TU WIEN Informatics

Robo-Smile

Entwicklung eines Gesichtsausdruck-Feedback-Systems für eine Emotions-Lernplattform für Kinder mit ASD

DIPLOMARBEIT

zur Erlangung des akademischen Grades

Diplom-Ingenieurin

im Rahmen des Studiums

Biomedical Engineering

eingereicht von

Nicole Melanie Weinert, BSc
Matrikelnummer 01425190

an der Fakultät für Informatik

der Technischen Universität Wien

Betreuung: Privatdoz. Dipl.-Ing. Dr.techn. Martin Kampel
Mitwirkung: Senior Lecturer Dipl.-Ing. Dr.techn. Michael Reiter

Wien, 9. März 2021

Nicole Melanie Weinert	Martin Kampel

Technische Universität Wien
A-1040 Wien ▪ Karlsplatz 13 ▪ Tel. +43-1-58801-0 ▪ www.tuwien.at

TU WIEN Informatics

Robo-Smile

Development of a Facial Expression Feedback System for an Emotion Learning Platform for Children with ASD

DIPLOMA THESIS

submitted in partial fulfillment of the requirements for the degree of

Diplom-Ingenieurin

in

Biomedical Engineering

by

Nicole Melanie Weinert, BSc
Registration Number 01425190

to the Faculty of Informatics

at the TU Wien

Advisor: Privatdoz. Dipl.-Ing. Dr.techn. Martin Kampel
Assistance: Senior Lecturer Dipl.-Ing. Dr.techn. Michael Reiter

Vienna, 9th March, 2021

Nicole Melanie Weinert	Martin Kampel

Technische Universität Wien
A-1040 Wien ▪ Karlsplatz 13 ▪ Tel. +43-1-58801-0 ▪ www.tuwien.at

Erklärung zur Verfassung der Arbeit

Nicole Melanie Weinert, BSc

Hiermit erkläre ich, dass ich diese Arbeit selbständig verfasst habe, dass ich die verwendeten Quellen und Hilfsmittel vollständig angegeben habe und dass ich die Stellen der Arbeit – einschließlich Tabellen, Karten und Abbildungen –, die anderen Werken oder dem Internet im Wortlaut oder dem Sinn nach entnommen sind, auf jeden Fall unter Angabe der Quelle als Entlehnung kenntlich gemacht habe.

Wien, 9. März 2021

Nicole Melanie Weinert

Danksagung

Mein aufrichtiger Dank geht an Privatdoz. Dipl.-Ing. Dr.techn. Martin Kampel, der die Betreuung meiner Arbeit übernommen hat und mich mit seinem hilfreichen Feedback unterstützt hat. Auch möchte ich mich bei Dipl.-Ing. Dr.techn. Michael Reiter bedanken, der mich im Zuge der Einreichung dieser Arbeit betreut hat. Weiters möchte ich mich bei Dipl.-Ing. Michael Platzer bedanken, der mich bei der Konzeption, Umsetzung und Erstellung dieser Arbeit unterstützt und ermutigt hat.

Diese Arbeit wäre ohne meine vorangegangene Bachelorarbeit nicht möglich gewesen, daher bin ich meinem ehemaligen Betreuer FH-Prof Dr. Andreas Drauschke aufrichtig dankbar, der unter anderem die Zusammenarbeit mit der Magistratsabteilung 10 (MA 10) ermöglicht hat. Darüber hinaus bin ich Mag Elvira Muchitsch und Frau Katharina Biebl dankbar, die mich bei der Gestaltung des pädagogischen Konzepts mit Fachwissen auf den Gebieten der Autismus-Spektrum-Störung (ASS) und des Sozialverhaltenstrainings unterstützt haben.

Acknowledgements

My sincere thanks go to Privatdozent Dipl.-Ing. Dr.techn. Martin Kampel, who took on my work's supervision and supported me with his helpful feedback. I also want to extend my gratitude to Dipl.-Ing. Dr.techn. Michael Reiter for supervising me during the submission of this work's proposal. Furthermore, I want to thank Dipl.-Ing. Michael Platzer for providing me with support, encouragement, and feedback throughout the design, implementation, and writing of this thesis.

This thesis would not have been possible without prior work conducted in the course of my bachelor thesis. Thus I am sincerely grateful to my former advisor FH-Prof. Dr. Andreas Drauschke for making the cooperation with the MA 10 possible. Furthermore, I am thankful to Mag. Elvira Muchitsch and Ms. Katharina Biebl, who provided expertise in the field of Autism Spectrum Disorder (ASD) and social behavior training during the design of the pedagogical concept.

Kurzfassung

Bei einem von vierundfünfzig Kindern wird eine Autismus-Spektrum-Störung (ASS) diagnostiziert. ASS ist eine neurologische Entwicklungsstörung, die mit sozialen Kommunikationsproblemen, Schwierigkeiten bei der sozialen Interaktion, sowie eingeschränkten und sich wiederholenden Verhaltensmustern einhergeht. Probleme im Sozialverhalten sind mit Schwierigkeiten bei der Erkennung von Emotionen in Gesichtsausdrücken anderer und dem Ausdrücken von Emotionen in der eigenen Mimik verbunden. Diese Arbeit beschreibt die Implementierung eines computergestützten Emotions-Feedback-Systems als Teil einer Lernplattform zur Förderung des Sozialverhaltens, mit der das Erkennen und Nachahmen von Emotionen in Gesichtsausdrücken trainiert wird. Das Spiel basiert auf einer früheren Arbeit zur Entwicklung eines pädagogischen Konzepts und Prototyps für ein computergestütztes Emotionslernspiel, welches auf die neurologischen Bedürfnisse von Kindern mit ASS zugeschnitten ist. Diese Arbeit befasst sich mit der Entwicklung einer Webcam-gestützten Emotionserkennung, welche kontinuierliches Feedback über die ausgedrückte Emotion in der Mimik des Kindes ermöglicht. Um den Kontroll-Algorithmus optimal auf die neurologischen Bedürfnisse von Kindern mit ASS zuzuschneiden, basiert die Emotionserkennung ausschließlich auf zuvor detektierten Orientierungspunkten im Gesicht. Darüber hinaus sind Limitierungen der Rechenzeit des Feedback-Systems essenziell, da die Erkennung ohne Verzögerung oder Pausieren des Webcam-Streams auf älteren oder schwächeren Computern laufen sollte, um das System für eine Vielzahl an Eltern, Betreuerinnen und Betreuern verfügbar zu machen. Da die Zielgruppe der Plattform Kinder sind, muss die Erkennung in der Lage sein, genaue Ergebnisse zu liefern, selbst wenn die Spielerin oder der Spieler unruhig ist, sich dreht oder bewegt. Daher wird spezieller Wert darauf gelegt, dass die Gesichtspunkte- und Emotionserkennung gegenüber einer Vielzahl von Gesichtsausdrücken und Posen robust sind. Das Ergebnis ist ein neuartiges, deep learning basierendes Feedback-System, welches anhand von Orientierungspunkten im Gesicht Live-Feedback zu ausgedrückten Emotionen geben kann. Die Arbeit präsentiert einen state-of-the-art Gesichtspunkt-Detektor, der besonders stabile und akkurate Ergebnisse bei großen Posenvariationen liefert und dennoch eine ausreichend kurze Ausführungszeit hat, um ein Live-Feedback basierend auf der Webcam-Übertragung zu ermöglichen. Darüber hinaus wird eine Emotionserkennung vorgestellt, bei der nur Gesichtsmarkierungen zur Klassifizierung verwendet werden.

Abstract

One in fifty-four children is diagnosed with Autism Spectrum Disorder (ASD). ASD is a neurodevelopmental disorder associated with impaired social communication and social interaction, as well as restricted and repetitive patterns of behavior. Difficulties in social behavior are linked to struggles in reading emotions in facial expressions of others and expressing feelings via facial expressions. This thesis provides an implementation of a computer-based emotion-feedback system conceptualized to be part of a learning platform for promoting social behavior by training to recognize and mimic emotions in facial expressions. The game is based on previous work on developing a pedagogical concept and prototype for a computer-based emotion-learning game fitted to the neurological needs of children with ASD. This thesis covers the development of a webcam-based emotion detection for continuous feedback of expressed emotions in the child's mimic. The emotion detection is based on facial landmarks to optimally be tailored to the neurological needs of children with ASD. Furthermore, the detection considers computational restrictions since it needs to run on webcam streams of older or weaker computers without delay or pauses to make the system accessible to parents and caregivers. As the platform's target audience are children, the detection must deliver accurate results even when they are restless, turn, or move. Thus, the emphasis is set on the facial landmark detection and the emotion recognition to be robust to various facial expressions and poses. The methods present a novel deep-learning-based facial expression feedback system capable of giving live feedback on expressed emotions. The algorithm contains a state-of-the-art facial landmark detector performing accurate detection during large pose variations with execution times allowing live webcam-based feedback, and an emotion detection using only facial landmarks as detector input.

Contents

CHAPTER 1

Introduction

1.1 Motivation and Application

One in fifty-four children is diagnosed with Autism Spectrum Disorder (ASD) [1]. ASD is a neurodevelopmental disorder associated with impaired social communication and social interaction, as well as restricted and repetitive patterns of behavior [2]. The tendency of repetition is further expressed through the preference of keeping up routines and firm procedures as well as a high sensitivity to change [2]. Difficulties in social behavior are linked to struggles in reading emotions in facial expressions of others and expressing feelings via facial expressions [3]. This difficulty in nonverbal and social skills are related to decreased social participation, leading to lower peer acceptance and atypical relationships [3].

ASD is characterized by deficits in face perception, which can lead to difficulties identifying information conveyed by faces such as identity, expressed emotion, and gaze direction [4]. Reasons for this are explained by atypical patterns of brain activity during social stimuli processing. The characterization of social stimuli processing is split into two main areas being social cognition and social motivation [4].

- **Social cognition:** Social cognition describes cognitive activities associated with the perception, understanding, and usage of cues communicating emotions and interpersonal information [5]. In ASD deficits in social cognition are argued to be linked to atypical eye-gaze processing and struggles of understanding the reasons behind emotional expressions of others [4].
- **Social motivation:** Social motivation includes social orienting (draw attention to socially relevant cues), motivation (seeking social cues and finding it rewarding [6]), and social maintaining (desire to maintain social bonds over sustained periods of time [6]) [4]. Difficulties in social motivation are implicated by drawing attention

to the face (in contrast to the background) and absent sustained attention to facial social cues.

1.1.1 Teaching facial expressions to children with ASD

Teaching facial expressions to children with ASD is essential and part of behavioral therapy. For example, in Austria, the focus of the therapy of children with ASD lies in the training of communication and social skills, the development of perceptions, the expansion of competences, and the treatment of secondary behavioral issues[1].

Asides from learning in therapy and school, there are various learning games available teaching children with ASD to recognize emotions. Grossard et al. review 31 serious games focusing on the promotion of social behavior of children with ASD [7]. Serious games are not designed for the sole purpose of entertainment, but for example aim to educate, convey information, or explore [8]. Of these 31 games, 16 explicitly targeted the recognition of emotions in images, drawings, audio, and video recordings. Only one game called LifeIsGame includes exercises promoting emotion production by letting the child match a shown facial expression via sketching on a canvas [9] [7].

The serious games CopyMe [10] and educative multimodal game for emotional imitation (JEMImE) [11] promote the production of facial expressions by additionally giving video-based feedback. CopyMe lets the player mimic the expressions in a given picture. The targeted emotions are happiness, sadness, anger, surprise, fear, and disgust, which are progressively introduced based on the game's difficulty level. JEMImE focuses on the imitation of the emotions happiness, anger, and sadness. The game is split into training and playing phases. During training, the child either has to mimic an avatar or has to express a specific emotion. During the playing phase, the child navigates an avatar in a virtual world. The child gets confronted with specific social scenarios and has to produce the expected facial expression fitted to the scenario.

1.1.2 Previous work: Robo-Smile

In an attempt to develop a serious emotion-learning game for children with ASD, prior work was conducted in cooperation with the Magistratsabteilung 10 (MA 10) and the special kindergarten (a kindergarten catering for children with special needs) Sobieskigasse 31 located in Vienna. The goal was to develop the pedagogical concept of the emotion learning game Robo-Smile [12] and a first prototype.

Robo-Smile stands out from other serious games, as it not only targets the recognition but also the imitation of facial expressions, whilst emphasizing specific patterns in the face, which aims to make the learning process more fitted to the neurological needs of children with ASD. As it currently does not include scenarios but instead focuses on the

[1] Autistenhilfe, "Therapie." [Online]. Available: https://www.autistenhilfe.at/autismus/therapie/. Accessed: 22 June 2020.

facial expressions alone, the target audience are children aged 8 to 16 years with ASD or children with severer struggles of distinguishing between and imitating facial expressions.

1.1.2.1 Hypotheses behind Robo-Smile

The main two hypotheses Robo-Smile is based upon are that children with ASD tend to learn to express and recognize emotions easier when linked to specific patterns in the face and that gamification and pattern-based feedback helps to improve learning.

Linking facial expressions to specific patterns in the face

- ASD is associated with atypical eye-tracking patterns during the perception of faces [4]. Facial areas containing the socially relevant features (eyes, nose, and mouth) are significantly less viewed compared to non-significant parts, thus impeding the detection of facial expressions [13]. By marking and abstracting socially-relevant facial features, the attention is diverted to the mouth and eye area, which is hypothesized to help detect emotions in facial expressions.
- The learning strategy utilizes the preference of keeping up routines and firm procedures [2] by linking the task of detecting and imitating emotions in facial expressions to specific and constant patterns.

Gamification and pattern-based feedback

- The usage of computer-aided systems for teaching social cues to children with ASD is recommended due to its predictability and the absence of struggles caused by pragmatics and anxiety due to human interaction [14].
- Viewing facial expressions and receiving feedback has shown to improve the perception of facial expressions in individuals with ASD. The improvement correlates with increased activity in the extended face perception network in areas linked to visual and attention processing [15].

The learning game aims to teach children to detect happiness, sadness, disgust, anger, fear, surprise, and neutral expressions by linking the emotion to specific patterns in the face.

1.1.2.2 Gameplay of Robo-Smile

The learning platform consists of two main games, aiming to teach the recognition and imitation of emotion in facial expressions.

- **Recognition task:** During the first game (see Figure 1.1), the child learns to pay attention to the main patterns in the face used to express emotions and recognize

the displayed feeling. In the course of this, the child is shown images, where either the drawn game character or a human expresses one of seven emotions. The game character (see Figure 1.2) is equipped with the basic facial features needed for expressing the Action Units (AUs) necessary for mimicking emotions. When photos of people are shown, the same patterns emphasized in the drawing can be marked in the picture to make it easier for the child to process and detect the given emotion.

- **Imitation task:** The second game (see Figure 1.3) focuses on imitating shown emotions. The imitation task uses the same image set as the recognition task. During the imitation, the child gets continuous feedback on which emotion it currently expresses. Furthermore, the same facial features, which are emphasized by the game character and can be marked in the photographs, can be overlayed onto the webcam stream of the visualized feedback. The emotion detection is based on the detected facial landmarks to make the learning method transparent with the learning platform's feedback.

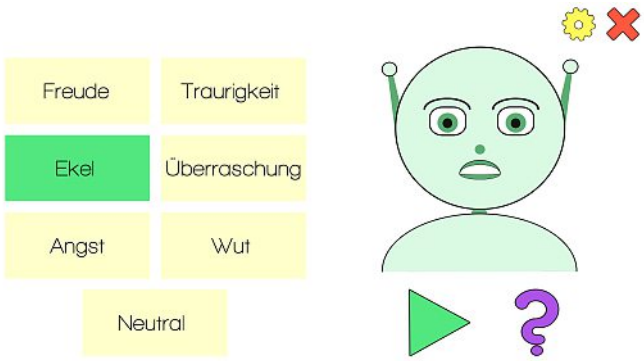


Figure 1.1: Graphical user Interface Robo-Smile: Recognition task where the user needs to detect which emotion is expressed in the shown image.

To make the game adjustable to the child's needs and preferences, the game character can look like a robot with bright colors and pointy ears or human-like with a variety of skin colors. Furthermore, which emotions, character variations, and photographs are shown during the given task can be selected in the settings to regulate the difficulty of the game.

1.1.2.3 Limitations

The prototype's current control mechanism is based on Dlib's [16] implementation of Kazemi and Sullivan's proposed one millisecond face alignment [17]. Based on the extracted locations of facial landmarks on the video stream, a Support Vector Machine (SVM) is used for classifying the emotions happiness, surprise, disgust, sadness, fear, anger, and neutral using Facial Action Coding System (FACS) inspired distances in the face. The implementation proves the feasibility of the gaming concept. However, it

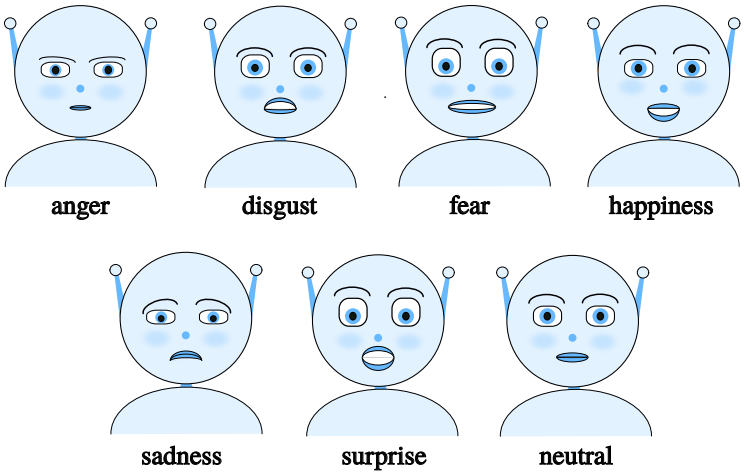


Figure 1.2: Main character of the game during the expression of the six basic emotions and neutral.

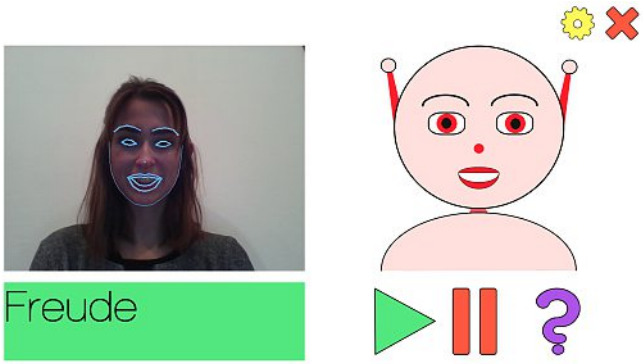


Figure 1.3: Graphical user Interface Robo-Smile: Imitation task where the user mimics the emotion expressed by the game character.

lacks the precision of detected facial landmarks during large pose variations and partial illumination conditions. In particular, the current mechanism fails to correctly detect facial landmarks during challenging conditions such as partly covered faces (e.g. glasses, beards), bad lighting conditions (weak lighting, the light source behind or to the left/right oft the person), and non-frontal face views. Since the pedagogical concept requires that the emotion classification is based on distances derived by the detected facial landmarks, high failure rates during the facial landmark detection step also lead to a high failure rate during the emotion classification. This makes continuous feedback impossible. However, it is needed for properly using the learning system. Thus the improvement of the detection accuracy and the system's overall robustness to a large variety of facial expressions and poses is necessary to facilitate the best possible gaming experience.

1.2 Aim of the Work and Research Questions

This thesis's focus lies in the redesign of the technological implementation of Robo-Smile's control mechanism, as the current mechanism fails to correctly detect facial landmarks and emotions during challenging conditions such as partly covered faces, bad lighting conditions, and side profiles. This thesis emphasizes on the improvement of the detection of facial expressions from live video transmissions to be used as a control mechanism for Robo-Smile to allow future studies, which are not part of this thesis, regarding the usability and the clinical relevance of the game. Similar to the previous control mechanism, the methodology developed in this thesis consists of two main parts to be compliant with Robo-Smiles's pedagogical concept.

- **Facial Landmark detection:** To emphasize specific areas in the face associated with conveying information necessary for recognizing emotion, the facial contours of the facial features such as eyebrows, eyes, nose, and mouth need to be marked. By marking those key-points and connecting them to form abstractions of eyes, nose, and mouth in the received video feedback, it is tried to draw the child's attention towards those regions. The regions are highlighted using facial landmarks, which need to be detected automatically by the control mechanism. As the platform's target audience are children, the detection must be capable of delivering accurate results even when the participant is restless, turns, or moves. Thus, the facial landmark detection needs to be robust to a large variety of poses, meaning that the detector must not fail during head rotations up to $\pm 90°$.

- **Emotion detection:** The learning platform focuses on teaching children the basic six emotions happiness, sadness, disgust, fear, anger, and surprise, as well as the detection of neutral faces. To give continuous feedback on the expressed emotion, an emotion detector capable of distinguishing those seven different states must be developed. Due to Robo-Smile's pedagogical concept, which suggests the linking of patterns with facial landmarks, the emotion classifier must operate on the coordinates of facial landmarks and information derived from these locations, such as distances.

- **Application considerations:** The emotion feedback system must continuously mark the main facial features and detect the expressed emotion on webcam stream data without noticeable delay and pauses. The human visual system needs one-fifteenth of a second [18], which corresponds to approximately 66.7 ms, to process one image. If a second image needs to be processed during these 66.7 ms, the images are perceived as continuous, thus allowing the impression of fluent movement. Thus, the chosen methods for detecting facial landmarks and classifying the emotion must, in sum, have execution times, which allow frame-wise video processing whilst guaranteeing the perception of movement continuity, requiring a minimum frame-rate of 15 frames per second (fps).

In order to make the system accessible to parents and caregivers, the detection cannot require additional expensive equipment (e.g. additional hardware such as high resolution cameras, lighting, Graphics Processing Units (GPUs)). However, as most commercially available laptops are already equipped with built-in cameras (so-called webcams), the plan is to take advantage of that fact and to develop the algorithm such that it can use these webcams. It cannot be assumed that parents or caregivers have access to expensive high performance computers equipped with GPUs, thus the system must run on webcam streams (or camera streams of similar image quality) of older or weaker computers with only a Central Processing Unit (CPU) without delay or pauses. Hence, computational restrictions are taken into consideration during the development of the control mechanism.

More specifically, this thesis focuses on solving the following research questions:

- Which combined facial landmark detector and emotion classifier can operate on a webcam stream without requiring additional computational resources (e.g., GPU) while still achieving a frame rate of minimum 15 fps?
- To achieve the best feedback accuracy, how must the system's architecture be designed to be suited for the learning platform?

1.3 My Contribution

This thesis's contribution is the development of a novel deep-learning-based facial expression feedback system capable of giving live feedback on expressed emotions. Furthermore, the resulting feedback system supports the pedagogical concept of Robo-Smile. It aims to ease the teaching of emotions to children with ASD by allowing a purely computer-aided live emotion feedback system. The game removes struggles caused by pragmatics and anxiety due to human interaction [14] and allows continuous feedback, which is linked to increased activity in the extended face perception network [15].

- **Literature overview:** A detailed literature review on the detection of facial landmarks, and the classification of emotions is given. State-of-the-art facial landmark detectors are identified, complying with the computational restrictions of using only a CPU while achieving execution times allowing live feedback. Based on this evaluation, the facial landmark detectors Convolutional Neural Network (CNN)-6 (with Wing Loss) [19], Tweaked Convolutional Neural Network (TCNN) [20], Multi-Center Learning (MCL) [21], and Task-Constrained Deep Convolutional Network (TCDCN) [22] [23] are considered suited to be used as part of Robo-Smile's feedback system. Furthermore, two main emotion classification approaches, being AU-based and purely pattern-based, are identified as promising for Robo-Smile's emotion classification.
- **Facial landmark detection:** By combining and adapting aspects of the state-of-the-art facial landmark detectors, an original algorithm is developed as part of

this work. The detector manages to achieve state-of-the-art results with a mean normalized error of 7.01, an improvement of 21.40 % over the methodology it is inspired by, and attains an average execution time of 0.0299 s (33 fps) allowing live webcam-based feedback running on CPUs only. The accurate facial landmark detection facilitates the marking of those key-points and connecting them to form abstractions of eyes, nose, and mouth overlayed onto the captured video stream, thus drawing the child's attention towards those regions.

- **Emotion detection:** In order to develop a unique pattern and facial-landmark-based emotion detector, AUs- and purely coordinate-based methodologies are designed, trained, and tested in order to find the best emotion classifier for the given facial-landmark-based input. Based on this evaluation, a new pattern and facial-landmark-based facial expression classifier is presented achieving a classification accuracy of 88.57 %. Linking specific facial landmark patterns to emotion feedback algorithms supports the preference of persons with ASD of keeping up routines and firm procedures.

1.4 Structure of the Work

The introduction (see chapter 1) of this thesis touches upon the pedagogical aspects and the neurological reasoning behind Robo-Smile's learning concept. Furthermore, an overview of the previous development and the aim of this thesis, which is to improve the control mechanism, are discussed, and the research questions of this thesis are formulated. Next, chapter 2 elaborates on the current state-of-the-art algorithms used for detecting facial landmarks (see section 2.1) and classifying emotions (see section 2.2). Furthermore, state-of-the-art methodologies are discussed concerning their applicability as part of Robo-Smilie's control mechanism. The methodology (see chapter 3) describes all designed architectures and conducted experiments to find the best architecture for the webcam-based feedback algorithm. Furthermore, the methodology used to develop the emotion feedback system is evaluated in chapter 4. The results are split into the performance evaluation of the facial landmark detector (see section 4.1), the evaluation of the emotion detector (see section 4.2), and the analysis of the overall execution times of the obtained emotion feedback system (see section 4.3). The results show that this thesis presents a novel facial landmark detector, which manages to achieve state-of-the-art detection performance and has sufficiently fast execution times to allow the frame-wise processing of a video stream. Furthermore, two unique strategies for developing an emotion detector using facial landmarks and information derived from these coordinates are presented. During the finding of suited methodologies and the evaluation, particular attention is paid to the conformity with Robo-Smile's educational concept. This thesis is concluded in chapter 5 by analyzing the findings of this work, stating its limitations, and giving an outlook on future challenges.

CHAPTER 2

State of the Art

Specific areas in the face associated with conveying information necessary for recognizing emotions are planned to be emphasized using the contours of facial features such as eyebrows, eyes, nose, and mouth. By marking those key-points and connecting them to form abstractions of eyes, nose, and mouth in the received video feedback, it is tried to draw the child's attention towards those regions. The highlighting of these areas is achieved using facial landmarks, which need to be detected automatically by the control mechanism. Furthermore, the pedagogical concept intends to use facial landmarks as the input of the emotion classification.

