BRNO UNIVERSITY OF TECHNOLOGY
VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

FACULTY OF INFORMATION TECHNOLOGY
FAKULTA INFORMAČNÍCH TECHNOLOGIÍ

DEPARTMENT OF COMPUTER SYSTEMS
ÚSTAV POČÍTAČOVÝCH SYSTÉMŮ

CREATING A PYTHON-BASED AUTOMATED SYSTEM FOR RECOGNIZING EMOTIONS FROM FACIAL EXPRESSIONS

VYTVOŘENÍ AUTOMATIZOVANÉHO SYSTÉMU ZALOŽENÉHO NA PYTHONU PRO ROZPOZNÁVÁNÍ EMOCÍ Z VÝRAZŮ OBLIČEJE

BACHELOR'S THESIS
BAKALÁŘSKÁ PRÁCE

AUTHOR **SAMUEL ZIMA**
AUTOR PRÁCE

SUPERVISOR **YASIR HUSSAIN**
VEDOUCÍ PRÁCE

BRNO 2024

BRNO UNIVERSITY OF TECHNOLOGY
FACULTY OF INFORMATION TECHNOLOGY

Bachelor's Thesis Assignment

156771

Institut:	Department of Computer Systems (DCSY)
Student:	**Zima Samuel**
Programme:	Information Technology
Title:	**Creating a Python-based Automated System for Recognizing Emotions from Facial Expressions.**
Category:	Image Processing
Academic year:	2023/24

Assignment:

1. Study human emotions and their effects on the face.
2. Identify the challenges and limitations of existing methods in the area of automated emotion recognition through a literature review.
3. Design a machine learning (including deep learning) model for automated detection of human emotion from facial images.
4. Implement the proposed machine learning model for emotion detection using face images in Python.
5. Test and validate the emotion detection model on dataset images.
6. Discuss the obtained results and their contribution.

Literature:

- Based on the supervisor's recommendation.

Requirements for the semestral defence:

- Fulfillment of Items 1 to 3 of the assignment.

Detailed formal requirements can be found at https://www.fit.vut.cz/study/theses/

Supervisor:	**Hussain Yasir**
Head of Department:	Sekanina Lukáš, prof. Ing., Ph.D.
Beginning of work:	1.11.2023
Submission deadline:	9.5.2024
Approval date:	30.10.2023

Abstract

This thesis examines facial expression recognition (FER) using deep learning by focusing on its application in devices with limited memory and computational resources. It begins by researching emotions and facial expressions from psychological, biological, and sociological perspectives. The core of this thesis involves the design and implementation of an automated FER system using the FER-2013 dataset. This system uses a customized SqueezeNet architecture enhanced with a simple bypass, dropout layers and batch normalization layers. This system achieves an accuracy of 66.37% on the FER-2013 dataset. For comparative analysis, this model was compared with a customized VGG16 architecture which achieved an accuracy of 65.09%. This thesis provides valuable insights into the development of smaller, more efficient machine learning models for FER which are usable in a wide range of devices, including low-performance CPUs and embedded devices.

Abstrakt

Táto práca skúma rozpoznávanie výrazu tváre (angl. facial expression recognition – FER) pomocou hlbokého učenia so zameraním na použitie v zariadeniach s obmedzenou pamäťou a výpočtovými zdrojmi. Začína výskumom emócií a výrazov tváre z psychologického, biologického a sociologického hľadiska. Jadro výskumu tvorí návrh a implementácia automatizovaného systému pre FER s použitím súboru dát FER-2013. Tento systém využíva prispôsobenú architektúru SqueezeNet rozšírenú o jednoduchý obchvat, vrstvy náhodného odpadu neurónov a vrstvy dávkovej normalizácie. Tento systém dosahuje na súbore dát FER-2013 presnosť 66,37 %. Pre porovnávaciu analýzu sa tento model porovnal s upravenou architektúrou VGG16, ktorá dosiahla presnosť 65,09 %. Táto práca poskytuje cenné poznatky o vývoji menších, efektívnejších modelov strojového učenia pre FER, ktoré sú použiteľné pre široké spektrum zariadení vrátane nízkovýkonných procesorov a vstavaných zariadení.

Keywords

facial expression recognition, emotions, facial expressions, anatomy of the face, convolutional neural networks, machine learning, deep learning, SqueezeNet, VGG16, embedded devices, FER-2013

Klíčová slova

rozpoznávanie výrazov tváre, emócie, výrazy tváre, anatómia tváre, konvolučné neurónové siete, strojové učenie, hlboké učenie, SqueezeNet, VGG16, vstavané zariadenia, FER-2013

Reference

ZIMA, Samuel. *Creating a Python-based Automated System for Recognizing Emotions from Facial Expressions.* Brno, 2024. Bachelor's thesis. Brno University of Technology, Faculty of Information Technology. Supervisor Yasir Hussain

Rozšířený abstrakt

Táto práca je zameraná na rozpoznávanie emócii z výrazov tváre pomocou hlbokého učenia so zreteľom na použitie v zariadeniach s obmedzenou pamäťou a výpočtovými zdrojmi. Ako prvé sú v tejto práci preskúmané emócie a ich vplyv na výrazy tváre.

Biologický základ emócii možno nájsť v mozgu, konkrétne v limbickom systéme, ktorý je zodpovedný za emočné reakcie. Z psychologického hľadiska sú emócie definované ako vrodené a prirodzené alebo pomocou teórie konštruovaných emócii. Emócie sa prejavujú najmä vo výrazoch tváre človeka, ale taktiež v gestikulácii a v reči.

Výrazy tváre sú formou emočnej reakcie a spôsobom komunikácie medzi ľuďmi. Výrazy tváre sú tvorené buď vedome alebo nevedome. Tieto vedomé prvky znamenajú, že človek je schopný predstierať nejakú emóciu v podobe výrazu tváre. Napriek tomu je možné pomocou nevedomých automatických mechanizmov zistiť, že daná emócia v podobe výrazu tváre nie je pravdivá. Jedným z týchto automatických mechanizmov sú aj takzvané mikrovýrazy, ktoré sú veľmi krátke a nevedomé výrazy tváre, ktoré prezrádzajú pravdivý emočný stav človeka. Ďalej je v tejto práci preskúmaná aj univerzálnosť výrazov tváre medzi kultúrami, ktorá ukazuje na možné malé odhýlky v rozpoznávaní a kategorizácii výrazov tváre medzi kultúrami.

Ako ďalšie je v tejto práci preskúmaná anatómia tváre, konkrétne svaly tváre. Tieto svaly tváre sú delené na žuvacie svaly a mimické svaly. Obidve skupiny svalov sú zodpovedné aj za vytváranie výrazov tváre.

Následne je v tejto práci preskúmaný dopad niektorých chorôb na výrazy tváre. Chorobami, ktoré ovlývňujú možnosť tvorby alebo rozpoznávanie výrazov tváre sú Alzheimerova choroba, Parkinsonova choroba, Bellova obrna a ďalšie.

Ďalej je v tejto práci preskúmaná aj dôležitosť výrazov tváre z pohľadu komunikácie a sociálneho života ľudí.

Jadro práce tvorí systém pre rozpoznávanie emócii z výrazov tváre. Súbor dát, ktorý tento systém používa je FER-2013. Predspracovanie obrázkov zo súboru dát FER-2013 je nasledovné. Najprv sú obrázky prevedené zo stupňa šedej do formátu RGB. Tieto obrázky sú následne normalizované a triedy jednotlivých obrázkov sú prevedené do kódu 1 z N (angl. one-hot encoding). Následne sú vypočítané váhy jednotlivých tried, keďže súbor dát FER-2013 je nevyvážený. Ako posledná je použitá technika rozširovania údajov, ktorá vytvára náhodné transformácie vstupných obrázkov.

Základom tohto systému je konvolučná neurónová sieť SqueezeNet, ktorá je rozšírená o jednoduchý obchvat, tri ohňové moduly, vrstvy náhodného odpadu neurónov a vrstvy dávkovej normalizácie. Táto architektúra taktiež používa krížovú entropiu ako stratovú funkciu a optimalizátor Adam.

Tento systém dosahuje na súbore dát FER-2013 presnosť 66,37 %. Pre porovnávaciu analýzu sa tento model porovnal s upravenou architektúrou VGG16, ktorá dosiahla presnosť 65,09 %. Táto práca poskytuje cenné poznatky o vývoji menších, efektívnejších modelov strojového učenia pre rozpoznávanie emócii z výrazov tváre, ktoré sú použiteľné pre široké spektrum zariadení vrátane nízkovýkonných procesorov a vstavaných zariadení.

Creating a Python-based Automated System for Recognizing Emotions from Facial Expressions

Declaration

I hereby declare that this Bachelor's thesis was prepared as an original work by the author under the supervision of Yassir Hussain. I have listed all the literary sources, publications and other sources, which were used during the preparation of this thesis.

.....
Samuel Zima
May 1, 2024

Acknowledgements

I would like to express my gratitude to my supervisor Yasir Hussain for his valuable advice throughout the course of this research. His guidance helped me to overcome difficult challenges that I faced in this research and enabled me to focus on important parts of this thesis to successfully complete it.

Contents

List of Figures

Chapter 1

Introduction

Human beings are social creatures and as such we communicate and interact with each other daily. Communication allows us to pass information, share experiences, convey needs, express thoughts, and build relationships. Apart from exchanging information, expressing and perceiving emotions is an essential part of a successful communication. Emotion perception is helpful for understanding the intent and meaning of the message being communicated. By recognizing and acknowledging another person's emotions we can connect with them on an emotional level and respond appropriately.

Emotions are often conveyed through nonverbal cues such as facial expressions, body language, and tone of voice. Paying attention to these cues helps grasp the complete message being communicated, not just the words spoken. From these nonverbal cues facial expressions are the most prevalent form of expressing emotions. Facial expressions are a central element of human expressive behaviour and are particularly suitable for expressing emotions and communicating them to the outside world. The technique of recognizing a person's facial expression of an emotion is known as facial expression recognition.

Facial expression recognition can have benefits and applications across various fields. The most notable one being healthcare. Some diseases and conditions, such as Alzheimer's disease and depression, can manifest as changes in facial expressions and body language. Facial expression recognition can help to identify these changes and aid in diagnosis and monitoring of treatment effectiveness.

Facial expression recognition is a problematic topic even today because emotions remain one of the most fascinating and mysterious products of brain function. This is evident in the difficulty of defining emotions and the difficulty of linking these emotions to facial expressions.

The goal of this thesis is to study and find relations between emotions and facial expressions, identify the current methods, restrictions and challenges of facial expression recognition, and then create an automated system that will be able to recognize emotions from facial expressions.

Chapter 2

Human Emotions and Their Relations to Facial Expressions

2.1 Emotions

Emotions play a vital part in every person's daily life and they are an important aspect of human mental life. Emotions are helpful for communication and understanding between people. They are also helpful in building relationships with others. Furthermore, emotions are also very important to oneself as they can be the driving force for a person and their actions. On the other hand, emotions can also cause conflicts and a lack of understanding between people. Emotions can also be a destructive force for people who are suffering from mood or anxiety disorders. Despite all that the concept of emotion is still not clearly defined and there is no scientific consensus on the fundamental nature of emotions.

There are two distinct perspectives on the nature of emotions in the field of affective science. The first perspective being that emotions are biologically given action plans that help humans to navigate the complexities of the world. From this perspective emotions are decoupled reflexes, and are like reflexes and fixed action patterns [46]. Emotional reactions are quite stereotypical and a person cannot invent a new emotional expression or experience a new emotion that is not in the person's biological inventory to begin with. However, unlike with reflexes, people have some control over how emotions cause a certain behaviour typical for that emotion, and therefore the emotion and the behaviour can be separated or suppressed in time [14]. This view of emotions is also called natural kind. This natural kind view of emotions emphasises the crucial role that emotions play in influencing our decisions, remembering past experiences, and colouring our perception of the future. Emotions can cause certain behaviours in the person experiencing the emotion while the perception of emotions in others can cause adaptive behaviour on the part of the observer. In addition, the natural kind view of emotions also suggests that people are endowed with a set of basic emotions that are universally recognized and experienced, and that these emotions are triggered by distinct neural circuits [46]. Therefore, emotions like happiness, sadness, anger, fear, disgust and surprise are widely considered to be these universally recognized and experienced basic emotions [44]. Furthermore, in 2017 researchers identified 27 distinct categories of emotion based on videos intended to evoke a certain emotion [38].

The second perspective being the theory of constructed emotion. From this point of view emotions are constructed throughout the entire brain by multiple brain networks in collaboration. This construction includes interoception, which is the collection of senses

providing information to the organism about the internal state of the body, concepts and social reality. Interoceptive predictions produce basic and affective feelings between pleasure and displeasure (valence), and between high arousal and low arousal. Concepts include emotion concepts as cultural knowledge and social reality provides the language that makes the perception of emotion possible among people [22]. The dimensional approach of valence and arousal has led some scientist to think that what people perceive and experience as distinct emotions are actually conceptual constructions that emerge from categorization of a more basic psychological process called core affect. The core affect is experienced along with the dimensions of valence and arousal which are then categorised into an emotion category based on the cognitive appraisal of the current context [46]. Evidence for this constructed emotions perspective has come from an extensive meta-analysis which found out that that emotions are not localised to distinct brain regions, and that in fact multiple brain regions are active during emotional experiences across the range of distinct emotion categories [64].

These two perspectives have emerged from different research traditions and methodologies. Researchers on the side of emotions as natural kind typically utilise evidence from neuroscience studies on humans and animals, and they asses whether emotional states are associated with distinct neural or cognitive reactions such as facial expressions. On the other hand, researchers on the side of constructed emotions typically utilise evidence from the analysis of subjective experience of emotions in humans usually by means of self-report, and they also utilise evidence from neuroimaging studies [46].

I think that in order to better explain and understand emotions, emotional expressions and experiences both of these perspectives should be taken into account.

2.2 The Biology of Emotions

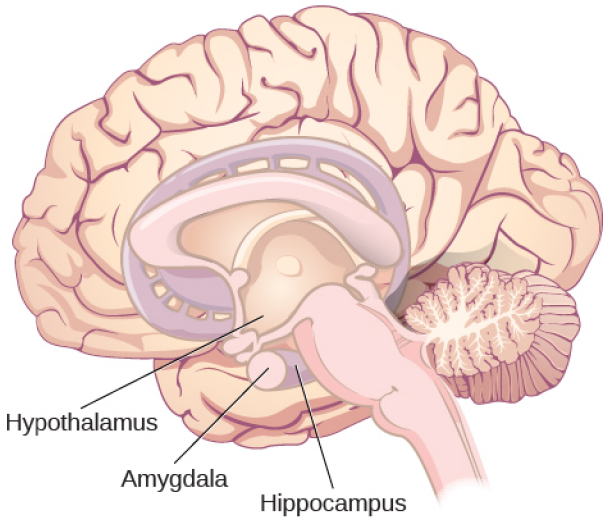

Figure 2.1: The limbic system [5].

The area of the brain which is involved in emotional and behavioural responses is called the limbic system. The limbic system is a group of interconnected structures located deep within the brain. The full list of structures that make up the limbic system is not consistently stated in literature, however the limbic system generally includes the hypothalamus,

hippocampus, amygdala, thalamus and limbic cortex. Experts still have a lot of questions about the brain's role in emotions and what areas of the brain are involved in different emotions, however they have pinpointed the origins of some common emotions like fear, anger and happiness [87].

There are several different systems in the brain that connect a stimulus with an emotional value or appraise the stimulus. Emotional systems do not exist in isolation and they communicate and influence each other. The first system involved with appraisal is the dopaminergic reward system. This system motivates people to repeat something that feels good. Another system involves the aforementioned amygdala. Amygdala are two small clusters of nerves that sit in each temporal lobe. The amygdala primarily mediates the responses of fear, anger and aggression. Furthermore, a structure of the brain called insula is also involved with an emotion of disgust [76] [87].

