T.C.

OSTIM TEKNIK UNİVERSITY

ENGINEERING FACULTY

EMOTION RECOGNITION USING CNN ARCHITECTURES

SEMESTER PROJECT

Selin ERTUĞRUL

190201020

Selim KOÇ

200201401

COMPUTER ENGINEERING

ANKARA 2024

T.C.

OSTIM TEKNIK UNİVERSITY

ENGINEERING FACULTY

EMOTION RECOGNITION

USING CNN ARCHITECTURES

SEMESTER PROJECT

Selin ERTUĞRUL

190201020

Selim KOÇ

200201401

COMPUTER ENGINEERING

ANKARA 2024

TERM PROJECT ACCEPTANCE AND APPROVAL

Students Name/Last Name: Selin Ertuğrul, Selim Koç

Student Number: 190201020, 200201401

Department: Computer Engineering

The student, whose explicit information is given above, realized in the Spring Semester of the 2023/2024 Academic Year “Emotion Recognition Using CNN Architectures” his titled work has been accepted as a bachelor's semester project.

Advisor Approval	**Dean Manager Approval**
(Name, Surname and Signature)	(Name, Surname and Signature)
Date of Approval:	Date of Approval:

ÖZET

CNN Mimarilerini Kullanarak Duygu Tanıma

Bu çalışma, yapay zekâ ve derin öğrenme tekniklerinin duygu tanıma yeteneklerini değerlendirmek amacıyla gerçekleştirilmiştir. İnsan duygularını anlama ve yüz ifadeleri aracılığıyla analiz etme ihtiyacı, teknolojinin gelişimiyle birlikte giderek artmaktadır. Bu bağlamda, sıfırdan eğitilen derin öğrenme modelleri kullanılarak duygu analizi gerçekleştirilmiştir. Çalışmanın odak noktası, farklı CNN modellerinin duygu tanıma görevindeki etkinliğini vurgulamaktır. CNN (Convolutional Neural Network) mimarisi, özellikle görüntü işleme alanında kullanılan, örüntü tanıma ve sınıflandırma gibi görevlerde etkili olan derin öğrenme modelidir. Modeller, derin öğrenme yaklaşımıyla öğrenilen bilgileri duygu tanıma görevine aktarmayı amaçlamaktadır. Çalışma, FER2013 veri seti üzerinde gerçekleştirilmiş ve modellerin performansı çeşitli metriklerle değerlendirilmiştir. Elde edilen sonuçlar, özellikle VGG16 modelinin diğer modellere kıyasla daha yüksek başarı oranına sahip olduğunu göstermektedir. Bu sonuçlar, yapay zekâ temelli duygu tanıma uygulamalarının gelecekteki potansiyelini ve derin öğrenme modellerinin bu alandaki etkisini anlamamıza katkı sağlamaktadır. Çalışmanın bulguları, duygu tanıma teknolojilerinin geliştirilmesi ve iyileştirilmesi konusunda araştırmacılara rehberlik edebilir. Yapay zekâ tabanlı duygu analizi, sağlık sektöründen akıllı yaşam alanlarına kadar geniş bir yelpazede uygulamalar bulabilir ve insan-makine etkileşimini daha etkili hale getirebilir. Bu bağlamda, Resnet-50 modelinin başarısı, duygu tanıma alanında derin öğrenme modellerinin kullanımının önemini vurgulamaktadır.

Anahtar Kelimeler: Analiz, Derin öğrenme, Duygu Tanıma, Model, Yapay zekâ

ABSTRACT

Emotion Recognition Using CNN Architectures

This study was carried out to evaluate the emotion recognition abilities of artificial intelligence and deep learning techniques. The need to understand human emotions and analyze them through facial expressions is increasing with the development of technology. In this context, sentiment analysis was performed using deep learning models trained from scratch. The focus of the study is to highlight the effectiveness of different CNN models in the emotion recognition task. CNN (Convolutional Neural Network) architecture is a deep learning model that is used especially in the field of image processing and is effective in tasks such as pattern recognition and classification. The models aim to transfer the information learned through the deep learning approach to the emotion recognition task. The study was carried out on the FER2013 dataset and the performance of the models was evaluated with various metrics. The results obtained show that especially the VGG16 model has a higher success rate compared to other models. These results contribute to our understanding of the future potential of artificial intelligence-based emotion recognition applications and the impact of deep learning models in this field. The findings of the study can guide researchers in the development and improvement of emotion recognition technologies. Artificial intelligence-based sentiment analysis can find applications in a wide range from the healthcare sector to smart living spaces and make human-machine interaction more effective. In this context, the success of the Resnet-50 model highlights the importance of using deep learning models in the field of emotion recognition.

Keywords: Analysis, Deep learning, Emotion Recognition, Model, Artificial intelligence

PREFACE

He is a valuable and precious advisor who shared his valuable knowledge with me in the preparation of this semester project, did his best to be useful to me with great interest every time I consulted him, whenever I had a problem I could consult him without hesitation, who did not spare his smiling face and sincerity from me, and I think I will benefit from the valuable information he gave me in my future professional life. I owe a debt of gratitude to Assoc. Prof. Dr. Murat Şimşek, who fulfills the requirements of his status as a teacher, and I present my gratitude.

CONTENTS

FIGURES LIST

TABLES LIST

ABBREVIATIONS

Some symbols and abbreviations used in this study are presented below with their explanations.

Abbreviations	Definitons
AI	Artificial İntelligence
ANN	Artificial Neural Network
CNN	Convolutional Neural Networks
DA	Sentiment Analysis
RSA	Similarity Analysis
ReLU	Rectified Linear Unit
RNN	Recurrent Neural Networks
FC	Fully Connected Layer
ResNet	Residual Network
ILSVRC	ImageNet Large Scale Visual Recognition Challenge
FER	Facial Emotional Recognation
GPU	Graphics Processing Unit
2D	Two Dimentional
VGG	Visual Geometry Group
RGB	Red, Green, Blue
V	Version
TP	True Positive
TN	True Negative
FP	False Positive
FN	False Negative
ROC	Receiver Operating Characteristic
AUC	Area Under the ROC Curve

1. INTRODUCTION

Faces are indispensable elements of human social interactions. A person's face is an expression that a person creates according to their interest in being evaluated positively or favorably by others (Lawrence et al. 2015). These expressions reflect the emotions people feel as they go about their lives.

The ability of facial expressions to reflect emotions plays a critical role in understanding the depths of human psychology and the inner worlds of individuals. The role of emotions in human psychology has a significant importance in shaping individuals' lives and social interactions. Emotions help people understand their inner world and relate to the external world. In this context, emotions play a vital role in human psychology at both individual and societal levels.

Emotions create various postures and meanings in the human body. In addition, facial expressions, tone of voice, gaze, movements and gestures are important in conveying emotions. In addition, the way we express emotions and their manifestations on our face may differ depending on gender, age, culture, environment, race and many other factors. But in general, facial expressions given to the same emotions have universal features (Ananthram, et al. 2020)

The emotions of living beings are reflected in their facial expressions as well as in their souls. In many cases, it can be difficult to detect human emotions just by speaking. Mental, psychological and emotional expressions, which often cannot be expressed, are conveyed to the other party through changes in the human face and body. Emotion recognition can be determined by expressing and interpreting the body language as a whole, based on the voice, gaze, gestures, facial expressions, in short, the whole posture of the person. The changes that occur when all the limbs in the person's face become different from their natural structure create only an idea at the point of determining the facial expression. The facial change that occurs should be considered as a whole and handled as a complex structure. Thus, healthy inferences can be made at the point of determining the expression.

Six of the facial expressions (happiness, sadness, fear, surprise, disgust and anger) have been accepted worldwide as a result of psychological research (Khaireddin & Chen, 2021). These facial expressions play an important role in both expressing emotions and understanding the emotional state of others. Faces are indispensable elements of human social interactions. A person's face is an expression that a person creates according to his or her interest in being evaluated positively or favorably by others (Lawrence et al., 2015). These expressions reflect the emotions people feel as they go about their lives. The ability of facial expressions to reflect emotions plays a critical role in understanding the depths of human psychology and the inner worlds of individuals.

From a psychological perspective, there is a very close relationship between facial expressions and emotional state. For example, a person's frown, pursed lips and lines on the forehead often reflect anger. Similarly, a wide smile and sparkling eyes often express happiness. These facial expressions can help one understand not only one's own emotional state, but also the emotional states of others.

The psychological impact of facial expressions plays a critical role in the identification, expression, sharing and understanding of emotions. The process of emotional expression often enhances communication between people and increases emotional interaction. At the same time, the ability to recognize accurate facial expressions can enhance the ability to empathize and understand other people's emotional experiences.

Differences in people's moods cause physiological changes in many systems in their bodies (Wei, 2013). They show these changes through methods such as body posture, facial expressions and voice (He et al., 2009). Stress, which is one of the situations that people express depending on their emotional states, has an important place in human life (Hartling et al., 2019). Therefore, depending on the emotional states determined from the face, the stress rate of the person can be determined and contribute to mental health.

Psychologists have been conducting extensive studies to understand the effects of facial expressions on emotional state and to better define people's emotions. According to the findings of research in the field of psychology, moderate levels of stress can help people solve some of their problems, while excessive stress can lead to various health problems, including brain damage (Wei, 2013). Therefore, depending on the emotional states determined from the face, the stress rate of the person can be determined and contribute to mental health.

The definition of facial expressions has an important place in artificial intelligence. In the health sector, mental states in patients and their depressive and autistic states can be detected. In addition, in the field of security, it has a very large area of use, from the security of people in exam systems. For example, fatigue can be detected according to the facial expressions of drivers and possible accidents can be prevented. Facial expressions are utilized in many areas such as detecting people in illegal situations and taking up hobbies according to their mood (Kobani, et al. 2015).

Nowadays, with the rapid adoption of technology in our lives, data becomes more complex and creates a complex state. In order to make more meaningful inferences from complex data, serious studies are being conducted on artificial intelligence (AI), image processing and emotion analysis. Face recognition analysis has the most comprehensive place among the research on recognizing human emotion. In particular, the automatic detection of emotions has attracted considerable attention due to the fact that the health system, smart living spaces, health areas such as autism, spectrum disorder, schizophrenia, and the interaction between human and computer and even the interaction between human and robot will be of great importance in the coming years.

At this point, facial emotion recognition is tasked with analyzing emotions between facial expressions. We can say that the realization of facial emotion analysis consists of two main steps. In particular, feature extraction is provided from the image resulting from the perception of the face, facial changes, cropping and resizing.

In this sense, emotions are classified or separated with deep learning methods such as artificial neural network (ANN) and other machine learning. Most commonly, ANNs and especially convolutional neural networks (CNNs) are used to extract existing features from images (Akhand, et al. 2021).

Nowadays, image processing studies in the fields of machine learning and deep learning have increased. Deep learning is a machine learning approach that uses multilayer artificial neural networks to extract meaningful features from complex data sets. Sentiment analysis on facial expressions is a popular task for machine learning and deep learning models. Deep learning methods are an effective tool for analyzing emotion from facial expressions. Facial expressions are first represented by basic features that can be extracted by image processing techniques. These features then feed a deep learning model that classifies the emotion in the facial expression. Deep learning, as the word “deep” in its name suggests, has a multi-layered structure and consists of a more layered structure than artificial neural networks. With these multiple networks, serious studies have been carried out in the field of image processing and have made great contributions to this field (Garcia et al., 2018).

In this study, we addressed the problem of emotion detection from facial expressions. In particular, we focused on six basic emotions. For emotion detection analysis, we used Resnet-50, Traditional CNN, VGG-16 and MobileNet-V3 learning methods from CNN methods. For this, we performed emotion detection after training and testing stages with deep learning methods using artificial neural network in the field of artificial intelligence. In this study, firstly, the purpose and necessity of the study are emphasized and necessary studies are made. In the second part, a large-scale literature study on the study was conducted. In the third section, the methods to be used with the collected materials were determined and the training and testing phases were carried out. In the last stage, the training and test results were compared with the deep learning methods used, the positive and negative aspects of these methods were determined, and the success levels obtained as a result of the studies were presented as a graph and confusion matrix.

2. LITERATURE REVIEW

Facial expressions, which play a significant role in communication among people, occur beyond individuals' control. Physical changes in facial features are crucial for healthy communication.

In this study, Çağlar Atılgan attempted to detect the basic emotions of individuals using artificial neural networks on real data. He utilized changes in specific facial regions, such as the mouth, chin, eyebrows, and eyes. By employing the triangulation method to mark these changes, he aimed to identify the transformations and relationships based on the distances between points. Using the Cohn-Kanade Extended image database, he trained the neural network with over 4,000 images, achieving a 65% accuracy rate. From the collected images, he successfully identified emotions like happy, surprised, sad, angry, surprised, and fearful, independent of attributes such as gender, age, race, and height (Atılgan, 2019).

Afshin Dehghan and colleagues (Dehghan et al. n.d.) designed a deep learning-based pipeline system with deep neural networks in their 2017 study to identify individuals' genders and facial expressions. The system was trained using a dataset containing a total of 4,000,000 images of 40,000 different individuals. For facial expression recognition, images from the FER2013 dataset were used. The images were semi-supervised and labeled according to gender and emotional expressions, and face alignment was performed on the labeled data. The system achieved a success rate of 61.3% in age prediction and 76.1% in facial expression recognition.

Another study focused on emotion analysis using the CNN deep learning model. Videos were recorded second by second, and the momentary changes in human faces were analyzed and plotted. After collecting the necessary emotion-laden materials, the best-performing CNN deep learning model was created, and the facial expressions in the collected videos were analyzed. A total of 61 films were watched, and thousands of images were collected. The films were chosen from various categories with different emotions, and photo frames featuring boredom, fear, happiness, calmness, surprise, anger, and sadness were cropped.

The Haarcascade technique was used for face detection. The detected faces were categorized according to emotions using Amazon's FaceRecognition web service. Although approximately 50,000 face images were collected, around 30,000 images were filtered out due to being low-quality or incorrect based on the Haarcascade and Amazon web service assessments. The training and testing with deep learning revealed that boredom and surprise were the most commonly confused emotions. A 60% accuracy rate was achieved when testing with the remaining five emotion images (Arığ & Metin, 2021).

Ray and Chakrabarti investigated the impact of emotions on learning. Technology-Enhanced Learning is a method that transforms the teaching-learning process, and the role of emotion is often neglected. However, emotion plays a significant role in human cognitive processes; hence, capturing the emotional state of the student is necessary for the learning process to be fully successful. This study proposes an Emotional Computing Module that captures emotional aspects in E-Learning systems, managing learning activities, timing, and the overall learning process. Human-Computer Interaction focuses on recognizing the student's emotional state and facilitates interaction between humans and computers. For this purpose, biophysical methods (heart rate, skin conductivity, blood pressure) and facial expression methods are used. This study explores how emotion transforms in the learning process and how emotion feedbacks are used to enhance learning experiences. A student emotion detection and automatic course selection model is proposed, combining two emotional features to create an effective Affective E-Learning System. The research shows positive results compared to existing systems, proving the effectiveness of the recognition system. From all these data and findings, it can be concluded that facial expressions provide significant and sharp clues about emotions (Ray & Chakrabarti, 2016).

In another study, participants observed dynamic facial expressions transitioning from a neutral expression to angry, happy, or sad expressions. A contrast effect was observed in expressions transitioning to a neutral expression. That is, a neutral expression starting with anger was slightly positively evaluated, while the same neutral expression starting with a smile was negatively evaluated. In the second experiment, sequentially presented static expressions also triggered contrast effects, but the effects following dynamic expressions were weaker. In the

third experiment, various facial movements at different degrees of anger and happiness expressions were evaluated. In slightly expressed movements (25% and 50% expression), balance and push effects were observed. Since the perceived expression was perceived as going beyond the target points, emotion ratings were higher in these movements. In the fourth experiment, sad facial expressions were evaluated, and both contrast and push effects were observed in dynamic expressions. These findings reveal new and effective contextual effects in dynamic facial expressions and highlight the importance of facial movements in socio-emotional communication. Experiment 1 was conducted with 24 undergraduate students at the University of California, Berkeley. Participants were shown video clips containing dynamic movements of angry, happy, and neutral facial expressions. When a neutral expression was reached at the end of the movement, the neutral expression was evaluated differently based on the initial emotional expression. Similarly, momentum effects were observed at slightly expressed points at the end of the movement. The experiment emphasized the contextual effects of dynamic facial expressions and the importance of facial movements in socio-emotional communication. According to the results of Experiment 1, participants were able to influence the emotion ratings of the starting and ending faces. A neutral expression moving from anger to happiness was evaluated as a slightly positive expression, while the same neutral expression starting with a smile was evaluated as negative. Significant bias effects were observed in two neutral conditions. A 100% angry neutral expression was evaluated more positively than the same neutral expression initially observed, while a 100% happy neutral expression was evaluated more negatively than the initial ratings. In Experiment 2, the degree of contrast effects obtained sequentially without dynamic movement was evaluated (Marian & Shimamura, 2013).