2.1 Facial Landmark Detection

Facial landmarks, also known as facial feature points or fiducial points, describe semantic facial points [24]. An example of a popular facial landmark representation is the Multi-PIE 68 points mark-up, which is shown in Figure 2.1. It can be seen that the mark-up abstracts the main facial features being eyebrows, eyes, nose, mouth, and the contour of the face. Automated facial landmark detection systems aim to map facial landmarks to image locations and label fiducial points in facial images.

2.1.1 Traditional Approaches

As facial landmark detectors are used in various applications such as facial recognition, facial expression analysis, age estimation, and gender classification [26] extensive research has been conducted in this area. Traditional facial landmark detection models can be classified into three major categories [27] [24].

- **Holistic models or Active Appearance Models (AAMs):**
  Holistic models such as AAM use statistical models for facial shape and global appearance information for detecting facial landmarks [28].

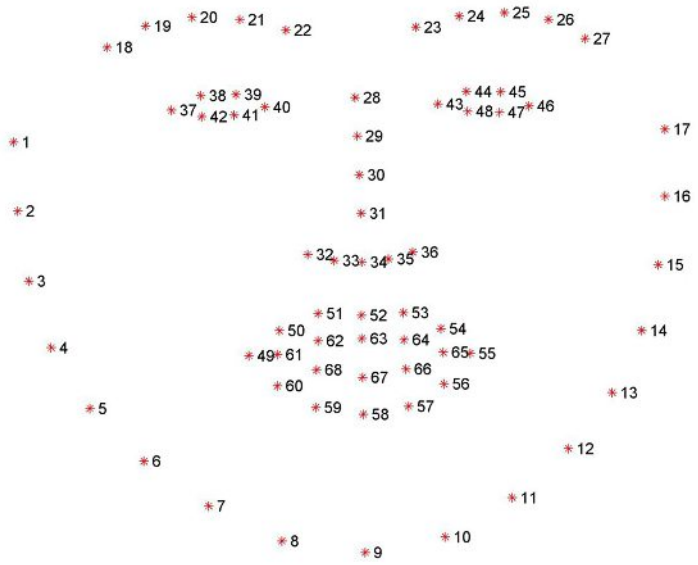


Figure 2.1: Multi-PIE 68 facial points mark-up [25]

- **Constrained Local Model (CLM)-based methods:**
  CLMs use a global shape model similar to AAM-based models, but with the distinction of additionally using local appearance information around each landmark [27].
- **Regression-based landmark detection methods**: Regression-based methods are usually not based on explicit shape models [27], but rather directly learn to map appearances in facial images to facial landmarks [24].

2.1.2 Facial Landmark Detection using Deep Learning

Recent advances in facial landmark detection algorithms show a trend of using deep-learning-based models [29], as these proved promising detection results. Similar to traditional methods for detecting facial landmarks, deep learning approaches can also be categorized as model-based or regression-based [22]. Model-based detectors try to fit created facial templates to a given input image, whereas regression-based methods try to improve detection iteratively using regression [22].

Table 2.1 summarizes the main deep-learning-based facial landmark detection methods and gives information on the execution time, as fast methods are essential for frame-based video detection. If not stated differently in the table, all given execution times only include the facial landmark detection and not the face detection needed prior to the facial landmark detector.

Deep-learning-based models are superior in accuracy but inferior in detection speed compared to regression and constrained local models [29]. However, a low computational cost and fast execution time are essential for processing webcam data. As the game

must be able to run fluently on weaker and older computers without GPU most of the methods described in Table 2.1 are not suited for the given task. However, the methods CNN-6 (using Wing Loss, 120 fps) [19], TCDCN (58.82 fps) [22] [23], MCL (57 fps) [21], Coarse-to-Fine Auto-Encoder (CFAN) (43.78 fps) [30], Face and Landmark Detector (FLDet) (19.60 fps) [31], Constrained Local Neural Field (CLNF) (2 fps) [32] are tested on a CPU.

Table 2.1: Summary of facial landmark detection methods and their execution time

Author (year)	**Method**	**Summary**	**Execution time and device used**
Xiong et al. (2013) [33]	Supervised Descent Method (SDM)	Combination of Supervised Descent method, which uses learned generic descent directions and Pose-Aware Models (PAMs); during tracking, the initialization is done using the landmark estimate of the previous frame	>30 fps (during tracking, without re-initialization), Intel i5-2400 CPU
Sun et al. (2013) [34]	Deep Convolutional Network Cascade for Facial Point Detection	Combination of deep and shallower convolutional neural networks	8.33 fps, 3.30 GHz CPU
Zhou et al. (2013) [35]	Extensive Facial Landmark Localization with Coarse-to-Fine Convolutional Network Cascade	Four-level Coarse-to-fine deep convolutional neural network cascade	not specified
Zhang et al. (2014 [22], 2016 [23])	TCDCN	Combination of facial landmark detection task with heterogeneous correlated tasks for improving detection performance	58.82 fps (5 landmarks), 55.56 fps (68 landmarks) Intel Core i5 CPU
Zhang et al. (2014) [30]	CFAN	Refinement of facial landmarks by cascading global and local Stacked Auto-encoder Networks	43.78 fps, Intel i7-3770 3.4 GHz CPU

Baltrušaitis et al. (2014) [32]	CLNF	Local Neural Field patch expert optimized using Non-Uniform Regularised Mean-Shift method	2 fps on in the wild data, 10 fps on Multi-PIE data, dual core Intel i7 3.5GHz
Ranjan et al. (2016) [36]	HyperFace	Combination of partly fused CNNs for simultanous face, landmark, pose and gender detection	10 fps (Fast-HyperFace), GTX TITAN X GPU
Xiao et al. (2016) [37]	Recurrent Attentive-Refinement Network (RAR)	Pipeline of cascaded regressions for progressive landmark position refinement	4 fps, Titan-Z GPU [21]
Jourabloo et al. (2016) [38]	Piecewise Affine-Warped Feature (PAWF), Direct 3D Projected Feature (D3PF)	Combination of cascaded CNN regressors and the 3D Morphable Model using two different features: PAWF, D3PF	1.67 fps (PAWF), 3.85 fps (D3PF), device not specified
Wu et al. (2016) [20]	TCNN	CNN model using additional fine-tuning of final layers	not specified
Zhang et al. (2016) [39]	Multitask Cascaded Convolutional Network (MTCNN)	Cascaded network with three stages	16 fps on 2.6 GHz CPU, 99 fps on Nvidia Titan Black GPU (including face detection)
Trigeorgis et al. (2016) [40]	Mnemonic Descent Method (MDM)	Coarse-to-fine shape refinement usind CNN-based and RNN-based units	not specified
Lv et al. (2017) [41]	Two-Stage Reinitialization (TSR)	A deep-regression-based network with two-stage re-initialization	83 fps, Nvidia Titan X GPU
Kowalski et al. (2017)	Deep Alignment Network (DAN)	Multiple deep network stages using landmark heatmaps	45 fps, GeForce GTX 1070 GPU
Yang et al. (2017) [42]	Stacked Hourglass Network	Model consisting of a supervised face transformation and four stacked Hourglass Networks	few seconds per frame, device not specified

He et al. (2017) [43]	Fully End-to-End Cascaded Convolutional Neural Network (FEC-CNN)	Cascaded CNN Network, which takes landmark areas of previous stage as input	10 fps, device not specified
Zhu et al. (2018) [44]	3D Dense Face Alignment (3DDFA)	A dense 3D Morphable Model (3DMM) is fitted using a Cascaded Convolutional Neural Network	15.65 fps, 3.4 GHz CPU and GTX TITAN X GPU
Feng et al. (2018) [19]	CNN using Wing Loss	A piece wise loss function called Wing Loss is used for training a CNN	150 fps (CNN-6), 20 fps (CNN-6/7), CPU
Shao et al. (2018) [21]	MCL	Deep learning network with multiple shape prediction layers	57 fps, Intel i5-6200U 2.3GHz CPU
Wu et al. (2018)	Look at Boundary (LAB)	Network using to boundary heatmaps transformed facial landmarks	16.67 fps, TITAN X GPU
Zhu et al. (2019) [45]	Occlusion-adaptive Deep Network (ODN)	Adaptation to the last residual unit of ResNet-18 using a geometry-aware module, a destillation module, and a low-rank learning module	1.79 fps/21.16 fps (ResNet-18), Titan Xp/ Jetson TX1 GPU [46]
Zhuang et al. (2019) [31]	A CPU Real-time Joint FLDet	Combination of a Rapidly Digested Backbone (RDB), a Lightweight Feature Pyramid Network (LFPN), and a Multi-task Detection Module	19.60 fps (includes face detection), Intel Xeon E5-2660v3 2.60GHz CPU

Based on the execution times and the network complexity, the methods CNN-6 (with Wing Loss), TCNN, MCL, and TCDCN are considered most suited for the control mechanism of Robo-Smile and thus are taken into consideration for choosing and designing the facial landmark detector used. They are explained in more detail in the following subsections.

2.1.3 Task-Constrained Deep Convolutional Network (TCDCN)

Zhang et al. [22] [23] propose the adaptation of a cascaded CNN to include information on the task. By utilizing auxiliary information on gender, pose, appearance (if the person wears glasses), and facial expression (if the person is smiling), they optimize the main task of localizing facial landmarks. They are using one CNN to train the detection of sparse

landmarks (5 points) and auxiliary attributes to expand their model to detect dense facial landmarks (e.g., 68 points). They propose the least square loss for the regression task of locating facial landmarks and cross-entropy as loss for the classification tasks of the auxiliary attributes. To facilitate training convergence, they use task-wise early stopping. Furthermore, Stochastic Gradient Descent (SGD) optimization, and Rectifier (ReLU) activation functions for the convolutional layers are used during the training procedure.

2.1.4 Wing Loss for Robust Facial Landmark Localization with CNN

Feng et al. [19] propose a new loss function, called Wing Loss, for robust facial landmark detection. They show that the same CNN-6 model trained using Ridge regression (L2), Lasso regression (L1) and smooth L1 loss functions perform well for large errors and suggest emphasizing on samples with small and medium errors using the Wing Loss function. The loss function is split to be suited for small and large errors. As seen in Equation 2.1 the function acts as a logarithmic function with offset for small errors and as L1 for large errors. The parameter w limits the area of non-linearity, ϵ describes the curvature of the wing loss function, and C is a constant linking the two functions. Similar, to TCDCN they use SGD optimization and ReLU activation functions for the convolutional layers.

$$wing(x) = \begin{cases} w \cdot \ln\left(1 + \frac{|x|}{\epsilon}\right) & if|x| < w, \\ |x| - C & otherwise \end{cases} \tag{2.1}$$

Furthermore, they propose a boosting strategy called Pose-based Data Balancing (PDB), capable of counteracting the problem of under-representing samples with large out-of-plane head pose variations in current image databases. They evaluate the performance of the wing function and PDB on two CNN models, a simple and fast CNN model (CNN-6) and a slower and more accurate two-stage landmark detector (CNN-6/7).

2.1.5 Tweaked Convolutional Neural Network (TCNN)

Wu et al. [20] present the fine-tuning of a vanilla CNN network, which is loosely based on TCDCN. They suggest that deeper layers of standard networks capture rough landmark locations and propose a tweaking process based on facial alignment. As seen in Equation 2.2 they use L2 normalized by the Inter-Ocular Distance (IOD) as loss function. P_i are the coordinates of the i^{th} facial landmark, with $i = 37$ and $i = 46$ being the landmarks of the outer corners of the eyes. $\hat{P}_i$ denotes the coordinates of the ground truth facial landmark locations. In contrast to TCDCN and CNN-6 with Wing Loss, Wu et al. use Adam optimization and the absolute hyperbolic tangent function as activation function.

$$L2_{normalized}(P_i, \hat{P}_i) = \frac{\|P_i - \hat{P}_i\|_2^2}{\|P_{37} - P_{46}\|_2^2} \tag{2.2}$$

2.1.6 Multi-Center Learning (MCL)

Shao et al. [21] propose a deep learning framework called MCL, which utilizes strong correlations between landmarks by using different shape prediction layers for semantically relevant facial landmark clusters. For clustering the 68 facial landmarks, seven groups are formed, thus separating the landmarks into the left eye, the right eye, the nose, the mouth, the contour of the chin region, the contour of the left side of the face, and the contour of the right side of the face. The first layers are shared by the various shape prediction layers and are composed of three max-pooling layers as suggested by Simonyan and Zisserman [47]. To ensure decreased model complexity, the multiple shape prediction layers are combined by using a model assembling function. Furthermore, during initial training, each landmark's loss is weighted to improve the detection performance of difficult facial landmarks. Their proposed loss function can be seen in Equation 2.3, where w_j is the weight of the j-th facial landmark, x_j and y_j represent the X- and Y-coordinates of the j-th landmark, $\hat{x_j}$ and $\hat{y_j}$ represent the ground truth X- and Y-coordinates of the j-th landmark, and $\hat{d}$ describes the ground truth IOD. The learning procedure uses SGD optimization, mini-batch sizes of 64 and Rectifier Activation functions for the convolutional layers.

$$E = \sum_{j=1}^{n} w_j \cdot \frac{(\hat{x_j} - x_j)^2 + (\hat{y_j} - y_j)^2}{2 \cdot d^2} \tag{2.3}$$

2.2 Emotion Detection

Nonverbal behavior plays an essential role in everyday life, as, for example, facial expressions alone contribute 55 % to the impact of a spoken message [48], which makes the detection of facial expressions essential for human communication. As a consequence, the recognition of emotion is well studied in various areas such as Human Computer Interactions (HCI) and Computer Vision [49]. For the recognition of emotion, various modalities can be utilized, such as text, sound, image or videos, and physiological signals [50].

Early research on emotions was conducted by Ekman and Friesen. Based on a cross-cultural study in the twentieth century, they defined six basic emotions, which they argued to be valid across all cultures [51]. However, current research contradicts the assumption proposed by Ekman and Friesen that the basic six emotions, happiness, sadness, surprise, disgust, fear, and anger, are culture unspecific [52]. Nevertheless, these emotions are used for teaching children with ASD to recognize emotion in facial expressions. Thus this thesis focuses on the six basic emotions and neutral expression.

2.2.1 Facial Action Coding System (FACS)

Emotion expression correlates with the activation of different facial muscles. Depending on the emotion, different muscles are activated, and thus the visualized facial expression varies. These observations can be classified using the FACS.

The FACS is considered one of the most popular emotion representations designed to facilitate objective measurements for facial expression analysis [48]. The FACS differentiates between 44 independent movements, which are referred to as AUs [53]. Which emotion correlates with each AU and muscular activity is described in Table 2.2.

The activation of single or multiple facial muscles leads to changes in the facial appearance. These visual appearance changes are described in more detail in Table 2.3. During the expression of emotions, AUs can appear isolated or as a combination of multiple AUs. The resulting appearance changes triggered by the combination of AUs are described as one entity since they influence each other.

2.2.2 Traditional Approaches

Automated emotion or affect recognition systems aim to detect facial actions or the emotions conveyed by the facial actions and are mostly based upon the previously described FACS [55]. Sariyanidi et al. [55] present a conceptional framework designed to analyze and compare facial affect recognition systems. Their design divides the overall task of facial expression analysis into four main parts.

- **Registration:** The registration step describes which information is gathered for the affect recognition systems. The registration can further be categorized depending on its output.
  - **Whole face registration:** The whole face is used for registration.
  - **Parts registration:** Facial parts such as eyes or mouth are used for registration.
  - **Points registration:** Fiducial points (or facial landmarks) are used for registration.

  Traditional methods rely on AAM for registering whole faces, parts of faces, or facial landmarks.

- **Representation:** The representation describes the modality of how the registration step is conducted. It can be separated into two main categories, being the spatial representation and the spatio-temporal representation.
  - **Spatial representation:** Image sequences are encoded frame-wise.
  - **Spatio-temporal representation:** A range of frames within a temporal window is analyzed as one entity.

- **Dimensionality Reduction:** The vast majority of traditional systems use dimensionality reduction techniques such as pooling, feature selection, and extraction.
- **Recognition:** The intended result of affect recognition systems is the labeling of emotion or visualized AUs. Most affect recognition systems use machine learning techniques such as SVM, which is one of the most popular methodologies for the given task.

2.2.3 Facial Expression Recognition using Deep Learning

Analogous to detecting facial landmarks, there is a shift from traditional models to deep-learning-based models for detecting facial expressions.

Similar to traditional models, the input data can either consist of a static image (spatial representation) or frames of a video sequence (spatio-temporal representation) [56]. Methods using spatio-temporal representation can either exploit the temporal relation explicitly or not. Methods, which do not explicitly exploit the temporal dependency usually operate on frame sequences, while methods exploiting the temporal relation explicitly take a range of frames in a temporal window as a single input [56]. The recognition performance of facial expression detectors increases if the input consists of image sequences instead of still-images. Thus current research focuses on the recognition of emotions in image sequences [57].

Traditional models require a feature extraction and feature selection step, which are independent of each other [56]. However, deep learning methods do not require those two steps to be separate, as they are capable of end-to-end facial expression recognition [56]. Nevertheless, algorithms exist which separate feature extraction and classification by using, e.g., deep learning methods for the feature extraction and an additional independent classification algorithm. The majority of algorithms harness the advantage of end-to-end classification but are not applicable for the present work since the game Robo-Smile aims to link patterns of facial landmarks to expressed emotions, thus approaches based on facial landmarks are preferable.

2.2.3.1 Emotion Detection using Facial Landmarks and Deep Learning

There are deep-learning-based architectures, which take additional information such as facial landmarks, gender, or pose into account [57] [49] [22] [23]. However, landmark-based facial expression detection is rarely studied since a proper deep-learning-based model capable of extracting sufficient information from facial landmarks has not yet been found [49].

Zhang et al. propose the TCDCN [22] [23] architecture, which combines auxiliary information on facial expression, gender, pose, and appearance to the task of facial landmark detection. As the control system of Robo-Smile must be able to output information on the locations of facial landmarks and the expressed emotion, this network architecture seems to be a promising option. However, it only distinguishes between smiling and

not smiling, making it difficult to estimate if an adaptation of the network to a facial expression detector capable of classifying all six basic emotions and neutral is possible. Furthermore, the image databases containing labeled facial landmarks and emotions required for training and testing the combined network currently do not exist. Hence this approach is discarded.

Jung et al. propose the Deep Temporal Appearance-Geometry Network (DTAGN), which is a combination of a Deep Temporal Geometry Network (DTGN) and a Deep Temporal Appearance Network (DTAN), which use joint fine-tuning and image sequences as input [57]. Ngoc et al. propose the usage of Directed Graph Neural Networks (DGNN), which also bases on image sequences as network input [49]. However, these networks are only partially suited for the current task since they have not yet been used on continuous video data but only on image sequences with a defined start and end time. Furthermore, the pedagogical concept of Robo-Smile does not rely on temporal information. Instead, it focuses on the patterns formed by the facial landmarks at specific points in time, thus making DTAGN and DTAN unfit for the development of this work.

2.2.4 Findings with Respect to the Applicability of State-of-the-Art Methods for Robo-Smile's Control Mechanism

The state-of-the-art facial landmark detectors CNN-6 (with Wing Loss), TCNN, MCL, and TCDCN feature execution times suited for live feedback without the usage of GPUs and thus are considered most suited for the control mechanism of Robo-Smile. All methods present different and unique aspects contributing to the state-of-the-art accuracy of these algorithms. Thus the combination and adaptation of these characteristics can facilitate the development of a new detector capable of improving the detection performance whilst achieving execution times similar to the base models.

Furthermore, AUs are used to facilitate objective measurements for facial expressions. Thus the applicability for a facial-landmark-based facial expression classifier in combination with traditional methods should be further evaluated. Additionally, as state-of-the-art emotion detectors utilize deep learning methodologies, utilizing this approach using facial landmark patterns (spatial arrangement of facial landmarks) could also lead to promising results, suggesting the need to further evaluate the applicability for the given task.

Table 2.2: Descriptions of AU and facial muscles associated with emotions [53] [54]. Optional AUs are writen in parentheses.

Emotion	AU numbers	Descriptor	Muscular Basis
Happiness	6	Cheek Raiser	Orbicularis Oculi, Pars Orbitalis
	12	Lip Corner Puller	Zygomatic Major
Sadness	1	Inner Brow Raiser	Frontalis, Pars Medialis
	(4)	Brow Lowerer	Depressor Glabellae, Depressor Supercilli, Corrugator
	15	Lip Corner Depressor	Triangularis
	(17)	Chin Raiser	Mentalis
Surprise	1	Inner Brow Raiser	Frontalis, Pars Medialis
	2	Outer Brow Raiser	Frontalis, Pars Lateralis
	5	Upper Lid Raiser	Levtor Palpebrae Superioris
	(25 or 26)	25: Lips Part	Depressor Labii, or Relaxation of Mentalis or Orbicularis Oris
		26: Jaw Drop	Masetter, Temporal and Internal Pterygoid Relaxed
Fear	1	Inner Brow Raiser	Frontalis, Pars Medialis
	2	Outer Brow Raiser	Frontalis, Pars Lateralis
	4	Brow Lowerer	Depressor Glabellae, Depressor Supercilli, Corrugator
	5	Upper Lid Raiser	Levtor Palpebrae Superioris
	7	Lid Tightener	Orbicularis Oculi, Pars Palebralis
	20	Lip Stretcher	Risorius
	(25 or 26)	25: Lips Part	Depressor Labii, or Relaxation of Mentalis or Orbicularis Oris
		26: Jaw Drop	Masetter, Temporal and Internal Pterygoid Relaxed
Anger	4	Brow Lowerer	Depressor Glabellae, Depressor Supercilli, Corrugator
	5	Upper Lid Raiser	Levtor Palpebrae Superioris
	24	Lip Pressor	Orbicularis Oris
	38	Nastril Dilator	-
Disgust	9	Nose Wrinkler	Levator Labii Superioris, Alaeque Nasi
	10	Upper Lip Raiser	Levator Labii Superioris, Caput, Infraorbitalis
	(25 or 26)	25: Lips Part	Depressor Labii, or Relaxation of Mentalis or Orbicularis Oris
		26: Jaw Drop	Masetter, Temporal and Internal Pterygoid Relaxed

Table 2.3: Description of facial changes during the expression of AUs and AU combinations associated with emotion.

AUs	Description
1 + 4	The medial eyebrow corners are raised and pulled together.
1 + 2	The entire eyebrow is raised upwards.
1 + 2 + 4	The entire eyebrow is raised upwards,whilst the eyebrows are slightly pulled together.
4	The eyebrows are lowered and pulled together.
6 + 12	The inner and outer orbicularis oculi muscle are contracted.
5	The upper eyelids are pulled up which leads to a wider opening of the eyes.
5 + 7	The upper and lower eyelids are raised. The lower eyelids are furthermore straightened slightly, which causes slight bulging
15	The lip corners are pulled downwards.
25	The lips part.
20 + 25	The lips are stretched whilst the lips part.
24	The lips are pressed together and narrowed.
10 + 25	The upper lip is slightly raised leading to parting lips.
38	The nostrils are dilated.
9	The nose is wrinkled, the skin on the nose bridge is lifted upwards and the nasal wings are pulled upwards.
17	The chin is pushed up.
26	By relaxing the jaw muscles, the jaw is lowered.

CHAPTER 3

Methodology

This work presents a control algorithm that uses the video input of a webcam for detecting the expressed emotion and giving feedback. For this purpose, the algorithm is split into two main steps. First, facial landmarks are localized, and then, based on these landmarks, the expressed emotion is classified. In order to allow the application to run on live webcam streams without causing noticeable delay or pauses, additional strategies must be considered to reduce the execution time. Several novel architectures are conceived for both facial landmark detection and emotion classification. By designing various experiments comparing these different system architectures, it is aimed to find the best-suited methodologies for the overall feedback system and the sub-algorithms for detecting facial landmarks and the emotion classification. This facilitates information on the best overall system and allows the comparison of different approaches and detection strategies.

All scripts used to develop the control mechanism and its evaluation are written in Python [58]. The image processing library OpenCV [59] is used for camera access, transmitting the video stream, and image pre-processing. Tensorflow[1] and scikit-learn [60] are used for detector training.

3.1 Facial Landmark Detection

During the imitation task, blue lines are superimposed on the child's face to highlight facial patterns for easier face and facial expression processing. For this purpose, salient facial key points following the Multi-PIE 68 points mark-up need to be detected by a facial landmark detection algorithm. This thesis considers Zhang et al.'s [23], Feng et al.'s [19],

[1] M. Abadi et al., "TensorFlow: Large-scale machine learning on heterogeneous systems," 2015, software available from tensorflow.org. [Online]. Available: https://www.tensorflow.org/. Accessed: 22 June 2020.

Shao et al.'s [21], and Wu et al.'s [20] deep-learning-based facial landmark detectors, and Zhu et al.'s. [20] method to generate large training sets emphasizing samples of profile views. By combining and adapting various implementation aspects of the different deep learning models, an attempt is made to find an accurate and robust facial landmark detector, which is in compliance with the given time and performance limits.

Based on the state of the art, twelve different original landmark detector designs are implemented and tested. All tested implementations are listed in Table 3.1 with each row describing the configuration used during each experiment. All experiments are numbered consecutively and differ from each other in either one or more aspects regarding the size of the input image and whether color images are used or not, if additional image augmentation is performed apart from artificially generating profile views, the performed normalization strategy, the design base on which the network architecture is built upon, the activation function used, and the loss function used. Furthermore, all listed configuration options are discussed in more detail in the following sections.

All implementations are trained on either a GeForce GTX 1080 or a Tesla T4 GPU.