Once the structures for appraisal associate a stimulus with an emotional value a reaction begins. The amygdala is connected to the hypothalamus and can stimulate an increased heart rate and increased blood pressure which are important physical symptoms of the emotion of fear or anger [76].

However, sometimes as an emotion manifests it must be regulated or suppressed. A person may try to suppress the emotion by not permitting the face or the body to naturally show what they are feeling. The orbitofrontal cortex is activated in these cases of emotional regulation [76].

In summary, emotions are not generated by one part of the brain, and they rely on multiple interwoven neural networks involving the limbic system and also other parts of the brain which appraise the external or internal stimuli, generate an emotional response, and if needed regulate the emotional response. Any damage to these brain areas can result in the inability to regulate, recognize or generate an emotion.

2.3 Facial Expression

The term facial expression describes visible muscle movements on the face that can be perceived and interpreted by other people. Facial expression can be considered as a form of emotional response and as a form of social communication. Both of these forms then shape a facial expression, although sometimes one form is more prevalent than the other. For example in involuntary spontaneous facial expressions the emotional response is the dominant form, and on the other hand in arbitrary facial expressions the social communication is the dominant form. The muscular mobility of the face which is highly developed in humans is controlled by a complex neural control that encompasses both automatic and volitional components [13].

2.3.1 Voluntary and Involuntary Facial Expressions

The volitional component enables individuals to consciously control their facial muscles, allowing them to mimic or feign various emotional states. This ability plays a significant role in social interactions, where individuals often modify their facial expressions to influence or deceive others. However, this deception is not always directed outwardly. Sometimes, people use facial expressions to deceive themselves as a coping mechanism in stressful situations [61].

On the other hand the involuntary or automatic aspect of facial expressions is controlled by automatic neural mechanisms that can override conscious control. These spontaneous

expressions often reveal a person's true emotional state, regardless of their attempts to conceal it. This emotional override does not have to be accompanied by large movements of facial muscles, it can instead involve very subtle or discreet facial muscle movements [61]. These subtle and discreet facial muscle movements are called microexpressions. Microexpressions are very brief and involuntary facial expressions that reveal person's genuine emotional state, even if they try to hide it. These expressions are typically rapid and subtle, making them difficult to detect without proper training or keen observation [85].

In an experiment conducted by Paul Ekman and Wallace V. Friesen they highlighted the significance of microexpressions in uncovering concealed emotions. They observed and filmed a psychiatric patient who attempted to conceal their emotional distress and simulated optimism, control of affect, and feelings of well being during an interview. Despite the patient's efforts, Ekman, Friesen, and the interviewer identified several microexpressions and nonverbal cues that clearly indicated deception from the patient. At the end of the interview, the patient suffered an emotional breakdown, cried and admitted to not feeling well. This confirmed the presence of concealed distress from the patient [45].

The ability to detect and interpret concealed emotions using microexpressions has a potential in various fields. Healthcare professionals can use microexpressions to better understand their patients' mental states and enhance empathy, and communication. This is also valuable in scenarios where patients may be unable or unwilling to verbally express their emotions. Similarly, microexpressions can be used in law enforcement and security. Understanding and observing microexpressions can aid in assessing truthfulness and detecting deception, which can be crucial during interrogations and negotiations. Furthermore, microexpressions can be also used in any profession that requires face-to-face interpersonal skills [66].

2.3.2 Universality of Facial Expressions

Facial expressions have also raised the question whether these facial expressions of emotions are universal across different cultures.

Charles Darwin was the first scientist who systematically examined the universality of facial expressions and conducted various experiments. He conducted the first intercultural comparisons of facial expressions. His experiments also included observing expressions of young children, observing the expressions of people born blind, and analyzing the emotional expressions of people that were suffering from a mental illness. From his observations Darwin found strong similarities in facial expressions, and drew the conclusion about the universality of facial expressions [93]. Further notable experiments were conducted by Paul Ekman. He conducted an experiment with isolated tribes of New Guinea which shown that the tribespeople could recognize facial expressions of emotion posed by westerners and vice versa. Another study from 2004 examined the expressions of 84 judo athletes from 35 countries at the Athens Olympic Games. The winners of the displayed smiles and expressions of happiness whereas losers displayed expressions of sadness, anger or disgust. These spontaneous expressions by individuals from different countries and cultures in a naturalistic setting were also evidence of the universality of facial expressions of emotions [54].

On the other hand people around the world observe, and under certain circumstances use facial expressions differently. A study has shown that Westerners represent each of the six basic emotions with a distinct set of facial movements, whereas Asians do not. When observing a facial expression Asian observers also tend to fixate on the eyes region and represent emotional intensity with dynamic eye activity, but Western observers distribute

their fixations evenly across the face [56]. This shows culture-specific, and not universal observation and representation of basic emotions. Furthermore, cultural display rules also challenge the universality of facial expressions. Display rules are social norms that help individuals manage and modify emotional expressions depending on the social situations. In a study conducted by Wallace V. Friesen Americans and Japanese viewed stressful films alone, and then in the presence of an older, higher status experimenter. When viewing the films alone both parties expressed their positive and negative emotions fully, but when they viewed the films in the presence of another person, the Japanese were more likely to smile in the presence of another person than when they were alone. The Japanese display rule to not express negative feelings in the presence of higher status person were activated, and therefore differences in expressions occurred between the Japanese and the Americans [54].

Culture therefore plays an important role in managing emotions and their expressions. It is also responsible for the different observation strategies for facial expressions used by people. However, spontaneous emotional reactions are produced unconsciously and person's cultural or social norms may not modify or manage the expression of emotion in time.

2.4 The Muscles of the Face

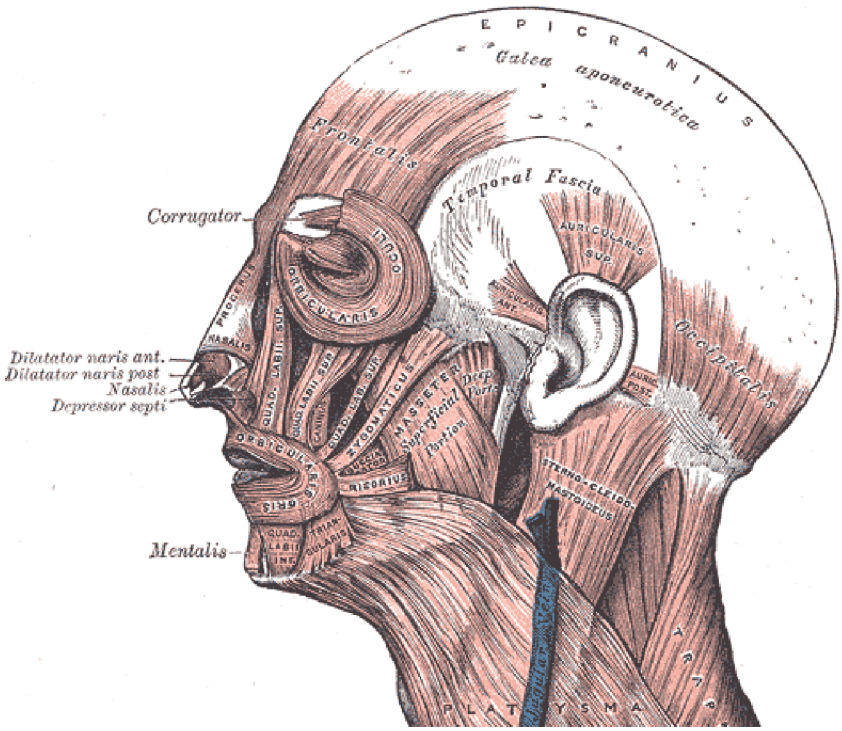

Figure 2.2: Overview of the facial muscles [92].

The ability for humans to communicate non verbally with so many different facial expressions is possible because of the complex structure of the muscles of facial expressions, also known as mimetic muscles.

The facial muscles are striated muscles and they are categorized into two groups of facial muscles. The first group being the aforementioned mimetic muscles and the second

group being the muscles of mastication. Despite this categorization of the facial muscles both groups of muscles usually act synchronously [92].

The mimetic muscles are innervated by the facial nerve, also known as the seventh cranial nerve. This nerve provides motor innervation of facial muscles that are responsible for facial expressions. This nerve is also divided into terminal branches where each of these branches innervate certain mimetic muscles [42].

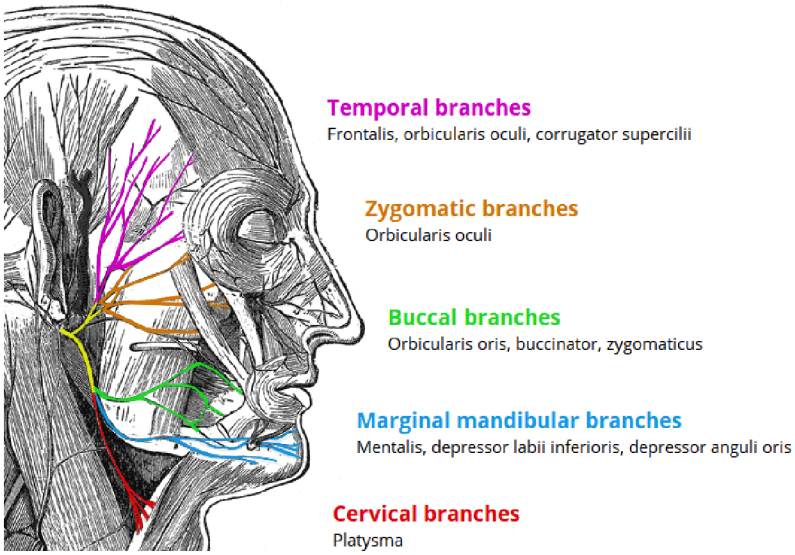

Figure 2.3: Innervation to the muscles of facial expression via the facial nerve (CN VII) [39].

The muscles of mastication are innervated by the trigeminal nerve, also known as the fifth cranial nerve. This trigeminal nerve is also divided into three main branches. One of these branches is the mandibular nerve which supplies motor innervation to all muscles involved in mastication [52].

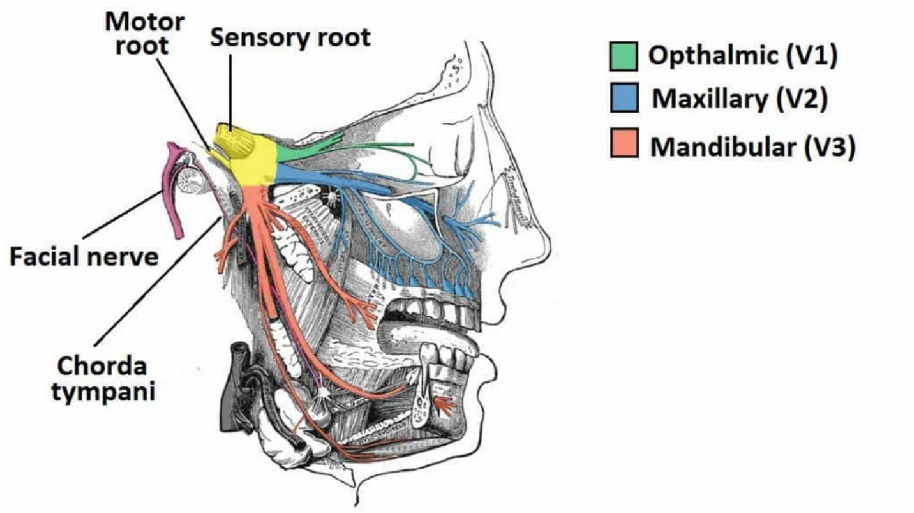

Figure 2.4: Overview of the deep distribution of the trigeminal nerve and its terminal branches [70].

The muscles of mastication are a group of muscles that are primarily responsible for the movement of the mandible (lower jaw) at the temporomanidbular joint. These muscles are divided into the primary muscles of mastication and the accessory muscles of mastication. The four primary muscles of mastication are the masseter, temporalis, lateral pterygoid and medial pterygoid [10].

The function of the masseter muscle is to elevate the mandible, approximate the teeth, and protrude or retract the mandible. The function of the temporalis muscle is to elevate the mandible and retract the mandible. The function of the lateral pterygoid is the depression of the mandible, and it also assists with side to side movement of the mandible. The function of the medial pterygoid muscle is to assist with elevation and protrusion of the mandible, and it also assists with side to side movement of the mandible [23].

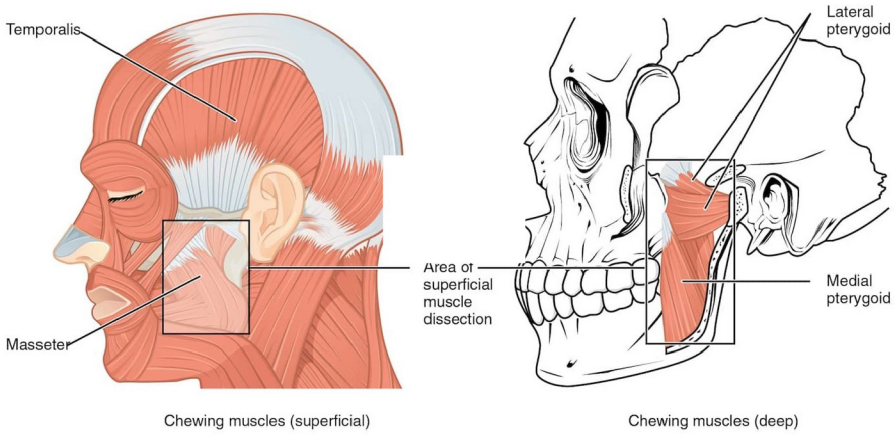

Figure 2.5: Overview of the muscles of mastication [69].

The mimetic muscles are located under the hypodermis layer of the skin on the face. These muscles originate from the bone of the skull and they are inserted onto the skin of the face. The mimetic muscles are the only muscles that are attached directly to the skin. The contraction of various groups of these muscles results in various facial expressions. The mimetic muscles are categorized into three groups [92] [60].

The first group of mimetic muscles are the orbital facial muscles. This group includes the occipitofrontalis, orbicularis oculi and corrugator supercilii. The occipitofrontalis muscle is a long and wide muscle which covers a large part of the skull. The functions of this muscle are raising the eyebrows and wrinkling the forehead. The orbicularis oculi muscle surrounds the eye socket and extends to the eyelid. The main function of this muscle is closing the eyelid. This muscle is also involved in the drainage of tears. The corrugator supercilii muscle is located above the eyelid and closer to the bridge of the nose. The function of this muscle is to lower the eyebrow and create vertical wrinkles on the bridge of the nose [60].

The second group of mimetic muscles are the nasal muscles. This group consists of procerus, nasalis and depressor septi nasi. The procerus muscle is located along the bridge of the nose. The function of this muscle is lowering the eyebrows and creating wrinkles on the bridge of the nose. The nasalis muscle is located on the side of the nose and its functions are compressing or widening the nasal opening and depressing the tip of the nose. The depressor septi nasi is located at the side of the nasal opening and its function is widening the nasal opening [60].

The third group of mimetic muscles are the oral muscles. This group includes the orbicularis oris, buccinator and other oral muscles. The orbicularis oris muscle surrounds the lips. The function of this muscle is closing the lips and puckering the lips. The buccinator muscle is located along the cheek area and its function is pulling the cheeks inward. The other oral muscles are responsible for actions like elevating the angle of the mouth, lowering the angle of the mouth, pulling the angles of the mouth, elevating the upper lip, lowering the lower lip, protruding the lower lip, wrinkling the chin and tensing the skin on the neck [90].

2.5 Impact of Diseases on Facial Expressions

a person's ability to produce or perceive and interpret facial expressions or emotional states of others can be altered by various diseases and disorders.