3. METHODS

3.1. Data

The data set named FER2013 was used in this end-of-semester project study. This data set has been downloaded from the Kaggle database. The data set consists of color images with 48x48 pixels. The data set is a difficult benchmark set used for emotion recognition of facial expressions. The data set consists of images indicating facial expressions of 7 different emotions. Images are a difficult data set and consist of images that are not distributed properly. There are a total of 35340 images. The number of images for each emotion is given in Table 3.1. Random images from the data set are given in Figure 3.1.

Sıra	Emotions	Number of Images
0	Angry	6989
1	Disgust	1766
2	Fear	6096
3	Happy	7849
4	Neutral	7011
5	Sad	6176
6	Surprise	4005

Table 3.1. Distribution of images in the data set according to classes

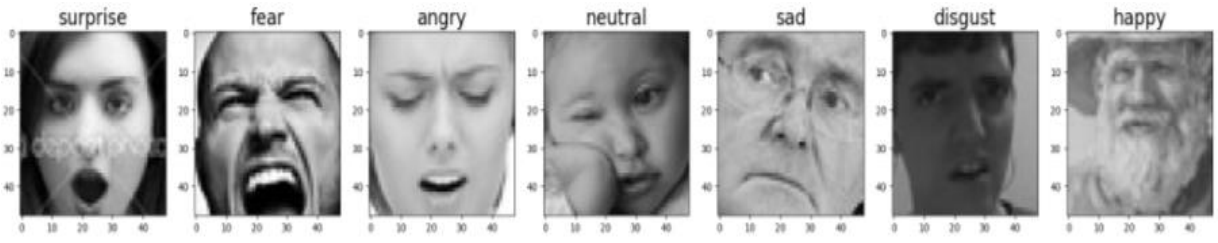

Figure 3.1. Facial Emotion

The bar graph of the distribution of the images in the dataset into classes is shown in figure 3.2. As can be seen from the figure, the most images are in the “Happy” class and the least images are in the “Disgust” class.

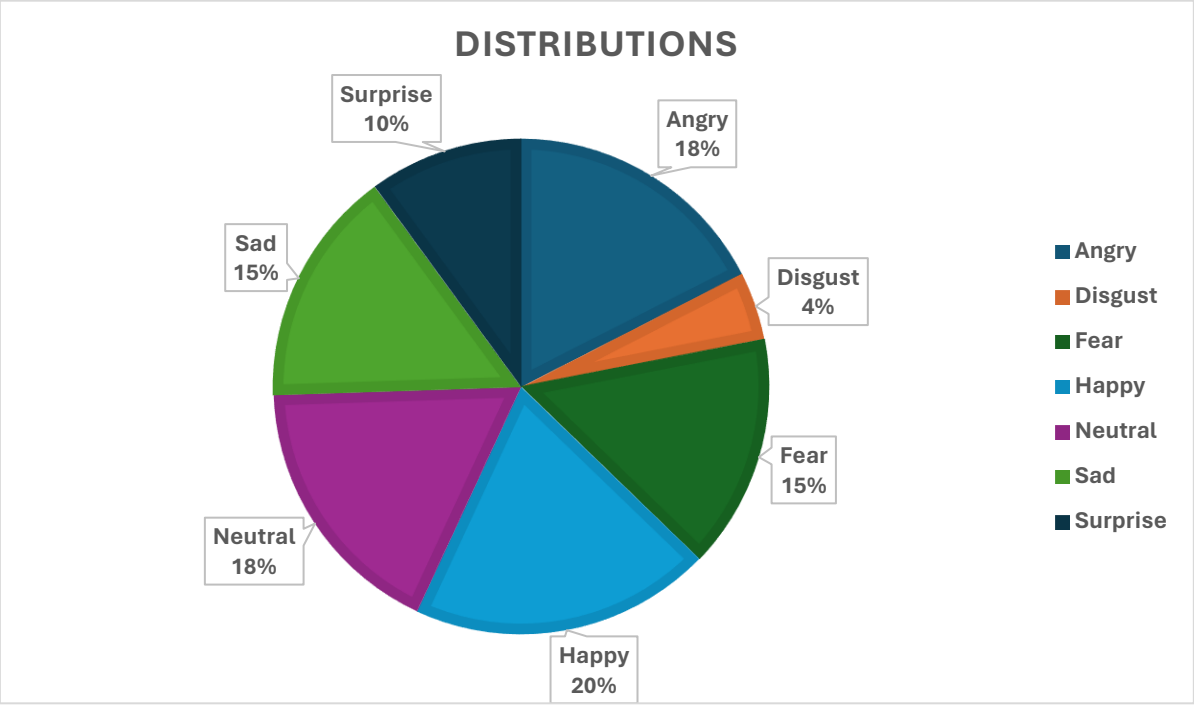

Figure 3.2. Graph of distribution of images in the dataset into classes

3.2 Theory

3.2.1 Deep Learning

Deep learning is a method used in the fields of artificial intelligence and machine learning. Basically, it uses multi-layered deep neural networks that allow the extraction of meaningful features from large and complex data sets. This method applies a multi-layered learning process to represent the complexity in the dataset and Decipher the relationships between the features.

3.2.2 Deep Neural Networks

Deep neural networks learn by adjusting the strengths of their connections to better convey input signals through multiple layers to neurons associated with the right general concepts.

When data is fed into a network, each artificial neuron that fires (labeled "1") transmits signals to certain neurons in the next layer, which are likely to fire if multiple signals are received. The process filters out noise and retains only the most relevant features.

3.2.3 Deep Neural Networks

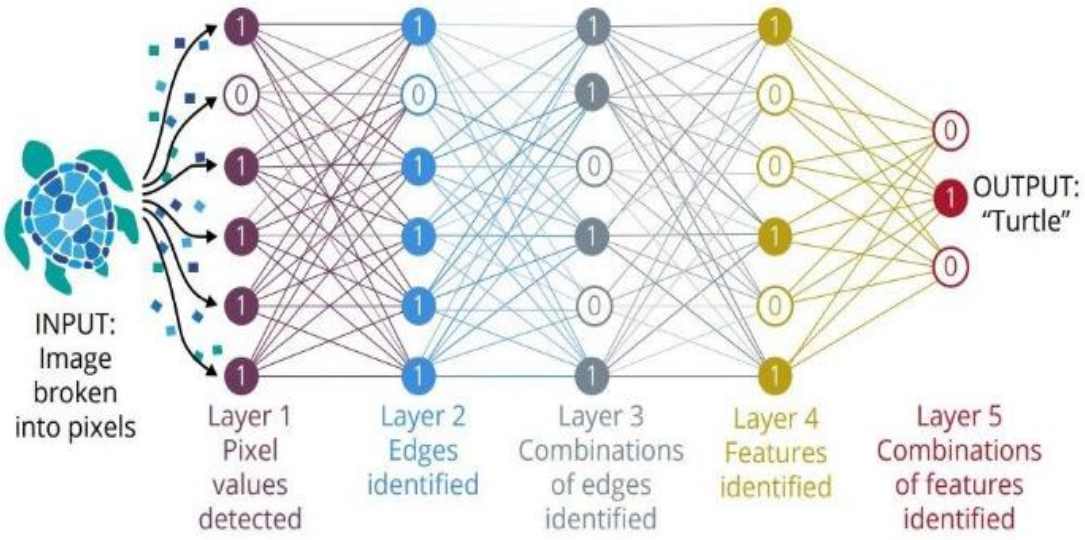

Figure 3.3 Deep Neural Network Architecture Diagram

Deep learning is carried out through deep neural networks. These neural networks consist of layers of interconnected neurons. Input data is fed to the neurons in the first layer, and the outputs from these neurons are passed to the next layer. In this way, each layer processes and represents the data differently, creating higher-level features. This allows the network to learn more complex and abstract features. In the final layer, these representations are used to obtain the desired output.

One of the key factors in the success of deep learning is its ability to learn from large datasets. Large datasets enable the network to see more examples, thereby allowing it to create more accurate and generalizable models. Additionally, deep learning models can perform complex computations quickly, thanks to powerful hardware support (such as GPUs) and large parallel processing capabilities.

Deep learning is particularly effective in tasks with large datasets and complex structures. It has been successfully used in many areas such as image recognition, natural language processing, and speech recognition.

3.2.4 Convolutional Neural Networks (CNN)

The CNN (Convolutional Neural Network) architecture is a feedforward neural network. It is primarily designed to extract features from two-dimensional images. CNN is a subfield of machine learning, a branch of computer science whose foundations were laid in the 1980s. It has the capacity to yield very good results in the learning process with large datasets (Güler, Çinar, & Cengil, 2017). It is one of the deep learning models that has shown great success, especially in tasks like image processing and data classification. CNNs began to be processed and trained after the 1990s, and by 1995, they started being used in image processing applications (Lawrence, Giles, & Chung, 1996). Today, the basic structure of CNNs, which have shown successful results in many fields, is seen below.

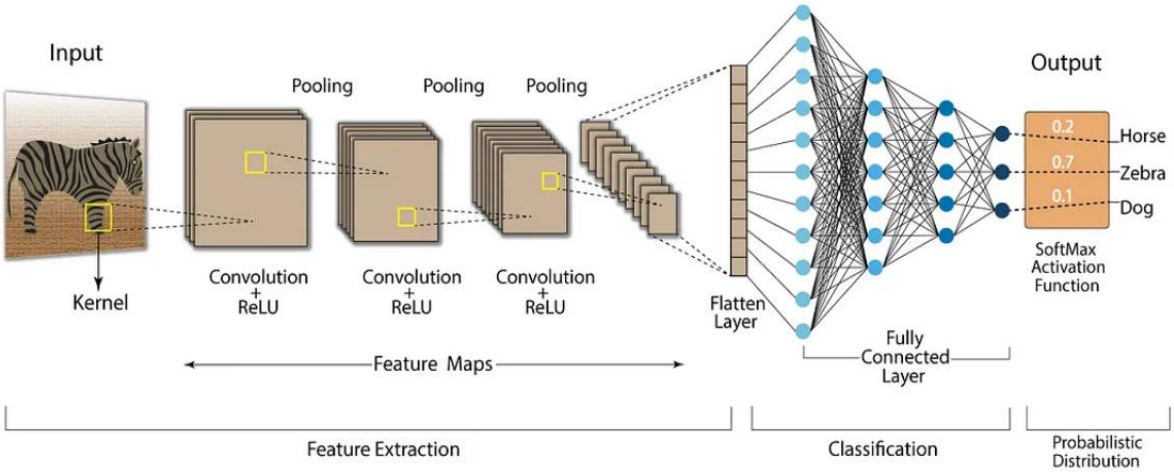

Figure 3.4. CNN Neural Architecture

CNN architecture consists of two main sections: Feature Extraction and Classification, along with several associated layers. The fundamental structure of CNN begins with the Convolution layer, where filtering operations are applied to input data. The values obtained after applying the ReLU activation function are then passed to the Pooling layer. The Pooling layer reduces the dimensionality of 2D matrices by transforming them into a linear vector. Subsequently, the extracted features are fed into the Fully-Connected layer for classification. In this layer, classification is performed using Softmax regression.

The learning model of CNN takes input images and defines the characteristics of objects, making them distinguishable and extracting relationships between objects. This process is carried out through convolution, pooling, fully connected, and classification layers. Thus, distinguishing features of images are identified, and artificial learning is conducted through CNN.

The effectiveness of CNN in visual data processing stems from its ideal structure for analyzing and classifying complex matrix data. This model can be considered as an application of matrix multiplication and achieves the best results, particularly in visual data processing scenarios. Figure 3.5 illustrates a basic CNN structure.

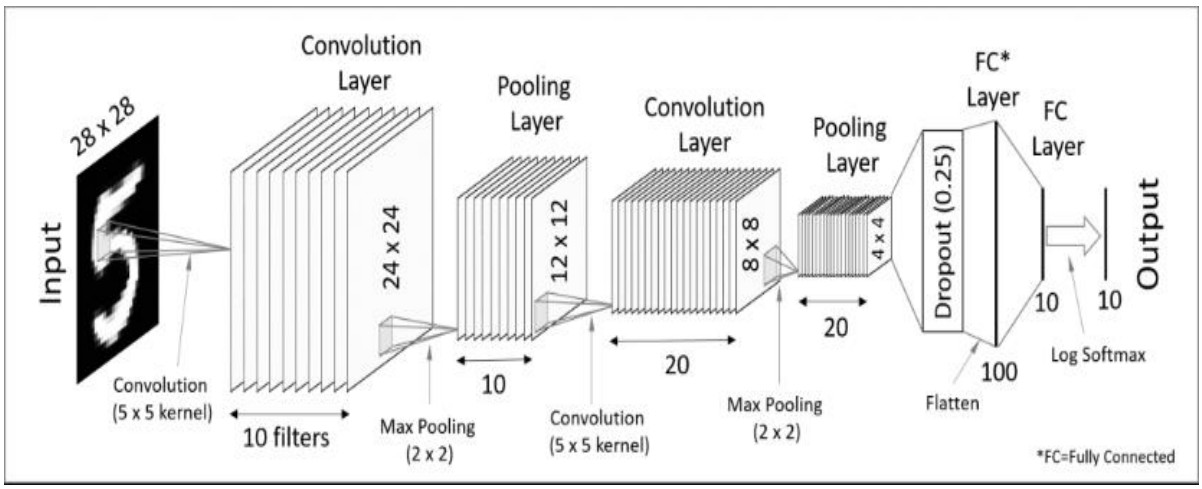

Figure 3.5. CNN Structure

The basic principle is to determine and process different features of images using specialized filters. These filters are typically trained to recognize edges, corners, and other significant features. Subsequently, the results obtained from these filters are transmitted between layers to generate more complex outcomes. CNN consists of multiple trained layers arranged consecutively to process data. The first layer is the input layer, which feeds data into the network. Then, in the convolution layer, data is processed based on various features. This layer plays a crucial role in tasks such as identifying features in images. Following that, classification and inference occur in the output layer, where final results are obtained. The overall structure of the network is adjusted and learned through an algorithm called backpropagation. This algorithm compares the network's predictions with actual labels, calculates errors, and updates the parameters of the network to minimize these errors.

This process enables the network to learn better features and make more accurate classifications. As a result, CNN achieves significant success in recognizing and classifying features in images, providing great convenience in various application areas such as image classification, object recognition, medical image analysis, autonomous vehicles, and many other specialized tasks. Additionally, they are effectively used in object detection applications to determine the positions of objects in images.

The convolution operation is a computation method specific to a filter's input image, essential for determining the qualities and important features in existing images. This method, which plays a crucial role in identifying features and processing images, is commonly used in image processing and particularly in deep learning fields. Convolutional neural networks generally consist of three fundamental layers. The first layer, the "Convolutional layer," highlights features in the data, making it easier to recognize patterns. Then, the "Pooling layer" summarizes the data in an original way, reducing its size to decrease processing time and costs. Finally, the "Full Connection Layer" combines all these features to produce results. These layers together form the basic structure of convolutional neural networks, facilitating the learning of features in complex visual data.

3.2.4.1 Input Layer

In the CNN architecture, the input layer is where the network starts, and it receives raw data at this layer. In other words, it's the starting point for the data that the model will use. Depending on the architectural structure of the model, input values should be carefully selected and organized because this is where the model begins processing the data, serving as the first step. This section, where the fundamental data representing the dataset is presented, contains the input values used for training the model. These data are used during the learning process of the network and are employed for the model to perform specific operations on the data. The input layer is where the data is accepted, and then it's passed on to the rest of the network. Therefore, the input layer is an essential part of CNN and the point where the model interacts with the data.

3.2.4.2 Convolutional Layer

Convolutional layers process input data using special filters. Initially, these layers process input data to extract basic features, such as edges and curves. Subsequent layers then focus on more complex features, reaching higher-level results like recognizing shapes and sizes of objects.

A convolutional layer creates a feature matrix by processing data with special filters. This matrix assists in feature extraction and highlights significant information in the input data. Particularly in tasks like image processing and recognition, convolutional layers play a critical role in detecting details.

The learning process primarily occurs in the convolutional layers of CNNs. During this process, the network's parameters are updated to minimize error values obtained from the feature matrix. This allows the network to learn more complex features and operate more effectively on data. Convolutional layers enable CNNs to learn features hierarchically from input data and generalize these features, contributing to the successful performance of CNNs.

