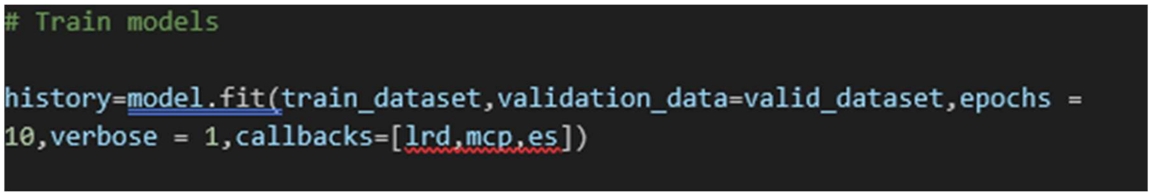
```
lrd = ReduceLROnPlateau(monitor = 'val_loss',patience = 20,verbose =
1,factor = 0.50, min_lr = 1e-10)

mcp = ModelCheckpoint('ResNet50model.h5')

es = EarlyStopping(verbose=1, patience=20)
```

Figure 4.7 Callback Functions

Monitors the validation loss ('val_loss') and reduces the learning rate if no improvement is observed for a specified number of epochs (patience). The learning rate is reduced by a factor of 0.50 when the condition is met. It is stated that the minimum learning rate is 1e-10 (min_lr). Finally, use the Keras API to assemble and fit a deep learning model. The model. Fit method, which creates batches of augmented images from the training set using the train_dataset data generator, is invoked to start the training process.

4.3.6 Train models

Assign the names X_valid and Y_valid to the validation data. Divide the training set's length by the batch size to set the steps_per_epoch parameter. Give the fit method the list of callbacks, which includes the ReduceLROnPlateau and Early Stopping callbacks that were previously defined. Use_multiprocessing should ultimately be set to true in order to take advantage of multiple CPU cores to speed up data generation.



```
# Train models

history=model.fit(train_dataset,validation_data=valid_dataset,epochs =
10,verbose = 1,callbacks=[lrd,mcp,es])
```

Figure 4.8 Train models

- **model.fit** is the function used to train the model.
- **train_dataset** is the training dataset generated using an **ImageDataGenerator**.
- **validation_data** is the validation dataset for monitoring model performance during training.
- **epochs=10** specifies the number of training epochs.
- **verbose=1** prints updates during training.
- **callbacks=[lrd, mcp, es]** includes the defined callback functions.

The **fit** method returns a **history** object, which contains information about the training process such as loss and accuracy over epochs. We use this information to analyze and visualize how our model is learning. Make sure that our model architecture, loss function, and optimizer are set up appropriately before calling the **fit** method. If there are any issues or error messages, we check and address them based on the specific context of code. Using the fit method to train your model on the training dataset (train_dataset) and validate it on the validation dataset (valid_dataset) for 10 epochs. You've also included the callback functions lrd (ReduceLROnPlateau), mcp (ModelCheckpoint), and es (EarlyStopping) to control the training process.

4.4 Experiment Result

4.4.1 Confusion matrix

The confusion matrix comprises a 7x7 grid that juxtaposes predicted labels with true labels across various emotions. The matrix's values denote the frequency of instances where a particular emotion was predicted compared to the true emotion. Cells along the diagonal signify accurately predicted emotions, indicating a match between the predicted and true labels. Conversely, off-diagonal cells indicate instances of misclassification, where the predicted label diverges from the true label.

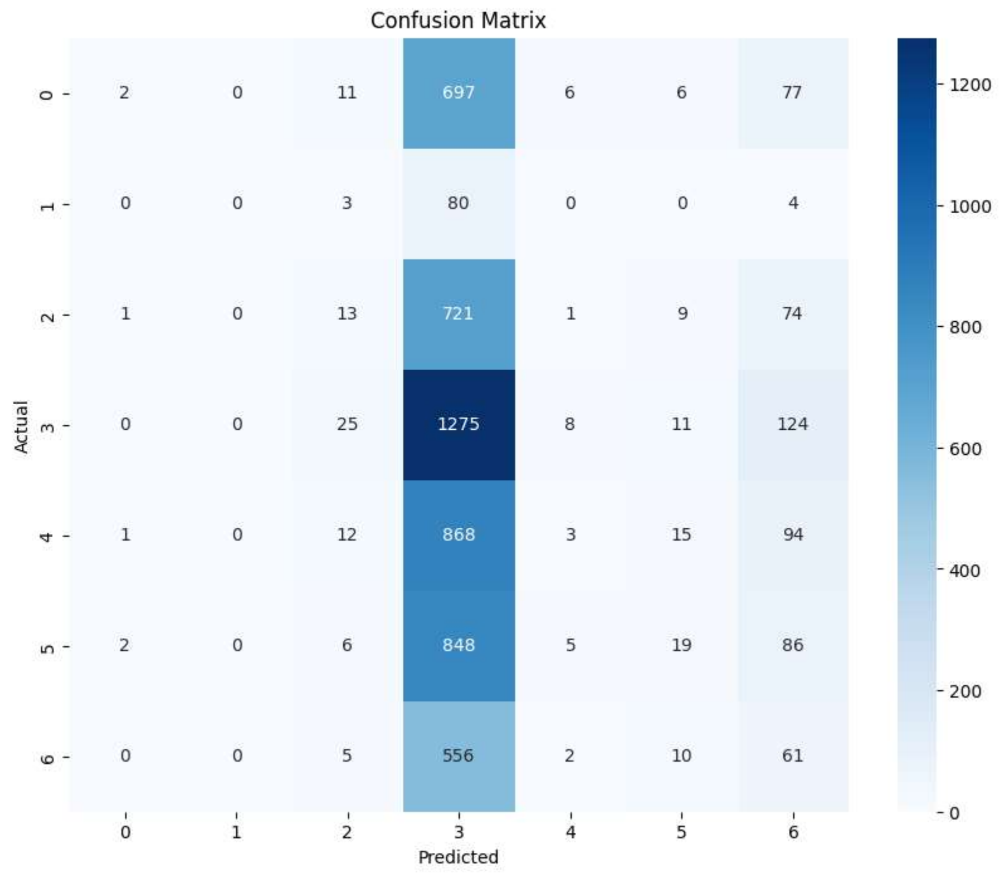


Figure 4.9 The average confusion matrix across ten epochs for the pre-trained Resnet50 model.

