Hochschule Neu-Ulm
University of Applied Sciences

Masterthesis

in the study course Artificial Intelligence and Data Analytics

Evaluation of Explainable AI to improve the interpretability of Facial Emotion Recognition

submitted by

Selina Lorch

Erstkorrektor:	Prof. Dr. Philipp Brune
Zweitkorrektor:	Prof. Dr. Stefan Faußer
Anschrift Verfasser:	Herdbruckerstraße 14, 89073 Ulm selina.lorch@student.hnu.de
Thema erhalten:	December 4, 2023
Arbeit eingereicht:	June 4, 2024

Abstract

Explainable Artificial Intelligence (XAI) has received considerable attention in recent years for enhancing the interpretability of machine learning models, particularly in the field of Facial Emotion Recognition (FER).
While facial emotion recognition models have achieved impressive levels of accuracy, their black-box nature presents challenges in understanding the underlying decision-making processes. This thesis examines the potential of XAI methods, specifically SHAP (Shapley Additive Explanations) (S. M. Lundberg and Lee 2017) and LIME (Local Interpretable Model-agnostic Explanations) (Marco Túlio Ribeiro, Singh, and Guestrin 2016), to enhance the interpretability of FER models.

A comparative evaluation was conducted using two established facial image datasets: FER 2013 (Kumar and Reddy n.d.) and RAF-DB (Bobojanov et al. 2023). The objective of the analysis was to examine the potential of XAI methods to idenitfy the key facial features that contribute to the models' predictions.
The results demonstrate that SHAP and LIME provide valuable insights into the models' decision-making processes. SHAP offers more consistent and stable visualizations than LIME. Both methods identified key facial regions, including the mouth, eyes, and cheeks, which align with known facial action units (Paul Ekman and Friesen 1971). This validates the models' interpretability.

The findings indicate the potential of XAI methods to bridge the gap between accuracy and interpretability in FER models, thereby contributing to the development of more transparent and trustworthy AI systems.
This research contributes to the advancement of XAI in FER by underscoring the importance of addressing ethical considerations in the deployment of these technologies (Cabitza, Campagner, and Mattioli 2022).
Future research should investigate the integration of additional XAI techniques, the utilization of diverse datasets, and the ethical implications of FER in real-world applications.

Contents

List of Figures

List of Tables

List of Abbreviations

AI Artificial Intelligence

AU Action Units

CK+ Cohn-Kanade Plus

CNN Convolutional Neural Network

DL Deep Learning

EM Expectation-Maximization

FACS Facial Action Coding System

FER Facial Emotion Recognition

GAIN Generative Adversarial Interpretations

Grad-CAM Gradient-weighted Class Activation Mapping

Guided Grad-CAM Guided Gradient-weighted Class Activation Mapping

HCI Human-Computer Interaction

LDL Label Distribution Learning

LIME Local Interpretable Model-Agnostic Explanations

ML Machine Learning

RAF-DB Real-world Affecitve Faces Database

ReLU Rectified Linear Unit

SFEW Static Facial Expressions in the Wild

SHAP Shapley Additive Explanations

SLIC Simple Linear Iterative Clustering

SVM Support Vector Machine

XAI Explainable Artificial Intelligence

1 Introduction

1.1 Background and Motivation

New Artificial Intelligence (AI) and Computer Vision technologies are constantly emerging. With these fast-paced advancements, numerous studies have been conducted with a focus on recognizing human emotions from facial images. However, accurately recognizing human emotions remains a challenge. The main problem with current AI models is their inability to provide a transparent and explainable decision-making process for the emotion recognition.

Facial emotions are outward expression of an internal emotional state. They play a crucial role in human communication and interaction (Kuttenreich, Piekartz, and Heim 2022). Therefore, the ability to recognize facial emotions is important in order to understand and respond to the emotional state of others (Gultekin et al. 2016). Several Facial Emotion Recognition (FER) models were developed to aid in this task.

FER models frequently employ Deep Learning (DL) techniques, notably Convolutional Neural Network (CNN), which have shown remarkable process in recognizing human emotions (Gao 2022). The advantage of DL methods is that they are self-learning systems which means that they can study non-linear relationships between the raw data and the target classes (Weitz et al. 2019).

Unfortunately, CNNs are often constructed as black boxes meaning they operate without disclosing how and why they recognize or misconstrue human emotions. This absence of explanation leads to practical and ethical issues (Guidotti et al. 2018).

FER has various applications in healthcare, Human-Computer Interaction (HCI) and marketing strategies. In terms of mental health diagnosis, the incorrect interpretation of human emotions can result in inaccurate diagnoses that may possibly impede the patient's treatment outcomes.

Currently, mental health institutions typically have numerous patients but inadequate staffing capacity. Therefore, models like the FER model could assist staff as long as the model's decisions are understandable. (Weitz et al. 2019).

Similarly, in HCI, a lack of interpretability may result in flawed user experiences or misguided responses, hindering effective communication between humans and machines. Furthermore, using inaccurate models of emotion recognition can lead to ineffective marketing and advertising strategies that impact consumer engagement and satisfaction.

To enhance the interpretability of those models means to understand the underlying decision-making process of the model and therefore to understand the internal logic (Guidotti et al. 2018).

As a result, there is a growing focus on developing methods to explain black box models and improve their interpretability (S. Luo et al. 2021). These methods are called Explainable Artificial Intelligence (XAI) methods. XAI therefore refers to the category

of AI that aims to provide a transparent and understandable decision-making process of AI models, addressing the issues of black box models (Gunning et al. 2019; Linardatos, Papastefanopoulos, and Kotsiantis 2021).
The goal of XAI methods is to improve the interpretability of DL models and to ensure that the user can comprehend, trust and effectively manage the output of AI systems. Applying those methods should provide human understandable explanations.
In the case of FER models, understanding the decision-making process of the model can provide an understanding on how to improve the model's performance and limitations (Shan Li and Weihong Deng 2022). Approaches like Shapley Additive Explanations (SHAP) and Local Interpretable Model-Agnostic Explanations (LIME) methods attempt to highlight the relevant regions in an image which can contribute to the FER model's prediction, therefore the application of XAI methods could potentially reveal which facial features are important for recognizing a specific emotion.
However, despite the potential of XAI methods, their application in FER remains relatively unexplored. Furthermore, the application of such methods remains a challenge in various domains (Adhi Tama et al. 2022) and there is a lack of a standardized evaluation criterion for these methods.
The performance of an AI model can be measured by its accuracy, precision and recall however, the XAI methods do not provide an evaluation metrics assessing the interpretability of the model.
Thus, the problem revolves around developing robust evaluation methods that quantify the extent to which XAI methods can enhance the interpretability of AI models and bridging the gap between accuracy and interpretability in emotion recognition models for facial images.
The challenge lies in developing a FER model that not only achieves high accuracy but also provides explanations that are comprehensible and actionable for the users (Shingjergji et al. 2022).

1.2 Overview of the work

This thesis examines the potential of XAI methods to enhance the interpretability of FER models.
The primary objective is to bridge the gap between the high accuracy of FER models and their interpretability, thereby making their decision-making processes more transparent and trustworthy for users. The work is structured into the following key sections:

1. Introduction:
   This section provides an overview of the importance of interpretability in AI models, particularly in the context of FER. It introduces the concept of XAI and its relevance to understanding and trusting FER models.

2. Literature Review:
   This section presents a review of existing research on FER, including its development, psychological theories, and the technical aspects of image classification and Machine Learning (ML) techniques.
   Furthermore, the examination of black-box models in FER is conducted, along with an investigation into the emerging field of XAI, which highlights methods such as SHAP and LIME.

3. Methodology:
   This section outlines the research methodology, including the technical environment, data collection, and pre-processing steps. It outlines the architectural design of the FER models employed and the implementation of XAI methods (SHAP and LIME). The methodology includes the training and evaluation of FER models on the FER 2013 and RAF-DB datasets.

4. Results:
   The results section presents the findings, including the performance metrics of the FER models and the interpretability assessments using SHAP and LIME.

5. Discussion:
   This section interprets the results, comparing the effectiveness of SHAP and LIME in enhancing the interpretability of FER models.

6. Conclusion and Outlook:
   The conclusion summarizes the key findings of the research, emphasizing the contributions of SHAP and LIME to the interpretability of FER models. The outlook section outlines future research directions.

2 Literature Review

2.1 Current state of research

2.1.1 Facial Emotion Recognition

Facial Emotion Recognition is a technology that involves the identification and categorization of human emotions based on facial expressions. This process employs Computer Vision and Artificial Intelligence to analyze facial features and patterns in order to derive emotions such as happiness, sadness, anger or surprise. The primary objective of FER is to enable machines to interpret and respond to human emotional states, thereby facilitating human-computer interaction and various applications in fields such as healthcare, marketing, and security. (Khaireddin and Z. L. Chen 2021)
The development of FER can be traced back to early psychological studies of facial expressions by researchers like Charles Darwin and Paul Ekman. (Perrett 2022; Paul Ekman and Friesen 1971) Due to the digitalization and the advancements in ML and Computer Vision the accuracy and robustness of FER systems have improved significantly.
However accurately recognizing facial emotions is still a difficult task even for humans. There are many factors that can potentially influence the decision. The human face consists of over 40 different muscles, all of which contribute to the different emotions displayed on the face.

2.1.2 Basic Emotion Theory

Paul Ekman´s Basic Emotion Theory proposes that there are a set of universal emotions that are biologically hardwired into humans and expressed similarly across all human cultures. (Paul Ekman and Friesen 1971)
He identified six primary emotions:

1. Happiness
2. Sadness
3. Fear
4. Disgust
5. Anger
6. Surprise

These emotions are considered to be basic emotions as they are thought to be universally recognized and have distinct facial expressions associated with them. (Paul Ekman and Friesen 1971)

The majority of the research in the field of FER is based on the Basic Emotion Theory and therefore the published FER datasets are also often annotated using the six basic emotions and neutral.
Nevertheless, contemporary psychological research challenges the Basic Emotion Theory. Many psychologists now believe that facial expressions cannot be classified into a single emotion but can also be a combination of multiple emotions, such as 'blissfully surprised'. Even Paul Ekman himself modified his perspective on the Basic Emotion Theory in recent studies. He introduces a new concept of 15 basic emotions that differ from one another by several characteristics, as illustrated in Figure 1. Those characteristics are shared by the

Table 3.1 Characteristics which distinguish basic emotions from one another and from other affective phenomena

1. Distinctive universal signals
2. Distinctive physiology
3. Automatic appraisal, tuned to:
4. Distinctive universals in antecedent events
5. Distinctive appearance developmentally
6. Presence in other primates
7. Quick onset
8. Brief duration
9. Unbidden occurrence
10. Distinctive thoughts, memories images
11. Distinctive subjective experience

Figure 1: Cutout from "Basic Emotion" by Paul Ekman (Paul Ekman 2013)

15 proposed emotions which are the following: amusement, anger, contempt, contentment, disgust, embarrassment, excitement, fear, guilt and pride. According to Paul Ekman, a family of related emotions can be derived from these 15 basic emotions. However, the psychologist notes that there is currently no evidence to support his findings. (Paul Ekman 2013)
Consequently, research in the field of FER continues to utilise the Basic Emotion Theory. Nevertheless, it is important to bear in mind the critiques surrounding the theory when examining the classification of facial expressions.

2.1.3 Facial Action Units

The human face is composed of over 40 muscles that are both structurally and functionally distinct, which contribute to the facial expressions. Each of these muscles is anatomically distinct and can innervate independently of one another. The specific configuration of facial muscles in a contracted or relaxed state determines the facial expression that is depicted. (Matsumoto and P. Ekman 2008)
Facial Action Units (AU) are defined as the movement of facial muscles or muscle groups that configure the facial expression of an emotion. (Yang et al. 2019) The findings of Paul Ekman also inspired the development of the Facial Action Coding System (FACS). This

system provides a detailed description of the criteria for observing and coding each AU, as well as the combination of the different AUs and the resulting emotion.
The codings system consists of 46 main AUs, 8 head movement AUs and 4 eye movement AUs. (Farnsworth 2022) The presence of one AU or the combination of several AUs determines the emotional classification of the facial expression.

2.1.4 Image Classification

As previously mentioned FER is a synthesis of the disciplines of Psychology and Technology. In this context, the term 'Technology' is used to refer to Computer Vision.
Computer Vision is a broad field that involves enabling machines to interpret and, to some extent, understand the visual world. Key tasks in Computer Vision include:

- Image Classification:
  The process of assigning a label to an entire image.
- Object Detection:
  The process of identifying and locating objects within an image.
- Image Segmentation:
  The process of dividing an image into segments to identify and locate objects within the image.

FER primarily falls under the category of image classification as it involves classifying a facial image into one or more emotional classes. However, it is often the case that FER employs a variety of image processing techniques that can overlap with those used in image segmentation. One such technique is Facial Landmark Detection, which involves the identification of key points on the face, such as the eyes, nose and mouth, that can assist in the accurate classification of emotions.
Image classification can be divided into 4 types of classification (SuperAnnotate 2023):

- Binary classification:
  The process of classifying images into on of two possible categories.
- Multiclass classification:
  The process of classifying images into one of more than two possible categories.
- Multilabel classification:
  The process of assigning multiple labels to an image, where each label represents a different category.
- Hierarchical classification:
  The process of classifying images using a hierarchy of classes, where each class can have subclasses.

Based on these types of classification, FER involves either multiclass classification or multilabel classification. (SuperAnnotate 2023)
The assumption underlying multiclass classification is that each facial image expresses a single emotion at a time. This method involves assigning one predefined emotional class to each image. However, the classification of facial images into a single emotional category may not fully capture the complexity of human emotions. As many psychologists posit, human emotions can be a confluence of several emotions simultaneously.
On the other hand, multilabel classification, allows each facial image to be assigned to mutliple emotional classes. This approach is more realistic as it acknowledges that facial expressions can convey a mixture of emotions. One approach for multilabel classification is the Label Distribution Learning (LDL). The LDL considers the intensity of each emotion, allowing for a nuanced representation of the facial expressions. (Shikai Chen et al. 2020)

2.1.5 Machine Learning Techniques in FER

The process of FER can employ a number of ML techniques to classify a facial expression into an emotional state.
Machine learning tasks can be divided into two categories:

- Unsupervised learning
- Supervised learning

In unsupervised learning, the objective is to identify hidden patterns or intrinsic structures in the input data, which is unlabelled. (Badillo et al. 2020)
In supervised learning, the objective is to make predictions on previously unseen data by learning a mapping from inputs to outputs. In supervised learning, the data used is comprised of input-output tuples, each of which contains an input example and the corresponding output label. (Badillo et al. 2020)
As illustrated in Figure 2, supervised learning can be further divided into regression and classification, depending on the nature of the labels.
Given that FER is image classification, the standard technique for training ML models to recognize human emotions involves supervised learning. As swhon in Figure 2, the conventional ML methods include Decision Trees, Support Vector Machine (SVM), Naive Bayes and K nearest neighbours.
DL is a subset of ML that employs artificial neural networks for complex tasks. DL models are currently the state-of-the-art in this field.
One of the most promising DL architectures is the CNN, which has been successfully implemented in multiple image processing applications (Weitz et al. 2019). As illustrated in Figure 3, the CNN is composed of an input layer, convolutional layers, pooling layers, fully connected layers, and an output layer. It is employed in image detection and classification

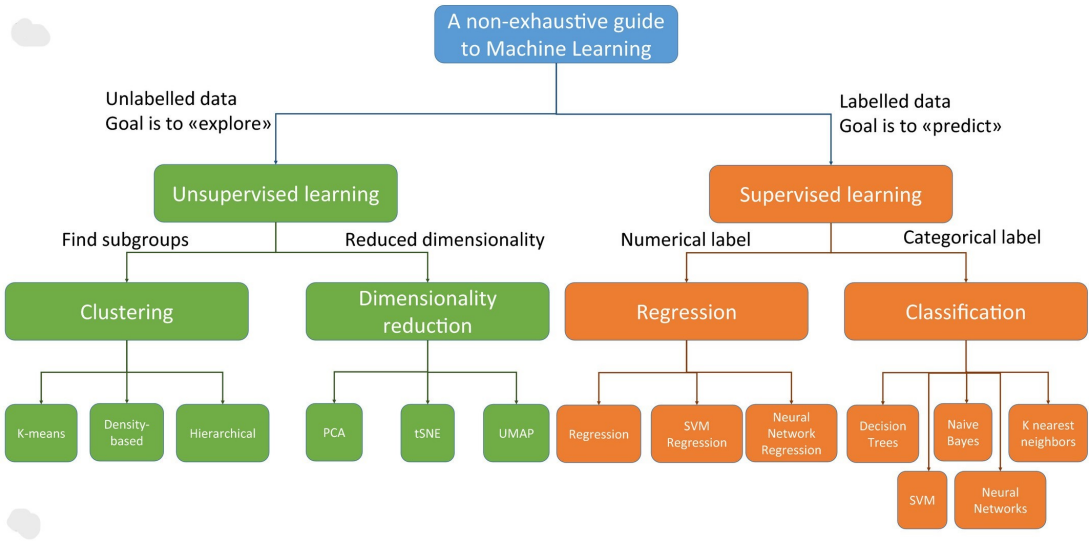

Figure 2: Taxonomy of main ML algorithms (Badillo et al. 2020)

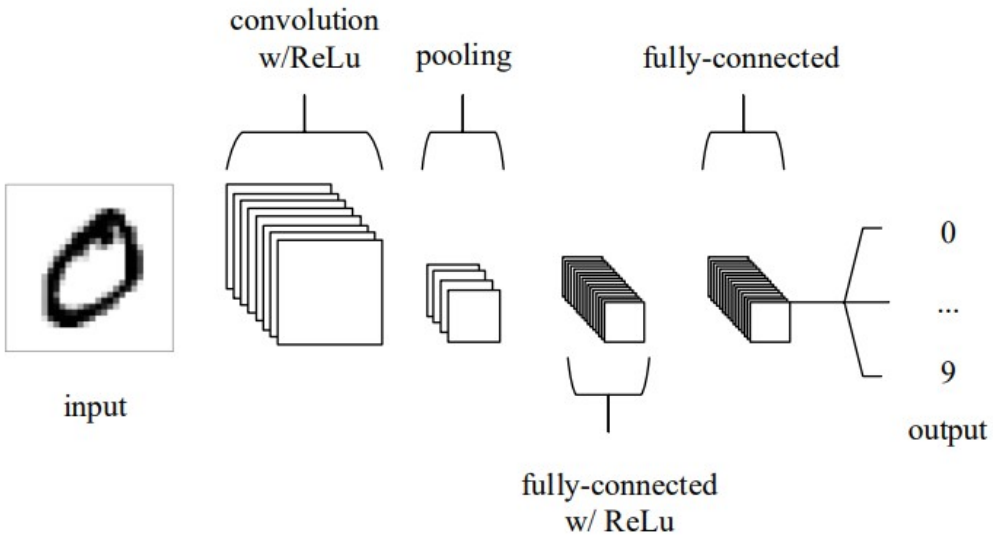

Figure 3: Simple CNN architecture (O'Shea and Nash 2015)

tasks, particularly due to its high accuracy compared to other machine learning algorithms. (O'Shea and Nash 2015)
The key components of the CNN have the following functions (O'Shea and Nash 2015):

- Input layer:
  Takes in the pixel values of the image.
- Convolutional layer:
  Applies filters to the input image to produce feature maps.
- Pooling layer:
  Performs downsampling operations along the spatial dimensions to reduce the

dimensionality of the feature maps.

- Fully connected layer:
  Integrates the features extracted by the previous layers and outputs the final classification scores.

- Output layer:
  Produces the final classification probabilities.

CNNs have seen significant advancement in recent years, including the development of various architectures and algorithms that accelerate the progresses. These advancements have led to the successful application of CNN models in various domains such as image classification and object detection.
Therefore several FER model architecture were developed over the years including AlexNet (Krizhevsky, Sutskever, and Hinton 2012) and ResNet (He et al. 2016). These architectures can be employed and modified for the purpose of training new FER models.
However, images are composed of a multitude of data and possess a complex architectural structure, rendering it challenging to comprehend the entirety of the processing involved in numerous applications. Consequently, in the context of CNNs, research has been concentrated on enhancing their structure with the objective of enhancing image recognition capabilities.
It is therefore crucial to comprehend the underlying decision-making process of the model in order to enhance its performance.

2.1.6 Black Box Models in FER

However, understanding the underlying decision-making process includes understanding the internal logic (Guidotti et al. 2018). DL models are often criticized for their lack of interpretability. Especially in practical applications that include high-stake decision-making scenarios it is crucial to be able to interpret the model's decision (O'Shea and Nash 2015).
Using FER models in the healthcare domain could be a valuable asset, however using the model as an aid for the diagnosis of a patient can only be done, if the decision-making process is comprehensible to the user, otherwise a wrong decision could have a severe negative impact on the patient´s outcome.

