National College of Ireland

# A Comparative Analysis for Recognizing Emotions from Facial Expressions

MSc Research Project
Artificial Intelligence

Noel Viji Thaliath
Student ID: x22185178

School of Computing
National College of Ireland

Supervisor: Dr. Muslim Jameel Syed

**National College of Ireland**
**Project Submission Sheet**
**School of Computing**

| | |
|---|---|
| **Student Name:** | Noel Viji Thaliath |
| **Student ID:** | x22185178 |
| **Programme:** | Artificial Intelligence |
| **Year:** | 2023 |
| **Module:** | MSc Research Project |
| **Supervisor:** | Dr. Muslim Jameel Syed |
| **Submission Due Date:** | 14/12/2023 |
| **Project Title:** | A Comparative Analysis for Recognizing Emotions from Facial Expressions |
| **Word Count:** | 4283 |
| **Page Count:** | 20 |

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

**ALL** internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

| | |
|---|---|
| **Signature:** | |
| **Date:** | 14th January 2024 |

**PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST:**

| | |
|---|---|
| Attach a completed copy of this sheet to each project (including multiple copies). | ☐ |
| **Attach a Moodle submission receipt of the online project submission**, to each project (including multiple copies). | ☐ |
| **You must ensure that you retain a HARD COPY of the project**, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer. | ☐ |

Assignments that are submitted to the Programme Coordinator office must be placed into the assignment box located outside the office.

| **Office Use Only** | |
|---|---|
| Signature: | |
| Date: | |
| Penalty Applied (if applicable): | |

# A Comparative Analysis for Recognizing Emotions from Facial Expressions

Noel Viji Thaliath
x22185178

**Abstract**

Emotions, categorized into anger, disgust, fear, gladness, neutrality, sadness, and surprise, significantly influence judgments and discussions on various issues. Deep learning, an artificial intelligence technique, can mimic the human brain's data analysis to identify patterns for judgments. It uses networks to comprehend unsupervised, unstructured or unlabeled data, surpassing machine learning when dealing with large amounts of data. Unlike traditional programs, which examine data in a linear fashion, deep learning systems use a hierarchical function to handle data in a nonlinear fashion. For this research, the models developed for experimentation is a CNN model, a hybrid CNN-LSTM model, and VGG-16 model. The overall performed model was the hybrid CNN-LSTM model which gave an accuracy of 42% for a balanced dataset and 62% for unbalanced dataset. The model that performed best with a high accuracy was the CNN model that gave up to 62% in testing but gave a very high 99% accuracy during its training phase.

***Keywords***— CNN, Deep Learning, Machine Learning, LSTM, VGG-16, OpenCV, FER, Artificial Intelligence

# 1 Introduction

Detecting an individual's emotional state has gained significant attention in recent years. Doctors and psychiatrists can assess a patient's mental state by observing facial expressions, posture, and conduct. Emotions significantly influence our judgments, thinking, attention, prosperity, and quality of life. Communication is facilitated by emotions and facial expressions, and the need to understand the full range of emotions has emerged in psychology and mental health, (Zadeh, et al., 2019).

## 1.1 Background

Facial Expression Recognition (FER) is a computer vision challenge for detecting and categorizing emotional emotions on the human face. The objective is to automate the process of identifying emotions in real time by analyzing the numerous features of the face. Using a webcam, we can take live shots of a person and utilize them as input images to discern emotions and determine their mental state using Computer Vision algorithms. Computer vision is the science of teaching computers to recognize and interpret objects and people in photographs and films. The doctor in the psychological field is seeking to comprehend the emerging elements of brains by linking the discipline to neuroscience. Psychologist's purpose is to understand the behavior of individuals or groups.

OpenCV is an open-source library that uses image and video processing components for various applications such as human emotion recognition, photo retouching, vehicle license plate identification, optical character recognition, and robotic vision.

## 1.2 Motivation

Many experts have conducted trials and investigations to predict outcomes in emotion identification systems, which are divided into two types: those that recognize emotions in still images and those that use snapshots. This research aids doctors in diagnosing a person's mental condition and providing necessary care and therapy.

# 2 Related Work

Facial Emotion Recognition (FER) is a significant field in computer vision, with numerous studies requiring systematic review. These include feature-based face recognition Extrusion methods, which can be hand-drawn features or feature releases using deep neural networks.

Convolution Neural Network models are almost universally used by academics to evaluate an individual's face mood from 2018 to 2022. The techniques' accuracy ranged between 80% and 90%. Emotion predictions can be achieved using two or four layers of CNN. Convolution is the initial step in extracting image characteristics, focusing on visual aspects among pixels with small input squares. It uses an image matrix and kernel. If images are large, the pooling layer restricts network parameters and processing. It collects properties from the immediate layer's feature map.

## 2.1 Facial Emotion Detection Using Convolutional Neural Networks

The paper by (Zadeh et al.; 2019) presents a deep learning-based framework for human emotion detection, which uses Gabor filters to extract features and classify them using a Convolutional Neural Network (CNN). The approach improves the speed and accuracy of CNN training. The method involves texture analysis, edge detection, and feature extraction, with the best results obtained at the edges and texture changes. The filtered image is sent into an AI model with a CNN architecture, which produces a vector with seven categories after several layers and learning processes. The deep neural network in the image uses a 6x6 filter convolution, then reduces dimensions to 128x128x6 using MaxPooling. A 16x16 filter size is applied, and a 120x120 filter size is used for convolving the data. Following that, the Flatten function is used to turn all of the data into a vector of size 432000. The vector is then decreased to seven, indicating the seven different types of emotional states, and converted into an 84-length vector. They employed the JAFFE database, which comprises 213 Japanese female model photos depicting seven emotional states of the face, including six modes, natural states of the face, and normal face. Their findings reveal that after 10 epochs, the recommended strategy achieves 86% accuracy, while the typical CNN method achieves 51%. After 15 epochs, the proposed approach achieves 92% accuracy, compared to 73% for regular CNN. After 25 epochs, the recommended approach achieves 97% accuracy, whereas its rival reaches 90%. In the end, the recommended strategy achieves 97% accuracy whereas the other achieves 91% accuracy.

This study explores the use of Convolutional Neural Network (CNN) to detect emotions based on facial expressions in images. The CNN model was developed and evaluated on three datasets: Facial Expression Recognition 2013 (FER2013), Cohn-Kanade Dataset (CK+), and Karolinska Directed Emotional Faces (KDEF). The datasets were sorted into seven emotion

groups, and the CNN architecture was built with four convolution layers and two totally connected layers. The data was then separated for training and testing, and the model was trained for emotion detection. The KDEF dataset was the best, with an accuracy of 0.82, precision of 0.84, recall of 0.82, and F1-score of 0.81. The most easily identifiable emotion classifications are disgust and happiness, while sad is the most difficult to recognize. The most commonly identified emotion in the FER2013 dataset is Happy, but not all pictures with the Happy label may be classified as Happy emotions. Images not classified as Happy are generally classified as Neutral and Sad emotions. In the CK+ and KDEF datasets, all photographs in the Happy emotion class can be recognized, while only one image in the Happy emotion class cannot be recognized. Sad is the most commonly misidentified emotion class in the CK+ and KDEF datasets, with most photographs in the Sad class identified as Angry images.

The researchers in Zhou et al. (2020) developed a lightweight CNN to detect facial emotions in real-time scenarios. They used multi-task cascaded convolutional networks (MTCNN) for face detection and segmented face coordinates for their facial emotions classification model. The model was tested on the FER-2013 dataset and the WIDER FACE dataset, which contains 32203 pictures and 393,703 faces. The WIDER FACE dataset is a benchmark dataset for face detection, with 61 event categories chosen by the Chinese University of Hong Kong. The sentiment classification achieved a 67% accuracy on the FER dataset, and the model architecture's weights were stored in an 872.9KB file. The true positive rate reached up to 95% when the other dataset was retrained on the MTCNN model.

In Vasantha et al. (2022) proposes applying algorithms such as Decision trees, Random Forest, ANN for detecting the mental health using certain facial emotion factors. They suggest to get the input data from any webcams or if someone uploads a facial image of the respected individual.

The research Bhatia et al. (2022) presents a customizable facial recognition model based on behavioral elements of a map combined with biometric visual information. The device uses geometric structures to match the identification system's template, but faces can be complex and require constant facial expressions. The emotion recognition device offers a flexible facial recognition version based on behavioral operations of a map with biometric visible capabilities.

## 2.2 Group Facial Emotion Detection using CNN

The study Kousalya et al. (2023) analyzed seven major human emotions in a group photo to predict the situation. The model uses dlib's HOG + Linear SVM approach to extract faces from the image and predict the environment. The input pictures are retrieved from the group image, and the model uses a 48x48 gray-scale image with 32 filters to generate a 48x48x32x1 matrix. The input matrix is then reduced to 24x24x64x1 before being randomly removed. The input matrix is then sent to the max-pooling layer, where it is reduced to 12x12x128x1 and then 12x12x512x1 before being reduced to 12x12x128x1. The input matrix is then converted into a 1D array, and the dense layer has 1024 neurons with ReLU activation functions. The signal is sent to the dropout layer, which causes half of the neurons to die. The information is sent to the dense layer, which has a Softmax activation function and outputs 7 neurons. The CNN model is used to save and test images, and the situation is forecasted using the class with the highest likelihood depending on the type of emotion shown in each image. The model achieves training accuracy of 89.66% and validation accuracy of 83.83%, surpassing the other four optimizers in all seven categories.