Each row corresponds to the true class, while each column represents the predicted class. Notably, the model seems to perform well in correctly predicting instances for the third and sixth classes, with 721 and 19 correct predictions, respectively. However, it struggles with accurate predictions for the first, fourth, and fifth classes, as evidenced by higher misclassification counts, such as 697, 1275, and 868 instances, respectively.

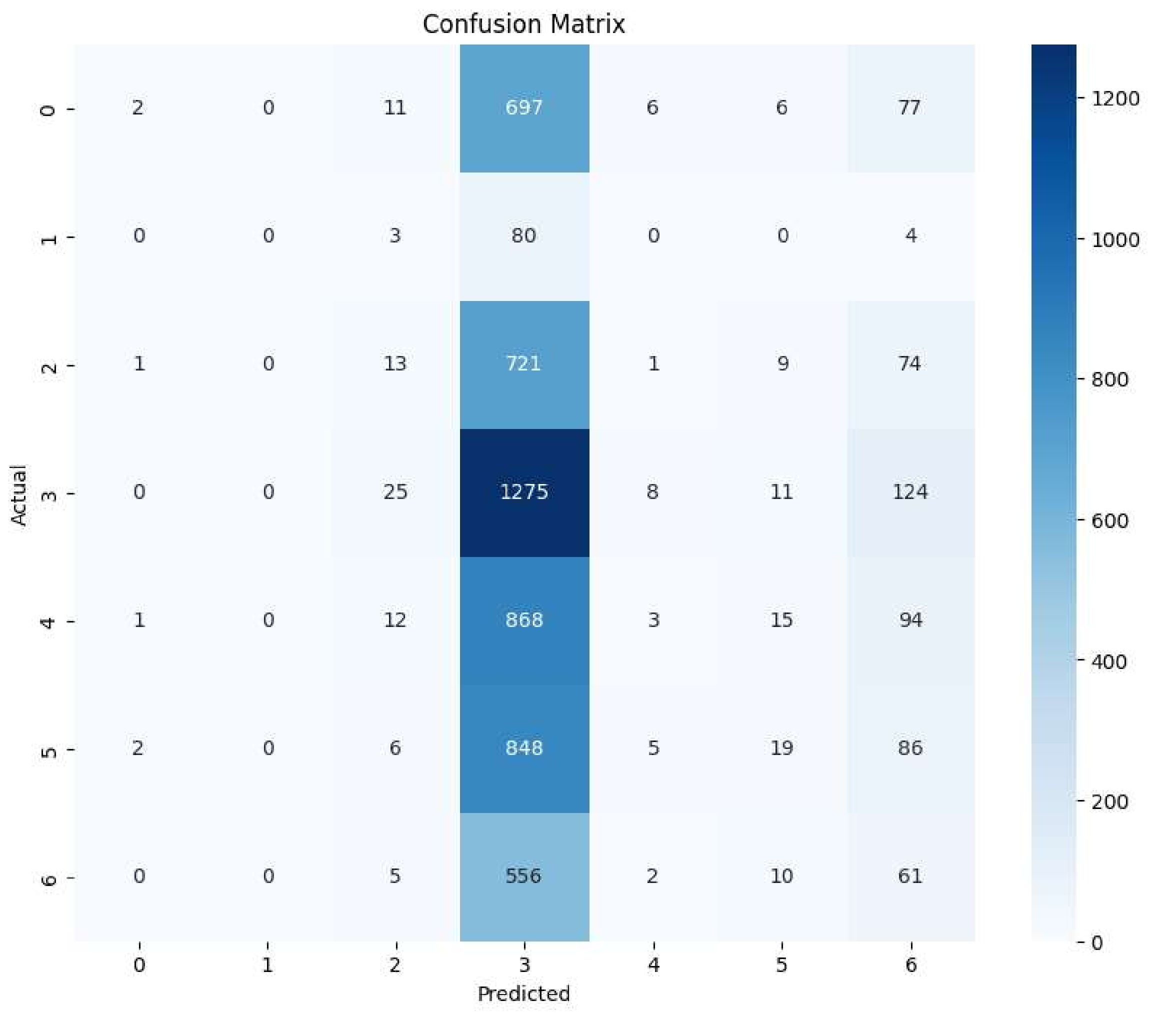


Figure 4.10 The average confusion matrix across ten epochs for the pre-trained VGG16 model.