Figure 4 presents an overview of the various ML techniques, categorized according to their interpretability and performance. As illustrated in the Figure 4, the higher the performance of the ML method, the lower the interpretability. The highest performance can be achieved by deep neural networks such as the CNN. However, it is notable that these networks exhibit the lowest interpretability. (O'Shea and Nash 2015).
As previously mentioned CNN models have a complex architecture and are therefore often

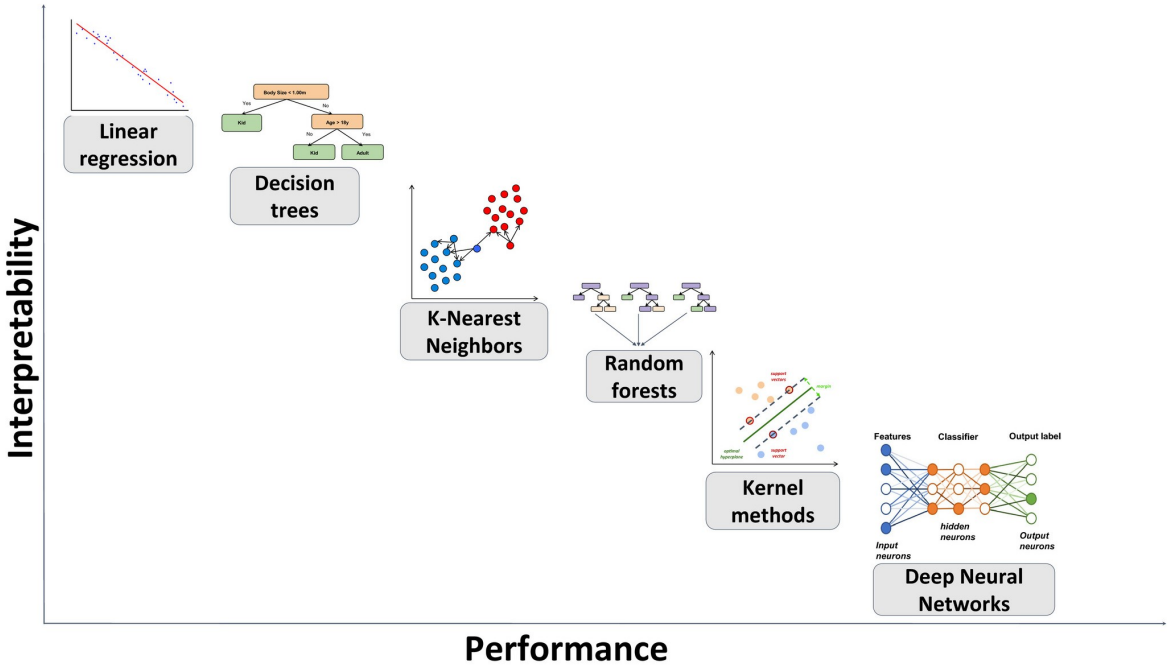

Figure 4: Overview of ML methods based on Interpretability and Performance (O'Shea and Nash 2015)

defined as black box models as they lack interpretability. Black boxes in the context of machine learning refer to models that conceal their internal logic from the user (Guidotti et al. 2018). Therefore, even though these models perform well in classifications tasks, comprehending the rationale behind their predictions remains challenging.
Especially in the case of FER models it is challenging to evaluate the models prediction. The FER models do not provide any explanation on what facial features influenced the models decision-making process for the prediction. This lack of interpretability restricts the practical application and the trustworthiness of the FER models.

2.1.7 Explainable Artificial Intelligence Methods

To improve interpretability and transparency a new concept called XAI was introduced. Explainable AI aims to provide more entail on the decision-making process of an AI-model. Therefore, those techniques try to shed light on black box models.
In order to understand XAI and how XAI methods can help develop an interpretable model, the term to interpret has to be defined. "To interpret means to give or provide the meaning or to explain and present in understandable terms some concept" (Guidotti et al. 2018).
This means that the provided explanations of the interpretation do not need to be explained further, they are self-contained. Therefore, a model that is interpretable provides explanations about its decision-making process.

As shown in Figure 5, XAI methods can be classified into two primary categories:

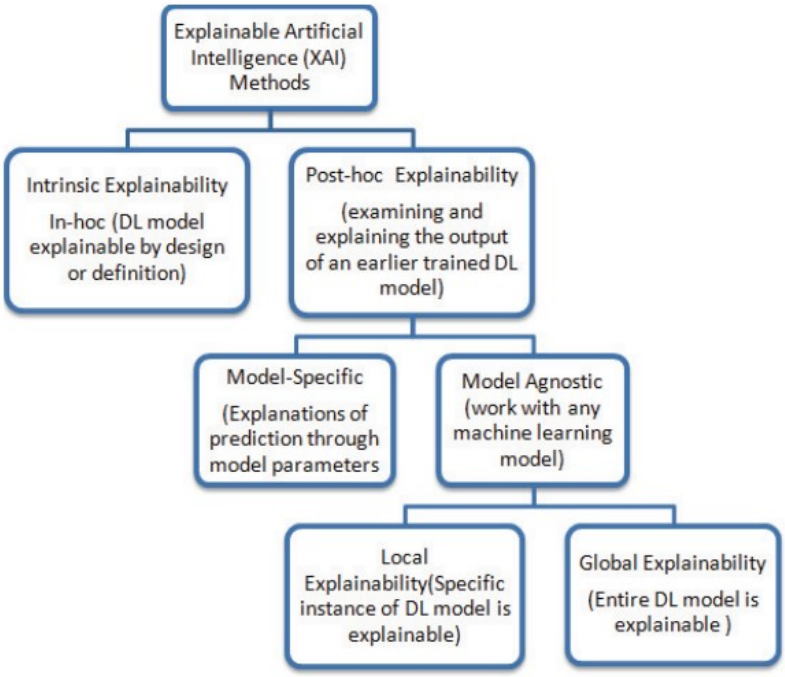

Figure 5: Classification of XAI methods (Joshi and Bagade 2023)

- Intrinsic (In-hoc) Explainability
- Post-hoc Explainability

As in-hoc XAI methods are DL models that are designed to be explainable, they cannot be employed for the evaluation of the interpretability of the FER models (Joshi and Bagade 2023).
Therefore the research focuses on post-hoc XAI methods. Model-specific XAI methods, such as Grad-CAM and Grad-CAM++, utilize the model's parameters to illustrate the rationale behind the predictions.
Model-agnostic XAI methods, such as LIME and SHAP, can be employed for any ML model. Consequently, these methods facilitate an unbiased comparison. The methods provide a uniform approach to interpretability, thereby ensuring that the evaluation is not biased by the specifics of any particular model type.

2.1.8 Shapley Additive Explanations

As previously mentioned SHAP is a model-agnostic XAI method.
The rationale behind the method is derived from the field of corporate game theory. The theory is concerned with the fair distribution of a payout won by a team consisting of a group of people. As each team member contributed differently, dividing the payout by the number of team members might be unfair. Consequently, the Shapley values, which

represent the average contribution of a player to the payout, were calculated. (Roth 1988) The XAI method SHAP makes use of these Shapley values. In the case of FER, the pixel values of the image represent the players, with the FER model serving as the game. The objective is to identify the contributions of each pixel to the prediction by calculating the Shapley values. (S. M. Lundberg and Lee 2017)

$$\phi_i(f, x) = \sum_{z' \subseteq x'} \frac{|z'|!(M - |z'| - 1)!}{M!} [f_x(z') - f_x(z' \setminus i)] \tag{1}$$

- $\phi_i(f, x)$: The Shapley value for a feature i in the context of the black box model f and the input datapoint x.
- $\sum_{z' \subseteq x'}$: The sum is taken over all possible subsets z' of the set x'. Here, x' is a simplified data input.
- $\frac{|z'|!(M-|z'|-1)!}{M!}$: This expression is a weighting factor that considers the number of permutations in which the subset z' can appear in a specific order. Here, $|z'|$ is the number of elements in the subset z', and M is the total number of features in the set x.
- $f_x(z')$: The output of the black box model f for the subset z' of x.
- $f_x(z' \setminus i)$: The output of the black box model f for the subset z' of x, excluding the element i.
- $z' \setminus i$: The subset z' without the element i. This represents the contribution of i when i is removed from the subset z'.

Therefore, the equation 1 sums the weighted contributions $[f_x(z') - f_x(z' \setminus i)]$, i.e. marginal contribution, for all possible subsets z' of x' to compute the Shapley value ϕ_i.
In image classification, SHAP helps in understanding the importance of each pixel or region of an image towards the final prediction. This is especially useful for debugging models, validating their robustness, and ensuring that they are not relying on spurious patterns.

2.1.9 Local Interpretable Model-Agnostic Explanations

LIME is another model-agnostic XAI method, as the name suggests. The goal of the XAI method is to provide a locally faithful explanation of the predictions. (Marco Túlio Ribeiro, Singh, and Guestrin 2016)
This approximation can be obtained using the following equation:

$$\xi(x) = \arg\min_{g \in G} \mathcal{L}(f, g, \pi_x) + \Omega(g) \tag{2}$$

- $\xi(x)$: This represents the explanation of the model for the instance x.
- $\arg\min_{g \in G}$: g is chosen from the set G such that the expression $\mathcal{L}(f, g, \pi_x) + \Omega(g)$ is minimized.
- $\mathcal{L}(f, g, \pi_x)$: This term represents the locality-aware loss function, which measures how unfaithful g is in approximating f in the locality defined by π_x.
- f: The black-box model being explained, mapping from the feature space R^d to R.
- g: The interpretable model used to approximate f locally.
- π_x: A proximity measure between an instance z and x, used to define locality around x.
- $\Omega(g)$: A complexity measure of the explanation g, ensuring that the explanation remains interpretable by humans.

The equation 2 aims to find an interpretable model g that minimizes the locality-aware loss function $\mathcal{L}(f, g, \pi_x)$ and the complexity measure $\Omega(g)$. This ensures that g is both a good local approximation of f and simple enough for humans to understand. (Marco Túlio Ribeiro, Singh, and Guestrin 2016)

$$\mathcal{L}(f, g, \pi_x) = \sum_{z, z' \in Z} \pi_x(z) \left(f(z) - g(z')\right)^2 \tag{3}$$

- $\mathcal{L}(f, g, \pi_x)$: The same locality-aware loss function from Equation 2.
- $\sum_{z, z' \in Z}$: Summation over the set Z of perturbed instances.
- $\pi_x(z)$: The proximity measure used to weight the loss based on how close z is to x.
- $f(z)$: The prediction of the black-box model f for instance z.
- $g(z')$: The prediction of the interpretable model g for instance z', where z' is a perturbed version of z.

Equation 3 calculates the locality-aware loss by summing the weighted squared differences betwenn the black box model´s predictions $f(z)$ and the interpretable model´s predictions $g(z')$ over perturbed instances. The proximity measure $\pi_x(z)$ ensures that instances closer to x are weighted more heavily. (Marco Túlio Ribeiro, Singh, and Guestrin 2016)
The LIME algorithm operates by modifying the input data around a given instance and observing the resulting changes in the model´s predictions. In the context of image classification, this entails the segmentation of the image into interpretable components, i.e. superpixels. These segments are perturbed, for instance by masking them out, to see how

the model´s prediction is affected. The interpretable model g, often a simple linear model, is then trained on these perturbed segments to approximate the behavior of the complex model f locally around the instance x. (Marco Túlio Ribeiro, Singh, and Guestrin 2016) Therefore, LIME helps to identify the most influential regions within an image for a given prediction, thereby offering insights into the underlying decision-making process of the model. This can be of particular value in identifying potential biases, validating model reliability, and ensuring transparency in applications such as medical image analysis, where interpretability is critical.

2.1.10 Ethical Considerations in FER

The field of FER is concerned with the analysis of images that depict humans. Consequently, ethical considerations for FER involve a number of crucial elements, including privacy, accuracy, potential biases and the reliability of the technology.
FER systems often involve capturing and analyzing facial data, which can lead to significant privacy issues. It is crucial to ensure that the facial data of an individual is not misused and that their privacy is respected. There is also a necessity to prevent the utilization of FER for discriminatory purposes, such as the identification of individuals based on race or ethnicity.
Furthermore, the accuracy of FER systems is a major concern, as misclassification of emotions can have serious implications. A number of studies have demonstrated that the process of detecting emotions from static images is often inconsistent, particularly when large and heterogeneous samples of observers are involved. (Cabitza, Campagner, and Mattioli 2022)
In addition, it is possible for FER systems to be biased based on the data they are trained on. If the training data lacks diversity, the system may perform poorly on certain demographic groups. This can result in unfair treatment and decisions based on inaccurate emotion recognition. (Cabitza, Campagner, and Mattioli 2022)
The deployment of FER technology in various sectors, such as surveillance and marketing, raises also ethical questions about its impact on individuals' autonomy and consent. For instance, the utilization of FER technology to monitor emotions in public spaces or workplaces can be seen as an invasion of privacy and can lead to these ethical dilemmas.
In conclusion, while the potential of FER technology to enhance human-computer interaction and various applications is considerable, it is imperative to address the ethical concerns related to privacy, accuracy, biases, and the broader implications of its usage to ensure that it is developed and deployed responsibly.

2.2 Research gap and research question

Even though Artificial Intelligence and Explainable AI have been studied a lot in recent years, there is still a gap between high accuracy and interpretability in AI-models. The lack of interpretability makes it hard to understand AI-models completely.
The first step to improve AI systems is to understand their weaknesses (Rosenfeld 2021). However, it is more difficult to perform weakness analysis on black box models and therefore it is necessary to combine AI-models with XAI methods.
In the area of FER, the adaption and effectiveness of XAI methods is relatively unexplored. The lack of interpretability of FER models raises ethical concerns, limiting the practical application of these models in sensitive domains such as healthcare and minimizes user trust.
When talking about trusting a model, there are two definitions of trust: trusting a prediction and trusting a model. In the case of trusting a prediction the question is whether the user trusts an individual prediction made by the model and based on this prediction would the user take some action. Trusting a model refers to the question whether the user trusts the model to behave in reasonable ways when used (Marco Túlio Ribeiro, Singh, and Guestrin 2016).
The proposed master thesis aims to answer the following research question:

RQ: "How can XAI methods improve the interpretability of Facial Emotion Recognition models, enabling a clearer understanding of the underlying features and decision-making processes?"

This research question emphasizes two core aspects:

- FER Model Development
- Evaluation of Explainable AI Techniques

These aspects will be explained in detail in the following sections.

3 Methodology

3.1 Type of Research

This thesis employs an applied research approach within the domain of AI and ML, with a specific focus on enhancing the interpretability of FER models through the use of XAI methods. The research integrates theoretical concepts of XAI with practical applications in order to address real-world challenges in understanding and trusting FER systems.
This research follows a quantitative and comparative design, involving the following key steps:

- Model Training and Evaluation:
  Training a FER model on standard datasets (FER 2013 and RAF-DB) and evaluating its performance in terms of accuracy, precision, recall, and F1-score.
- Application of XAI Methods:
  Applying SHAP and LIME to the trained FER model to generate explanations for its predictions.
- Metric-Based Assessment:
  Evaluating the interpretability of the model´s explanations using predefined metrics such as stability, simplicity, comprehensiveness, consistency with domain knowledge, computational efficiency, and visual interpretability.
- Comparative Analysis:
  Comparing the results of SHAP and LIME to determine their respective strengths and weaknesses in enhancing model interpretability.

3.2 Overview of the proposed FER Model

3.2.1 Technical Environment

The implementation and evaluation of the FER models and XAI methods were carried out on the HNU server. The details of the technical environment are the following:

- Operating System: Linux
- Distribution: Ubuntu 20.04.6 LTS (Long Term Support)
- NVIDIA Data Science Stack
- GPUs: 4 NVIDIA A100 80GB PCIe GPUs

As previously mentioned the server was equipped with the NVIDIA data science stack, which includes optimized libraries and tools for data science and ML workflows. The four

NVIDIA GPUs provide a substantial computational power for training DL models.
The code was developed in python, using Jupyter Notebooks as the development environment, offering an interactive interface for coding, debugging and visualizing the results.
Several python libraries were used in the implementation of the FER models and XAI methods. The key libraries included:

- Numpy: For handling arrays and operations.
- Pandas: For data manipulation and analysis.
- Sklearn: For data splitting and basic ML utilities.
- Keras: For building and training the model.
- Tensorflow: As the backend for keras.
- Matplotlib: For plotting and visualizing data and results.
- Skimage: For advanced image processing tasks.
- Shap: For implementing SHAP values.
- Lime: For implementing LIME.

The setup provided a robust and scalable environment for implementing and evaluating the FER models and XAI methods, leveraging the computational power of the NVIDIA A100 GPUs and the comprehensive support provided by the NVIDIA data science stack. This environment ensured efficient model training, testing, and validation processes, crucial for developing high-performance and interpretable FER models.

3.2.2 Data collection

In the context of this master thesis on FER, the decision was made to utilize existing datasets rather than collecting new data. This choice was driven by several reasons:

- Larger Dataset Availability:
  Existing datasets provide access to significantly larger volumes of data than would be feasible to collect independently within the scope of this research. Large datasets are crucial for training robust and accurate models and implementing XAI methods.
- Ensuring High Variety in Demographics:
  Established datasets often contain a wide range of demographic diversity, including variations in age, gender and ethnicity. This diversity is essential for developing an unbiased and effective FER model.

- Time Constraints:
  Data collection is a time-intensive process that involves recruiting participants, capturing data and annotating the facial expressions. Given the limited time frame, relying on existing datasets allows for more efficient use of time, focusing the efforts on the model development and XAI implementation and analysis.

- Annotation Reliability:
  High-quality annotations are crucial for training accurate FER models. However, ensuring reliable annotations is challenging without a large team of experts and sufficient time. Existing datasets often come with professionally annotated data, ensuring a higher reliability and consistency.

- Accessibility to Large Amounts of Data:
  Gathering a large amount of high-quality data independently is challenging. Existing datasets mitigate this issue by providing accessible data that can be immediately utilized for research purposes.

The research area of FER contains a multitude of datasets that have been developed and made available over the years. In the process of sourcing data for the FER model, several factors were taken into consideration to ensure the selection of the most appropriate datasets. These factors include:

- Size:
  The dataset needed to be large enough to provide a robust foundation for training and testing the model, ensuring statistical significance in the results.

- Balance and Diversity:
  It was essential to choose datasets that encompassed a diverse range of expressions, demographics and ethnicity to mitigate biases.

- Real-World Applicability:
  The datasets were evaluated for their relevance to real-world scenarios, ensuring that the emotional expressions captured were realistic and applicable to real-world applications.

- Quality of Annotations:
  High-quality and accurate annotations were crucial to ensure that the data could be reliably used for training and evaluating the model.

- Accessibility:
  The datasets needed to be publicly accessible in terms of availability and ease of use.

After a thorough evaluation based on these criteria, the following datasets were selected for further considerations:

- FER 2013
- AffectNet
- Real-world Affective Faces Database Real-world Affecitve Faces Database (RAF-DB)
- Cohn-Kanade Plus Cohn-Kanade Plus (CK+)
- Static Facial Expressions in the Wild Static Facial Expressions in the Wild (SFEW)

These datasets collectively offer a comprehensive and diverse collection of facial expressions. The FER 2013 dataset consists of 35,887 gray-scaled images that offer a wide range of expressions. As the dataset is considered as a benchmark in the FER community it is publicly available and easy to access. However due to the controlled environments and the gray-scaled images the dataset provides a less realistic point of view. The labels of the FER 2013 dataset are based on the basic emotions and the annotations are relatively consistent but lack detailed annotations. (Kumar and Reddy n.d.)
AffectNet contains over one million facial images collected from the internet that capture real-world variability with various poses, lighting and occlusions. The dataset is therefore diverse with 7 primary and 11 secondary emotions.(Mollahosseini, Hassani, and Mahoor 2017) Initially the images were annotated by single annotators but the label quality improved when the images were re-annotated by crowd-sourcing. (Kim and Wallraven 2021) However the dataset requires licensing and is therefore less accessible than other datasets.
The RAF-DB dataset consists of 29,672 facial images that offer a wide variety of facial expressions from diverse demographics ensuring realistic conditions. (Bobojanov et al. 2023) The images were annotated through crowd-sourcing with 40 different taggers that led to reliable and varied labels. (Bobojanov et al. 2023) RAF-DB is also publicly available and easy to use.
CK+ contains 593 video sequences from 123 subjects. However the dataset is limited in its diversity and less applicable to real-world scenarios due to the controlled setting. (Wadhawan and Gandhi 2021) The dataset consists of high-quality annotations with detailed facial action units and emotion labels and is publicly available and highly accessible.
The SFEW dataset consists of 1,766 images that are publicly available. The images were retrieved from static frames of the AFEW database by computing key frames based on facial point clustering. (Shan Li and Weihong Deng 2018) The dataset offers a high diversity and is highly applicable to real-world scenarios as natural and varied expressions were captured. SFEW provides basic emotion labels but can be inconsistent due to the natural collection of images.

As previously mentioned the accessibility of the AffectNet dataset is limited and due to

its extensive size it can be challenging to handle, process and use this dataset.
The CK+ contains high-quality images that were captured in a controlled environment derived from both static and dynamic sequences and is therefore ideal for controlled studies. However the dataset is less applicable to real-world scenarios.
SFEW is highly applicable to real-world scenarios due to the natural and varied expressions captured in the images collected from movies. However the dataset is smaller than other FER datasets and even though it provides basic emotion labels, they can be inconsistent due to the natural collection of the images.
Because of the above mentioned points the three datasets were not chosen for the research.

Based on the 5 selection criteria RAF-DB and FER 2013 are the ideal datasets for the evaluation of the model and the XAI methods.
FER 2013 is larger than many other FER datasets and can be used to train robust models as it offers images with variations in lighting, occlusions and facial orientation, reflecting real-world conditions. The variability enhances the model´s ability to generalize well to real-world scenarios. (Sahoo et al. 2022) As the FER 2013 is widely recognized as a benchmark dataset, the designed model can be compared against a large body of existing work, facilitating an evaluation and validation of new models and techniques. (Khaireddin and Z. L. Chen 2021) As well as that the various previous pre-processing and augmentation techniques can improve the model´s robustness and performance. Those techniques are well-documented and can be leveraged by new researchers to enhance their models.
As mentioned before the RAF-DB contains images that are collected from the internet and therefore they are reflecting a wide variety of real-world scenarios. Due to the diversity and naturalness of the images the dataset has a high relevance to real-world applications. Furthermore, the images were specifically annotated for pose and occlusion attributes which is valuable for developing robust models and therefore it enhances the ability of models to handle challenging conditions such as varying poses and occlusions. (Ma, Sun, and Shutao Li 2023)

The FER 2013 was created using a collection of images from the Google image search engine. The images were gathered by querying for specific faces that matched a set of 184 emotion-related keywords such as "blissful" or "enraged". The keywords were then combined with additional terms related to gender, age and ethnicity. The OpenCV face recognition algorithm was employed to identify and delineate bounding boxes around each face within the collected image. The labeling process involved human annotators that categorized each image into on predefined emotion category based on the seven basic emotions. (Goodfellow et al. 2013)
The images contained in the RAF-DB dataset were collected from the internet, representing thousands of individuals. The images were sourced from a variety of online platforms in

order to ensure a wide range of expressions and demographics. Each image was labeled by approximately 40 human annotators using a crowd-sourcing platform which ensures reliability and validity of the emotion labels. The Expectation-Maximization Expectation-Maximization (EM) algorithm was then used to filter out the unreliable labels in order to identify and correct the mislabeled images ensuring high-quality annotations.(Shan Li, Weihong Deng, and Du 2017)

3.2.3 Data Pre-Processing

For the evaluation of XAI to improve the interpretablity of FER, the FER 2013 and RAF-DB datasets underwent several pre-processing steps. Data pre-processing is a critical step in the development of ML models, particularly in the context of FER. Effective pre-processing ensures that the data fed into the model is clean, standardized, and suitable for training, which ultimately enhances the model´s performance and interpretability.