Table 3.1: Summary of tested implementations during training of the facial landmark detector

#	**Input**	**Augmentation**	**Normalization**	**Network design base**	**Activation function**	**Loss function**
1	224 × 224 px BGR	no additional augmentation	per-channel mean normalization, scaling $[-1, 1]$	Vanilla TCNN with additional dense layer	abstanh	L2-based (area of 0.5 %)
2	224 × 224 px BGR	no additional augmentation	per-channel mean normalization, scaling $[-1, 1]$	Vanilla TCNN with additional dense layer	abstanh	L2-based (area of 1 %)
3	224 × 224 px BGR	no additional augmentation	per-channel mean normalization, scaling $[-1, 1]$	Vanilla TCNN with additional dense layer	abstanh	L2-based (area of 2 %)
4	224 × 224 px BGR	no additional augmentation	per-channel mean normalization, scaling $[-1, 1]$	Vanilla TCNN with additional dense layer	abstanh	L2-based (area of 3 %)
5	224 × 224 px BGR	additional mirroring	per-channel mean normalization, scaling $[-1, 1]$	Vanilla TCNN with additional dense layer	abstanh	L2 based (area of 2 %)

6	50×50 BGR	no additional augmentation	per-channel mean normalization, scaling $[-1, 1]$	Vanilla TCNN	abstanh	L2-based (area of 2 %)
7	50×50 px BGR	no additional augmentation	per-channel mean normalization, scaling $[-1, 1]$	Vanilla TCNN	abstanh	wing loss
8	124×124 px BGR	no additional augmentation	per-channel mean normalization, scaling $[-1, 1]$	Vanilla TCNN	abstanh	wing loss
9	50×50 px Grayscale	no additional augmentation	batch normalization, scaling $[-1, 1]$	MCL	ReLU	L2-based (area of 2 %)
10	50×50 px Grayscale	no additional augmentation	batch normalization, scaling $[-1, 1]$	MCL	abstanh	L2-based (area of 2 %)
11	50×50 px Grayscale	no additional augmentation	batch normalization, scaling $[-1, 1]$	MCL	ReLU	wing loss
12	50×50 px Grayscale	no additional augmentation	batch normalization, mean centering and normalization using the standard deviation	MCL	ReLU	wing loss

3.1.1 Image Database and Data Augmentation

The facial landmark detection must be robust to large pose variations. Thus an image database containing a variety of different poses is essential. However, most facial landmark databases such as AFLW [61], LFPW [62], HELEN [63], and IBUG [25] contain only medium pose-variations. A contradicting example is the Database Annotated Facial Landmarks in the Wild (AFLW) [61], which contains images with large pose variations, but only 21 annotated landmarks and does not include covered and thus not visible landmarks. Reasons for the lack of current databases containing fully annotated samples with large pose variations are that current initialization algorithms used for manual annotation struggle with profile images and that occluded landmarks must be guessed and, therefore, are difficult to annotate [64]. Zhu et al. [64] solve this issue by presenting

a method capable of artificially generating labeled profile views of heads turned up to 90°.

The 300W-LP database [64] is an extension of the 300-W dataset [65] [25] [66] and contains 61255 samples including augmented versions using Zhu et al.'s methodology of generating labeled profile views of the IBUG [25], AFW [67], LFPW [62], and HELEN [63] database. For training purposes the 300W-LP database [64] is split into training (85 %) and validation set (15 %). For testing purposes the private and public 300-W [65] [25] [66] test sets are used. The public test set is furthermore split into the common (HELEN [63] and LFPW [62]) and the challenging (IBUG [25]) subset.

For all designs except design #5, Zhu et al.'s [64] method of generating artificial samples with large head pose variations is the only image augmentation technique used. However, to further increase the samples of the training set during design #5, each sample image and the corresponding set of 68 landmarks are flipped, and thus the training set is doubled.

3.1.2 Preprocessing

Images used need to be preprocessed for further training to exclude image areas without faces and to further center and normalize the training samples to improve the training performance.

Image databases used for training can include more than one face per image and large areas of background. However, facial landmark detectors operate on face images as input. Thus, as a first step, faces in images need to be extracted. OpenCV [59] offers a Haar-cascade-based and a deep-learning-based face predictor. The deep-learning-based predictor provides superior detection accuracy as seen in Table 4.1, especially during large pose variations. Thus, the deep-learning-based face detector is used to extract the faces from the sample images. Visual inspection of the face detector's performance showed that the resulting bounding boxes cropped facial features, such as noses in profile images. As a workaround, the bounding boxes are enlarged. In X-direction, a border of 10 % of the original bounding box's width is added to both sides of the face. In Y-direction, a border of 5 % is added to the chin area and a border of 15 % to the forehead. Images with falsely detected bounding boxes are not discarded, but the database's bounding boxes are used for extraction instead.

Depending on the Network Design, which is either based on the Vanilla TCNN-Network or the MCL-Network, the input image is a color image using the Blue Green Red (BGR) color space or a grayscale image respectively.

All images used for training need to be normalized to increase the facial landmark detector's training performance. Depending on the network and training attempt, various normalization techniques are used.

- **Designs** #1-#8**:** Each color channel's mean pixel value is computed over all extracted face image samples of the training data set to center and normalize the

input data for training. The per-channel means are subtracted from each image afterward. Furthermore, all pixel values are normalized on a per-channel basis to values between −1 and 1.

- **Designs** #9-#11**:** Based on Shao et al.'s proposed input image normalization, which is used during the training of the MCL network [21], each pixel value is normalized to values between −1 and 1. Furthermore, Batch Normalization layers are used throughout the network. Ioffe and Szegedy propose the method of Batch normalization in order to accelerate the training process of deep learning networks by reducing the internal covariance shift [68]. This is achieved by normalizing the activation of a previous layer for each batch. Thus the inputs of a layer are transformed to have a constant mean activation of 0 and a constant activation standard deviation of 1 for each mini-batch.

- **Designs** #12**:** Each grayscale image is centered using the mean pixel value and standardized using each training image's standard deviation to center and standardize the training's input data. Similar to design #9-#11 Batch Normalization Layers are used throughout the network. This normalization strategy is inter alia used during the training of the TCDCN model.

Furthermore, since the detected faces' resolution can vary depending on the sample used, all extracted faces need to be re-sized to guarantee a uniform input.

- **Designs** #1-#5**:** Similar, to He et al.'s Residual Network (ResNet) [69], the input image is re-sized to 224 × 224 px during the designs #1-#5.

- **Designs** #6**,** #7 **and** #9-#12**:** The input image is re-scaled to 50 × 50 px during the designs #6, #7, and #9 to #12, which is used during the training of the MCL model [21].

- **Design** #8**:** During the design #8 the input face image is scaled to 124 × 124 px, as it represents a value between 50 × 50 px and 224 × 224 px.

3.1.3 Network Design and Training

The facial landmark detector's speed is essential to guarantee the utilization of the detector on video stream data. Furthermore, experiments are conducted using three different Network architectures, which are based on the architectures TCNN [20], and MCL [21] and are adapted to guarantee fast execution times and accurate prediction during large pose variations.

- **Vanilla TCNN:** The Vanilla TCNN described by Wu. et al. [20] is loosely based on Zhang et al.'s TCDCN [22] [23]. The network is chosen as a base due to its fast execution time, allowing real-time performance on a CPU. The original

implementations, however use additional information [22] [23] or fine tuning [20] and are furthermore only trained to detect 5 facial landmarks. Two different adaptations are trained and tested, which vary in the number of dense layers used. The adapted models are shown in Figure 3.1 and Figure 3.2.

  - In Figure 3.1 the TCNN based and adapted model architecture consisting of five Convolutional Layers, five Max-Pooling Layers, and three Fully Connected Layers is shown. Furthermore, Absolute Tangens Hyperbolicus (abstanh) function units are used after each Convolutional Layer. The network's input is the facial color image in BGR color space extracted by the face detector. The network outputs a vector containing the coordinates of the 68 facial landmarks. No padding is used for each Convolutional and Max-Pooling Layer. The described network architecture is used during the designs #1 to #5.
  - In Figure 3.2 the TCNN based and adapted model architectures consisting of five Convolutional Layers, five Max-Pooling Layers, and two Fully Connected Layers are shown. Furthermore, abstanh function units are used after each Convolutional Layer. The network's input is the facial color image in the BGR color space extracted by the face detector. The network outputs a vector containing the coordinates of the 68 facial landmarks. Even padding to the left/right or above/below is used during Convolutional Layers and Max-Pooling Layers to maintain consistent input and output dimensions. The described network architecture is used during design #8 (input image size: 124×124 px, shown in the upper part of the figure) and designs #6 and #7 (input image size: 50×50 px, shown in the lower part of the figure).

- **MCL:** Shao et al. [21] propose the method of Deep Multi-Center Learning, which consists of shared layers, various shape prediction layers for predicting clusters of landmarks, and one final shape prediction layer for predicting all 68 facial landmarks. The network is chosen as a base since the model demonstrates real-time performance while handling complex occlusions and appearance variations [21]. Only the shared layers, which consist of three stacks of two convolutional layers [47] each followed by a max-pooling layer, one stack of three convolutional layers, and one Global Average Pooling Layer, are used to decrease the model's complexity and execution time even further. Furthermore, one final dense layer is added to enable the detector to output all 68 facial landmarks. The adapted model is shown in Figure 3.3. Summarizing, it consists of nine Convolutional Layers, three Max-Pooling Layers, one Global Average Pooling Layer, and one Fully-Connected Layer. Furthermore, similar to the original paper [21] Batch Normalization is used after each convolutional layer to improve the convergence of the network. The network's input is the facial grayscale image extracted by the face detector. The network outputs a vector containing the coordinates of the 68 facial landmarks. Even padding to the left/right or above/below is used during Convolutional Layers and Max-Pooling Layers to maintain consistent input and output dimensions. The described network architecture is used during the designs #9 to #12.

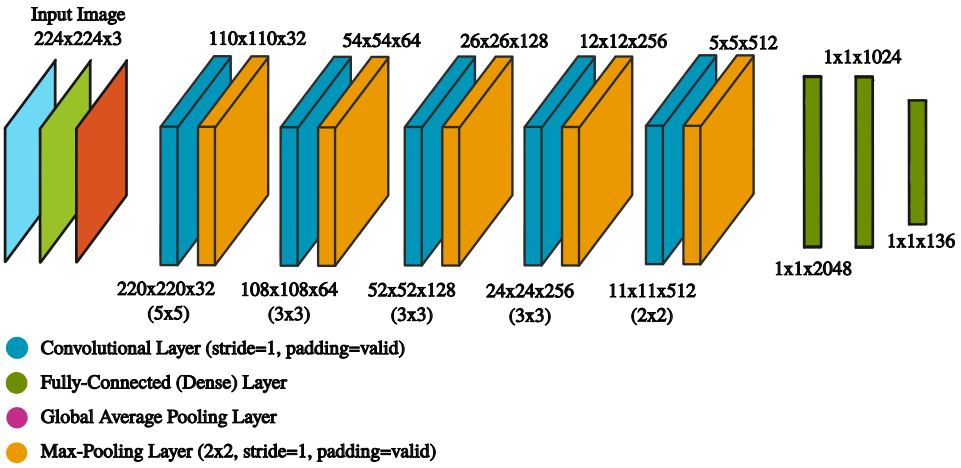


Figure 3.1: TCNN based and adapted model architecture used during designs #1 to #5.

Experiments are performed using two different loss functions, being an adaptation of Wu et al.'s [20] proposed loss function and Feng et al.'s [19] proposed wing loss function to find the best model.

- **L2-based loss function using landmark areas:** Insufficient training data and inconsistent annotations are thought to be the main reasons for current facial landmark detectors to lack precision despite good accuracy [70]. Precision describes how narrowly the results are distributed, whereas accuracy describes the mean measurements' distance to the ground truth value. Thus, applying an image-based detector on each frame of a video can lead to jitter and detection instability [70]. The loss function is adjusted to allow annotation error in close proximity of the ground-truth manual annotation without increasing the loss to counteract the variance in annotations and the resulting instabilities during video annotations.
- **Wing Loss:** The split Wing loss function shows superior performance during facial landmark localization when compared to L2, L1 and smooth L1 as it emphasizes on optimizing small and medium range errors [19]. As Feng et al. suggest, the parameters of the Wing loss are set to $w = 10$ and $\epsilon = 2$ during designs #7, #8, #11 and #12.

The batch sizes used during the training of a specific network are not always listed. However, Wu et al. and Shao et al. [21] suggest the usage of mini-batch sizes of 8 [19] and 64 [21] samples, respectively. Empirical testing showed that using batch sizes of 64 samples performed well for all designs and different network architectures used during this work. SGD optimization is used in order to optimize facial landmark detector models such as MCL [21], TCDCN [22] [23], and CNN-6 with wing loss. However, throughout all designs, Adam optimization is used as suggested by Wu et al. [20] as it proved to lead to fast learning and accurate regression results.

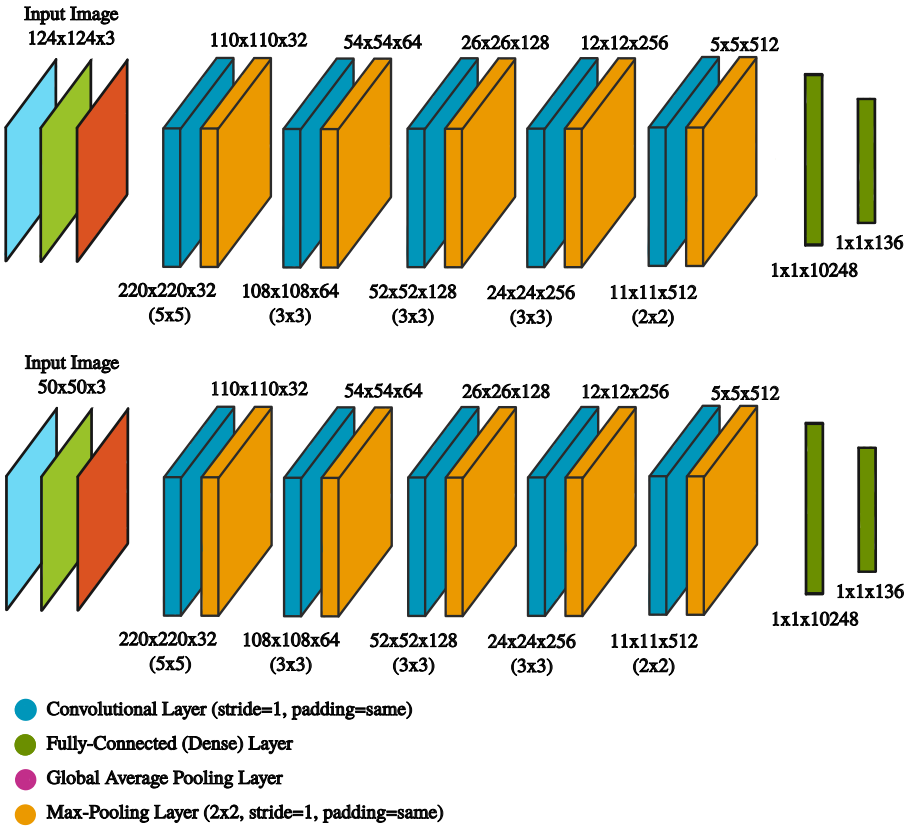


Figure 3.2: TCNN based and adapted model architectures used during design #8 (shown in upper part of the figure) and designs #6 and #7 (shown in lower part of the figure).

The learning rate is decreased automatically when the validation loss stops improving. Furthermore, if decreasing the learning rate does not improve the learning progress, early stopping of the training is used to improve the facial landmark detector's learning.

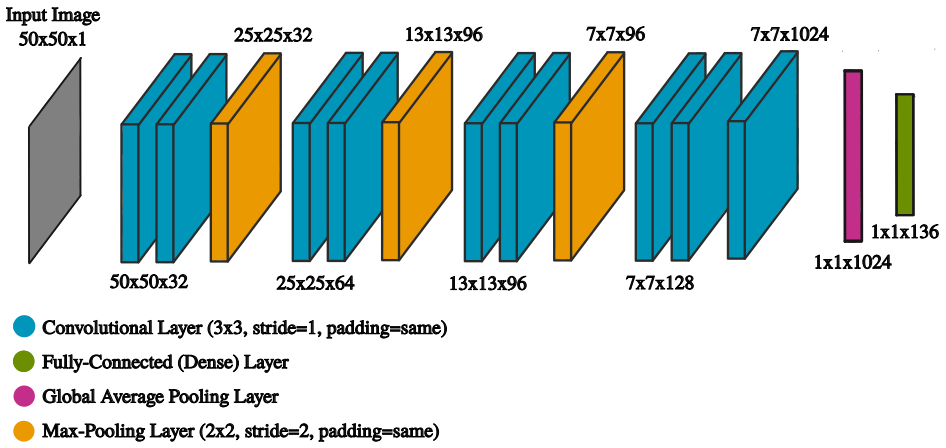


Figure 3.3: MCL based and adapted model architecture used during designs #9 to #12.

3.2 Emotion Detection

The learning platform focuses on teaching children the basic six emotions happiness, sadness, disgust, fear, anger, and surprise, as well as the detection of neutral faces. Therefore, an emotion detector capable of distinguishing those seven different states is needed. However, end-to-end models for detecting emotions are not preferable in this case since this thesis aims to link patterns of facial landmarks to expressed emotions. Thus, the emotion detector's input is limited to the facial landmarks and information, which can be derived by facial landmarks such as distances between points. As profile views partly hide facial landmarks, the detection emphasizes on frontal face recordings. This contradicts the emphasis on robustness to pose variation set during the development of the facial landmark detector. However, the decision is made to guarantee the accuracy of the detector, as state-of-the-art facial expression detectors struggle with occlusion and pose variation due to the lack of sufficiently labeled training data [56].

3.2.1 Emotion Detection from Facial Landmarks using AU

AU are used to facilitate objective measurements for facial expression. Thus this facial expression detector design is based on information derived by the FACS system. Due to the restriction of using only facial landmarks and information derived from facial landmarks such as distances between points, the AUs need to be described using this information alone. Thus, distances in the face must be found, which correspond to AUs needed to express the basic six emotions. However, not all AUs can be expressed using facial landmarks alone. For example, wrinkles around the nose during AU 9 can not be described by the Multi-Pie 68 facial points mark-up. Thus these aspects are neglected. Furthermore, several distances can be used to describe the same AU. Thus

feature selection is performed. Traditional approaches are used for the final classification of facial expressions.

3.2.1.1 Image Databases

As the game user should mimic emotions and thus learn how to act out different feelings, an image dataset containing acted emotions is preferable to in the wild image databases. Thus, the Radboud Facial Expression Database [71] is used for training and testing the emotion detector. The database contains images of 67 probands, including Caucasian males, females, and children, as well as Moroccan Dutch males [71]. During training, only frontal facial views are used since profile views partly hide facial landmarks. Furthermore, the resulting dataset is shuffled and split into training (70 %) and testing set (30 %). No additional validation set is used, as the classifiers' hyperparameters are optimized using cross-validated grid search.

3.2.1.2 Feature Selection and Preprocessing

For the development of a facial expression detector based on facial landmarks and AU features must be identified, which optimally describe the AUs associated with the basic six emotions. Therefore, various distances in the face are computed, which can be separated by the facial feature they are describing. A summary of which AUs can be used to describe the different emotions and what facial feature is affected during the expression is shown in Table 3.2. However, several distances can be used to describe the same AU. Thus univariate statistical tests are performed in order to find the best-suited parameter. Each facial change described by an AU linked to one of the basic six emotions is represented by the feature corresponding to the movement with the largest F-values computed using Analysis of Variance (ANOVA).

Table 3.2: Summary of affected facial features and expressed AUs associated with emotions.

Emotion	AUs	Eyebrows	Eyes	Mouth	Nose	Jaw
Happiness	6, 12		6 + 12			
Sadness	1, 4, 15, 17	1 + 4		15		17
Surprise	1, 2, 5, 25, 26	1 + 2	5	25		26
Fear	1, 2, 4, 5, 7, 20, 25, 26	1 + 2 + 4	5 + 7	20 + 25		26
Anger	4, 5, 24, 38	4	5	24	38	
Disgust	9, 10, 25, 26			10 + 25	9	26

Eyebrows: The eyebrows are affected during the expression of sadness (AU 1 + 4: the medial eyebrow corners are raised and pulled together), surprise (AU 1 + 2: the entire eyebrow is raised upwards), fear (AU 1+2+4: the entire eyebrow is raised upwards, while the eyebrows are slightly pulled together) and anger (AU 4: the eyebrows are lowered and

pulled together). In summary, the medial, lateral, and nasal eyebrow position and the distance between the eyebrows must be described by these distances. In order to describe these facial changes, the following distances between facial landmarks are identified.

- **Lateral eyebrow corners - eye orientation line:** This measurement describes the average of the left and right minimum distance between the lateral eyebrow corner and an imagined line between the nasal and lateral eye corners (minimum distance between #27 and line through #43 and #46, minimum distance between #18 and line through #40 to #37).
- **Center eyebrow corners - eye orientation line:** This feature describes the average of the left and right minimum distance between the center eyebrow corner and an imagined line between the nasal and lateral eye corners (minimum distance between #25 and line through #43 and #46, minimum distance between #20 and line through #40 to #37).
- **Nasal eyebrow corners - eye orientation line:** This measurement describes the average of the left and right minimum distance between the nasal eyebrow corner and an imagined line between the nasal and lateral eye corners (minimum distance between #23 and line through #43 and #46, minimum distance between #22 and line through #40 to #37).
- **Distance between inner eyebrow corners:** This parameter describes the distance between the nasal eyebrow corners (distance between #22 and #23).
- **Lateral eye corners - lateral eyebrow corners:** This parameter is the average distance between the left and right lateral eye corner and the left and right lateral eyebrow corner (distance between #46 and #27, distance between #37 and #18).
- **Nasal eye corners - nasal eyebrow corners:** This feature is the average distance between the left and right nasal eye corner and the left and right nasal eyebrow corner (distance between #43 and #23, distance between #40 and #22).

Eyes: The eyes are affected during the expression of happiness (AU $6 + 12$: contracting of inner and outer orbicularis oculi muscle), fear (AU $5 + 7$: the upper and lower eyelids are raised; the lower eyelids are furthermore straightened slightly, which causes slight bulging) as well as surprise, and anger (AU 5: the upper eyelids are pulled up which leads to a wider opening of the eyes). Thus, distances must describe changes in eye height and width and changes in the height of the upper and lower half of the eyes.

- **Upper eye corners - eye orientation line:** This feature describes the average of the left and right minimum distance between the upper eye corners and an imagined line between the nasal and lateral eye corners (minimum distance between #45 and line through #43 and #46, minimum distance between #44 and line

through #43 and #46, minimum distance between #38 and line through #40 and #37, minimum distance between #39 and line through #40 and #37).

- **Lower eye corners - eye orientation line:** This measurement describes the average of the left and right minimum distance between the lower eye corners and an imagined line between the nasal and lateral eye corners (minimum distance between #47 and line through #43 and #46, minimum distance between #48 and line through #43 and #46, minimum distance between #42 and line through #40 and #37, minimum distance between #41 and line through #40 and #37).
- **Width of the eyes:** The feature describes the average of the left and right distance between the nasal and lateral eye corners (distance between #43 and #46, distance between #40 and #37).
- **Upper eyelid height:** This feature is the average distance between the left and right center eye landmarks and the corresponding center eyebrow landmarks (distance between #45 and #26, distance between #44 and #24, distance between #38 and #21, distance between #39 and #19).

Mouth: The mouth is affected during the expression of sadness (AU 15: the lip corners are pulled downwards), surprise (AU 25: the lips part), fear (AU $20 + 25$: the lips are stretched while the lips part), anger (AU 24: the lips are pressed together and narrowed), and disgust (AU $10 + 25$: the upper lip is slightly raised leading to parting lips). In summary, the mouth's height and width, the position of the mouth corners in relation to the center of the mouth, the distance between the upper mouth corners and the nose, and whether the lips are parted or not must be described.

- **Lateral mouth corners - lateral eye corners:** This measurement describes the average distance of the left and right lateral mouth corner and lateral eye corner (distance between #55 and #46, distance between #49 and #37).
- **Inner upper lip corner - mouth orientation line:** This feature describes the minimum distance between the inner upper lip corner (#63) and the line trough the outer lateral mouth corners (line through #55 and #49).
- **Inner lower lip corner - mouth orientation line:** This parameter describes the minimum distance between the inner lower lip corner (#63) and the line trough the outer lateral mouth corners (line through #55 and #49).
- **Outer upper lip corner - mouth orientation line:** This measurement describes the minimum distance between the outer upper lip corner (#52) and the line trough the outer lateral mouth corners (line through #55 and #49).
- **Outer lower lip corner - mouth orientation line:** This feature describes the minimum distance between the outer lower lip corner (#63) and the line trough the outer lateral mouth corners (line through #55 and #49).

- **Width of the mouth:** Distance between the lateral, outer lip corners (#55 and 49).
- **Height of the mouth:** Distance between the nasal, outer upper and lower lip landmarks (#52 and 58).
- **Nasal mouth center - mouth orientation line:** This measurement describes the minimum distance between the center of the mouth, meaning the center of the nasal inner, outer, upper and lower mouth landmarks (#52, #63, #67, and #58) and the line trough the outer lateral mouth corners (line through #55 and #49).
- **Outer lower lip corner - chin:** This measurement describes the distance between the outer lower lip corner (#58) and the chin (#9).
- **Lateral lip corners - chin:** This measurement describes the minimum distance between the chin (#9) and the imagined line through the lateral eye corners (line through #55 and #49).
- **Lips part:** This measurement describes the distance between the inner upper (#63) and lower (#67) lip corner.
- **Tip of the nose - upper mouth corner:** This measurement describes the distance between the tip of the nose (#31) and the upper mouth corner (#52).

Nose: The nose is affected during the expression of anger (AU 38: the nostrils are dilated) and disgust (AU 9: the nose is wrinkled, skin on the nose bridge is lifted upwards, and the nasal wings are pulled upwards). However, not all changes in the area of the nose can be described using facial landmarks alone since facial landmarks do not capture wrinkles or the nostrils. Thus only the upward pulling of the nasal wings can be described.

- **Nasal wings - nasal eye corners:** This measurement describes the average distance between the left and right nasal wings and the nasal eye corners (distance between #36 and #43, distance between #32 and #40).

Jaw: The jaw is affected during the expression of sadness (AU 17: the chin is pushed up), surprise, fear, and disgust (AU 26: by relaxing the jaw muscles, the jaw is lowered). Summarizing, the jaw's position with respect to the center of the face needs to be described.

- **Tip of the nose - chin:** This measurement describes the distance between the tip of the nose (#31) and the chin (#9).

The facial landmark coordinates received by the facial landmark detector are scaled to values between 0 and 1. These coordinates are used to compute the distances describing the AUs. All distances are then normalized by the IOD. Furthermore, all features are mean-centered and scaled to have unit variance.

3.2.1.3 Classifier Design and Training

In order to find the best emotion classifier, popular classification methods are trained and evaluated. The hyperparameters of the classifier models are optimized using a cross-validated grid search over a corresponding parameter grid.