## 2.3 Facial Emotion Detection using an Emotion Detection Tool with Eye Gesture support

The study by Kumari and Deshmukh (2023) presents a Python and OpenCV-based tool that tracks eyeball movements and detects emotions while driving. The tool generates the driver's emotional state throughout their journey. The first step involves determining the user's face from a webcam using the 68 face landmark file, which is highly accurate and works even in low light conditions. The next step is detecting the eye position and analyzing the face to detect the driver's emotion. The model can detect six emotions: angry, fear, happy, neutral, sad, and surprise. The test was conducted on the proposed model and compared with four other models, resulting in an accuracy of nearly 70

## 2.4 Facial Emotion Detection using YOLOv7, Faster R-CNN and SSD

The study compares the performance of YOLOv7 for facial expression recognition in Thai elderly adults (Khajontantichaikun et al.; 2023), comparing it to Faster R-CNN and SSD models. The models were trained and tested using a dataset of 900 face photos of Thai seniors. YOLOv7 outperformed the other models with a mean average accuracy of 0.95, while Faster R-CNN and SSD had mean average precision of 0.86 and 0.84, respectively. To avoid over-fitting, picture preprocessing and data augmentation techniques were used. The study employed faster R-CNN ResNet101 V1 640x640, SSD ResNet101 V1 FPN 640x640, and YoLOv7 models for learning from labeled images after preprocessing and data augmentation. According to the test findings, YOLOv7 has the best mAP performance (0.95), followed by Faster R-CNN (0.87) and SSD (0.84).

# 3 Methodology

In this section, we discuss the data collection, preprocessing the image dataset and transformations, and the models that are used for conducting the research for this analysis. The aim of this section is to give a detailed description of the research design and methods. We have compared the proposed CNN model against the hybrid CNN-LSTM model and the VGG-16 model. Since the image database has unbalanced classes in it, we have also created a balanced version of this dataset, and tested them against both balanced and unbalanced classes.

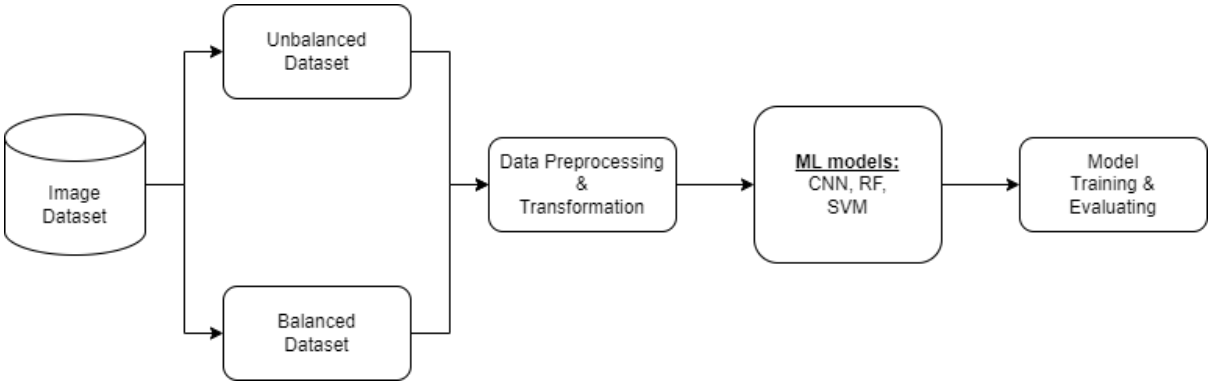



Figure 1: Research Process Flow

## 3.1 Data Collection

The image dataset was taken from Shah (2020). The dataset uses png file format. The data consists of seven classes, anger, disgust, fear, happiness, neutral, sadness, and surprise. Since the classes in the dataset are unbalanced, a subset of this dataset with balanced classes are additionally created and the experiments are performed on them and compared with the unbalanced class.

## 3.2 Data Pre-processing and Transforming

In this stage we perform data preprocessing where the balanced and unbalanced classes are read separately and then we re-scale and transform the images to a consistent size, which can improve the performance of certain algorithms. Re-scaling an image involves changing its dimensions, usually by scaling it up or down, while maintaining its aspect ratio. We use the ImageDataGenerator() function to load in the images, train, and then perform data augmentation on image data.

## 3.3 Modelling Algorithms Used

For this research thesis, I have used and analysed three main models. A 4-layer CNN model, a 5-layer CNN-LSTM hybrid model and a VGG-16 network model. Later we compare their performance against balanced and unbalanced classes on the input dataset.

### 3.3.1 CNN model

A CNN (Convolutional Neural Network) model is a deep learning approach that is often used for picture classification and identification. It is inspired by the way neurons respond to particular portions of the visual field in animals' visual cortex. CNN models are made up of layers such as convolutional layers, pooling layers, and fully connected layers. Convolutional layers extract features like edges, textures, and forms from images using filters. Pooling layers sample feature maps, reducing spatial dimensions, and completely linked layers classify collected characteristics.

CNN models have excelled in a variety of computer vision applications, including object identification, facial recognition, and picture segmentation. They have also been used to other fields, such as natural language processing and speech recognition, by tailoring the architecture to the task at hand.

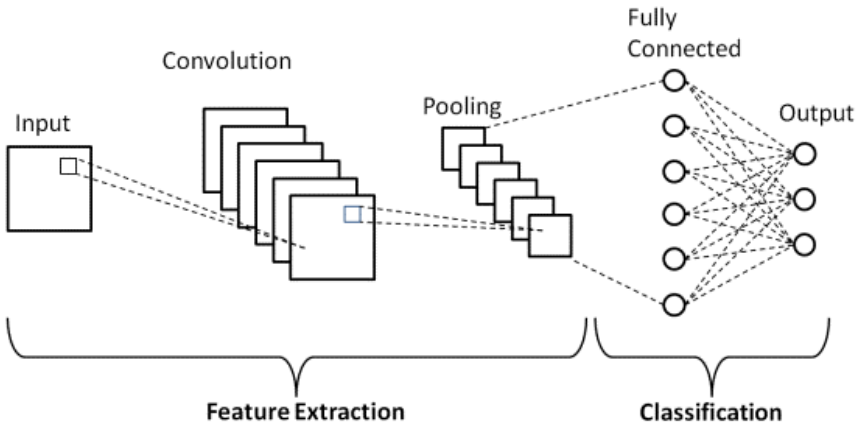



Figure 2: CNN Model

### 3.3.2 LSTM Model

Long Short-Term Memory (LSTM) is a form of recurrent neural network (RNN) architecture widely used for processing sequential data, such as time series data or natural language text. Unlike standard RNNs, LSTMs may choose forget or recall information from past time steps, making them more efficient and effective at dealing with long-term data dependencies.

The input data in an LSTM model is fed into a memory cell, which saves the information and utilizes it to create an output. There are three gates in the memory cell: an input gate, an output gate, and a forget gate. The input gate decides what fresh information to add to the memory cell, the output gate decides what to output, and the forget gate decides what to delete. This selective memory mechanism enables LSTMs to keep a more lasting recollection of prior events while processing new information.

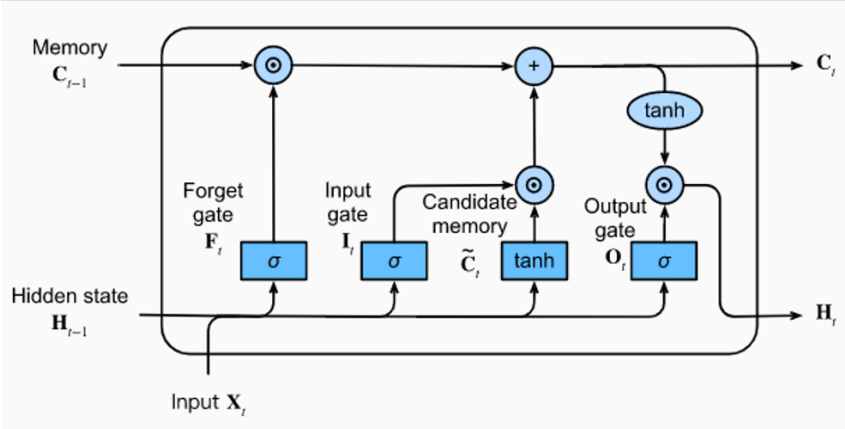



Figure 3: A cell of the LSTM model

LSTM models are made up of an input layer, one or more LSTM layers, and an output layer. The input layer receives the input data, and each succeeding LSTM layer processes the output of the preceding layer. To obtain the final output, the output of the final LSTM layer is passed via a fully connected layer and a soft-max activation function.

### 3.3.3 VGG-16 model

VGG-16 is a 16-layer convolutional neural network (CNN) architecture developed by the Visual Geometry Group at the University of Oxford (Simonyan and Zisserman; 2014). It was a top performer in the 2014 ImageNet Large Scale Visual Recognition Challenge. The model consists of 13 convolutional layers, three fully linked layers, and five maximum pooling layers. The convolutional layers use small filters (3x3) with a stride of one and padding of one.