The FER 2013 dataset was obtained from Kaggle. The dataset was provided in a CSV file containing both the pixel values of the images and the corresponding emotion labels. This CSV file was uploaded to the Jupyter Notebook environment for further processing steps. The data was read from the CSV file using pandas and the emotion labels and pixel values were extracted into two separate numpy arrays. As shown in Figure 6 the pixel values,

```
# Creating empty numpy array for images
X = np.zeros((pixels.shape[0], 48*48))
print(X.shape)

# Filling X with pixel values for every image
for ix in range(X.shape[0]):
    p = pixels[ix].split(' ')
    for iy in range(X.shape[1]):
        X[ix, iy] = int(p[iy])
```

Figure 6: Re-shaping pixel values for each image in the dataset

initially stored as strings, were converted into arrays of integers. Each image´s pixel values were re-shaped into a 48x48 pixel grayscaled image. The defined numpy array 'X' was filled with these reshaped pixel values for each image in the dataset by iterating over the rows and columns of 'X'.
The dataset was then split into three subsets: 70% for training, 15% for testing and 15% for validation. This splitting ensured that the model had enough data to learn from while also being able to generalize unseen data. As shown in Figure 7, the splitting was performed

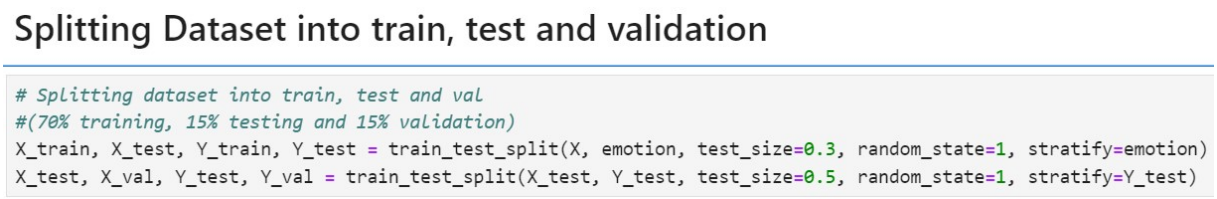

Figure 7: Splitting dataset into train, test and validation using sklearn train_test_split function

using sklearn´s 'train_test_split' function, ensuring a stratified split to maintain the proportion of the different emotional classes in each subset.
The input data 'X' was then re-shaped to include the channel dimension required by the standard CNN model. Because the images are grayscaled, the channel dimension was set to 1. Therefore the final shape of the input data was: (number of images, 48, 48, 1).
The emotion labels were converted into one-hot encoded vectors using 'to_categorical' from keras.utils, resulting in a shape of (number of images, number of emotion classes) for the labels.
In the last step of the data pre-processing, the pixel values were normalized by dividing 'X_train', 'X_test' and 'X_val' by 255, scaling them to a range between 0 and 1.

The RAF-DB dataset was obtained from the official website of the dataset. (H. Wang and W. Deng 2024) The received dataset consists of two subsets: the single-label subset which contains 7 classes of basic emotions such as happiness, sadness or angry and the two-tab subset which contains 12 classes of compound emotions such as happily surprised or sadly fearful. For the training and building of the FER model, the 15,339 aligned images of the single-label subset were used.
The RAF-DB dataset underwent a series of pre-processing steps. While some of these were similar to those used for the FER 2013 dataset, there were also specific steps unique to RAF-DB. As with the FER 2013 dataset, the images and labels for RAF-DB were uploaded to the Jupyter Notebook environment. However, the emotion labels for RAF-DB were stored in a separate text file called 'list_partition_label.txt'.
The text file contained the filenames and corresponding emotion labels and was read using pandas to create a data frame with image filenames and their associated labels. As shown in Figure 8 an empty list 'label_elements' was created and the code iterated over each line of the text file containing the labels. For each line the code split the line at the substring '.jpg', creating a list where the second element, i.e. the emotion label of the image, is accessed. The label was then converted to an integer and appended to the 'label_elements' list. The elements were then subtracted by 1 to convert the labels into zero-based indexing, which is standard python indexing. The images were extracted from a directory containing the names of the image files and stored in a list.

```
# Extracting Labels from list_partition_label

with open(data_path_labels + '/list_patition_label.txt') as f:
    label_lines=f.readlines()

label_lines.sort()
print(label_lines[0:5])

label_elements=[]

for i in label_lines:
    label_elements.append(int(i.split(".jpg")[1][1]))

print(label_elements[0:5])

#Adapting to python index (-1)
labels = [element - 1 for element in label_elements]
print(labels[0:5])
```

Figure 8: Extraction emotion labels for RAF-DB from 'list_partition_label.txt'

The splitting of the dataset into train, test and validation subsets as well as the data categorization and normalization were done using the same methods as for the FER 2013 dataset. Therefore providing: 3 data subsets with 70% for training, 15% for testing and 15% for validation, one-hot encoded labels and pixel values scaled to a range between 0 and 1.
These pre-processing steps ensured that the datasets were appropriately prepared for training a CNN model.

3.2.4 Model architecture

The chosen architecture of the FER models using the FER 2013 and RAF-DB datasets is a CNN designed to effectively handle the complexities of facial expression data, leveraging multiple convolutional layers to extract features and dense layers for the classification. The architecture is based on the AlexNet model (Krizhevsky, Sutskever, and Hinton 2012), which first introduced stacking CNN layers directly and the proposed model in the paper "Face Value: On the Impact of Annotation (In-)Consistencies and Label Ambiguity in Facial Data on Emotion Recognition". (Gebele, Brune, and Faußer 2022)
As shown in Figure 9 the model was build using the keras Sequential API, which allows for a layer by layer addition of various neural network components.
The architecture is divided into four input layers and one flatten layer. The input layers consist of two convolutional layers, which are followed by one max pooling layer. The utilization of 'same' padding ensures that the output feature maps have the same spatial

```
# Building FER model
model = Sequential()

model.add(Conv2D(32, (5, 5), padding="same", activation='relu', input_shape=X_train.shape[1:4]))
model.add(Conv2D(32, (5, 5), padding="same", activation='relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))

model.add(Conv2D(64, (3, 3), padding="same", activation='relu'))
model.add(Conv2D(64, (3, 3), padding="same", activation='relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))

model.add(Conv2D(128, (3, 3), padding="same", activation='relu'))
model.add(Conv2D(128, (3, 3), padding="same", activation='relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))

model.add(Conv2D(256, (3, 3), padding="same", activation='relu'))
model.add(Conv2D(256, (3, 3), padding="same", activation='relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))

model.add(Flatten())
model.add(Dense(128, activation='relu'))
model.add(Dropout(0.4))

model.add(Dense(64, activation='relu'))
model.add(Dropout(0.4))
model.add(Dense(7 , activation='softmax'))
```

Figure 9: FER model architecture (FER 2013 dataset)

dimensions as the input, achieved through the padding of the input. This is beneficial as the multiple layers could otherwise result in a significant reduction in the spatial dimensions of the feature maps. Furthermore, it simplifies the architectural design by eliminating the necessity for manual calculation of layer dimensions following each convolution operation. The Rectified Linear Unit (ReLU) is the most commonly used activation function in CNNS due to its effectiveness. ReLU allows the CNN to learn complex patterns and is computationally inexpensive. Furthermore the activation function reduces the likelihood of the vanishing gradient problem and reduces the risk of overfitting as it outputs zeros for any negative input. (Y. Wang et al. 2020)

Max pooling is a commonly used technique in CNNs due to several benefits that enhance the performance and efficiency of these models. Max pooling reduces the size of feature maps in a neural network by taking the highest value in each small region. Therefore pooling operation helps in reducing the number of parameters in the CNN, thereby speeding up the training process and inference times. (Rafiq et al. 2022)

By downsampling the feature maps, max pooling acts as a form of regularization and thereby reduces the risk of overfitting. This is particularly useful in tasks where the training data may be limited. (Rafiq et al. 2022)

Because of that, the max pooling operation is preferred over the average pooling function

as it increases the performance of the CNN significantly. (Rafiq et al. 2022)

As shown in Figure 9, the last block of the architecture consists of flattening, two dense layers and two dropouts. The flatten layer transforms the two dimensional matrix of the features into a one dimensional vector, which is necessary for the following dense layers. (Wu, Xie, and J.-H. Luo 2016)
The dense layers perform the final classification by combining features from the convolutional layers. Using Dropouts prevents the model from overfitting.
The last layer of the model is a softmax layer with seven elements corresponding to the seven emotional classes. The choice of softmax for the output in the CNN was driven by several reasons such as probability distribution, ensuring mutually exclusive classes and training stability. Softmax converts the output values into probabilities, making it easier to interpret the outputs and ensures that the sum of the output probabilities is 1. Furthermore the function helps in stabilizing the training process by providing smooth gradients. (Krizhevsky, Sutskever, and Hinton 2012)
The architecture is designed to capture spatial hierarchies in images through convolutional layers, pooling layers to reduce dimensionality, and dense layers for classification. This approach effectively balances model complexity and performance. (Gebele, Brune, and Faußer 2022)
The model was compiled using the Adam optimizer with a learning rate of 0.0001. The loss function used was categorical cross-entropy, suitable for multi-class classification problems, and the primary metric for the evaluation was the validation accuracy. (Gebele, Brune, and Faußer 2022)
The training process included several callbacks to enhance the performance and manage overfitting. As shown in Figure 10, the keras ModelCheckpoint callback was used to save the best model based on the validation accuracy for the model evaluation. Furthermore the ReduceLROnPlateau callback was used to reduce the learning rate when a plateau in the validation accuracy is detected. If there was no positive change in the validation accuracy for three epochs in a row the learning rate was automatically minimized by 0.1. The process stopped if the learning rate reached the defined minimum.
The training and validation data were fed into the model using data generators, ensuring efficient memory usage and on-the-fly data augmentation. As shown in Figure 11 the parameters 'featurewise_center', 'featurewise_std_normalization', 'samplewise_center', 'samplewise_std_normalization', 'horizontal_flip' and 'vertical_flip' were all set to 'False'. The parameters 'featurewise_center' and 'samplewise_center' control whether to center the input data by subtracting the mean value. By keeping those parameters 'False' the original pixel intensity distribution is not altered and therefore the natural variability in the facial expressions is maintained. (Chollet 2016)
To control whether to divide the input data by the standard deviation, the parameters

```
# Defining callbacks

model_callbacks = [
    # Only saving the best model based on validation accuracy
    tf.keras.callbacks.ModelCheckpoint(filepath='Model/model_evaluation_xai_FER.hdf5',
                                       save_best_only=True, monitor='val_accuracy',
                                       mode='max', verbose=1),
    # Defining conditions for stopping the training and reducing the learning rate
    ReduceLROnPlateau(monitor='val_accuracy',
                      patience=3,
                      factor=0.1,
                      min_lr=0.000000001,
                      min_delta=0.0001,
                      verbose=1)
]
```

Figure 10: Definition of callbacks

```
# Building and Fitting ImageDataGenerator

datagenerator_train = ImageDataGenerator(
        featurewise_center=False,
        featurewise_std_normalization=False,
        samplewise_center=False,
        samplewise_std_normalization=False,
        horizontal_flip=False,
        vertical_flip=False)

datagenerator_val = ImageDataGenerator()

datagenerator_train.fit(X_train)
```

Figure 11: Definition of keras ImageDataGenerator

'featurewise_std_normalization' and 'samplewise_std_normalization' were set. Similar to the centering parameters, not normalizing the standard deviation preserved the characteristics of the data. This can be crucial for recognizing slight changes in the facial expressions. (Chollet 2016)
The parameters 'horizontal_flip' and 'vertical_flip' define whether to randomly flip the input data horizontally or vertically. In facial recognition tasks, flipping might lead to unrealistic variations as human faces are not typically symmetrical. (Chollet 2016)
Even though no augmentation or pre-processing was done using the keras ImageDataGenerator, it still provides a consistent data handling interface and can handle batch processing efficiently. The data generator was also defined to make it easier to add augmentations and pre-processing steps in the future.

The model was trained for 50 epochs with a batch size of 128 and steps per epoch were calculated as the number of training samples divided by the batch size. This definition ensured an equal amount of weight updates for both models. (Gebele, Brune, and Faußer 2022)

As previously mentioned the architecture was chosen based on its ability to balance the model complexity and performance. Alternative architectures, such as deeper networks like ResNet or more sophisticated techniques like attention mechanisms, were considered. However, these were deemed unnecessary due to the relatively small size of the two datasets and the added computational complexity.
The chosen architecture balances complexity and performance, ensuring the model is capable of learning rich feature representations without overfitting. The use of convolutional layers for feature extraction and dense layers for classification has proven effective in similar tasks and provides a strong foundation for FER. (Gebele, Brune, and Faußer 2022)

3.3 XAI methods implemented

As previously stated, XAI methods are of critical importance for comprehending and interpreting the decisions made by ML models. For FER models, a variety of XAI methods can be employed, each with its own respective strengths and weaknesses.
Therefore the following XAI methods were considered (Marco Túlio Ribeiro, Singh, and Guestrin 2016; Roth 1988; K. Li et al. 2018; Selvaraju et al. 2016):

- LIME
- SHAP
- Gradient-weighted Class Activation Mapping (Grad-CAM)
- Guided Gradient-weighted Class Activation Mapping (Guided Grad-CAM)
- Generative Adversarial Interpretations (GAIN)

LIME is a model-agnostic, local explanation method that approximates the model locally around the prediction to explain individual predictions. The input data is perturbed, and the resulting changes in the output are observed to build a simpler, more interpretable model, such as a linear model, that mimics the complex model´s behavior in that local region. This method is beneficial for understanding the nuances of individual predictions and is useful for debugging and identifying the factors influencing a particular decision. (Marco Túlio Ribeiro, Singh, and Guestrin 2016)
The XAI method SHAP is also a model-agnostic, local explanation method where Shapley

values are derived from cooperative game theory, providing a unified measure of feature importance by considering all possible combinations of features. The method attributes each feature´s contribution to the prediction by computing Shapley values, ensuring consistency and fairness. This approach is effective for both global and local explanations, offering comprehensive insights into feature importance and interactions. (Roth 1988)
In contrast, Grad-CAM represents a model-specific method for visual explanations. The model employs gradients of the target concept, such as the predicted emotion, flowing into the final convolutional layer to generate a coarse localization map. The map thus highlights the crucial regions in the image that are important for the prediction. Grad-CAM is particularly well-suited to image-related tasks, such as identifying which facial features contribute to the prediction of emotional states. (Selvaraju et al. 2016)
Guided Grad-CAM is a model-specific method for visual explanations that combines Grad-CAM with guided backpropagation to provide more detailed and visually interpretable maps. While Grad-CAM identifies the regions of importance, guided backpropagation offers high-resolution details for these regions, enhancing the clarity and interpretability of the explanation. This approach offers the most fine-grained visual interpretability in image tasks, providing more detailed insights than Grad-CAM. (Selvaraju et al. 2016)
Finally, GAIN is a model-specific method for visual and generative explanations. The method employs a generative adversarial network to generate interpretable masks that highlight the most relevant regions within the input image. The objective is to generate explanations that are in close alignment with human understanding. The utility of GAIN lies in its capacity to provide explanations that are human-like in nature, particularly in the context of complex image tasks. (K. Li et al. 2018)

The selection of the most appropriate XAI methods is dependent on the specific requirements of the FER models and their explanations. The two principal categories of explanation are:

- Local and individual prediction explanations:
  - LIME:
    Offers simplicity and ease of understanding for the individual predictions.
  - SHAP:
    Offers more comprehensive and theoretically grounded explanations, which are useful for both local and global insights.
- Visual explanations of image data:
  - Grad-CAM:
    Is an effective method for identifying the regions of an image that are most influential in the model's decision-making process.

- Guided Grad-CAM:
  Provides more detailed visual explanations, integrating the advantages of Grad-CAM and guided backpropagation.
- GAIN:
  Offers the advantage of providing human-like and interpretable masks, rendering it an effective tool for complex visual tasks.

Therefore, the choice depends on whether the visual interpretability or a more feature-centric understanding is prioritized. For a thorough evaluation, using a combination of these methods might provide the most comprehensive insights into the FER models interpretability. However, due to time constraints, only the two XAI methods, LIME and SHAP, were selected for implementation.
By using both LIME and SHAP, the strengths of each method can be leveraged to gain a more comprehensive understanding of the FER model. LIME and SHAP can provide distinct insights into the same prediction. While LIME may provide a relatively straightforward approximation for a specific instance, SHAP can offer a more comprehensive and theoretically grounded explanation. This diversity enhances the interpretability analysis.
The fact that both methods are model-agnostic ensures that they can be applied to any type of FER model that is chosen for use or comparison. This flexibility allows the interpretability framework to be adapted to a variety of model types and scenarios.
Furthermore, both XAI methods provide a visual representation of the image data, whereby the most influential regions are highlighted based on the prediction made by the FER model.

3.3.1 Shapley Additive Explanations (SHAP)

To evaluate the SHAP method and the interpretability of the FER model, several variations of the method were implemented. Overall, the SHAP method implementation was similar for both FER models, utilizing the shap library (S. Lundberg 2018).
The first step of the implementation process involved calculating the Shapley values for each pixel of the image, thus providing the individual contribution of each pixel to the prediction. The Figure 12 illustrates the implementation of these Shapley values exemplary for the first image of the FER 2013 test dataset and its model´s prediction. The following steps are included in the implementation process (S. Lundberg 2018; S. M. Lundberg and Lee 2017):

- Selecting a background dataset:
  After loading the model, a background dataset is selected from the training data. As shown in Figure 12, a random sample of 500 instances of the training dataset

```
model = keras.models.load_model('Model/model_evaluation_xai_FER.hdf5')

# Selecting background dataset
background = X_train[np.random.choice(X_train.shape[0], 500, replace=False)]

# Creating Deep Explainer to explain predictions of the model
explainer = shap.DeepExplainer(model, background)

# Iterating through test dataset and calculating shap values
for xi in range(10):
    instance_to_explain = np.expand_dims(X_test[xi], axis=0)

    preds = model.predict(instance_to_explain)
    emotion_value = get_class(preds)
    emotion = int(emotion_value[0][0])

    shap_values = explainer.shap_values(instance_to_explain)
    shap.image_plot(shap_values, instance_to_explain, show = False)
    plt.title(f'Prediction: {emotional_classes[emotion]}')
    plt.show()
```

Figure 12: Implementation of SHAP without masker

was chosen to serve as the background for the SHAP explainer.
The background dataset represents the distribution of the data on which the model was trained on, i.e. 'X_train' and is used by the SHAP explainer to generate reference values for explaining the model´s predictions.
The size of the background dataset should be large enough to capture the variability and distribution of the training data, but small enough to keep the computational cost manageable. Therefore the 500 instances were chosen for the background.

- Creating the DeepExplainer:
  After setting the background, the SHAP DeepExplainer was created using the model and the background dataset. As the name states, the DeepExplainer is suitable for DL models such as the trained FER models.
  The explainer uses the background dataset to approximate the expected value of the predictions without certain features. For the DL models, the DeepExplainer utilizes the gradients of the model´s output with respect to its input features. By using the gradient information the explainer, can provide more accurate explanations compared to methods that treat the model as a black box. (S. Lundberg 2018)

- Iterating through the test dataset:
  As illustrated in Figure 12, the initial ten images of the test dataset are iterated through in order to calculate the Shapley values for each instance. For each instance, the following steps were carried out:

- Instance preparation:
  The current test instance was expanded to align with the input dimensions anticipated by the model.
- Model prediction:
  The previously trained FER model was employed to generate a prediction based on the input data, resulting in an array of probabilities. These predictions were then used to determine the emotional class of the instance, utilising the pre-defined 'get_class' function.
- Calculating Shapley values:
  The Shapley values for the current instance were calculated utilizing the DeepExplainer. The method 'explainer.shap_values()' was employed to calculate the Shapley values for the given instance. The input was an instance of the test dataset to be explained and the output were the Shapley values, calculated using the previously mentioned equation 1. These values were then stored in the 'shap_values' variable.
- Plotting Shapley values:
  The Shapley values were plotted using the 'shap.image_plot' method and the emotional class of the displayed image was set as the title of the plot.

The output of the code snippet illustrated in Figure 12, provides a visual explanation of the FER model´s prediction for the specific instance.
The central part of the plot is the displayed image, which requires an explanation. Overlaid on the image are the Shapley values, which highlight the areas of the image that contributed to the model´s predictions. These contributions are shown with colours:

- Red areas:
  These areas indicate features that have a positive impact on the model´s prediction. Therefore it represents the positive Shapley values of the features that push the prediction towards the predicted emotional class.