It is evident that the model performs well for certain classes, such as the fourth class with 714 correct predictions, while struggling with others, as indicated by higher off-diagonal values. The most frequent misclassifications appear to involve the third and sixth classes, with 388 and 582 instances respectively, being misclassified as class four. Overall, a more detailed analysis of precision, recall, and F1 scores for each class would provide a more comprehensive evaluation of the model's performance. In VGG16

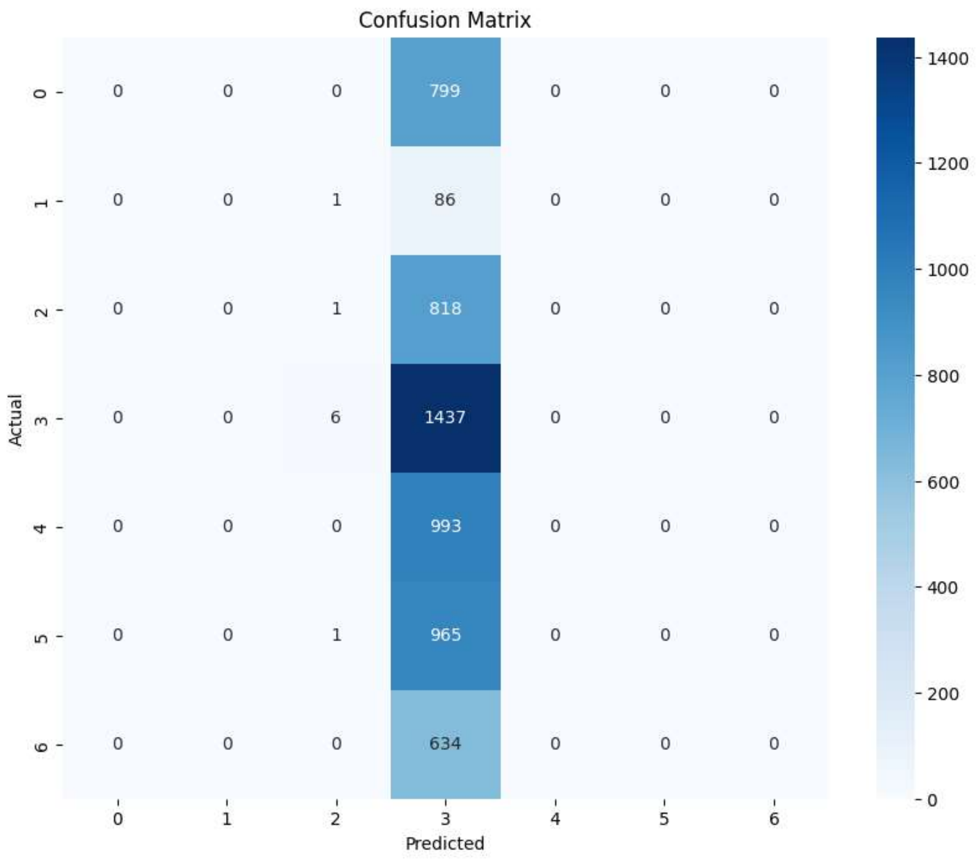


Figure 4.11 the average confusion matrix across ten epochs for the pre-trained InceptionV3 model.

Model consistently predicts class 3, with a count of 1437, while seldom predicting other classes. This suggests a significant imbalance in the model's predictions, potentially favoring a specific class. The counts of zero in several cells also raise concerns about the model's inability to predict certain classes.

4.4.2 Accuracy and loss

Further analysis and fine-tuning may be needed to enhance the model's performance, especially if certain classes are more challenging or underrepresented in the dataset.

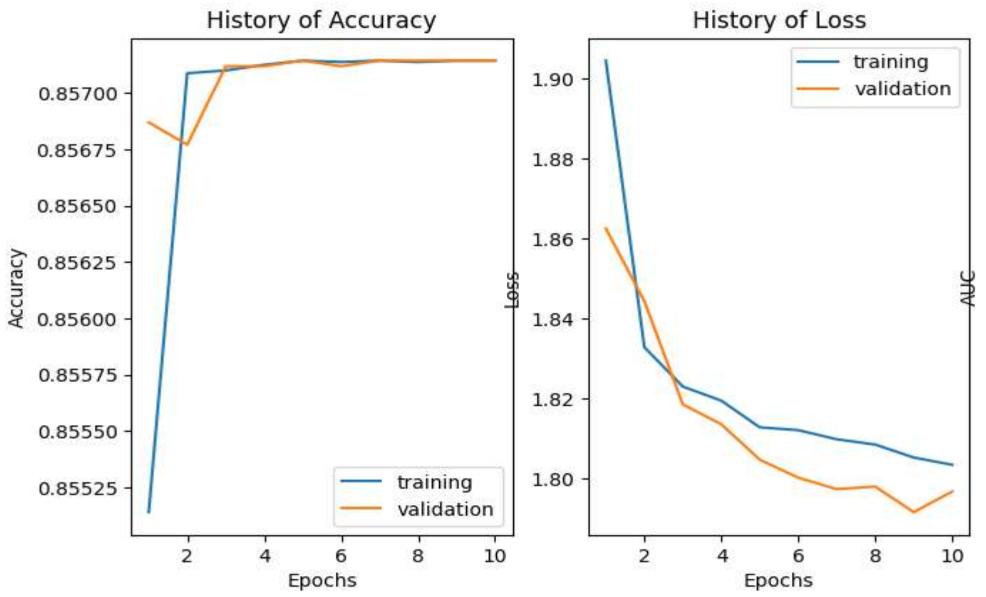


Figure 4.11The training accuracy and loss of the pre-trained Resnet50 model were evaluated over 10 epochs using the Fer2013.csv dataset.

These final training and validation accuracies provided indicate that the machine learning model achieved an accuracy of approximately 85.7% on both the training and validation datasets. This implies that the model performed consistently well on both the data it was trained on and unseen validation data.