One of the fundamental enhancement of the VGG-16 model is the employment of tiny filters of uniform size across the network. This enables the network to learn more complicated features by stacking many convolutional layers with tiny filters rather than utilizing bigger filters with a greater receptive field. The VGG-16 model has been widely utilized as a baseline architecture for image classification tasks, as well as a feature extractor for additional computer vision applications such as object recognition and semantic segmentation. It is accessible in many deep learning frameworks, including TensorFlow and PyTorch, and may be fine-tuned for individual tasks or datasets.

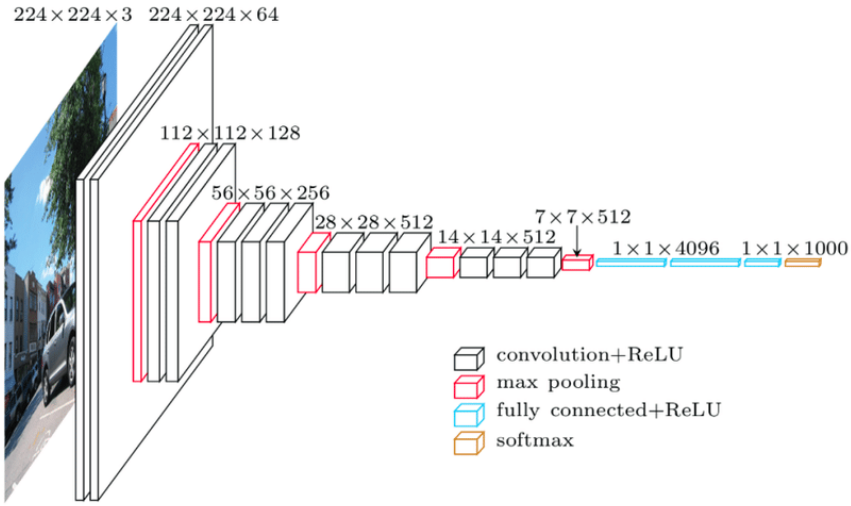



Figure 4: VGG-16 Network Model

# 4 Design Specification

This section explains the system that we have implemented for this research which is shown in Figure 5, and would discuss each phase in detail.

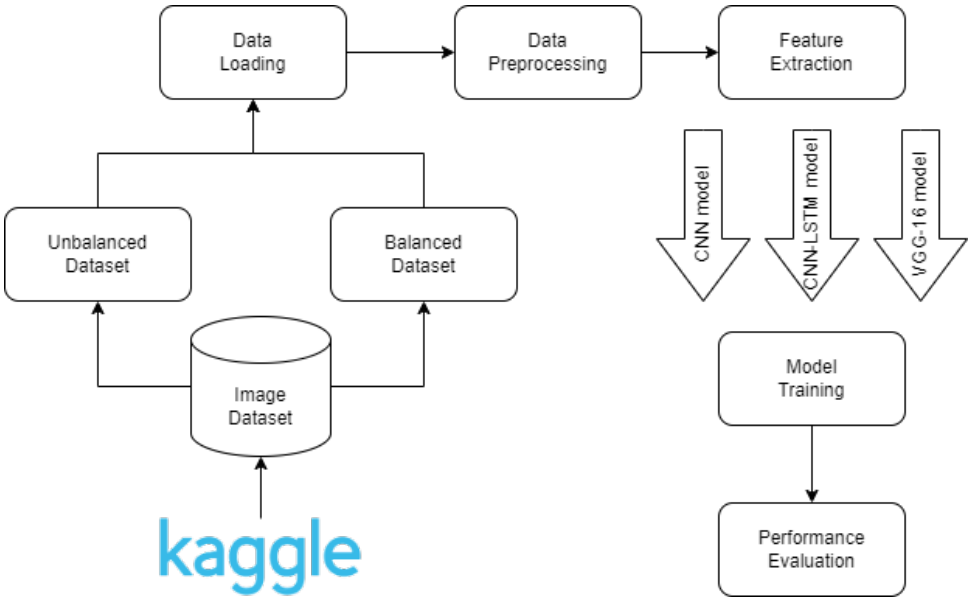



Figure 5: Project Architecture

The dataset was originally downloaded from Kaggle's website Shah (2020). The dataset obtained from this location has unbalanced classes, so for the experiment's purpose we created a balanced class and ran all the models with both balanced and unbalanced classes. The images consists of seven basic human emotions: anger, disgust, fear, happiness, neutral, sadness, and surprise, which are used as labels for classifying these emotions by the implemented models. The next step is splitting the train dataset into 70-30 ratio for training and validation, and will use the test dataset for testing the model's performance. The models are trained with two different epochs, mainly 50 and 100. The evaluation was done using confusion matrix, classification report, accuracy, and ROC scores.

# 5 Implementation

This section deals with the development of all the three machine learning models that I have tried while implementing the research.

## 5.1 CNN Model

The implemented CNN model consists of four interconnected conv2d layers along with batch normalization, activation, max pooling layers. Before the next convolutional layer gets connected to the model a 20% dropout of neurons present in the current layer. A dropout layer is a sort of neural network layer used to avoid over-fitting. When a model grows too complicated, it begins to remember the training data rather than learning generalizable patterns. As a result, the model would perform well on training data but it would be worse on fresh, unknown data. During training, a dropout layer would randomly set a percentage of the neurons in the layer to zero. Since different neurons are employed during various training iterations, the model is forced to learn numerous representations of the data. This prevents the model from being overly reliant on any particular neuron or set of neurons, and pushes it to discover more generalizable patterns in the data.

The enter output generated by the third dropout layer is then flattened and then passes through a series of densely connected layers, until the final dense layer generates an output of 7 units which corresponds to the seven classified outputs. The structure of the CNN model is shown in Figure 6.

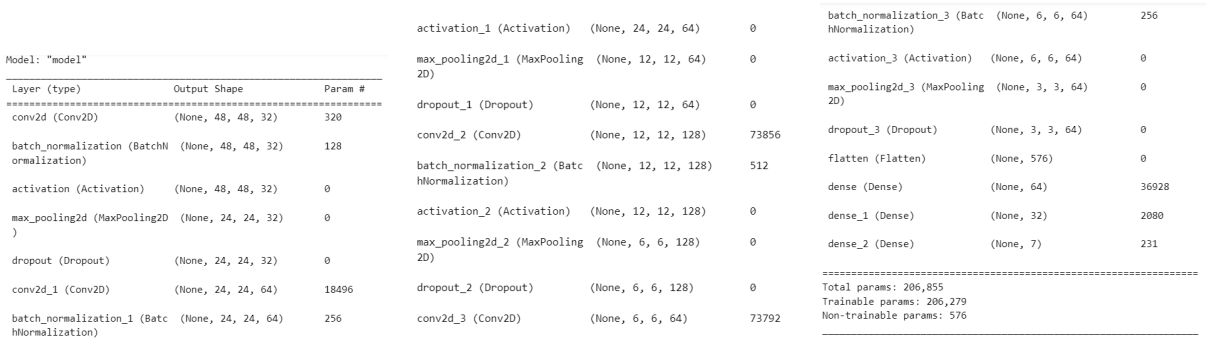



Figure 6: CNN Architecture

## 5.2 CNN-LSTM Model

The hybrid CNN-LSTM model has of five block interconnected convolutional-2d layer followed by batch normalization and activation. In this hybrid model the max pooling layers are attached to the second, fourth and the final convolutional layer. After this layer the output layer from the CNN layer is reshaped from a 2-D format of (None, 4, 4, 128) to a single dimension of (None, 16, 128). This is then given as the input to two LSTM layers of 128 cells followed by 64 cells. The output of this layer then moves on to a series of a densely connected layers which then terminates into the final output layer of seven predicted classes of emotions. The structure of of the hybrid CNN-LSTM model is shown in Figure 7.

| Layer (type) | Output Shape | Param # |
|---|---|---|
| conv2d (Conv2D) | (None, 46, 46, 32) | 320 |
| batch_normalization (BatchNormalization) | (None, 46, 46, 32) | 128 |
| activation (Activation) | (None, 46, 46, 32) | 0 |
| conv2d_1 (Conv2D) | (None, 46, 46, 64) | 18496 |
| batch_normalization_1 (BatchNormalization) | (None, 46, 46, 64) | 256 |
| activation_1 (Activation) | (None, 46, 46, 64) | 0 |
| max_pooling2d (MaxPooling2D) | (None, 23, 23, 64) | 0 |
| conv2d_2 (Conv2D) | (None, 21, 21, 64) | 36928 |
| batch_normalization_2 (BatchNormalization) | (None, 21, 21, 64) | 256 |
| activation_2 (Activation) | (None, 21, 21, 64) | 0 |
| conv2d_3 (Conv2D) | (None, 21, 21, 128) | 73856 |
| batch_normalization_3 (BatchNormalization) | (None, 21, 21, 128) | 512 |
| activation_3 (Activation) | (None, 21, 21, 128) | 0 |
| max_pooling2d_1 (MaxPooling2D) | (None, 10, 10, 128) | 0 |
| conv2d_4 (Conv2D) | (None, 8, 8, 128) | 147584 |
| batch_normalization_4 (BatchNormalization) | (None, 8, 8, 128) | 512 |
| activation_4 (Activation) | (None, 8, 8, 128) | 0 |
| max_pooling2d_2 (MaxPooling2D) | (None, 4, 4, 128) | 0 |
| reshape (Reshape) | (None, 16, 128) | 0 |
| lstm (LSTM) | (None, 128) | 131584 |
| reshape_1 (Reshape) | (None, 2, 64) | 0 |
| lstm_1 (LSTM) | (None, 64) | 33024 |
| dense (Dense) | (None, 200) | 13000 |
| dropout (Dropout) | (None, 200) | 0 |
| dense_1 (Dense) | (None, 7) | 1407 |