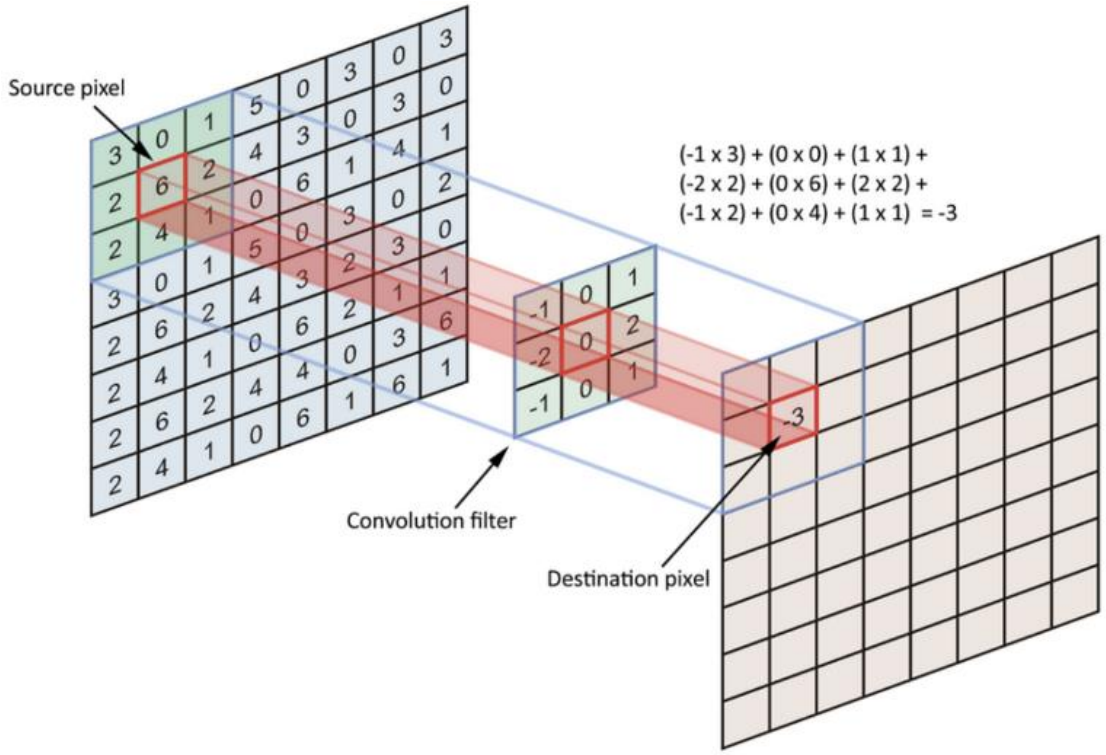

Figure 3.6. Convolution layer feature extraction

As seen from the above diagram, the convolution process occurs as follows. Firstly, a 8x8 input data matrix and a 3x3 filter are used. This filter performs a cross-correlation operation at each position while sliding over the input matrix. The resulting values from these multiplications are summed up, and this process is repeated across a 3x3 window, thus creating a feature map. Another function of the feature map is to contain weight values representing different features in the image. These features are used to capture visual characteristics such as edge detection and color changes. Convolutional layers successfully perform complex visual recognition tasks by utilizing these features as the fundamental structure of deep learning networks.

3.2.4.3 Activation Layer

The activation layer is a layer found in artificial neural networks and is used to determine the output of each neuron. This layer contains activation functions, which are mathematical functions applied to the input values to determine the output values calculated by the neurons.

An activation function provides a nonlinear transformation when determining the output of a neuron. This allows artificial neural networks to model more complex relationships. Without these nonlinear transformations, neural networks would struggle to model sufficiently complex structures if they were limited to linear functions. Therefore, activation functions overcome this limitation by keeping neuron outputs within a limited range and applying a nonlinear transformation. As a result, neurons continue the learning process of the network by transmitting this processed output signal to the next layer.

The most commonly used activation functions include ReLU (Rectified Linear Activation), sigmoid, and tanh (Hyperbolic Tangent). The nature of the problem determines the chosen activation function. Thus, the problem should be differentiable, sensitive, nonlinear, and within specific ranges. A sensitive activation function is especially important in fields like image processing, where matrix operations are applied. Otherwise, the learning speed may be reduced. Below are the mathematical equations and graphs of activation functions used in CNN and deep learning architectures.

Sigmoid Fuction

The sigmoid activation function is a nonlinear activation function graphically shaped like an S-curve. This function, frequently used in the layers of artificial neural networks, compresses input values into the range of 0 to 1, presenting outputs in a probability-like form. This characteristic makes it particularly preferred for binary classification problems. The sigmoid activation function helps the network incorporate nonlinear features, thereby enhancing its ability to learn complex relationships. Additionally, it is differentiable (Sharma, Sharma, & Athaiya, 2017), which allows for more effective learning processes.

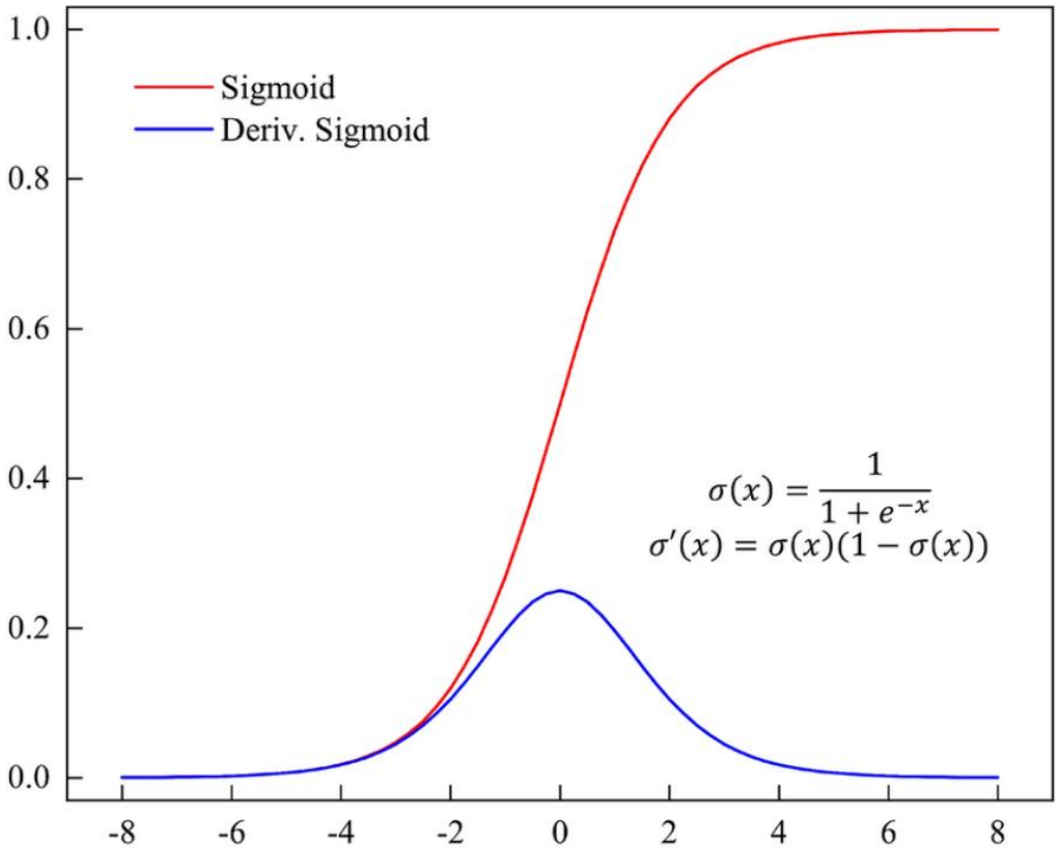

Figure 3.7 Sigmoid Function and Derivative Graphic

Hyperbolic Tangent Function

The hyperbolic tangent function takes values between -1 and 1 and forms an S-shaped curve passing through the center 0. The hyperbolic tangent function has a more symmetrical structure than the sigmoid function. This activation function produces output in the range [-1, +1] and is especially used in binary classification problems. Having a wider value range compared to the sigmoid function allows neural networks to learn faster and increase efficiency. Thanks to this feature, the hyperbolic tangent activation function increases the performance of neural networks by enabling them to learn in a wider spectrum and faster.

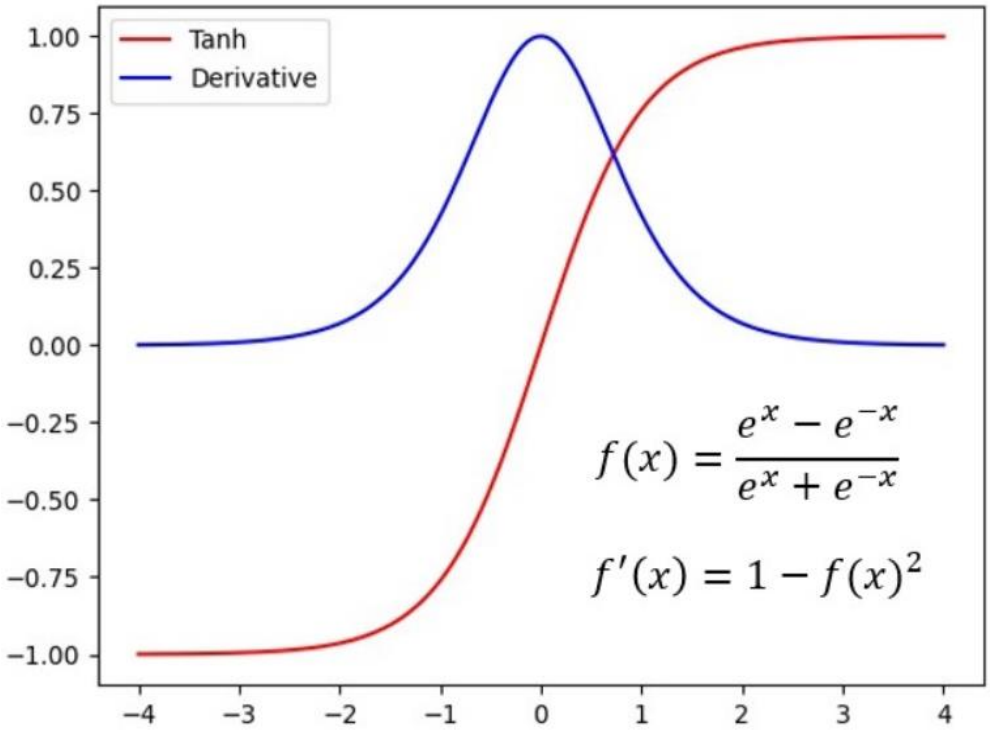

Figure 3.8 Hyperbolic Tangent Function and Derivative Graphic

While the tanh activation function may be more efficient than the sigmoid activation function, it encounters the same problems as the sigmoid function during backpropagation. In case of very large or very small values, the derivative of the tanh function gradually approaches zero, making it difficult to train the neural network. As an exponential function, it is computationally costly. However, the tanh function is a useful activation function in hidden layers to pass better input values to the next hidden layer.

ReLU Function

ReLU (Rectified Linear Unit) is a widely used activation function in deep learning models. ReLU works by comparing the input to a threshold value, passing positive inputs as they are while setting negative inputs to zero. In other words, positive values pass through, and negative values become zero. ReLU is simple to compute, which speeds up the training process and helps reduce the vanishing gradient problem. However, it can cause some neurons to constantly output zero, known as the "Dead ReLU" problem. To prevent this, variants like Leaky ReLU are used, which add a small slope to negative inputs, keeping all neurons active.

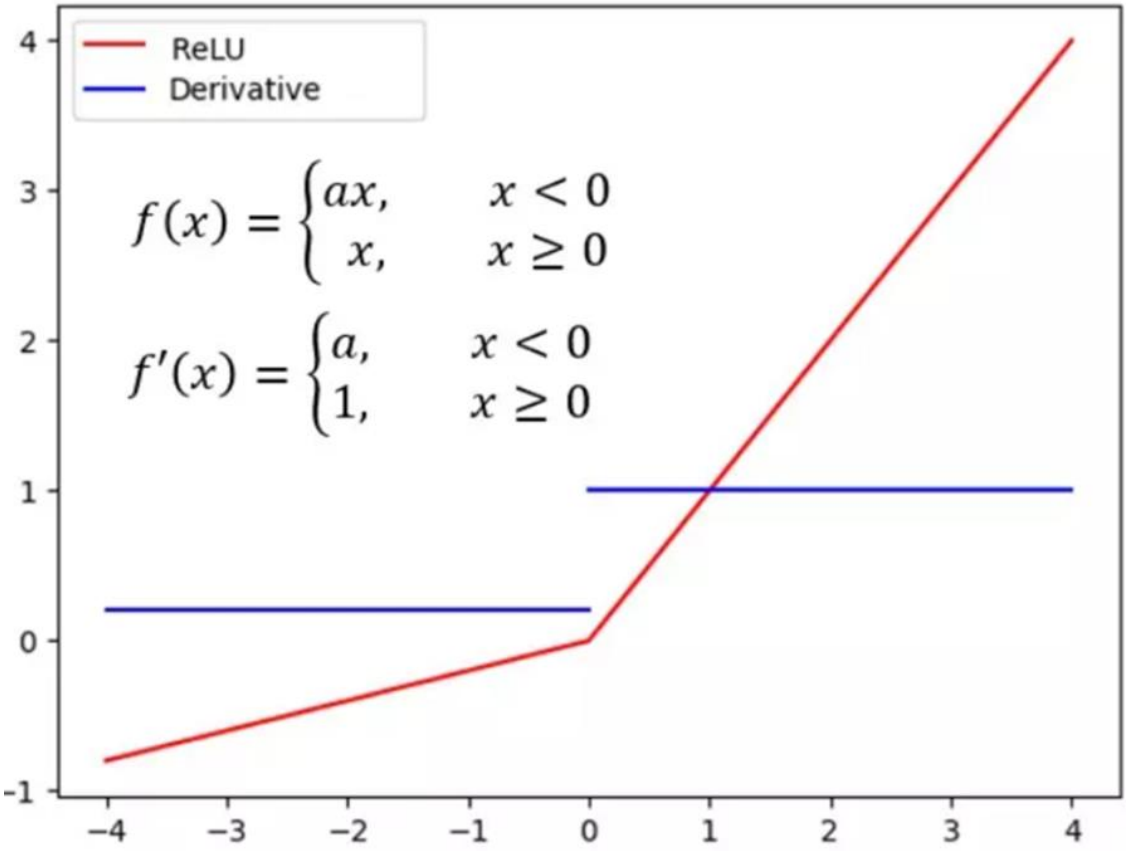

Figure 3.9 ReLU Function and Derivative Graphic

Leakly Relu may be particularly useful in overcoming ReLU's so-called "dying ReLU" problem. The “dying ReLU” problem may arise because the ReLU function produces zero output with negative inputs, with the result that some neurons are not active at all during training and these neurons do not contribute to learning. The fact that Leaky ReLU is more flexible against this problem makes the learning process more effective.

3.2.4.4 Pooling Layer

The pooling layer optimizes large datasets by reducing the number of weights, thus accelerating the learning process and lowering costs, thereby enhancing functionality. No training occurs in this layer. Instead, it maintains the features and the training process takes place after the convolution layer. The pooling layer shortens the training time, saving both time and resources.

During the pooling process, a window or filter is defined over the input data and moved with a certain stride. This window scans the input data, and pooling is applied to each region. This way, the features of the input data are collected or reduced, and the maximum or average value of each window is taken. This process reduces the size of the input data while preserving important features. The most commonly used types of pooling are max pooling and average pooling.

Max pooling selects the maximum value within each stride window, emphasizing important features of the input data and reducing its size. Average pooling, on the other hand, takes the average of the values within each stride window. This also reduces the size of the input data while generally preserving important features.

The pooling layer is used to identify important features in the feature map and allows the network to operate with fewer parameters. This enables the network to train faster and use less memory. Additionally, the pooling process can help improve the network's learning and prevent overfitting.

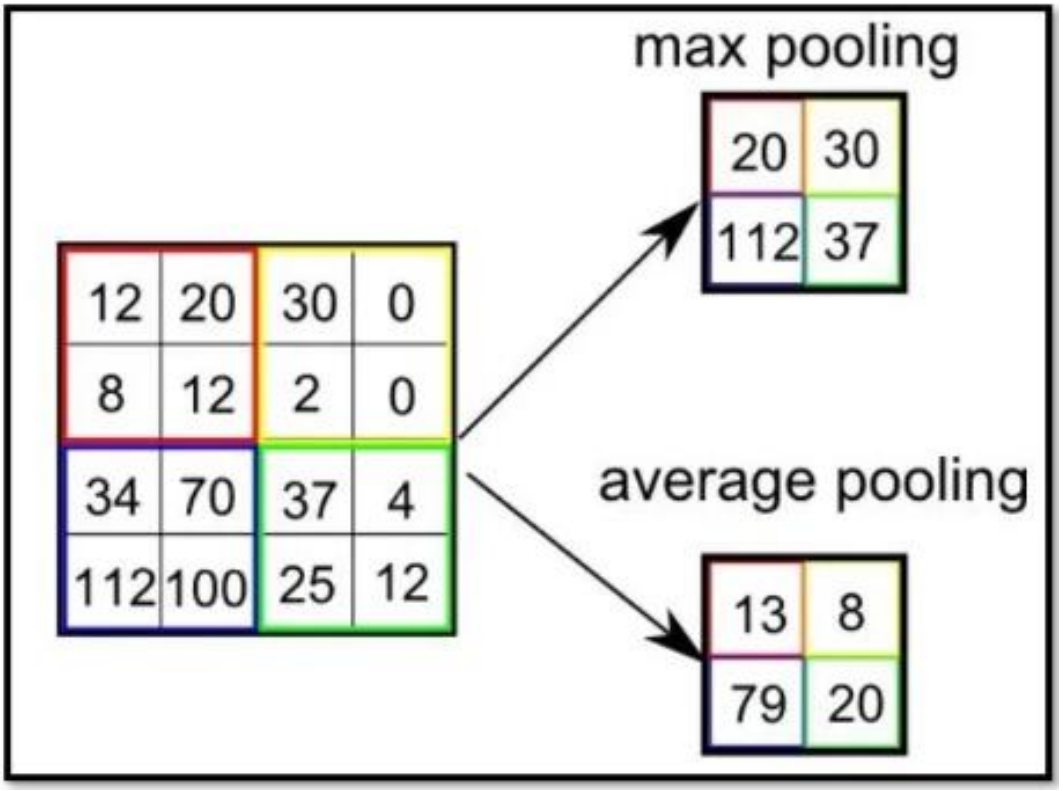

Figure 3.10 Max Pooling and Average Pooling Methods

The figure above shows the maximum pooling and average pooling methods. When 2x2 pooling is applied on the indices in the data matrix, the maximum values in the matrix data are taken, reducing the size and reducing the processing load. In the average pooling method, the average of the values in the input data to which a 2x2 matrix is applied is taken to create a new matrix.