- **Tree classifier:** A decision tree classifier facilitates the supervised learning of simple decision rules, which allow a classification procedure, which is simple, easy to understand, and can be visualized. Since these rules could also be used for teaching children with ASD, this classifier seems promising for the given task. In order to find the best tree classifier for the facial expression classification, the following hyperparameters are optimized:
  - **Maximum depth:** This parameter limits the depth of the tree. The values used for the parameter grid search are set to all integer values in the range from 3 to 10.
  - **Criterion:** This hyperparameter describes which function should be used for measuring the quality of a split. The scikit-learn [60] implementation supports the Gini impurity and the entropy-based information gain.
  - **Splitter:** This parameter decides which strategy is used to choose node splits. The scikit-learn [60] implementation supports choosing the best split and choosing the best random split.
- **Random forest classifier:** A random forest classifier represents an ensemble of tree classifiers. The implementation of scikit-learn [60] combines these tree classifiers by computing the average of their probabilistic prediction and allows the tuning of the following hyperparameters:
  - **Maximum depth:** This parameter limits the depth of the tree. The values used for the parameter grid search are set to all integer values in the range from 3 to 10.
  - **Number of estimators:** Describes how many estimators are used. The parameter grid used for tuning this hyperparameter is $[10, 30, 50, 100, 300, 500]$.
  - **Criterion:** This hyperparameter describes which function should be used for measuring the quality of a split. The scikit-learn [60] implementation supports the Gini impurity and the entropy-based information gain.
- **AdaBoost classifier:** An implementation of Zhu et al. Multi-class AdaBoost [72] is used to build a boosted ensemble using a tree classifier as a base estimator. The following hyperparameters are tuned to optimize the classifier:
  - **Maximum depth:** This parameter limits the depth of the base classifier. The values used for the parameter grid search are set to all integer values in the range from 3 to 10.

  - **Criterion:** This hyperparameter describes which function the base estimator uses for measuring the quality of a split. The scikit-learn [60] implementation supports the Gini impurity and the entropy-based information gain.
  - **Splitter:** This parameter decides which strategy is used by the base estimator to choose node splits. The scikit-learn [60] implementation supports choosing the best split and choosing the best random split.
  - **Maximum number of estimators:** This hyperparameter describes the maximum number of estimators at which the boosting procedure is stopped. The parameter grid used for tuning this hyperparameter is $[10, 30, 50]$.
  - **Learning rate:** The learning rate describes the factor by which each classifier's influence is decreased and thus facilitates the controlling of the contribution of each tree to the prediction of the ensemble. The parameter grid is set to values between 0.1 and 2 in increments of 0.1.

- **K-nearest neighbors classifier:** This classifier decides based on the test set's data points in closest proximity to the data point, which needs classification.
  - **Number of neighbors:** This parameter describes how many neighbors are used for voting. The chosen parameter grid for tuning this hyperparameter is set to $[3, 5, 7]$.
  - **Metric:** The Euclidean distance, the Manhattan distance, and the Minkowski distance are tested during hyperparameter optimization.

- **SVM:** SVMs aim to classify unseen data by dividing sample data in the feature space based on their category while aiming to maximize the distances to the nearest training-data points. This classification model is considered most popular for traditional facial expression classification tasks [55].
  - **Classification strategy:** This parameter describes whether the multiclass problem should be solved according to the one-vs-rest or one-vs-one scheme.
  - **Regularization parameter:** The regularization parameter specifies the trade-off between the avoidance of missclassified training samples and a larger-margin separating hyperplane. The parameter grid is set to 10^x with x ranging from -5 to 5 in integer increments.
  - **Kernel:** Describes which kernel function is used. It is tested during hyperparameter tuning, whether a linear kernel function, a third-degree polynomial kernel function, or a radial basis function kernel is best suited for classifying the facial expressions.
  - **Kernel coefficient:** Specifies the kernel coefficient necessary for the radial basis function and the polynomial kernel function. The scikit-learn [60] implementation supports the options "auto" (inverse of the number of features) and "scale" (inverse of the product of the number of features and the variance of the test set).

3.2.2 Emotion Detection from Facial Landmarks using Coordinate Patterns and Deep Learning

Inspired by DTAGN [57] and DGNN [49] this classifier design tries to map the patterns formed by facial landmarks directly to the expressed emotion. Thus, the emotion detector uses the coordinates of facial landmarks as input. Similar to DTAGN [57] and DGNN [49] a deep learning approach is used for the development. Experiments are conducted on which facial landmarks result in the best detection performance to find the best model for this task.

- 68 **landmarks:** All salient facial key points emitted by the facial landmark detector following the Multi-PIE 68 points mark-up (see Figure 3.4) are used as input for the emotion detector.
- 51 **landmarks:** As the contour of the face does not necessarily add additional information and thus might not be beneficial to the detector's performance, all facial landmarks making up the contour of the face are excluded, and only the remaining landmarks are used as input for the emotion detector (landmarks #18 through #68). The mark up is shown in Figure 3.5.
- 21 **landmarks:** To further reduce the dimension of the emotion detector's input vector, the facial landmarks used are limited to 21 landmarks that are particularly involved during emotion expression. This includes the landmarks of the eyebrows (#18, #20, #22, #23, #25, #27), the eyes (center between #38 and #39, #41 and #42, #44 and #45, #47 and #48), the tip of the nose (#31) and the main salient features of the mouth (#49, #52, #55, #58, #63, #67). All 21 facial landmarks are marked in Figure 3.6.

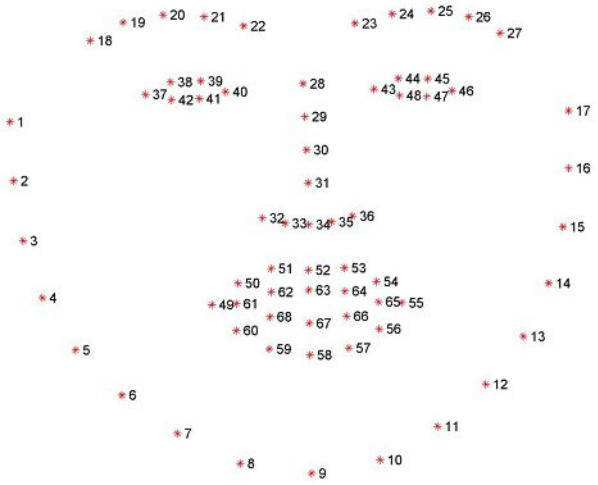


Figure 3.4: 68 points mark up [25]

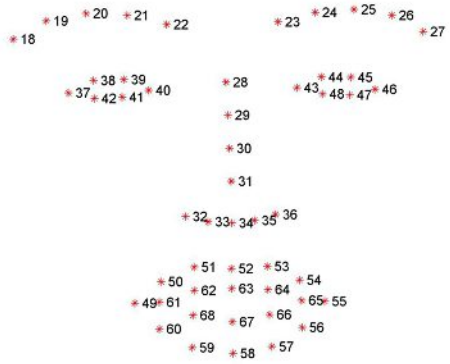


Figure 3.5: 51 points mark up [25]

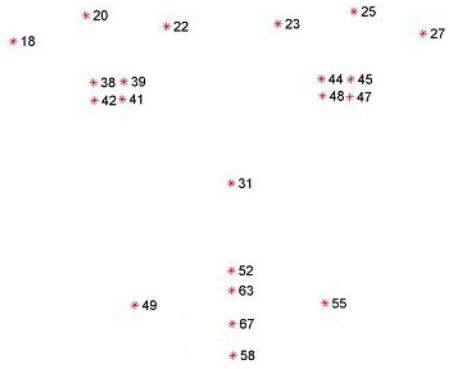


Figure 3.6: 21 points mark up [25]

3.2.2.1 Image Databases and Data Aaugmentation

The Radboud Facial Expression Database [71] is used for training and testing the emotion detector. During training, only frontal facial views are used since profile views partly hide facial landmarks. Furthermore, the resulting dataset is shuffled and split into training (70 %), validation (15 %), and testing set (15 %).

The facial expression database provides relatively few images, thus to prevent the deep learning network from overfitting, the training set is altered to increase the number of samples [57]. To do so, all images of the training set are flipped and in addition each image is rotated by every angle in $\{-15°, -10°, -5°, \pm 0°, 5°, 10°, 15°\}$, resulting

in fourteen variations of each image in the test set. Furthermore, Gaussian noise $N(0, \sigma^2) = N(0, 0.01^2)$ is added to each landmark coordinate.

$$\tilde{x}_i = \bar{x}_i + z_i \tag{3.1}$$

$$z_i \sim N(0, \sigma^2) = N(0, 0.01^2) \tag{3.2}$$

3.2.2.2 Preprocessing

The XY-coordinates of the landmarks have large value variations and thus are not suited as direct input for a deep learning network [57]. As a result, normalization of the input data is needed. The same normalization procedure used by Heechul et al. is [57] applied to the input landmarks (see equation 3.3). First, the data is centered by the coordinates of the nose's tip (landmark number 31). Second, the centered coordinates are normalized by dividing each coordinate by the standard deviation of the current frame's corresponding coordinate. Since the tip of the nose is used for the normalization process, the coordinates of landmark #31 are not used as input for the emotion detector.

$$\bar{x}_i = \frac{x_i - x_{31}}{\sigma_x} \tag{3.3}$$

$$\bar{y}_i = \frac{y_i - y_{31}}{\sigma_y} \tag{3.4}$$

3.2.2.3 Network Design and Training

Current state-of-the-art emotion detectors focus on image and image-sequence-based models. However, image or image-sequence-based detectors are not compliant with the pedagogical concept of Robo-Smile. Different network architectures are evaluated to find a model capable of classifying expressed facial expressions. An overview of trained and tested networks as well as their structure is listed in Table 3.3. The columns describe the different network architectures, which are numbered consecutively and differ in the number of layers and the individual layers' structure. The rows describe the consecutively numbered layers.

Dropout layers are used during training, which randomly delete input units to prevent overfitting and improve the given network's generalization ability. Batch Normalization and ReLU are used after each dense layer to improve the network's convergence. Furthermore, similar to the facial landmark detector training, batch sizes of 64 samples and Adam optimization are used during the training of all variations of input features and network architectures.

Similar to the implemented facial landmark detectors, the learning rate is decreased automatically when the validation loss stops improving. Furthermore, if decreasing the learning rate does not improve the learning progress, early stopping of the training is used.

Table 3.3: Overview on structure of trained and tested emotion detector networks.

Layer #	**Network 1**	**Network 2**	**Network 3**	**Network 4**	**Network 5**	**Network 6**
1	Dense (7 units)	Dense (124 units)	Dense (1024 units)	Dense (1024 units)	Dense (2048 units)	Dense (100 units)
2	Batch Normal-ization	Dropout (5 %)	Dropout (5 %)	Dropout (5 %)	Dropout (5 %)	Dropout (5 %)
3	Softmax	Batch Normal-ization	Batch Normal-ization	Batch Normal-ization	Batch Normal-ization	Batch Normal-ization
4		Dense (7 units)	Dense (7 units)	Dense (1024 units)	Dense (1024 units)	Dense (100 units)
5		Batch Normal-ization	Batch Normal-ization	Dropout (5 %)	Dropout (5 %)	Dropout (5 %)
6		Softmax	Softmax	Batch Normal-ization	Batch Normal-ization	Batch Normal-ization
7				Dense (7 units)	Dense (7 units)	Dense (7 units)
8				Batch Normal-ization	Batch Normal-ization	Batch Normal-ization
9				Softmax	Softmax	Softmax

3.3 Application Considerations

The feedback system must have a minimum frame-rate of 15 fps to allow the perception of continuous movement, which is difficult to achieve with sequential application of face, facial landmark, and emotion detection. The overall system design emphasis lies on reducing the computation time, which is accomplished by two main concepts.

- **Alternating detectors:** The detection of the face and facial landmarks are computationally expensive, as seen in Table 4.18. Thus the sequential application on each frame is not possible. The computation time can be cut in half by alternating the face detection and the facial landmark detection on every other frame. When a new frame is transmitted via the webcam stream, the face detector is applied. If a face is found, the bounding box is kept the same for the second transmitted frame. The bounding box's content is extracted from the second transmitted frame and passed on to the facial landmark detector. Since the face is unlikely to leave the

face bounding box between two consecutive frames with a frame rate above 15 fps, the assumption can be made that the bounding box remains valid.

- **Optical flow:** The facial landmarks must be computed and marked in each individual frame to retain the perception of fluid movement. However, the frame-wise computation of facial landmarks is unfeasible due to the limited computation time. Thus, the computationally cheaper method of tracking the previously detected facial landmarks using optical flow and, more specifically, Bouguet's implementation of the Lucas Kanade Feature Tracker algorithm using pyramids [73] is used. During this step, the previous image and facial landmarks are used to track the new frame's points. One backtracking step is applied after each optical step to guarantee the optical flow step's accuracy. Therefore, the current frame and the resulting facial landmarks are tracked in regard to the previous image. Afterward, the results of the backtracking step are compared to the points of the facial landmark detector. If the points differ, the optical flow step is considered incorrect and discarded.

Figure 3.7 shows the flow graph of the resulting feedback mechanism. During the imitation task, the computer's video capture device is accessed, and frames are passed on for further processing. The face detector is applied on the first transmitted frame and, more importantly, on every frame with an even frame count. If a face is detected and the frame count is odd, the section of the image containing the face is passed on to the facial landmark detector. Therefore, the bounding box of the face detector of the previous frame is used. If the frame count is even, the facial landmarks are tracked using optical flow. Furthermore, the emotion detector is applied. The facial landmark detector's results applied on the previous frame are used for the tracking and as input for the emotion detector. The frame count is increased after the processing of every frame with a detected face. Thus, the face detector is applied on every transmitted frame when no face is detected since the counter is not increased. If no face is detected, no further processing is necessary. This is done to detect new faces faster and to keep the computation time per frame stable. The counter is reset continuously to prevent overflowing.

3.4 Overall Concept

As required by the pedagogical concept, Robo-Smile's control mechanism is mainly composed of two steps, the facial landmark detection step and the facial expression detection step.

- **Facial landmark detection:** Aspects of the state-of-the-art facial landmark detector methodologies TCNN [20], MCL [21], TCDCN [22] [23], and Wing Loss [19] are combined in order to develop an original facial landmark detection algorithm improving the detection performance, whilst maintaining similar execution times. The suggested experiments cover different input image sizes, color and grayscale

input images, different network design bases and adjustments, two different activation functions (abstanh and ReLU), as well as various loss functions. Furthermore, Zhu et al.'s [64] 3D-Face-Model-based solution for increasing training samples in profile views is utilized for image augmentation.

- **Emotion detection:** The emotion detection is dependent on the previous facial landmark detection since Robo-Smile's emotion feedback system utilizes the preference of persons with ASD of keeping up routines and firm procedures [2] by linking specific facial landmark patterns to facial expressions. Thus, the facial expression classifier must be built upon the previous facial landmark detection step. Therefore, two main approaches are envisaged to find the methodology best suited for correctly classifying expressed emotions based on the locations of facial landmarks or information such as distances, which can be derived from these coordinates.
  - **Emotion detection from Facial Landmarks using AUs:** By utilizing the AUs described by the FACS, an emotion detection algorithm is suggested, which bases on distances in the face associated with specific AUs active during the expression of emotions. The F-values are computed using ANOVA to find the best distance features. The optimal classifier is determined by comparing the performance of a tree classifier, an AdaBoost classifier, a random forest classifier, a k-nearest neighbors classifier, and a SVM. The hyperparameters of the corresponding classifier are optimized using a cross-validated grid search over a pre-defined parameter grid.
  - **Emotion detection from Facial Landmarks using Coordinate Patterns and Deep Learning:** Inspired by the deep learning facial expression classifier DTAGN [57] and DGNN [49], several network designs are proposed, which use the facial landmark coordinates as input for classifying facial expressions.

  The Radboud [71] facial expression database is used for training and testing the emotion classifier, as it comprises acted facial expressions and contains images of children.

To guarantee execution times allowing the feedback mechanism to run with a minimum of 15 fps, sequential executions of the face, facial landmark, and emotion detectors might not suffice. Thus an architecture is described, allowing the decrease of execution time by alternating the detectors and using a computationally cheaper optical flow facial landmark tracking step whenever the facial landmark detector is not applied.

All described experiments aim to optimize Robo-Smile's control mechanism and require an evaluation not only with respect to the detection accuracy but also in terms of execution times, as a compromise between accuracy and speed must be achieved.

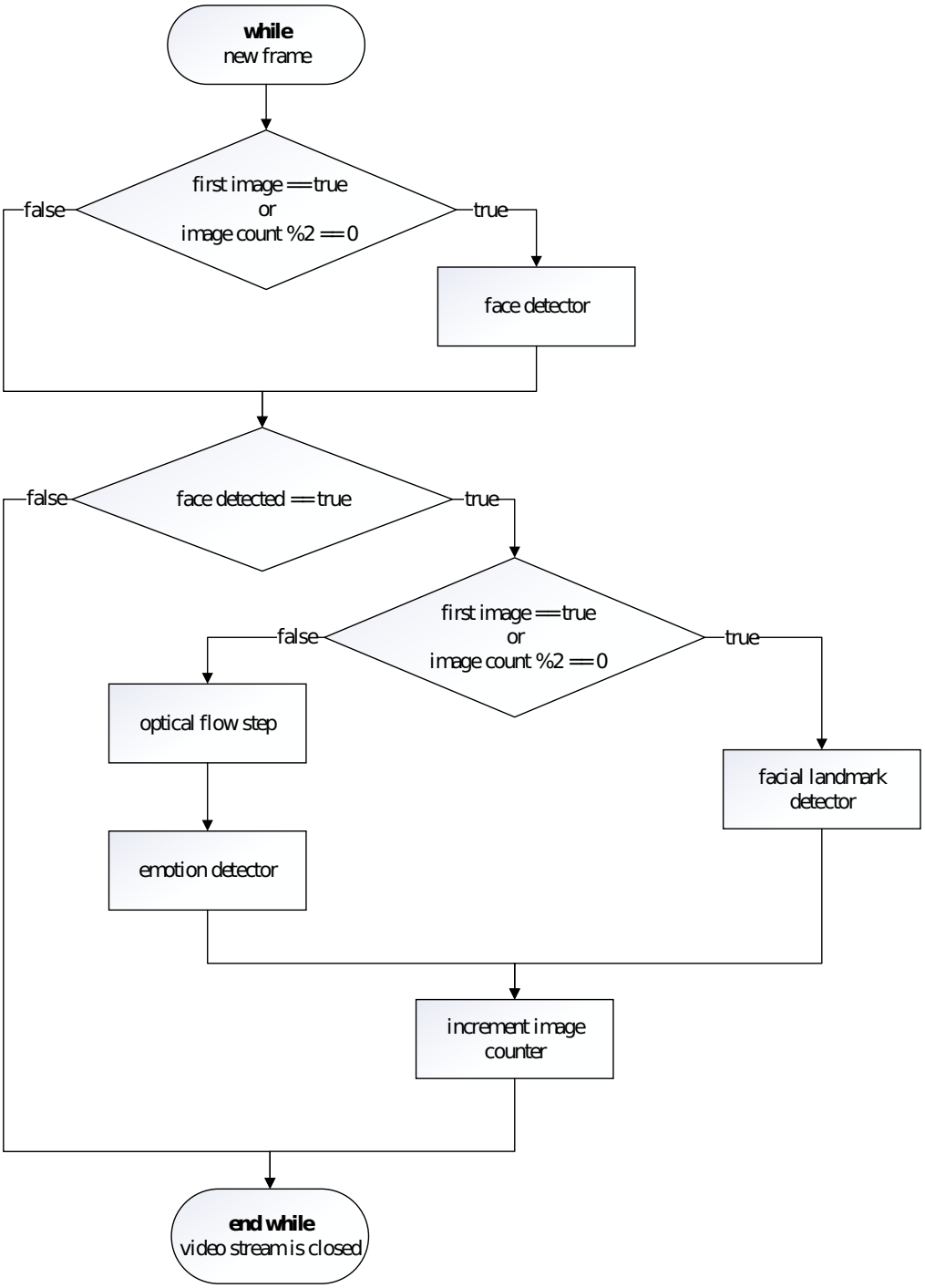


Figure 3.7: Flow graph of the feedback mechanism showing the process of alternating detectors.

CHAPTER 4

Results

4.1 Facial Landmark Detection

4.1.1 Face Detection

As facial landmark detectors operate on facial images, a robust and accurate face detection is needed as a first step. OpenCV [59] offers two different face detectors, which are tested using the 300W-LP database [64], which consists of the augmented 300-W database [65] [25] [66] using Zhu et al.'s 3D-Face-Model-based solution for increasing training samples in profile views. If 90 % of facial landmarks are inside the bounding box resulting from the face detector, a face is considered detected. Overall, 61225 images are used for testing the detection performance. The results are listed in Table 4.1 and show that the deep-learning-based face detector is superior compared to the Haar-cascade-based face detector. The deep-learning-based face detector is better suited for detecting faces during large pose variations since the Haar-cascade-based detector only manages to detect 29378 of the overall 61225 images, which leads to a relative detection rate of 47.98 %. In comparison, the deep-learning-based detector correctly detects faces in 58299 images, which corresponds to a relative detection rate of 95.22 %. Thus, the deep-learning-based face detector is used for pre-processing during the detection of facial landmarks.

Table 4.1: Detection rate comparison of a Haar-cascade-based and a deep-learning-based face detector. Both tested detectors are provided by OpenCV [59].

Face detector	**Detected faces**	
	Absolute	**Relative (in %)**
Haar-cascade-based	29378	47.98
Deep-learning-based	58299	95.22

4.1.2 Facial Landmark Detection Accuracy

The facial landmark analysis focuses on first evaluating the various detectors obtained by the different experiments listed in Table 3.1 and comparing the performance with the facial landmark detector provided by Dlib, which has been used in the previous version of Robo-Smile. Second, the derived facial landmark detector providing the best results is compared to state-of-the-art architectures.

For the performance analysis, the tested CNN designs are trained on the 300-W database [65] [25] [66] augmented using Zhu et al.'s 3D-Face-Model-based solution for increasing training samples with profile views. Testing is done on the public and private test sets of the 300W [65] [25] [66] database. The private test set consists of 300 images shot indoors and 300 images shot outdoors. However, the public test set can be separated into a common subset and a challenging subset. The common subset is the combination of the test sets of the Helen [63] and LFPW [62] database. The Helen dataset [63] consist of 330 test images, the LFPW [62] of 224 test images (originally 300 images, but only 224 re-annotated images are provided by IBUG [25]). As the landmark markup of the original Helen [63] and LFPW [62] database differs from the markup used for the 300-W challenge [65] [25] [66], the re-annotated landmark locations are used. The IBUG [25] test set, which is comprised of 135 face images, is considered to be the challenging subset of the 300-W public test set [65] [25] [66].

The error metrics most commonly used for evaluating and comparing facial landmark localization algorithms is the normalized error (see Equation 4.1) calculated over each sample of a given test set. The error is normalized using the IOD, which corresponds to the distance between the eyes' outer corners. N denotes the number of facial landmarks used. P_i are the coordinates of the i^{th} facial landmark, with $i = 37$ and $i = 46$ being the landmarks of the outer corners of the eyes. $\hat{P}_i$ denotes the coordinates of the ground truth facial landmark locations.

$$\mathrm{NE}_{IOD} = \frac{\sum_{i=1}^{N} \|P_i - \hat{P}_i\|_2}{\|P_{37} - P_{46}\|_2} \tag{4.1}$$

In literature, the normalized mean error or the normalized median error calculated over a given test set is used to compare facial landmark detectors. In order to compare the results of the designs listed in Table 3.1, the mean, the standard deviation and the median normalized error are computed over the test sets of the HELEN [63], LFPW [62], IBUG [25] and 300 W [65] [25] [66] databases.

In order to evaluate the accuracy of a given test set, the faces need to be extracted. The same face detector is used to test the self-trained facial landmark detectors and Dlib's 68 point shape predictor. However, it must be noted that the face detector did not manage to predict all faces correctly. If the bounding box resulting from the face detector does not include all 68 ground truth facial landmarks, the image is neglected for the given analysis. Thus the given test set is reduced to facilitate the testing under similar conditions as

during the planned application. The detection of the face is necessary before the facial landmark localization step. The failure rate of the deep-learning-based face detector for each test set is listed in Table 4.2.

Table 4.2 describes the overall image count and how many images are falsely detected for each test set. Each row corresponds to one test set. The image libraries HELEN [63] and LFPW [62] contain image conditions, which are considered easy, whilst the databases IBUG [25] and 300 W [65] [25] [66] (indoor and outdoor) databases contain images with conditions considered more difficult. This is also reflected in the deep-learning-based face detector's performance, as the image databases considered more difficult lead to higher failure rates.

Table 4.2: The failure rate of Open CV's deep-learning-based face detector for the HELEN [63], LFPW [62], IBUG [25] and 300W [65][25][66] test set.

Test set	**Overall image count**	**Count of falsely detected faces**	
		Absolute	**Relative in %**
HELEN	330	48	14.55
LFPW	224	47	20.98
IBUG	135	77	57.04
300W (indoor)	300	155	51.67
300W (outdoor)	300	191	63.67

Table 4.3 shows the calculated mean, standard deviation, and median normalized error over the tests sets HELEN [63], LFPW [62], and IBUG [25]. Each row describes a different detector used during computation of the normalized error. Except for the Dlib detector, all detectors are numbered. The numbers correspond to the detector's design. Similarly, Table 4.4 shows the calculated mean, standard deviation, and median of the normalized error over the IBUG 300-W facial landmark dataset [65] [25] [66], which is further split into indoor and outdoor images. As seen in Table 4.3 the rows describe the detectors used, where the detectors are numbered corresponding to the design the detector resulted from, with the only exception being the Dlib detector. The Dlib detector is used for comparison purposes since this facial landmark detector is used during Robo-Smiles's first prototype. Thus a comparison to the existing system is possible. The Dlib detector is an implementation of Kazemi and Sullivan's proposed one-millisecond face alignment and is trained on the IBUG 300-W facial landmark dataset [65] [25] [66]. The best median, standard deviation, and median measurements of the normalized error overall compared detectors are written in bold letters.

Table 4.3 shows that the standard deviation of all self-trained models is significantly smaller, demonstrating a better precision of the deep-learning-based methods than the regression-tree-based Dlib predictor. Model #12 shows an improvement in the standard deviation of the normalized error with respect to Dlib of 72.79 % computed over the HELEN test set, of 56.74 % computed over the LFPW test set, and of 72.19 % computed over the IBUG test set. However, the mean and median error of Dlib's predictor are

Table 4.3: Mean, standard deviation and median normalized error calculated over the test sets of the HELEN [63], LFPW [62], IBUG [25], and 300W [65][25][66] databases.