The first disease that has an impact on production or perception of facial expressions is Alzheimer's disease. Alzheimer's disease is a neurodegenerative disease primarily known for its impact on memory and cognitive functions. However, its effects extend beyond cognitive decline to include significant alterations in emotional processing and social interactions. One of the key challenges faced by individuals with Alzheimer's is a diminished ability to recognize and interpret facial expressions. As Alzheimer's disease progresses, there is a notable deterioration in the ability to perceive or understand emotional states from facial expressions. This decline is often attributed to the neurodegenerative processes affecting areas of the brain involved in emotion recognition, such as the amygdala, the prefrontal cortex and others. The impairment in recognizing facial expressions of emotions in others can lead to misunderstandings and difficult social relationships. Moreover, individuals with Alzheimer's may also exhibit changes in their ability to express emotions facially. The degeneration of neural pathways can lead to a reduced range and intensity of facial expressions, making it difficult for them to convey their own emotions effectively. This change can be especially challenging for caregivers and family members, as it hinders their ability to understand and empathize with the emotional state of their loved ones with Alzheimer's disease [67] [51].

The second example of a disorder that affects facial expressions is Bell's palsy. Bell's palsy is a neurological disorder characterized by sudden onset of paralysis or weakness on one side of the face, which can significantly impact production of facial expressions. This condition is caused by a dysfunction of the seventh cranial nerve, also known as the facial nerve (CN VII), which is responsible for innervating the mimetic muscles that control facial expressions. Notably, Bell's palsy is the most common cause of facial paralysis [6].

The third example of a disease that affects facial expressions is Parkinson's disease. Parkinson's disease is a neurodegenerative disease that is also associated with reduction in facial movements. In later stages of the disease this disease can severely restrict a person's ability to produce facial expressions. This condition of not being able to produce facial expressions is called hypomimia or face masking. Furthermore, Parkinson's disease also affects a person's ability to perceive facial expressions and interpret their meaning or the emotion behind the facial expression. This impairment can be attributed to the neurodegenerative processes affecting areas of the brain involved in emotion recognition and social cognition. As a result, social interactions can become more challenging, as the ability to perceive and respond appropriately to non-verbal communication is diminished [75] [61].

Another example of a disorder that affects person's ability to perceive and interpret facial expressions is major depressive disorder. A study has shown that people suffering

from major depressive disorder tend to evaluate ambiguous and neutral facial expression as intensely more sad or less happy. Furthermore, they have increased attention towards sad facial expressions and less attention for happy expressions. Lastly, they showed reduced general accuracy in recognising sad and happy facial expressions [25].

Apart from the four aforementioned diseases and disorders deficits in perception of facial expressions are also found in people with schizophrenia, borderline personality disorder, multiple sclerosis, and alcohol use disorder [61].

2.6 Importance of Facial Expressions

Human beings are social creatures and as such person's everyday life is filled with multitude of social interactions with other people. These social interactions are carried out by verbal and non-verbal communication. Facial expressions are as stated before an important aspect of non-verbal communication.

The first importance of facial expressions in social interactions is so called emotion mirroring or emotion contagion. A person tends to unconsciously mimic the behaviour of another person that he or she is communicating with or observing. This mirroring creates greater alignment of the communicants, and enhances communication and understanding. Furthermore, if a person observes a fearful face for example, the observer is more likely to be more vigilant because there might be something to be afraid of. The facial expression of fear also enhances vigilance, the eyes widen which increases the visual field and the widening of nostrils enhances the sense of smell. Therefore, by observing and mimicking other person's expression of fear, the observer becomes more vigilant and enhances certain senses or sensory signals [47].

Another importance of facial expressions in social interactions is the usage of these facial expressions as communicative signals. In the dynamic of a conversation, participants react to the spoken content not just with words, but also through a variety of facial expressions. These expressions serve as an immediate and often subconscious feedback mechanism. The person speaking can perceive these non-verbal cues and gain valuable insight into the listener's reactions and emotions. This awareness allows the speaker to adapt the pace and the content of their dialogue to better align with the listener's mood and level of understanding. For instance, a puzzled look might encourage further explanation, while a nod might encourage continuation. This emotional awareness enhances the depth and effectiveness of communication, and it inspires a more empathetic and responsive interaction. Emotional awareness can be very helpful both in personal and professional life. It enables more meaningful relationships and effective exchange of ideas. Furthermore, emotional awareness is not an innate skill, however it can be developed and refined over time like any other soft skill [61] [86].

Chapter 3

Facial Expression Recognition

In the ever evolving field of artificial intelligence and machine learning, facial expression recognition is an important area of research that combines the complexities of human emotion with machine learning. This field aims to enable machines to interpret and classify human facial expressions. This technology offers various benefits like aiding in psychological research, helping law-enforcement, enhancing user experience in electronic devices and improving interpersonal communication and understanding. Despite the rapid advancements, this field faces many challenges such as understanding and categorization of emotions, variability in expressions across different cultures and the need for robust datasets. This chapter focuses on dataset used in this thesis, emotion classification, emotion model and methodologies of facial expression recognition.

3.1 Dataset

The FER-2013 dataset which stands for Facial Expression Recognition 2013, is a significant resource in the field of computer vision and artificial intelligence. It was introduced at the International Conference on Machine Learning in 2013. The FER-2013 dataset consists of 35 887 grayscale images, each with a resolution of 48x48 pixels. The faces have been automatically registered, so that the face is more or less centred and occupies about the same amount of space in each image. These images are categorized into seven distinct facial expressions: anger, disgust, fear, happiness, sadness, surprise, and neutral. The dataset was compiled to help in the development and benchmarking of machine learning models capable of recognizing emotions from facial expressions. This dataset is notable for its emphasis on capturing a wide range of demographics and spontaneous expressions, rather than posed expressions, which makes it a valuable resource for training more realistic and effective facial expression recognition systems. The images in FER-2013 are sourced from the internet, which contributes to the diversity in age, ethnicity, and lighting conditions [82].

I have chosen the FER2013 dataset because it primarily captures spontaneous expressions, and because it has many images for each of the six basic emotions. However, the dataset is unbalanced in terms of the number of images for each emotion. Therefore, to address this, some data augmentation of certain expressions will be necessary. I also think that this dataset is suitable for implementation of a deep learning facial expression recognition system.

	AN	DI	FE	HA	NE	SA	SU
MultiPie	0	22696	0	47338	114305	0	19817
MMI	1959	1517	1313	2785	0	2169	1746
CK+	45	59	25	69	0	28	83
DISFA	436	5326	4073	28404	48582	1024	1365
FERA	1681	0	1467	1882	0	2115	0
SFEW	104	81	90	112	98	92	86
FER2013	4953	547	5121	8989	6198	6077	4002

* AN, DI, FE, HA, Ne, SA, SU stand for Anger, Disgust, Fear, Happiness, Neutral, Sadness, Surprised respectively.

Figure 3.1: Number of images per each expression in databases [68].

3.2 Emotion Classification

Emotion classification is an important field within psychological research which encompasses the systematic categorization of emotions. This field which is rooted in psychological studies, neurological and biological research identifies core emotions such as happiness, sadness, anger, fear, surprise, and disgust. Recent advancements in this field extend the categories of emotions, and acknowledge more complex emotions such as trust, anticipation, love and many other emotions. These classifications are not only human constructs, but they reflect the underlying physiological and neurological processes. Diverse methodologies, ranging from observational studies to neuroimaging contribute to our understanding of how emotions manifest and influence human behavior. This multilayered approach highlights the complex interactions between cognitive processes and emotional responses, indicating the complexity of human emotional experience.

Emotion classification in psychology is broadly segmented into two primary models: discrete categories and dimensional models. The discrete categories model argues that emotions are distinct and categorizable, typically identified as basic emotions like happiness, sadness, anger, fear, surprise, and disgust. This perspective, rooted in the works of psychologists like Paul Ekman, suggests that these emotions are universal with identifiable facial expressions and physiological responses. On the other hand there are dimensional models such as Russell's circumplex model of affect. The dimensional model perspective proposes that emotions can be characterized on a two dimensional axis by valence (pleasant-unpleasant) and arousal (activated-deactivated). This approach emphasizes the fluency and spectrum of emotions, rather than distinct categories. Both models offer unique insights into emotional complexity, with discrete categories underscoring the fundamental aspects of emotional experience, and dimensional models highlighting the nuanced interaction and degrees of emotional states [73] [50].

3.3 Emotion Model

In this thesis I will be focusing on facial expressions of happiness, sadness, anger, surprise, disgust and fear which make up the FER-2013 dataset along with a neutral expression. The facial expressions of these emotions are explained in the following table.

Number	Emotion class	Explanation of the facial expression
1	Happiness	upward curving of the corners of the mouth (smiling), crow's feet around the eyes, reduction in the size of the eye opening, relaxed eyebrows
2	Sadness	downturned corners of the mouth, slightly closed eyes, unfocused or downward gaze, inner eyebrows slightly upward and drawn together
3	Anger	tightened or closed jaw, glaring eyes with tightened eyelids, furrowed eyebrows
4	Surprise	opened mouth, eyes wide open, raised eyebrows
5	Disgust	upper lip raised, nose wrinkling, raised cheeks, narrowed eyes
6	Fear	eyebrows raised and drawn together, eyes wide open, increased jaw tension or slightly opened mouth, pale face

Table 3.1: Explanation of classes in the FER-2013 dataset.

3.4 Facial Expression Recognition Using Machine Learning

Facial expression recognition (FER) through machine learning represents a significant progress in the area of artificial intelligence and computer vision, offering various possibilities in fields such as psychological or medical research, security, and interactive media. Central part of this domain is the application of sophisticated machine learning models, which are capable of deciphering complex patterns in facial features and mapping these features to specific emotional states. By mapping facial muscle movements and its patterns, machine learning models are trained to classify expressions into categories like happiness, sadness, anger, surprise, disgust, and fear. The accuracy and efficiency of these models are continuously enhanced by advancements in dataset quality, data preprocessing and different machine learning architectures.

The general approach to facial expression recognition using machine learning typically consists of: face detection, feature extraction, model selection and training.

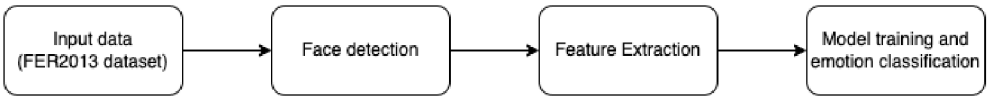

Figure 3.2: The general steps of facial expression recognition.

3.4.1 Face Detection

Face detection is the first step in facial expression recognition, focusing on accurately locating and identifying human faces within an image. This process typically involves algorithms that can discern facial features from various backgrounds and lighting conditions. Traditional methods like the Viola-Jones algorithm use Haar cascades to detect faces quickly. Modern approaches largely rely on deep learning, especially Convolutional Neural Networks (CNNs), which offer greater accuracy and robustness. These networks are trained on large datasets of images, enabling them to recognize a wide range of facial types and orientations. Once a face is detected, its position and size are used to isolate it from the rest of the image, allowing for more focused and accurate subsequent analysis of the facial expressions [37] [62]. This isolation is important for effective facial expression recognition, as it ensures that the algorithms analyzing and classifying facial expressions are focused only on the relevant facial features without interference from surrounding elements.

3.4.2 Feature Extraction

These techniques can be broadly categorized into geometric and appearance-based methods. Geometric methods focus on identifying specific landmarks on the face, such as the corners of the mouth or the edges of the eyebrows, and quantifying changes in these features. This approach effectively captures the structural alterations associated with different expressions. On the other hand, appearance-based methods analyze changes in the texture and appearance of the face, often utilizing techniques like Gabor filters, Local Binary Patterns (LBP), and Histogram of Oriented Gradients (HOG). These methods are capable of recognizing subtle variations in facial expressions by examining changes in skin texture and wrinkles. Moreover, with the advent of deep learning, convolutional neural networks (CNNs) have become increasingly popular in this domain. They automatically extract hierarchically structured features, learning both details and facial attributes related to specific facial expressions. This integration of geometric, appearance-based, and deep learning methods offers a complex and more robust approach to facial expression recognition which enhances the system's accuracy and generalization capabilities [81] [79].

3.4.3 Models

These models are designed to interpret and classify human emotions based on facial expressions, using a variety of algorithms ranging from traditional machine learning techniques to advanced deep learning networks. Traditional models like Support Vector Machines (SVMs) and Random Forests have been employed for their effectiveness in pattern recognition and classification. However, the advent of deep learning has brought forward more sophisticated and more complex architectures such as Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs). These architectures are particularly capable in handling the complexities of the different features in facial expressions and image classification. These models are trained on large datasets containing a wide array of human faces, capturing

a diverse range of emotions under various conditions. The challenge lies in accurately interpreting subtle changes in expressions and ensuring the models are unbiased and effective across different demographics. The applications of these models range from enhancing user experience in various electronic devices to aiding mental health professionals in diagnosing and treating emotional and psychiatric disorders. The development and refinement of these models continues to be a dynamic and evolving area of research, pushing the boundaries of machine learning and facial expression recognition [79] [81].

3.5 Machine Learning versus Deep Learning

In the realm of traditional machine learning, the approach to FER is characterized by a reliance on manual feature extraction and the use of established algorithms like Support Vector Machines (SVM) or K-Nearest Neighbors (KNN). This methodology requires a deep understanding of the facial features that are most indicative of various emotions and it typically involves less computational complexity. Machine learning methods are advantageous in scenarios with limited training data, as they require fewer data points to train effectively. However, they may lack the complex understanding necessary to interpret the more complex and subtle variations in facial expressions [37] [81].

On the other hand, deep learning, especially through the use of Convolutional Neural Networks (CNNs), offers an innovative approach to FER. By automatically extracting features directly from raw images, deep learning models are capable of processing and learning from high-dimensional data. This ability enables them to capture a wide range of facial expressions of emotions, from the most obvious to the most subtle ones. Deep learning models, due to their layered architecture, can learn hierarchical feature representations which makes them exceptionally good at handling the intricacies involved in interpreting and classifying facial expressions. They require larger datasets for training, but they are more effective in diverse and real-world scenarios, where facial expressions vary greatly across individuals and contexts [63] [68].

Deep learning models are generally less interpretable than the traditional machine learning models, which can be a drawback in applications where understanding the decision-making process of the model is crucial. However, the trade-off comes with significantly enhanced accuracy when deep learning models are used. This makes deep learning-based FER systems more effective and reliable in practical applications [63] [68].

In summary, while machine learning offers a more straightforward, less data-intensive approach to FER, it may fall short in dealing with the complexity and variability in facial expressions. In contrast, deep learning brings a more sophisticated, data-driven methodology, capable of comprehending the full spectrum of human emotions with greater depth and accuracy.

3.6 Facial Expression Recognition Using Deep Learning

The general approach to facial expression recognition using deep learning typically consists of: preprocessing, model selection and training.

3.6.1 Data Preprocessing

Data preprocessing is an important step in developing a facial expression recognition system using a neural network architecture. This phase involves several key tasks to prepare the

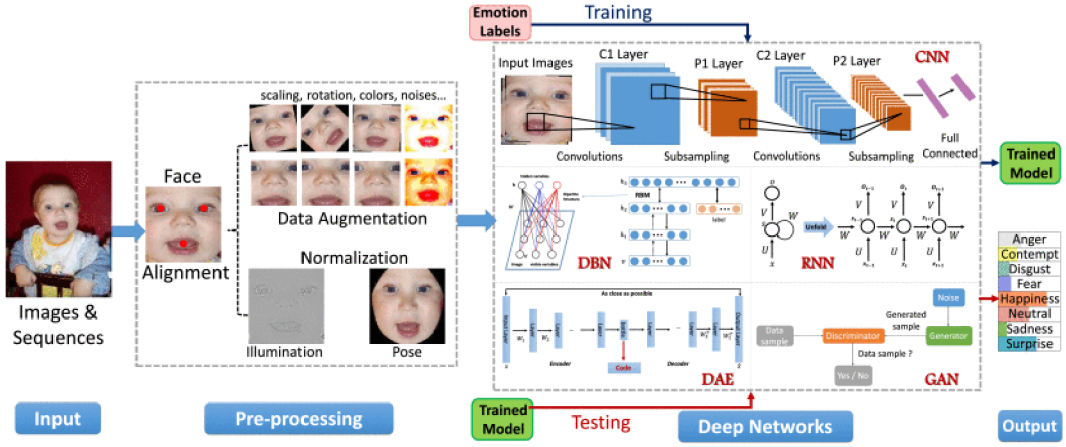

Figure 3.3: The general pipeline of deep facial expression recognition systems [63].

facial image dataset for effective training. Firstly, images are resized to a consistent size to ensure uniform input dimensions for the neural network. Pixel values are normalized, typically scaling them to a range of 0 to 1 or -1 to 1, to aid in the convergence of the network during training. Data augmentation techniques such as rotating, scaling, and flipping images are applied to increase the dataset's size, variability, and to balance the different categories of images. This process also involves splitting the dataset into distinct sets for training, validation and testing. The training and validation set is used to train the model and the test set to evaluate the model's performance. Proper preprocessing is important for training more accurate models, especially in a complex task like facial expression recognition, where variations in lighting, angles, and facial features can significantly impact model performance [12] [63].