3.2.4.5 Fully Connected Layer

The Fully Connected Layer takes the features learned in the previous layers of the neural network model as a matrix and turns this matrix into a vector by flattening it and is used to calculate the output probabilities from the input data. Usually in classification or regression problems, this layer results in a softmax activation function and each output neuron represents the probability that the image belongs to a particular class for the classification problem.

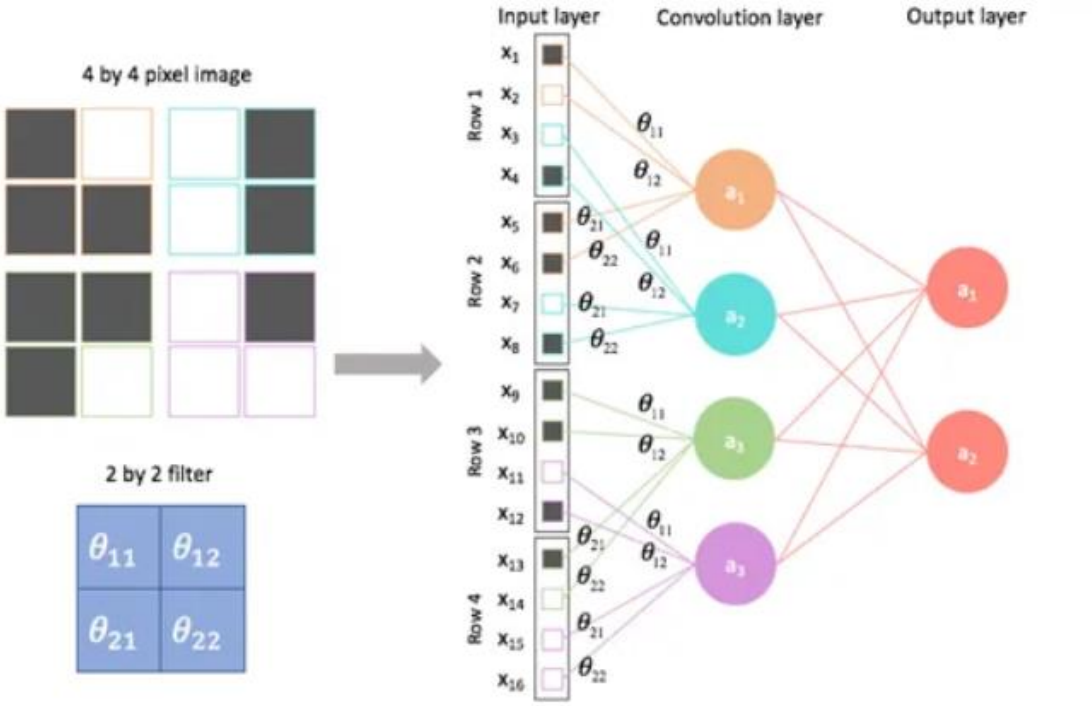

Figure 3.11 Fully Connected Layer

The Fully Connected Layer increases the learning capacity of the network and helps model complex relationships between input data and output. This layer is widely used in various applications of deep learning models.

3.2.5 CNN Models Used in the Project

3.2.5.1 Traditional Cnn

The traditional Convolutional Neural Network (CNN) architecture is particularly effective for tasks such as image recognition and classification. This architecture typically consists of a series of layers: convolutional layers, pooling layers, and fully connected layers.

Traditional CNNs usually have a limited number of layers and employ the Rectified Linear Unit (ReLU) activation function. ReLU sets negative values to zero while leaving positive values unchanged. During the training process, backpropagation and gradient descent algorithms are used to update the network's weights and biases.

The traditional CNN architecture serves as an effective starting point for image processing and classification tasks. However, modern CNN models (e.g., VGG, ResNet) offer superior performance through various improvements such as increased depth, innovative layer structures, normalization techniques, and data augmentation methods. These advancements enable tackling more complex and challenging tasks in the field of deep learning.

In this study, it uses the classic CNN architecture. Initially, the model starts with `Sequential()` and then sequential layers are added. The model starts with a series of `Conv2D` layers. Each `Conv2D` layer applies 2D convolution operations and then uses "relu" as the activation function. After each `Conv2D` layer, a `BatchNormalization()` layer is added. These layers normalize the output of the convolutional layers and help reduce overfitting. Then, a `MaxPool2D` layer is added. This reduces the output of the convolutional layers and emphasizes important features. Towards the end of the model, fully connected layers are added. These include a layer that flattens the features from the previous layers (`Flatten()`) and then hidden layers (`Dense()`) are defined. The final layer is created with the `softmax` activation function to determine the output classes.

This creates the Classic CNN model. Feature extraction, increasing the depth of the model, and classification operations are performed. ImageDataGenerator is used for data augmentation and normalization. A custom F1 score function is defined and the model is compiled with the Adam optimization algorithm. The training process is performed with ReduceLROnPlateau and ModelCheckpoint callback functions. When training is completed, the model and weights are saved, and performance graphs are created.

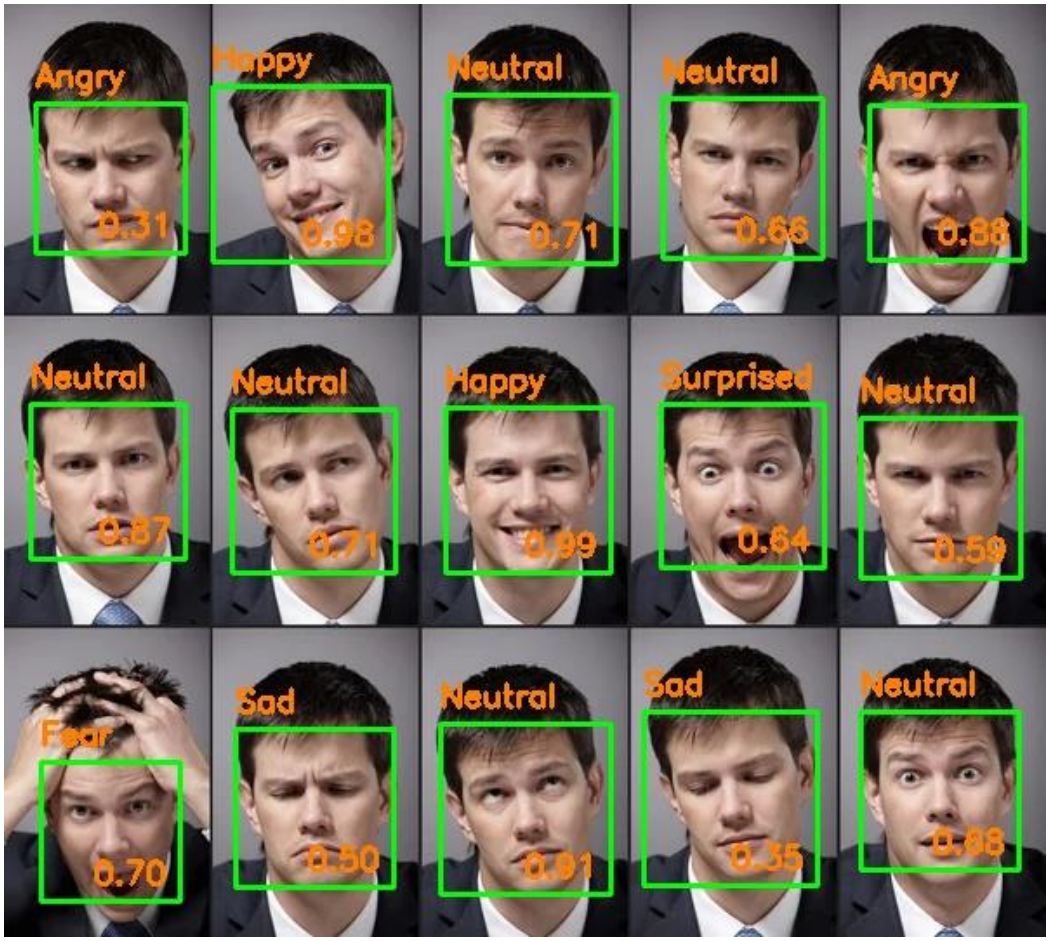

Figure 3.12 Output of CNN Model

3.2.5.2 ResNet

ResNet (Residual Network), He and his colleagues developed in 2016, is one of the deep learning models. The challenges encountered in deep learning training have led He and his colleagues to work on new models. In order to overcome these challenges, they created the ResNet architecture.

Deep learning training is generally a long process, and the number of layers used may be limited. ResNet comes into play at this point, providing solutions such as residual connections to make the process faster and smoother. The main feature of ResNet is that the data passing through each layer is added to the original input data with residual connections. These connections work by directly adding the output of the previous layer to the input of the next layer. This means that information from earlier layers is directly transmitted to later layers. These connections allow the network to be deeper and ensure smoother training.

Thus, ResNet provides much larger advantages compared to other architectural models. The ResNet model is implemented with skip connections, consisting of ReLU and two or three layers among architectures.

Another fact reached during the applications is that the ResNet model gives better results in image classification than other previously used models. As a result, it can be seen that the image features obtained by ResNet are good.

The residual block in ResNet succeeds when the input data sizes are equal to the output data sizes. Additionally, each ResNet block consists of two layers (for ResNet-18 and ResNet-34 networks) or three layers (for ResNet-50 and ResNet-101 networks). Figure 3.12 The general structure of the ResNet50 model can be seen.

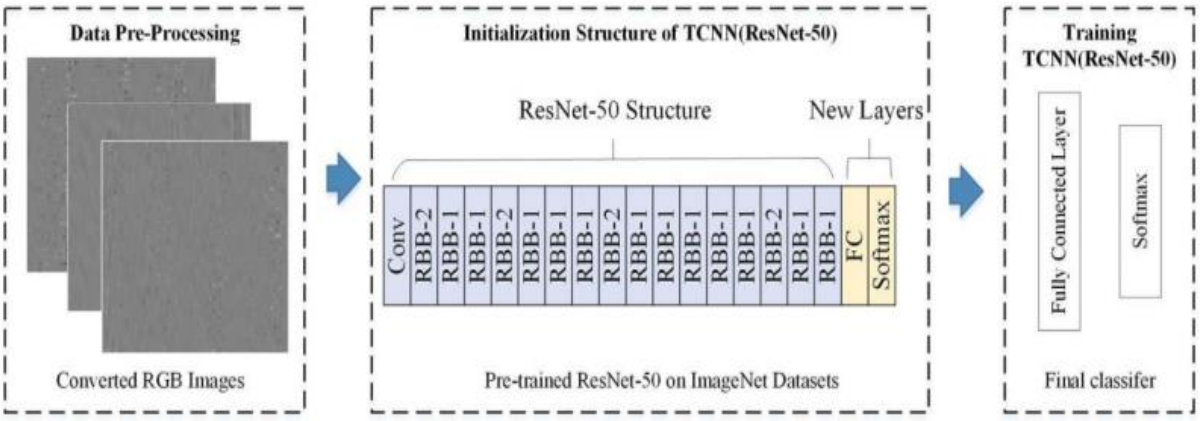

Figure 3.13 TCNN(ResNet-50) Flowchart

ResNet-50,

ResNet-50, short for Residual Network-50, is built on the fundamental idea of the model: while deeper networks can potentially achieve better performance, training them becomes more challenging. To address this challenge, it was proposed to add a residual connection between the input and output of each layer, suggesting that deeper networks could be easily trained.

ResNet-50 is a model with 50 layers. Its foundation lies in connecting each layer of the network with a special structure called "residual blocks," even though the network is deeper, to achieve better performance. These blocks, present in every layer of the network, provide a type of "skip connection" or "shortcut connection" that facilitates learning. These connections add the input of the layer directly to its output, making it easier for the network to learn and enabling the creation of deeper structures.

The creation of ResNet-50 can be achieved using methods such as transfer learning or training from scratch, with the utilization of a pretrained model. By using a pretrained model, the network is typically trained on a large dataset and then fine-tuned for a specific task.

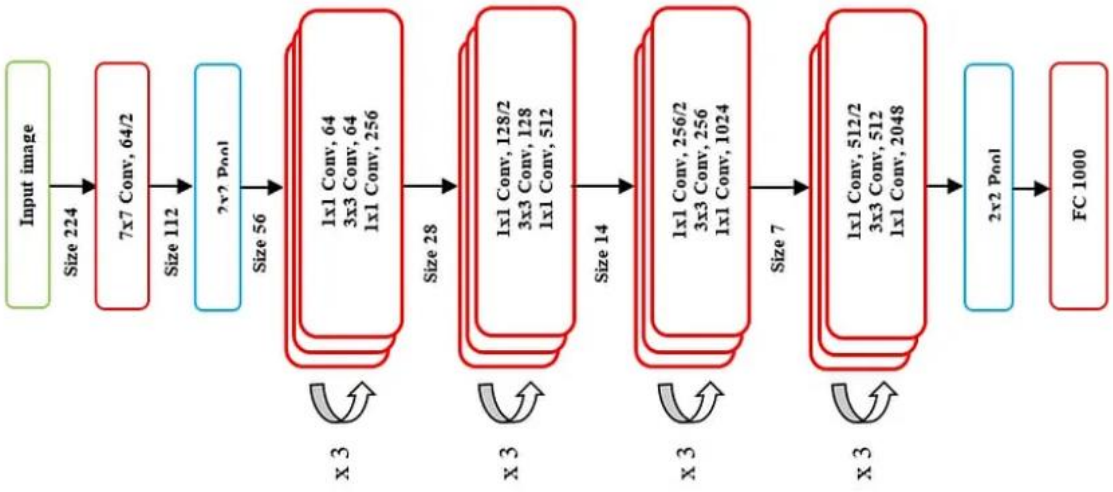

Figure 3.14 ResNet-50 Architecture

As seen in the figure, the ResNet-50 architecture includes the following elements.

• A convolution with a core size of 7 * 7 and 64 different cores and all with a step of size 2 gives 1 layer.

• Next, maximum pooling with step size 2 appears. 31

• The next convolution has 1 * 1.64 kernels, then 3 * 3.64 kernels, and finally 1 * 1.256 kernels. These three layers are repeated a total of 3 times, giving 9 layers in this step.

• Then we see 1 * 1,128 nuclei, then 3 * 3,128 nuclei and finally 1 * 1,512 nuclei, this step is repeated 4 times, which gives 12 layers in this step.

• After this comes a core of 1*1,256 and two more cores of 3*3,256 and 1*1,1024, and this is repeated 6 times, giving a total of 18 layers.

• And then again a core of 1*1.512, two more 3*3.512 and 1*1.2048 and this is repeated 3 times giving a total of 9 layers.

• After this an average pool is made followed by a fully connected layer containing 1000 nodes and finally terminated with a softmax function which gives 1 layer. In fact, activation functions and maximum/average pooling layers do not count. So the sum of this gives a Deep network with 1 + 9 + 12 + 18 + 9 + 1 = 50 layers.

This study employs TensorFlow and Keras to implement a ResNet50 model for image classification tasks. Initially, essential libraries are imported, and a custom f1_score function is defined to calculate the F1 score metric based on precision and recall values. Data generators are created to load and preprocess training and testing images, including augmentation techniques like rotation and horizontal flipping. The ResNet50 model, pretrained on ImageNet, is instantiated and added to a Sequential model along with additional layers for classification.

The model architecture includes convolutional layers, max-pooling layers, dropout layers, and dense layers. It is compiled with the Adam optimizer and categorical crossentropy loss function. During training, callbacks such as ReduceLROnPlateau and ModelCheckpoint are utilized for dynamic learning rate adjustment and model checkpointing, respectively. The training process is executed using the model.fit function, monitoring metrics such as accuracy, loss, precision, recall, and F1 score on both training and validation sets. After training, the model is saved in both .h5 and .weights.h5 formats. Finally, the model's performance metrics are visualized using matplotlib, providing insights into the training progress and validation performance.