#	HELEN			LFPW			IBUG		
	mean	**std**	**median**	**mean**	**std**	**median**	**mean**	**std**	**median**
1	3.69	1.11	3.50	3.59	1.34	3.19	7.40	3.09	6.24
2	3.79	1.16	3.54	3.66	1.44	3.26	7.24	2.70	6.22
3	3.64	1.09	3.50	3.51	1.28	3.11	7.03	2.53	6.23
4	3.84	1.19	3.63	3.61	1.32	3.28	7.26	2.90	6.32
5	3.47	1.00	3.36	3.63	1.38	3.28	7.17	2.97	6.39
6	3.88	1.27	3.66	3.94	1.48	3.65	7.55	3.29	6.71
7	3.83	1.28	3.50	3.90	1.59	3.45	7.05	3.19	6.11
8	4.56	2.07	4.12	5.44	2.52	4.95	9.67	5.88	7.72
9	3.54	1.01	3.36	3.40	1.15	3.16	6.49	2.52	5.60
10	5.47	2.05	5.05	5.56	2.20	5.34	9.67	4.86	8.49
11	3.18	0.85	3.09	3.07	1.04	2.81	5.99	2.40	5.31
12	3.05	**0.80**	2.93	2.89	**0.93**	2.69	**5.83**	**2.35**	**5.12**
Dlib	**2.34**	2.94	**1.68**	**2.65**	2.15	**2.19**	8.67	8.45	5.36

smaller for the databases HELEN [63] and LFPW [62] compared to the best model of design #12. The normalized mean error of model #12 with respect to Dlib increases by 30.34 % on the HELEN test set and by 9.06 % on the LFPW test set. Furthermore, the median normalized error of model #12 with respect to Dlib increases by 74.40 % on the HELEN test set and by 22.83 % on the LFPW test set. However, this is not true for the IBUG [25] test set since the self-trained model outperforms Dlib's predictor (increase in the mean performance of 32.76 % and in the median performance of 4.48 %).

The evaluation listed in Table 4.4 shows that model #12 results in the best mean performance for the private 300 W [65] [25] [66] test set during the given comparison of self-trained models and Dlib's shape predictor. Model #12 manages to reduce the calculated mean normalized error by 9.58 % on the indoor training set and by 15.28 % on the outdoor training set compared to the second-best mean measurements of the Dlib facial landmark predictor. The standard deviation of all models developed in this work is significantly smaller compared to Dlib's predictor. Model #11 shows a reduction of the standard deviation of 63.62 % calculated over the indoor test set and 66.94 % calculated over the outdoor test set with respect to Dlib. The model with the best mean performance results in a slightly higher standard deviation compared to model #11, however it still shows an average reduction of the standard deviation of 60.15 % calculated over the indoor test set and of 63.67 % calculated over the outdoor test set with respect to Dlib. However, Dlib shows better median detection rates than the deep-learning-based models, as model #12 results in a median measurement of 4.73 indoor and 4.65 outdoor. In contrast, Dlib results in a median measurement of 3.88 indoor and 4.22 outdoor. Since the median measurements are less prone to be affected by outliers than the mean

measurement, this and the detector's measured standard deviation suggest a higher failure rate of Dlib's detector compared to all self-trained models.

Table 4.4: Mean, standard deviation and median normalized error calculated over the private 300 W test set [65] [25] [66].

#	300 W (indoor)			300 W (outdoor)		
	mean	**std**	**median**	**mean**	**std**	**median**
1	5.98	2.55	5.44	6.28	2.88	5.57
2	6.17	2.78	5.58	6.43	2.90	6.08
3	5.80	2.22	5.33	6.06	2.36	5.84
4	6.09	2.44	5.50	6.20	2.58	6.01
5	5.78	2.33	5.14	6.04	2.72	5.71
6	6.19	2.81	5.50	6.31	2.78	5.75
7	6.22	2.92	5.74	6.33	2.57	5.68
8	7.22	3.27	6.60	7.33	2.78	6.99
9	5.62	2.15	5.19	5.88	2.23	5.33
10	9.72	4.86	8.64	9.93	4.91	8.57
11	5.25	**1.99**	4.92	5.40	**2.02**	5.03
12	**5.00**	2.18	4.73	**5.10**	2.22	4.65
Dlib	5.53	5.47	**3.88**	6.02	6.11	**4.22**

In summary, the results listed in Table 4.3 and Table 4.4 show that the self-trained model with design #12 yields the best results compared to all other self-trained models computed over all test sets. Thus the model is used for further evaluation enabling a better comparison with Dlib's shape predictor.

Assessing the error distribution for each image in a given test set, the advantage in the precision of the deep-learning-based detector with design #12 compared to Dlib's shape prediction becomes apparent as seen in Figure 4.1 and Figure 4.2.

Figure 4.1 shows the normalized error of each image sample visualized using box plots for the Dlib detector (green) and model #12 (orange) calculated separately over the public 300 W test set. The 300 W test set is analyzed over the common subset (HELEN and LFPW), the challenging subset (IBUG), and the full set. It can be seen that, especially for datasets that are considered difficult, the detection of Dlib's shape predictor fails more often, as more outliers exist. This behavior elucidates the fact that model #12 leads to an overall more precise detection, with a slight disadvantage in accuracy, as a minor bias can be observed compared to Dlib's shape predictor during test samples considered easy. However, during the evaluation on the challenging subset, #12 demonstrates superiority in precision and accuracy compared to Dlib.

Figure 4.2 shows the normalized error visualized using box plots for the Dlib detector (green) and model #12 (orange) calculated separately over the private 300 W test set. The 300 W private test set is separated into the indoor and outdoor subset. Similar to

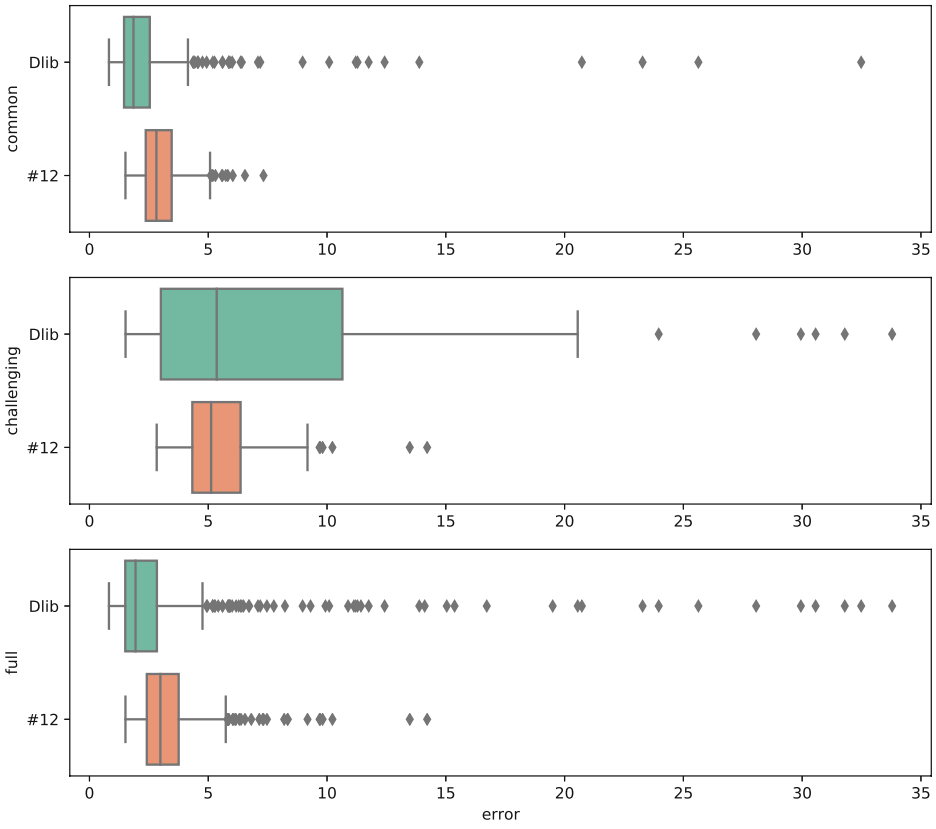


Figure 4.1: Boxplot graph of the normalized error of each sample comparing the accuracy of facial landmark detectors provided by Dlib and model #12, tested on the 300 W public test set [65] [25] [66].

the public test set analysis, it can be seen that, especially for datasets considered more difficult, the detection of Dlib's shape predictor fails more often, as more outliers exist. This suggests superiority in the precision of model #12 compared to Dlib. However, Dlib's facial landmark detector is considered to be slightly more accurate for the private test set since the average normalized error is lower compared to detector #12.

The performance of the best self-developed model (#12) is compared with state-of-the-art deep learning models and Dlib's shape predictor. In contrast to the previous evaluation, the face detector is not used for pre-processing. The reason for this is the high failure rate (see Table 4.2), which would falsify the comparison with state-of-the-art methods as not the full test set would be used. Thus, the bounding boxes provided by the databases are taken instead of the bounding boxes computed by the face detector. The bounding

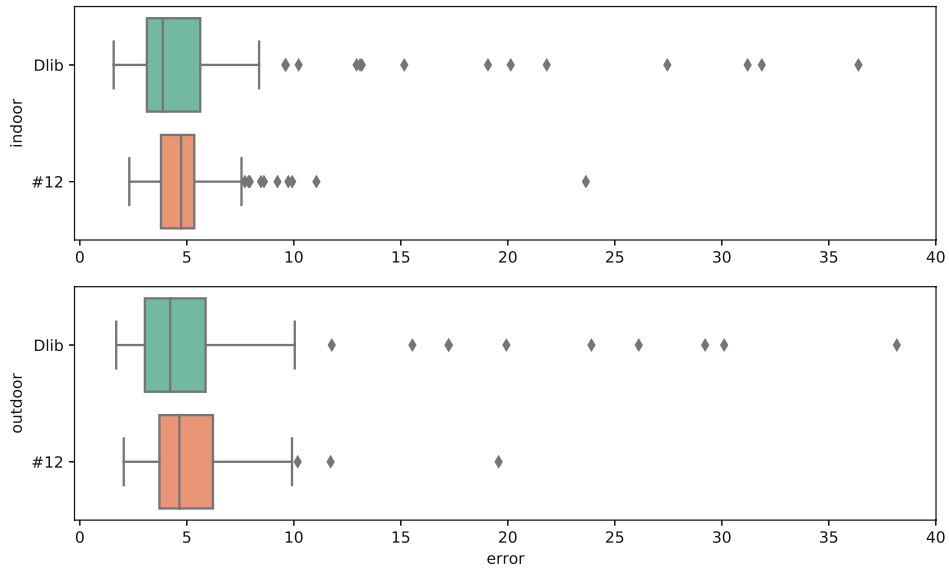


Figure 4.2: Boxplot graph of the normalized error comparing the accuracy of facial landmark detectors provided by Dlib and model #12, tested on the 300 W private test set [65] [25] [66].

boxes computed by the detector are bigger compared to the ground truth bounding boxes. Thus the ground truth bounding box is enlarged. In X-direction, a border of 10 % of the ground truth's bounding box width is added on the left and right sides of the face. In Y-direction, a border of 5 % is added to the chin area and a border of 10 % to the forehead.

Table 4.5 shows the mean normalized error computed over the public 300 W test set [65] [25] [66] for state-of-the-art facial landmark detectors and the detector #12 developed in this work. For the common subset, the Dlib detector reaches best results with a mean of 3.04, closely followed by Dong et al. Convolutional Pose Machine (CPM) with added Supervision-by-Registration (SBR) achieving a mean normalized error of 3.28. Model #12 has an increased mean error by 19.08 % compared to Dlib and a 10.37 % increase compared to CPM with added SBR. However, when looking at the mean normalized error for the challenging subset model #12 outperforms the current state-of-the-art detectors by 51.72 % in comparison to Dlib and by 7.52 % in comparison to CPM with added SBR. During observation of the full 300 W public test set, detector #12 delivers the second-best detection accuracy. CPM with added SBR achieves the smallest average normalized error with 4.10. It must be noted that no execution times can be cited for CPM with added SBR. However, since the model uses a CPM, which has an execution time of 33 ms tested on a GPU (GTX 1080 Ti) [74], it can be assumed

that the model can not be used for live-feedback on devices without a powerful GPU. Thus, model #12 is the detector with the best detection performance on the full test set while being compliant with the application's restrictions as part of Robo-Smile.

Table 4.5: Mean error normalized using the IOD for state-of the art facial landmark detectors. All measurements are taken from Zhang et al. [23] and Dong et al. [70].

Method	300 W (public)		
	Common	**Challenging**	**Full**
CDM [75]	10.10	19.54	11.95
DRMF [76]	6.65	19.79	9.22
RCPR [77]	6.18	17.26	8.35
CFAN [30]	5.50	16.78	7.69
ESR [78]	5.28	17.00	7.58
SDM [33]	5.57	15.40	7.52
LBF [79]	4.95	11.98	6.32
CFSS [80]	4.73	9.98	5.76
TCDCN [22] [23]	4.80	8.60	5.54
MDM [40]	4.83	10.14	5.88
TSR [41]	4.36	7.56	4.99
CPM [81]	3.39	8.14	4.36
CPM+SBR [70]	3.28	7.58	**4.10**
Dlib [16]	**3.04**	14.52	5.29
# 12	3.62	**7.01**	4.28

Since some publicized models are not evaluated using the mean normalized error, Table 4.6 compares state-of-the-art methods using the median normalized error. For the common subset of the 300 W public test set, Dlib's shape predictor obtains the best median normalized error results with a measurement of 1.95. The model presented in this work (#12) has a larger median normalized error of 3.43 and therefore achieves moderate detection accuracy in comparison to all detectors listed in this table. However, for the challenging subset, Dlib achieves a median normalized error of 8.98, which is a worse detection accuracy compared to model #12, which achieves a median normalized error of 6.30. The model performing best on the challenging subset is the Convolutional Experts Constrained Local Model (CE-CLM) with a median normalized error of 5.35. This model manages execution times not allowing the application on live camera data. However, recent advancements of CE-CLM developed as part of the OpenFace 2.0 Toolkit optimize the model to achieve frame rates improving the original model by a factor of 30 (30-40 Hz frame rates running on a quad-core 3.5 GHz Intel i7-2700K processor, and 20 GHz frame rates on a Surface Pro 3 laptop with a 1.7 GHz dual-core Intel i7-4650U processor) [82].

Overall, the model presented in this thesis reaches moderate median normalized error results and good mean normalized error results compared with state of the art. Hence, this model improves the previous implementation of Robo-Smile's control mechanism and

Table 4.6: Median error normalized using the IOD for state-of the art facial landmark detectors. All measurements are taken from Zadeh et al. [83].

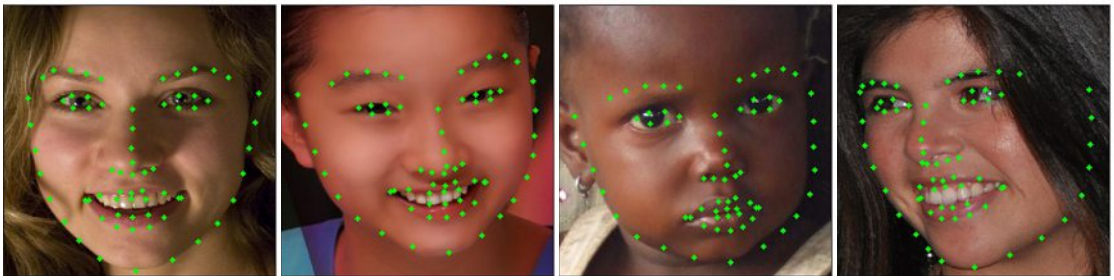
Method	300 W (public)	
	Common	Challenging
CLNF [32]	3.47	6.37
DRMF [76]	4.97	10.36
CFSS [80]	3.20	5.97
TCDCN [22] [23]	4.11	6.87
3DDFA [44]	7.27	12.31
CE-CLM [83]	3.14	**5.38**
Dlib [16]	**1.95**	8.98
# 12	3.43	6.30

achieves state-of-the-art detection accuracy while being compliant with the execution time restrictions, which can be traced back to the planned application.

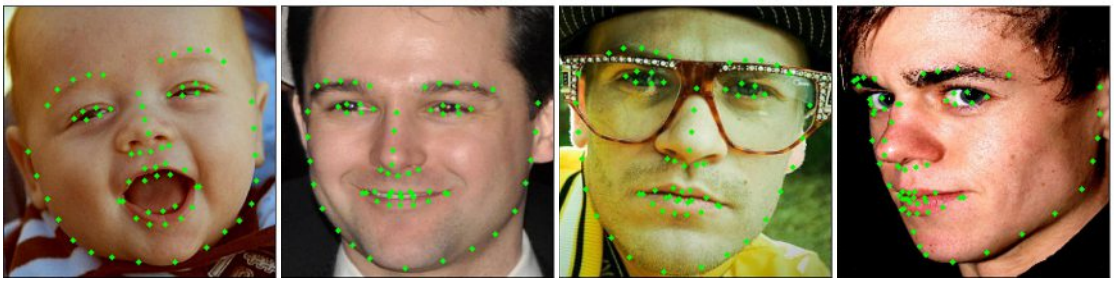
Finally, the performance of the detector (# 12) is shown on samples of the HELEN [63] (see Figure 4.3) and IBUG [25] (see Figure 4.4) test set in order to emphasize the detector's robustness to different environments. The image databases do not include information on the age, gender, or ethnicity of the portrayed people. Thus this information is manually estimated. The samples are chosen manually to cover images showing persons of different ages, genders, facial expressions, ethnicities, and head poses. Furthermore, images are selected showing different lighting conditions and partly covered faces. The detector's results are marked in green. Furthermore, all displayed images are cropped to show the output of the face detector only.

Due to the facial landmark detector's application as part of Robo-Smile, the detector must predict the fiducial points in children's facial images accurately. It is assumed that the images b, c, e, k, m and o of Figure 4.3 show faces of children of different age, pose, ethnicity and facial expression. None of the children are wearing glasses. However, parts of the facial contour are covered by hands in image m, and the tongue covers parts of the mouth contour in image o. Image c shows a light source to the child's right, while the lighting is frontal or close to frontal, in the remaining images containing children's faces. It must be noted that visual inspection shows no noticeable difference in the prediction accuracy of images containing children compared to facial images containing adults or seniors. Furthermore, no noticeable difference in accuracy between different ethnicities can be seen in these images.To show the detector's performance during conditions considered difficult, an excerpt of images of the IBUG [25] image database is visualized in Figure 4.4. Images a and b both show female faces with a light source to the left of the face. Due to the bad lighting conditions, parts of the contour of the face's left side cannot be identified, and parts such as the left eye and eyebrow in image b are challenging to detect. Image b also shows the limits of the facial landmark detector since there is an observable error

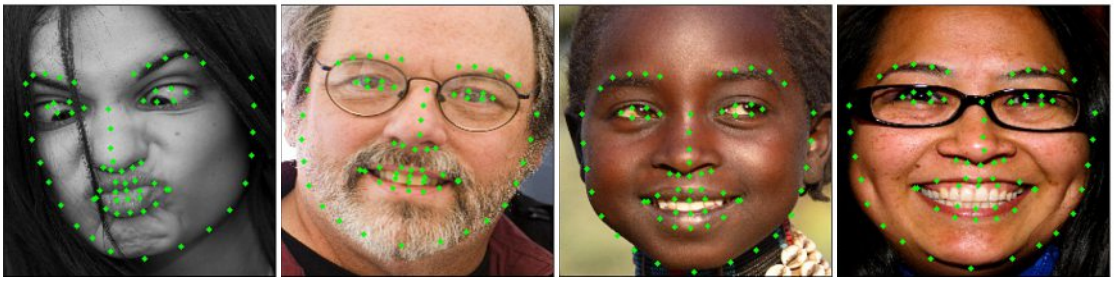
during the detection of the left eye and eyebrow. Images e, g, j, and p show people's side views with partly covered facial features due to the head pose.



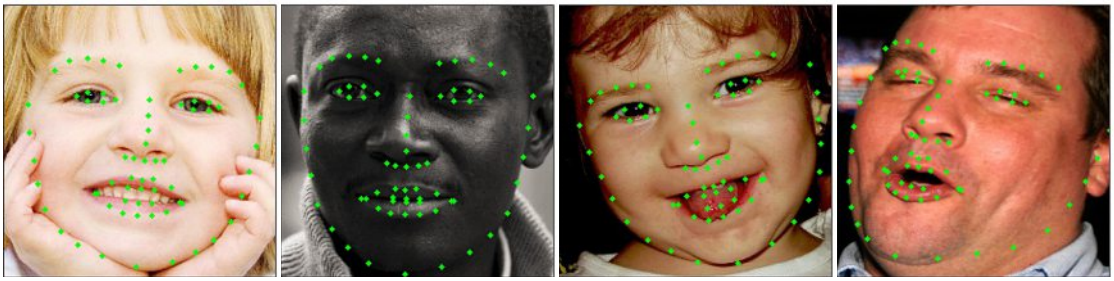
(a) 297461011_1.jpg (b) 3020509517_1.jpg (c) 3173028919_2.jpg (d) 3213167447_1.jpg



(e) 2981942448_1.jpg (f) 3213221949_1.jpg (g) 3022230063_1.jpg (h) 3016219064_1.jpg

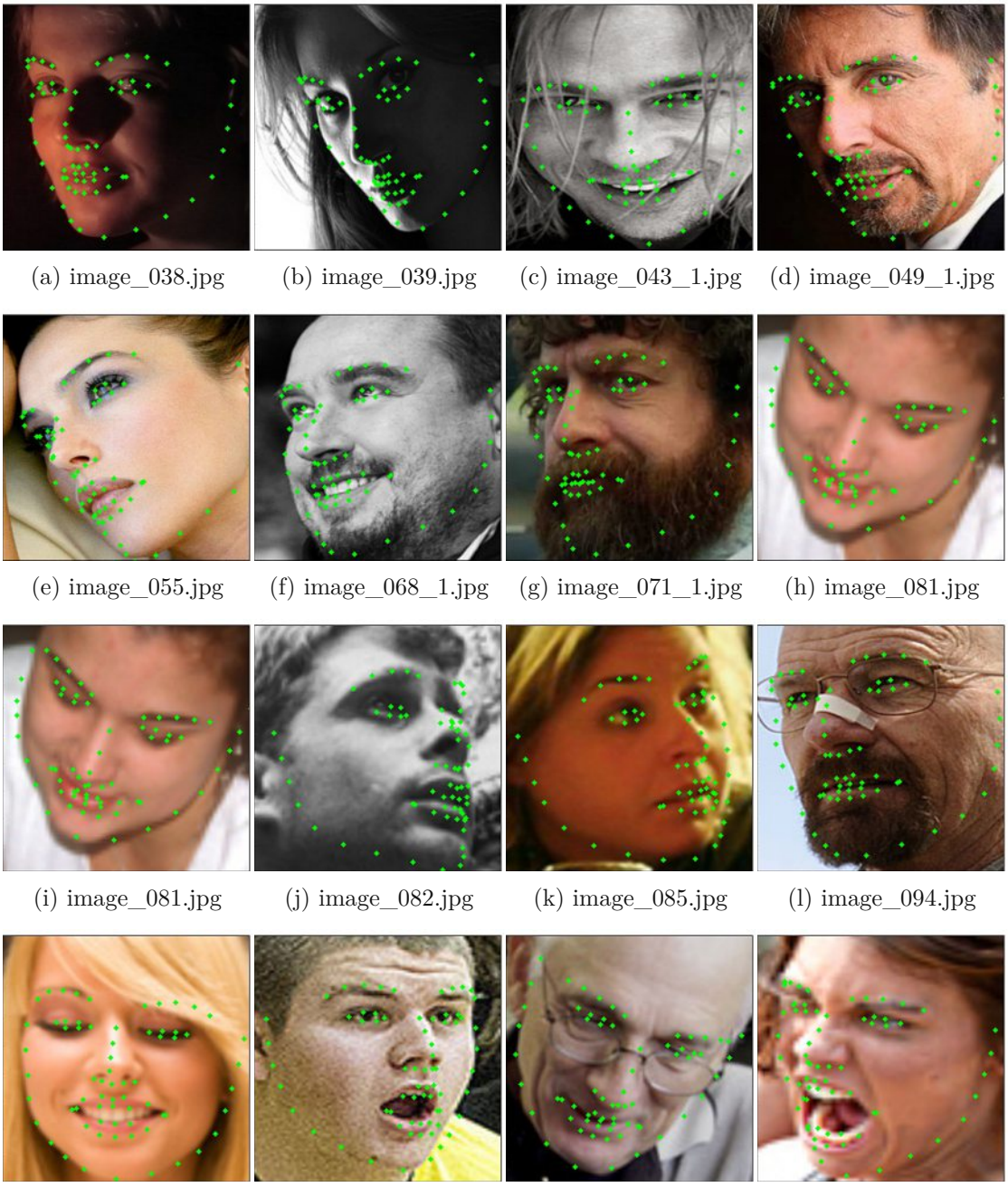


(i) 3175828165_1.jpg (j) 3052055699_1.jpg (k) 3239637522_1.jpg (l) 3138240967_1.jpg



(m) 3002568151_2.jpg (n) 3036934213_1.jpg (o) 3251963224_1.jpg (p) 3236428731_1.jpg

Figure 4.3: Detection performance on sample images of the HELEN [63] test set.



(a) image_038.jpg (b) image_039.jpg (c) image_043_1.jpg (d) image_049_1.jpg

(e) image_055.jpg (f) image_068_1.jpg (g) image_071_1.jpg (h) image_081.jpg

(i) image_081.jpg (j) image_082.jpg (k) image_085.jpg (l) image_094.jpg

(m) image_99.jpg (n) image_100_05.jpg (o) image_113.jpg (p) image_117.jpg

Figure 4.4: Detection performance on sample images of the IBUG [25] test set.

4.1.3 Timing Behavior of Facial Landmark Detector

Besides analyzing the detection error, the execution times of the self-trained models and Dlib's shape predictor are evaluated (see Table 4.7). For this purpose, the execution times for each sample in the HELEN [63], LFPW [62], IBUG [25], and private 300 W test sets [65] [25] [66] are measured. The execution time measurements do not include the execution times of the pre-processing procedure containing the face detection and needed image processing steps, such as scaling and color conversion, but exclusively the execution time of the detector itself. The tests are computed using the scripting language Python and run on an Intel Core i7-6500U processor with a CPU frequency of 2.5 GHz.