3.6.2 Model Architecture

Designing an architecture based on Convolutional Neural Networks (CNN) or utilizing deep learning for facial expression recognition is a process that involves constructing a network capable of accurately identifying and classifying various facial expressions of emotions. The architecture typically consists of several layers, including convolutional layers, pooling layers, and fully connected layers. Convolutional layers usually consist of numerous filters which are essential for extracting different features from the facial images, such as edges and textures. These features are very important in recognizing facial expressions. Pooling layers reduce the spatial dimensions of the extracted features, helping in reducing the computational load. As the network deepens, it becomes capable of recognizing more complex patterns. After several convolutional and pooling layers, the network usually transitions to fully connected layers, which interpret these high-level features and perform the classification task. The final layer often uses a softmax function for multi-class classification of facial expressions. The design of the CNN architecture is therefore a very important step that significantly influences the effectiveness, accuracy and computational load of the facial expression recognition system [12] [63].

Chapter 4

Designing Facial Expression Recognition System

In this thesis I decided to design and implement the facial expression recognition system using deep learning. More precisely I want to implement this system using a smaller Convolutional Neural Network architecture with fewer parameters, so that the entire model can be trained on low-performing CPUs with limited computational resources and that it can be deployed and applicable in a variety of devices including embedded devices.

In this thesis I will be using the FER-2013 dataset for my facial expression recognition system. The dataset is described in section 3.1. The individual facial expressions of emotions which are covered by the FER-2013 dataset are described in table 3.1 along with a brief overview of the distinct features of individual facial expressions.

4.1 Dividing the Dataset

First of all, before I design the neural network model itself, the FER-2013 dataset should be divided into three separate datasets. The first dataset should be the **training dataset** which should be the largest of all the three datasets. It will be used for training the implemented neural network model, so that the model can understand the patterns and relationships within the data, and therefore learn to make predictions or decisions without being explicitly programmed to perform a specific task. The training process involves repeatedly adjusting the model's parameters to enhance its decision-making and predictive capabilities. Furthermore, a larger training dataset allows the model to encounter a wide range of data that facilitate better generelization capabilities and reduce the risk of overfitting [3] [43].

The second dataset should be the **validation dataset**. This dataset should be used alongside the training dataset in the training phase to assess how the neural network model performs on the data that it wasn't trained on. The validation dataset is also used to determine how the model is learning and adapting to the data overall. This dataset also helps to identify the signs of overfitting, where the model performs well on the training data, but poorly on the validation data. Moreover, the validation dataset enables an iterative process of improvement. Based on its performance on training and this dataset, adjustments can be made to the model's architecture, input data or training process to enhance its accuracy and effectiveness.[31] [3].

The third and last dataset should be the **testing dataset**. This dataset will be used after the neural network model is already trained and validated. This dataset offers an unbiased evaluation of the model's effectiveness. Since the model has never encountered this data during its learning phases, its performance on the testing dataset is a strong indicator of its real-world applicability. Lastly, the analysis of the model's performance on the testing dataset helps in identifying specific challenges and limitations of the current model and provides useful information for future improvements [31] [3].

4.2 Data Preprocessing

After the FER-2013 dataset is split into the training, validation and testing dataset, I need to preprocess the images and categories from the individual datasets before they can be used by the neural network model.

4.2.1 Normalization

The first step in data preprocessing is normalizing the data, so that each image and its pixels are translated to the same range of values, typically to a range between 0 and 1 or between -1 and 1. Normalized data enhance the model's performance and accuracy. Normalization scales down the wide range of input values and helps the neural network to treat all features equally, which prevents any feature from disproportionately influencing the model's learning process. Furthermore, normalization speeds up the convergence of the training process and allows the model to learn and generalize more effectively [59].

4.2.2 One-hot Encoding

Next step in data preprocessing is to convert the categories of images from the created datasets to a form which can be easily interpreted by the neural network. One-hot encoding is a method used in machine learning to convert categorical data. It involves converting each categorical value into a new binary row and assigning a 1 or 0 in the row's columns. Each category is represented by a row or a vector containing 1 in the position of the correct category and 0 in all other positions. This approach is particularly useful for handling data without any ordinal relationship or dependencies between categories, as it avoids artificial hierarchies among categories. In machine learning models, this technique helps in processing non-numeric categories and maintaining the categorical distinctions clear and explicit for the algorithm [2].

4.2.3 Data Augmentation

Another step that I likely need to do before the data can be inputted and processed by the neural network is data augmentation because the FER-2013 dataset is imbalanced as it can be seen in figure 3.1.

Data augmentation is a technique used in machine learning that artificially increases the training dataset by creating modified copies of the existing data in the training dataset. This is achieved by applying various transformations to the existing data to generate new and modified data. In image processing, as is the case for me in this thesis, these transformations might include rotating, flipping, scaling, or changing the color of images. Data augmentation is beneficial for multiple reasons. The first reason is to simply add more data from which the model can learn. Another reason is to reduce the risk of overfitting and create

more variability in the data. Furthermore, data augmentation increases the generalization capabilities of the model and can help to resolve class imbalances between categories in the dataset. Last but not least, data augmentation reduces the costs necessary for collecting and labeling new data [19] [41].

On the other hand, if data augmentation is not done correctly augmented data can mislead the model and lower the model's accuracy and generalization capabilities. Therefore, it is important to choose plausible or realistic data augmentation because inappropriately augmenting data might lead to unrealistic features that the model might learn as valid patterns. Moreover, data augmentation and training on larger datasets can lead to increased computational costs and training time [9].

4.3 Architecture of the Model

After the preprocessing I need to choose and implement the model's architecture. As I already mentioned at the beginning of this chapter I decided to implement the facial expression recognition system using a smaller and more lightweight Convolutional Neural Network. I found out that implementing and training any kind of machine learning model is an iterative process. Therefore, at first I should try and train different kinds of Convolutional Neural Networks with different number of convolutional, pooling and fully connected layers. I should also try out different activation functions for the model's layers. Furthermore, I should experiment with the number of neurons in each layer, and then observe and asses the performance of the model on the training and validation datasets.

4.4 Training the Model

The next step after I have the convolutional neural network model prepared is to start the training process. However, there are still some parameters that have to be chosen and experimented with for the training of the model.

4.4.1 Activation Function

An activation function is a mathematical function applied to the output of a neuron in a neural network. This function decides whether the neuron should be activated or not based on whether each neuron's input is important or not in the process of prediction. The primary role of the activation function is to transform a set of input values received by a neuron or a layer of neurons into an output value. This output is then send to subsequent layers or neurons in the network. Furthermore, this function is able to add non-linearity into the network. This feature is essential because many real-world data patterns are non-linear and linear models would struggle to capture such complex patterns [21] [58].

There are various activation functions and I think should experiment with multiple activation function and assess their performance with my model.

4.4.2 Loss Function

Loss functions in machine learning are fundamental concepts that are used to measure the error between predicted values and actual values. They play an important role in the training algorithm, as they provide a quantifiable measure of how well the model is performing. In addition, loss function improves the model by directing the model to adjust

it's weights during training. By iteratively minimizing the loss, the model can enhance its predictions and accuracy, which is the primary goal of machine learning model training. The objective of training a machine learning model is to find the set of parameters that minimize the loss function [89].

There are number of different loss functions that are broadly divided into two main groups. These groups are the loss functions for regression tasks and loss functions for classification tasks. Regression loss functions are used in tasks where the model is expected to predict continuous values. On the other hand, classification loss functions are used in tasks where the model is expected to predict discrete labels which correspond to a specific class in the dataset [16].

My task is to create a system which classifies images into discrete categories, therefore I will be using a classification loss function in the training phase of my model.

4.4.3 Optimizer

Optimizers in machine learning are algorithms or methods used to change the attributes of the machine learning model such as weights and learning rate in order to reduce and minimize the loss function and enhance the model's performance. These specialized algorithms facilitate the learning process of neural networks by iteratively adjusting the model's weights based on the feedback received from the data. There are multiple optimizers and each one differs in how they change the learning rate, and the approach in processing gradients during the training phase which allows for more efficient movement towards the optimal solution. This variation in approaches allows each optimizer to be more suitable for specific types of datasets and architectures [48].

Choosing the right optimizer for the task is also important, therefore I should also try out different kinds of optimizers and assess which one performs the best with my dataset and my model.

4.4.4 Batch Size and Epochs

Batch size is a hyperparameter which refers to a number of training instances or samples used in one iteration of the training phase. When training a model the training dataset is usually divided into smaller batches, and after the model processes the entire batch its weights are updated and another batch is given to the model. a smaller batch size often leads to more frequent updates and it can reduce the chance of overfitting and enhance the generalization capabilities of the model. However, it can also make the learning process longer due to more updates. On the other hand, a larger batch size enables faster computation by efficiently using computational resources as more data are processed at once. However, this can lead to less detailed updates which might impact the model's ability to reach the best generalization and performance [33] [40].

The number of epochs is a hyperparameter that defines the number of times that the learning algorithm will go through the entire training dataset during the training phase. During each epoch, the model has the chance to learn from every instance or sample in the dataset and make adjustments to its parameters. One epoch usually consists of one or more batches. More epochs mean the model has more opportunities to learn and adjust its weights which potentially leads to better performance. However, too many epochs can lead to overfitting, where the model performs very good on the training dataset and performs poorly on unseen data [33] [7].

In summary, it is important to balance the batch size with the number of epochs to achieve the best performance and accuracy from the model. Therefore, I should start with smaller number of epochs and smaller to medium batch size, e.g. 128, 256 or 512. After the training phase I should assess the results and then increase or decrease the number of epochs and batch size based on the results.

4.5 Evaluating the Model

The final step after a successful training phase and its assessment is the evaluation of the model on unseen data which provides an unbiased interpretation of the models performance and real-world applicability. For the evaluation of the model I will use the following metrics.

4.5.1 Accuracy

Accuracy is a metric that measures how often a machine learning model correctly predicts the outcome. In other words, it is the ratio of the number of correct predictions to the total number of predictions made. Accuracy is a statistical metric that shows how well a classification model can correctly identify the class labels of a given dataset. Mathematically accuracy can be defined as [1]:

$$\text{Accuracy} = \frac{\text{Correct predictions}}{\text{All predictions}}$$

4.5.2 Precision

Precision is a metric that measures how often does the machine learning model correctly predict the positive class. Specifically, it calculates the proportion of true positives, i.e. correctly predicted positive instances, out of all the instances that the model predicted as positive, which includes both true positives and false positives, i.e. instances which were wrongly predicted as positive. Mathematically precision can be defined as [77] [1]:

$$\text{Precision} = \frac{\text{True positives}}{\text{True positives} + \text{False positives}}$$

A high precision score means the classifier is good at avoiding false positive predictions, however it may come at the expense of higher false negative rates [77].

4.5.3 Recall

Recall, also known as sensitivity is a metric that quantifies the ability of a model to correctly identify all relevant instances within a dataset. In other words, it measures how often a machine learning model correctly identifies true positives from all the actual positive samples in the dataset. Mathematically recall can be defined as [77] [53]:

$$\text{Recall} = \frac{\text{True positives}}{\text{True positives} + \text{False Negatives}}$$

This metric works well for problems that have imbalanced classes since it is focused on the model's ability to find objects of the target class. This metric is important in scenarios where missing a positive instance carries significant consequences and you want to find all objects of the target class [1].

4.5.4 F1-score

The F1-score is an important metric in machine learning, particularly in classification tasks as it provides a balanced measure of a model's precision and recall. F1-score is the harmonic mean of these two metrics, which means it gives an equal weight to both precision and recall. F1-score combines both precision and recall into a single metric, providing a more comprehensive evaluation of the model's performance. Mathematically F1-score can be defined as [77] :

$$\mathrm{F1} = 2 \cdot \frac{\mathrm{Precision} \cdot \mathrm{Recall}}{\mathrm{Precision} + \mathrm{Recall}}$$

F1-score metric is especially useful in situations where an precision and recall is equally important. This metric is particularly also valuable in datasets with imbalanced class distributions. The F1-score serves as a single measure to gauge the effectiveness of the model in terms of reliability in its predictions [53].

4.5.5 Confusion Matrix

Confusion matrix is a performance evaluation tool in machine learning which represents the accuracy of a classification model. Confusion matrix is an N x N matrix used for evaluating the performance of a classification model, where N is the total number of target classes. The diagonal cells from the top-left to the bottom-right represent correct predictions by the model where the predicted class matches the actual correct class. The non-diagonal cells indicate errors made by the model. This matrix layout allows for a comprehensive and interpretable visual representation of how well the model is performing across all classes [24].

Chapter 5

Implementation of the Facial Expression Recognition System

For the implementation of the facial expression recognition system I have chosen to use the Tensorflow and Keras libraries for Python. Tensorflow is a library that offers extensive options for creating machine learning models, training them iteratively and exporting them. TensorFlow provides robust capabilities to deploy machine learning models on a variety of environments such as servers, edge devices, smartphones, microcontrollers, CPUs and GPUs [11].

Keras is an API that works along with Tensorflow. The goal of Keras is to reduce the congnitive load and offer consistent and simple APIs which minimize the number of user actions required for common use cases. Furthermore, Keras focuses on code conciseness, maintainability and deployability. The codebases creating with Keras are usually smaller, more readable and are easier to deploy on various environments. Keras also offers a straightforward way to create custom machine learning architectures or to simply choose from a variety of well-known and established machine learning architectures [35].

5.1 Data Preprocessing

First of all, as described in section 4.1 I have to import the FER-2013 dataset and then divide the dataset into training, validation and testing datasets. For this exact purpose I am using the Deeplake library for Python.

Deeplake is a specialized Python library that is designed for managing and utilizing datasets in machine learning applications. It primarily focuses on providing a robust and efficient means of accessing, storing and preprocessing large datasets that are commonly used in machine learning projects. The core functionality of Deeplake revolves around its ability to interact with a variety of database systems which makes it highly adaptable for different data storage environments. It simplifies fetching, transformation, and preparation of datasets for training and evaluation of machine learning models. This makes it a valuable tool for those working with extensive datasets, as it makes the data handling process easier and allowing for more focus on model development and performance optimalization [49]. Furthermore, Deeplake offers an easy way to access and download well-known datasets for machine learning models such as the FER-2013 dataset that I am using in this thesis. The FER-2013 dataset is also already divided into the training, validation and testing dataset in the Deeplake databse [34].

After loading the FER-2013 training and validation datasets to local variables, the individual tensors for images and labels must be separated and extracted for further preprocessing.

The separated and extracted images from the training and validation datasets are subsequently converted to RGB representation. This is necessary because the original images are grayscale, therefore they only have one colour channel and I have found out that certain layers from the Keras library need three colour channels to work correctly. This conversion from grayscale to RGB representation is done with the `convert_to_rgb` function. This function transforms the grayscale images to three colour channel images by duplicating the single channel across the three channels which makes the images compatible with functions that expect RGB input.