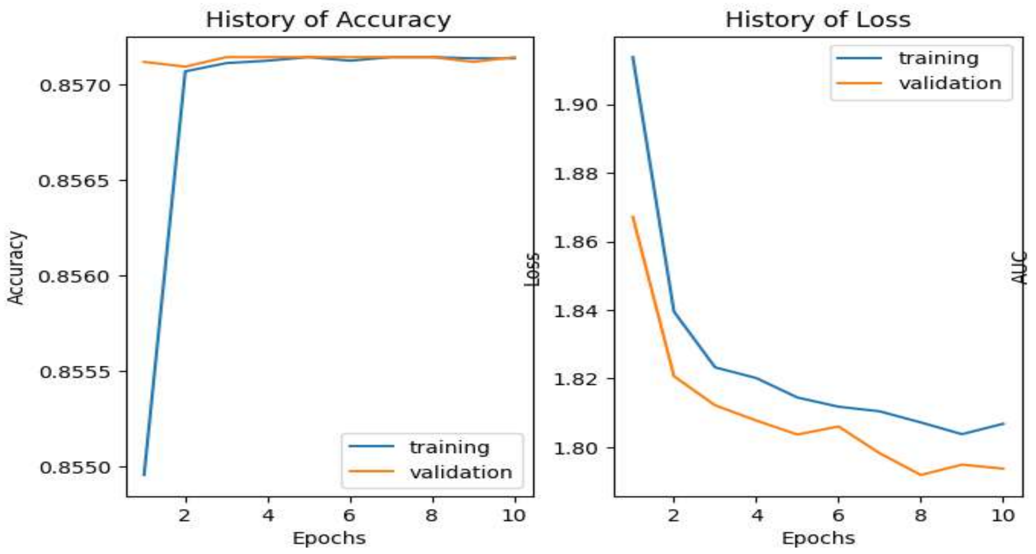


Figure 4.12 The training accuracy and loss of the pre-trained VGG16 model were evaluated over 10 epochs using the Fer2013.csv dataset.

The training and validation accuracies provided offer insights into the performance of a machine learning model during its training process. The final training accuracy stands at 85.9%, indicating that the model correctly predicted the training dataset's target values with high accuracy. The validation accuracy closely follows at 85.8%, suggesting that the model generalizes well to unseen data, maintaining a consistent level of accuracy. These values together indicate that the model has learned effectively from the training data and demonstrates a robust performance on both the training and validation sets.

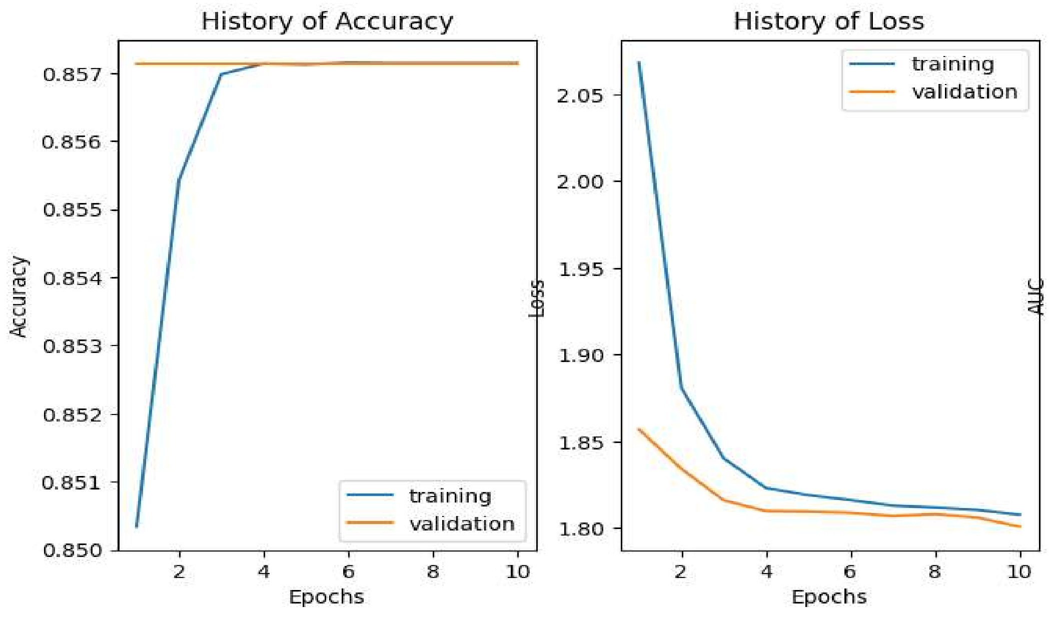


Figure 4.13 the training accuracy and loss of the pre-trained InceptionV3 model were evaluated over 10 epochs using the Fer2013.csv dataset.

The model's overall prediction accuracy, or the percentage of correctly identified cases, is indicated by the reported accuracy rate of 25.05%. With a precision of 7.89%, it is clear that there are difficulties in precisely identifying positive cases because only a tiny percentage of cases that the model predicted as positive are indeed positive. The model's recall value of 25.05% highlights a modest sensitivity to positive cases and shows that it can capture some of the real positive examples.

4.4.3 Each emotion Precision values

The precision of positive predictions is measured. It is defined as the proportion of accurately predicted positive observations to all positive predictions. An average precision value for all classes is provided by the overall precision.

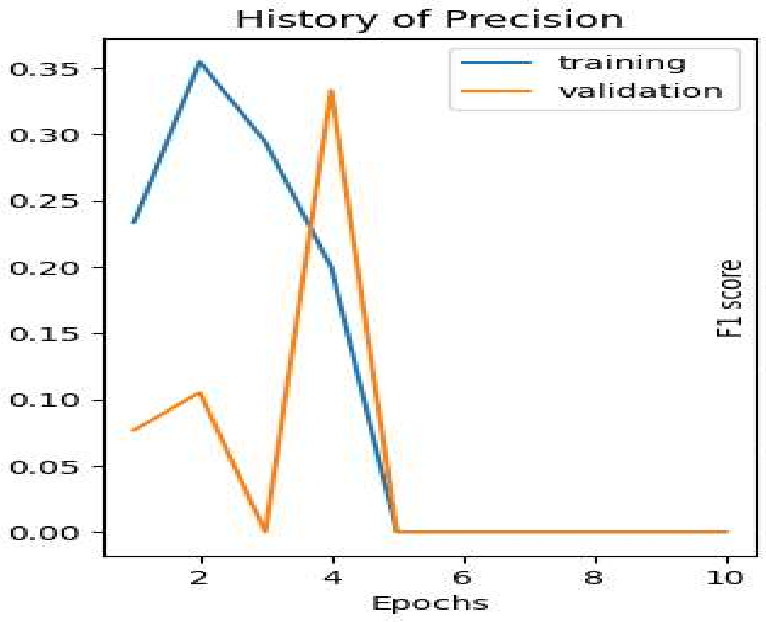


Figure 4.13 The overall precision of 21.40% indicates the average precision across all classes in the emotion recognition Resnet50 model.