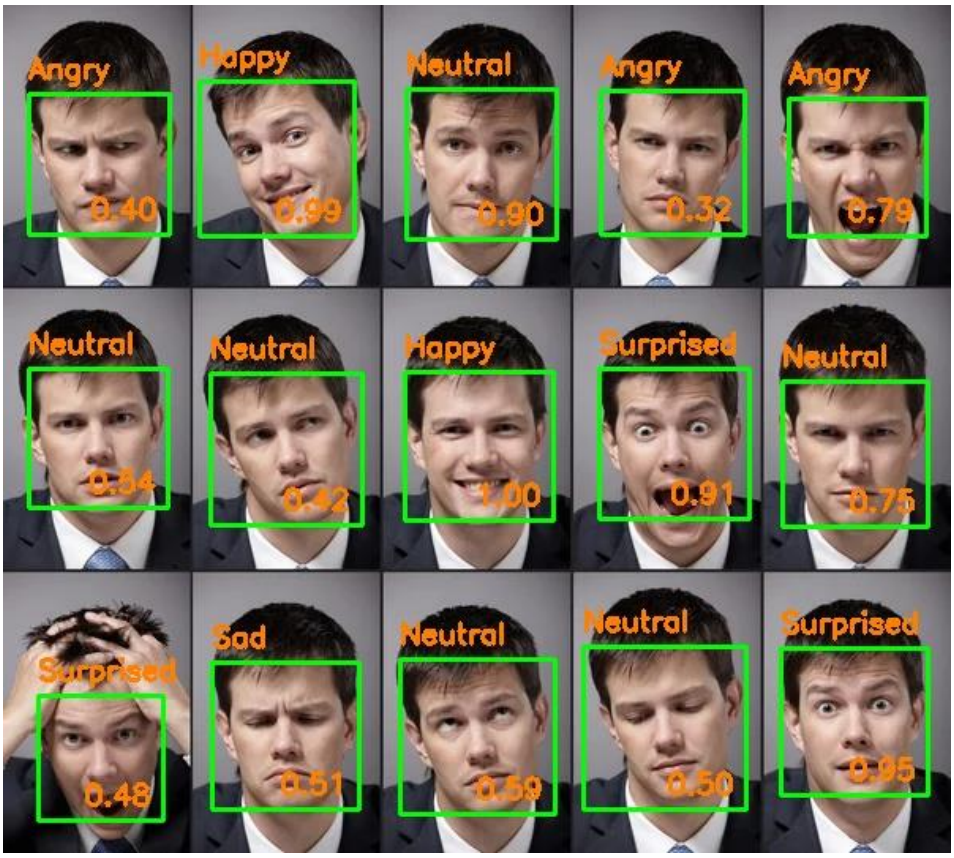

Figure 3.15 Output of Resnet-50 Model

The results obtained show that ResNet-50 has higher prediction accuracies than other deep learning models and traditional methods.

3.2.5.3 VGG

VGG, a deep learning model developed by the Visual Geometry Group at the University of Oxford, is known for its significant success in the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) in 2014. The term "VGG" stands for the abbreviation of this group.

One of the most distinctive features of VGG is its highly homogeneous structure in terms of depth. The model is presented in two different variants consisting of 16 or 19 convolutional layers. These convolutional layers sequentially process the data to learn more complex features. Following these convolutional layers, fully connected layers and finally classification layers are included to complete the model.

The success of VGG is significant in demonstrating the usability of deep networks in tasks such as image classification. Additionally, its homogeneous structure allows the model to be easily understandable and reusable. These characteristics have made VGG a significant milestone in the field of deep learning. Figure 3.14 illustrates the general structure of the VGG-16 model.

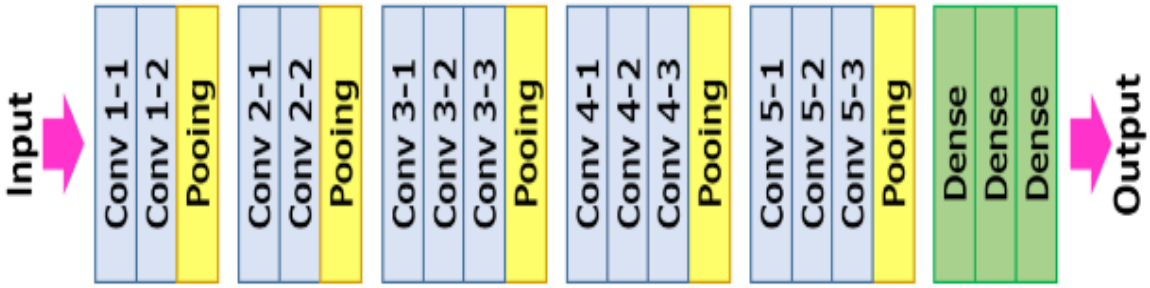

Figure 3.16 VGG-16 Model Structure

VGG-16

VGG-16 is a widely used deep learning model, particularly in the field of image classification. With a 16-layer deep neural network architecture, VGG-16 includes small 3x3 convolutional layers and 3 fully connected layers. It also employs max-pooling operations. Trained on the ImageNet dataset, VGG-16 has demonstrated high performance in the ImageNet image classification competition. Therefore, it achieves good results in general image recognition tasks. It is utilized in various computer vision applications such as image classification, object detection, image recognition, and visual feature extraction. Some advantages of VGG-16 include its high classification accuracy, transferable features, and the simplicity and regularity of its structure. In conclusion, VGG-16 is a powerful and successful deep learning model widely used in image classification and other computer vision applications.

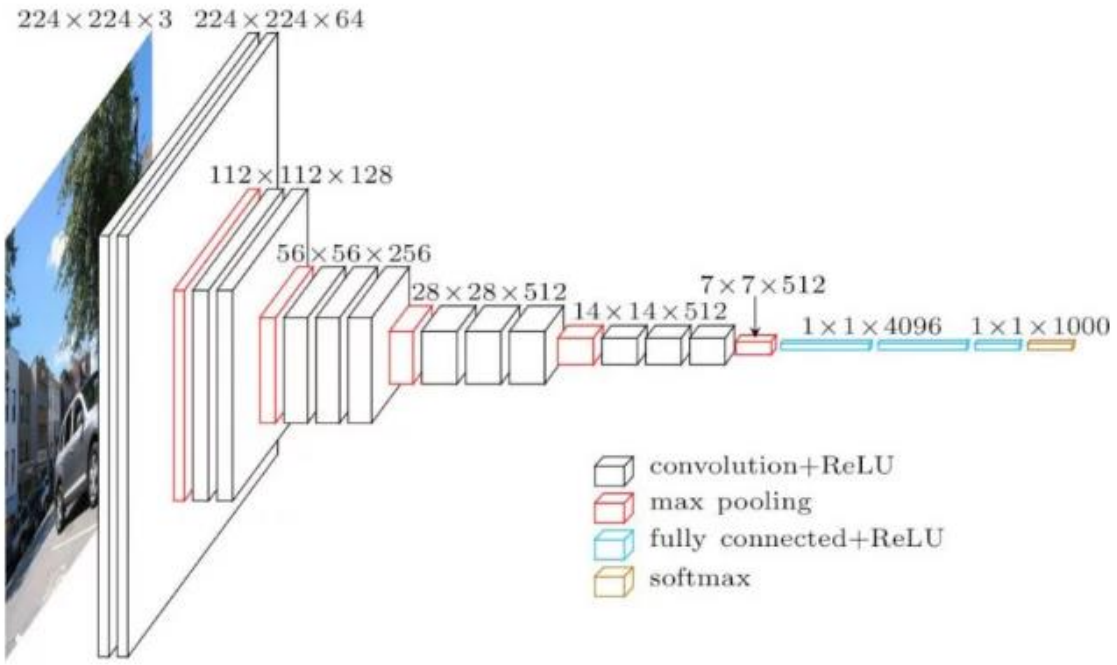

Figure 3.17 VGG-16 model architecture

The input to the Conv1 layer is a fixed-size 224 x 224 RGB image. The image is passed through a stack of convolutional layers where filters have a very small receptive field: 3×3 (left/right, up/down, smallest size to capture the central concept). In one of the configurations, it also uses 1×1 convolution filters that can be seen as a linear transformation (followed by non-linearity) of the input channels. The convolution stride is fixed to 1 pixel; the spatial padding of the input layer is such that the spatial resolution is preserved after convolution, meaning the padding is 1 pixel for 3×3 transformation layers. Spatial pooling is carried out by five max-pooling layers following some of the transformation layers (not all transformation layers are followed by max-pooling). Max-pooling is performed over a 2×2 pixel window with a stride of 2 pixels.

Three Fully Connected (FC) layers follow the stack of convolutional layers: the first two have 4096 channels each, and the third performs the 1000-way ILSVRC classification, hence it has 1000 channels. The last layer is the softmax layer. The configuration of fully connected layers is the same across all networks. All hidden layers are equipped with non-linear rectification (ReLU).

This code creates, trains, and evaluates a deep learning model using TensorFlow and Keras. It begins by importing necessary libraries and defining a custom f1_score function. The load_data function uses ImageDataGenerator to apply data augmentation to the training data and normalizes both training and test data. The build_model function defines a VGG16-like architecture with several convolutional and max pooling layers, followed by fully connected layers. The model is compiled with the Adam optimizer and categorical_crossentropy loss function, including accuracy, precision, recall, and f1_score as metrics.During training, callbacks like ReduceLROnPlateau and ModelCheckpoint are used to adjust the learning rate and save the best model, respectively. The model is trained using the model.fit function, tracking metrics such as accuracy, loss, precision, recall, and f1_score for both training and validation sets. After training, the model summary is printed, and the metrics are visualized using matplotlib. The trained model is saved as VGG16_deneme_2.h5, completing the process of developing, training, evaluating, and saving the convolutional neural network.

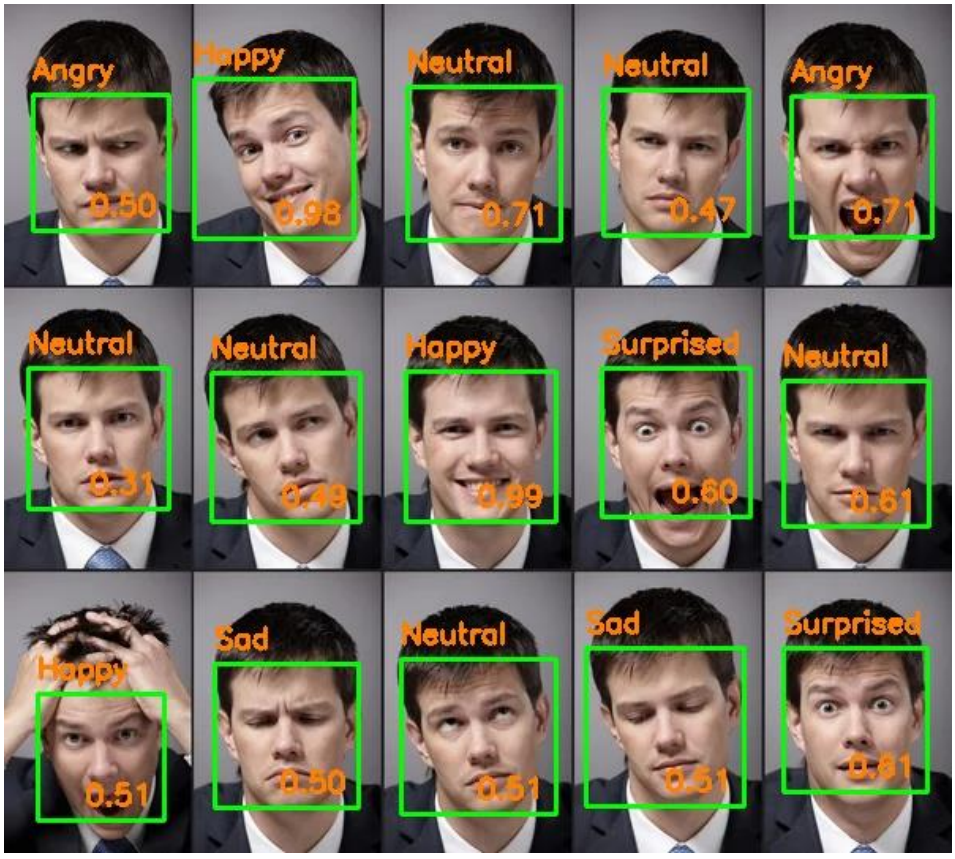

Figure 3.18 Output of VGG-16 Model

3.2.5.4 MobileNet

MobileNet is an effective deep learning model designed for mobile and embedded devices. It has an architectural structure that includes fewer parameters and calculations compared to traditional convolution, using Depthwise convolution and Pointwise convolution. In this way, it offers a light and fast architecture. MobileNet provides high classification accuracy for mobile devices and provides successful results in tasks such as image classification, object detection and segmentation. Various model variants (MobileNet-V1, MobileNet-V2, Mobilenet-V3 etc.) are offered for different accuracy and speed requirements.

MobileNet models are used in many areas such as mobile applications, embedded systems, edge computing applications, robotics and image recognition systems. It has advantages such as low computation and memory requirements, fast operating performance, high accuracy and flexible model options. Figure 3.16 illustrates the general structure of the MobileNet-V3 model.

MobileNet-V3

MobileNet-V3 is a next-generation deep learning architecture developed by Google. This model was designed to overcome the performance and efficiency limitations of previous MobileNet versions (V1 and V2). The main goal of MobileNet v3 is to provide high performance, low power consumption and small model size for mobile and embedded systems.

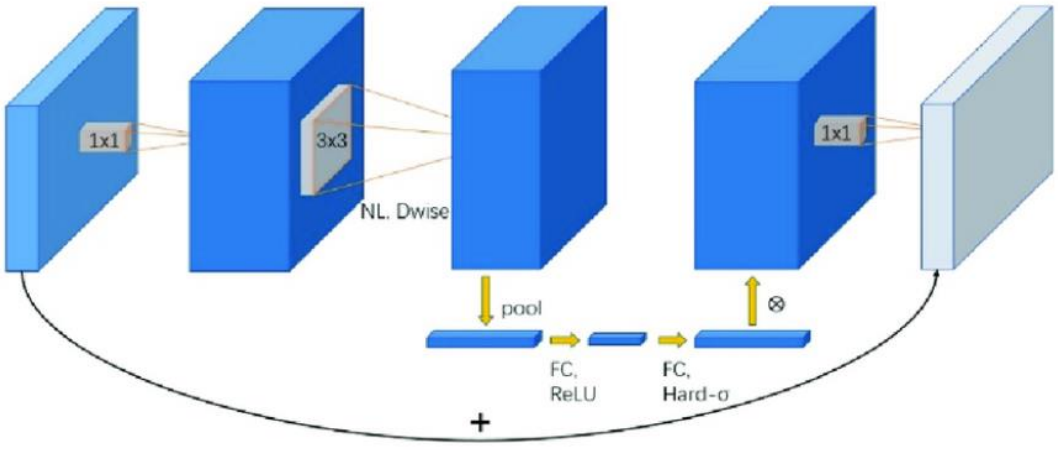

Figure 3.19 Structure of the MobileNet-V3 Model

The basic components of the MobileNet-V3 architecture consist of a series of specialized layers. The input layer starts learning the initial characteristics of the input data by applying a standard convolutional process. Then, Depthwise Separable Convolutional Layers come into play. These layers split the standard convolutional process into two separate steps: Depthwise Convolution and Pointwise Convolution. Depthwise Convolution performs convolution for each input channel separately, so that inter-channel interaction is preserved. Pointwise Convolution, on the other hand, uses the output of the Depthwise convolution to generate new feature maps, allowing information to be exchanged between channels. This two-step approach significantly reduces the computational cost and reduces the model size. Squeeze-and-Excitation Blocks increase the feature learning capacity of the model by modeling the relationships between channels. These blocks compute the importance of channels and emphasize important channels. Finally, the Hard Swish activation function provides more linear feature learning and runs faster than ReLU. All these layers come together, It unlocks MobileNet-V3's features such as high performance, low power consumption and small model size, making it effectively usable on mobile devices.

In this study, utilizes TensorFlow and Keras to implement a MobileNet-V3 model for image classification tasks. Initially, the required libraries are imported, and a custom f1_score function is defined to compute the F1 score metric based on precision and recall values. Data generators are set up to load and preprocess training and testing images, including various augmentation techniques like rotation, shifting, and flipping. The MobileNetV3Small model, pretrained on ImageNet, is instantiated and integrated into a Sequential model along with additional layers for classification. The model's top layers are unfrozen to allow fine-tuning during training.

The model is compiled with the Adam optimizer and categorical crossentropy loss function. During training, callbacks such as ReduceLROnPlateau and ModelCheckpoint are used for dynamic learning rate adjustment and model checkpointing, respectively. The training process is executed using the model.fit function, monitoring metrics such as accuracy, loss, precision, recall, and F1 score on both training and validation sets. After training, the model is saved in both .h5 and .weights.h5 formats. Finally, the model's performance metrics are visualized using matplotlib, including accuracy, loss, precision, recall, and F1 score over epochs, providing insights into the model's training progress and validation performance.