In the course of evaluating the timing behavior of Dlib's shape predictor and all self-trained models listed in Table 3.1, the mean and standard deviation of the execution times are calculated for all test set images (see Table 4.7). Furthermore, the distributions of execution time measurements of the Dlib shape predictor and the best derived facial landmark detector (#12) are visualized in Figure 4.5. The evaluation highlights the superiority of Dlib's detector in regards to speed. Since Dlib's shape predictor is an implementation of Kazemi and Sullivan's proposed one-millisecond face alignment [17], an execution time of one millisecond is expected. However, an average execution time of three times the expected one-millisecond duration is measured. This discrepancy can be traced back to implementation details and the environment in which the evaluation is run.

As seen in Table 2.1, the current state-of-the-art deep-learning-based facial landmark detectors have not yet been able to reach execution times similar to the regression-based [29] detector proposed by Kazemi and Sullivan. However, it must be noted that all self-trained models are suited for running on a CPU while reaching average execution times exceeding the lower limit of 15 fps. Thus the perception of movement continuity during frame-wise video processing is achieved.

Table 4.7: Mean and standard deviation computed over all samples of the HELEN [63], LFPW [62], IBUG [25], and private 300 W test sets [65] [25] [66] running on a CPU with a frequency of 2.6 GHz. Furthermore, the average fps are listed.

#	Execution time		
	mean in s	**std in s**	**fps**
1	0.0516	0.0117	19.38
2	0.0519	0.0084	19.25
3	0.0517	0.0081	19.34
4	0.0514	0.0030	19.44
5	0.0521	0.0109	19.21
6	0.0268	0.0029	37.37
7	0.0274	0.0140	36.46
8	0.0338	0.0124	29.55
9	0.0294	0.0040	34.06
10	0.0297	0.0055	33.65
11	0.0298	0.0123	33.54
12	0.0299	0.0125	33.41
Dlib [16]	**0.0030**	**0.0001**	**329.82**

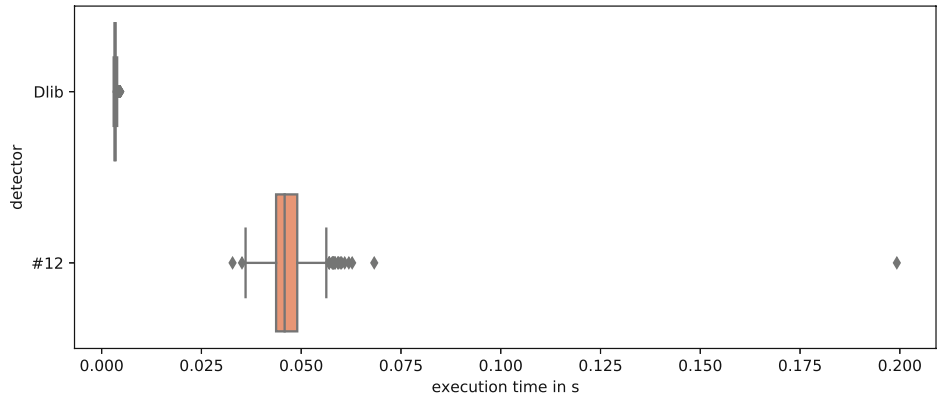


Figure 4.5: Boxplot graph of the execution times measured over all samples of the HELEN [63], LFPW [62], IBUG [25], and private 300 W test sets [65] [25] [66] running on a CPU with a frequency of 2.6 GHz.

4.2 Emotion Detection

4.2.1 Emotion Detection from Facial Landmarks using AUs

4.2.1.1 Feature Selection

The emotion classifier should use distances between facial landmarks as input. Therefore, these facial distances must be able to describe the main facial changes. Since several facial distances can describe these movements, the distances with the highest F-Scores are chosen, as they are most promising to describe the facial changes associated with the corresponding emotion. The result of the ANOVA test is shown in Figure 4.6. It must be noted that all parameters have F-scores proving their statistical significance for classifying emotion with a confidence level of 99 % since all corresponding p-values are significantly smaller than 0.01 (see Table 4.8).

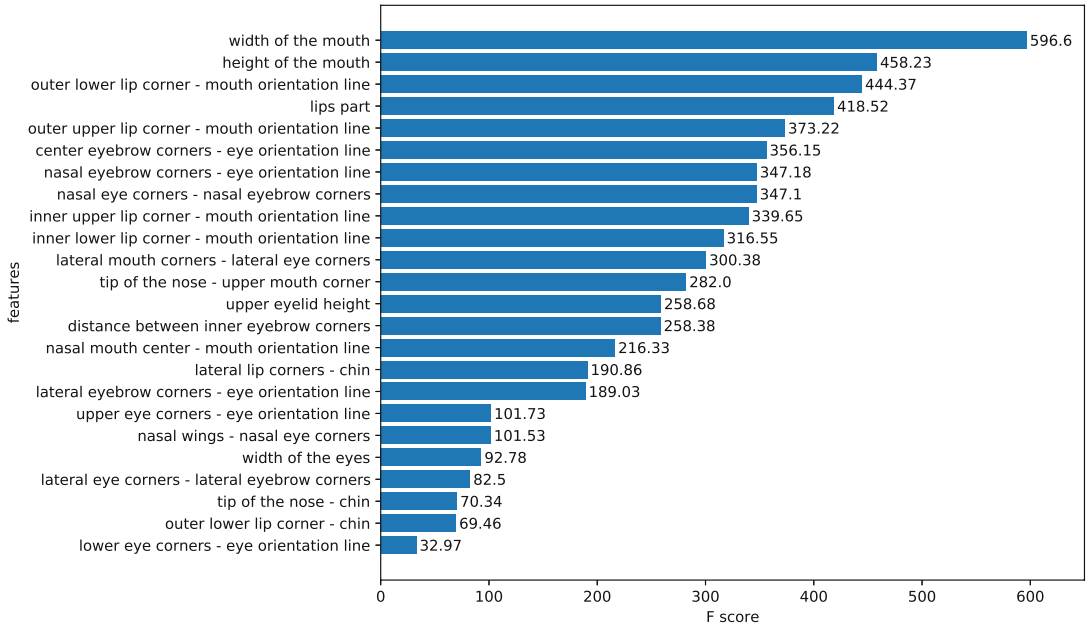


Figure 4.6: Results of the ANOVA test used for feature selection.

- **Eyebrows:**
  - The positions of the nasal eyebrow landmarks (#22 and #23) must be described since these points are raised during the expression of sadness, fear, and surprise and lowered during the expression of anger. These movements can be described by the features “nasal eyebrow corners - eye orientation line” and “nasal eye corners - nasal eyebrow corners”. Since, the F-score of the feature “nasal eyebrow corners - eye orientation line” is slightly higher (F-score ≈

347.18) compared to the parameter “Nasal eye corner - nasal eyebrow corner” (F-score ≈ 347.10) the distance “Nasal eyebrow corner - eye orientation line” is selected.

- The locations of the center landmarks (#20 and #25) must be described, since these points are raised during the expression of surprise and fear, and lowered during the expression of anger. The movement of the center eyebrows can be described by the feature “center eyebrow corners - eye orientation line”, which has a F-score of approximately 356.15.
- The coordinates of the lateral eyebrow landmark (#18 and #27) must also be described since these points are raised during the expression of surprise and fear and lowered during the expression of anger. The features “lateral eye corners - lateral eyebrow corners” (F-score ≈ 82.50) and “lateral eyebrow corners - eye orientation line” (F-score ≈ 189.03) can be used to describe these movements. Due to the higher F-score the distance “lateral eyebrow corners - eye orientation line” is selected.
- The eyebrows are pulled together during the expression of sadness, fear, and anger, leading to a decreased distance between the nasal eyebrow corner landmarks (#22 and #23), which are described by the feature “distance between inner eyebrow corners” (F-score ≈ 258.38).

Summarizing, the parameters “nasal eyebrow corners - eye orientation line”, “center eyebrow corners - eye orientation line”, “lateral eyebrow corners - eye orientation line”, and “distance between inner eyebrow corners” are selected for the description of the eyebrow movements.

- **Eyes:**
  - The upper eyelids are raised during the expression of fear, surprise, and anger, which can be described by the parameter “upper eye corners - eye orientation line” (F-score ≈ 101.73). The upper eyelid movements can also be described using the average height of the eyelid, represented by the parameter “upper eyelid height”. However, this parameter does not describe the eyelid directly but with respect to the eyebrows. Due to this parameter’s high F-score (F-score ≈ 258.68) compared to all scores of features describing the eye movements and to depict the interaction between eyebrow and eyelid, this feature is selected.
  - Furthermore, the lower eyelids are raised during the expression of fear. Hence, the distance “lower eye corners - eye orientation line” is introduced. The associated F-score for this parameter is comparably low (F-score ≈ 32.97). However, since the parameter is the only one capable of describing this subtle facial movement, it is selected for further use during classification.
  - During the expression of happiness, the inner and outer orbicularis oculi muscles are contracted, leading to an increased eye size. Which can be

described by the distances, width, and height of the eyes. The width of the eyes is described by the feature "width of the eye" (F-score ≈ 92.78). The eyes' height can be composed by the lower and upper eye halves described by the parameters "upper eye corners - eye orientation line" and "lower eye corners - eye orientation line". Thus no additional parameter is selected.

Summarizing, the parameters "upper eye corners - eye orientation line", "upper eyelid height", "lower eye corners - eye orientation line", and "width of the eye" are selected.

- **Mouth:**
  - The lip corners are pulled downwards during the expression of sadness and pulled upwards during happiness. The parameters "inner upper lip corner - mouth orientation line" (F-score ≈ 339.65), "inner lower lip corner - mouth orientation line" (F-score ≈ 316.55), "outer upper lip corner - mouth orientation line" (F-score ≈ 373.22), "outer lower lip corner - mouth orientation line" (F-score ≈ 444.37), "lateral mouth corners - lateral eye corners"(F-score ≈ 444.37), "lateral lip corners - chin" (F-score ≈ 190.86), and "nasal mouth center - mouth orientation line" (F-score ≈ 216.33) can be used to describe this movement. The associated F-scores suggest that the usage of the outer upper and lower lip corners are most meaningful. Both values are chosen instead of only one because using both values allows the separate observation of the upper and lower lip. This would not be necessary to distinguish between happiness and sadness alone since the upper and lower lip behave similarly. However, it is assumed to help during the classification of the remaining emotions since the normalized parameter distribution shows distinctive differences between the emotions (see Figure 4.7 and Figure 4.8).
  - The lips are stretched during the expression of fear and narrowed during the expression of anger, which can be described using the feature "width of the mouth" (F-score ≈ 596.60). During anger, the lips are not only narrowed but also pressed together, leading to not only an increased mouth width but also height. This change can be described by the feature "height of the mouth" (F-score ≈ 458.23). The features "height of the mouth" and "width of the mouth" showed to be most meaningful compared to the other features, as their F-scores rank highest (see Figure 4.6).
  - The lips part during the expression of surprise, fear, and disgust, which can be described using the parameter "lips part" (F-score ≈ 418.52).
  - During the expression of disgust the upper lip is raised and the distance between the lower lip and tip of the nose decreases. This change can be described by the parameter "tip of the nose - upper mouth corner" (F-score ≈ 282.00).

Summarizing, the features "outer upper lip corner - mouth orientation line", "outer lower lip corner - mouth orientation line", "width of the mouth", "height of the

mouth", "lips part", and "tip of the nose - upper mouth corner" are selected for the classification procedure.

- **Nose:**
  - During the expression of disgust, the nasal wings are pulled upwards, which can be described using the parameter "nasal wings - nasal eye corners". The associated F-score is 101.53.
- **Jaw:**
  - The jaw is pushed up during the expression of sadness and lowered during surprise, fear, and disgust. To describe these movements the distance "tip of the nose - chin" (F-score $\approx$ 70.34) can be used. The upwards movement of the chin during the expression of sadness also leads to a narrowing of the distance between the chin and the outer lower lip corner (#58), which is described by the parameter "outer lower lip corner - chin" (F-score $\approx$ 69.46). However, the distribution of the feature "outer lower lip corner - chin" (see Figure 4.9) shows that the differences between sadness and all emotions except happiness are negligible since the curves mostly overlap. Thus, only the parameter "tip of the nose - chin" is selected to describe the jaw movements.

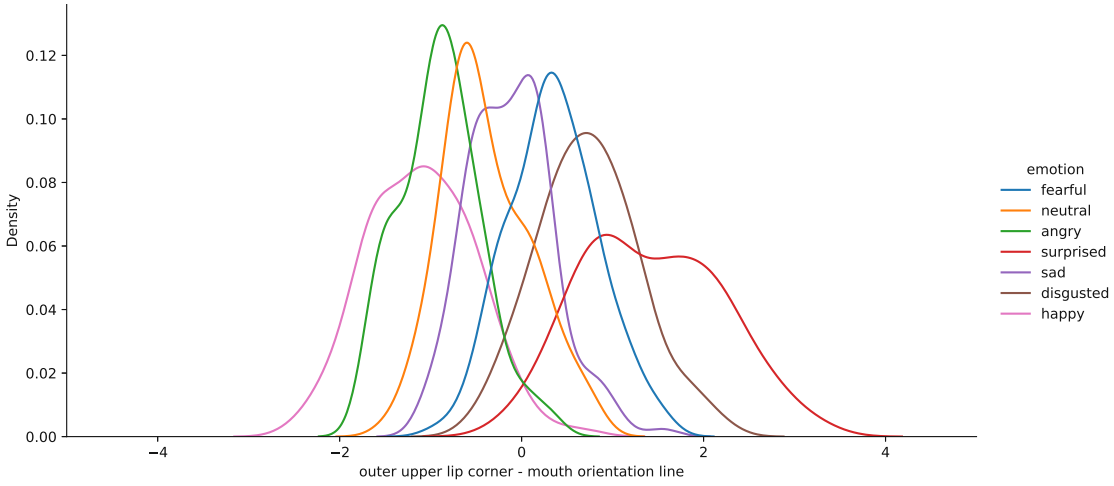


Figure 4.7: Distribution of the normalized parameter "outer upper lip corner - mouth orientation line".

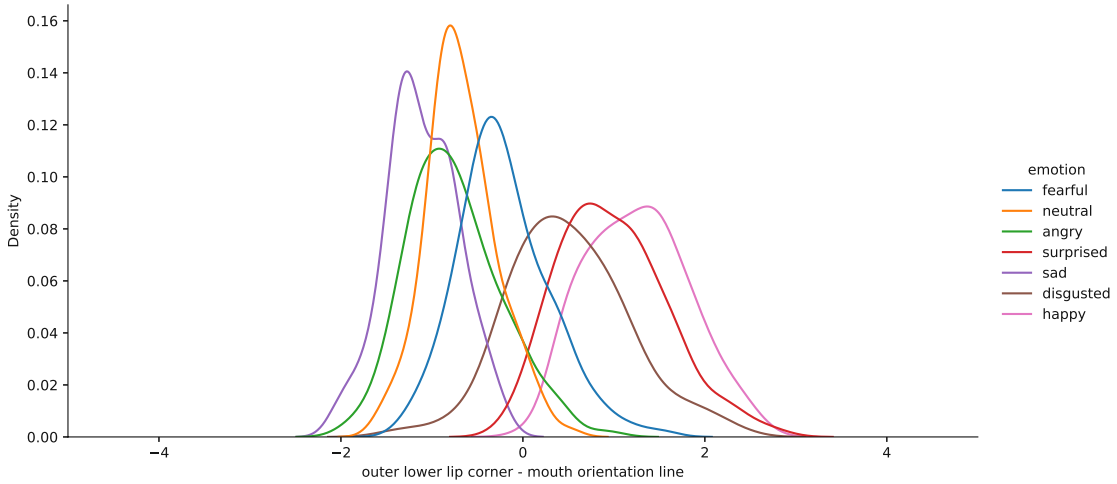


Figure 4.8: Distribution of the normalized parameter “outer lower lip corner - mouth orientation line”

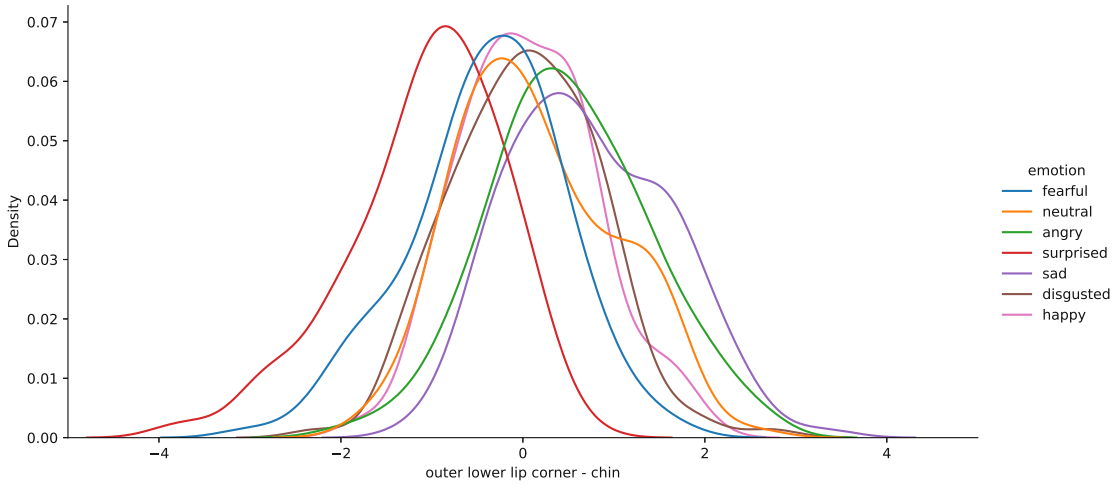


Figure 4.9: Distribution of the normalized parameter “outer upper lip corner - mouth orientation line”.

Table 4.8: Resulting F-scores and corresponding p-values of the ANOVA test used for feature selection.

Parameter	**F-score**	**P-value**
Lateral eyebrow corners - eye orientation line	189.0334	1.88E-159
Center eyebrow corners - eye orientation line	356.1529	1.76E-241
Nasal eyebrow corners - eye orientation line	347.1766	8.50E-238
Distance between inner eyebrow corners	258.3792	2.07E-197
Lateral eye corners - lateral eyebrow corners	82.5016	1.73E-83
Nasal eye corners - nasal eyebrow corners	347.0989	9.15E-238
Upper eyelid height	258.6843	1.45E-197
Nasal wings - nasal eye corners	101.5343	3.35E-99
Tip of the nose - upper mouth corner	281.9973	6.29E-209
Tip of the nose - chin	70.3435	8.20E-73
Upper eye corners - eye orientation line	101.7343	2.32E-99
Lower eye corners - eye orientation line	32.9730	2.71E-36
Width of the eyes	92.7801	4.07E-92
Lateral mouth corners - lateral eye corners	300.3789	1.78E-217
Inner upper lip corner - mouth orientation line	339.6458	1.18E-234
Inner lower lip corner - mouth orientation line	316.5480	1.01E-224
Outer upper lip corner - mouth orientation line	373.2172	2.59E-248
Outer lower lip corner - mouth orientation line	444.3748	1.23E-274
Width of the mouth	596.5971	0
Height of the mouth	458.2252	2.13E-279
Nasal mouth center - mouth orientation line	216.3345	3.22E-175
Outer lower lip corner - chin	69.4632	5.10E-72
Lateral lip corners - chin	190.8614	1.51E-160
Lips part	418.5208	1.91E-265

4.2.1.2 Emotion Detection Classifier Selection

Based on the feature selection process, 16 distance measurements are computed. All parameters are then mean-centered and scaled to unit variance. Different classifiers are evaluated to find the best classifier for the distinction between the basic emotions and neutral. All classifiers are tested using 61 images per emotion (overall 427 images) and evaluated using measurements describing the detector's precision, recall, and the F-beta score.

- **Precision:** Precision describes a detector's accuracy, meaning how many elements are correctly classified as a given class in relation to all elements that are correctly and falsely classified (see Equation 4.2).

$$\text{Precision} = \frac{\text{True Positive}}{\text{True Positive} + \text{False Positive}} \tag{4.2}$$

- **Recall:** Recall describes the relevant portion of elements which are classified as a given class (see Equation 4.3).

$$\text{Recall} = \frac{\text{True Positive}}{\text{True Positive} + \text{False Negative}} \tag{4.3}$$

- **F-beta score:** This parameter describes the weighted, harmonic mean of the precision and recall measurements [60]. During the analysis, precision and recall are weighted equally ($\beta = 1$) since both parameters are equally important for the classification of facial expressions (see Equation 4.4). If no additional weights are used during mean computation ($\beta = 1$) the resulting measurement is also referred to as F_1 score (see Equation 4.5). The F-beta and the F_1 score can result in values between 0 and 1, with a value of 1 being the best possible result.

$$\text{F-beta score} = (1 + \beta^2) \cdot \frac{\text{Presicion} \cdot \text{Recall}}{\beta^2 \cdot \text{Presicion} + \text{Recall}} \tag{4.4}$$

$$\text{F}_1 = 2 \cdot \frac{\text{Presicion} \cdot \text{Recall}}{\text{Presicion} + \text{Recall}} \tag{4.5}$$

Tree classifier:
The cross-validated grid search results in the usage of a maximal tree depth of 8, Gini impurity for measuring the quality of a split, and random splitting. The resulting tree classifier achieves an overall accuracy on the training set of 77.14 %. The evaluation on the test set yields an accuracy of 76.35 %, which is slightly lower than the accuracy of the training set. The main classification metrics are visualized in Table 4.9, showing the precision, recall and $\mathbf{F}_1$-score for each class. It can be noted that the classifier is unable to classify every emotion equally well. The detector achieves noticeable lower values for precision, recall, and $\mathbf{F}_1$-score for the emotions neutral, sadness, and fear in contrast to the remaining emotions of happiness, disgust, surprise, and anger.

Table 4.9: Classification report of tree classifier

Emotion	Precision	Recall	$\mathbf{F}_1$-score
Angry	0.77	0.75	0.76
Disgusted	0.9	0.87	0.88
Fearful	0.66	0.69	0.67
Happy	0.98	0.95	0.97
Neutral	0.53	0.66	0.58
Sad	0.71	0.56	0.62
Surprised	0.87	0.87	0.87

The confusion matrix shown in Figure 4.10 underlines the fact that the tree classifier struggles, especially with correctly classifying the expression of sadness. At the same time, emotions, such as happiness and surprise, are rarely mispredicted. Furthermore, the matrix gives insight into what types of errors the classifier is prone to yield. The confusion matrix shows that the emotion sadness is misclassified as neutral, fearful and angry, since only 56 % of images of persons expressing sadness are correctly classified as sadness, while 25 % are detected as neutral, 11 % as fearful and 8 % as angry. The expression of neutral is only classified correctly in 66 % of cases, whilst it is wrongly classified as anger in 13 %, as sadness in 11 %, and as fear in 10 % of the cases. Fearful is correctly classified in 69 % of the cases. The remaining cases are misclassified as surprise (13 %), disgust (5 %), sadness(3 %), and anger (2 %). Anger is correctly classified in 75 % with a tendency to being misclassified as neutral (15 %), sadness (8 %), and fear (2 %). This suggests that the classifier struggles with distinguishing the emotions of neutral, anger, and sadness, which can be attributed to only minor facial differences between these emotions. In contrast, the emotions happiness (correctly classified in 95 % of cases), surprise (correctly classified in 87 % of cases), and disgust (correctly classified in 87 % of cases) are associated with more distinct facial changes and achieve better classification accuracy.

Random forest classifier:
The parameters leading to the best results during cross-validated grid search are a maximum depth of 9, a maximal number of estimators of 50, and the usage of entropy-based information gain. The trained random forest classifier achieves an accuracy of 84.69 % on the training set and of 83.84 % on the test set. This shows an improvement in the detection accuracy on the test set of 9.81 % with respect to the trained tree classifier. The main classification metrics are visualized in Table 4.10 and show the precision, recall and $\mathbf{F}_1$-score for each class. The evaluation shows that the classifier manages to classify the emotion of happiness in all test images correctly. Furthermore, the emotions of disgust ($\mathbf{F}_1 = 0.96$) and surprise ($\mathbf{F}_1 = 0.9$) are correctly classified in almost all of the cases. Summarizing, an improvement in $\mathbf{F}_1$-score of all emotions can be observed compared to the usage of a single tree. However, the classifier still struggles with distinguishing neutral expressions correctly ($\mathbf{F}_1 = 0.68$).

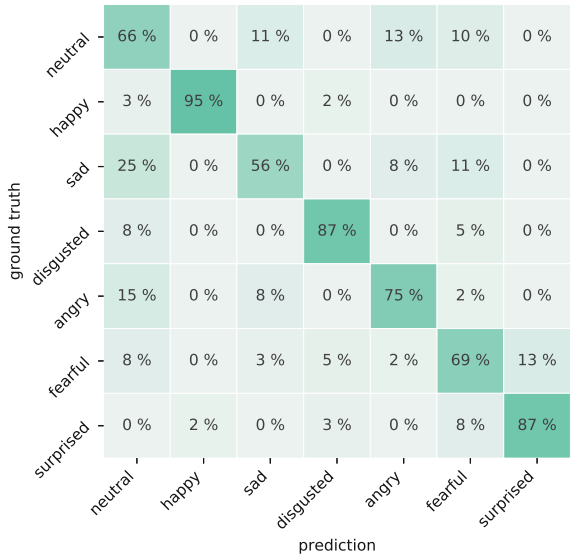


Figure 4.10: Normalized confusion matrix using a tree classifier.

Table 4.10: Classification report using a random forest classifier

Emotion	Precision	Recall	F_1-score
Angry	0.74	0.87	0.8
Disgusted	0.97	0.95	0.96
Fearful	0.73	0.84	0.78
Happy	1	1	1
Neutral	0.8	0.59	0.68
Sad	0.78	0.69	0.73
Surprised	0.88	0.93	0.9

The confusion matrix is shown in Figure 4.11. The matrix emphasizes that the tree classifier struggles with correctly classifying neutral facial expressions, whilst happiness is never predicted wrong. It can be observed that neutral facial expressions are most likely to be misclassified as anger (21 %), sadness (10 %), and fear (10 %). The expression of sadness is only classified correctly in 69 % of cases, whilst it is wrongly classified as fear in 15 %, as sadness in 8 %, and as fear in 8 % of the cases. All remaining emotions are correctly classified in at least 84 % of cases (fear: 84 %, anger: 87 %, surprise: 93 %, disgust: 95 %, happiness: 100 %).