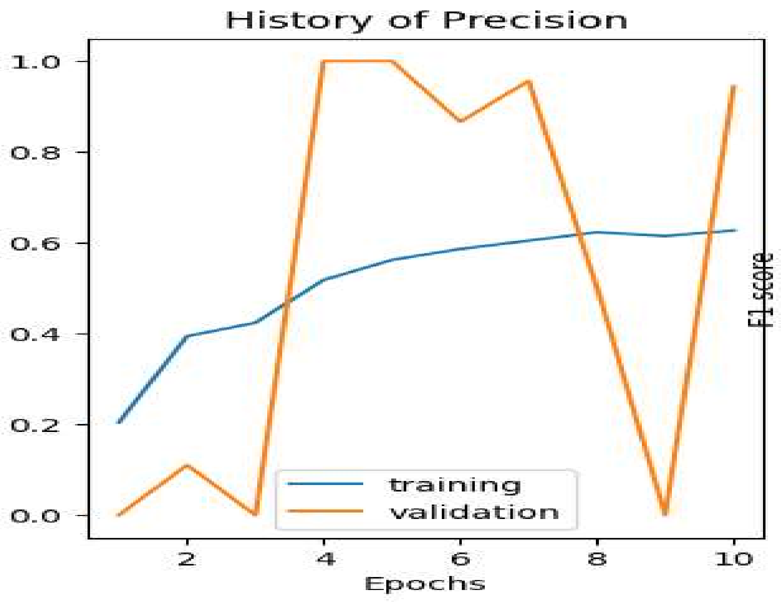
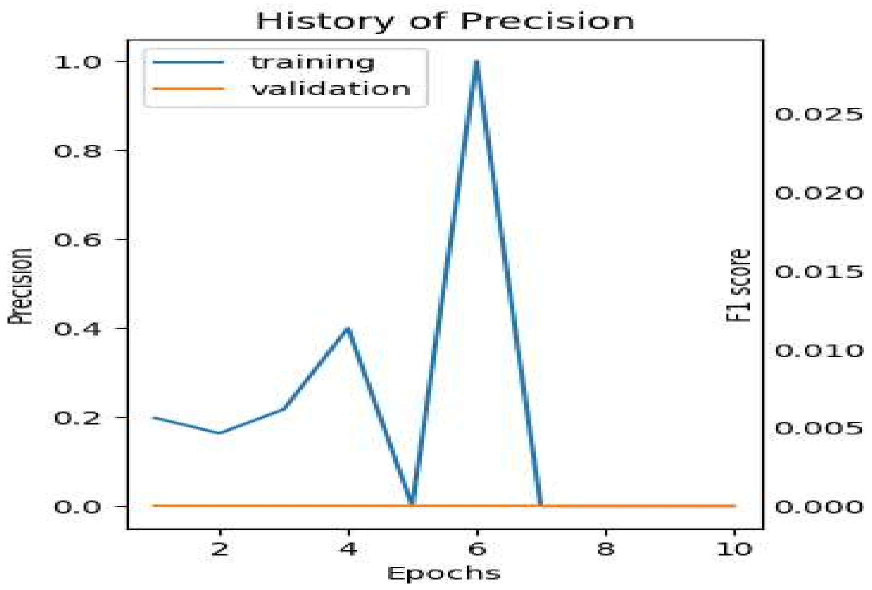


Figure 4.14 The overall precision of 21.40% indicates the average precision across all classes in the emotion recognition VGG16 model.

Figure 4.15 The overall precision of 7.89% indicates the average precision across all classes in



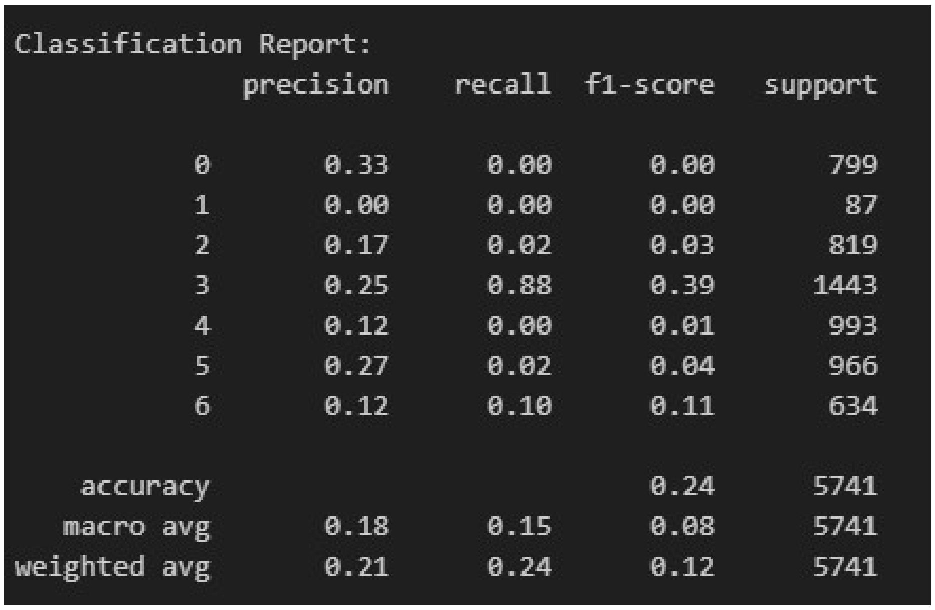
the emotion recognition InceptionV3 model.