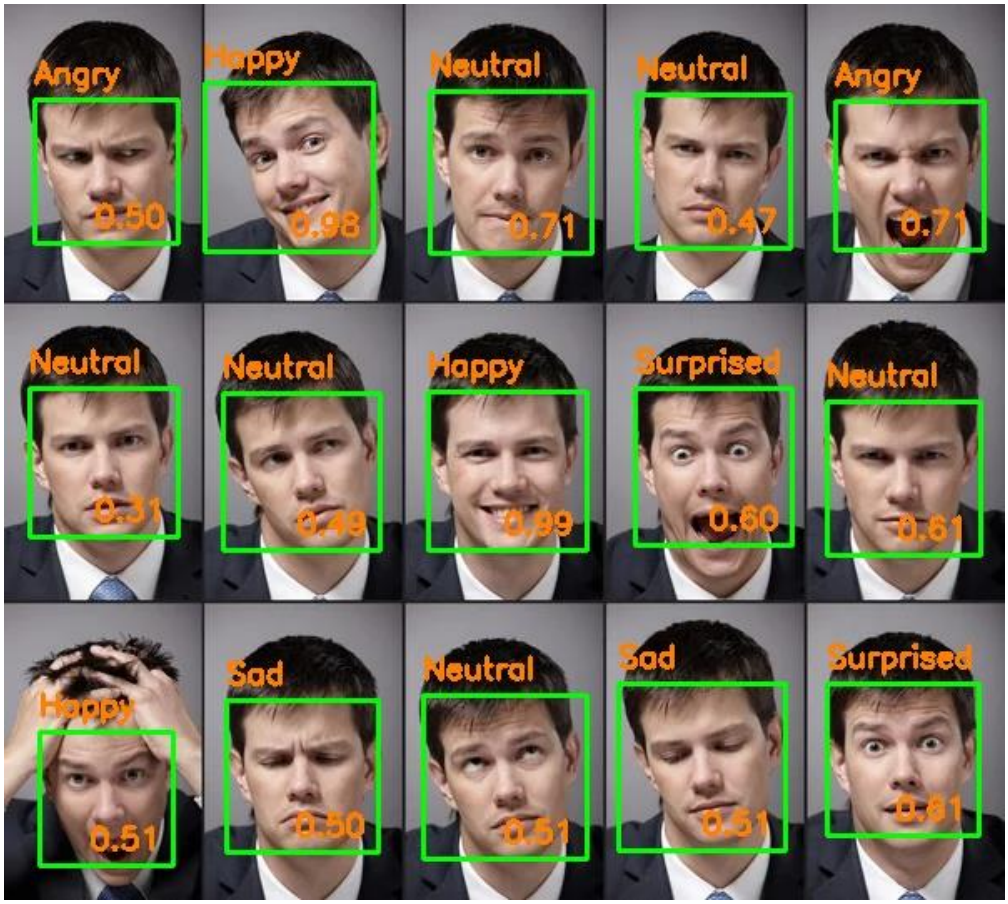

Figure 3.20 Output of MobileNet-V3 Model

3.2.6 Performance Measures

Performance metrics are used to evaluate the classifier performance of deep learning models. There are many evaluation metrics. The performance values in these metrics may be strong in some and low in others. However, it can be said that the success of a model depends not only on these criteria but also on the number of data assigned to the training and test classes. Through these metrics, models are predicted and their actual values are compared. The performance metrics used in this semester project are as follows.

3.2.6.1 Confusion Matrix:

Confusion Matrix is a matrix used to evaluate classification performance. It shows correct and misclassifications by comparing actual classes and predicted classes. This matrix allows for an in-depth analysis of the performance of the classification model.

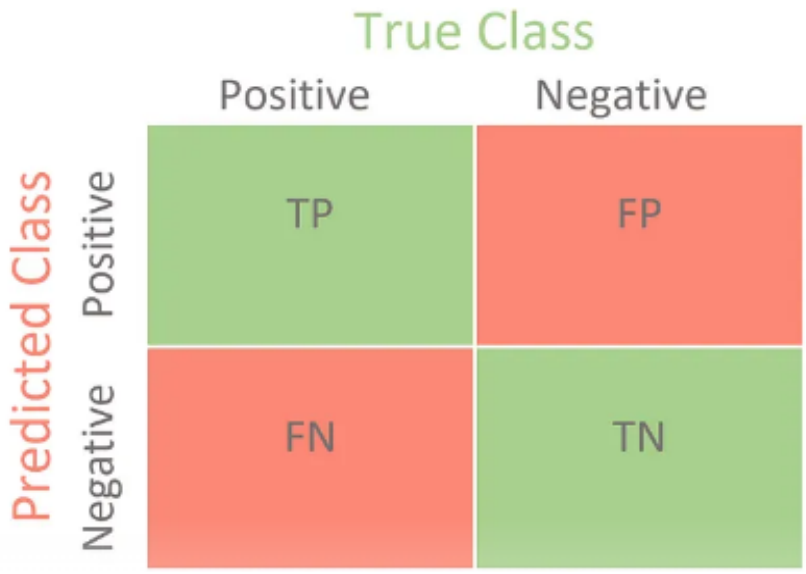

Figure 3.21 Model of Confusion Matrix

True Positives (TP): The model predicted a label and matches correctly as per ground truth.

True Negatives (TN): The model does not predict the label and is not a part of the ground truth.

False Positives (FP): The model predicted a label, but it is not a part of the ground truth (Type I Error).

False Negatives (FN): The model does not predict a label, but it is part of the ground truth. (Type II Error).

3.2.6.2 Accuracy:

Accuracy is the percentage of times the classification model correctly classifies all instances. This metric takes into account false positive and false negative examples and represents the overall model performance.

Accuracy =$(TP+TN)$ / $(TP+FP+FN+TN)$

3.2.6.3 Precision:

Precision is the proportion of positive samples that the model correctly classifies. This measure shows the impact of false positive samples and represents the reliability of the model.

Precision=$TP/(TP+FP)$

3.2.6.4 Recall :

Recall indicates how many true positive examples the model correctly classifies. This measure shows the impact of false negative examples and represents the comprehensiveness of the model.

Recall=$TP/(TP+FN)$

3.2.6.5 F1-Score:

The F1-Score is the harmonic mean of precision and sensitivity. It is a balanced indicator of precision and sensitivity and provides an overall assessment of classification performance.

F1-Score = 2 $*$ (precision * sensitivity) / (precision + sensitivity)

4. ACTIVITY- TIME TABLE

ACTIVITY	MONTHS				
	February	March	April	May	June
Determining the topic					
Source research					
Creation of graduation project content					
Application of models for the graduation project					
Interpretation of graduation project models and statistical analysis					
Writing the first two chapters determined in the content of the graduation project					
Writing sections related to the model in the final project content					
Completion of all sections specified in the final assignment content					

Table 4.1 Activity-Time Table

5. FINDINGS

The emotional state of living beings is evident not only through verbal expressions, but also through facial expressions. A person's emotional state is revealed throughout the body, reflecting the mood of the moment. Understanding human emotions through speech alone is often a challenging and time-consuming process. Nowadays, with the rapid development of technology, data has become increasingly complex and this complexity has led to intensive studies on artificial intelligence (AI), image processing and sentiment analysis.

In the research for recognizing human emotions, face recognition analysis has one of the most extensive fields. Humankind is conducting research in a wide range of areas, from healthcare to smart living, from autism and spectrum disorders to schizophrenia, especially in the automatic detection of emotions. In this field, facial emotion recognition in particular is responsible for performing emotion analysis between facial expressions. Facial emotion analysis takes place based on factors such as perception of the face, changes, blinks and size variations. In this semester project work, emotion analysis from facial image expressions has been carried out.

In this study, the focus is on detecting emotions from facial expressions. In particular, six basic emotions are emphasized. Emotion detection analysis was performed with deep learning methods using Resnet-50, Traditional CNN, VGG-16 and MobileNet-V3 learning methods from CNN methods.

These deep learning models have common architectures that are frequently used for tasks such as image classification and feature extraction. These models usually have a multilayer structure and include convolution-based layers. Convolution layers learn and extract important features by applying filters on images, which enables the model to recognize patterns and features in visual data. Furthermore, the activation functions often used in these models, especially ReLU, enhance the learning ability of the network. Pooling layers reduce the size of feature maps, preventing over-learning and reducing computational cost. The last layers are usually fully connected layers and are used for classification.

These models are usually trained based on transfer learning principles and adapt to the training data using various optimization algorithms. These features make deep learning models such as ResNet-50, VGG-16, Classic CNN and MobileNet effective and powerful, making them the preferred choices in many visual recognition and classification tasks.

The metrics 'accuracy', 'loss', 'precision', 'recall', 'AUC', 'f1_score' were used to evaluate the performance of the models. The hyperparameters of the models are shown in Table 5.1.

Hiperparametre	**Value**
batch_size	64
epoch	45
Metrics	'accuracy', 'loss', 'precision', 'recall', 'AUC', 'f1_score'
optimizer	Adam
Learning_rate	0.001
Loss function	CategoricalCrossentropy

Table 5.1 Hyperparameters

Traditional CNN, Resnet-50, VGG-16 and MobileNet-V1 models were used with 80%-20% training-test ratios. The dataset named FER2013 was used to test the models. This dataset was downloaded from Kaggle database. The dataset consists of 48x48 pixel color images. The dataset is a hard benchmark set used for emotion recognition of facial expressions.

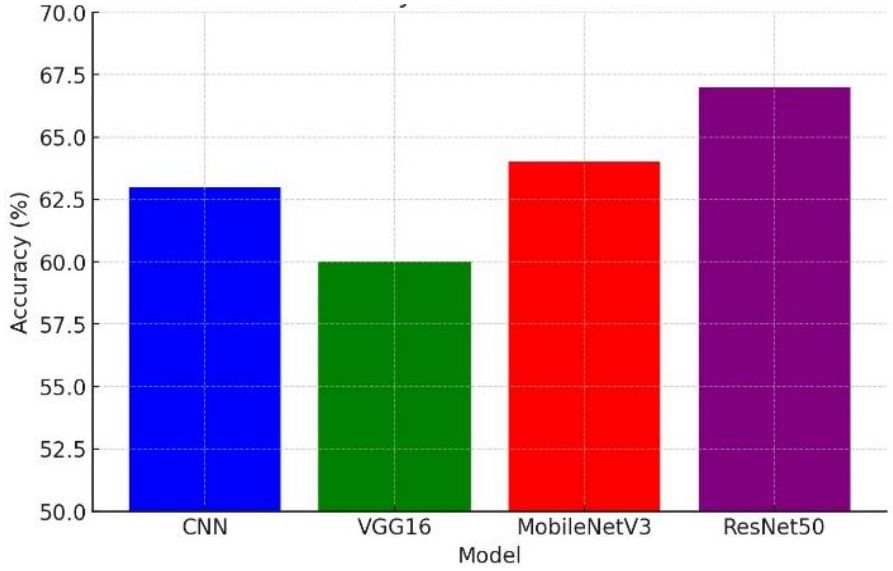

Figure 5.1 Accuracy of Models

Looking at Figure 5.1, the highest success rate for model accuracy is observed with the ResNet50 model as “0.67”. Looking at the success graphs of CNN models, it is seen that the most unsuccessful model is VGG-16.

The success graphs of the models are given in Figure 5.2 according to iteration values. In addition, the graphs of the loss values of the models are shown in Figure 5.3.

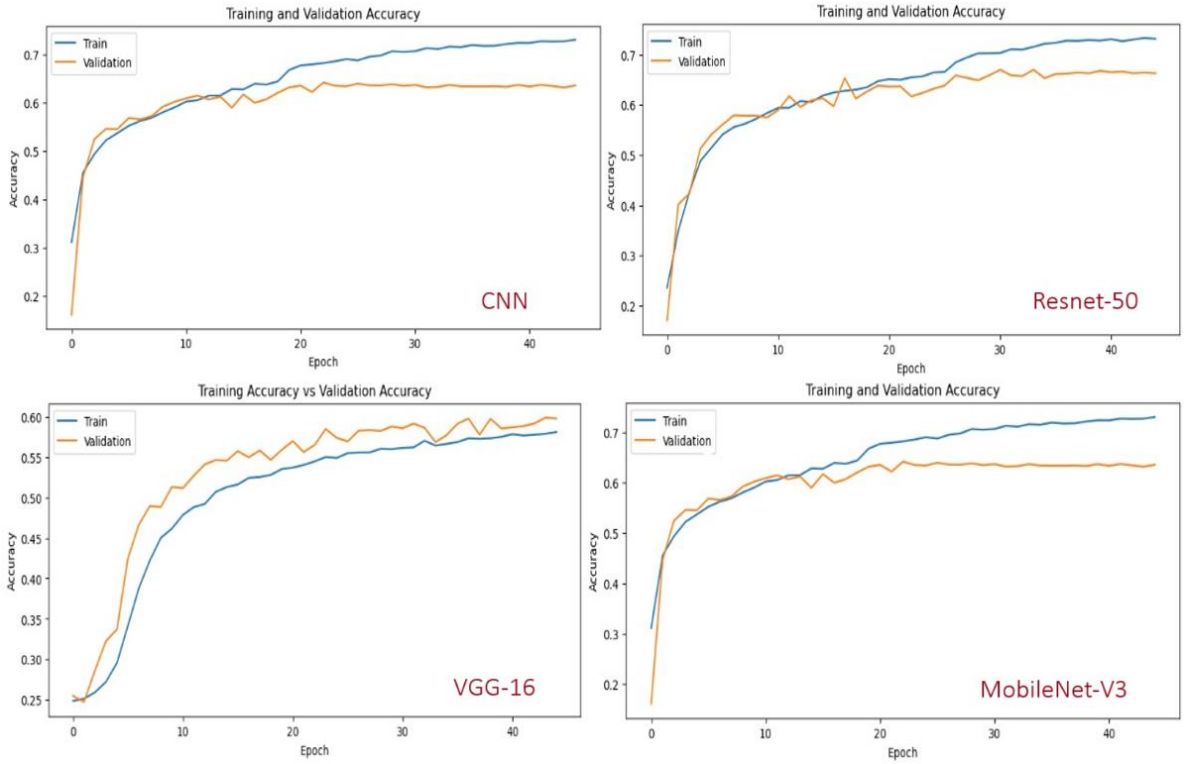

Figure 5.2. Accuracy Value of Models

In the Resnet Graph, the training accuracy has exceeded 0.7 and the verification accuracy has stabilized at approximately 0.68. This model offers a higher verification accuracy than others. Since the ResNet50 Accuracy graph offers the highest verification accuracy value, the ResNet model performed better than other models according to this graph. This model offers better generalization and accuracy.

The training loss for CNN continuously decreases while the validation loss seems to stabilize around the 20th epoch. The training loss is lower than the validation loss. For VGG, both training and validation losses are close to each other, around 1.2, but the validation loss is slightly lower than the training loss. The training loss for MobileNet continuously decreases, and the validation loss also seems to stabilize around the 20th epoch. The training loss is lower than the validation loss. For ResNet, both training and validation losses are very close to each other and stabilize around 1.0. The ResNet loss graph shows the best results compared to the other models. The close alignment of training and validation losses, along with the low loss values around 1.0, indicates that the model generalizes well and does not overfit. Therefore, the ResNet model demonstrates the best performance in terms of loss.

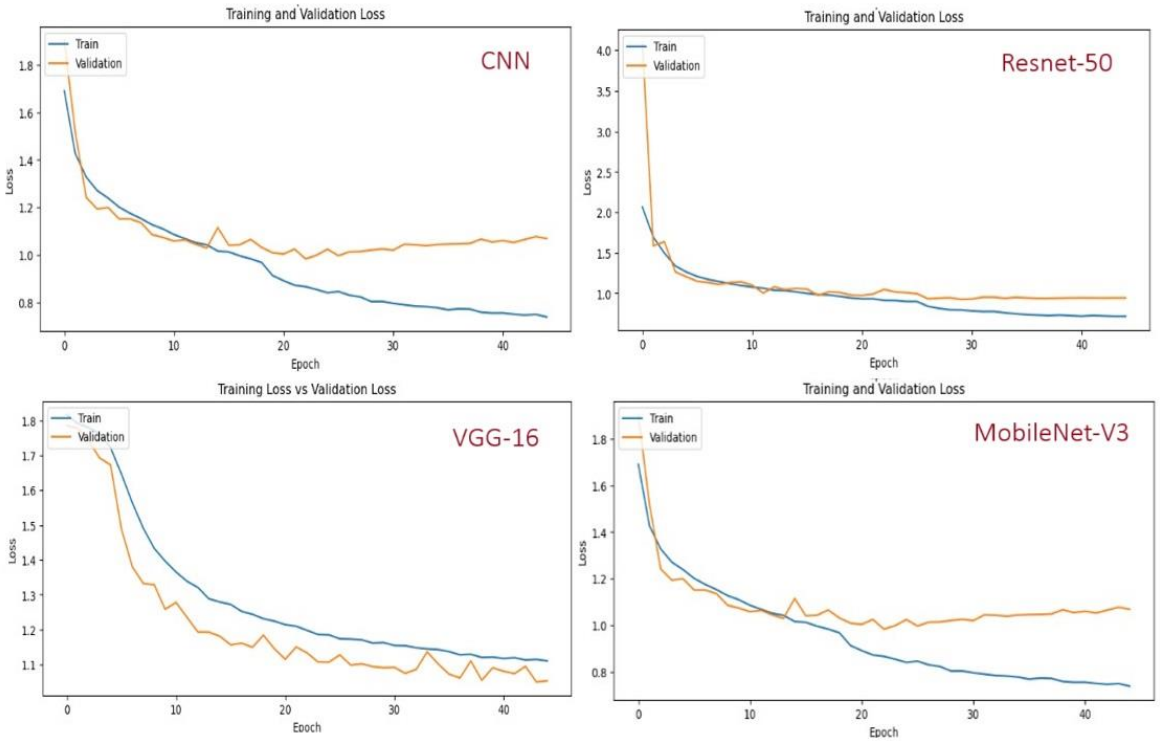

Figure 5.3. Loss Value of Models

The Receiver Operating Characteristic (ROC) curve is a visualization tool used to evaluate the performance of classification models in detail. By examining the trade-off between a model's true positive rate and false positive rate, this curve shows how the model performs at different classification thresholds. The ROC curve is often used to assess the classification accuracy of a model, as well as to better understand the behavior of the model, especially in data sets with unbalanced class distributions. The ROC curve plots a curve showing the sensitivity and specificity values obtained by a model under each threshold value. The closer the curve is to the upper left corner, the better the performance of the model. In this case, both the true positive rate and false positive rate are high. As the curve approaches the lower left corner, the performance of the model decreases because the sensitivity decreases while the specificity increases.