- Blue areas:
  These areas indicate features that have a negative impact on the model´s prediction. Therefore it illustrates the negative Shapley values that indicate features that push the prediction away from the predicted emotional class.

The implementation of the SHAP method without using a masker was also implemented for each emotional class individually by using a conditional check. However, the condition check only considered the predicted emotional class.
To further analyze the SHAP method and the FER models decision-making-processes, the SHAP explanations of the misclassifications were also looked at using a similar

implementation process. This was done for each pixel of the image and with a masker. The SHAP method was then further implemented using a masker with a blur effect to evaluate its impact on the interpretability. This implementation process, shown in Figure 13, also involved iterating through the instances of the test datasets. However, in this case the whole test datasets were evaluated.

Identical to the first SHAP implementation, the instance that needed to be explained was

```
model = keras.models.load_model('Model/model_evaluation_xai_FER.hdf5')

for xi in range(len(X_test)):
    instance_to_explain = np.expand_dims(X_test[xi], axis=0)

    preds = model.predict(instance_to_explain)
    emotion_value = get_class(preds)
    emotion = int(emotion_value[0][0])

    if emotional_classes[emotion] == 'Happy':
        masker = shap.maskers.Image("blur(15, 15)", X_test[xi].shape)
        explainer = shap.Explainer(model,
                                   masker,
                                   output_names = emotional_classes)
        shap_values = explainer(instance_to_explain,
                                max_evals = 1000,
                                batch_size = 50,
                                outputs=shap.Explanation.argsort.flip[:4])
        shap.image_plot(shap_values, instance_to_explain)
```

Figure 13: Implementation of SHAP with masker "blur"

expanded to align with the input dimensions anticipated by the model and the emotional class of the instance was determined, using the previously mentioned 'get_class' function. The next steps of the process differed from the first implementation. These steps included the following:

- Condition check:
  The Shapley values were calculated and plotted for each emotional class individually. Therefore a condition check was conducted. If the predicted emotional class corresponded to the chosen emotion, the Shapley values were calculated and plotted. This approach facilitated an understanding of the model's behaviour for each class and enabled the evaluation of the class-specific interpretability.

- Defining a masker:
  A SHAP image masker with a blur effect was defined. The blur kernel size was set to (15, 15) but different kernel sizes were also tested to observe variations in the interpretability. These variations are explained later on.
  The blur masker is employed to obscure specific areas of the image by applying a

blurring effect. This approach helps to maintain the overall structure of the image while removing fine details, thus allowing the model to still receive a meaningful input. (S. Lundberg 2018; S. M. Lundberg and Lee 2017)
In order to calculate the Shapley values for the image, the blur masker systematically blurs different regions of the image and observes how these changes affect the model's predictions. Consequently, the method enables the determination of the importance of each region of the image, with the potential identification of facial features that contribute positively to the prediction. (S. Lundberg 2018; S. M. Lundberg and Lee 2017)

- Creating the SHAP Explainer:
  The SHAP Explainer was created using the model and the masker. The output names were set to the emotional classes. Similar to the previously mentioned DeepExplainer, this explainer was used to generate explanations for the predictions made by the FER model.

- Calculating the Shapley values:
  Similar to the previous shap implementation process, the Shapley values for the current instance were calculated by utilizing the SHAP Explainer.
  However, in this case several more paramters were defined for the calculation:

  - 'max_evals':
    This parameter specifies the maximum number of evaluations of the FER model that may be employed in the calculation of the Shapley values. A greater number of evaluations is generally associated with more accurate Shapley values, although this increases the computational time. The value of 1000 has been selected in order to achieve an optimal balance between the accuracy of the results and the computational efficiency of the process. (S. Lundberg 2018)
  - 'batch_size':
    This parameter specifies the number of samples to be processed concurrently during the computation of the Shapley values. A batch size of 50 signifies that the model will process 50 samples simultaneously. This can be modified according to the memory and computational capabilities of the system. It can be observed that larger batch sizes may result in a reduction in computation time, although this may be accompanied by an increase in the required memory. (S. Lundberg 2018)
  - 'outputs = shap.Explanation.argsort.flip[:4]':
    This parameter determines which outputs, i.e. emotional classes, are to be explained. 'Shap.Explanation.argsort.flip[:4]' selects the top four classes with

the greatest contribution to the prediction. This is particularly useful in the context of multi-class classification problems. (S. Lundberg 2018)

- Plotting the Shapley values:
  Identical to the previous implementation process, the Shapley values were shown using 'image_plot'.

As previously stated, the impact of varying kernel sizes for the masker blur on the interpretability of the model's predictions was investigated. By varying the kernel size, the degree of blur applied to the image changes, which may highlight different aspects of the feature importance.
To evaluate the interpretability of the FER models using SHAP with different kernel sizes of the masker 'blur', the image sizes and their nature, i.e. grayscaled or RGB, need to be considered.

FER 2013 dataset (48x48, Grayscale) Given the smaller dimensions of the images (48x48) and the fact that they are grayscaled, it is important to select the kernel sizes with great care in order to ensure that the resulting images are both meaningful and interpretable, while not losing significant details. Therefore useful kernel sizes to evaluate are the following:

- Small kernel (5,5):
  This kernel size allows for a fine-grained interpretability without much blurring.

- Medium kernel (9,9):
  The medium kernel size provides a balance between detail and interpretability.

- Large kernel (15,15):
  This kernel size offers more generalization and can help to identify broader regions of importance.

RAF-DB dataset (100x100, RGB) In the case of larger RGB images, it is possible to utilise kernel sizes that are relatively larger due to the increased resolution and colour channels. It would be beneficial to consider the following kernel sizes when evaluating the efficacy of a given approach:

- Small kernel (7,7):
  This kernel size allows for a fine-grained interpretability without much blurring.

- Medium kernel (15,15):
  The medium kernel size provides a balance between detail and interpretability.

- Large kernel (25,25):
  This kernel size offers more generalization and can help to identify broader regions of importance.

```
# Loading the model
model = keras.models.load_model('/home/lorch/Model/model_evaluation_xai_FER.hdf5')

# Defining the kernel sizes
kernel_sizes = [(5, 5), (9, 9), (15, 15)]

output_dir = '/home/lorch/Output/FER 2013/SHAP/Different kernel sizes'

for xi in range(100):
    instance_to_explain = np.expand_dims(X_test[xi], axis=0)

    preds = model.predict(instance_to_explain)
    emotion_value = get_class(preds)
    emotion = int(emotion_value[0][0])

    if emotional_classes[emotion] == 'Happy':
        shap_images = []  # List to store paths to SHAP images

        # Looping through the kernel sizes
        for size in kernel_sizes:
            masker = shap.maskers.Image(f"blur({size[0]}, {size[1]})", X_test[xi].shape)
            explainer = shap.Explainer(model, masker, output_names=emotional_classes)
            shap_values = explainer(instance_to_explain,
                                    max_evals=1000,
                                    batch_size=50,
                                    outputs=shap.Explanation.argsort.flip[:1])

            shap_image_path = os.path.join(output_dir,
                                           f'kernel_size_{size[0]}x{size[1]}_instance_{xi}.png')
            shap.image_plot(shap_values, instance_to_explain, show=False)
            plt.savefig(shap_image_path)
            shap_images.append(shap_image_path)
            plt.close()

        # Ploting all SHAP images together for better evaluation
        fig, axs = plt.subplots(1, 3, figsize=(15, 5))

        for idx, img_path in enumerate(shap_images):
            img = Image.open(img_path)
            axs[idx].imshow(img)
            axs[idx].set_title(f"Kernel Size {kernel_sizes[idx]}")
            axs[idx].axis('off')

        plt.show()
```

Figure 14: Implementation of evaluation of kernel sizes

Figure 14 shows the methodology employed to assess the influence of distinct kernel sizes on the Shapley values associated with the FER model, exemplified by the FER model trained on the FER 2013 dataset.
Identical to the previous implementation process, the following steps were done:

- Loading the FER model
- Iterating through instances of the test dataset

- Expanding the dimension of the instance to be explained
- Determining emotional class for the instance
- Using a conditional check to only use images with chosen emotional class
- Defining 'masker', 'explainer' and 'shap_values'
- Using 'image_plot' to show the Shapley values

However, in this case, the masker was defined three times using different pre-defined kernel sizes. The kernel sizes were identical to the previously listed small, medium and large kernel sizes for both the FER 2013 and the RAF-DB dataset.
As illustrated in Figure 14, the multiple definitions of the masker, explainer and Shapley values were achieved through the use of a for loop. Furthermore, in order to present the image plots in a linear sequence and to save them for further evaluations, the SHAP plots were stored in a pre-defined directory until they could be displayed.
This results in the generation of a single figure, comprising all the saved SHAP images, with subplots. Each subplot corresponds to a different kernel size, thereby enabling a visual comparison of the impact of the Shapley values.

3.3.2 Local Interpretable model-agnostic Explanations (LIME)

As with the SHAP method, the LIME method was implemented in several variations in order to evaluate the method and the interpretability of the FER model. The implementation of the LIME method was similar for both FER models, utilizing the lime library and one implementation without the use of the lime library. (Marco Tulio Ribeiro, Singh, and Guestrin 2016)
All implementations of the LIME method utilizing the lime library employed a uniform underlying process, which is shown in Figure 15, with only a few minor variations. The following steps were included in the implementation process (Marco Túlio Ribeiro, Singh, and Guestrin 2016; Marco Tulio Ribeiro, Singh, and Guestrin 2016):

- Loading the pre-trained FER model from the specified file path.
- Initializing the LimeImageExplainer:
  The initialization of the LimeImageExplainer represents a crucial step in the process of setting up the necessary environment for the generation of explanations for image model predictions. This initalization creates an instance of the explainer, prepares it to handle images, and encapsulates the methods required for segmentation, perturbation, prediction, and explanation.

```python
#Loading pre-trained model
model = load_model('/home/lorch/Model/model_evaluation_xai_RAF_DB.hdf5')

# Initializing LIME Image Explainer
explainer = lime_image.LimeImageExplainer()

# Segmentation into superpixel with SLIC (Simple linear iterative clustering)
segmenter = SegmentationAlgorithm('slic', n_segments=50, compactness=10, sigma=1)

instance_to_explain = np.expand_dims(X_test[0], axis = 0)

preds = model.predict(instance_to_explain)
emotion_value = get_class(preds)
emotion = int(emotion_value[0][0])

instance_to_explain = X_test[0]

#Displaying the segmented image
segments = segmenter(instance_to_explain)

# Explain the prediction
explanation = explainer.explain_instance(instance_to_explain,
                                         classifier_fn=lambda x: model.predict(x),
                                         top_labels=1,
                                         hide_color=0,
                                         num_features = 10,
                                         num_samples=500,
                                         segmentation_fn = segmenter,
                                         model_regressor = Ridge())

# Display the explanation
temp, mask = explanation.get_image_and_mask(explanation.top_labels[0],
                                            positive_only=True,
                                            num_features=5,
                                            hide_rest=True)

fig, axes = plt.subplots(1, 3, figsize=(15, 5))

# Plot the segmented image
axes[0].imshow(instance_to_explain)
axes[0].set_title("Original Image, Prediction: {}".format(emotional_classes[emotion]))
axes[0].axis('off')

axes[1].imshow(mark_boundaries(instance_to_explain, segments))
axes[1].set_title("Segmented Image")
axes[1].axis('off')

# Plot the LIME explanation
axes[2].imshow(mark_boundaries(temp, mask))
axes[2].set_title("LIME Explanation")
axes[2].axis('off')

plt.show()
```

Figure 15: Implementation of LIME method exemplary for RAF-DB FER model

- Segmentation into superpixels with SLIC:
  The Simple Linear Iterative Clustering (SLIC) was used for the image segmentation

of the chosen instance. The following parameter were defined for the segmentation:

  - 'n_segments':
    The number of segments to be employed in the division of the image. The optimal number of segments was determined to balance between over-segmentation and under-segmentation for the different images of the two datasets.
  - 'compactness':
    This parameter controls the compactness of the segments. A moderate value ensures that the segments are neither excessively compact nor dispersed.
  - 'sigma':
    The smoothing parameter for the segmentation. A small value results in a slight smoothing effect.

- Extracting the emotional class of the instance

- Explaining the predictions:
  The 'explain_instance' method in LIME´s LimeImageExplainer class generated a local explanation for a given instance by:

  - Perturbation:
    LIME generated a set of perturbed instances by making small changes to the input instance. In the case of image data, this process involved modifying the superpixels.
  - Prediction:
    The black box model was employed to predict the outcomes for these perturbed instances.
  - Weighting:
    Each perturbed instance was assigned a weight based on its proximity to the original instance.
  - Local model training:
    A local interpretable model was trained on the perturbed instances, utilizing their features and the corresponding predictions from the black box model.
  - Explanation:
    The coefficients of this linear model indicated the importance of the features, i.e. superpixels, in the prediction made by the black box model.

  The following parameters were defined for the 'explain_instance':

  - 'classifier_fn':
    Function to get model´s prediction.

  - 'top_labels':
    Sets the number of top predicted labels to be explained.
  - 'hide_color':
    Sets the color to hide the superpixels. For both models this was set to 0, i.e. black.
  - 'num_features':
    Number of superpixels to highlight in the explanation.
  - 'num_samples':
    Number of samples to generate for each explanation. In this case it is important to find a balance between explanation accuracy and computation time.
  - 'segmentation_fn':
    Segmentation function used to divide the images.
  - 'model_regressor':
    Defines the local interpretable model to be used. The default model is Ridge regression.

- Displaying the images:
  The 'get_image_and_mask' function was used to generate an image and mask for the explanation. The following parameters determined the visual output of the LIME method:
  - 'positive_only':
    If this parameter is set to 'True', only the positive contributions are shown in the explanation.
  - 'num_features':
    Sets the number of top features, i.e. superpixels, to be displayed in the explanation.
  - 'hide_rest':
    If this parameter is set to 'True' the rest of the image is hidden to focus on the important superpixels.

- Plotting the results:
  The original image to be explained, the segmented image and the LIME explanation were visualized using 'imshow'.

Because the lime library does not work with grayscaled images, the FER 2013 test dataset images had to be converted into RGB images. This was done by using the 'rgb2gray' and 'gray2rgb' modules from skimage.
The implementation process depicted in Figure 15 was also carried out in a similar manner

to the SHAP implementation, with each emotional class being addressed individually. This also included looking at the missclassifications.
Furthermore, the different local interpretable model´s were evaluated based on the improvement of the interpretability of the FER models explanations. This was done by implementing the LIME method without the lime library and with different interpretable models such as:

- Linear regression:
  Stadard linear regression.
- Ridge regression:
  Linear model that includes a regularization term to prevent overfitting.
- Lasso regression:
  Linear model similar to Ridge, but it employs L1 regularization. This can result in the creation of sparse models, where some coefficients are exactly zero.
- ElasticNet:
  Combines both L1 and L2 regularization, providing a balance between Ridge and Lasso.

In order to implement the LIME method without the lime library, it was necessary to create the perturbation first. Therefore, a perturbation function was defined, as shown in Figure 16. (Marco Tulio Ribeiro, Singh, and Guestrin 2016) The function was defined to create

```
def create_perturbations(img, segments, num_perturb=500):
    active_pixels = np.unique(segments)
    perturbations = np.random.binomial(1,
                                       0.5,
                                       size=(num_perturb,
                                             len(active_pixels)))
    perturbed_images = np.tile(img, (num_perturb, 1, 1, 1))

    for i, pert in enumerate(perturbations):
        for pixel, active in zip(active_pixels, pert):
            if active == 0:
                perturbed_images[i][segments == pixel, :]
                = np.mean(img[segments == pixel, :], axis=(0, 1))
    return perturbations, perturbed_images
```

Figure 16: Definition of function for the Perturbation

the perturbed versions of an input image by masking certain segments, i.e. superpixels. The first step of the function included identifying the unique segment labels in the

segmented image, representing the superpixels.
As shown in Figure 16, the perturbations were generated using a binary matrix with the dimension (number of perturbations, length of active pixels). The function employs a binomial distribution with a probability of 50% to determine whether each segment should be active, i.e. 1, or inactive, i.e. 0, in each perturbation.
The perturbed images were initialized by duplicating the original image a specified number of times. The specified number was the number of perturbations.
The function then iterated over each perturbation and its corresponding active and inactive segments. If the segment was inactive, the segment´s pixels in the perturbed image were replaced with the mean color of the segment from the original image. In contrast to previous implementations of LIME, the inactive segments were not replaced with a black segment. Therefore the method´s contribution to the interpretability of the FER model was analyzed using both options.
At the end of the function, the binary matrix of the perturbations and the array of the perturbed images were returned.
In order to receive the explanations, the 'explain' function was defined as partly illustrated in Figure 17. The 'explain' function had two parameters:

- 'model': FER model to be explained.
- 'img': Input image to be explained.

The implementation included the following steps:

- Image segmentation:
  The input image was stored in the variable 'original_image' for further steps.
  Identical to the previous LIME implementation, the 'SegmentationAlgorithm' function was used to segment the input image into superpixels using the SLIC algorithm.
  The segmented image was stored in the variable 'segments'.
- Perturbation generation:
  Using the pre-defined perturbation function, the perturbed versions of the image were generated.
- Model predictions on perturbations:
  Similar to the previous LIME implementation the model´s predictions were generated.
- Identification of top predicted class:
  The class with the highest prediction probability was selected and stored in the 'top_class' variable.
- Training a linear model:
  In this section, the local interpretable model was trained to approximate the relationship between the perturbations and the probabilities of the top predicted class.

```python
def explain(model, img):
    original_image = img

    # Segmentation into superpixels with SLIC
    segmenter = SegmentationAlgorithm('slic', n_segments=48, compactness=10, sigma=1)
    segments = segmenter(original_image)

    # Creating perturbations with pre-defined function
    perturbations, perturbed_images = create_perturbations(original_image, segments)
    # Predicting perturbed images
    predictions = model.predict(rgb2gray(perturbed_images))

    # Taking probabilities for the top predicted class
    img = np.expand_dims(img, axis=0)
    original_prediction = model.predict(rgb2gray(img))
    top_class = original_prediction.argmax()
    target_class_probs = predictions[:, top_class]

    # Training linear model
    clf = LinearRegression()
    clf.fit(perturbations, target_class_probs)
    coefficients = clf.coef_

    # Showing coefficients on superpixels
    num_top_features = 4
    top_features = np.argsort(coefficients)[-num_top_features:]
    mask = np.zeros(segments.shape)
    for feature in top_features:
        mask[segments == feature] = 4
```

Figure 17: Definition of function for the explanations

As previously mentioned this step was done with multiple local interpretable models. The coefficients of these models, indicated the importance of each superpixel.

- Visualization of important superpixels:
  The most important superpixels were then identified by sorting the coefficients and selecting the top number of features. Furthermore, a mask was created to highlight these important superpixels in the segmented image.

3.4 Evaluation metrics

The following steps were taken to evaluate the models´ performances and the improvement of the interpretability in the context of FER.

The model evaluation process included defining a function for single class predictions, converting the one-hot-encoded labels to integer class labels, creating and displaying a

confusion matrix and calculating various performance metrics such as accuracy, precision, recall and F1-score. Furthermore, a classification report was created to receive a comprehensive view of the models performances.
The above mentioned evaluation process was done for the model with the highest validation accuracy of both datasets FER 2013 and RAF-DB.

The interpretability evaluation process included looking at various interpretability metrics such as stability, simplicity, comprehensiveness, consistency, computational efficiency and visual interpretability.
The process was done for both SHAP and LIME and the FACS was used to measure the consistency with the domain knowledge.

3.4.1 Performance metrics

To evaluate the models predictions, a function was defined that extracts the single class predictions from the model's output probabilities. As shown in Figure 18, the function

```
# Defining function to get class of one prediction
def get_class(preds):
  pred_class = np.zeros((preds.shape[0],1))

  for i in range(len(preds)):
    pred_class[i] = np.argmax(preds[i])

  return pred_class

# Checking successful definition of function
pred_class = get_class(preds)
```

Figure 18: Definition function get_class

'get_class' iterated over the predicted probabilities and assigned the class with the highest probability to each instance. This step was crucial for transforming the model's output into a format that was suitable for the comparison with the true labels.
The dataset's true labels were converted into one-hot-encoded vectors during the data preprocessing process. To facilitate the comparison with the predicted labels, this conversion was reversed, resulting in integer class labels.
In order to receive a visual representation of the model´s performance, a confusion matrix was created and displayed. The confusion matrix presents the counts of true positives, false positives, true negatives, and false negatives. Therefore it helps to identify specific classes where the model performs well and those where the model struggles. Nevertheless, it should be noted that the matrix itself is not a performance matrix per se. However, it is a useful tool for understanding the model´s performance across the different emotional classes and was therefore included in the evaluation process.

To evaluate the model´s performance the following performance metrics were used (Pedregosa et al. 2011):

- Accuracy:
  Accuracy is defined as the proportion of true results, including true positives and true negatives, among the total number of cases examined. This allows for the general assessment of the model's performance across all classes.

- Micro Precision:
  Micro Precision calculates the precision, i.e. the ratio of true positive predictions to the total positive predictions, globally by counting the total true positives and false positives.

- Micro Recall:
  Micro Recall computes the recall, i.e. the ratio of true positive predictions to the actual positive instances, globally by counting the total number of true positives and false negatives.

- Micro F1-score:
  Micro F1-score is the harmonic mean of Micro Precision and Micro Recall. The metric is calculated globally by counting the total true positive, false negatives and false positives.

- Macro Precision:
  Macro Precision calculates the precision for each class individually and then takes the average. Each class is treated equally, regardless of its size.

- Macro Recall:
  Macro Recall calculates the recall for each class individually and finds their unweighted mean.

- Macro F1-score:
  Macro F1-score is the harmonic mean of Macro Precision and Macro Recall. It treats each class equally and calculates a single performance metrics across all classes.