Total params: 457,863
Trainable params: 457,031
Non-trainable params: 832

Figure 7: CNN-LSTM Hybrid Architecture

## 5.3 VGG-16 Model

The base model of this model would be taken along with the normal weights of imagenet dataset. After the model is loaded, then we freeze all the layers of this network and we add the facial image dataset as the modified input layer, and then we re-training the model on it. The VGG-16 model, with 138 million parameters, consists of 13 convolutional layers and 3 fully linked layers. Convolutional layers extract features from input pictures, while fully linked layers classify. The architecture diagram in Figure 8 illustrates the model's structure that was used in Simonyan and Zisserman (2014).

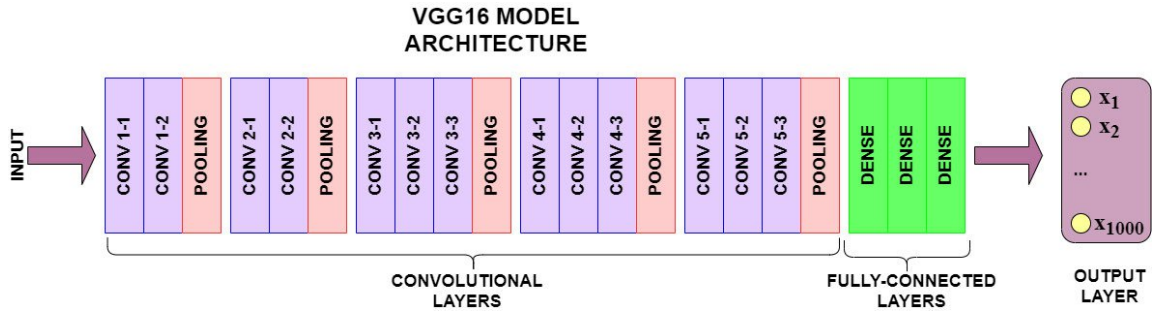



Figure 8: VGG-16 Architecture

# 6 Evaluation and Discussion

This section provides us with the critical assessment for the performance of the three different machine learning models that I have used for classifying the facial emotions to seven categories that are anger, disgust, fear, happiness, neutral, sadness and surprise. For analyzing the model's performance I used the the metrics given below:

- Accuracy: Accuracy is defined as the ratio of the number of correct predictions to that of the total number of predictions that can be made.
- Precision: It is a metric that is being used to evaluate the performance of a classification task. It can be defined as the ratio of true positive predictions to that of the total number of the positive predictions.
- Recall: It can be defined as the ratio of true positive predictions against the total number of actual positive cases that are present in the dataset.
- F1 score: It is the harmonic mean of precision and recall and would lie between 0 and 1. A higher value indicates that the model has a better performance score.

- Classification Report: It is tool that generates a summary of the performance of a machine learning model when it is used to perform classification tasks. It generally includes several evaluation metrics like precision, recall, and F1 score for each class in the dataset.
- ROC Score: It can be defined as the trade-off between the true positive rate and false positive rate at different thresholds.
- Confusion Matrix: It summarizes the predictions made by the trained model against that of the actual true labels of the given data, thereby providing a way to assess the accuracy and confusion of the given model.

In the experiment phase we used a 4-layer CNN model, a hybrid CNN-LSTM model, and the VGG-16 model and executed them against both balanced and unbalanced classes. We also ran and analyzed them against various epochs: 50 and 100 epochs.

## 6.1 Experiment 1: CNN model

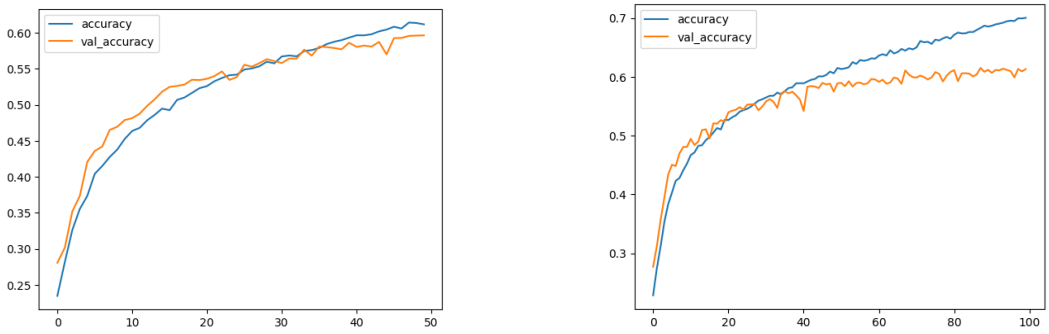
The following are the results for the experiment when used on unbalanced classes followed by the performance graphs and confusion metrics below:

| Evaluation Parameters | | |
|---|---|---|
| No. of Epochs | Accuracy | ROC Score |
| 50 | 60.60% | 0.88635 |
| 100 | 61.58% | 0.89251 |

Table 1: Performance for Unbalanced classes

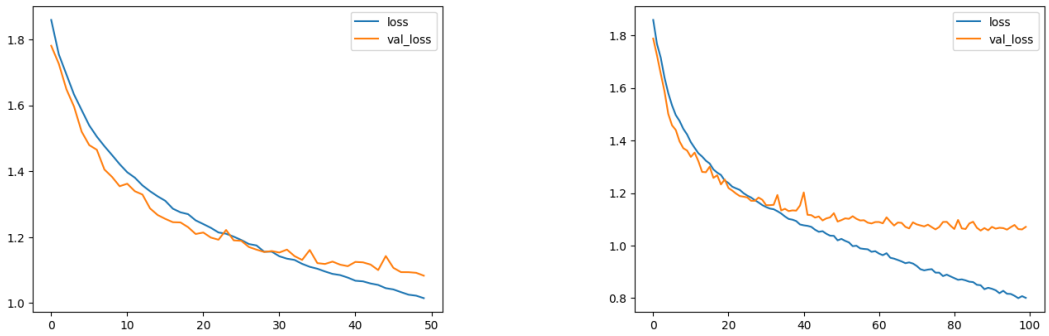
| Classes | Precision | | Recall | | F1 Score | |
|---|---|---|---|---|---|---|
| | 50 epochs | 100 epochs | 50 epochs | 100 epochs | 50 epochs | 100 epochs |
| 0 | 0.55 | 0.53 | 0.54 | 0.50 | 0.54 | 0.51 |
| 1 | 0.70 | 0.58 | 0.35 | 0.33 | 0.46 | 0.42 |
| 2 | 0.46 | 0.50 | 0.28 | 0.32 | 0.35 | 0.36 |
| 3 | 0.77 | 0.86 | 0.87 | 0.82 | 0.82 | 0.84 |
| 4 | 0.54 | 0.55 | 0.63 | 0.64 | 0.58 | 0.59 |
| 5 | 0.46 | 0.44 | 0.48 | 0.56 | 0.47 | 0.49 |
| 6 | 0.72 | 0.75 | 0.73 | 0.78 | 0.73 | 0.76 |