4.5 F1-score

The F1-score values provided for each class offer a comprehensive assessment of the classification model's performance across seven different categories. Resnet50 Class 3 stands out with the highest F1-score of approximately 39.3%, indicating a balanced performance in terms of precision and recall. However, Class 1 exhibits a F1-score of 0.0%, suggesting that the model did not correctly classify any instances for this class, emphasizing a significant limitation. The overall F1-score, calculated by considering the harmonic mean of precision and recall across all classes, is 0.1225. In VGG16, classes 0, 1, 4, and 5 exhibit F1-scores of 0.0%, indicating that the model struggled to find a balance between precision and recall for these classes, possibly due to an inability to effectively capture true positives or minimize false positives and false negatives. Class 3 stands out with the highest F1-score of approximately 38.3%, indicating a relatively balanced performance in terms of precision and recall. The overall F1-score, calculated as the harmonic mean across all classes, is 0.1137, providing a consolidated measure of the model's effectiveness in handling both false positives and false negatives across all classes.

4.5.1 Classification report of each class or emotion

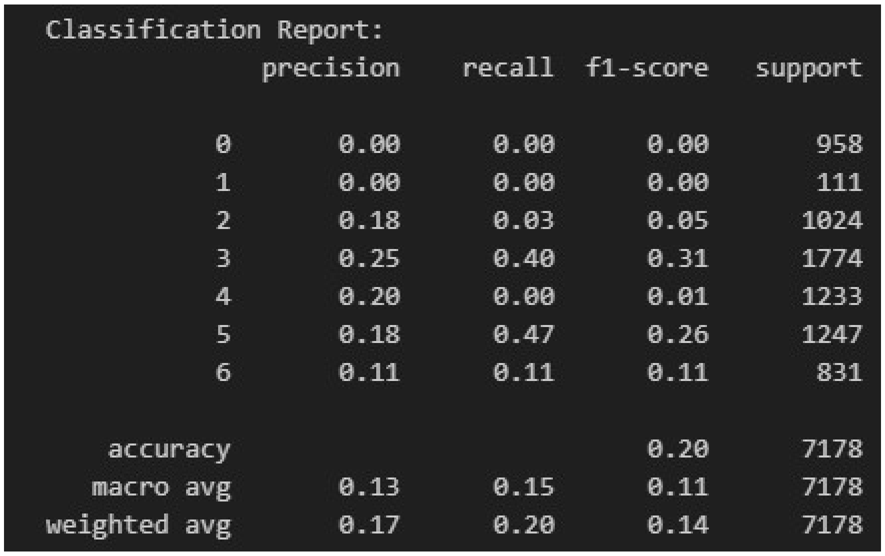
The classification report offers a thorough overview of diverse performance metrics for individual classes in a classification task. It usually includes the F1-score, support values, recall, and precision for every class here's is our classification report based on the precision values we provided:



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

           0       0.33      0.00      0.00       799
           1       0.00      0.00      0.00        87
           2       0.17      0.02      0.03       819
           3       0.25      0.88      0.39      1443
           4       0.12      0.00      0.01       993
           5       0.27      0.02      0.04       966
           6       0.12      0.10      0.11       634

    accuracy                           0.24      5741
   macro avg       0.18      0.15      0.08      5741
weighted avg       0.21      0.24      0.12      5741
```

Figure 4.16 Resnet50 Picture showing the sequential model's categorization report

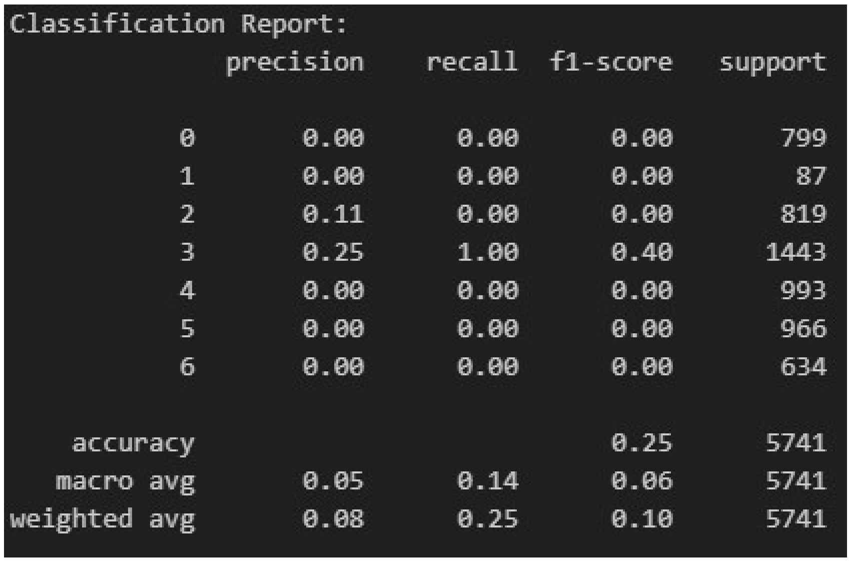
While Class 3 demonstrates relatively higher precision (25%) and recall (88%), suggesting better identification of positive instances, Classes 0, 1, and 4 exhibit poor precision and recall, with Class 0 having the highest precision but low recall. The overall accuracy is 24%, with the macro and weighted average F1-scores indicating a generally modest model performance.



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

           0       0.00      0.00      0.00       958
           1       0.00      0.00      0.00       111
           2       0.18      0.03      0.05      1024
           3       0.25      0.40      0.31      1774
           4       0.20      0.00      0.01      1233
           5       0.18      0.47      0.26      1247
           6       0.11      0.11      0.11       831

    accuracy                           0.20      7178
   macro avg       0.13      0.15      0.11      7178
weighted avg       0.17      0.20      0.14      7178
```