The next step after the images have been converted to RGB representation is to normalize the images from the training and validation datasets for the model. This is done by dividing the training and validation dataset images by 255 which normalizes the values of pixels of each image between the values of 0 and 1, as originally the values of not normalized images were between 0 and 255.

Another step is to perform one-hot encoding for the labels for the images in training and validation datasets. This is done by using the `to_categorical` function from Keras library which performs the one-hot encoding on the original numerical label of the target class. For example, in the FER-2013 dataset the facial expression of anger is encoded as a numerical value of 0, and it has seven categories for facial expressions, so the chain of encoding from the actual class name to one-hot encoding looks like this:

$$\text{Anger} \longrightarrow 0 \longrightarrow [1, 0, 0, 0, 0, 0, 0]$$

5.1.1 Data Augmentation

After the images from the training and validation datasets have been normalized and their labels converted to one-hot encoding I implemented the data augmentation. Data augmentation is implemented using the `ImageDataGenerator` class from the Keras library. The `ImageDataGenerator` class generates batches of tensor image data with real-time data augmentation which effectively increases the diversity of the training dataset. This data augmentation class includes transformations like rotations, shifts, shear, zoom, flips, and other transformations which help in making the model more robust to different variations in new data. The process involves specifying the desired transformations and then integrating the augmented data into the training process which often leads to better generalization capabilities and improved model performance. Furthermore, the `ImageDataGenerator` class also reduces overfitting because it exposes the model to a wider range of variations of the input data.

Personally, in data augmentation I am using rotations up to 20 degrees, very mild width and height shifts, shear and zoom. I am also using the horizontal flip transformation, so that the faces on the images can be flipped and symmetrical images of faces and facial expressions can be created.

5.1.2 Class Weights

During the implementation I have found out that data augmentation alone isn't enough for the class imbalance in FER-2013 dataset. Therefore, I decided to also use class weights

along with data augmentation to help combat the bias for the majority classes in the training dataset during the training phase.

Class weights in machine learning is a technique used to combat class imbalances in datasets where certain classes are underrepresented compared to other classes. In a typical dataset, each class might not be equally represented which can lead to biases in the model. This causes that the model performs poorly on the underrepresented classes. Class weights are a possible solution to this problem. By assigning different weights to individual classes, the model can be encouraged to pay more attention to the underrepresented classes. This is particularly useful in classification problems. The weights can be set manually or automatically based on the frequency of the classes. By adjusting the learning process to focus more on the minority classes, class weights can help in achieving a more balanced and fair performance across all classes [91].

For this purpose I am using the `compute_class_weight` function from the scikit-learn library. Scikit-learn is an another open-source machine learning library for Python which is well-known for its simplicity, efficiency, and accessibility. One of its key strengths is its extensive documentation and examples. Scikit-learn is widely used and popular due to its versatility and ease of use, making it an important part of the Python machine learning capabilities [72].

The mathematical definition for computing class weights and the formula that the `compute_class_weights` uses is as follows [91]:

$$w_j = \frac{n}{k * n_j}$$

Here the w_j is the calculated weight for the class j. The n is the total number of samples, k is the number of classes and n_j is the number of samples in the j class.

The computed class weights are then added as a parameter to the training function of the model.

5.2 Model Architecture – SqueezeNet

The architecture that I decided to implement based on my specifications is the SqueezeNet architecture. More precisely, I implemented a customized version of the SqueezeNet architecture for the facial expression recognition.

SqueezeNet is a convolutional neural network (CNN) designed to provide competitive accuracy while using significantly fewer parameters compared to other architectures. This makes it very efficient for deployment in environments where computational resources are limited, such as on smartphones or embedded systems. The key part and innovation in the SqueezeNet architecture is the use of so called fire modules. These modules consist of a squeeze layer which uses 1x1 convolutional filters to reduce the depth. The squeeze layer if followed by an expand layer which combines the 1x1 with 3x3 convolutional filters. This design drastically reduces the number of parameters in the network. Despite its compact size, SqueezeNet achieves accuracy levels comparable to larger networks like AlexNet which makes it a significant development in optimizing deep learning models for efficiency without sacrificing too much performance [55].

SqueezeNet architecture (5.3) starts with a standalone convolutional layer. This layer is then followed by eight fire modules and the architecture ends with a final convolutional layer. The network is quite deep due to the number of these layers, but it retains its efficiency regarding the number of parameters [18].

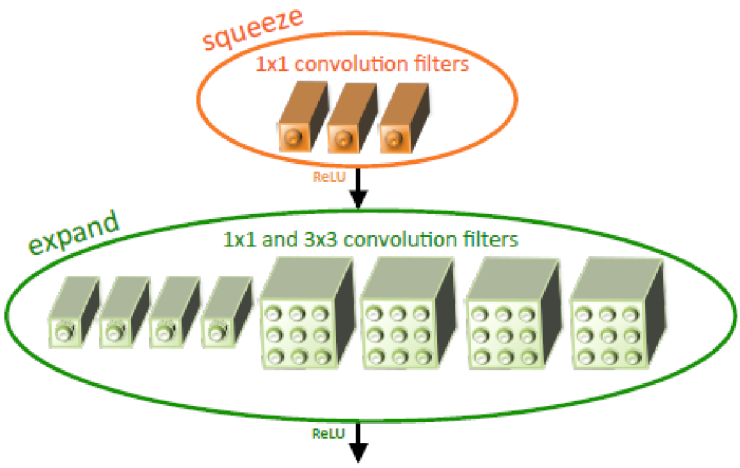

Figure 5.1: Microarchitectural view: Organization of convolution filters in the Fire module. [55].

These convolutional layers in CNNs perform convolution on the input image. They apply a filter on the input image and extract important features such as edges or specific shapes and patterns. The filter moves across the input image and performs the convolution on the parts of the image that the filter is currently covering. The output of each convolutional layer is the so called feature map. This feature map is a transformed version of the input image where the most prominent features detected by the filter are highlighted. Multiple filters can be used in a single convolutional layer and each of these filters is designed to detect different features from the image. These multiple convolutional layers allow the CNN to capture complex hierarchies and patterns in the image [30].

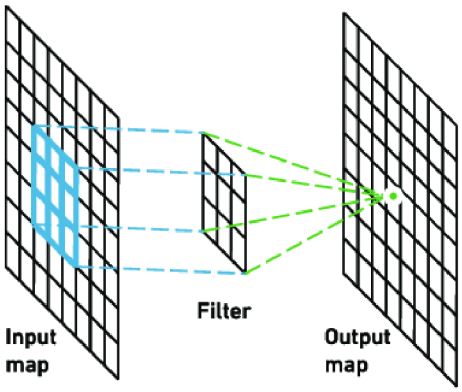

Figure 5.2: Outline of the convolutional layer [94].

After each convolutional layer a ReLU, i.e. Rectified Linear Unit, activation function is used. This function is a calculation that returns the input value if the input value is positive and if the value is negative, then it outputs the value of 0. This simplicity of the ReLU function accelerates the training process compared to other activation functions like sigmoid or tanh. Furthermore, it provides a solution to the problem of vanishing gradients [29].

The vanishing gradient problem in CNNs arises during the training phase, specifically from the way gradients are computed and used to update the weights of the CNN. Learning

in a CNN occurs through backpropagation where errors are propagated backwards from the output to adjust the weights of the CNN. However, if the derivative of the activation function used in the convolutional layers is very close to 0, it leads to small gradients. This causes the weights in the convolutional layers to change very little, making it difficult to effectively train certain convolutional layers in the CNN. If ReLU is used as the activation function in a CNN, the value of the derivative will have values of 0 or 1 which prevents the gradient from vanishing. Therefore, the use of ReLU function prevents the gradients from vanishing [57].

The ReLU function has become widely used in many CNNs because of its efficiency. Mathematically the ReLU function can be defined as follows [8]:

$$f(x) \ = \ max(0, \ x)$$

Next layers used in the SqueezeNet architecture are the Pooling layers that are added after the third, seventh and ninth fire module.

A pooling layer is a layer that is added after the convolutional layer. More specifically, it is added after the activation function has been applied to the outputted feature maps. Pooling layers provide various benefits for the CNN. They play an important role in reducing spatial dimensions which enable the model to learn different features from the dataset. The limitation of the outputted feature maps from convolutional layers is that they record precise positions of features in the input image. This means that when the position of the feature changes, the resulting feature map will be different [28]. Pooling layers address this problem by making the CNN invariant to changes in the input images from transformations such as rotations or shifts. Furthermore, pooling operations downsample the feature maps which results in a decrease of parameters and computational load for the subsequent layers. There are two types of pooling layers and those are the max pooling layer or the average pooling layer [17]. The SqueezeNet architecture uses max pooling layer after the third and seventh fire module and global average pooling layer after the final convolutional layer [55].

The max pooling layer works by sliding a specified window across the inputted feature map and taking the maximum value within that window [17].

The global average pooling layer calculates the average of all the values in each feature map. Essentially, it takes the entire feature map and compresses it into a single average value [17].

After the final convolutional layer which has a number of filters equal to the number of classes, and the ReLU activation function and global average pooling, the output is passed through the softmax activation function which calculates the probabilities for all classes. The mathematical definition of the softmax function is as follows [83]:

$$softmax(x_i) = \frac{e^{x_i}}{\sum_{j=1}^{K} e^{x_j}}$$

Here the x_i is the output of the global average pooling layer for the i-th class. e^{x_i} is the exponential of x_i. K represents the number of classes. $\sum_{j=1}^{K} e^{x_j}$ is the sum of the exponential scores for all classes. This sum ensures that the softmax function outputs sum up to 1. Output of this function represents the probability that the input belongs to class i.

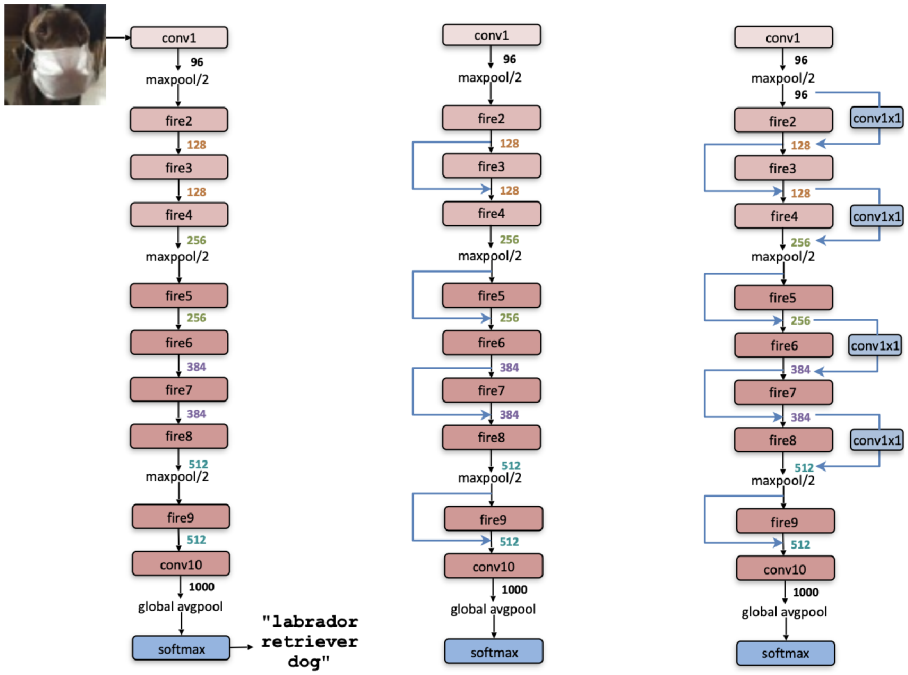

Figure 5.3: Macroarchitectural view of our SqueezeNet architecture. Left: SqueezeNet; Middle: SqueezeNet with simple bypass; Right: SqueezeNet with complex bypass [55].

5.3 Customized SqueezeNet

For my facial expression recognition system I used the SqueezeNet with simple bypass as a base model for my implementation.

The architecture with simple bypass allows the input of the fire module to be added to its output. This helps to reduce the problem with vanishing gradients which is described in section 5.2. Furthermore, the addition of the bypass usually helps to stabilize the learning process of the model and can lead to faster convergence and better performance. It can also improve the accuracy of the model, as the bypass enables features that could potentially be lost in the convolutional layers to be carried from one fire module to another in the model [55]. In my model the bypass connection is used in all fire modules, if the channels of the output and input of the fire module match.

The first addition that I have made to the original SqueezeNet architecture with bypass is that I added additional three fire modules, so that the model has slightly more parameters and more convolutional layers that can extract features from the input images.

Another addition that I have made to the fire modules of the SqueezeNet is adding a Dropout layer to the convolutional layers of the fire module after I experienced a problem with overfitting during the training phase. Dropout is a regularization technique that is used to reduce the overfitting of the model and enhance the model's generalization capabilities. This method works by randomly removing or dropping inputs to another layer. Specifically, it sets a number of input units to zero at each update of the model during the training phase. In my case, this dropout layer sets random pixels in the outputted feature maps from the convolutional layer to zero. This random dropping of inputs is controlled by the dropout rate which represents the probability with which the inputs or pixels are dropped [27] [20].

The dropout rate which I use in my model is 0.2. That means that each pixel has a 20 % probability of being dropped.

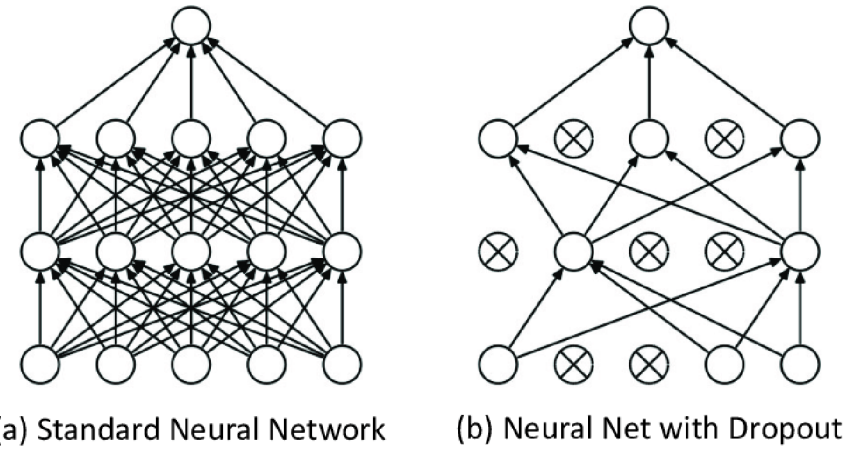

Figure 5.4: An illustration of the dropout mechanism within the proposed CNN. (a) Shows a standard neural network with 2 hidden layers. (b) Shows an example of a thinned network produced by applying dropout, where crossed units have been dropped. [80].

Last addition that I have made to the fire modules of the SqueezeNet is adding Batch normalization layer after the squeeze convolutional layer in the fire module because I wanted to increase the training speed of the model. Batch normalization is a technique that improves the training speed and model stability. Furthermore, batch normalization can act as a form of regularization that reduces overfitting. By normalizing the input by normalizing and scaling the output from the previous layer, batch normalization allows each layer of the network to learn more independently. During the training phase batch normalization reduces the so called internal covariate shift, which is the change in the distribution of inputs due to the updating of weights in previous layers. This normalization is performed for each training mini-batch and it involves calculating the mean and variance [84] [26].

When given inputs x for a mini-batch of size m, batch normalization transforms each input x_i as follows [71] [65]:

1. Firstly, the mean μ_B of the mini-batch is calculated.

$$\mu_B = \frac{1}{m}\sum_{i=1}^{m} x_i$$

2. After that, the variance σ_B^2 of the mini-batch is calculated.

$$\sigma_B^2 = \frac{1}{m}\sum_{i=1}^{m}(x_i - \mu_B)^2$$

3. Next the input data x_i are transformed into normalized data $\hat{x}_i$ using the calculated mean and variance.

$$\hat{x}_i = \frac{x_i - \mu_B}{\sqrt{\sigma_B^2 + \epsilon}}$$

Here, the ϵ is a very small constant that is added to avoid division by zero.