AdaBoost classifier:
The cross-validated grid search showed the best results using a maximum depth of 9, a

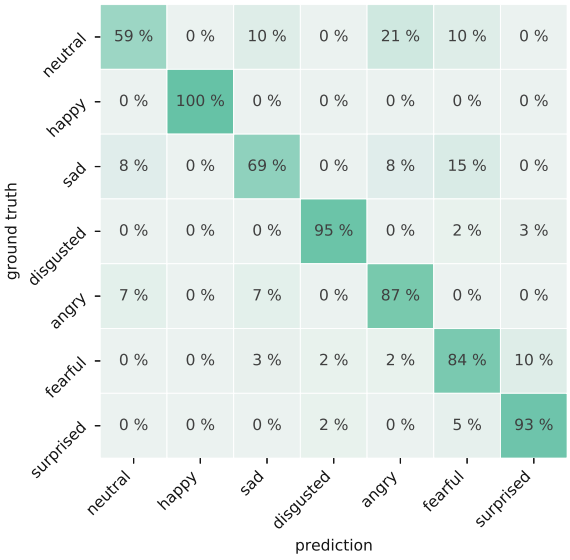


Figure 4.11: Normalized confusion matrix using a random forest classifier.

maximum number of estimators of 50, a learning rate of 1.1, Gini impurity for measuring the quality of a split, and choosing the best split as hyperparameters for training the AdaBoost classifier. The trained classifier achieves an accuracy of 87.04 % on the training set and an accuracy of 86.65 % on the test set. The main classification metrics are shown in Table 4.11 listing the precision, recall, and F_1-score for each class. It can be observed that the classifier achieves the best results for the class of happiness (F_1=1). The detector achieves the worst measurements for the emotion of sadness and neutral expression, where a F_1-score of 0.7 is measured.

Table 4.11: Classification report using an AdaBoost classifier

Emotion	**Precision**	**Recall**	**F_1-score**
Angry	0.93	0.89	0.91
Disgusted	0.95	0.98	0.97
Fearful	0.76	0.84	0.8
Happy	1	1	1
Neutral	0.69	0.72	0.7
Sad	0.74	0.66	0.7
Surprised	0.92	0.9	0.91

The confusion matrix computed on the test set is shown in Figure 4.12 and emphasizes the struggle of the AdaBoost classifier to correctly classify the expression of neutral,

which is only correctly classified in 74 % of all test images, and the emotion of sadness, which is only classified correctly in 77 % of cases. Neutral faces are most likely to be misclassified as anger in 13 %, as sadness in 8 %, and as fear in 5 % of cases. The expression of sadness is misclassified as neutral in 10 %, as fear in 10 %, and as anger in 3 % of the cases. In contrast, the emotion of happiness is correctly classified in 100 % of the tested images. The emotions of surprise and disgust are only misclassified in 5 % of the cases.

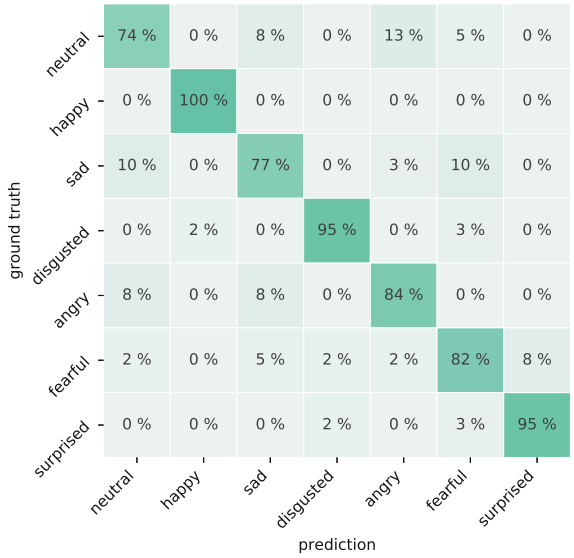


Figure 4.12: Normalized confusion matrix using an AdaBoost classifier.

K-nearest neighbors classifier:
The trained k-nearest neighbors classifier achieving the best results uses an Euclidean distance measurement and uses the nearest seven neighboring values for making decisions. The model achieves an accuracy of 82 % on the training set and of 85 % on the test set. The precision, recall, and F_1-score the k-nearest neighbors classifier achieves are listed for each emotion in Table 4.12. The measurements show that the classifier accomplishes F_1-scores above 0.9 for the emotions of anger, surprise, disgust, and happiness, suggesting good detection performance. However, in comparison, it struggles with the classification of sadness and neutral facial expressions ($F_1 = 0.7$). During the classification of neutral expressions, the detector shows a slightly better sensitivity (recall = 0.72) compared to the detector's precision (presicion = 0.69). In contrast the detector achieves a slightly worse sensitivity (recall = 0.66) compared to it's precision (presicion = 0.74).

Table 4.12: Classification report of K-nearest neighbors classifier

Emotion	Precision	Recall	F_1-score
Angry	0.93	0.89	0.91
Disgusted	0.95	0.98	0.97
Fearful	0.76	0.84	0.8
Happy	1	1	1
Neutral	0.69	0.72	0.7
Sad	0.74	0.66	0.7
Surprised	0.92	0.9	0.91

The confusion matrix shown in Figure 4.13 shows that the detector correctly classifies the emotion of sadness in 66 % of cases and is most likely to be misclassified as neutral (23 %) and as fear (11 %). Neutral is correctly classified in 72 % of cases and is most likely misclassified as sadness (15 %), anger (7 %), and fear (7 %).

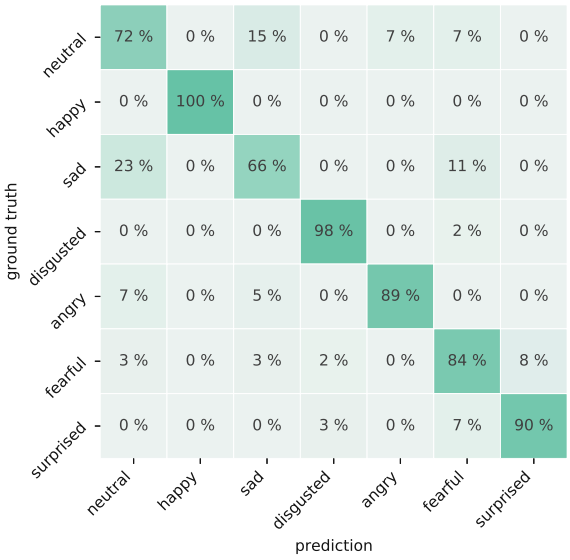


Figure 4.13: Normalized confusion matrix using a k-nearest neighbors classifier.

SVM:
The cross-validated grid search suggests using the one-vs-rest scheme, a regularization parameter of $C = 100$, a radial basis kernel function, and the inverse of the product of the number of features and the variance of the test set is used as kernel coefficient. The detector trained using these hyperparameters achieves a detection accuracy of 87.96 % on

the training set and a detection accuracy of 87.12 % on the test set. The main classification metrics are visualized in Table 4.13, showing the precision, recall and F_1-score for each class. It can be noticed, that the trained SVM achieves best overall detection performance of the emotions happiness ($F_1 = 1$), surprise ($F_1 = 0.93$), and disgust ($F_1 = 0.98$), whilst slightly struggling with the detection of neutral facial expressions ($F_1 = 0.74$).

Table 4.13: Classification report of SVM.

Emotion	**Precision**	**Recall**	**F_1-score**
Angry	0.84	0.92	0.88
Disgusted	0.98	0.97	0.98
Fearful	0.81	0.77	0.79
Happy	1	1	1
Neutral	0.76	0.72	0.74
Sad	0.8	0.77	0.78
Surprised	0.91	0.95	0.93

The confusion matrix shown in Figure 4.14 underlines the fact that the SVM struggles with correctly classifying the expression of neutral, whilst emotions, such as happiness, disgust and surprise are rarely predicted wrong. Furthermore, the confusion matrix shows that the emotion of neutral is misclassified as anger, sadness, and fear, with only 72 % of images of persons with neutral facial expressions being correctly classified as neutral, whilst 13 % are detected as anger, 11 % as sadness and 3 % as fear. The expression of fear and sadness are correctly classified in 77 % of cases. The detector achieves to correctly classify the emotion of anger in 92 %, surprise in 95 %, disgust in 97 %, and happiness in 100 % of test images.

Table 4.14 summarizes the detection performances of all tested classifiers. It can be seen that the SVM achieves the best detection performance with a detection accuracy of 87.13 % compared to the AdaBoost classifier (86.65 %), the k-nearest neighbors classifier (85.48 %), the random forest classifier (83.84 %), and the tree classifier (76.35 %).

Table 4.14: Summary of detection accuracy achieved by the tested classifiers.

Classifier	**Detection accuracy on test set in %**
Tree classifier	76.3466
Random forest classifier	83.8407
AdaBoost classifier	86.6511
K-nearest neighbors classifier	85.4801
SVM	87.1294

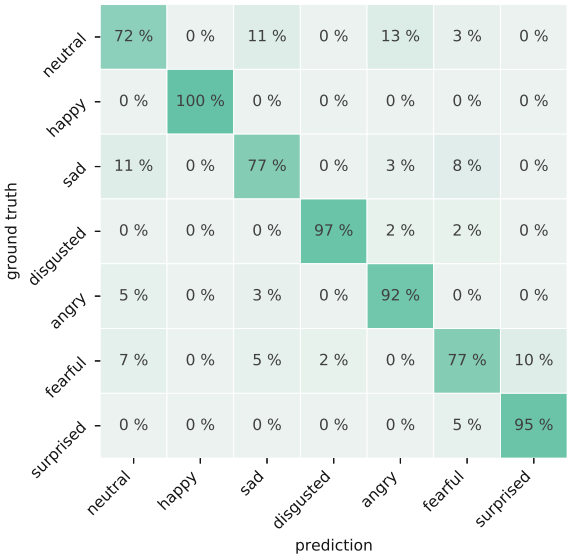


Figure 4.14: Normalized confusion matrix using a SVM.

4.2.1.3 Timing Behavior

Asides from analyzing the detection performance of each tested classifier, the timing behavior is evaluated. Therefore, the execution time of each detector is measured for each image of the emotion test set. The mean execution time and the standard deviation for each trained emotion detector are listed in Table 4.15. The measurements only include the emotion classification while excluding the time it takes to detect the face, the facial landmarks, and the distances used as input for the classifiers.

Table 4.15: Mean and standard deviation of execution times in seconds.

Classifier	mean in s	std in s	fps
Tree classifier	0.0027	0.0003	368.6209
Random forest classifier	0.0076	0.0004	131.4013
AdaBoost classifier	0.0073	0.0005	137.3858
K-nearest neighbors classifier	0.0036	0.0003	279.0895
SVM	0.0029	0.0002	349.9066

4.2.2 Emotion Detection from Facial Landmarks using Coordinate Patterns and Deep Learning

4.2.2.1 Emotion Detection Accuracy

The developed emotion detector is based on the facial landmarks derived by the facial landmark detection step to support Robo-Smile's pedagogical concept. As model #12 shows the best results fitted to the current task, the derived emotion detector bases on the facial landmarks computed using this predictor. To find the best-suited model, six different architectures (see Table 3.3) are tested using either all 68, 51, or 21 facial landmarks as input for the detector. The Radboud Facial Expression Database [71] is used for training, validating, and testing the emotion detector.

In Table 4.16 the overall accuracy in percentage is listed for each configuration possibility. It can be seen that Network 4, which uses the locations of 21 facial landmarks as input, leads to the best result of 88.57 %.

Table 4.16: Accuracy in percentage of the tested network architectures and facial landmark configurations.

# of land-marks	Network 1	Network 2	Network 3	Network 4	Network 5	Network 6
68	66.19 %	81.43 %	83.81 %	82.38 %	87.62 %	86.19 %
51	60.48 %	84.76 %	84.29 %	83.33 %	85.71 %	86.19 %
21	60.0 %	80.48 %	83.81 %	88.57 %	84.76 %	86.67 %

The performance of each network is further evaluated in regard to the classification performance of each class, as seen in the normalized confusion matrices in Figure 4.15 using all 68 facial landmarks as input, Figure 4.16 using 51 facial landmarks as input, and Figure 4.17 using 21 facial landmarks as input. The analysis shows that all detectors struggle to distinguish between neutral, sadness, and fear, which can be explained by the similarity of these two facial expressions. However, happiness, disgust, and surprise can be easily distinguished, as most trained networks manage to classify these three emotions correctly.

(a) Network 1

(b) Network 2

(c) Network 3

(d) Network 4

(e) Network 5

(f) Network 6

Figure 4.15: Normalized confusion matrices visualizing the performance of the trained emotion detectors using 68 facial landmarks as input.

(a) Network 1 (b) Network 2

(c) Network 3 (d) Network 4

(e) Network 5 (f) Network 6

Figure 4.16: Normalized confusion matrices visualizing the performance of the trained emotion detectors using 51 facial landmarks as input.

(a) Network 1

(b) Network 2

(c) Network 3

(d) Network 4

(e) Network 5

(f) Network 6

Figure 4.17: Normalized confusion matrices visualizing the performance of the trained emotion detectors using 21 facial landmarks as input.

4.2.2.2 Timing Behavior

Aside from analyzing the tested emotion detectors' accuracy, the derived models' timing behaviors are evaluated. Therefore, the self-trained emotion classification models' execution times are calculated for each test set's image sample. In Table 4.17, the mean execution time for each trained classifier is listed.

Table 4.17: Mean execution time in seconds of the trained network architectures and facial landmark configurations.

# of landmarks	Network 1	Network 2	Network 3	Network 4	Network 5	Network 6
68	0.0257	0.0261	0.0259	0.0259	0.0264	0.0259
51	0.0259	0.0259	0.0262	0.0262	0.0427	0.0402
21	0.0252	0.0253	0.0254	0.0253	0.0258	0.0395

4.3 Execution Time Analysis of the Overall Feedback System

The overall feedback system runs on the live webcam-video transmission. Aside from the frame transmission, the feedback system is comprised of a face, facial landmark, and emotion detection. The face and emotion detection are alternated with the facial landmark detection. As the facial landmarks are marked on each frame, the facial landmark detector is exchanged with a much faster optical flow tracking step when the face and emotion detector is applied.

Since both networks show best detection accuracy of all experiments conducted in this work, the best performing self-trained facial landmark detector (model #12) and the coordinate-pattern- and deep-learning-based emotion detector (network architecture 4, input of 21 facial landmark locations) are used to implement the overall system. The implemented feedback system's temporal behavior is evaluated on 5000 consecutive frames transmitted from the webcam running on a CPU with a frequency of 2.6 GHz. The mean, standard deviation, and the resulting fps are listed in Table 4.18. The distribution of measured execution times are shown in Figure 4.18. The program sections face detection, landmark detection, and emotion detection contain the needed image processing, while the program sections emotion detector and face detector describe only the respective detector. It can be seen that the execution times of the face detector, facial landmark detector, and emotion detector are similar. The optical flow step, however, is one order of magnitude faster.

Table 4.18: The mean execution time, it's standard deviation, and the average fps for each program section and the overall system is listed.

Program section	**mean in s**	**std in s**	**fps**
Face detection	0.0374	0.0065	26.7189
Landmark detection	0.0356	0.0080	28.1237
Landmark detector	0.0341	0.0080	29.3275
Optical flow step	0.0027	0.0002	364.8069
Emotion detection	0.0284	0.0079	35.1822
Emotion detector	0.0280	0.0079	35.7389
Overall	0.0536	0.0186	18.6396

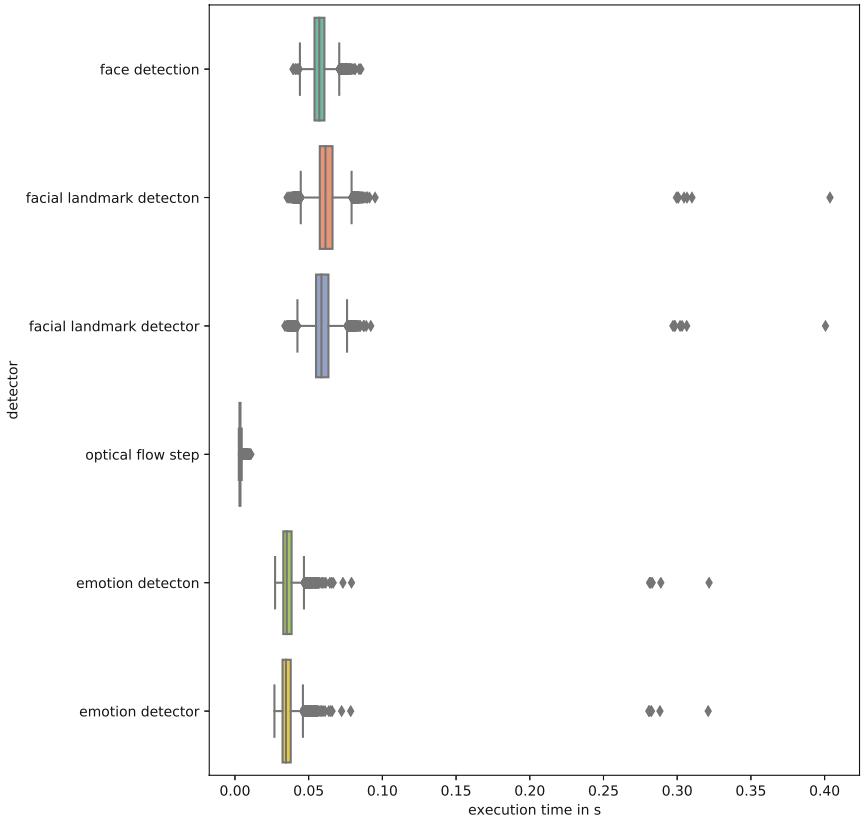


Figure 4.18: Boxplot graph of the execution times measured over all samples of the test set running on a CPU with a frequency of 2.6 GHz.

CHAPTER 5

Discussion and Conclusion

5.1 Discussion

A feedback system suited for the usage as part of the serious game Robo-Smile is implemented in this work. As the results have shown, the developed architecture can accurately and robustly detect facial landmarks even during large pose variations and classify expressed emotion based on the previously detected facial landmarks while achieving execution times allowing live webcam-based feedback. Hence, the requirements of the intended application have been fulfilled.

5.1.1 Facial Landmark Detection

Robo-Smile's pedagogical concept includes the need to detect the main facial features in the webcam stream. Therefore this thesis' focus includes the detection of facial landmarks. The developed facial landmark detector with the best results regarding detection accuracy is model #12. The model shows state-of-the-art detection accuracy while having sufficiently fast execution times to still allow frame-wise processing of a video stream without noticeable delay or pauses. The model with the best performance is loosely based on the MCL network and consists of three stacks of two Convolutional layers each [47], followed by a max-pooling layer, one stack of three Convolutional Layers, one Global Average Pooling Layer, and one Fully-Connected Layer. Batch Normalization and ReLU are used after each Convolutional Layer to improve the network's convergence. As input a 50×50 px grayscale image is used, which is furthermore frame-wise mean-centered and normalized using the standard deviation. For training, batch sizes of 64 samples and the 300W-LP database [64], which extends the 300-W dataset [65] [25] [66] to contain samples of large pose variations by utilizing Zhu et al.'s image augmentation method [64], are used. Furthermore, Adam optimization is used. The learning rate is decreased automatically if the validation loss starts stagnating. Furthermore, the training is stopped early if decreasing the learning rate does not improve the learning process.

The main differences between the developed detector (#12) and the MCL network, which the network is based upon, are discussed in more detail in the following points.

- **Image Databases and data augmentation:** Both networks use the 300 W database [65] [25] [66] for training the detector to detect 68 facial landmarks. However, Shao et al. MCL increase the diversity of training samples by rotation, uniform scaling, translation, horizontal flip, and JPEG compression [21], while this thesis exploits Zhu et al.'s image augmentation to artificially generate samples of different head poses.

- **Preprocessing:** The input images of the MCL network are normalized to $[-1, 1]$ by subtracting 128 and dividing by 128 [21], while the detector #12 uses frame-wise mean centering and normalization using the standard deviation.

- **Network design and training:** This thesis' network architecture is loosely based on Shoa et al.'s MCL [21] model. The MCL architecture can be sectioned into shared layers, multiple shape prediction layers trained to predict a fraction of all 68 landmarks, and one final shape prediction layer. The proposed model utilizes the shared layers and extends the network with a fully Connected Layer to extract features. Aside from the number of shape prediction layers, the training procedure differs as well. Shao et al. focus on the optimization of difficult landmarks first by weighting the distance-based loss (see Equation 2.3) of each landmark individually [21]. This thesis uses Wing loss, which emphasizes small and medium-range errors [19]. Furthermore, different optimizers are used as the MCL approach uses SGD, whilst this work uses Adam optimization. Both propose the usage of batch sizes of 64. Finally, Shao et al. reduce the learning rate after a specified number of iterations, while model #12 was trained by reducing the learning rate with respect to the validation loss.

The model adaptation decreased its complexity and further improved the detection accuracy as the MCL architecture results in a mean normalized error calculated over the IBUG [25] test set of **8.51**, while this thesis presents a mean normalized error of **7.01** tested on the same test set. With respect to execution time, MCL reports an average speed of **57 fps** tested on an i5-6200U CPU with a clock frequency of 2.3 GHz. The proposed network of this thesis demonstrates an average execution time of **33.41 fps** tested on an i7-6500U CPU with a clock frequency of 2.5 GHz. However, as the network architecture proposed in this thesis is a scaled-down version of the network proposed by Shao et al. the discrepancies in speed can be traced back to the implementation of the network and not the network itself. MCL is implemented using the deep learning framework Caffe [84] [21], while this thesis is implemented using tensorflow. Dong el al's CPM with SBR [70] shows superior mean normalized error performance on the full 300W test set [65] [25] [66], however, they do not state the model's execution time. Since the model uses a CPM, which has an execution time of 33 ms tested on a GPU (GTX

1080 Ti) [74], it can be assumed that the model will not be applicable for generating live-feedback on devices without a powerful GPU.

The proposed architecture capable of detecting 68 facial landmarks not only shows superior behavior compared to the MCL-network, but also in comparison to state-of-the-art networks listed in Table 4.5 and Table 4.6. Dlib's mean and median error normalized using the IOD is smaller for the common subset. However, for images of the challenging subset of the public 300 W dataset [65] [25] [66], which are considered more difficult, model #12 shows better performance than Dlib's shape predictor. A closer look at the error distribution visualized in Figure 4.1 and Figure 4.2 shows that more outliers exist in the results of Dlib's shape predictor, indicating superiority in the precision of the developed model #12. Applying an image-based detector with low precision on each frame of a video can lead to jitter and detection instability [70]. Thus high precision is advantageous for the control system developed in this thesis.

The current image augmentation used during the facial landmark detector training is limited to Zhu et al.'s [44] methodology of generating labeled profile views. However, popular augmentation methods such as rotation and mirroring, which are, for example, used during training of the emotion detector, are not utilized, which can be taken into account during further development.

Summarizing, this thesis presents a state-of-the-art architecture capable of precisely detecting 68 facial landmarks with execution times allowing the detection of more than 30 fps. More precise detector results are preferable, especially during frame-wise detection, as fluctuation in the detection disrupts the perception of fluid tracking. As interruptions in the tracking results are distracting, it is especially bad for the given use-case. Furthermore, as the children are asked to imitate facial expressions and are likely to move and turn, the detection during challenging poses is essential. Thus, the usage of detector #12 is preferred for the developed feedback mechanism, as it proves to perform well during large pose variations.

5.1.2 Emotion Detection

State-of-the-art emotion detection algorithms focus on the detection of facial expressions in image sequences instead of still images [57]. However, as this thesis aims to develop an image and facial-landmark-based classifier, the comparison to state of the art is not applicable. This thesis describes two main approaches to find the best detector for the application as part of Robo-Smile's emotion feedback system.

- **AU-based emotion detection:** Facial distances are used to describe the main facial changes associated with facial expressions. ANOVA tests are conducted to find the best features suited for classifying emotions. Based on the analysis, 16 features are selected and used as input for training a tree classifier, a random forest classifier, an AdaBoost classifier, a k-nearest neighbors classifier, and a SVM. The detector, capable of achieving best classification results, uses a SVM for

distinguishing the six basic emotions and the expression of neutral. The designed network achieves a detection accuracy of 87.13 %.

- **Coordinate-pattern- and deep-learning-based emotion detection:** The second emotion classification approach described in this thesis is inspired by DTAGN [57] and DGNN [49] and tries to map the patterns formed by facial landmarks directly to the expressed emotion. The usage of 68, 51, and 21 facial landmarks as input and six different network architectures are evaluated. The detector, which results in the best detection performance, uses 21 facial landmarks as input and the network architecture #4. This configuration achieves a detection accuracy of 88.57 %.

Both methodologies can distinguish emotion in facial expressions and thus can be used as part of Robo-Smile's emotion feedback system. However, since the coordinate-pattern- and deep-learning-based emotion detection has a slightly better detection accuracy (improvement of 1.65 % with respect to the AU-based emotion detection), it is used as part of the final application.

All trained detectors use the facial landmarks of a single image as input. However, further developments should take landmark-based state-of-the-art approaches into account, which use image sequences [57] [49]. As the goal of the game is not the detection of spontaneous emotion, starting and ending in a neutral state, but rather to mimic a given emotion for several seconds, the applicability of using image sequences for the emotion detection must be further evaluated.

In the course of the facial landmark detectors' training, the image database is augmented using Zhu et al.'s [44] methodology of generating labeled profile views to contain a large variety of pose variations. However, during the training of the emotion detectors, the Radboud [71] database is not augmented using this methodology, as side profiles cover up part of the facial landmarks. However, additional experiments on the impact of additional augmentation to include various poses should be considered for further development of the feedback system.