- Weighted Precision:
  Weighted Precision calculates the precision for each class, weights it by the number of true instances for each class, and then averages these values.

- Weighted Recall:
  Weighted Recall calculates the recall for each class, weights it by the number of true instances for each class, and then averages these values.

- Weighted F1-score:
  Weighted F1-score is the harmonic mean of Weighted Precision and Weighted Recall. It accounts for the class distribution by weighting each class's F1-score by the number of true instances in each class.

The chosen performance metrics provided a detailed view of the model´s performance across the different aspects of the classification.
In order to evaluate the models performances on each class individually, a classification report was included in the evaluation process. The report contained the precision, recall and f1-score of each class individually. Furthermore it also showed the support numbers, i.e. the number of actual occurrences of each class in the dataset. As well as that the overall accuracy, macro precision, macro recall, macro F1-score, weighted precision, weighted recall and weighted F1-score were displayed in the classification report. These metrics were identical to the previously calculated performance metrics.
The classification report complements the confusion matrix and performance metrics, providing a detailed performance analysis for each emotional class.

By following these evaluation process steps, a comprehensive analysis of the FER models was ensured. Furthermore the strengths and areas for improvement were identified during this process.

3.4.2 Interpretability metrics

This section provides a detailed description of the methodology used to evaluate the interpretability metrics for SHAP and LIME. The selected metrics are:

- Stability:
  Stability assesses whether similar or the same inputs receive similar or the same explanations. To evaluate this metric, the SHAP and LIME methods were applied mutltiple times to the same image in order to see the changes in the explanation. Furthermore, the variance, standard deviation and mean of the explanations were measured in order to gain more insights on the stability of the XAI methods.

- Simplicity:
  The simplicity of the explanation is determined by the ease with which a human can comprehend it. As previously mentioned, different number of features and different kernel sizes are used to evaluate the trade-off between the depth of the explanation and the explanation´s accuracy to ensure that simplicity does not compromise the quality of the explanation.

- Comprehensiveness:
  This metric evaluates how well the explanation captures all the relevant aspects of the

model´s decision for a particular prediction. However, this metric is hard to measure as it is not clear which facial features are important for the FER model´s prediction. Therefore, this metric can only be measured by using the domain knowledge, for instance the FACS.

- Consistency with domain knowledge:
  The consistency with domain knowledge criterion assesses the degree to which the explanations align with the existing domain knowledge and expectations. This metric was measured by comparing the facial features highlighted by the XAI methods with those known to be important according to the FACS.

- Computational efficiency:
  The concept of computational efficiency is defined as the time and computational resources required to generate explanations. The time taken to generate the explanations using LIME and SHAP were measured and the computational resources were asssessed.

- Visual interpretability:
  The visual interpretability of an explanation is determined by the extent to which it can be effectively conveyed through visual means. Therefore, visual representations of the explanations provided by SHAP and LIME were generated and assessed.

These metrics are crucial for understanding the effectiveness and reliability of the XAI methods based on the two FER models.

3.5 Implementation of the Facial Action Coding System

The FACS provides a comprehensive framework for categorizing facial movements by their appearance on the face.
As previously mentioned, the FACS decomposes the facial expressions into individual components, associated with the specific muscle movement.
Therefore, the FACS can help clarify which facial movements contribute to the model´s prediction. Furthermore, the coding system also provides the validation whether the model´s detected facial expressions align with the human-coded AUs. This ensures that the model´s interpretations are consistent with the established psychological frameworks. Therefore, the FACS was implemented for the evaluation of the XAI methods for the FER models, which offered a structured approach to interpret the facial expressions. (Cheong and Jolly 2022) The Figure 19 shows the implementation of the FACS using the py-feat library. The implementation contained the following steps (Cheong and Jolly 2022):

- Initializing the detector:
  As show in the Figure 19, an instance of the py-feat 'Detector' class was created with various models specified for different tasks:

```
detector = Detector(
    face_model="retinaface",
    landmark_model="mobilefacenet",
    au_model='xgb',
    emotion_model= "resmasknet",
    facepose_model="img2pose",
)

for xi in range(10):

    instance_to_explain = X_test[xi]

    instance_to_explain_rgb = np.squeeze(instance_to_explain, axis=-1)
    instance_to_explain_rgb = gray2rgb(instance_to_explain_rgb)

    img = instance_to_explain_rgb
    img_array = (img*255).astype(np.uint8)
    img_pil = Image.fromarray(img_array)
    img_pil = img_pil.resize((224, 224))

    img_pil.save('path_to_save_resized_image.jpg')
    single_face_img_path = 'path_to_save_resized_image.jpg'

    single_face_prediction = detector.detect_image(single_face_img_path)

    figs = single_face_prediction.plot_detections(poses=True)
    figs = single_face_prediction.plot_detections(faces='aus', muscles=True)
```

Figure 19: Implementation of the FACS

– 'face_model':
  Model used for the face detection, for instance 'retinaface'. The model utilizes a single-stage detector with pixel-level facial landmark localization. RetinaFace can detect faces with various orientations and scales. (J. Deng et al. 2019)

– 'landmark_model':
  Model used for the landmark detection, for instance 'mobilefacenet'. MobileFaceNet is a lightweight DL model designed for efficient face recognition. (Sheng Chen et al. 2018)

– 'au_model':
  Model used for the Action Unit detection, for instance 'xgb'. XGB, i.e. XGBoost, is used to predict the presence and intensity of AUs from facial images. (Cheong and Jolly 2022)

– 'emotion_model':
  Model used for emotion detection, in this case 'resmasknet'. ResMaskNet is a DL model developed for FER. It combines residual learning with attention mechanisms to improve the recognition of subtle facial expressions. The model uses ResNet to improve the feature extraction and mask-based attention to focus on relevant regions of the face. (Cheong and Jolly 2022)

– 'facepose_model':
Model used for the face pose estimation, here exemplary 'img2pose'. Img2pose is a DL model for face pose estimation. It predicts the 3D pose of a face from a 2D image by estimating the rotation and translation vectors that align the face with a 3D model. (Albiero et al. 2020)

These parameters can be adjusted, using different pre-trained models included in the py-feat library.

- Selection and preparation of the instance to explain:
A for loop was used to iterate through the test datasets. In the case shown in Figure 19, the first ten instances of the FER 2013 test dataset were looked at.
As in previous implementation processes, the instance to explain was selected and for the FER 2013 dataset, the grayscaled image instance was converted to a RGB image by squeezing the last dimension and using the 'gray2rgb' function of skimage. For the RAF-DB dataset, this step was not neccessary.
The image pixel values were scaled to the range of (0,255), changed to the 'unint8' type, converted from a numpy array to a PIL image and resized to 224x224 pixels in order to fit the requirements of the detector.
Because the 'detect_image' function only works with a file path, the prepared image was temporarily saved to a specified path.

- Prediction:
The 'detect_image' method was used to make predictions on the saved image contained in the image path file.

- Plotting the detection:
The 'plot_detection' method of py-feat was used to generate plots of the detection. The first use of the method included poses in the plot, and the second line included AUs and muscles.

The implementation process resulted in the output of two plots. The first plot showed the detected face along with the estimated face poses. Therefore it illustrated the image with a bounding box of the face detection, an outline of the facial landmarks detected and overlaid 3-dimensional axes representing the pitch, yaw and roll of the face. The plot also provided the AUs and the emotional class detected in the image. (Cheong and Jolly 2022) The second plot showed a template face with the highlighted AUs and muscle activation. The plot also included the identical AUs and emotional classes detected in the explained image. (Cheong and Jolly 2022)

4 Results

4.1 Performance of the FER models

In order to perform the evaluation of the FER models, both models with the highest validation accuracy were loaded using the keras library. This methodology ensured that the models with the highest validation accuracy were used for the predictions.
The model´s performance on the test datasets of FER 2013 and RAF-DB was summarized by various classification metrics. The following sections provide a detailed description of these results.

4.1.1 Performance of the FER model trained on FER 2013 dataset

The predictions on the test dataset of FER 2013 were generated and verified for consistency in shape and length.
The pre-defined 'get_class' function converted the probability outputs into class predictions, ensuring the model´s outputs were interpretable.
Figure 20 shows the confusion matrix, which provides a visual representation of the true

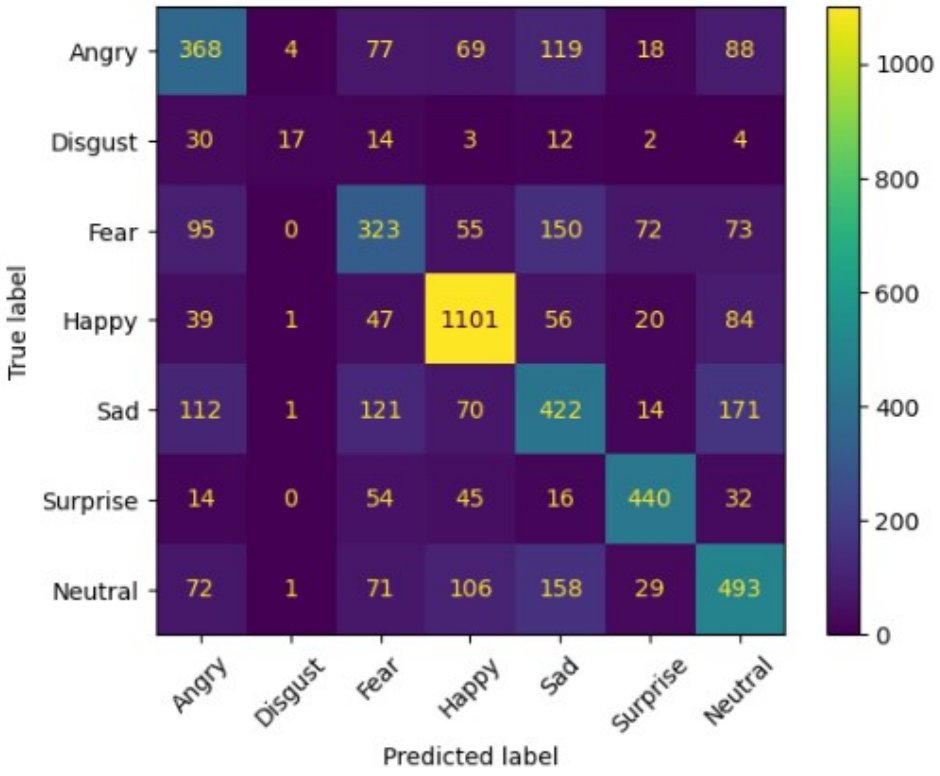

Figure 20: Confusion matrix of the FER model trained on the FER 2013 dataset

labels in comparison to the predicted labels. The colour intensity in the matrix is indicative of the frequency of predictions, thereby aiding in the identification of misclassified instances and overall model performance trends.

The FER model demonstrated a strong performance in recognizing the 'Happy' emotion, correctly identifying 1101 out of 1348 instances. However, the model frequently confused 'Angry' with 'Sad' and 'Fear'. The 'Disgust' emotion was often misclassified, most commonly as 'Anger'. Similarly, 'Fear' was sometimes mislabeled as 'Sad' or 'Angry'. The 'Sad' emotion was often mistaken for 'Angry', 'Neutral', and 'Fear'. While 'Surprise' was generally well classified, it was occasionally misclassified as 'Neutral' or 'Fear'. Lastly, 'Neutral' was often confused with 'Sad' and 'Happy'. As Figure 21 illustrates, the model

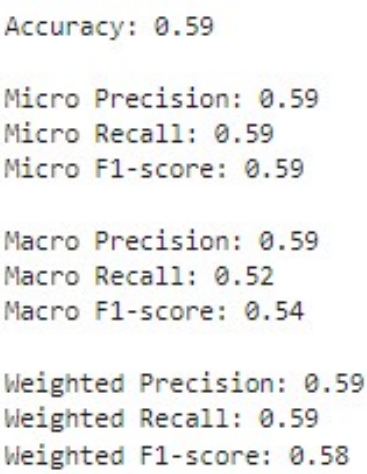

Figure 21: Performance metrics of the FER model trained on the FER 2013 dataset

trained on the FER 2013 dataset shows moderate performance with an overall accuracy of 59%.
Micro precision, recall and F1-score all have a value of 0.59, indicating that considering each individual prediction equally, the model performs at a moderate level.
The macro precision, recall and F1-score show the performance across classes without considering class imbalance, with scores of 0.59, 0.52, and 0.54 respectively.
Weighted precision, recall and F1-score, all around 0.59, suggest that the model maintains a consistent level of performance across the emotional classes when accounting for the number of instances in each class.
The classification report depicted in Figure 22 presents the precision, recall and F1-score for each emotional class, along with the number of true instances of each class, i.e. the support. This report offered insights into the model's ability to identify each emotion.

4.1.2 Performance of the FER model trained on RAF-DB dataset

Identical to the evaluation process of the FER model trained on the FER 2013 dataset, the predictions on the test dataset of RAF-DB were generated and verified for consistency in shape and length and the pre-defined 'get_class' function converted the probability outputs into class predictions.
To get a first overview of the model´s performance, the confusion matrix was build. As

Classification report

	precision	recall	f1-score	support
Angry	0.50	0.50	0.50	743
Disgust	0.71	0.21	0.32	82
Fear	0.46	0.42	0.44	768
Happy	0.76	0.82	0.79	1348
Sad	0.45	0.46	0.46	911
Surprise	0.74	0.73	0.74	601
Neutral	0.52	0.53	0.53	930
accuracy			0.59	5383
macro avg	0.59	0.52	0.54	5383
weighted avg	0.59	0.59	0.58	5383

Figure 22: Classification report of the FER model trained on the FER 2013 dataset

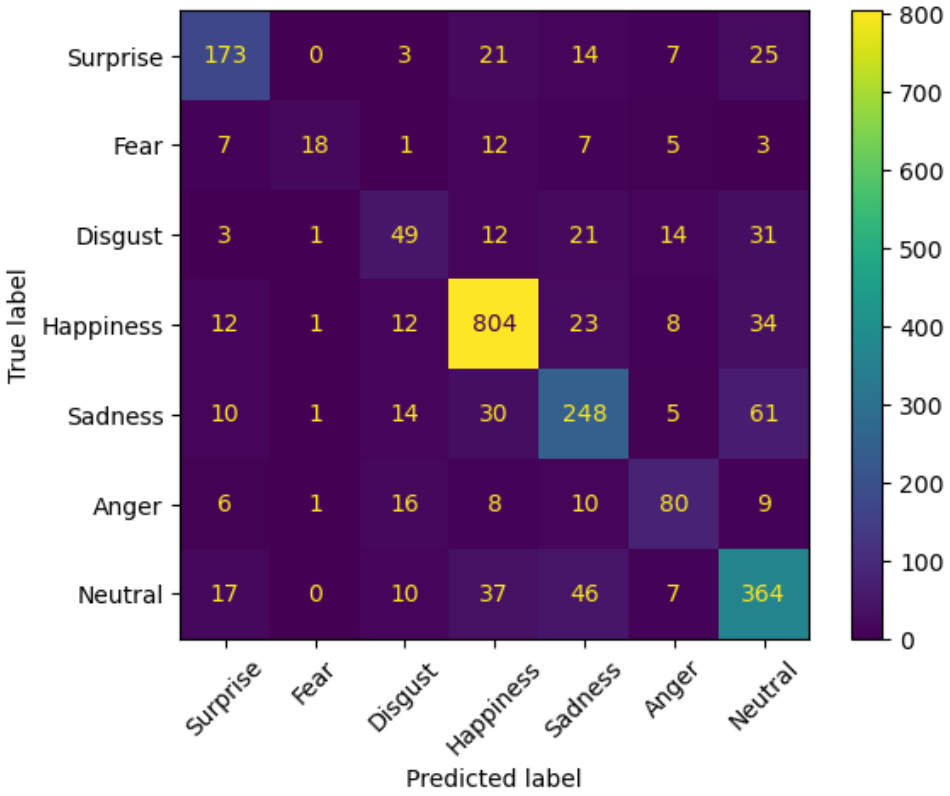

Figure 23: Confusion matrix of the FER model trained on the RAF-DB dataset

illustrated in Figure 23, the model demonstrated an accurate identification of instances of 'Happiness', with 804 out of 894 instances correctly classified. However, 'Fear' was often misclassified as other emotions, with only 18 out of 53 instances correctly labelled. The emotional class of 'Neutral' was sometimes confused with 'Sadness' and 'Happiness'. The emotional classes of 'Disgust' and 'Anger' were frequently misclassified.

Figure 24 shows the performance metrics of the FER model trained on the RAF-DB

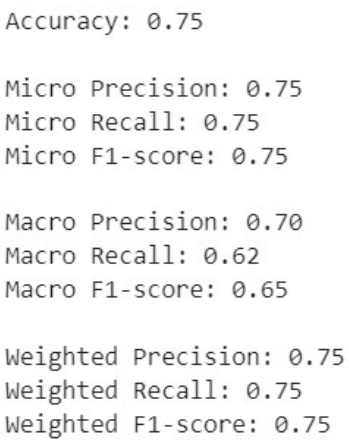

Figure 24: Performance metrics of the FER model trained on the RAF-DB dataset

dataset. With an overall accuracy of 75%, the classification model demonstrates a solid performance.

The micro precision, recall and F1-score are all 75%, indicating that, when each individual prediction is considered equally, the model performs well.

Macro precision, recall and F1-score show a more balanced view of the performance across the emotional classes. With the values of 70%, 65% and 65%, they highlight the differences in performance between the emotional classes.

The metrics weighted precision, recall and F1-score are also all at 75%, indicating that the model maintains a consistent level of performance across the emotional classes.

Figure 25 illustrates the classification report. As previously mentioned the report presents the precision, recall, F1-score and support for each emotional class.

Classification report

	precision	recall	f1-score	support
Surprise	0.76	0.71	0.73	243
Fear	0.82	0.34	0.48	53
Disgust	0.47	0.37	0.42	131
Happiness	0.87	0.90	0.88	894
Sadness	0.67	0.67	0.67	369
Anger	0.63	0.62	0.62	130
Neutral	0.69	0.76	0.72	481
accuracy			0.75	2301
macro avg	0.70	0.62	0.65	2301
weighted avg	0.75	0.75	0.75	2301

Figure 25: Classification report of the FER model trained on the RAF-DB dataset

4.2 Performance of the XAI methods

In order to perform the evaluation of the XAI methods, both FER models with the highest validation accuracy were loaded as previously explained.
The XAI methods performance on improving the interpretability of the FER models was analyzed by using several variations of both methods. The following section provides a detailed description of the results.

4.2.1 Shapley Additive Explanations (SHAP)

For the FER 2013 model´s predictions, the SHAP method was first used to calculate the Shapley values of each pixel of an image individually. As previously explained this was done by not using a masker. The Figure 26 pairs facial images with their corresponding SHAP

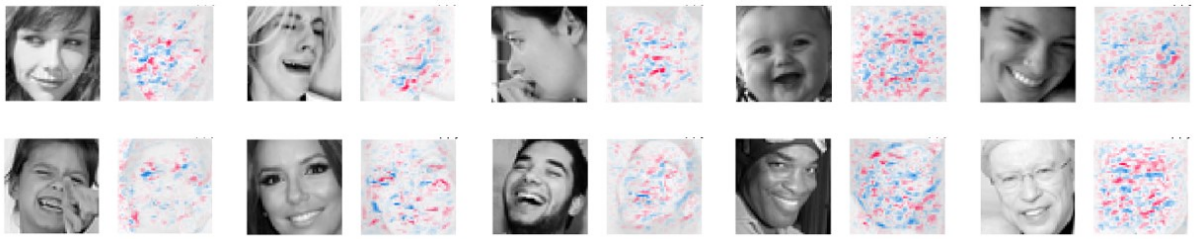

Figure 26: Shapley values calculated withot masker of emotional class 'Happy'

visualizations, which highlight the regions influencing the prediction of the emotional class 'Happy'.
As each pixel´s contribution to the prediction was looked at, specific facial regions that influence the prediction were hard to identify. Across most of the images shown in Figure 26, only the face was highlighted, indicating that the FER model made its decisions based on facial features.
When examining the SHAP plot closely, the facial regions such as the mouth, eyes and cheeks tended to show red regions. However, these regions were often surrounded by blue areas.
Therefore the 'blur' masker was used to receive more interpretable visualizations and therefore improving the model´s interpretability as well. Figure 27 shows 10 pairs containing facial images of the RAF-DB test dataset with the predicted emotional class 'Happy' with the corresponding SHAP visualization. Due to the 'blur' masker the highlighted areas were more generalized, providing a more human-like explanation of the model´s prediction.
All of the images in Figure 27 contained red highlights in the area of the mouth, indicating that this facial feature was an indicator for the emotional class 'Happy'. Furthermore, some images also highlighted the forehead and cheeks in red.
These findings aligned with the findings of the SHAP visualizations without the masker

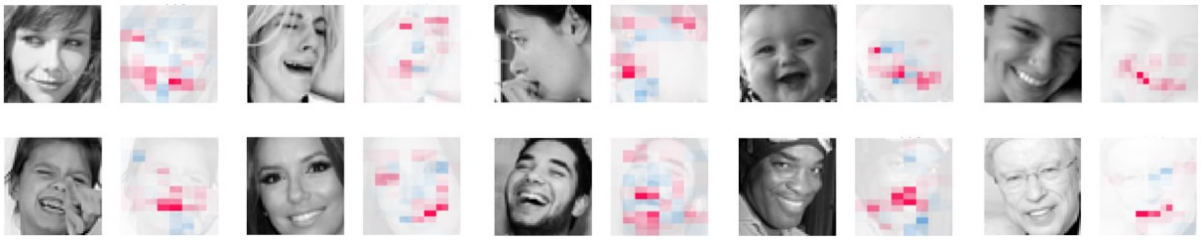

Figure 27: Shapley values calculated with masker of emotional class 'Happy'

for the emotional class 'Happy' except for the eyes.
However, in some images, parts of the forehead appeared in blue, suggesting these areas may slightly detract from the model´s prediction of 'Happy'. Occasionally, parts of the cheeks and jawline showed blue areas as well. This indicated that in some cases, these areas had a negative impact on the prediction.
In order to see if these findings of the emotional class 'Happy' aligned with the knowledge of Pyschology, the FACS were analyzed.
Figure 28 illustrates one output of the FACS implementation for one instance of the FER