The AUC (Area Under the ROC Curve) value refers to the area under the ROC curve and is often used to measure the classification performance of the model with a single number. The closer the AUC value is to 1, the better the performance of the model. ROC analysis provides a comprehensive evaluation of a model's performance metrics such as accuracy, precision and specificity, especially in areas such as medical diagnostics, machine learning and fraud detection.

For CNN AUC-ROC, the lowest AUC value is 0.83 for "fearful," and the highest AUC value is 0.97 for "happy." For VGG AUC-ROC, the lowest AUC value is 0.78 for "fearful," and the highest AUC value is 0.96 for "happy." For MobileNet AUC-ROC, the lowest AUC value is 0.81 for "fearful," and the highest AUC value is 0.97 for "happy." For ResNet AUC-ROC, the lowest AUC value is 0.84 for "fearful," and the highest AUC value is 0.98 for "happy." The ResNet AUC-ROC graph shows the best results compared to the other models as it presents the highest AUC values. The ResNet model demonstrates superior classification performance by showing higher or equal AUC values across all categories. Notably, with the highest AUC value of 0.98 in the "happy" category, it shows the best performance.

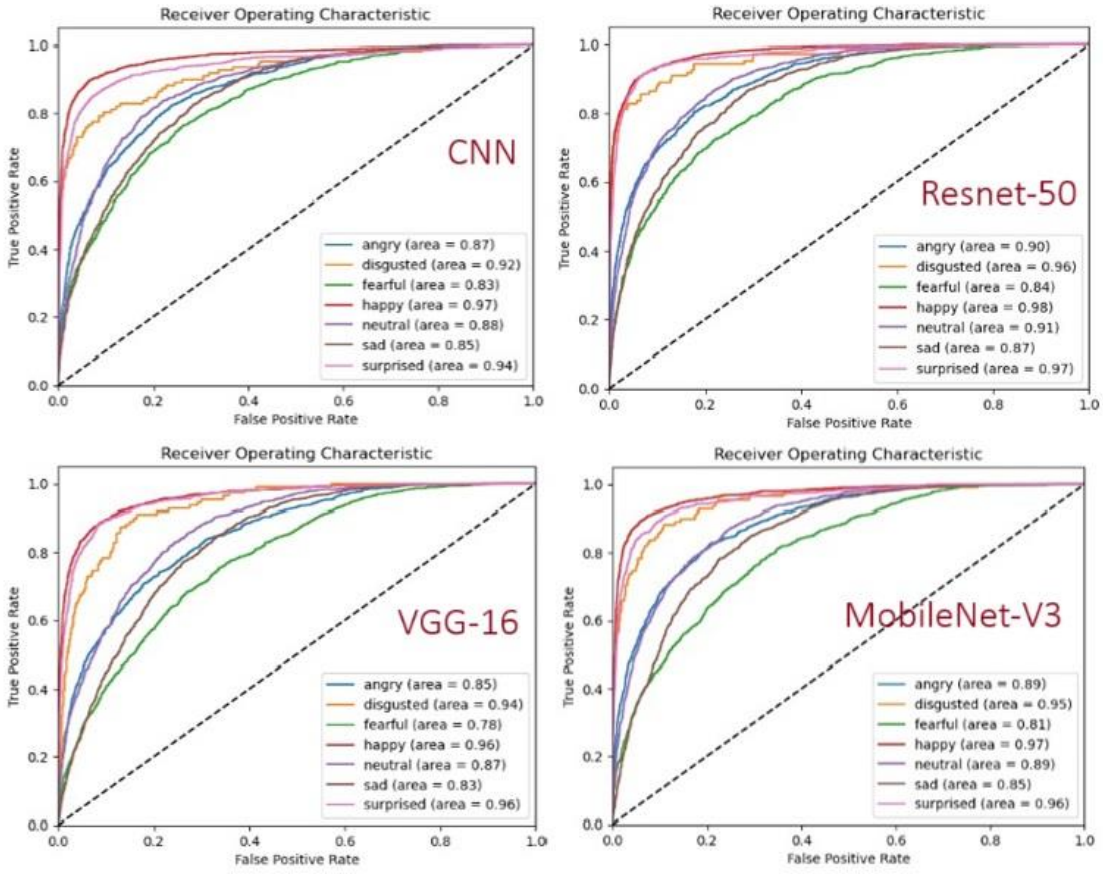

Figure 5.4. ROC Curves of Models

Precision is defined as a critical metric that evaluates the performance of classification models and generally measures how many of the samples predicted as positive are actually positive. This metric aims to minimize the model's false positive predictions (False Positives) and is therefore particularly important in scenarios where false alarms are costly or undesirable.

Precision represents the proportion of True Positives to total positive predictions, and a high precision value indicates that the instances that the model predicts as positive are in fact mostly positive. This implies that the model has a low false positive rate, thus emphasizing the model's ability to reliably predict positives. However, precision alone may not be a sufficient measure of performance, so it is often evaluated in combination with other metrics such as recall, F1-score and ROC curve. High precision implies that the model tends to correctly recognize classes, but this value should be interpreted in balance with other metrics. Presicion plots of the models are given in Figure 5.5.

The CNN precision shows a difference between the training and validation precision values, with the validation precision value stabilizing around 0.7. For the VGG precision, the training and validation precision values are quite close, both stabilizing around 0.7-0.75. In the case of MobileNet precision, the training precision value is close to 0.8, while the validation precision value stabilizes around 0.7. The ResNet precision shows that the training and validation precision values are quite close and stabilize around 0.7-0.75. When we look at the VGG precision and ResNet precision graphs, we see that the training and validation precision values of both models are very close to each other and remain stable. This indicates that the models generalize well and do not overfit.

However, the VGG precision graph stands out as the model with the least difference between the training and validation precision values. Therefore, the VGG precision graph provides better results compared to the others, as the model's training and validation performance is consistent, offering more reliable results in terms of the model's overall performance.

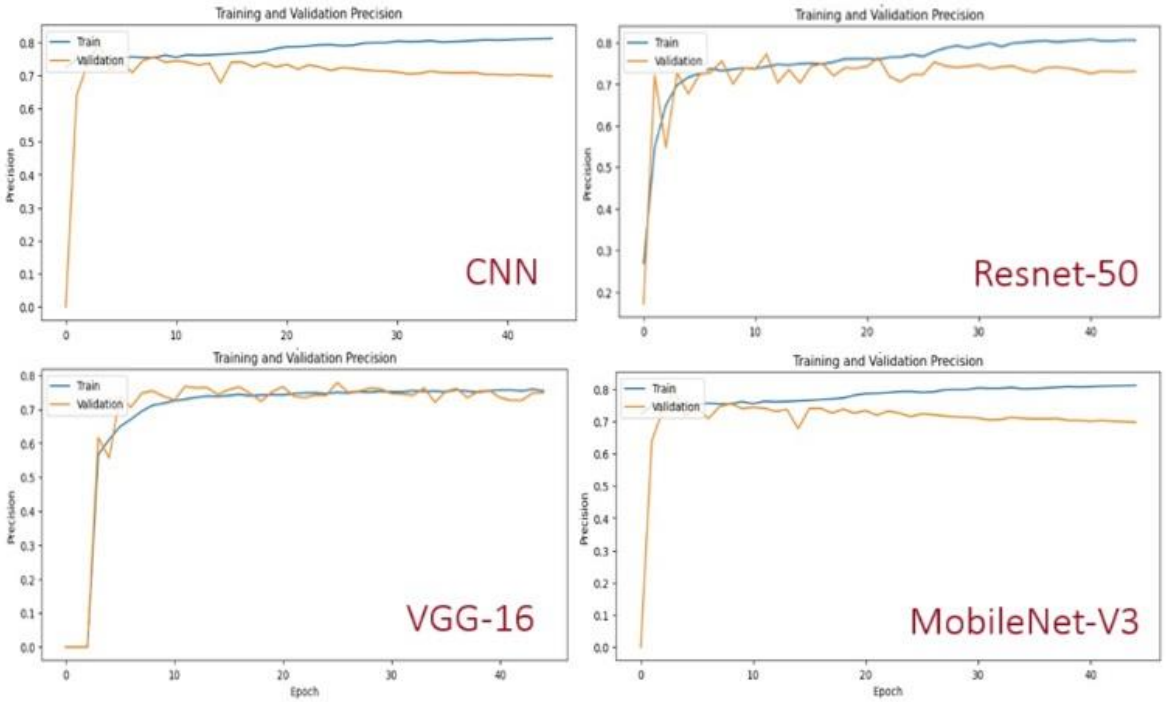

Figure 5.5 Presicion Value of Models

Recall is considered an important metric for evaluating the performance of classification models and usually measures the proportion of true positive cases out of the total positive cases. A high recall value indicates the ability of the model to accurately detect true positives, which is especially important if the model does not miss positive cases, i.e. has a low false negative rate. However, an increase in recall usually leads to a decrease in precision, so these two metrics are in a trade-off relationship against each other. Recall is often important in applications that require precision, such as medical diagnostics and emergencies, and is used in conjunction with other performance metrics to understand the overall effectiveness of a model. The sensitivity plots of the models are given in Figure 5.6.

The CNN recall values for both training and validation are around 0.6, with validation values slightly lower than the training ones. In contrast, the VGG recall values show a notable difference, with validation recalls stabilizing around 0.4, significantly higher than the training recalls. On the other hand, both MobileNet and ResNet have training and validation recall values around 0.6, showing a close similarity between the two sets.

Analyzing the MobileNet and ResNet recall graphs, it is evident that both models exhibit minimal disparity between their training and validation recall values. This similarity indicates that both models generalize well and avoid overfitting, reflecting their effectiveness in capturing the underlying patterns in the data.

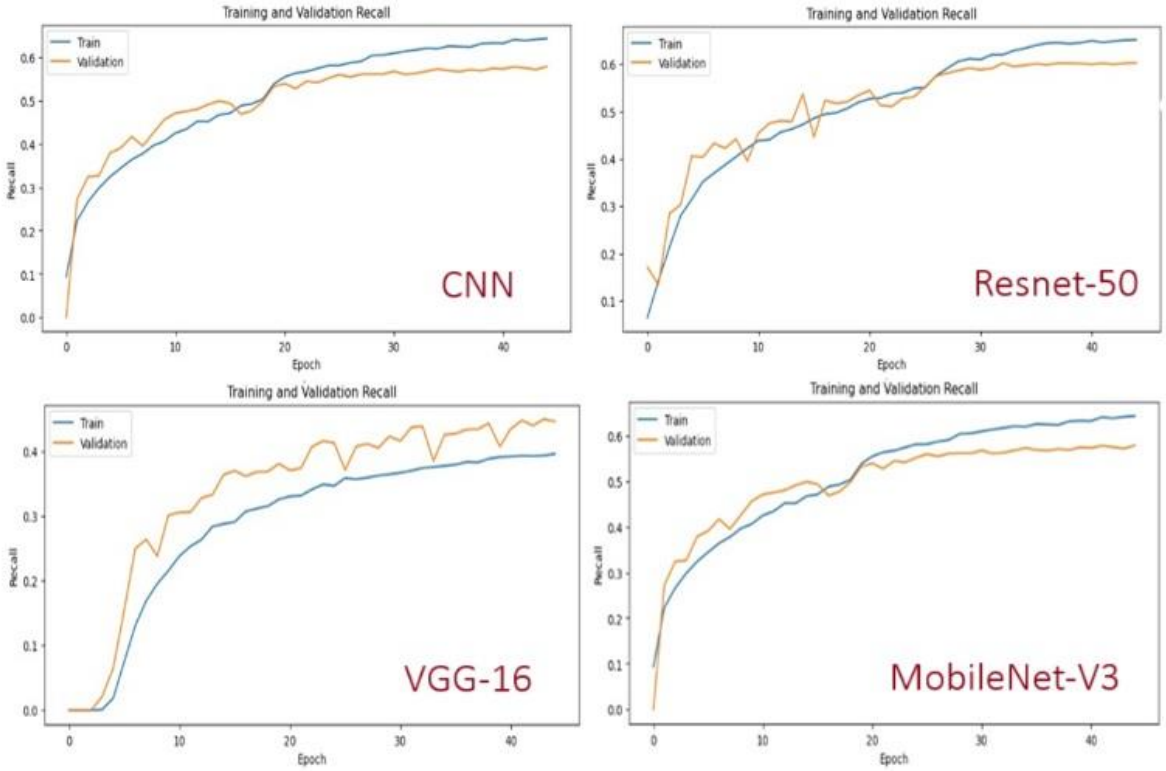

Figure 5.6 Recall Value of Models

F1-score is a metric used to evaluate the performance of a classification model and represents the harmonic mean of the precision and recall metrics. This metric balances both the model's ability to minimize false positives and its ability to minimize false negatives. The F1-score is important to ensure that the model is both accurate and focused, especially in datasets with unbalanced class distributions. A high F1-score value indicates cases where the model effectively optimizes both precision and recall, which implies that the overall performance of the model is balanced. F1-score is preferred to evaluate the model's ability to successfully recognize positive and negative classes, especially in areas such as medical diagnostics, information security and fraud detection. The value plots of the models are given in Figure 5.7.

ResNet F1 Score graph gave the best results compared to other models. The fact that the training and validation F1 scores are close to each other indicates that the model generalizes well and does not overfit. Therefore, the ResNet model shows the best performance.

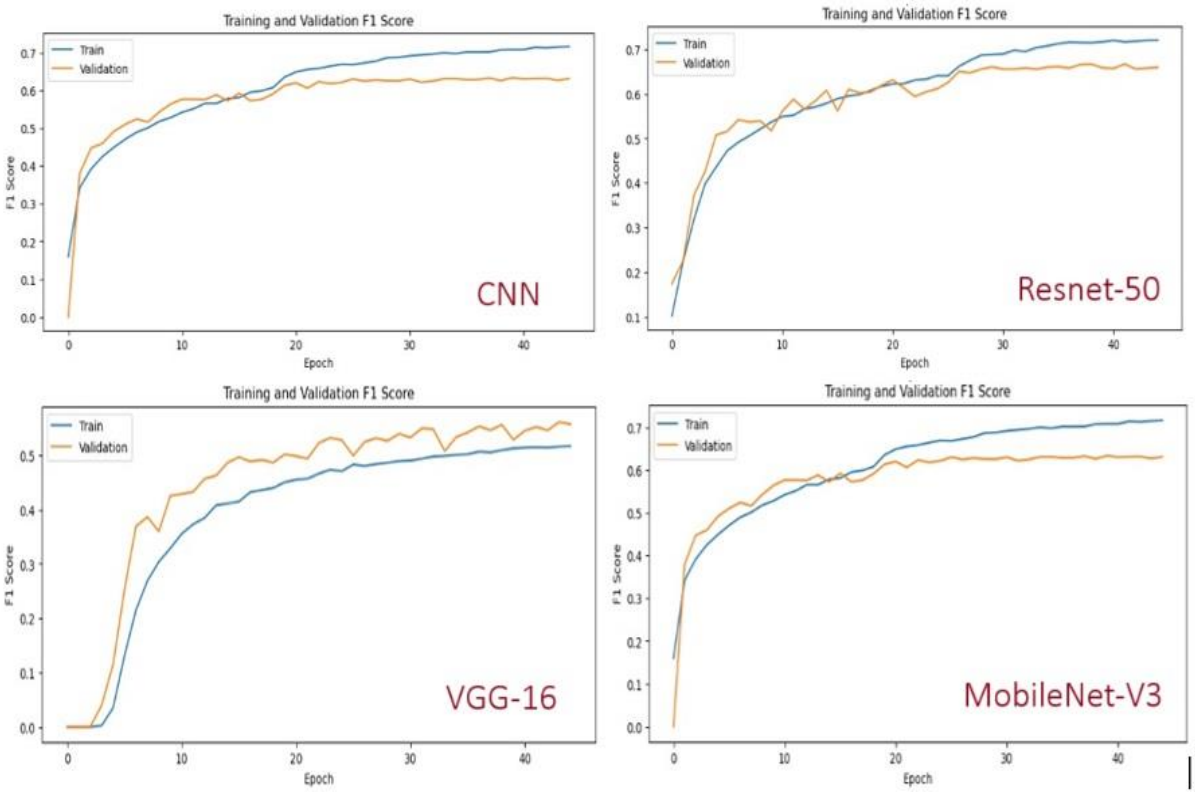

Figure 5.7 F1-Score Value of Models

There are 7,177 images for the FER2013 dataset in the 20% test set that the CNN structure never sees. Therefore, the confusion matrix obtained over 140 images for each emotion is shown in Figure 5.8. As can be seen from the figure, the most successful results are obtained for happy (87%), disgusted (82%), surprised (78%), angry (60%), sad and neutral (57%), and the least successful results are obtained for fearful (56%).