Table 2: Metrics for model with 50 and 100 epochs respectively



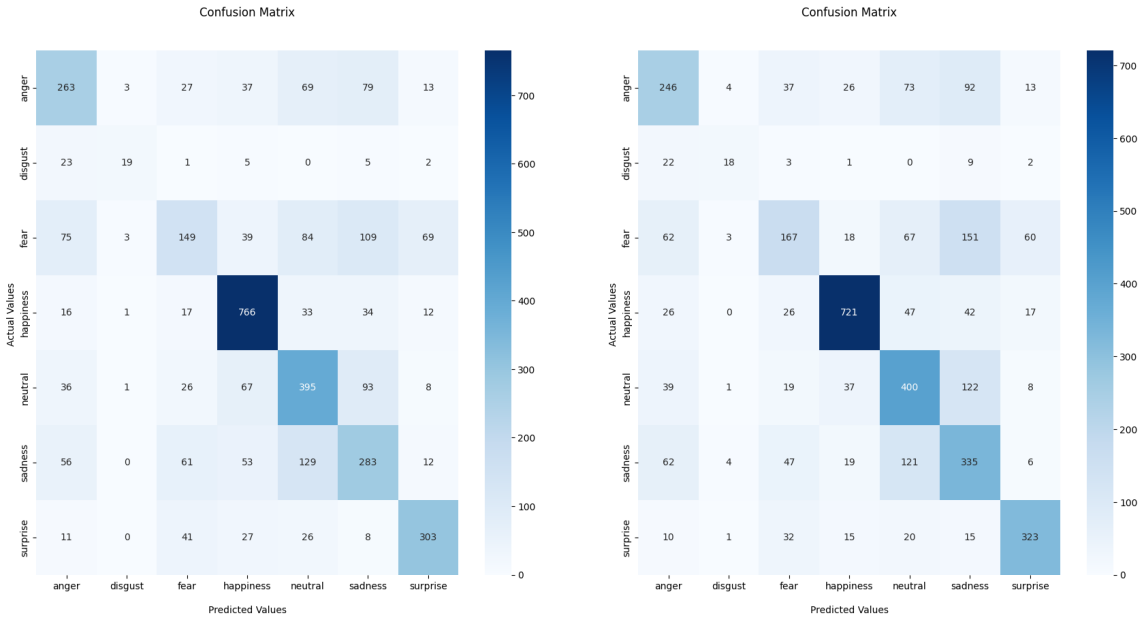

(a) 50 epochs (b) 100 epochs

Figure 9: Accuracy of Unbalanced Classes




(a) 50 epochs (b) 100 epochs

Figure 10: Loss of Unbalanced Classes




(a) 50 epochs (b) 100 epochs

Figure 11: Confusion matrix of Unbalanced Classes

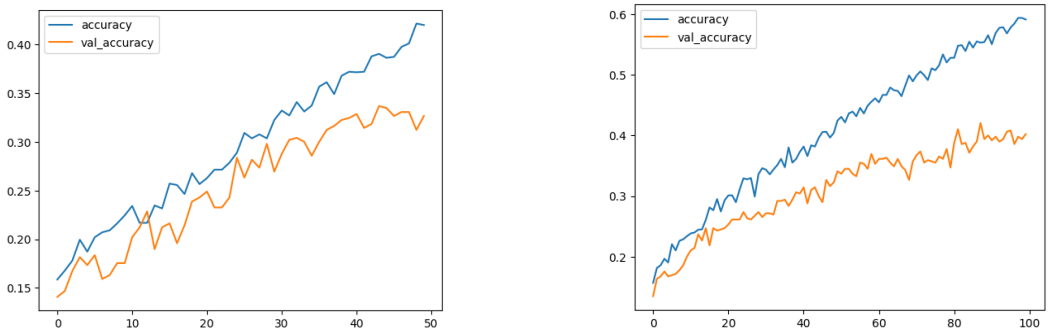
The following are the results for the experiment when used on balanced classes followed by the performance graphs and confusion metrics below:

| Evaluation Parameters | | |
|---|---|---|
| No. of Epochs | Accuracy | ROC Score |
| 50 | 35.87% | 0.72335 |
| 100 | 42.06% | 0.77907 |

Table 3: Performance for Balanced classes

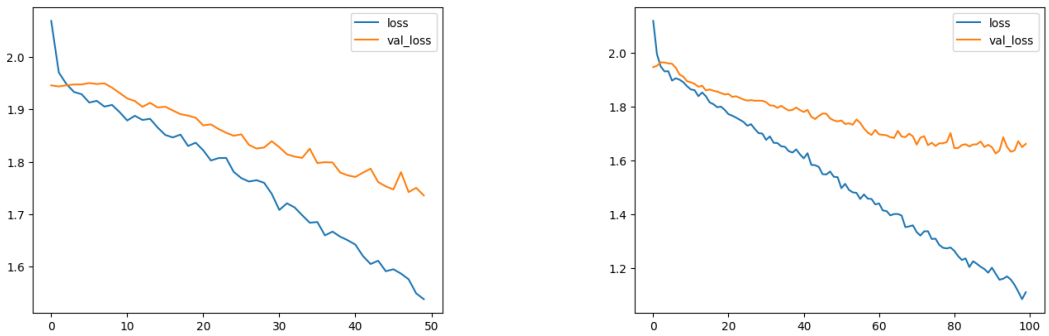
| Classes | Precision | | Recall | | F1 Score | |
|---|---|---|---|---|---|---|
| | 50 epochs | 100 epochs | 50 epochs | 100 epochs | 50 epochs | 100 epochs |
| 0 | 0.33 | 0.23 | 0.12 | 0.12 | 0.17 | 0.16 |
| 1 | 0.41 | 0.44 | 0.41 | 0.68 | 0.41 | 0.53 |
| 2 | 0.43 | 0.37 | 0.11 | 0.20 | 0.18 | 0.26 |
| 3 | 0.31 | 0.42 | 0.68 | 0.64 | 0.42 | 0.51 |
| 4 | 0.31 | 0.39 | 0.38 | 0.41 | 0.34 | 0.40 |
| 5 | 0.22 | 0.26 | 0.17 | 0.18 | 0.19 | 0.21 |
| 6 | 0.53 | 0.63 | 0.67 | 0.72 | 0.60 | 0.68 |

Table 4: Metrics for model with 50 and 100 epochs respectively



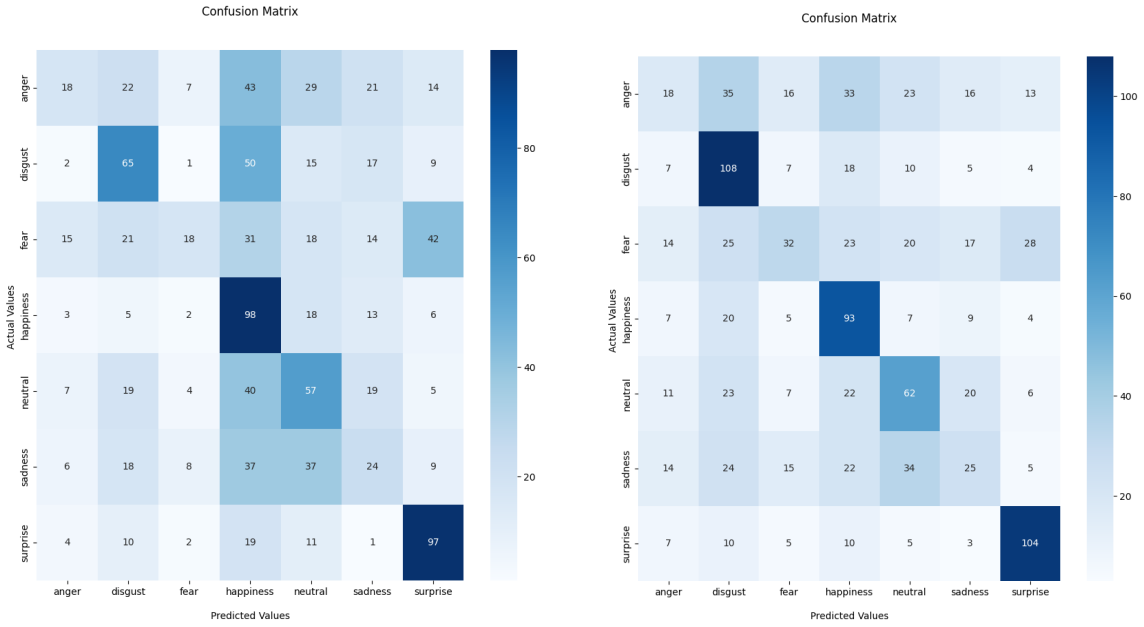

(a) 50 epochs (b) 100 epochs

Figure 12: Accuracy of Balanced Classes




(a) 50 epochs (b) 100 epochs

Figure 13: Loss of Balanced Classes




(a) 50 epochs (b) 100 epochs

Figure 14: Confusion matrix of Balanced Classes

## 6.2 Experiment 2: Hybrid CNN-LSTM model

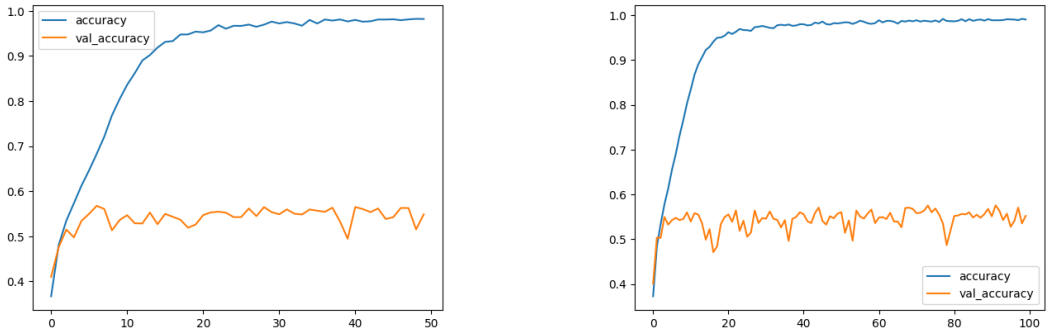
The following are the results for the experiment when used on unbalanced classes followed by the performance graphs and confusion metrics below:

| Evaluation Parameters | | |
|---|---|---|
| No. of Epochs | Accuracy | ROC Score |
| 50 | 55.59% | 0.84927 |
| 100 | 54.19% | 0.84060 |