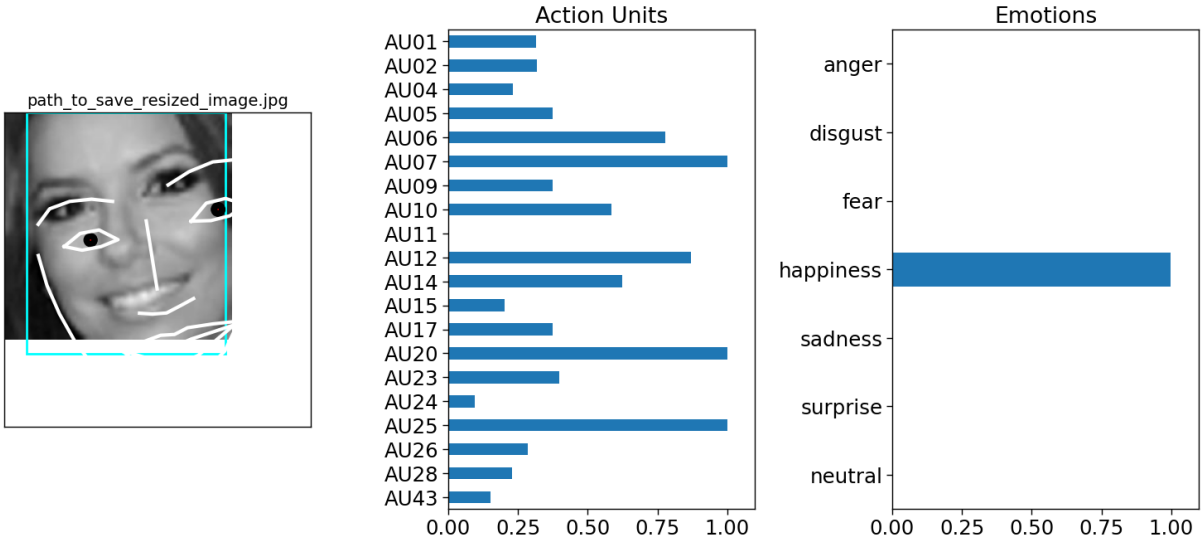

Figure 28: Output of the FACS implementation for one instance of the FER 2013 test dataset

2013 test dataset with the emotional class prediction 'Happy'. The most present AUs in the image were:

- AU06 (Cheek Raiser)
- AU07 (Lid Tightener)
- AU12 (Lip Corner Puller)

- AU20 (Lip Stretcher)
- AU25 (Lips part)

Therefore, the red highlights of the mouth across all images and the red highlights of the cheeks in some images aligned with the findings of the FACS analysis.
The AUs 6, 12 and 25 are indicators for the emotional class 'Happy' (Cheong and Jolly 2022). The AU20 and AU07 are usually indicators for the emotional class 'Fear'. Therefore, these AUs should have been highlighted in blue by the SHAP method. When looking at the image of the test dataset in Figure 27, the image contained blue highlights on the right eye and left side of the lip. Therefore, to some extent the negative contributed AUs were also highlighted blue in the SHAP visualizations.
It should be noted that the face detection shown in Figure 28, had a slight misplacement.
The Figure 29 shows outtakes of the SHAP visualizations without a masker of the emotional class 'Disgust'. As the confusion matrix showed, the emotional class 'Disgust' was often misclassified by the FER 2013 model.
Similar to the images shown in Figure 26, common areas of all visualizations highlighted

Figure 29: Shapley values calculated withot masker of emotional class 'Disgust'

by the Shapley values were the mouth and the eyes. Furthermore the nose also appeared to be highlighted in red across all images. This indicated that the ML model relies on these regions to identify 'Disgust'.
However, the same areas of the face were also highlighted in blue, suggesting that these areas did not provide a positive contribution to the emotional class.
Figure 30 depicts the blurred Shapley values of the emotional class 'Disgust' for the

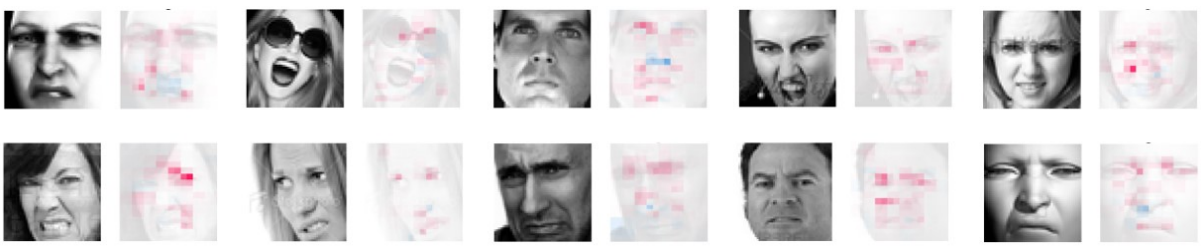

Figure 30: Shapley values calculated with masker of emotional class 'Disgust'

FER 2013 test dataset. Compared to the masker output of the emotion 'Happy' the Shapley value´s highlights were less intense in color. This suggested that the values of the contributions were smaller or widely distributed.
However, nearly all images had red areas around the mouth and eyes, thereby aligning with the findings of the SHAP visualizations without a masker. Furthermore, some images showed red highlights in the areas around the cheeks and brows, which could have indicated a small positive contribution to the model´s prediction.
4 out of the 10 images showed blue areas below the nose in the upper lip region. This could have been an indicator for another emotional class, therefore detracting the model´s prediction of 'Disgust'.
Figure 31 shows the output of the FACS implementation for one instance of the dataset

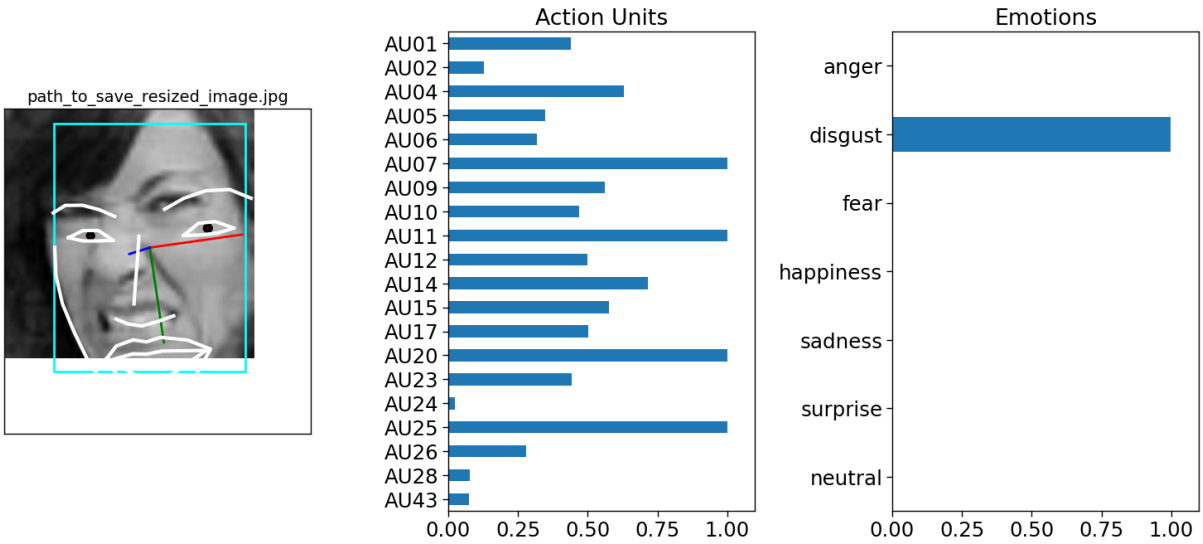

Figure 31: Output of the FACS implementation for one instance of the FER 2013 test dataset

with the emotional class 'Disgust'. The most present AUs were:

- AU04 (Brow Lowerer)
- AU07 (Lid Tightener)
- AU11 (Nasolabial Deepener)
- AU14 (Dimpler)
- AU20 (Lip Stretcher)
- AU25 (Lips part)

The combination of the AUs 6 (Cheek Raiser), 9(Nose Wrinkler), 11, 15(Lip Corner Puller) and 17(Chin Raiser) is usually an indicator for the emotional class 'Disgust'. The output

of the FACS only showed one of these AUs, i.e. AU 11. However, the SHAP visualization of the image showed red highlights in the area of the brows, the lip corner and the cheeks. Therefore, the SHAP method identified a few more AUs that need to be present for the facial expression of 'Disgust'. (Cheong and Jolly 2022)
In order to understand why the model misclassified some images, the false positive predicted images were looked at.

As illustrated in Figure 32, the true label of the image is 'Angry'. However, the FER

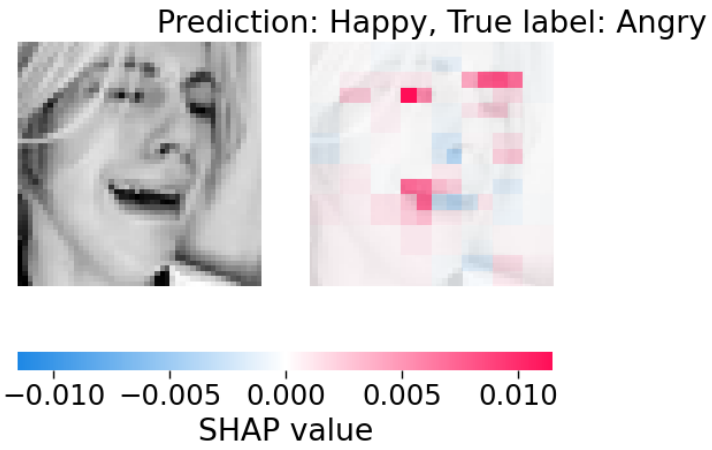

Figure 32: SHAP Output of a falsely predicted image

2013 model´s prediction was 'Happy'. As previously mentioned the AU12 (Lip corner Puller) plays a crucial role for the facial expression of happiness. The misclassified image showed red highlighted areas at the left lip corner. Therefore this could have been one reason why the model misclassified the image.
Furthermore, the combination of AUs that characterizes anger contains the AU04 (Brow Lowerer), however in this image, the brows seem to be risen and provide a positive contribution to the prediction of 'Happy'. This can also be an indicator for the misclassification.
To ensure that the right kernel size for the masker 'blur' was chosen, a variation of kernel sizes was analyzed.

The Figure 33, shows these variations, by illustrating 3 outputs of the SHAP method using the kernel size of 5x5, 9x9 and 15x15.
The 5x5 kernel had Shapley values that were spread out, with some concentration around the mouth and eyes. The values were relatively small in magnitude, therefore these values might not have provided enough details for the model´s prediction.
The 9x9 kernel showed a higher magnitude. However, the impact regions were only on one side of the face, suggesting that the kernel size only focused on some details.
The kernel size of 15x15 appeared to provide the best distribution of the Shapley values, thereby providing details on various facial regions. This kernel was used for the evaluation of the SHAP method as it provided the most insights on the model´s decision-making-process.

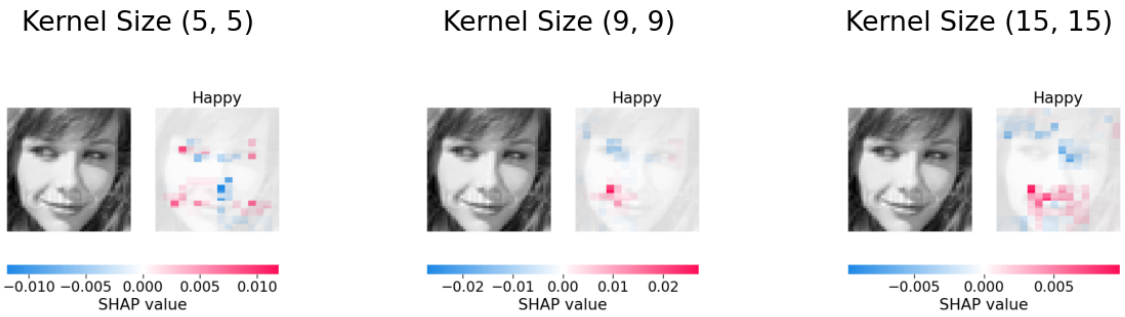

Figure 33: SHAP Output using different kernel sizes of the masker 'blur'

In order to evaluate the stability of the SHAP method, the Shapley values were calculated with and without a masker for one instance 10 times and the mean, standard deviation and variance were measured.

Figure 34 shows the SHAP output for these measurements. The mean Shapley values

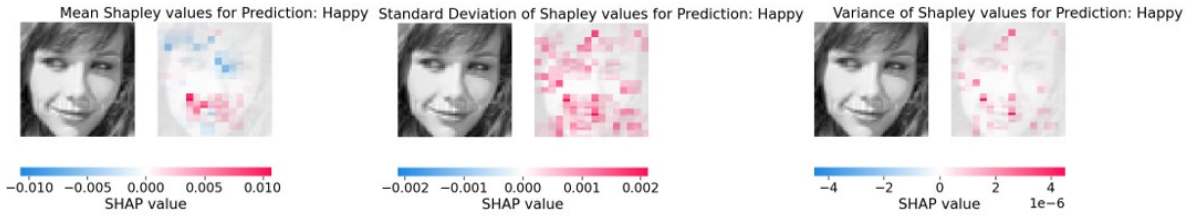

Figure 34: SHAP Output for mean, standard deviation and variance

provided an average importance of each pixel for predicting the emotional class 'Happy'. The blue and red regions indicated pixels that contribute negatively and positively, respectively, to the prediction. Therefore this image showed that the mouth region, especially the lip corner, played a crucial role for the indication of the emotional class 'Happy'.
For the standard deviation and the variance, lower values indicated stability of the SHAP method. In the standard deviation plot most pixels had a low value, indicating consistent SHAP values across multiple iterations. The few areas with higher standard deviation did not dominate the image and were scattered across the image and therefore they did not significantly affect the overall stability. For the variance plot, most pixels appeared to be light pink or not highlighted at all, therefore there was only a small variance in the Shapley values, indicating that the method was stable.

Identical to the FER 2013 model, the Shapley values of the RAF-DB model´s predictions were first calculated for each pixel of an image without the use of a masker. However, as illustrated in Figure 35, with a maximum value of 0.0013, the Shapley values calculated for the first instance of the RAF-DB dataset were below the threshold for visualisation.

Therefore, the Shapley values of the RAF-DB model were only calculated using a masker.

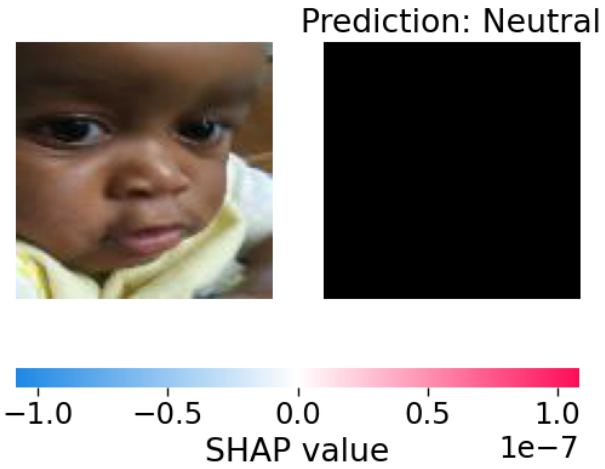

Figure 35: Output of the Shapley values of first instance of the RAF-DB test dataset

As previously explained, the Shapley values were calculated using the 'blur' masker with a kernel size of (25,25) for each emotional class individually.

Figure 36 illustrates an outtake of the SHAP method by showing the original images

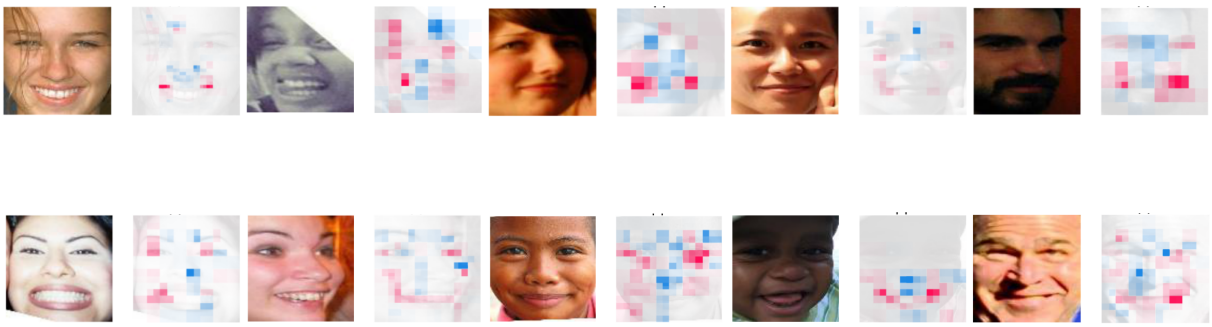

Figure 36: Shapley values calculated with masker of emotional class 'Happiness'

with the corresponding SHAP visualization. Each pair consists of a facial image with the emotional class 'Happiness' and the SHAP plot that highlights the regions influencing the prediction of the emotional class.

Across all images, red highlights were prominent around the mouth, particularly around the corners of the mouth. This suggested that the lip corners were a strong indicator of 'Happiness'. Furthermore, there were also positive contributions around the eyes in most of the pairs. This indicated that a particular shape of the eyes for instance eye crinkles could have been another indicator for the emotional class.

The blue areas, i.e. negative contributions were hard to link to one particular facial region.

However, the forehead and noise area were blue in some of the pairs, therefore these regions might have been an indicator for other emotional classes. These findings aligned with the findings using the FER 2013 model.
To evaluate the alignment with the knowledge of Psychology, the FACS implementation was also looked at for the RAF-DB test dataset.

The Figure 37 shows the FACS implementation of one instance with the emotional class

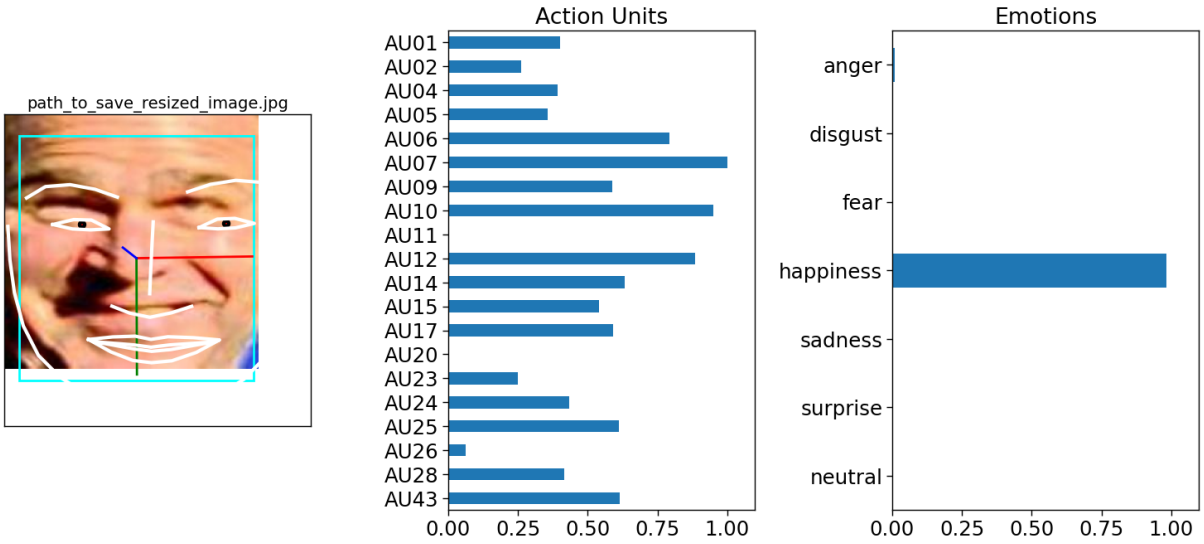

Figure 37: Output of the FACS implementation for one instance of the RAF-DB test dataset

'Happiness'. The most prominent AUs of the image were:

- AU06 (Cheek Raiser)
- AU07 (Lid Tightener)
- AU10 (Upper Lip Raiser)
- AU12 (Lip Corner Puller)

As previously mentioned, AU 6 and 12 are indicator for the facial expression of happiness according to the FACS (Cheong and Jolly 2022). The lip corners were also identified as an indicator by the SHAP method, therefore this finding aligned with the domain knowledge. Furthermore, the SHAP visualization of the image also showed positive contributions in the cheek and mouth region. Both these regions are characteristics for the facial expression of happiness according to the FACS (Cheong and Jolly 2022). Therefore, the SHAP method provided explanations that the model´s predictions aligned with the domain knowledge of the psychologists.

Given that the emotional class 'Disgust' exhibited a higher incidence of misclassification, its SHAP output, shown in Figure 38, was subjected to closer examination. Across most

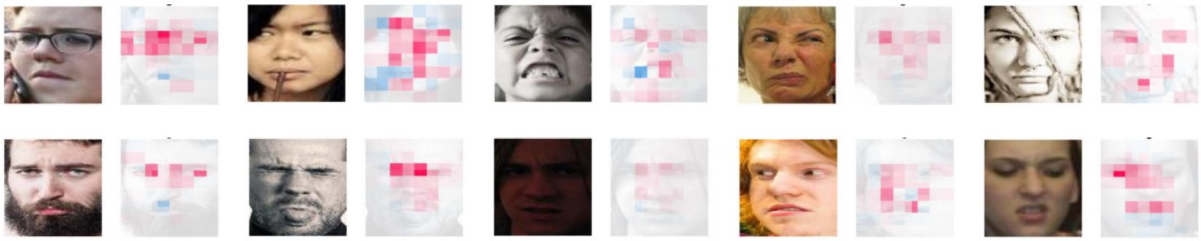

Figure 38: Shapley values calculated with masker of emotional class 'Disgust'

images, red highlights were prominent around the mouth, nose and eyes. This suggested that a specific facial expression in these regions could have been an indicator for the emotional class 'Digsust'. As well as that, the brow area was also often highlighted in a red color.