Figure 4.17 VGG16 Picture showing the sequential model's categorization report

The classification report reveals a model with limited performance in predicting emotional classes, as evidenced by low precision, recall, and F1-scores across most classes. Classes 0, 1, and 4 exhibit particularly poor performance, with precision and recall both at 0.00%, indicating the model's inability to correctly identify positive instances for these classes. Although Class 3 demonstrates relatively better precision 25% and recall 40%, the overall model accuracy remains low at 20%. The macro and weighted average F1-scores also indicate an overall subpar model performance.



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

           0       0.00      0.00      0.00       799
           1       0.00      0.00      0.00        87
           2       0.11      0.00      0.00       819
           3       0.25      1.00      0.40      1443
           4       0.00      0.00      0.00       993
           5       0.00      0.00      0.00       966
           6       0.00      0.00      0.00       634

    accuracy                           0.25      5741
   macro avg       0.05      0.14      0.06      5741
weighted avg       0.08      0.25      0.10      5741
```

Figure 4. 18 InspectionV3 Picture showing the sequential model's categorization report

Unfortunately, the model demonstrates significant limitations, as indicated by low precision, recall, and F1-score values for most classes. Notably, classes 0, 1, 2, 4, 5, and 6 exhibit negligible precision, recall, and F1-score values, suggesting a failure to accurately classify instances for these categories. Class 3 stands out with a higher recall of 1.00, indicating the model's success in capturing all positive instances, but precision and F1-score remain relatively modest. The overall accuracy is 25%, underscoring the model's general struggles.

Model	Precision	Recall	F1-Score
Vgg16	16.60%	19.87%	14.26%
Resnet50	21.40%	23.92%	12.25%
inspectionV3	7.89%	25.05%	10.10%

Table 4.1 Results from the FER-2013 test set summary after ten epochs.

The evaluation of three convolutional neural network architectures, VGG16, GoogleNet (V3), and ResNet50, based on precision, recall, F1-Score, and accuracy reveals distinct performance characteristics. ResNet50 stands out with the highest precision, recall, and F1-Score, indicative of a more balanced and accurate classification of instances. VGG16 demonstrates moderate performance across the metrics, while GoogleNet (V3) achieves the highest overall accuracy. However, the latter exhibits lower precision and F1-Score, emphasizing potential trade-offs between precision and recall. The reported accuracy values of 22.66% for VGG16, 25.05% for GoogleNet (V3), and 23.92% for ResNet50 provide an overall measure of correctness in class predictions. The results underscore the importance of considering a range of metrics for a comprehensive evaluation, guiding further adjustments or fine-tuning to optimize each model's performance.

Chapter 5

Conclusion and Future Work

The purpose of this study was to evaluate how well various deep learning models performed when interpolating images of various sizes to detect facial emotions, the findings above demonstrate how the choice of architecture and image size affects the accuracy of deep learning models in computer vision applications. In this work, we used to train three well-known models inspectionV3, VGG16, and ResNet50 for picture resizing, and the models' accuracy was also assessed using two distinct image sizes.

Among these models, ResNet50 demonstrated the highest precision at 21.40%, reflecting its ability to accurately classify instances positively. Additionally, ResNet50 exhibited a recall of 23.92%, signifying its capability to effectively identify a substantial proportion of true positive instances. The F1-Score, which balances precision and recall, for ResNet50 was 12.25%, indicating a relatively balanced performance in capturing both false positives and false negatives. VGG16, while displaying slightly lower precision 16.60% and recall 19.87% than ResNet50, achieved a higher F1-Score of 14.26%, suggesting a better balance between precision and recall. On the other hand, InceptionV3 yielded the lowest precision 7.89% and F1-Score 10.10%, despite having the highest recall at 25.05%. These results underscore the trade-offs between precision and recall in each model, emphasizing the importance of selecting a model aligned with specific task requirements and priorities. The study shows how important it is to choose the right deep-learning architecture and image size for a given job. In order to improve the robustness and accuracy of deep learning models, it also highlights the necessity of continued computer vision research and development, particularly in the areas of interpolation, feature extraction, and data augmentation strategies.

Selecting a deep learning architecture and image size, task-specific considerations are paramount. Various tasks, such as image classification, face and emotion detection, and segmentation, necessitate tailored architectures like VGG, ResNet, Inception, or others, each designed with varying complexities. Balancing model complexity against computational resources is crucial, as deeper architectures capture more intricate patterns but demand increased computational power. Transfer learning proves advantageous for tasks with limited data, allowing fine-tuning of pre-trained models, such as those from ImageNet, and mitigating the need for extensive annotated datasets. The choice of image size involves a trade-off between capturing details and computational efficiency, with considerations for real-time requirements.