Table 5: Performance for Unbalanced classes

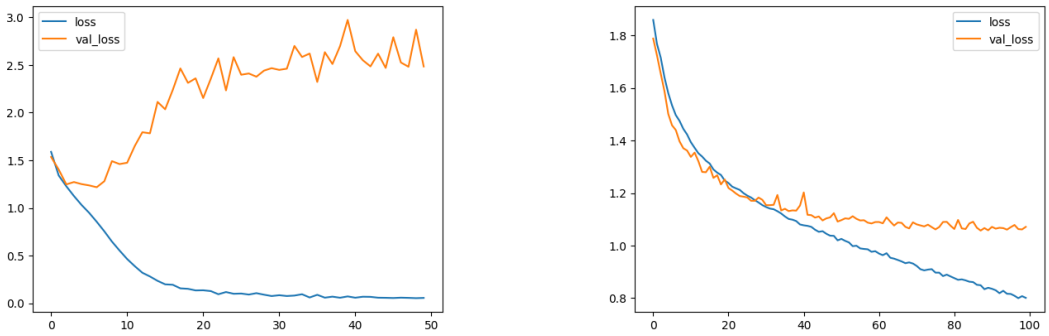
| Classes | Precision | | Recall | | F1 Score | |
|---|---|---|---|---|---|---|
| | 50 epochs | 100 epochs | 50 epochs | 100 epochs | 50 epochs | 100 epochs |
| 0 | 0.51 | 0.47 | 0.44 | 0.43 | 0.47 | 0.45 |
| 1 | 0.54 | 0.61 | 0.35 | 0.36 | 0.42 | 0.45 |
| 2 | 0.37 | 0.49 | 0.36 | 0.21 | 0.36 | 0.30 |
| 3 | 0.77 | 0.77 | 0.78 | 0.75 | 0.78 | 0.76 |
| 4 | 0.47 | 0.42 | 0.59 | 0.69 | 0.53 | 0.52 |
| 5 | 0.45 | 0.39 | 0.31 | 0.38 | 0.37 | 0.38 |
| 6 | 0.61 | 0.73 | 0.78 | 0.68 | 0.68 | 0.70 |

Table 6: Metrics for model with 50 and 100 epochs respectively



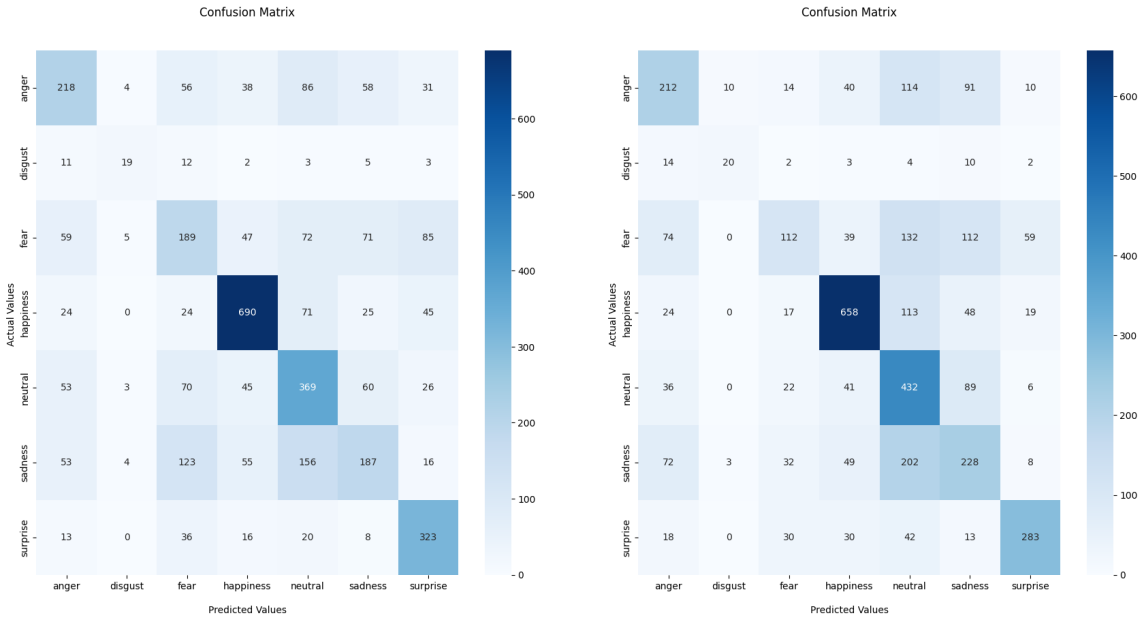

(a) 50 epochs (b) 100 epochs

Figure 15: Accuracy of Unbalanced Classes




(a) 50 epochs (b) 100 epochs

Figure 16: Loss of Unbalanced Classes




(a) 50 epochs (b) 100 epochs

Figure 17: Confusion matrix of Unbalanced Classes

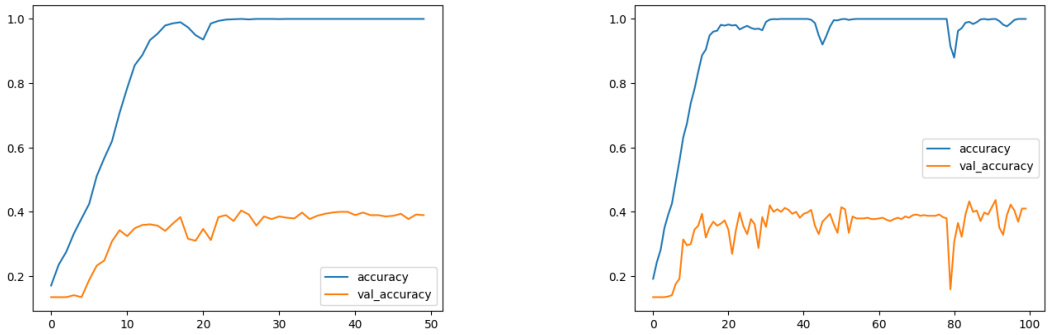
The following are the results for the experiment when used on balanced classes followed by the performance graphs and confusion metrics below:

| Evaluation Parameters | | |
|---|---|---|
| No. of Epochs | Accuracy | ROC Score |
| 50 | 40.76% | 0.76334 |
| 100 | 40.85% | 0.76141 |

Table 7: Performance for Balanced classes

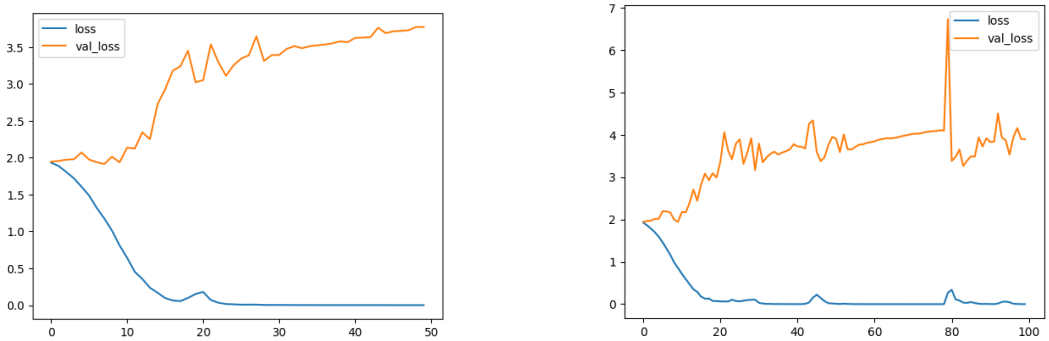
| Classes | Precision | | Recall | | F1 Score | |
|---|---|---|---|---|---|---|
| | 50 epochs | 100 epochs | 50 epochs | 100 epochs | 50 epochs | 100 epochs |
| 0 | 0.34 | 0.33 | 0.28 | 0.24 | 0.31 | 0.28 |
| 1 | 0.13 | 0.13 | 0.55 | 0.58 | 0.21 | 0.21 |
| 2 | 0.23 | 0.24 | 0.30 | 0.32 | 0.26 | 0.27 |
| 3 | 0.74 | 0.70 | 0.53 | 0.56 | 0.62 | 0.62 |
| 4 | 0.37 | 0.37 | 0.41 | 0.43 | 0.39 | 0.40 |
| 5 | 0.30 | 0.31 | 0.28 | 0.28 | 0.29 | 0.30 |
| 6 | 0.61 | 0.70 | 0.60 | 0.54 | 0.60 | 0.61 |