5.1.3 Application Considerations

The detection of the face, facial landmarks, and the expressed emotion are time-consuming. However, to run the feedback system on the live webcam transmission, the overall execution time must allow the image processing of at least 15 fps. Otherwise, the perception of fluent movement can not be guaranteed. Both the emotion and facial landmark detector manage execution times exceeding the 15 fps limitation. However, as seen in Table 4.18 the combined sequential average execution time would exceed this limit. This can be seen in Equation 5.1. The resulting overall execution time t is the sum of the execution time of the face detector t_{face}, facial landmark detector $t_{landmark}$, and emotion detection $t_{emotion}$. All program sections' execution times take the needed image processing to adjust the input image to fit the corresponding detector into account. The

resulting time of 0.1014 s, which corresponds to 9.8619 fps, violates the limitations of 15 fps. Thus, it emphasizes the need for alternating detectors and alternating the facial landmark detector with the less time-consuming optical flow step.

$$t = t_{face} + t_{landmark} + t_{emotion} = 0.0374\text{s} + 0.0356\text{s} + 0.0284\text{s} = 0.1014\text{s} \mathrel{\widehat{=}} 9.8619\text{fps} \quad (5.1)$$

The considerations made to speed up the system's execution time proves successful, as a mean of 18.64 fps could be attained, calculated over 5000 consecutive frames transmitted by the computer's internal webcam running on a CPU with a frequency of 2.6 GHz. It must be noted that the execution times listed in Table 4.18 are larger compared to previous analysis running on the corresponding test set (see Table 4.17 and Table 4.18). This discrepancy can be explained by the fact that the transmission of frame-wise video data of the computer's internal webcam is running in the background during testing of the overall system.

5.2 Conclusion

This thesis presents a novel deep-learning-based facial expression feedback system capable of giving live feedback on expressed emotions. The implementation allows a purely computer-aided live emotion feedback system, which removes struggles caused by pragmatics and anxiety due to human interaction [14], and allows continuous feedback linked to increased activity in the extended face perception network [15].

- **Facial Landmark detection:** An original algorithm is developed by combining and adapting aspects of existing facial landmark detectors. This detector not only manages to achieve state-of-the-art results, improving the methodologies it is inspired by, especially during large pose variations, but also attains execution times allowing live webcam-based feedback running on CPUs only. The accurate facial landmark detection facilitates the marking of those key-points. It connects them to form abstractions of eyes, nose, and mouth overlayed onto the captured video stream, hypothesized to draw the child's attention towards those regions. Rotated heads, partly covered faces, and bad lighting conditions are considered challenging for facial landmark detectors. The challenging subset of the private 300W image database is comprised of such challenging conditions. For this subset the facial landmark detector presented in this work outperforms state-of-the-art methodologies with a mean normalized error of 7.01 (improvement of 7.85 % compared to TSR, and 8.13 % compared to CPM with SBR). Hence, the requirement of delivering accurate results during large pose variations is met, which is essential when the participant is restless, turns, or moves. Therefore, the presented algorithm is considered to be suited for the intended use by children.
- **Emotion detection:** In order to develop a unique pattern and facial-landmark-based emotion detector, AUs- and purely coordinate-based methodologies are designed, trained, and tested in order to find the best emotion classifier for the

given facial-landmark-based input. This evaluation led to the development of a new pattern and facial-landmark-based facial expression classifier capable of distinguishing between the expressions of happiness, sadness, disgust, fear, anger, surprise, and neutral with an accuracy of 87.13 % (AU-based methodology) and 88.57 % (purely coordinate-based methodology). All presented classifiers use only the previously detected facial landmarks or distances derived by these landmarks as input. Thus the developed classifiers are compliant with Robo-Smile's pedagogical concept.

- **Overall feedback mechanism:** Struggles in recognizing emotion in faces of children with ASD are connected to atypical eye-tracking patterns [4], as socially relevant features such as eyes, nose, and mouth are significantly less viewed compared to non-significant features [13]. Thus, emphasizing socially relevant facial features and drawing attention to them is hypothesized to help detect emotions in facial expressions. By basing the emotion classification purely on information and patterns derived by the facial landmarks, the feedback algorithm supports the preference of keeping up routines and firm procedures [2] by linking the task of detecting and imitating emotions in facial expressions to specific and constant patterns. These facial patterns are furthermore emphasized by the marked facial landmarks in the feedback the child receives. The algorithms used for detecting facial landmarks and classifying the emotion must, in sum, have execution times, which allow frame-wise video processing while guaranteeing the perception of movement continuity, requiring a minimum frame-rate of 15 fps. By alternating the detectors and alternating the facial landmark detector with the less time-consuming optical flow step, an average of 18.64 fps could be attained, calculated over 5000 consecutive frames transmitted by the computer's internal webcam running on a CPU with a frequency of 2.6 GHz, thus proving that the presented system does not require additional expensive hardware in the form of GPUs.

Summarizing, this thesis presents a feedback system architecture, consisting of a novel facial landmark detection algorithm achieving results equivalent to comparable state-of-the-art detectors, twice as good as the previous prototype, and a purely facial-landmark-based emotion detector suited for usage as part of Robo-Smile, thus allowing the answering of the research questions.

- Which combined facial landmark detector and emotion classifier can operate on a webcam stream without requiring additional computational resources (e.g. GPU) while still achieving a frame rate of minimum 15 fps?
- To achieve the best feedback accuracy, how must the system's architecture be designed to be suited for the learning platform?

The system architecture leading to the best feedback accuracy is the combination of the self-designed and trained facial landmark detector #12 and the deep-learning-based emotion detector using coordinate patterns of facial landmarks as input. Thus,

these methods are best suited for the usage in Robo-Smile. By additionally exploiting the strategies of alternating facial landmark detectors and substituting computational expensive steps such as the facial landmark detection with faster, less accurate steps such as optical flow, execution times above 15 fps can be achieved.

5.3 Further Challenges

This thesis focuses on the technical realization of a robust feedback system capable of giving continuous feedback on the expressed emotion based on the facial landmarks alone. By basing the emotion detection on facial landmarks, the learning game aims to teach emotions in a way that is tailored to the neurological needs of children with ASD. The main hypotheses behind the learning game are that children with ASD tend to learn to express and recognize emotions easier when linked to specific patterns in the face and the assumption that gamification and pattern-based feedback helps to improve learning. These hypotheses are supported by current literature. However, to further support the assumptions made during the design of the pedagogical concept, studies need to be conducted to test the game's effectiveness and the learning strategies behind the game. Furthermore, the usability and the possibility to integrate the game in the education and daily life of children with ASD need to be further evaluated. By presenting a control mechanism overcoming previous limitations, this thesis lays the foundation needed to conduct further studies testing the usability and effectiveness of the serious game Robo-Smile.

List of Figures

List of Tables

Acronyms

3DDFA 3D Dense Face Alignment. 13, 51

3DMM 3D Morphable Model. 13

AAM Active Appearance Model. 9, 10, 16

abstanh Absolute Tangens Hyperbolicus. 22, 23, 26, 41

ANOVA Analysis of Variance. 30, 41, 56, 61, 79, 85, 87

ASD Autism Spectrum Disorder. ix, xiii, 1–3, 7, 8, 15, 34, 82, 83

ASS Autismus-Spektrum-Störung. vii, xi

AU Action Unit. 4, 7, 8, 16–20, 29–33, 41, 56, 80–82, 87

BGR Blue Green Red. 22–24, 26

CDM Cascaded Deformable Shape Mode. 50

CE-CLM Convolutional Experts Constrained Local Model. 50, 51

CFAN Coarse-to-Fine Auto-Encoder. 11, 50

CFSS Coarse-to-Fine Shape Searching. 50, 51

CLM Constrained Local Model. 10

CLNF Constrained Local Neural Field. 11, 12, 51

CNN Convolutional Neural Network. 7, 11–14, 18, 27, 44

CPM Convolutional Pose Machine. 49, 50, 78, 81

CPU Central Processing Unit. 7, 8, 11–13, 25, 54, 55, 74, 75, 78, 81, 85–87

D3PF Direct 3D Projected Feature. 12

DAN Deep Alignment Network. 12

DGNN Directed Graph Neural Networks. 18, 36, 41, 80

DRMF Discriminative Response Map Fitting. 50, 51

DTAGN Deep Temporal Appearance-Geometry Network. 18, 36, 41, 80

DTAN Deep Temporal Appearance Network. 18

DTGN Deep Temporal Geometry Network. 18

ESR Explicit Shape Regression. 50

FACS Facial Action Coding System. 4, 16, 29, 41

FEC-CNN Fully End-to-End Cascaded Convolutional Neural Network. 13

FLDet Face and Landmark Detector. 11, 13

fps frames per second. 6, 7, 11–13, 39–41, 54, 55, 69, 74, 75, 78–83, 87, 88

GPU Graphics Processing Unit. 7, 11–13, 18, 22, 49, 50, 78, 79, 82

HCI Human Computer Interactions. 15

IOD Inter-Ocular Distance. 14, 15, 33, 44, 50, 51, 79, 87

JEMImE educative multimodal game for emotional imitation. 2

L1 Lasso regression. 14, 27

L2 Ridge regression. 14, 22, 23, 27

LAB Look at Boundary. 13

LBF Local Binary Features. 50

LFPN Lightweight Feature Pyramid Network. 13

MA 10 Magistratsabteilung 10. vii, ix, 2

MCL Multi-Center Learning. 7, 11, 13, 15, 18, 23–27, 40, 77–79

MDM Mnemonic Descent Method. 12, 50

MTCNN Multitask Cascaded Convolutional Network. 12

ODN Occlusion-adaptive Deep Network. 13

PAM Pose-Aware Model. 11

PAWF Piecewise Affine-Warped Feature. 12

PDB Pose-based Data Balancing. 14

RAR Recurrent Attentive-Refinement Network. 12

RCPR RobustCascaded Pose Regression. 50

RDB Rapidly Digested Backbone. 13

ReLU Rectifier. 14, 23, 38, 41, 77

ResNet Residual Network. 25

SBR Supervision-by-Registration. 49, 50, 78, 81

SDM Supervised Descent Method. 11, 50

SGD Stochastic Gradient Descent. 14, 15, 27, 78

std standard deviation. 46, 47, 55, 69, 75

SVM Support Vector Machine. 4, 17, 35, 41, 67–69, 79, 86, 88

TCDCN Task-Constrained Deep Convolutional Network. 7, 11, 13, 14, 17, 18, 25, 27, 40, 50, 51

TCNN Tweaked Convolutional Neural Network. 7, 12–14, 18, 22–26, 40

TSR Two-Stage Reinitialization. 12, 50, 81

Bibliography

[1] M. Maenner, K. Shaw, J. Baio, A. Washington, M. Patrick, M. DiRienzo, D. Christensen, L. Wiggins, S. Pettygrove, J. Andrews, M. Lopez, A. Hudson, T. Baroud, Y. Schwenk, T. White, C. Rosenberg, L.-C. Lee, R. Harrington, M. Huston, and P. Dietz, "Prevalence of autism spectrum disorder among children aged 8 years — autism and developmental disabilities monitoring network, 11 sites, united states, 2016," *Morbidity and Mortality Weekly Report. Surveillance Summaries*, vol. 69, pp. 1–12, 03 2020.

[2] W. H. Organization, *The ICD-10 classification of mental and behavioural disorders: clinical descriptions and diagnostic guidelines*. Geneva,Switzerland: World Health Organization, 1992.

[3] B. Chamberlain, C. Kasari, and E. Rotheram-Fuller, "Involvement or isolation? the social networks of children with autism in regular classrooms," *Journal of autism and developmental disorders*, vol. 37, pp. 230–42, 03 2007.

[4] J. Nomi and L. Uddin, "Face processing in autism spectrum disorders: From brain regions to brain networks," *Neuropsychologia*, vol. 71, 03 2015.

[5] Y. Suchy and J. A. Holdnack, "Chapter 8 - assessing social cognition using the acs for wais–iv and wms–iv," in *WAIS-IV, WMS-IV, and ACS* (J. A. Holdnack, L. W. Drozdick, L. G. Weiss, and G. L. Iverson, eds.), pp. 367 – 406, San Diego: Academic Press, 2013.

[6] C. Chevallier, G. Kohls, V. Troiani, E. Brodkin, and R. Schultz, "The social motivation theory of autism," *Trends in cognitive sciences*, vol. 16, pp. 231–9, 03 2012.

[7] C. Grossard, O. Grynspan, S. Serret, A.-L. Jouen, K. Bailly, and D. Cohen, "Serious games to teach social interactions and emotions to individuals with autism spectrum disorders (asd)," *Computers & Education*, vol. 113, 05 2017.

[8] D. Djaouti, J. Alvarez, and J.-P. Jessel, "Classifying serious games: the g/p/s model," *Handbook of Research on Improving Learning and Motivation through Educational Games: Multidisciplinary Approaches*, 01 2011.

[9] T. Fernandes, S. Alves, J. Miranda, C. Queirós, and V. Orvalho, "Lifeisgame:a facial character animation system to help recognize facial expressions," vol. 221, 10 2011.

[10] N. Harrold, C. T. Tan, D. Rosser, and T. Leong, "Copyme: an emotional development game for children," 04 2014.

[11] C. Grossard, S. Hun, A. Dapogny, E. Juillet, F. Hamel, H. Jean-Marie, J. Bourgeois, H. Pellerin, P. Foulon, S. Serret, O. Grynszpan, K. Bailly, and D. Cohen, "Teaching facial expression production in autism: The serious game jemime," *Creative Education*, vol. 10, pp. 2347–2366, 01 2019.

[12] N. M. Weinert, *Development of a training platform for the promotion of social behavior in children with autism*. Bachelor's thesis, University of Applied Sciences FH Technikum Wien, 5 2017.

[13] K. Pelphrey, N. Sasson, J. Reznick, G. Paul, B. Goldman, and J. Piven, "Visual scanning of faces in autism," *Journal of autism and developmental disorders*, vol. 32, pp. 249–61, 09 2002.

[14] J. Daniels, N. Haber, C. Voss, J. Ouillon, S. Tamura, A. Fazel, A. Kline, P. Washington, J. Phillips, T. Winograd, C. Feinstein, and D. Wall, "Feasibility testing of a wearable behavioral aid for social learning in children with autism," *Applied Clinical Informatics*, vol. 09, pp. 129–140, 01 2018.

[15] S. Bölte, D. Hubl, S. Feineis-Matthews, D. Prvulovic, T. Dierks, and F. Poustka, "Facial affect recognition training in autism: Can we animate the fusiform gyrus?," *Behavioral neuroscience*, vol. 120, pp. 211–6, 03 2006.

[16] D. E. King, "Dlib-ml: A machine learning toolkit," *Journal of Machine Learning Research*, vol. 10, pp. 1755–1758, 2009.

[17] V. Kazemi and J. Sullivan, "One millisecond face alignment with an ensemble of regression trees," in *Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition*, CVPR '14, p. 1867–1874, 2014.

[18] P. Read and M.-P. Meyer, eds., *Restoration of Motion Picture Film*. Waltham, Massachusetts: Butterworth-Heinemann, 2000.

[19] Z. Feng, J. Kittler, M. Awais, P. Huber, and X. Wu, "Wing loss for robust facial landmark localisation with convolutional neural networks," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2235–2245, 2018.

[20] Y. Wu, T. Hassner, K. Kim, G. Medioni, and P. Natarajan, "Facial landmark detection with tweaked convolutional neural networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 12, pp. 3067–3074, 2018.

[21] Z. Shao, H. Zhu, X. Tan, Y. Hao, and L. Ma, "Deep multi-center learning for face alignment," *Neurocomputing*, vol. 396, pp. 477 – 486, 2020.

[22] Z. Zhang, P. Luo, C. C. Loy, and X. Tang, "Facial landmark detection by deep multi-task learning," 09 2014.

[23] Z. Zhang, P. Luo, C. C. Loy, and X. Tang, "Learning deep representation for face alignment with auxiliary attributes," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 38, no. 5, pp. 918–930, 2016.

[24] N. Wang, X. Gao, D. Tao, H. Yang, and X. Li, "Facial feature point detection: A comprehensive survey," *Neurocomputing*, vol. 275, pp. 50–65, 2018.

[25] C. Sagonas, G. Tzimiropoulos, S. Zafeiriou, and M. Pantic, "300 faces in-the-wild challenge: The first facial landmark localization challenge," *2013 IEEE International Conference on Computer Vision Workshops*, pp. 397–403, 2013.

[26] B. Johnston and P. Chazal, "A review of image-based automatic facial landmark identification techniques," *EURASIP Journal on Image and Video Processing*, vol. 2018, p. 86, 09 2018.

[27] Y. Wu and Q. Ji, "Facial landmark detection: A literature survey," *International Journal of Computer Vision*, vol. 127, p. 115–142, May 2018.

[28] T. Cootes, G. Edwards, and C. Taylor, "Active appearance models," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23, pp. 681 – 685, 07 2001.

[29] Y. wu and Q. Ji, "Facial landmark detection: A literature survey," *International Journal of Computer Vision*, 05 2018.

[30] J. Zhang, S. Shan, M. Kan, and X. Chen, "Coarse-to-fine auto-encoder networks (cfan) for real-time face alignment," in *European Conference on Computer Vision* (D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars, eds.), (Cham), pp. 1–16, Springer International Publishing, 2014.

[31] C. Zhuang, S. Zhang, X. Zhu, Z. Lei, J. Wang, and S. Z. Li, "Fldet: A cpu real-time joint face and landmark detector," in *2019 International Conference on Biometrics (ICB)*, pp. 1–8, 2019.

[32] T. Baltrusaitis, P. Robinson, and L. Morency, "Constrained local neural fields for robust facial landmark detection in the wild," in *2013 IEEE International Conference on Computer Vision Workshops*, pp. 354–361, 2013.

[33] X. Xiong and F. De la Torre, "Supervised descent method and its applications to face alignment," in *2013 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 532–539, 2013.

[34] Y. Sun, X. Wang, and X. Tang, "Deep convolutional network cascade for facial point detection," in *2013 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3476–3483, 2013.

[35] E. Zhou, H. Fan, Z. Cao, Y. Jiang, and Q. Yin, "Extensive facial landmark localization with coarse-to-fine convolutional network cascade," in *2013 IEEE International Conference on Computer Vision Workshops*, pp. 386–391, 2013.

[36] R. Ranjan, V. M. Patel, and R. Chellappa, "Hyperface: A deep multi-task learning framework for face detection, landmark localization, pose estimation, and gender recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 41, no. 1, pp. 121–135, 2019.

[37] S. Xiao, J. Feng, J. Xing, H. Lai, S. Yan, and A. Kassim, "Robust facial landmark detection via recurrent attentive-refinement networks," vol. 9905, pp. 57–72, 10 2016.

[38] A. Jourabloo and X. Liu, "Large-pose face alignment via cnn-based dense 3d model fitting," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4188–4196, 2016.

[39] K. Zhang, Z. Zhang, Z. Li, and Y. Qiao, "Joint face detection and alignment using multitask cascaded convolutional networks," *IEEE Signal Processing Letters*, vol. 23, no. 10, pp. 1499–1503, 2016.

[40] G. Trigeorgis, P. Snape, M. A. Nicolaou, E. Antonakos, and S. Zafeiriou, "Mnemonic descent method: A recurrent process applied for end-to-end face alignment," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[41] J. Lv, X. Shao, J. Xing, C. Cheng, and X. Zhou, "A deep regression architecture with two-stage re-initialization for high performance facial landmark detection," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3691–3700, 07 2017.

[42] J. Yang, Q. Liu, and K. Zhang, "Stacked hourglass network for robust facial landmark localisation," in *2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pp. 2025–2033, 07 2017.

[43] Z. He, M. Kan, J. Zhang, X. Chen, and S. Shan, "A fully end-to-end cascaded cnn for facial landmark detection," in *2017 12th IEEE International Conference on Automatic Face Gesture Recognition (FG 2017)*, pp. 200–207, 05 2017.

[44] X. Zhu, X. Liu, Z. Lei, and S. Z. Li, "Face alignment in full pose range: A 3d total solution," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 41, no. 1, pp. 78–92, 2019.

[45] M. Zhu, D. Shi, M. Zheng, and M. Sadiq, "Robust facial landmark detection via occlusion-adaptive deep networks," in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3481–3491, 2019.

[46] S. Bianco, R. Cadene, L. Celona, and P. Napoletano, "Benchmark analysis of representative deep neural network architectures," *IEEE Access*, vol. 6, p. 64270–64277, 2018.

[47] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv 1409.1556*, 09 2014.

[48] M. Pantic and L. J. M. Rothkrantz, "Automatic analysis of facial expressions: the state of the art," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 12, pp. 1424–1445, 2000.

[49] Q. T. Ngoc, S. Lee, and B. Song, "Facial landmark-based emotion recognition via directed graph neural network," *Electronics*, vol. 9, p. 764, 2020.

[50] C. Marechal, D. Mikołajewski, K. Tyburek, P. Prokopowicz, L. Bougueroua, C. Ancourt, and K. Węgrzyn-Wolska, *Survey on AI-Based Multimodal Methods for Emotion Detection*, pp. 307–324. Cham: Springer International Publishing, 2019.

[51] P. Ekman and W. V. Friesen, "Constants across cultures in the face and emotion," *Journal of Personality and Social Psychology*, vol. 17, no. 2, pp. 124–129, 1971.

[52] S. Li and W. Deng, "Deep facial expression recognition: A survey," *IEEE Transactions on Affective Computing*, p. 1–1, 2020.

[53] P. Ekman and E. Rosenberg, *What the Face Reveals: Basic and Applied Studies of Spontaneous Expression Using the Facial Action Coding System (FACS)*. Series in Affective Science, Oxford University Press, 2005.

[54] D. Matsumoto and P. Ekman, "Facial expression analysis," *Scholarpedia*, vol. 3, no. 5, p. 4237, 2008.

[55] E. Sariyanidi, H. Gunes, and A. Cavallaro, "Automatic analysis of facial affect: A survey of registration, representation, and recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 6, pp. 1113–1133, 2015.

[56] S. Li and W. Deng, "Deep facial expression recognition: A survey," *IEEE Transactions on Affective Computing*, p. 1–1, 2020.

[57] H. Jung, S. Lee, J. Yim, S. Park, and J. Kim, "Joint fine-tuning in deep neural networks for facial expression recognition," in *2015 IEEE International Conference on Computer Vision (ICCV)*, pp. 2983–2991, 2015.

[58] G. Van Rossum and F. L. Drake, *Python 3 Reference Manual*. Scotts Valley, CA: CreateSpace, 2009.

[59] G. Bradski, "The OpenCV Library," *Dr. Dobb's Journal of Software Tools*, 2000.

[60] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[61] M. Köstinger, P. Wohlhart, P. M. Roth, and H. Bischof, "Annotated facial landmarks in the wild: A large-scale, real-world database for facial landmark localization," in *2011 IEEE International Conference on Computer Vision Workshops (ICCV Workshops)*, pp. 2144–2151, 2011.

[62] P. N. Belhumeur, D. W. Jacobs, D. J. Kriegman, and N. Kumar, "Localizing parts of faces using a consensus of exemplars," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 12, pp. 2930–2940, 2013.

[63] V. Le, J. Brandt, Z. Lin, L. Bourdev, and T. S. Huang, "Interactive facial feature localization," in *Proceedings of the 12th European Conference on Computer Vision - Volume Part III*, ECCV'12, p. 679–692, 2012.

[64] X. Zhu, Z. Lei, X. Liu, H. Shi, and S. Li, "Face alignment across large poses: A 3d solution," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 146–155, 06 2016.

[65] C. Sagonas, E. Antonakos, G. Tzimiropoulos, S. Zafeiriou, and M. Pantic, "300 faces in-the-wild challenge: database and results," *Image and Vision Computing*, vol. 47, pp. 3 – 18, 2016.

[66] C. Sagonas, G. Tzimiropoulos, S. Zafeiriou, and M. Pantic, "A semi-automatic methodology for facial landmark annotation," in *2013 IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pp. 896–903, 2013.

[67] X. Zhu and D. Ramanan, "Face detection, pose estimation, and landmark localization in the wild," in *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2879–2886, 2012.

[68] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proceedings of the 32nd International Conference on Machine Learning* (F. Bach and D. Blei, eds.), vol. 37 of *Proceedings of Machine Learning Research*, pp. 448–456, 2015.

[69] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, 2016.

[70] X. Dong, S.-I. Yu, X. Weng, S.-E. Wei, Y. Yang, and Y. Sheikh, "Supervision-by-registration: An unsupervised approach to improve the precision of facial landmark detectors," pp. 360–368, 06 2018.

[71] O. Langner, R. Dotsch, G. Bijlstra, D. Wigboldus, S. Hawk, and A. Knippenberg, "Presentation and validation of the radboud face database," *Cognition and Emotion - COGNITION EMOTION*, vol. 24, pp. 1377–1388, 12 2010.

[72] J. Zhu, S. Rosset, H. Zou, and T. Hastie, "Multi-class adaboost," *Statistics and its interface*, vol. 2, pp. 349–360, 02 2006.

[73] J. yves Bouguet, "Pyramidal implementation of the lucas kanade feature tracker," *Intel Corporation, Microprocessor Research Labs*, 2000.

[74] D. Osokin, "Global context for convolutional pose machines," *ArXiv*, vol. abs/1906.04104, 2019.

[75] X. Yu, J. Huang, S. Zhang, W. Yan, and D. N. Metaxas, "Pose-free facial landmark fitting via optimized part mixtures and cascaded deformable shape model," in *2013 IEEE International Conference on Computer Vision*, pp. 1944–1951, 2013.

[76] A. Asthana, S. Zafeiriou, S. Cheng, and M. Pantic, "Robust discriminative response map fitting with constrained local models," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 3444–3451, 06 2013.

[77] A. Asthana, S. Zafeiriou, S. Cheng, and M. Pantic, "Robust discriminative response map fitting with constrained local models," in *2013 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3444–3451, 2013.

[78] X. Cao, Y. Wei, F. Wen, and J. Sun, "Face alignment by explicit shape regression," in *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2887–2894, 2012.

[79] S. Ren, X. Cao, Y. Wei, and J. Sun, "Face alignment at 3000 fps via regressing local binary features," in *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1685–1692, 2014.

[80] Shizhan Zhu, Cheng Li, C. C. Loy, and X. Tang, "Face alignment by coarse-to-fine shape searching," in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4998–5006, 2015.

[81] S. Wei, V. Ramakrishna, T. Kanade, and Y. Sheikh, "Convolutional pose machines," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4724–4732, 2016.

[82] T. Baltrusaitis, A. Zadeh, Y. C. Lim, and L. Morency, "Openface 2.0: Facial behavior analysis toolkit," in *2018 13th IEEE International Conference on Automatic Face Gesture Recognition (FG 2018)*, pp. 59–66, 2018.

[83] A. Zadeh, T. Baltrusaitis, and L.-P. Morency, "Convolutional experts constrained local model for facial landmark detection," pp. 2051–2059, 07 2017.

[84] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell, "Caffe: Convolutional architecture for fast feature embedding," *MM 2014 - Proceedings of the 2014 ACM Conference on Multimedia*, p. 675–678, 06 2014.