Blue areas were less prominent in the images shown in Figure 38. However, in some of the images the left side of the mouth or the upper lip were highlighted in blue. This could have been an indicator for another emotional class and therefore a potential misclassification. As previously mentioned, this analysis was done for every single emotional class.

Figure 39 shows the FACS implementation for an instance with the emotional class

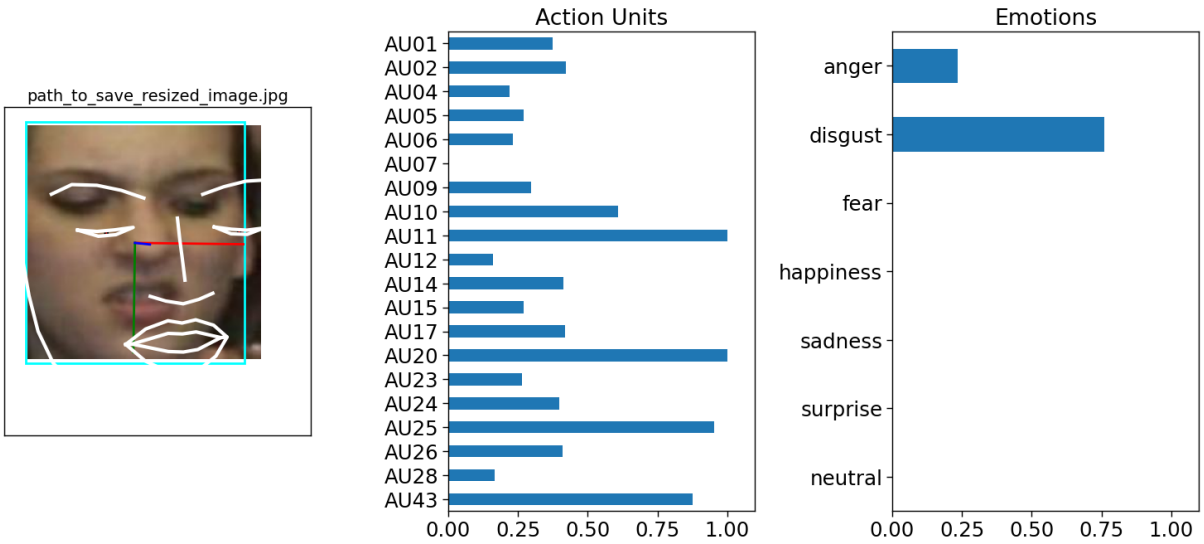

Figure 39: Output of the FACS implementation for one instance of the RAF-DB test dataset

'Disgust'. According to the implemented AU detection the following AUs were important:

- Au11 (Nasolabial Deepener)
- AU20 (Lip Stretcher)
- AU25 (Lip Part)

- AU43 (Eyes Closed)

As previously mentioned the AUs 6, 9, 11, 15 and 17 characterize the facial expression of disgust. However, similar to the findings using the FER 2013 model, only the AU11 was present in the image according to the AU detection implemented.
Despite that, when looking at the SHAP visualization the lip corners and the nose were identified as indicators for the model´s prediction of 'Disgust'. Therefore the SHAP method provided a better explanation of the model´s prediction, thereby improving the model´s interpretablity.
Furthermore, to ensure the effectiveness of the kernel size of the 'blur' masker, the different named kernel sizes were evaluated by analyzing there SHAP outputs.
Identical to the FER 2013 model, the misclassifications of the model were also evaluated based on the SHAP visualization of the misclassified images.

Illustrated in Figure 40 is a misclassified instance of the RAF-DB test dataset. The

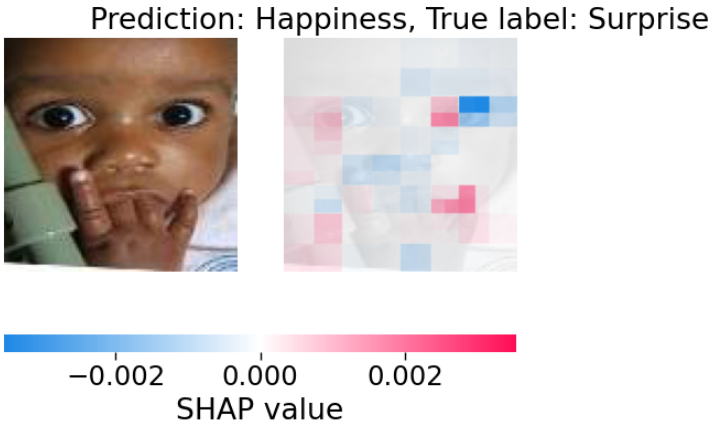

Figure 40: SHAP Output of a falsely predicted image

RAF-DB model predicted the emotional class 'Happiness', however the true label of the class was 'Surprise'. The Shapley values of the image highlighted he eyes and lip corner of the face as positive contributions.
As previously mentioned the lip corner is one AU that characterizes the emotion of happiness. The image contained a hand before the lip part, therefore the model probably concentrated only on the lip corners as the mouth part could not be identified easily as it was covered. Thus, only the lip corners were highlighted for the positive contribution, indicating the emotional class 'Happiness'.
As well as that the SHAP visualization also showed that the foreign object in the image was also contributing positively to the model´s prediction. This could have potentially influenced the model´s prediction immensely. However, in this case it was hard to interpret the model´s decision-making-process as it was unclear how the object was identified by the model or if it was mistaken as an AU that characterizes the facial expression of happiness.

The Figure 41 shows three SHAP plots for the emotional class 'Happiness' with three

Figure 41: SHAP Output using different kernel sizes of the masker 'blur'

different kernel sizes: 7x7, 15x15 and 25x25. The purpose of the visualization was to evaluate the impact of the different kernel sizes on the FER model´s interpretation and therefore to determine the best kernel size.
For the kernel size of 7x7, the Shapley values ranged from -4 e^{-7} to 4 e^{-7}, indicating that the contributions were very small. However, the values still showed a significant level of detail in some facial regions.
There were distinct regions with both positive and negative contributions, suggesting that specific features in these areas had a notable impact on the model´s prediction.
The Shapley values for the 15x15 kernel ranged from approximately -0.005 to 0.005, providing a broader distribution of the contributions. The highlighted regions were less defined compared to the smaller kernel. However, the values appeared to provide a more smoothed interpretation.
For the 25x25 kernel, the Shapley values ranged from approximately -0.002 to 0.002, thereby further reducing the size of contributions. The Shapley values were even more diffused compared to the other two kernels, indicating that this kernel size offered the most generalized view of the model´s prediction.
As illustrated in Figure 41, the 7x7 and 15x15 kernel provided localized and detailed features that contributed the model´s prediction. However, in several cases of the implementation of the SHAP method using these two kernels, the SHAP visualization did not provide any Shapley values. Therefore, the 25x25 kernel was chosen to evaluate the SHAP method.
Additionally, the 25x25 kernel also provided more entail on the contribution of different facial regions, thereby providing more insights into the FER models predictions.

Overall, the SHAP method provided valuable insights into the model´s decision-making-process by aiding to identify important facial features that contributed to the model´s predictions. Therefore the interpretability of these black box models were improved by

implementing the SHAP method.

4.2.2 Local interpretable model-agnostic Explanations (LIME)

The LIME provides explanations for the individual predictions of the FER models. However, the output of LIME varies for each iteration. This is due to the fact that LIME generates these explanations by creating perturbations of the input image and observes how the model´s predictions change.
The method perturbs the images by turning of some of the image´s segments, i.e. superpixels, therefore each perturbation is a new version of the image.
Because of that, for the evaluation of the LIME method, the stability of the method was looked at first. Similar to the stability measurement of the SHAP method, mean, standard deviation and variance of the LIME method were calculated for ten iterations.
The Figure 42 shows six images with the following order:

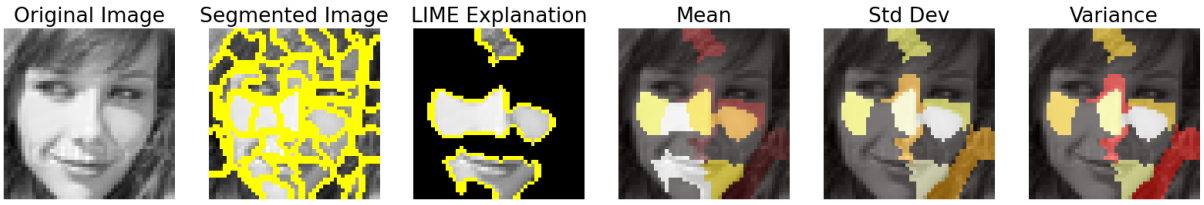

Figure 42: LIME Output for mean, standard deviation and variance

- Image of the FER 2013 test dataset to be explained
- Segmentation into superpixels
- 5 top superpixels that contribute most to the model´s prediction (LIME explanation) based on the last iteration
- Mean of the LIME calculations
- Standard deviation of the LIME calculations
- Variance of the LIME calculations

The mean image showed that certain facial regions were consistently highlighted as important across the iterations, with varying intensities. The darker red areas represented regions that had been consistently marked as important across the iterations. The white and yellow areas were also highlighted as important but with less consistency compared to the darker red areas.
For the standard deviation plot, bright yellow and white areas illustrated areas with a high standard deviation and thereby indicated areas with a high variability in the importance

score across the iterations. This suggested instability in the LIME explanations for these regions.
The darker areas indicated low variability. This suggested more stability in the LIME explanations for these regions.
The bright yellow and red areas in the variance plot indicated high variability in the importance scores across the iterations, similar to the standard deviation. This emphasized the areas of instability even more.
The darker areas illustrated low variance, which indicated low variability, suggesting more stable regions.
However, it should be noted, that the measurements were plotted based on the 5 top superpixels. Therefore, even though the LIME method showed some instability the overall conclusion was that the method was still stable enough to provide valuable explanations.
The same measurements were evaluated for the RAF-DB model, shown in Figure 43. Identical to the FER 2013 model´s LIME explanations, the Figure 43 shows the image to

Figure 43: LIME Output for mean, standard deviation and variance

be explained with its superpixels, the last LIME explanation of the iteration and the three measurements:

- Mean
- Standard deviation
- Variance

The stability of the LIME explanations of the RAF-DB dataset was similar to the stability of the FER 2013 LIME explanations. However, the RAF-DB dataset´s explanations seemed to have a lower standard deviation and lower variance overall, which indicated that they were more stable.
Using the default local interpretable model Ridge, the LIME explanations were evaluated by implementing the method with the lime libary for each emotional class.
Figure 44 shows ten triplets consisting of:

- Image of the FER 2013 test dataset to be explained
- Segmentation of the image (superpixels)

Figure 44: LIME explanations for model´s preddiction of emotional class 'Happy'

- LIME explanation based on the top 5 superpixels

Most of the LIME explanations focused on the mouth area, which was consistent with the FACS as the lip corners are a strong indicator for happiness according to the coding system (Cheong and Jolly 2022).
Many of the LIME visualizations also highlighted areas around the eyes and cheeks. As previously mentioned the AU6 (Cheek raiser) also indicates the facial expression of happiness according to the FACS.
In some cases regions of the forehead were also highlighted, indicating that this was also an important region for the model´s prediction of 'Happy'. However this does not align with the knowledge of the psychologists.
The Figure 45 also shows ten triplets for the LIME explanation of the model´s prediction

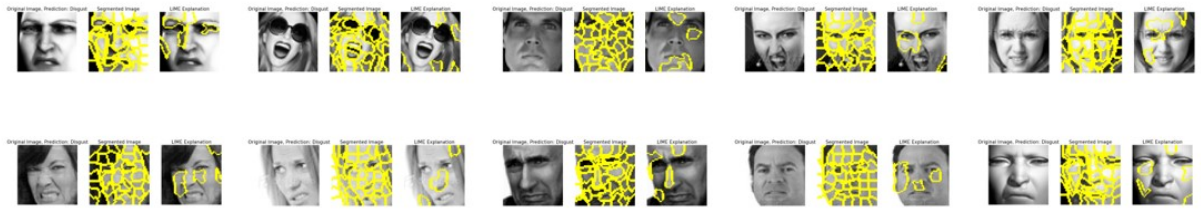

Figure 45: LIME explanations for model´s preddiction of emotional class 'Disgust'

'Disgust'. The LIME explanations frequently highlighted the nose and mouth regions. This aligned with the previous findings. Furthermore, it aligned with the FACS that states that the facial expression of disgust is often associated with nose wrinkling and some configuration of the mouth.
Several of the explanations also emphasized areas around the eyes and brows. However, the brows were more often associated with facial expressions such as fear or sadness. Therefore, this could have been an indicator for a misclassified image.
In 3 cases the LIME explanation also highlighted regions that were not part of the face, thereby making it hard to understand the underlying decision-making-process of the model.

The Figure 46 also shows the previously explained triplets for LIME explanations based

on the RAF-DB model´s predictions for the emotional class 'Happiness'.
Nearly all LIME explanations highlighted some part of the cheek, which is according to

Figure 46: LIME explanations for model´s preddiction of emotional class 'Happiness'

the FACS an indicator for the facial expression of happiness. Some of the images also highlighted parts of the mouth and eyes.
However, the lip corners, which are a characteristic of happiness, were only highlighted in 3 of the 10 images. Furthermore, similar to the explanations of the FER 2013 model, some LIME explanations highlighted regions that were not part of the face, thereby making it hard to understand the model´s decision-making-process in these cases.
However, these explanations were configured by using the ridge regression, therefore it could have been possible that this local interpretable model was not the right fit for the FER model´s predictions.
Because of that, other local interpretable models were evaluated, which is explained later on.
The Figure 47 shows the previously mentioned triplets for the emotional class 'Disgust'.
The LIME explanations frequently highlighted the nose and mouth regions, which was

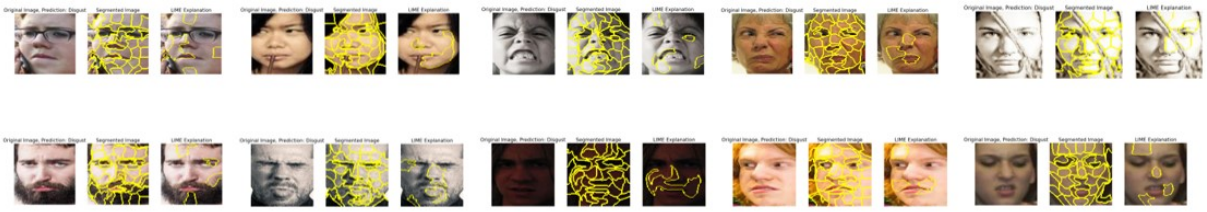

Figure 47: LIME explanations for model´s preddiction of emotional class 'Disgust'

similar to the findings of the FER 2013 model´s explanations.
Also similar to the FER 2013 model´s explanations, several instances highlighted areas around the eyes and brows.
Therefore, according to the LIME explanations, the model correctly focused on the key areas associated with the facial expression of disgust. However, the explanations contained some misclassified images and therefore the model did possibly not focus on the right facial regions for these emotions. This could have been due to the fact, that as previously explained these misclassified images contained AUs that characterize the predicted emotion.

To evaluate the different local interpretable models and thereby the LIME method further, the XAI method was implemented without the lime library and with different local interpretable models. The results of this process is shown in Figure 48. The triplets order is

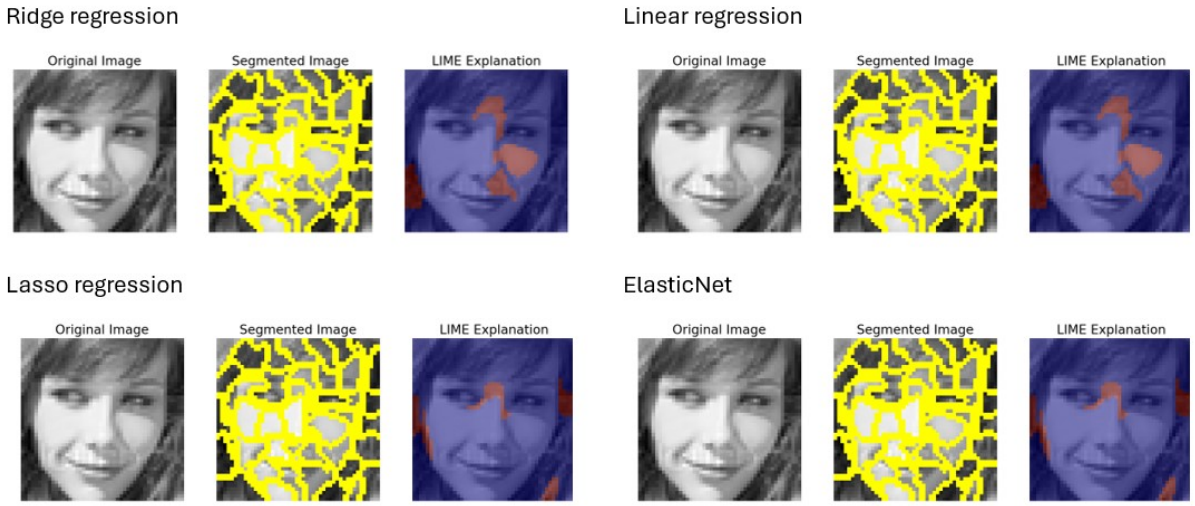

Figure 48: LIME explanations for different local interpretable models

identical to the previous ones. The LIME explanations were shown by highlighting the top contribution in red and the rest of the image in blue.
Ridge regression and linear regression highlighted the same facial features as important. These regions included the right cheek, the right upper lip or even lip corner and the area between the brows. Except for the brows, these highlights aligned with the FACS. (Cheong and Jolly 2022)
However, one area that was not part of the face was also highlighted by both local models. The lasso regression and ElasticNet also had identical highlighted facial features.However, these local models highlighted only the upper nose area and areas outside the face. Therefore, these local interpretable models were possibly not the right fit for the black box model´s predictions.

The same process was also done for the LIME explanations of the RAF-DB model as shown in Figure 49. In contrast to the results of the FER 2013 model explanations, the LIME explanations for the ridge regression and linear regression differed.
The ridge regression identified the left part of the lip and the right cheek as important facial features for the RAF-DB model´s prediction. This partially aligned with the FACS as the parts of the lip corners and cheeks were included in the explanation.
The linear regression identified part of the right cheek and a small part of the left cheek close to the nose as important facial features.
In the case of the lasso regression and ElasticNet, both local interpretable models highlighted the eye region and the left side of the chin and mouth as important facial features.

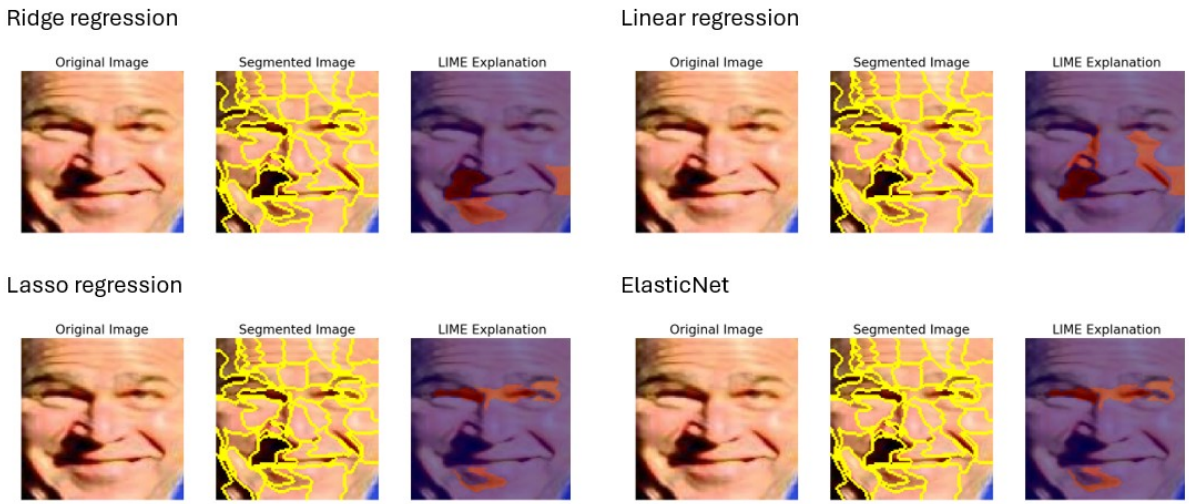

Figure 49: LIME explanations for different local interpretable models

However, the eyes are not a characterization for the facial expression of happiness.

Overall the LIME explanation provided facial features that appeared to be indicators for the respective emotional class. However, in some cases they did not align with the overall knowledge of FER.
Nevertheless, the XAI method provided visual explanations for the FER model´s predictions and therefore improved the interpretability of the FER model to some extent.