4. Lastly, the scaling and shifting transformations are applied using the γ and β parameters. These parameters are learned along with other parameters of the model during the training phase.

$$y_i = \gamma \, \hat{x}_i + \beta$$

The output y_i is the new normalized input into the next layer of the model.

The entire customized SqueezeNet architecture with bypass is implemented in the `SqueezeNet` and `fire_module_with_bypass` functions.

5.4 Training

After implementing the entire model architecture, the model needs to be trained. The configuration of the model for training is implemented using the `compile` method. The training itself is implemented using the `fit` method. As I described in section 4.4 another important part of implementation is choosing the loss function and optimizer which are used by the model during the training phase.

5.4.1 Loss Function – Cross Entropy

Entropy and cross entropy are concepts in information theory which are often used in machine learning for evaluating and optimizing the performance of classification models like CNNs, including models that were designed for multi-label classification tasks.

Entropy in the field of information theory and machine learning is a measure of unpredictability or randomness within a system. Specifically, it is the average uncertainty, surprise, or information characteristics to the possible outcomes. In short, entropy measures the uncertainty that is associated with random events within the system. The mathematical definition of entropy $H(X)$ is as follows [78] [88]:

$$H(X) = -\sum_{i=1}^{K} p(x_i) \log p(x_i)$$

Here the $p(x_i)$ represents the probability of class i occurring in the dataset. Constant K is the number of classes.

The cross entropy loss function is an important element in training CNNs, especially when the CNN is dealing with classification tasks that involve multiple classes, i.e. multi-label classification. This loss function measures the difference between the predicted probability output by the CNN and the true class, represented as a one-hot encoded array or vector. For each input, the CNN predicts probabilities across all classes of the provided dataset and the categorical cross entropy loss function computes the logarithm of the probability assigned to the true class. Essentially, this function penalizes the model more heavily when the model assigns a low probability to the correct class. This encourages the model to adjust its weights to increase the accuracy of its predictions [78] [88]. Mathematically the categorical cross entropy loss function L is defined as follows [4]:

$$L = -\sum_{i=1}^{K} y_i \log p(i)$$

Here the K constant is the number of classes. The y_i is the binary indicator, i.e. 0 or 1, if the class label is the correct classification for the current observation or image. The $p(i)$ is the predicted probability that the current observation or image belongs to class i.

5.4.2 Optimizer – Adam

Adam optimizer, i.e. Adaptive Moment Estimation, is an optimization algorithm used in training machine learning models. Adam is a popular optimizer that is used in many machine learning tasks because it is straightforward to implement, computationally efficient and requires little memory. It works well across a wide range of problems and typically converges faster than other optimizers [32].

Adam optimizer is a combination of two other optimization algorithms. Those being the Stochastic Gradient Descent with momentum, i.e. SGD with momentum, and Root Mean Squared Propagation, i.e. RMSProp. Momentum speeds up the training by accelerating gradients in the right direction by adding a fraction of the previous gradient to the current one. In RMSProp parameters with high gradients get smaller update steps, and low gradients get bigger update steps. In short, momentum is about accelerating in consistent directions and RMSProp is based on controlling the potential overshooting by modulating the step size [15].

Mathematical definition of Adam optimization algorithm is as follows [74] [15]:

1. Firstly, the gradient g_t is calculated using the derivative of the loss function L and derivative of weights w at time t.

$$g_t = \frac{\delta\ L}{\delta\ w_t}$$

2. Secondly, the momentum is calculated.

$$m_t = \beta_1 m_{t-1} + (1 - \beta_1) g_t$$

   Here the m_t is the average of the gradients. Using averages makes the algorithm converge towards the global minimum faster. The β_1 is a decay rate for the momentum, and typically it is set to 0.9.

3. Next, the RMSprop is calculated.

$$v_t = \beta_2 v_{t-1} + (1 - \beta_2) g_t$$

   Here the v_t is the average of the squared gradients. The squared gradients mean that when the variance of gradients is high the learning rate is reduced, and when the variance of gradients is low the learning rate is increased. The β_2 is a decay rate for the squared gradients, and typically it is set to 0.999.

4. Another step is to perform the so called bias correction for the calculated m_t and v_t.

$$\hat{m}_t = \frac{m_t}{1 - \beta_1^t}$$

$$\hat{v}_t = \frac{v_t}{1 - \beta_2^t}$$

   This is done so that the m_t and v_t will not be biased towards 0 and to prevent high oscillations near the global minimum.

5. Finally, the weights of the model are updated using the bias corrected m_t and v_t.

$$w_{t+1} = w_t - \alpha \frac{\hat{m}_t}{\sqrt{\hat{v}_t} + \epsilon}$$

Here, the α is the learning rate of the model and ϵ is a very small constant that is used to avoid division by zero.

5.4.3 Model Checkpoint

During the first couple training experiments I discovered that I should save the model during the training phase. This way I could train the model for more epochs and not worry about overfitting, as I would have saved the model which had the best accuracy and loss metrics.

For this exact purpose I used the `ModelCheckpoint` class from Keras library. This class allows to save the model during the training phase after certain conditions that I defined are met. I added parameters to the `ModelCheckpoint` class, so that it monitors model's accuracy on the validation dataset. Furthermore, I added a parameter that saves the model only when the accuracy on the validation dataset improves. This way I can train for more epochs and know that only the best performing model is saved.

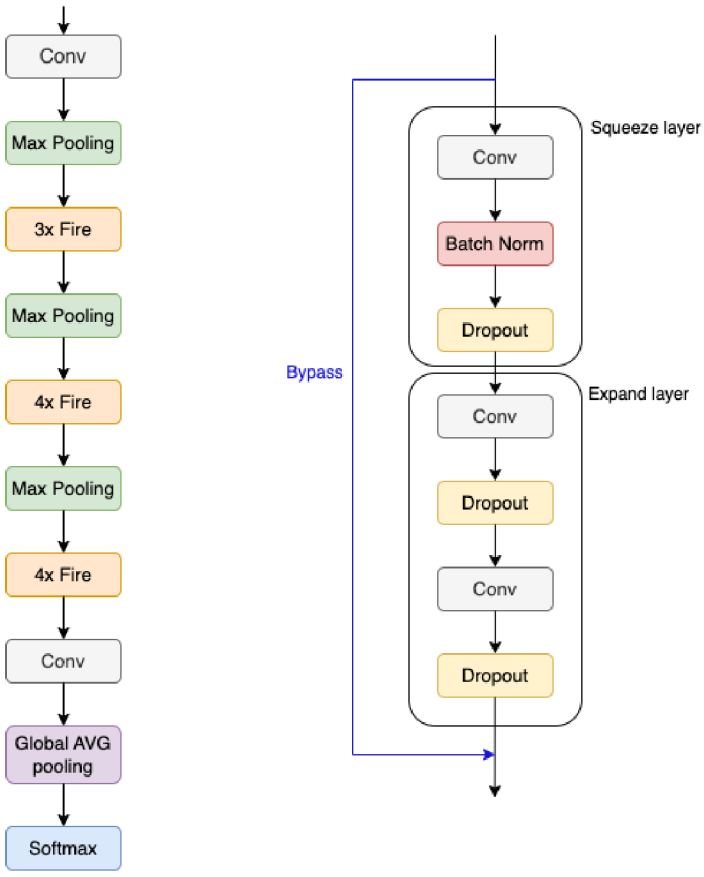

Figure 5.5: A diagram of the implemented architecture. Left: Architecture of the entire model. Right: Architecture of a single fire module.

Chapter 6

Training and Evaluation

As I already mentioned implementation of machine learning models is an iterative process that requires experimentation with various hyperparameters and architecture of the model.

First of all, all training and evaluation of models was done using either my laptop or Google Colab. On my laptop the models were trained on my laptop's CPU, which in this case is an Intel Core i9 2.3 GHz 8-core processor. In Google Colab I used various GPUs for training my models. This GPUs were the NVIDIa T4, NVIDIa L4 Tensor Core and NVIDIa A100 Tensor Core. I used the Google Colab GPUs to greatly increase the training speed, so that I could make more frequent modifications to the implemented models. The duration of a single epoch for the customized SqueezeNet architecture on my laptop's CPU was around 125 second and on Google Colab's L4 GPU it was around 25 seconds.

Before settling for the SqueezeNet architecture I tried using various different convolutional neural networks (CNN). The Keras library allows accessible and easy way to use well-known and widely used deep learning models [36].

6.1 ResNet50 and EfficientNet

After implementing all of the data preprocessing I started with the ResNet50 architecture. This architecture does not follow the specification that I set for a smaller network with fewer parameters, but I wanted to see how does a deeper network with a lot of parameters perform. I started with training the ResNet50 model that was already trained on the ImageNet dataset with just modifying the fully connected layers to fit my FER-2013 dataset and fine-tune it for my classification task. I quickly discovered that the model overfits extremely just after 20 epochs. After that, I added dropout to the fully connected layers, but the model still overfitted and was not very stable.

After that I gave up on ResNet50 and tried the EfficientNet architecture to see how it performs. The EfficientNet model performed better than the ResNet50, however it still overfitted slightly and the loss metric was too high. Furthermore, I wanted to use a smaller network with fewer parameters. Therefore, the ResNet50 and EfficientNet architectures were not suitable for my specification and applicable in in wide range of devices including devices with limited memory and computational capabilities.

6.2 VGG16 and VGG8

Next up I moved to a smaller and well-known VGG16 architecture. Firstly, I tried to fine-tune the VGG16 model that was trained on the ImageNet dataset. This approach performed similarly to the ResNet50 architecture in the fact that it overfitted. After that I tried to use a VGG16 architecture that was not pre-trained and implemented the VGG16 architecture using the Keras API, so that I could modify the entire architecture of the model. This way I added dropout layers after every convolutional block and fully connected layers. Furthermore, I added batch normalization at the beginning of every convolutional block. This model performed much better than the pre-trained one, although it still overfitted slightly and I wanted to try even smaller networks.

Subsequently, I tried to remove some convolutional layers from the VGG16 architecture to create a smaller VGG8 network. This architecture performed very similarly to the VGG16 architecture without considerable improvement in performance on the FER-2013 dataset. I also modified this VGG8 architecture with dropout layers after the fully connected layers and added batch normalization after every convolutional layer. Unfortunately, this model usually overfitted more than the VGG16 architecture and the accuracy metrics were similar.

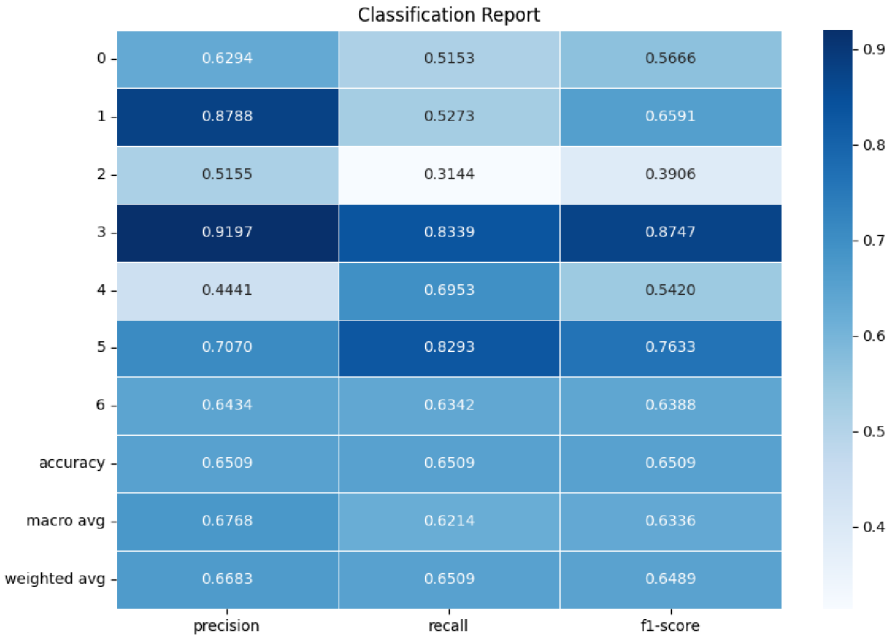

Figure 6.1: Classification report of the trained VGG16 architecture. Labels: 0 = Anger, 1 = Disgust, 2 = Fear, 3 = Happiness, 4 = Sadness, 5 = Surprise, 6 = Neutral.

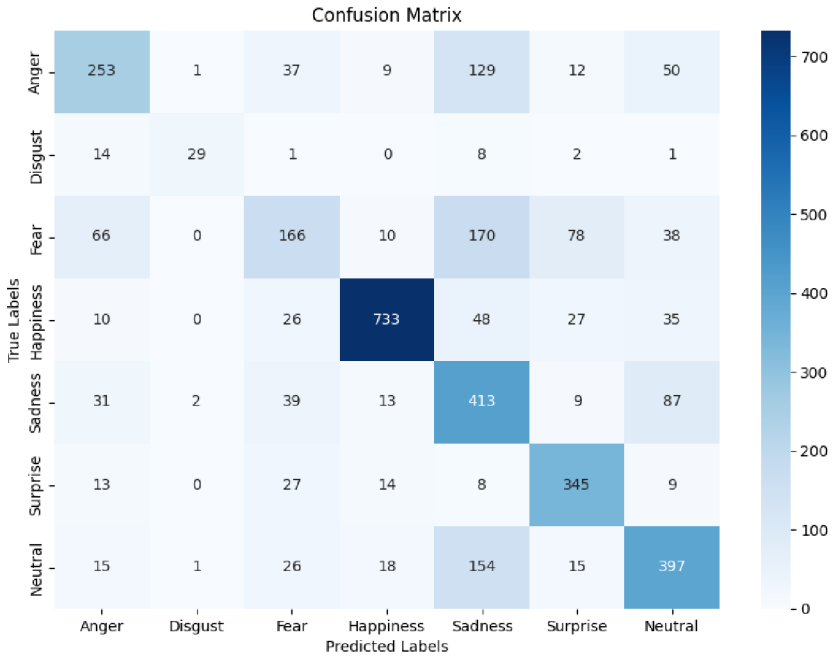

Figure 6.2: Confusion matrix of the trained VGG16 architecture.

6.3 SqueezeNet

After I tried the aforementioned architectures I decided to use the SqueezeNet architecture which was the perfect candidate based on my specification.

Firstly, I started experimenting and training using the original SqueezeNet architecture without any bypass connections between fire modules. However, like the other models this model suffered from overfitting and achieved lower accuracy than the other models. Therefore, I chose to implement the SqueezeNet architecture with bypass to combat the vanishing gradient problem and increase the accuracy. This proved to be a good step because the accuracy of the model improved, but the problem with overfitting still remained. That is why I also added the dropout layers with 0.2 dropout rate after each convolutional layer in the fire modules. This addition helped with the overfitting, but the accuracy of the model dropped slightly. To address this I also added the batch normalization layer after the first convolutional layer in the fire modules. This helped with the learning convergence of the model and slightly increased the accuracy. Lastly, I changed the batch size from 512 to 256 to enable the model to learn more from the training dataset and enhance the model's generalization capabilities.