Table 8: Metrics for model with 50 and 100 epochs respectively



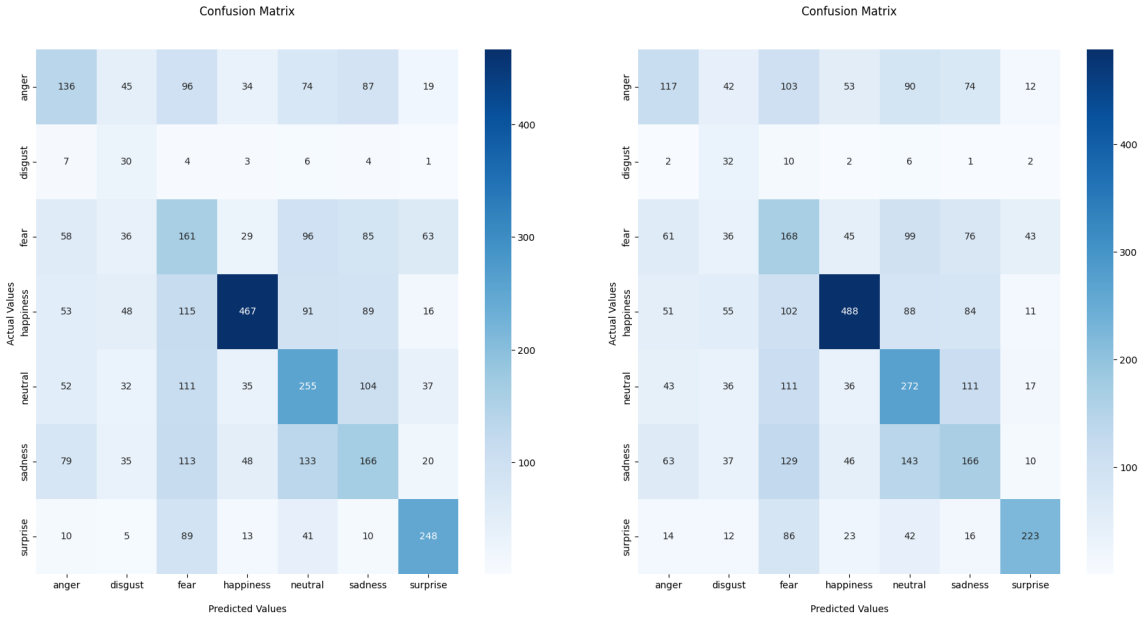

(a) 50 epochs (b) 100 epochs

Figure 18: Accuracy of Balanced Classes




(a) 50 epochs (b) 100 epochs

Figure 19: Loss of Balanced Classes




(a) 50 epochs (b) 100 epochs

Figure 20: Confusion matrix of Balanced Classes

## 6.3 Experiment 3: VGG-16 model

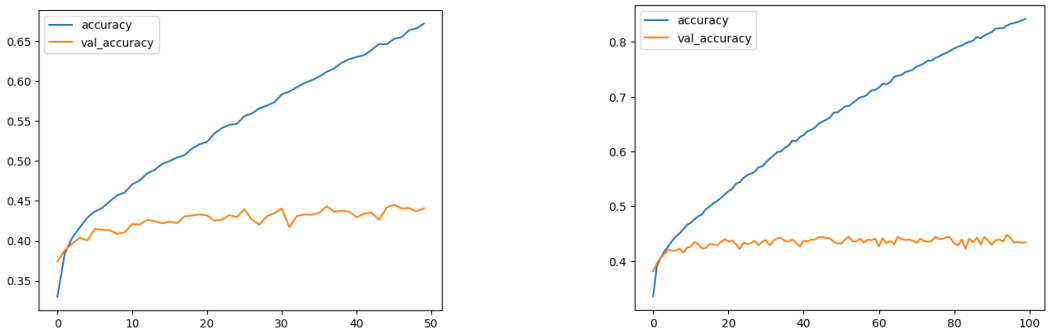
The following are the results for the experiment when used on unbalanced classes followed by the performance graphs and confusion metrics below:

| Evaluation Parameters | | |
|---|---|---|
| No. of Epochs | Accuracy | ROC Score |
| 50 | 44.19% | 0.77403 |
| 100 | 44.66% | 0.76642 |

Table 9: Performance for Unbalanced classes

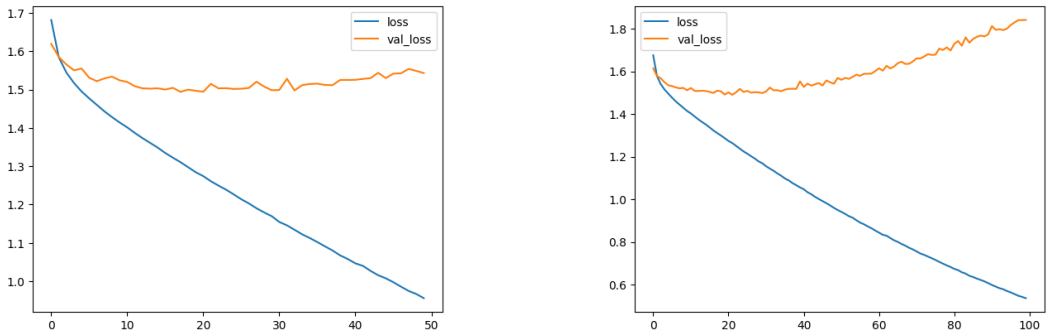
| Classes | Precision | | Recall | | F1 Score | |
|---|---|---|---|---|---|---|
| | 50 epochs | 100 epochs | 50 epochs | 100 epochs | 50 epochs | 100 epochs |
| 0 | 0.36 | 0.39 | 0.32 | 0.31 | 0.34 | 0.35 |
| 1 | 0.60 | 0.40 | 0.22 | 0.36 | 0.32 | 0.38 |
| 2 | 0.37 | 0.36 | 0.29 | 0.34 | 0.32 | 0.35 |
| 3 | 0.50 | 0.55 | 0.63 | 0.57 | 0.56 | 0.56 |
| 4 | 0.41 | 0.41 | 0.39 | 0.44 | 0.40 | 0.42 |
| 5 | 0.36 | 0.35 | 0.41 | 0.40 | 0.38 | 0.37 |
| 6 | 0.65 | 0.60 | 0.55 | 0.59 | 0.59 | 0.59 |

Table 10: Metrics for model with 50 and 100 epochs respectively



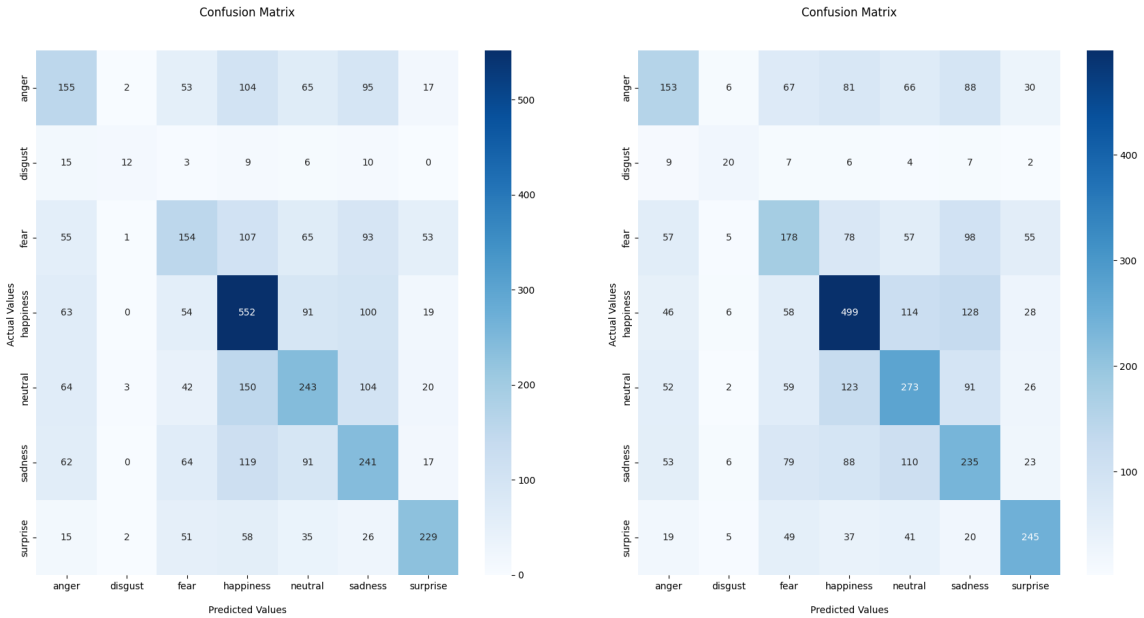

(a) 50 epochs (b) 100 epochs

Figure 21: Accuracy of Unbalanced Classes




(a) 50 epochs (b) 100 epochs

Figure 22: Loss of Unbalanced Classes




(a) 50 epochs (b) 100 epochs

Figure 23: Confusion matrix of Unbalanced Classes

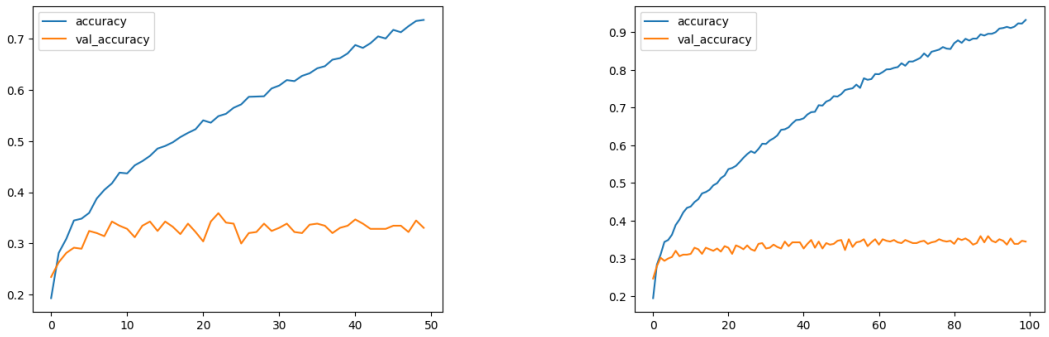
The following are the results for the experiment when used on balanced classes followed by the performance graphs and confusion metrics below:

| Evaluation Parameters | | |
|---|---|---|
| No. of Epochs | Accuracy | ROC Score |
| 50 | 32.54% | 0.72038 |
| 100 | 32.63% | 0.71035 |