References

[1] N. Roopa, " Deep learning for facial expression-based emotion recognition." 2019.

[2] W. K. T. Md. Zia Uddin, "Facial Expression Recognition Using Salient Features and Convolutional Neural Network," 2017.

[3] D. Liliana, "Emotion recognition from facial expression using deep convolutional neural network.," 2019.

[4] L. a. C. R. Bejjagam, "Convolutional neural networks with multiclass classification and Bayesian optimization for hyperparameter tuning are used to recognize facial emotions..," 2022.

[5] M. L. C. L. P. a. P. J. Kumbure, "Machine learning techniques and data for stock market forecasting," 2022.

[6] X. Z. Y.-P. T. Y. S. a. J. Z. Jiwen Lu, "Neighborhood repulsed metric learning for kinship verification," 2014.

[7] A. C. K. Sushmitha, "Facial Emotion Detection," vol. Volume 5, 2019.

[8] A. A. Shervin Minaee, "Deep-Emotion: Facial Expression Recognition Using Attentional Convolutional Network," 2019.

[9] M. H. M. A. A. Z. M. F. G. a. T. Uddin, ". A facial expression recognition system using robust face features from depth videos and deep learning," 2017.

[10] C. A. a. M. B. T. S. Wingenbach, " Correction: A collection of videos expressing low, intermediate, and high intensity emotions serve as the validation for the Amsterdam dynamic facial expression set–bath intensity variations (ADFES-BIV)," Vols. vol. 11, no. 12, 2016.

[11] [Online]. Available: Https://www.libguides.wits.ac.za/c.php?j=693518&p=4914913. [Accessed 14 06 2023].

[12] J. S. Gero, "Research Methods for Design Science Research: Computational and Cognitive Approaches," 2006.

[13] M. A. E. B. A. A. S. R. &. A. A. Ozdemir, "Realtime emotion recognition from facial expressions using cnn architecture," 2019.

[14] S. R. A. N. M. B. H. a. A. M. Yahya, "Comparison of Convolutional Neural Network Architectures for Face Mask Detection. International Journal of Advanced Computer Science and Applications," 2021.

[15] A. A.-J. J. a. S. J. Rodríguez-Fuertes, "Measuring the candidates' emotions in political debates based on," 2022.

[16] G. Boesch, "VGG Very Deep Convolutional Networks (VGGNet)–What you need to know.," 2021.

[17] J. Brownlee, "Deep learning with Python: develop deep learning models on Theano and TensorFlow using Keras.," 2016.

[18] Simonyan, K., & Zisserman, A, "Very deep convolutional networks for large-scale image recognition," 2014.

[19] T. A. ResNet-50, "towardsdatascience," 2022. [Online]. Available: https://towardsdatascience.com/the-annotated-resnet-50-a6c536034758. [Accessed 23 01 2023].

[20] Yuesheng, F., Jian, S., Fuxiang, X., Yang, B., Xiang, Z., Peng, G., ... & Shengqiao, X, " Circular fruit and vegetable classification based on optimized GoogLeNet," 2021.

[21] Jasitha, P., Dileep, M. R., & Divya, M, "Venation based plant leaves classification using GoogLeNet and VGG," 2019.

[22] Lin, C., Li, Y., Liu, H., Huang, Q., Li, Y., & Cai, Q., "Power Enterprise Asset Estimation Algorithm Based on Improved GoogLeNet," 2020.

[23] Ketwongsa, W., Boonlue, S., & Kokaew, U, "A new deep learning model for the classification of poisonous and edible mushrooms based on improved alexnet convolutional neural network," 2022.

[24] Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., ... & Rabinovich, A., " Going deeper with convolutions," 2015.

[25] M. A. Islam, "Comparative Analysis of Pre-Trained," 2023.

[26] Chauhan, N. K., & Singh, K, "A review on conventional machine learning vs deep learning," Vols. In 2018 International conference on computing, power and communication technologies , 2018.

[27] N. J. S. N. V. G. S. J. P. a. B. P. Gupta, ".Comparison of fluoride ion release and alkalizing potential of a new bulk-fill alkasite," 2019.

[28] S., & Chakraborty, A., Sarkar, C, Sharma, " A Novel Multi-Task Learning Framework for Facial Emotion Recognition." vol. IEEE Transactions, 2020.

[29] A. C. S. S. S. S. A. K. D. &. S. R. Bhattacharyya, "A deep learning model for classifying human facial expressions from infrared thermal images.," *Scientific reports,* vol. 1, p. 20696, 2021.

[30] L. V. F. S. E. V. P. D. S. M. &. L. C. P. Lévêque, "Evaluating the facial expression recognition robustness of deep neural networks and humans," *Electronics,* p. 4030, 2022.

[31] M. A., Papaioannou, Zafeiriou, S., Kollias, D., Nicolaou, A., Zhao, G., &, " Aff-Wild: The Challenge of Valence and Arousal in the Wild." 2017.

[32] J. &. G. T. Buolamwini, "Gender shades: Intersectional accuracy disparities in commercial gender classification.," no. In Conference on fairness, accountability and transparency.

[33] Z. C. Lipton, "The Mythos of Model Interpretability," pp. 31-57, 2016.

[34] S. Q. L. Q. H. S. J. &. J. J. Liu, "Path aggregation network for instance segmentation.," *In Proceedings of the IEEE conference on computer vision and pattern recognition,* pp. 8759-8768, 2018.

[35] KerasNLP, "keras," 16 06 2021. [Online]. Available: https://keras.io/guides/. [Accessed 16 06 2023].

[36] "Keras Applications," [Online]. Available: https://keras.io/api/applications/. [Accessed 17 1 2024].

[37] A. a. H. G. Krizhevsky, "ImageNet Classification with Deep," 2012.

[38] J. a. A. B. Fernández Villaverde, "An empirical comparison of seven," 2014.

[39] RGB, W, "Why OpenCV Using BGR Colour Space Instead of RGB," Stackoverflow.com, 2016. [Online]. Available: : http://stackoverflow.com/questions/14556545/whyopencv-using-bgr-colour-space-instead-of-rgb . [Accessed 16 06 2023].

[40] X. Z. L. a. J. by Fan Sun, "Multi-Object Detection in Security Screening Scene Based on Convolutional Neural Network," 2022.