5 Discussion

5.1 Performance of the FER models

The evaluation of the FER models trained on the FER 2013 and RAF-DB datasets provides valuable insights into their performance across different emotional classes.
As previously mentioned, the FER model trained on the FER 2013 dataset achieved an overall accuracy of 59%, while the RAF-DB model performed better with an accuracy of 75%. This discrepancy suggests that the RAF-DB model is more effective at classifying emotions, potentially due to differences in the characteristics of the datasets used.
When looking at the different emotional classes, it becomes clear that both models perform exceptionally well in identifying 'Happy' instances. The FER 2013 model has a higher absolute number of correctly classified 'Happy' instances due to a larger test set size, but the RAF-DB model maintains high accuracy and fewer misclassifications overall.
The differentiation of 'Angry' from other emotional classes, in particular 'Sad', is a challenge for both models. However, the RAF-DB model shows fewer misclassifications compared to the FER 2013 model, indicating a better performance in this category.
The emotional class 'Disgust' is frequently misclassified by both models into various other emotions. The RAF-DB model demonstrates marginally better performance, but both models require improvement in this area.
Furthermore, both models have difficulty correctly identifying 'Fear', often confusing it with 'Sad' and other emotions. The RAF-DB model shows a better precision for 'Fear', indicating more confidence in its predictions despite a lower recall.
The RAF-DB model performs better in recognizing 'Sad' instances, with fewer misclassifications compared to the FER 2013 model. This indicates a more accurate understanding of 'Sad' facial expressions.
Both models effectively classify 'Surprise', but the RAF-DB model demonstrates fewer misclassifications, indicating a slightly better performance.
The RAF-DB model shows significantly better performance in classifying 'Neutral' expressions, with higher recall and fewer misclassifications compared to the FER 2013 model.

The Table 1 shows the ranks of the emotional classes based on their F1-scores for the both the RAF-DB and FER 2013 models, from highest to lowest.
With the exception of the emotional classes 'Surprise' and 'Neutral', both models exhibit identical rankings. It is therefore evident that both models are unable to identify emotions such as fear and disgust correctly.
The findings suggest that the training data may contain overlapping facial features, particularly for emotions such as 'Angry', 'Fear' and 'Disgust', which makes it challenging to categorize them as a single emotional class. Furthermore, these misclassifications indicate a need for more balanced data representations in order to ensure that the model is trained

Rank	RAF-DB Model	FER 2013 Model
1	Happy	Happy
2	Neutral	Surprise
3	Surprise	Neutral
4	Sad	Sad
5	Angry	Angry
6	Fear	Fear
7	Disgust	Disgust

Table 1: Ranking of Emotional Classes According to Accurate Classification

on all emotional classes equally.
However, these are just hypothesis based on the results of the model evaluation. In order to gain a better understanding of the limitations of the FER models, it is useful to look at more than the models predictions. Implementing the XAI methods can be helpful to better understand the missclassification of the emotional classes and to identify further processes to improve the FER models.
Even though, the performance of the model trained on the FER 2013 dataset is only moderate, it can be valuable for the evaluation of the XAI methods.

5.2 Performance of the XAI methods

The research aimed to enhance the interpretability of two FER models using XAI methods, specifically SHAP and LIME.
The SHAP method, both with and without maskers, provided detailed visualizations of the FER models decision-making-processes. For the emotion 'Happy', the Shapley values calculated for the FER 2013 model without the masker highlighted the mouth regions as crucial for predicting the emotional class. However, the presence of blue regions around the mouth and eyes suggested some confusion in these areas.
Using a blur masker in both models, the red regions were more generalized around the mouths and cheeks, aligning with known AUs such as AU06 (Cheek Raiser) and AU12 (Lip Corner Puller), which are associated with the facial expression of happiness. This alignment with psychological knowledge confirms the model´s reliance on these facial features for its predictions. (Cheong and Jolly 2022)
Similar to the emotional class 'Happiness', the emotional class of disgust was analyzed with and without a masker. The Shapley values calculated for the FER 2013 model without the masker indicated a focus on the mouth and eyes, with some negative contribution areas highlighted in blue.
The use of the masker also showed clearer red highlights around the mouth, nose and eyes in this case. This indicated that these facial areas are important for the models predictions of 'Disgust'. The alignment with AUs like AU09 (Nose Wrinkler) and AU10 (Upper Lip

Raiser) further validated these findings.
The SHAP method demonstrated stability across the iterations with larger kernel sizes such as 15x15 for the FER 2013 model and 25x25 for the RAF-DB model, providing consistent and distinct highlights around the important facial features such as the mouth and eyes.
Especially with the blur masker, the SHAP visualizations were straightforward to interpret as they clearly highlighted the important facial features. The simplicity was maintained without compromising the quality of the explanations, providing valuable insights into the models decision-making-processes.
Generating the SHAP explanations was computationally intensive but manageable.
Overall, SHAP provided stable, comprehensive and visually interpretable explanations that aligned with the domain knowledge, particularly when using larger kernel sizes.

The LIME explanations showed some variability across iterations with certain regions consistently highlighted as important. The standard deviation and variance measurements indicated areas of instability, suggesting that while LIME is useful for generating explanations, its stability can be inconsistent. Therefore, SHAP was found to be more stable than LIME.
With regard to the emotion of happiness, LIME frequently highlighted the mouth area, particularly the lip corners, which aligns with the FACS indicators. Nevertheless, some explanations also indicated that the models considered additional regions, such as the eyes and cheeks, to be important.
In the case of the emotion disgust, the LIME explanations frequently highlighted the nose and mouth regions, which is consistent with the SHAP analysis and the FACS indicators. Nevertheless, some misclassified images demonstrated regions unrelated to facial expressions, suggesting potential areas of improvement for the FER models.
The explanations provided by LIME were relatively straightforward to comprehend, displaying the superpixels that contribute the most to the model's predictions. However, the variability in the explanations could occasionally complicate the process of interpretation.
Therefore, both XAI methods provided simple and comprehensible visual explanations of the models decision-making-processes, but SHAP had a slight advantage due to its consistency.
It should also be noted that LIME highlighted critical regions but sometimes missed less obvious features due to its segmentation approach.
Furthermore, the LIME explanations were generally consistent with the domain knowledge, with a focus on the mouth and eyes for 'Happiness' and the nose and mouth for 'Disgust'. Nevertheless, some variability did result in instances of misalignment.
The efficacy of four local interpretable models (Ridge Regression, Linear Regression, Lasso Regression, and ElasticNet) was evaluated using LIME. The ridge regression and linear

regression models consistently highlighted facial features that align with the known facial AUs. However, Lasso Regression and ElasticNet frequently identified irrelevant regions, suggesting that these models may not be optimal for explaining FER predictions.
Overall, LIME demonstrated computational efficiency and reasonable visual interpretability, with the ability to highlight important facial regions. However, there was some variability and occasional misalignments.

The utilization of SHAP and LIME facilitates the transparency of FER models, thereby enhancing the comprehensibility of their decision-making processes. This can enhance user confidence in AI-driven emotion recognition systems. Furthermore, the alignment of SHAP explanations with domain knowledge serves to validate the model´s reliance on appropriate facial features for emotion recognition, thereby ensuring that the model´s predictions are based on scientifically recognized indicators.

However, the findings are based on two specific datasets, specifically FER 2013 and RAF-DB. Therefore, these findings might not generalize to other datasets or real-world scenarios without further validation.

6 Conclusion and Outlook

This thesis investigated the potential of XAI methods, specifically SHAP and LIME, to improve the interpretability of FER models.
The research primarily focused on models trained on two datasets:

- FER 2013
- RAF-DB

The research process involved training FER models on these two datasets. The models were evaluated based on their performance using standard metrics, including accuracy, precision, recall, and F1-score.
Following this, the SHAP and LIME XAI methods were employed to generate explanations for the models' predictions.
The findings indicate that both SHAP and LIME significantly contribute to the transparency of FER models, enabling a better understanding of the model´s decision-making processes.

The SHAP method provided detailed visualizations of the FER models' decision-making processes, which highlighted critical facial regions such as the mouth, eyes, and cheeks. These findings align with known facial AUs for the specific facial emotion.
The method demonstrated consistency and stability. The blur mask and a larger kernel size enhanced the interpretability of the results by generalizing the highlighted regions, thereby confirming the model's reliance on scientifically recognized facial features.
The LIME method provided straightforward and comprehensible visual explanations by identifying the superpixels that contributed most to the model's predictions. Nevertheless, LIME exhibited some variability in its explanations across iterations, suggesting occasional instability.
Despite this limitation, LIME consistently highlighted important facial regions, aligning with domain knowledge, although it sometimes identified additional or irrelevant regions.

In conclusion, both XAI methods were effective in improving the interpretability of FER models. However, SHAP demonstrated a slight advantage due to its stability and comprehensive explanations.

The findings of this research pave the way for several future directions in the development and application of FER models.
As previously mentioned, the current findings are based on two specific datasets. Future research should involve applying LIME and SHAP to a broader range of datasets to validate the generalizability of these findings.

Furthermore, the findings could be used to improve the XAI methods. For instance the implementation of LIME without a library could be evaluated closer, using different parameters each time as well as integrating even more local interpretable models.
In addition, combining SHAP and LIME with other XAI methods such as Grad-CAM, Guided Grad-CAM and GAIN could provide more comprehensive insights on the models decision-making-processes.
As well as that, the explanations of both XAI methods could be evaluated further by focusing even more on the false positive and false negative, to gain a deeper understanding of why the model misclassified these particular instances. This evaluation could be combined with the AU detection. For this, either the py-feat library could be used or an AU detection model could be trained. The detector of the py-feat library already provided valuable insights, however the method could be further evaluated in order to receive better outputs, that identify the exact facial landmarks and therefore the exact AUs.
Addressing the ethical considerations related to privacy, bias and accuracy still remains crucial. Future research should therefore focus on developing FER models that are not only interpretable but also fair and reliable across diverse demographic groups.

In summary, the research demonstrates the significant potential of XAI methods, such as LIME and SHAP, to improve the interpretability of FER models. Through continued research, these techniques can achieve broader acceptance and trust, ultimately enhancing their applicability in various real-world applications.

References

Adhi Tama, Bayu et al. (Mar. 2022). "An EfficientNet-Based Weighted Ensemble Model for Industrial Machine Malfunction Detection Using Acoustic Signals". In: *IEEE Access* 10, pp. 34625–34636. DOI: 10.1109/ACCESS.2022.3160179.

Albiero, Vitor et al. (2020). "img2pose: Face Alignment and Detection via 6DoF, Face Pose Estimation". In: *CoRR* abs/2012.07791. arXiv: 2012.07791. URL: https://arxiv.org/abs/2012.07791.

Badillo, Solveig et al. (2020). "An introduction to machine learning". In: *Clinical pharmacology & therapeutics* 107.4, pp. 871–885.

Bobojanov, Sukhrob et al. (2023). "Comparative Analysis of Vision Transformer Models for Facial Emotion Recognition Using Augmented Balanced Datasets". In: *Applied Sciences* 13.22. ISSN: 2076-3417. DOI: 10.3390/app132212271. URL: https://www.mdpi.com/2076-3417/13/22/12271.

Cabitza, Federico, Andrea Campagner, and Martina Mattioli (2022). "The unbearable (technical) unreliability of automated facial emotion recognition". In: *Big Data & Society* 9.2, p. 20539517221129549. DOI: 10.1177/20539517221129549. eprint: https://doi.org/10.1177/20539517221129549. URL: https://doi.org/10.1177/20539517221129549.

Chen, Sheng et al. (2018). "MobileFaceNets: Efficient CNNs for Accurate Real-time Face Verification on Mobile Devices". In: *CoRR* abs/1804.07573. arXiv: 1804.07573. URL: http://arxiv.org/abs/1804.07573.

Chen, Shikai et al. (June 2020). "Label Distribution Learning on Auxiliary Label Space Graphs for Facial Expression Recognition". In.

Cheong, Jin Hyun and Eshin Jolly (2022). *Detecting facial expressions from images*. https://py-feat.org/basic_tutorials/02_detector_imgs.html. Accessed: 2024-06-02.

Chollet, François (2016). "Building powerful image classification models using very little data". In: Accessed: 2024-05-25. URL: https://blog.keras.io/building-powerful-image-classification-models-using-very-little-data.html.

Deng, Jiankang et al. (2019). "RetinaFace: Single-stage Dense Face Localisation in the Wild". In: *CoRR* abs/1905.00641. arXiv: 1905.00641. URL: http://arxiv.org/abs/1905.00641.

Ekman, Paul (2013). "Basic Emotions". In: URL: https://www.paulekman.com/wp-content/uploads/2013/07/Basic-Emotions.pdf.

Ekman, Paul and Wallace V Friesen (1971). "Constants across cultures in the face and emotion". In: *Journal of Personality and Social Psychology* 17.2, pp. 124–129. DOI: 10.1037/h0030377. URL: https://doi.org/10.1037/h0030377.

Farnsworth, Bryn (2022). "Facial Action Coding System (FACS) - A Visual Guidebook". In: Accessed: 2024-05-28.

Gao, Bo (Feb. 2022). "Application of Convolutional Neural Network in Emotion Recognition of Ideological and Political Teachers in Colleges and Universities". In: *Scientific Programming* 2022. DOI: 10.1155/2022/4667677.

Gebele, Jens, Philipp Brune, and Stefan Faußer (2022). "Face Value: On the Impact of Annotation (In-)Consistencies and Label Ambiguity in Facial Data on Emotion Recognition". In: pp. 2597–2604. DOI: 10.1109/ICPR56361.2022.9956230.

Goodfellow, Ian J. et al. (2013). "Challenges in Representation Learning: A report on three machine learning contests". In: arXiv: 1307.0414 [stat.ML].

Guidotti, Riccardo et al. (2018). "A Survey Of Methods For Explaining Black Box Models". In: *CoRR* abs/1802.01933. arXiv: 1802.01933. URL: http://arxiv.org/abs/1802.01933.

Gultekin, Gozde et al. (Dec. 2016). "Facial emotion recognition ability: psychiatry nurses versus nurses from other departments". In: *Clinical and investigative medicine. Médecine clinique et experimentale* 39, p. 27503. DOI: 10.25011/cim.v39i6.27503.

Gunning, David et al. (Dec. 2019). "XAI—Explainable artificial intelligence". In: *Science Robotics* 4, eaay7120. DOI: 10.1126/scirobotics.aay7120.

He, Kaiming et al. (2016). "Deep Residual Learning for Image Recognition". In: pp. 770–778. DOI: 10.1109/CVPR.2016.90.

Joshi, Pranjali P. and Anant M. Bagade (2023). "Explainable Artificial Intelligence Techniques for Image Classification Models in Diverse Domains". In: pp. 1–7. DOI: 10.1109/ICCUBEA58933.2023.10392188.

Khaireddin, Yousif and Zhuo Liang Chen (2021). "Facial Emotion Recognition: State of the Art Performance on FER2013". In: *ArXiv* abs/2105.03588. URL: https://api.semanticscholar.org/CorpusID:234334063.

Kim, Doo Yon and Christian Wallraven (2021). "Label quality in AffectNet: results of crowd-based re-annotation". In: *CoRR* abs/2110.04476. arXiv: 2110.04476. URL: https://arxiv.org/abs/2110.04476.

Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E Hinton (2012). "ImageNet Classification with Deep Convolutional Neural Networks". In: 25. Ed. by F. Pereira et al. URL: https://proceedings.neurips.cc/paper_files/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf.

Kumar, K Pavan and Y Shankar Reddy (n.d.). "FACIAL EMOTION RECOGNITION USING MACHINE LEARNING". In: ().

Kuttenreich, Anna-Maria, Harry Piekartz, and Stefan Heim (July 2022). "Is There a Difference in Facial Emotion Recognition after Stroke with vs. without Central Facial Paresis?" In: *Diagnostics* 12, p. 1721. DOI: 10.3390/diagnostics12071721.

Li, Kunpeng et al. (2018). "Tell Me Where to Look: Guided Attention Inference Network". In: *CoRR* abs/1802.10171. arXiv: 1802.10171. URL: http://arxiv.org/abs/1802.10171.

Li, Shan and Weihong Deng (2018). "Deep Facial Expression Recognition: A Survey". In: *CoRR* abs/1804.08348. arXiv: 1804.08348. URL: http://arxiv.org/abs/1804.08348.

– (2022). "Deep Facial Expression Recognition: A Survey". In: *IEEE Transactions on Affective Computing* 13.3, pp. 1195–1215. DOI: 10.1109/TAFFC.2020.2981446.

Li, Shan, Weihong Deng, and JunPing Du (2017). "Reliable Crowdsourcing and Deep Locality-Preserving Learning for Expression Recognition in the Wild". In: pp. 2584–2593. DOI: 10.1109/CVPR.2017.277.

Linardatos, Pantelis, Vasilis Papastefanopoulos, and Sotiris Kotsiantis (2021). "Explainable AI: A Review of Machine Learning Interpretability Methods". In: *Entropy* 23.1. ISSN: 1099-4300. DOI: 10.3390/e23010018. URL: https://www.mdpi.com/1099-4300/23/1/18.

Lundberg, Scott (2018). "SHAP (SHapley Additive exPlanations)". In: Documentation available at https://shap.readthedocs.io/en/latest/.

Lundberg, Scott M. and Su-In Lee (2017). "A unified approach to interpreting model predictions". In: *CoRR* abs/1705.07874. arXiv: 1705.07874. URL: http://arxiv.org/abs/1705.07874.

Luo, Siwen et al. (2021). "Local Interpretations for Explainable Natural Language Processing: A Survey". In: *CoRR* abs/2103.11072. arXiv: 2103.11072. URL: https://arxiv.org/abs/2103.11072.

Ma, Fuyan, Bin Sun, and Shutao Li (Apr. 2023). "Facial Expression Recognition With Visual Transformers and Attentional Selective Fusion". In: *IEEE Transactions on Affective Computing* 14.2, pp. 1236–1248. ISSN: 2371-9850. DOI: 10.1109/taffc.2021.3122146. URL: http://dx.doi.org/10.1109/TAFFC.2021.3122146.

Matsumoto, D. and P. Ekman (2008). "Facial expression analysis". In: *Scholarpedia* 3.5. revision #88993, p. 4237. DOI: 10.4249/scholarpedia.4237.

Mollahosseini, Ali, Behzad Hassani, and Mohammad H. Mahoor (2017). "AffectNet: A Database for Facial Expression, Valence, and Arousal Computing in the Wild". In: *CoRR* abs/1708.03985. arXiv: 1708.03985. URL: http://arxiv.org/abs/1708.03985.

O'Shea, Keiron and Ryan Nash (2015). "An Introduction to Convolutional Neural Networks". In: *CoRR* abs/1511.08458. arXiv: 1511.08458. URL: http://arxiv.org/abs/1511.08458.

Pedregosa, F. et al. (2011). "scikit-learn: Machine Learning in Python". In: Accessed: 2024-05-27.

Perrett, David (2022). "Representations of facial expressions since Darwin". In: *Evolutionary Human Sciences* 4, e22. DOI: 10.1017/ehs.2022.10.

Rafiq, Musaib et al. (2022). "Efficient Implementation of Max-Pooling Algorithm Exploiting History-Effect in Ferroelectric-FinFETs". In: *IEEE Transactions on Electron Devices* 69.11, pp. 6446–6452. DOI: 10.1109/TED.2022.3207114.

Ribeiro, Marco Tulio, Sameer Singh, and Carlos Guestrin (2016). "LIME: Local Interpretable Model-agnostic Explanations". In: Accessed: 2024-06-01. URL: https://lime-ml.readthedocs.io/en/latest/.

Ribeiro, Marco Túlio, Sameer Singh, and Carlos Guestrin (2016). ""Why Should I Trust You?": Explaining the Predictions of Any Classifier". In: *CoRR* abs/1602.04938. arXiv: 1602.04938. URL: http://arxiv.org/abs/1602.04938.

Rosenfeld, Avi (2021). "Better Metrics for Evaluating Explainable Artificial Intelligence". In: AAMAS '21, pp. 45–50.

Roth, Alvin E (1988). "Introduction to the Shapley value". In: *The Shapley value*, pp. 1–27.

Sahoo, Goutam Kumar et al. (2022). "Deep Leaning-Based Facial Expression Recognition in FER2013 Database: An in-Vehicle Application". In: pp. 1–6. DOI: 10.1109/INDICON56171.2022.10040121.

Selvaraju, Ramprasaath R. et al. (2016). "Grad-CAM: Why did you say that? Visual Explanations from Deep Networks via Gradient-based Localization". In: *CoRR* abs/1610.02391. arXiv: 1610.02391. URL: http://arxiv.org/abs/1610.02391.

Shingjergji, Krist et al. (2022). "Interpretable Explainability in Facial Emotion Recognition and Gamification for Data Collection". In: *2022 10th International Conference on Affective Computing and Intelligent Interaction (ACII)*, pp. 1–8. URL: https://api.semanticscholar.org/CorpusID:253420469.

SuperAnnotate (2023). "What is image classification? Basics you need to know". In: Accessed: 2024-05-28.

Wadhawan, Rohan and Tapan K. Gandhi (2021). "Landmark-Aware and Part-based Ensemble Transfer Learning Network for Facial Expression Recognition from Static images". In: *ArXiv* abs/2104.11274. URL: https://api.semanticscholar.org/CorpusID:233387977.

Wang, H. and W. Deng (2024). "RAF-DB: The Real-world Affective Faces Database". In: Accessed: 2024-01-21. URL: http://www.whdeng.cn/raf/model1.html#dataset.

Wang, Yingying et al. (2020). "The Influence of the Activation Function in a Convolution Neural Network Model of Facial Expression Recognition". In: *Applied Sciences* 10.5. ISSN: 2076-3417. DOI: 10.3390/app10051897. URL: https://www.mdpi.com/2076-3417/10/5/1897.

Weitz, Katharina et al. (June 2019). "Deep-learned faces of pain and emotions: Elucidating the differences of facial expressions with the help of explainable AI methods". In: *tm - Technisches Messen* 86. DOI: 10.1515/teme-2019-0024.

Wu, Jianxin, Chen-Wei Xie, and Jian-Hao Luo (2016). "Dense CNN Learning with Equivalent Mappings". In: *ArXiv* abs/1605.07251. URL: https://api.semanticscholar.org/CorpusID:18198362.

Yang, Jiannan et al. (2019). "Facial Expression Recognition Based on Facial Action Unit". In: pp. 1–6. DOI: 10.1109/IGSC48788.2019.8957163.

7 Appendix

The neccessary jupyter notebooks and image data is provided seperately.