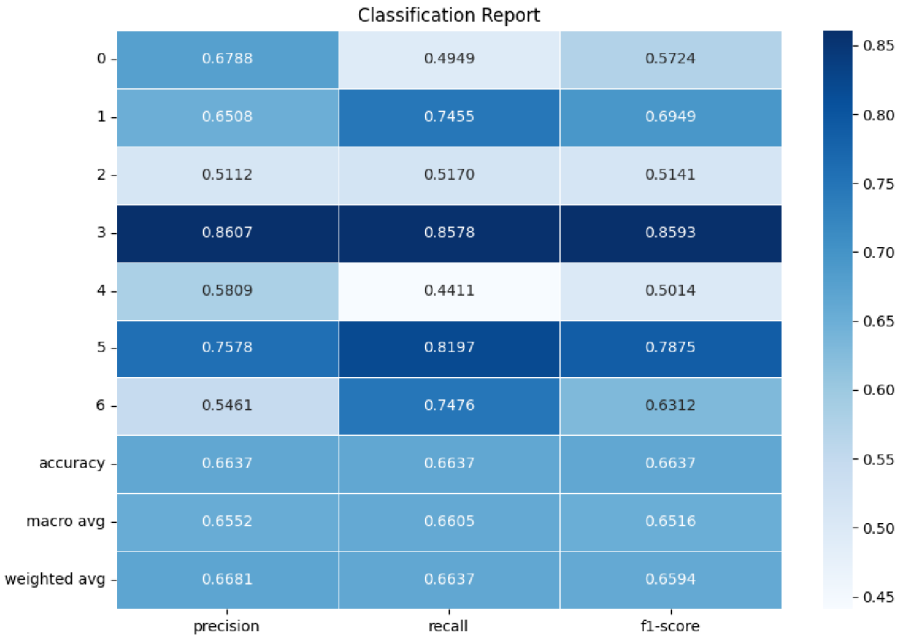

Figure 6.3: Classification report of the trained SqueezeNet architecture. Labels: 0 = Anger, 1 = Disgust, 2 = Fear, 3 = Happiness, 4 = Sadness, 5 = Surprise, 6 = Neutral.

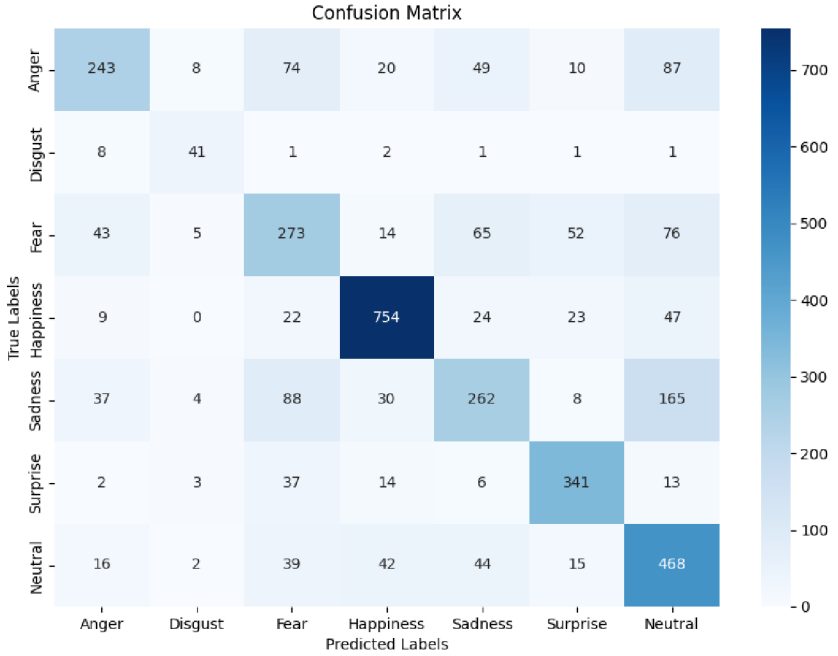

Figure 6.4: Confusion matrix of the trained SqueezeNet architecture.

6.4 Evaluation

In the classification report of the SqueezeNet architecture (6.3) the facial expressions of emotions with the lowest F1-scores are the expressions of sadness and fear. In the confusion matrix of the SqueezeNet architecture (6.4) we can see that the expression of sadness has been mostly confused with the neutral expression because they share common features. Furthermore, we can see that the expression of fear was the hardest to classify overall because the expression of fear has features which can make it easily confused with other emotions like surprise, sadness or even neutral emotion. On the other hand, the facial expression of emotions with the highest F1-scores are the expression of happiness, followed by the expression of surprise. The reason for having the highest F1-score is that the expressions of happiness and surprise have the most unique features out of all the expressions of emotions in the FER-2013 dataset which make them the easiest to classify.

In the classification report of the VGG16 architecture (6.1) the facial expressions of emotions with the lowest F1-scores are the expressions of fear and sadness. In the confusion matrix of the VGG16 architecture (6.2) we can see that the expression of sadness has been heavily confused with the neutral expression. Furthermore, we can see that the expression of anger was mostly confused with the expression of sadness, and that the neutral expression was mostly confused with the expression of sadness. On the other hand, the facial expression of emotions with the highest F1-scores are the expression of happiness, followed by the expression of surprise.

If we compare both confusion matrices, we can see that the SqueezeNet architecture makes less confusions between the facial expressions of emotions, as the confusion matrix of the SqueezeNet is more diagonal than the confusion matrix of VGG16. Next, if we compare the classification reports, we can see that the F1-score for the expression of fear is 12 % higher in the Squeezenet architecture. The F1-scores of other expressions are similar in both SqueezeNet and VGG16 architectures. The overall accuracy of the implemented SqueezeNet architecture is 66.37 % and the the accuracy of the implemented VGG16 architecture is 65.09 %. The implemented SqueezeNet model has 1 377 655 parameters and the size of the whole model is 5.26 MB. The implemented VGG16 model has 28 913 223 parameters and the size of the whole model is 110.3 MB.

Chapter 7

Conclusion

In this thesis I studied the psychological background of emotions and facial expressions. I found out that there are multiple approaches to categorizing and defining emotions and facial expressions. I researched the biological background of emotions and the anatomical background for the muscles of the face. Furthermore, I studied the sociological aspects of facial expression of emotions such as the universality of facial expressions, the so called microexpressions and the overall importance of facial expressions in our lives. Last but not least, I also researched the impact of certain diseases on facial expressions.

Another part of this thesis was designing and implementing an automated facial expression recognition system using machine learning. My main goal was to use or create a small CNN with fewer parameters, so that the model could be applicable in embedded devices with limited memory and computational resources. That is why after training and experimenting with various CNNs, such as Resnet and VGG16, I mainly focused on the SqueezeNet architecture which I customized to serve my specific requirements. The results of the customized SqueezeNet architecture prove that the SqueezeNet architecture can be utilized in facial expression recognition and it's results can be comparable to other much larger CNNs such as the VGG16.

In conclusion, this thesis provides valuable insights into facial expression recognition (FER). It establishes a foundation for future work aimed at not only enhancing the accuracy of automated FER systems but also at developing smaller, more efficient machine learning models suitable for use in embedded devices.

Chapter 8

Future Work

In future studies, I suggest using different dataset such as the AffectNet dataset for architectures used in this thesis or similar architectures and evaluate these architectures on the chosen dataset and compare the results.

Furthermore, it is essential to make further improvements in accuracy of automated facial expression recognition systems by experimenting with different smaller and more efficient networks such as MobileNet. Additionally, accuracy could be improved by further modifying the proposed or similar architectures with additional layers, hyperparameter optimization and utilizing lasso regression or ridge regression apart from dropout and batch normalization to combat overfitting.

Last but not least, since this thesis also puts an emphasis on the usage of automated FER systems in embedded devices, it is important to research the capabilities and applicability of facial expression recognition systems on devices such as Raspberry Pi.

Bibliography

[1] Accuracy vs. precision vs. recall in machine learning: what's the difference? *EvidentlyAI*. EvidentlyAI. [Online; viewed 09.04.2024]. Available at: https://www.evidentlyai.com/classification-metrics/accuracy-precision-recall.

[2] One-hot Encoding. *Deepchecks*. Deepchecks. [Online; viewed 08.04.2024]. Available at: https://deepchecks.com/glossary/one-hot-encoding/.

[3] Training, Validation and Test Sets: How To Split Machine Learning Data. *KILI TECHNOLOGY*. KILI TECHNOLOGY. [Online; viewed 08.04.2024]. Available at: https://kili-technology.com/training-data/training-validation-and-test-sets-how-to-split-machine-learning-data.

[4] Loss Functions. *ML Glossary*. ML Glossary. 2017. [Online; viewed 13.04.2024]. Available at: https://ml-cheatsheet.readthedocs.io/en/latest/loss_functions.html.

[5] The Biology of Emotions. University of Central Florida. 2020. [Online; viewed 19.10.2023]. Available at: https://pressbooks.online.ucf.edu/lumenpsychology/chapter/the-biology-of-emotions/.

[6] Bell's Palsy. *National Institute of Neurological Disorders and Stroke*. november 2023. [Online; viewed 9.11.2023]. Available at: https://www.ninds.nih.gov/health-information/disorders/bells-palsy.

[7] Epochs, Batch Size, Iterations - How are They Important to Training AI and Deep Learning Models. *SabrePC*. SabrePC. february 2023. [Online; viewed 09.04.2024]. Available at: https://www.sabrepc.com/blog/Deep-Learning-and-AI/Epochs-Batch-Size-Iterations.

[8] Rectified Linear Unit (ReLU). *Deepchecks*. deepchecks. june 2023. [Online; viewed 11.04.2024]. Available at: https://deepchecks.com/glossary/rectified-linear-unit-relu/.

[9] What are Data augmentation techniques : [2024 update]. *UbiAI*. ubiAI. november 2023. [Online; viewed 09.04.2024]. Available at: https://ubiai.tools/what-are-the-advantages-anddisadvantages-of-data-augmentation-2023-update/.

[10] Muscles of Mastication — Physiopedia,. *Physiopedia*. Physiopedia. 2024. [Online; viewed 26.10.2023]. Available at: https://www.physio-pedia.com/Muscles_of_Mastication.

[11] Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z. et al. *TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems*. 2015. Available at: https://www.tensorflow.org/.

[12] Abdullah, S. M. S. and Abdulazeez, A. M. Facial Expression Recognition Based on Deep Learning Convolution Neural Network: A Review. *Journal of Soft Computing and Data Mining*. Penerbit UTHM. april 2021, vol. 2, no. 1. DOI: 10.30880/jscdm.2021.02.01.006. ISSN 2716-621X.

[13] Adolphs, R. Recognizing Emotion from Facial Expressions: Psychological and Neurological Mechanisms. *Behavioral and Cognitive Neuroscience Reviews*. 2002, vol. 1, no. 1, p. 21–62. DOI: 10.1177/1534582302001001003. PMID: 17715585.

[14] Adolphs, R. Emotion. *Current Biology*. Elsevier BV. july 2010, vol. 20, no. 13, p. R549–R552. DOI: 10.1016/j.cub.2010.05.046. ISSN 0960-9822.

[15] Agarwal, R. Complete Guide to the Adam Optimization Algorithm. *Built In*. Built In. september 2023. [Online; viewed 13.04.2024]. Available at: https://builtin.com/machine-learning/adam-optimization.

[16] Alake, R. Loss Functions in Machine Learning Explained. *Datacamp*. Datacamp. november 2023. [Online; viewed 09.04.2024]. Available at: https://www.datacamp.com/tutorial/loss-function-in-machine-learning.

[17] Arham, M. Diving into the Pool: Unraveling the Magic of CNN Pooling Layers. *KDnuggets*. KDnuggets. september 2023. [Online; viewed 11.04.2024]. Available at: https://www.kdnuggets.com/diving-into-the-pool-unraveling-the-magic-of-cnn-pooling-layers.

[18] avidrishik. A Guide to SqueezeNet Architecture: Compressed Neural Network. *Medium*. Medium. august 2021. [Online; viewed 11.04.2024]. Available at: https://medium.com/@avidrishik/squeezenets-architecture-compressed-neural-network-7741d24ca56f.

[19] Awan, A. A. A Complete Guide to Data Augmentation. *Datacamp*. Datacamp. november 2022. [Online; viewed 09.04.2024]. Available at: https://www.datacamp.com/tutorial/complete-guide-data-augmentation.

[20] baeldung and Aibin, M. How ReLU and Dropout Layers Work in CNNs. *Baeldung*. Baeldung. march 2024. [Online; viewed 12.04.2024]. Available at: https://www.baeldung.com/cs/ml-relu-dropout-layers.

[21] Baheti, P. Activation Functions in Neural Networks [12 Types and Use Cases]. *V7labs*. v7labs. may 2021. [Online; viewed 09.04.2024]. Available at: https://www.v7labs.com/blog/neural-networks-activation-functions.

[22] Barrett, L. F. *How emotions are made*. Houghton Mifflin, march 2017. ISBN 9780544133310.

[23] Basit, H., Tariq, M. A. and Siccardi, M. A. *Anatomy, Head and Neck, Mastication Muscles*. Treasure Island (FL): StatPearls Publishing, june 2023. [Online; viewed 26.10.2023]. Available at: https://www.ncbi.nlm.nih.gov/books/NBK541027/.

[24] Bhandari, A. Understanding and Interpreting Confusion Matrix in Machine Learning (Updated 2024). *Analytics Vidhya*. AnalyticsVidhya. january 2024. [Online; viewed 09.04.2024]. Available at: https://www.analyticsvidhya.com/blog/2020/04/confusion-matrix-machine-learning/.

[25] Bourke, C., Douglas, K. and Porter, R. Processing of Facial Emotion Expression in Major Depression: A Review. *Australian & New Zealand Journal of Psychiatry*. SAGE Publications. august 2010, vol. 44, no. 8, p. 681–696. DOI: 10.3109/00048674.2010.496359. ISSN 1440-1614.

[26] Brownlee, J. A Gentle Introduction to Batch Normalization for Deep Neural Networks. *Machine Learning Mastery*. Machine Learning Mastery. december 2019. [Online; viewed 12.04.2024]. Available at: https://machinelearningmastery.com/batch-normalization-for-training-of-deep-neural-networks/.

[27] Brownlee, J. A Gentle Introduction to Dropout for Regularizing Deep Neural Networks. *Machine Learning Mastery*. Machine Learning Mastery. august 2019. [Online; viewed 12.04.2024]. Available at: https://machinelearningmastery.com/dropout-for-regularizing-deep-neural-networks/.

[28] Brownlee, J. A Gentle Introduction to Pooling Layers for Convolutional Neural Networks. *Machine Learning Mastery*. Machine Learning Mastery. july 2019. [Online; viewed 11.04.2024]. Available at: https://machinelearningmastery.com/pooling-layers-for-convolutional-neural-networks/.

[29] Brownlee, J. A Gentle Introduction to the Rectified Linear Unit (ReLU). *Machine Learning Mastery*. Machine Learning Mastery. august 2020. [Online; viewed 11.04.2024]. Available at: https://machinelearningmastery.com/rectified-linear-activation-function-for-deep-learning-neural-networks/.

[30] Brownlee, J. How Do Convolutional Layers Work in Deep Learning Neural Networks? *Machine Learning Mastery*. Machine Learning Mastery. april 2020. [Online; viewed 11.04.2024]. Available at: https://machinelearningmastery.com/convolutional-layers-for-deep-learning-neural-networks/.

[31] Brownlee, J. What is the Difference Between Test and Validation Datasets? *Machine Learning Mastery*. Machine Learning Mastery. august 2020. [Online; viewed 08.04.2024]. Available at: https://machinelearningmastery.com/difference-test-validation-datasets/.

[32] Brownlee, J. Gentle Introduction to the Adam Optimization Algorithm for Deep Learning. *Machine Learning Mastery*. Machine Learning Mastery. january 2021. [Online; viewed 13.04.2024]. Available at: https://machinelearningmastery.com/adam-optimization-algorithm-for-deep-learning/.

[33] Brownlee, J. Difference Between a Batch and an Epoch in a Neural Network. *Machine Learning Mastery*. Machine Learning Mastery. august 2022. [Online; viewed 09.04.2024]. Available at: https://machinelearningmastery.com/difference-between-a-batch-and-an-epoch/.

[34] Carrier, P.-L. and Courville, A. FER2013 Dataset. *Deeplake*. Deeplake. Available at: https://datasets.activeloop.ai/docs/ml/datasets/fer2013-dataset/.

[35] Chollet, F. et al. *Keras*. 2015. Available at: https://keras.io.

[36] Chollet, F. et al. *Keras Applications*. 2015. Available at: https://keras.io/api/applications/.