The most confused emotion classes were fearful-sad and neutral-sad with 17%. The least confused emotion classes were sad-disgusted, disgust-fearful, disgust-neutral and neutral-surprised with 1%.

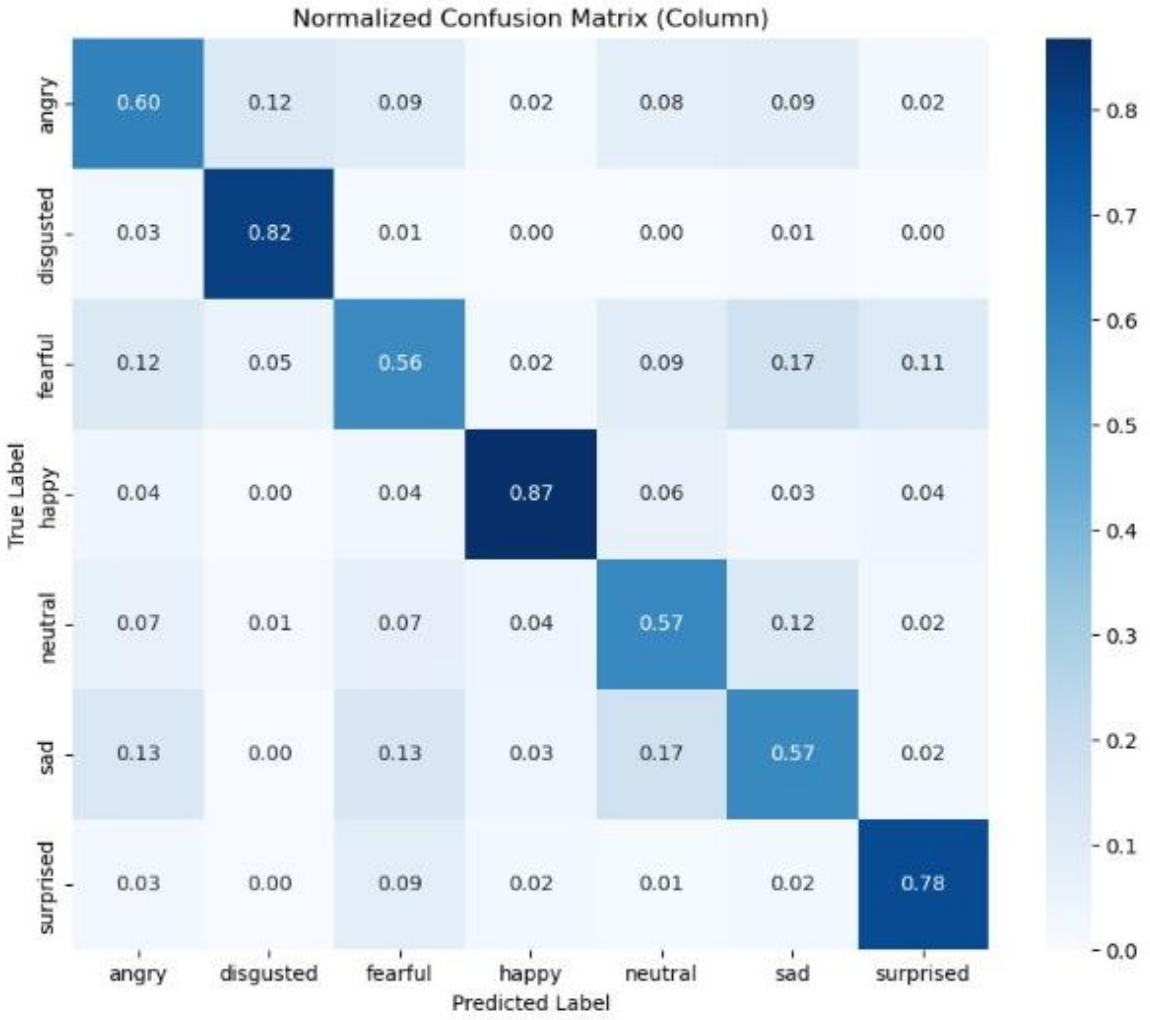

Figure 5.8 Confusion Matrix of Resnet-50 Model

6. DISCUSSION

In this study, the critical role of facial expressions in understanding emotions is emphasized. Facial expressions are important indicators that reflect individuals' inner worlds and psychological states. Emotions help people understand their inner world and relate to the outside world, and therefore are of great importance in human psychology. Facial expressions, tone of voice, gaze, movements and gestures play a critical role in communicating these emotions. Although they vary across different genders, ages, cultures and races, facial expressions have universal characteristics that are generally given to the same emotions. The psychological impact of facial expressions plays a critical role in identifying, expressing, sharing and understanding emotions. In this context, emotion detection from facial expressions makes it possible to better understand the emotional states of both individuals and communities and to develop interventions based on this understanding.

In this context, emotional intelligence and emotion recognition technologies play an important role in enriching and deepening human-machine interaction. Today, artificial intelligence algorithms have advanced emotion recognition capabilities. These algorithms can be trained to understand and interpret emotions in various modalities such as facial expressions, voice tone, text analysis. In this way, machines have the capacity to form similar emotional bonds with humans (Rázuri, et al. 2013). Emotional intelligence can be defined as the ability of machines to establish an emotional connection with humans, going beyond the ability to simply perform tasks. At this point, deep learning models play an important role in training emotion recognition algorithms. Learning models such as ResNet can improve emotion recognition performance by utilizing knowledge learned from large data sets in similar tasks (Akhand, et al. 2021).

This semester project work focused on seven basic emotions in particular, addressing the merits of emotion recognition. These emotions include happiness, sadness, normal, fear, anger, disgust and surprise. The success values obtained were achieved with the deep learning methods used and the effectiveness of these methods in emotion recognition performance was evaluated.

The extensive literature review reveals that emotion recognition technology can be used in many areas from healthcare to security applications, from education to human-machine interaction. Especially in the healthcare sector, emotion recognition systems can play an important role in monitoring the emotional state of patients and early detection of conditions such as anxiety and depression. The deep learning methods used in the training and testing phases were evaluated over different CNN methods and the results obtained were presented through charts and confusion matrix. These results show that deep learning methods improve the performance of emotion recognition, especially when the knowledge learned from large datasets is used in similar tasks.

In conclusion, this study has highlighted the future potential of artificial intelligence by addressing its emotion recognition capabilities. AI's advances in emotional intelligence can make human-machine interaction richer and more meaningful. Research on recognizing emotions through human gestures and facial expressions has become a major focus of interest in artificial intelligence (AI), image processing and sentiment analysis. Understanding human emotional expressions is often a complex and time-consuming process, but technological advances have made it more efficient and faster (Akçelik, et al. 2021).

Developing technology, especially innovations in artificial intelligence and deep learning, have led to the emergence of various applications in emotional intelligence. In this study, Traditional CNN, Resnet-50, VGG-16 and MobileNet-V1 deep learning models are used for emotion recognition.

The common function of these models is to perform recognition, classification and other visual processing tasks by extracting complex features from visual data with high accuracy and generalization capability. The main objective of this study is to evaluate the emotion recognition performance of the four models.

The hyperparameters used in the training and testing phases of the models were carefully selected and determined through experiments. “Adam” was used as the optimization algorithm, the learning rate was set to “0.001” and the Categorical Crossentropy loss function was preferred. The dataset used in the training and testing phases is taken from FER2013, a challenging benchmark set for emotion recognition. The performance values obtained for the training set were evaluated with various metrics, in particular success, precision, sensitivity, ROC curve and F1-Measure (Table 5.1). The highest success rate among the models used was ResNet50 with “0.68”.

These results show that ResNet50 is more successful in the emotion recognition task compared to other models. In addition to performance metrics, ROC curve analysis was used to evaluate the classification performance of the models in detail. Looking at the ROC curves, it is seen that ResNet50 has the best performance for both training and test datasets. The lowest performance was observed for the MobileNet-V3 model. When metrics such as precision, recall and F1-score are also examined, it is seen that ResNet50 has the highest values of 68%. These metrics play an important role in evaluating the overall performance of the model by balancing the false positive and false negative rates.

In conclusion, this study evaluated the performance of CNN models in the field of sentiment recognition and revealed that especially ResNet50 is more effective compared to other models.

7. CONCLUSION

The main goal of this study is to evaluate emotion recognition capabilities using deep learning techniques. Understanding emotional expressions plays an important role in human-machine interaction and technological advances in this field are increasing the impact of emotion recognition applications. In this study, emotion analysis is performed using deep learning models. The common function of these models is to perform recognition, classification and other visual processing tasks by extracting complex features from visual data with high accuracy and generalization capability. The main objective of this study is to evaluate the emotion recognition performance of the four models used.

The study evaluates the emotion recognition performance of these models. The dataset used in the training and testing phases is taken from Fer2013, a challenging benchmark set for emotion recognition. The performance of the models is evaluated with various metrics such as performance, precision, sensitivity, ROC curve and F1-Measure.

The results show that especially the Resnet-50 model has a higher success rate compared to the other models. One of the most important factors affecting the success rate will be the architecture and complexity of each model. Each model has a different architecture, which can lead to different feature extraction capabilities and overall performance. For example, deeper models such as ResNet-50 and VGG-16 generally perform better, while the lighter models Traditional CNN and MobileNet-V3 may have lower accuracy values.

One limitation of this study is that the distribution of images for each emotion in the dataset is very uneven. This makes it difficult to adapt deep learning models to the dataset. Another limitation is that the images in the dataset are 48x48 pixels and the images are colorless. This shows that the images do not contain enough information to express emotions.

8. SOURCES

Taş, S. (Ocak-2024). Yüz İfadelerinden Duygu Tespiti: Aktarım Derin Öğrenme Metotları İle. Yüksek Lisans Tezi, Batman Üniversitesi, Bilgi Teknolojileri Anabilim Dalı, Lisansüstü Eğitim Enstitüsü, Batman.

Akar, F., & Akgül, İ. (2022). Emotion Recognition from Facial Expressions by Deep Learning Model. Iğdır Üniversitesi Fen Bilimleri Enstitüsü Dergisi, 12(1), 69-79. DOI: 10.21597/jist.976577

Safalı, Y., & Avaroğlu, E. (2021). Derin Öğrenme ile Yüz Tanıma ve Duygu Analizi. Mersin Üniversitesi, Mühendislik Fakültesi, Bilgisayar Mühendisliği Bölümü, Mersin, Türkiye. DOI: 10.31590/ejosat.1010450

Rázuri, JG, Sundgren, D., Rahmani, R., &Cardenas, AM (2013, Kasım). Yapay sinir ağına dayalı birleştirilmiş görüntülerde yüz ifadesi analizi yoluyla otomatik duygu tanıma. 2013 yılında 12. Meksika Uluslararası Yapay Zekâ Konferansı (s. 85- 96). IEEE

Ananthram, A., Saravanakumar, K. K., Huynh, J., & Beigi, H. (2020). Multi-modal emotiondetection with transfer learning. arXivpreprint arXiv:2011.07065.

Lawrence, K., Campbell, R. ve Skuse, D. (2015). Age, gender, and puberty influence the development of facial emotion recognition. Frontiers in Psychology, 6(June), 1–14. doi:10.3389/fpsyg.2015.00761

Khaireddin, Y. ve Chen, Z. (2021). Facial Emotion Recognition: State of the Art Performance on FER2013. http://arxiv.org/abs/2105.03588 adresinden erişildi

Wei, C. Z. (2013). Stress emotion recognition based on RSP and EMG signals. Advanced Materials Research, 709, 827–831. doi:10.4028/www.scientific.net/AMR.709.827

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 770-778)

Hartling, C., Fan, Y., Weigand, A., Trilla, I., Gärtner, M., Bajbouj, M., … Grimm, S. (2019). Interaction of HPA axis genetics and early life stress shapes emotion recognition in healthy adults. Psychoneuroendocrinology, 99, 28–37. doi:10.1016/j.psyneuen.2018.08.030

Akhand, MAH, Roy, S., Siddique, N., Kamal, MAS ve Shimamura, T. (2021). Derin CNN'de transfer öğrenmeyi kullanarak yüz duygu tanıma. Elektronik, 10 (9), 1036.

Ananthram, A., Saravanakumar, K. K., Huynh, J., & Beigi, H. (2020). Multi-modal emotiondetection with transfer learning. arXivpreprint arXiv:2011.07065.

Khaireddin, Y. ve Chen, Z. (2021). Facial Emotion Recognition: State of the Art Performance on FER2013. http://arxiv.org/abs/2105.03588 adresinden erişildi.

Hartling, C., Fan, Y., Weigand, A., Trilla, I., Gärtner, M., Bajbouj, M., … Grimm, S. (2019). Interaction of HPA axis genetics and early life stress shapes emotion recognition in healthy adults. Psychoneuroendocrinology, 99, 28–37. doi:10.1016/j.psyneuen.2018.08.030

Garcia, A., Orts-Escolano, S., Oprea, S., Villena-Martinez, V., Martinez Gonzalez, P., & Garcia-Rodriguez, J. (2018). A survey on deep learning techniques for image and video semantic segmentation. AppliedSoft Computing, 70, 41-65

Dehghan, A., Ortiz, E. G., Shu, G. ve Masood, S. Z. (2017). DAGER: Deep Age, Gender and Emotion Recognition Using Convolutional Neural Network. http://arxiv.org/abs/1702.04280 adresinden erişildi.

Ray, A. ve Chakrabarti, A. (2016). Teknolojinin tasarımı ve uygulanması, biyo-fiziksel ve yüz ifadesinin füzyonunu kullanarak etkili öğrenmeyi mümkün kıldı. Journal of Education Technology&Society, 19 (4), 112-125.

Pantic, M. Machine analysis of facial behaviour: naturalistic and dynamic. Marian, DE ve Shimamura, AP (2013). Dinamik yüz ifadeleri üzerindeki bağlamsal etkiler. Amerikan Psikoloji Dergisi, 126 (1), 53-66

"New theory cracks open the black box of deep neural networks." (2017). Wired. https://www.wired.com/story/new-theory-deep-learning/ [Accessed 8 March 2018].

Cengil, E., Çinar, A., & Güler, Z. (2017, September). A GPU-based convolutional neural network approach for image classification. In 2017 International Artificial Intelligence and Data Processing Symposium (IDAP) (pp. 1-6). IEEE.

S. Lawrence, C. L. Giles and Ah Chung Tsoi, "Convolutional neural networks for face recognition," Proceedings CVPR IEEE Computer Society Conference on Computer Vision and Pattern Recognition, San Francisco, CA, USA, 1996, pp. 217-222

Harman, Samed. "Makine Öğrenmesi | Ördü Kader Ağlarını Bölüm 2— Evrişimli Sinir Ağları." Medium. https://medium.com/@samedharman/makine-öğrenmesi-ördü-kader-ağlarını-bölüm-2-evrişimli-sinir-ağları

alfo1995. "CNN--convolutional-neural-network-image-recognition." GitHub. https://github.com/alfo1995/CNN--convolutional-neural-network-image-recognition

Stewart, Matthew, PhD. "Simple Introduction to Convolutional Neural Networks." Towards Data Science. https://towardsdatascience.com/simple-introduction-to-convolutional-neural-networks-cdf8d3077bac

Sharma, S., Sharma, S. ve Athaiya, A. (2017). Sinir ağlarında aktivasyon fonksiyonları. Veri Bilimine Doğru, 6 (12), 310-316

"Sigmoid activation function and its derivative." ResearchGate. https://www.researchgate.net/figure/Sigmoid-activation-function-and-its-derivative_fig6_358908601

"Activation Functions in Neural Networks." SuperAnnotate. https://www.superannotate.com/blog/activation-functions-in-neural-networks

Türkoğlu, M., Hanbay, K., Sivrikaya, I. S., ve Hanbay, D. (2021). Derin evrişimsel sinir ağı kullanılarak kayısı hastalıklarının sınıflandırılması. Bitlis Eren Üniversitesi Fen Bilimleri Dergisi, 9(1), 334-345.

Gözütok, Hilal. "Evrişimli Sinir Ağları (Convolutional Neural Networks - CNN)." Medium. https://hilalgozutok.medium.com/evri%C5%9Fimli-sinir-a%C4%9Flar%C4%B1-convolutional-neural-networks-cnn-e61470e9bdb1

"The Annotated ResNet-50" by Géron, Aurélien. Towards Data Science. https://towardsdatascience.com/the-annotated-resnet-50-a6c536034758

"VGG16 Derin Öğrenme Ağı", Neurohive, https://neurohive.io/en/popular-networks/vgg16/

https://www.researchgate.net/figure/Illustration-of-the-structure-of-MobileNetv3-block_fig2_365884327

Tan, M., Chen, B., Pang, R., Vasudevan, V., Sandler, M., Howard, A., & Le, Q. V. (2019). MobileNetV3: Searching for Optimal Neural Architectures for Mobile and Resource-Constrained Devices. 2019 IEEE/CVF International Conference on Computer Vision (ICCV), 1314–1324. [Şekil 2. MobileNetv3 blokunun yapısının illüstrasyonu]. https://doi.org/10.1109/ICCV.2019.00140