Table 11: Performance for Balanced classes

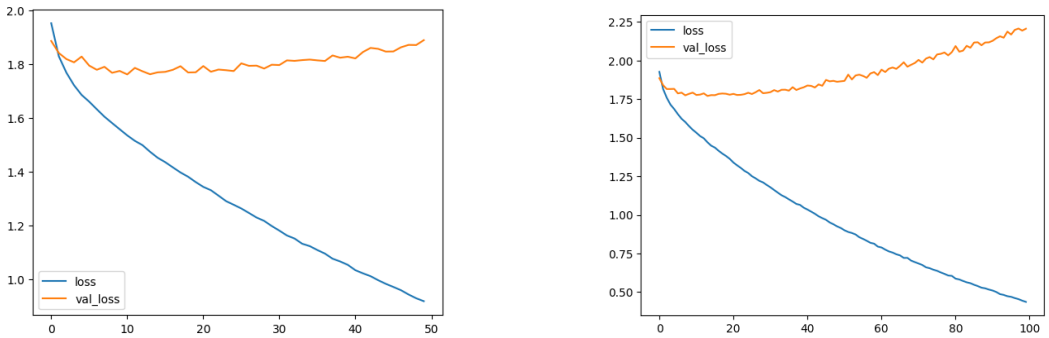
| Classes | Precision | | Recall | | F1 Score | |
|---|---|---|---|---|---|---|
| | 50 epochs | 100 epochs | 50 epochs | 100 epochs | 50 epochs | 100 epochs |
| 0 | 0.27 | 0.27 | 0.24 | 0.24 | 0.25 | 0.28 |
| 1 | 0.09 | 0.12 | 0.67 | 0.67 | 0.16 | 0.20 |
| 2 | 0.28 | 0.28 | 0.14 | 0.26 | 0.19 | 0.27 |
| 3 | 0.53 | 0.49 | 0.31 | 0.31 | 0.39 | 0.38 |
| 4 | 0.31 | 0.31 | 0.33 | 0.36 | 0.32 | 0.33 |
| 5 | 0.30 | 0.29 | 0.37 | 0.28 | 0.33 | 0.28 |
| 6 | 0.43 | 0.44 | 0.57 | 0.52 | 0.49 | 0.48 |

Table 12: Metrics for model with 50 and 100 epochs respectively



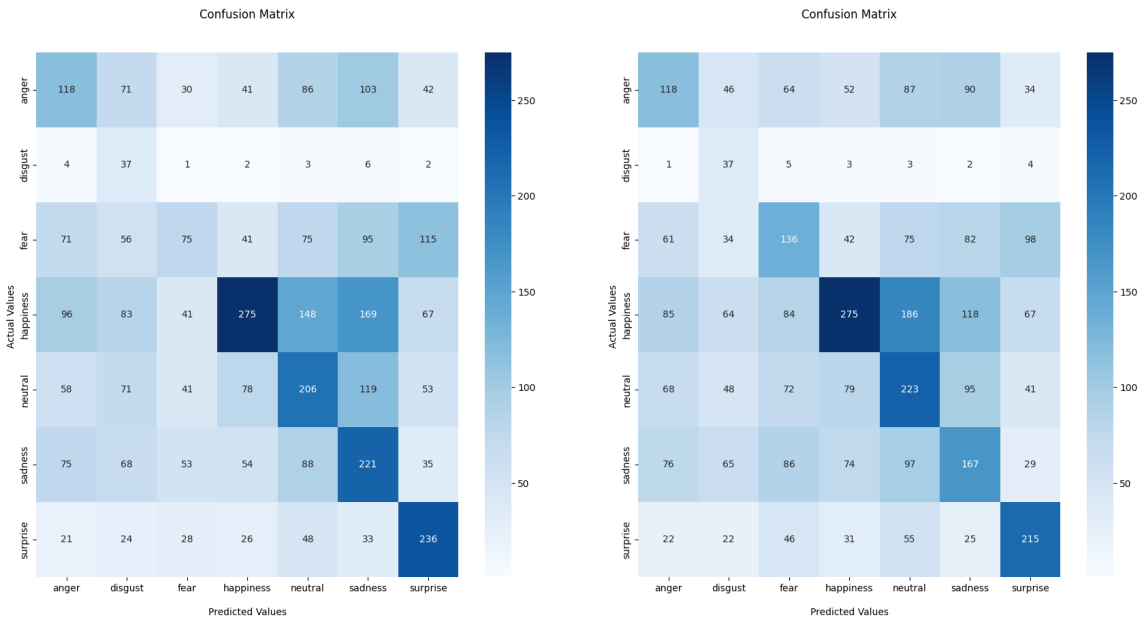

(a) 50 epochs (b) 100 epochs

Figure 24: Accuracy of Balanced Classes




(a) 50 epochs (b) 100 epochs

Figure 25: Loss of Balanced Classes




(a) 50 epochs

(b) 100 epochs

Figure 26: Confusion matrix of Balanced Classes

## 6.4 Discussion

This section provides us with an overview of the obtained results. After analyzing the results, the VGG model performed the worst out of the three giving around 33% for balanced classes and 45% for the unbalanced classes. The best performed overall model was the CNN-LSTM model that gave fairly better results in both balanced classes, around 41% and around 56% for unbalanced classes. But it is fair to say that the CNN model performed very well for the unbalanced class dataset giving the highest accuracy of up to 62%. But it did perform decently for the balanced class dataset up to 42% when the epoch count was set to 100, it gave a lower accuracy of 36% for 50 epochs. This study has helped in the field of medicine to help the doctors and staff to monitor the patient's mental well-being through-out their treatment period, especially if they are going through some tragic event this technology would help them to find them and thereby provide the necessary help.

# 7 Conclusion and Future Work

The research gave that the three models even though they worked very well while training, but not in the testing period was maybe due to the lack of available and properly labelled testing dataset. When comparing the various epoch count and further fine-tuning the model based on the input data might help in improving the overall performance of the models. The model that received the highest accuracy was the CNN model with up to 62% on the training dataset when the model was run for 100 epochs. Followed by this is the hybrid CNN-LSTM model that gave an accuracy of up to 56% while testing. But during its training phase, for both balanced and unbalanced classes it was able to achieve up to 99% as its accuracy score. Even though VGG-16 was built for other use-case scenario and is pre-trained on 'imagenet' dataset, but it could still be able to get a 75%-80% accuracy in its training phase means, that there might be further room for improvement.

Maybe the future availability of large amounts of datasets or development of updated versions of the facial emotion recognition dataset and a more fairly balanced version of the dataset,

might help to improve the accuracy of the classifying model. The need for investment and development in this field has grown very high in the recent years due to to rise of suicide cases, and some traumatic incident that a family member might be going through do to accidents, gun violence, and drug abuse which can degrade a person's mental well-being and their psychological condition. With proper tools and providing treatment before the golden hour would be the reason that they can stand up and fight back against their condition. Advanced development in the field of computer vision and image recognition would help in creating better tools with less compilation or faster preprocessing methods would help in its advancements.

# References


Bhatia, J. K., Singh, J. P., Singh, P. K. and Chauhan, V. K. (2022). Emotion detection using facial expressions, *2022 11th International Conference on System Modeling & Advancement in Research Trends (SMART)*, IEEE, pp. 1491–1498.

Khajontantichaikun, T., Jaiyen, S., Yamsaengsung, S., Mongkolnam, P. and Chirapornchai, T. (2023). Facial emotion detection for thai elderly people using yolov7, *2023 15th International Conference on Knowledge and Smart Technology (KST)*, IEEE, pp. 1–4.

Kousalya, K., Mohana, R., Kumar, B. K., Jithendiran, E., Kanishk, R., Logesh, T., Kumar, E. R. and Sahithya, B. (2023). Group emotion detection using convolutional neural network, *2022 OPJU International Technology Conference on Emerging Technologies for Sustainable Development (OTCON)*, IEEE, pp. 1–6.

Shah, M. (2020). Facial expression recog image ver of (ferc)dataset.
**URL:** *https://www.kaggle.com/datasets/manishshah120/facial-expression-recog-image-ver-of-fercdataset*

Simonyan, K. and Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition, *arXiv preprint arXiv:1409.1556* .

Vasantha, S., Md, A. et al. (2022). Emotion detection using facial image for behavioral analysis, *2022 International Conference on Futuristic Technologies (INCOFT)*, IEEE, pp. 1–7.

Zadeh, M. M. T., Imani, M. and Majidi, B. (2019). Fast facial emotion recognition using convolutional neural networks and gabor filters, *2019 5th Conference on Knowledge Based Engineering and Innovation (KBEI)*, IEEE, pp. 577–581.

Zhou, N., Liang, R. and Shi, W. (2020). A lightweight convolutional neural network for real-time facial expression detection, *IEEE Access* **9**: 5573–5584.