[37] Chouhayebi, H., Riffi, J., Mahraz, M. A., Yahyaouy, A. and Tairi, H. Facial expression recognition Using Machine Learning. In: *2021 Fifth International Conference On Intelligent Computing in Data Sciences (ICDS)*. IEEE, October 2021. DOI: 10.1109/icds53782.2021.9626709.

[38] Cowen, A. S. and Keltner, D. Self-report captures 27 distinct categories of emotion bridged by continuous gradients. *Proceedings of the National Academy of Sciences*. Proceedings of the National Academy of Sciences. september 2017, vol. 114, no. 38. DOI: 10.1073/pnas.1702247114. ISSN 1091-6490.

[39] Curry, L. The Facial Nerve (CN VII). *TeachMe Anatomy*. TeachMe Anatomy. july 2023. [Online; viewed 24.10.2023]. Available at: https://teachmeanatomy.info/head/cranial-nerves/facial-nerve/.

[40] Devansh. How does Batch Size impact your model learning. *Medium*. Medium. january 2022. [Online; viewed 09.04.2024]. Available at: https://medium.com/geekculture/how-does-batch-size-impact-your-model-learning-2dd34d9fb1fa.

[41] Dilmegani, C. What is Data Augmentation? Techniques and Examples in 2024. *AIMultiple*. AIMultiple. december 2023. [Online; viewed 09.04.2024]. Available at: https://research.aimultiple.com/data-augmentation/.

[42] Dulak, D. and Naqvi, I. A. *Neuroanatomy, Cranial Nerve 7 (Facial)*. Treasure Island (FL): StatPearls Publishing, july 2023. [Online; viewed 20.10.2023]. Available at: https://www.ncbi.nlm.nih.gov/books/NBK526119/.

[43] Eelbode, T., Sinonquel, P., Maes, F. and Bisschops, R. Pitfalls in training and validation of deep learning systems. *Best Practice Research Clinical Gastroenterology*. Elsevier BV. june 2021, 52–53. DOI: 10.1016/j.bpg.2020.101712. ISSN 1521-6918.

[44] Ekman, P. Are there basic emotions? *Psychological Review*. American Psychological Association (APA). 1992, vol. 99, no. 3, p. 550–553. DOI: 10.1037/0033-295x.99.3.550. ISSN 0033-295X.

[45] Ekman, P. and Friesen, W. V. Nonverbal Leakage and Clues to Deception†. *Psychiatry*. Informa UK Limited. february 1969, vol. 32, no. 1, p. 88–106. DOI: 10.1080/00332747.1969.11023575. ISSN 1943-281X.

[46] Fox, E. Perspectives from affective science on understanding the nature of emotion. *Brain and Neuroscience Advances*. SAGE Publications. january 2018, vol. 2. DOI: 10.1177/2398212818812628. ISSN 2398-2128.

[47] Frith, C. Role of facial expressions in social interactions. *Philosophical Transactions of the Royal Society B: Biological Sciences*. The Royal Society. december 2009, vol. 364, no. 1535, p. 3453–3458. DOI: 10.1098/rstb.2009.0142. ISSN 1471-2970.

[48] Gupta, A. A Comprehensive Guide on Optimizers in Deep Learning. *Analytics Vidhya*. AnalyticsVidhya. january 2024. [Online; viewed 09.04.2024]. Available at: https://www.analyticsvidhya.com/blog/2021/10/a-comprehensive-guide-on-deep-learning-optimizers/.

[49] Hambardzumyan, S., Tuli, A., Ghukasyan, L., Rahman, F., Topchyan, H. et al. Deep Lake: a Lakehouse for Deep Learning. 2023. Available at: https://pypi.org/project/deeplake/.

[50] Harmon Jones, E., Harmon Jones, C. and Summerell, E. On the Importance of Both Dimensional and Discrete Models of Emotion. *Behavioral Sciences*. MDPI AG. september 2017, vol. 7, no. 4, p. 66. DOI: 10.3390/bs7040066. ISSN 2076-328X.

[51] Heilman, K. M. and Nadeau, S. E. Emotional and Neuropsychiatric Disorders Associated with Alzheimer's Disease. *Neurotherapeutics*. Elsevier BV. january 2022, vol. 19, no. 1, p. 99–116. DOI: 10.1007/s13311-021-01172-w. ISSN 1878-7479.

[52] Huff, T., Weisbrod, L. J. and Daly, D. T. *Neuroanatomy, Cranial Nerve 5 (Trigeminal)*. Treasure Island (FL): StatPearls Publishing, november 2022. [Online; viewed 24.10.2023]. Available at: https://www.ncbi.nlm.nih.gov/books/NBK482283/.

[53] Huilgol, P. Precision and Recall | Essential Metrics for Machine Learning (2024 Update). *AnalyticsVidhya*. AnalyticsVidhya. december 2023. [Online; viewed 09.04.2024]. Available at: https://www.analyticsvidhya.com/blog/2020/09/precision-recall-machine-learning/.

[54] Hwang, H. and Matsumoto, D. Evidence for the Universality of Facial Expressions of Emotion. In: Mandal, M. K. and Awasthi, A., ed. *Understanding Facial Expressions in Communication: Cross-cultural and Multidisciplinary Perspectives*. New Delhi: Springer India, 2015, p. 41–56. DOI: 10.1007/978-81-322-1934-7_3. ISBN 978-81-322-1934-7.

[55] Iandola, F. N., Han, S., Moskewicz, M. W., Ashraf, K., Dally, W. J. et al. *SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and <0.5MB model size*. arXiv, 2016. DOI: 10.48550/ARXIV.1602.07360. Available at: https://arxiv.org/abs/1602.07360.

[56] Jack, R. E., Garrod, O. G. B., Yu, H., Caldara, R. and Schyns, P. G. Facial expressions of emotion are not culturally universal. *Proceedings of the National Academy of Sciences*. Proceedings of the National Academy of Sciences. april 2012, vol. 109, no. 19, p. 7241–7244. DOI: 10.1073/pnas.1200155109. ISSN 1091-6490.

[57] Jacob, T. Vanishing Gradient Problem: Causes, Consequences, and Solutions. *KDnuggets*. KDnuggets. june 2023. [Online; viewed 11.04.2024]. Available at: https://www.kdnuggets.com/2022/02/vanishing-gradient-problem.html.

[58] Jain, V. Everything you need to know about "Activation Functions" in Deep learning models. *Towards Data Science*. Towards Data Science. december 2019. [Online; viewed 09.04.2024]. Available at: https://towardsdatascience.com/everything-you-need-to-know-about-activation-functions-in-deep-learning-models-84ba9f82c253.

[59] Jaiswal, S. What is Normalization in Machine Learning? A Comprehensive Guide to Data Rescaling. *Datacamp*. Datacamp. january 2024. [Online; viewed 08.04.2024]. Available at: https://www.datacamp.com/tutorial/normalization-in-machine-learning.

[60] Jones, O. The Muscles of Facial Expression. *TeachMe Anatomy*. december 2023. [Online; viewed 26.10.2023]. Available at: https://teachmeanatomy.info/head/muscles/facial-expression/.

[61] Klingner, C. M. and Guntinas Lichius, O. Mimik und Emotion. *Laryngo-Rhino-Otologie*. Georg Thieme Verlag KG. may 2023, vol. 102, S 01, p. S115–S125. DOI: 10.1055/a-2003-5687. ISSN 1438-8685.

[62] Li, H., Lin, Z., Shen, X., Brandt, J. and Hua, G. A convolutional neural network cascade for face detection. In: *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, June 2015. DOI: 10.1109/cvpr.2015.7299170.

[63] Li, S. and Deng, W. Deep Facial Expression Recognition: A Survey. *IEEE Transactions on Affective Computing*. Institute of Electrical and Electronics Engineers (IEEE). july 2022, vol. 13, no. 3, p. 1195–1215. DOI: 10.1109/taffc.2020.2981446. ISSN 2371-9850.

[64] Lindquist, K. A., Wager, T. D., Kober, H., Bliss Moreau, E. and Barrett, L. F. The brain basis of emotion: A meta-analytic review. *Behavioral and Brain Sciences*. Cambridge University Press (CUP). may 2012, vol. 35, no. 3, p. 121–143. DOI: 10.1017/s0140525x11000446. ISSN 1469-1825.

[65] Mao, L. Batch Normalization Explained. *Lei Mao's Log Book*. Lei Mao's Log Book. july 2019. [Online; viewed 12.04.2024]. Available at: https://leimao.github.io/blog/Batch-Normalization/.

[66] Matsumoto, D. and Hwang, H. S. Evidence for training the ability to read microexpressions of emotion. *Motivation and Emotion*. Springer Science and Business Media LLC. april 2011, vol. 35, no. 2, p. 181–191. DOI: 10.1007/s11031-011-9212-2. ISSN 1573-6644.

[67] McLellan, T., Johnston, L., Dalrymple Alford, J. and Porter, R. The recognition of facial expressions of emotion in Alzheimer's disease: a review of findings. *Acta Neuropsychiatrica*. Cambridge University Press (CUP). october 2008, vol. 20, no. 5, p. 236–250. DOI: 10.1111/j.1601-5215.2008.00315.x. ISSN 1601-5215.

[68] Mollahosseini, A., Chan, D. and Mahoor, M. H. Going deeper in facial expression recognition using deep neural networks. IEEE. march 2016, p. 1–10. DOI: 10.1109/WACV.2016.7477450.

[69] Padilla, M. How to Measure Orofacial Pain With a Muscle Tenderness Exam. *Herman Ostrow School of Dentistry of USC*. [Online; viewed 26.10.2023]. Available at: https://ostrowonline.usc.edu/orofacial-pain-muscle-tenderness-exam/.

[70] Pazhaniappan, N. The Trigeminal Nerve (CN V). *TeachMe Anatomy*. TeachMe Anatomy. july 2023. [Online; viewed 24.10.2023]. Available at: https://teachmeanatomy.info/head/cranial-nerves/trigeminal-nerve/.

[71] Peccia, F. Batch normalization: theory and how to use it with Tensorflow. *Towards Data Science*. Towards Data Science. september 2018. [Online; viewed 12.04.2024]. Available at: https://towardsdatascience.com/batch-normalization-theory-and-how-to-use-it-with-tensorflow-1892ca0173ad.

[72] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B. et al. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*. 2011, vol. 12, p. 2825–2830. Available at: https://scikit-learn.org/stable/index.html.

[73] POSNER, J., RUSSELL, J. A. and PETERSON, B. S. The circumplex model of affect: An integrative approach to affective neuroscience, cognitive development, and psychopathology. *Development and Psychopathology*. Cambridge University Press (CUP). september 2005, vol. 17, no. 03. DOI: 10.1017/s0954579405050340. ISSN 1469-2198.

[74] prakharr0y. What is Adam Optimizer? *GeeksforGeeks*. GeeksforGeeks. march 2024. [Online; viewed 13.04.2024]. Available at: https://www.geeksforgeeks.org/adam-optimizer/.

[75] Prenger, M. T. M., Madray, R., Van Hedger, K., Anello, M. and MacDonald, P. A. Social Symptoms of Parkinson's Disease. *Parkinson's Disease*. Hindawi Limited. december 2020, vol. 2020, p. 1–10. DOI: 10.1155/2020/8846544. ISSN 2090-8083.

[76] Pressman, P. and Metrus, N. R. The Science of Emotions. *Verywell Health*. Verywell Health. february 2022. [Online; viewed 19.10.2023]. Available at: https://www.verywellhealth.com/the-science-of-emotions-2488708.

[77] Punia, R. How to Measure the Performance of Your Machine Learning Models: Precision, Recall, Accuracy, and F1 Score. *Medium*. Medium. april 2023. [Online; viewed 09.04.2024]. Available at: https://medium.com/@mycodingmantras/how-to-measure-the-performance-of-your-machine-learning-models-precision-recall-accuracy-and-f1-855702df048b.

[78] Pykes, K. Cross-Entropy Loss Function in Machine Learning: Enhancing Model Accuracy. *Datacamp*. datacamp. january 2024. [Online; viewed 13.04.2024]. Available at: https://www.datacamp.com/tutorial/the-cross-entropy-loss-function-in-machine-learning.

[79] Raut, N. Facial emotion recognition using machine learning. San Jose State University Library. 2019. DOI: 10.31979/etd.w5fs-s8wd.

[80] Roffo, G. *An illustration of the dropout mechanism within the proposed CNN*. ResearchGate, may 2017. [Online; viewed 12.04.2024]. Available at: https://www.researchgate.net/figure/9-An-illustration-of-the-dropout-mechanism-within-the-proposed-CNN-a-Shows-a_fig23_317277576.

[81] Rudovic, O. *Machine learning techniques for automated analysis of facial expressions*. Imperial College London, december 2013. DOI: 10.25560/24677. Available at: http://spiral.imperial.ac.uk/handle/10044/1/24677.

[82] Sambare, M. FER-2013. kaggle. 2020. Available at: https://www.kaggle.com/datasets/msambare/fer2013/data.

[83] Saxena, S. Introduction to Softmax for Neural Network. *AnalyticsVidhya*. AnalyticsVidhya. october 2023. [Online; viewed 11.04.2024]. Available at:

https://www.analyticsvidhya.com/blog/2021/04/introduction-to-softmax-for-neural-network/.

[84] Saxena, S. Introduction to Batch Normalization. *Analytics Vidhya*. AnalyticsVidhya. february 2024. [Online; viewed 12.04.2024]. Available at: https://www.analyticsvidhya.com/blog/2021/03/introduction-to-batch-normalization/.

[85] Scheve, T. What are microexpressions? *HowStuffWorks*. november 2023. [Online; viewed 01.03.2024]. Available at: https://science.howstuffworks.com/life/microexpression.htm.

[86] Segal, J., Smith, M., Robinson, L. and Boose, G. Nonverbal Communication and Body Language. *HelpGuide*. [Online; viewed 01.03.2024]. Available at: https://www.helpguide.org/articles/relationships-communication/nonverbal-communication.htm.

[87] Seladi Schulman, J. and Kapur, S. S. *What Part of the Brain Controls Emotions?* Healthline, july 2018. [Online; viewed 19.10.2023]. Available at: https://www.healthline.com/health/what-part-of-the-brain-controls-emotions.

[88] Shah, D. Cross Entropy Loss: Intro, Applications, Code. *V7Labs*. V7Labs. january 2023. [Online; viewed 13.04.2024]. Available at: https://www.v7labs.com/blog/cross-entropy-loss-guide.

[89] Shankar297. Understanding Loss Function in Deep Learning. *Analytics Vidhya*. AnalyticsVidhya. april 2024. [Online; viewed 09.04.2024]. Available at: https://www.analyticsvidhya.com/blog/2022/06/understanding-loss-function-in-deep-learning/.

[90] Shennan, J. Muscles of Facial Expression. *Geeky Medics*. april 2020. [Online; viewed 26.10.2023]. Available at: https://geekymedics.com/muscles-of-facial-expression/.

[91] Singh, K. How to Improve Class Imbalance using Class Weights in Machine Learning? *Analytics Vidhya*. AnalyticsVidhya. july 2023. [Online; viewed 10.04.2024]. Available at: https://www.analyticsvidhya.com/blog/2020/10/improve-class-imbalance-class-weights/.

[92] Westbrook, K. E., Nessel, T. A., Hohman, M. H. and Varacallo, M. *Anatomy, Head and Neck: Facial Muscles*. Treasure Island (FL): StatPearls Publishing, september 2022. [Online; viewed 20.10.2023]. Available at: https://www.ncbi.nlm.nih.gov/books/NBK493209/.

[93] Wolf, K. Measuring facial expression of emotion. *Dialogues in Clinical Neuroscience*. Informa UK Limited. december 2015, vol. 17, no. 4, p. 457–462. DOI: 10.31887/dcns.2015.17.4/kwolf. ISSN 1958-5969.

[94] Yakura, H. *Outline of the convolutional layer*. ResearchGate, march 2018. [Online; viewed 11.04.2024]. Available at: https://www.researchgate.net/figure/Outline-of-the-convolutional-layer_fig1_323792694.