HOCHSCHULE
MITTWEIDA
University of Applied Sciences

MASTER THESIS

Ms.
Navina Halbe, B.Sc.

Analyse der forensischen Aufbereitung biometrischer Gesichtsmerkmale für die digitale Nutzer Authentifikation

Mittweida, October 2023

Faculty of **Applied Computer Sciences and Biosciences**

MASTER THESIS

Analyse der forensischen Aufbereitung biometrischer Gesichtsmerkmale für die digitale Nutzer Authentifikation

Author:
Navina Halbe

Course of Study:
Cybercrime/Cybersecurity

Seminar Group:
CY21wC-M

First Examiner:
Prof. Ronny Bodach

Second Examiner:
Marius Eggert, M.Sc.

Submission:
Mittweida, 06.10.2023

Defense/Evaluation:
Mittweida, 2023

Faculty of **Applied Computer Sciences and Biosciences**

MASTER THESIS

Analysis of the Forensic Preparation of Biometric Facial Features for Digital User Authentication

Author:
Navina Halbe

Course of Study:
Cybercrime/Cybersecurity

Seminar Group:
CY21wC-M

First Examiner:
Prof. Ronny Bodach

Second Examiner:
Marius Eggert, M.Sc.

Submission:
Mittweida, 06.10.2023

Defense/Evaluation:
Mittweida, 2023

Bibliographic Description:

Halbe, Navina:
Analyse der forensischen Aufbereitung biometrischer Gesichtsmerkmale für die digitale Nutzer Authentifikation. – 2023. – 81 S.
Mittweida, Hochschule Mittweida – University of Applied Sciences, Faculty of Applied Computer Sciences and Biosciences, Master Thesis, 2023.

Referat:

Die Biometrie ist eine beliebte Methode für die Zugriffsbeschränkung auf sensible Daten. Sie bietet ein hohes Maß an Nutzerfreundlichkeit, da sich Nutzer kein Passwort merken müssen. Mit dem immer häufigeren Einsatz biometrischer Systeme erhöhte sich auch die Anzahl der Angriffe auf die Schwachstellen solcher Systeme. Andererseits bietet das Ausnutzen dieser Schwachstellen auch neue Möglichkeiten in Bezug auf die Strafverfolgung.

In dieser Thesis werden Ansätze evaluiert, wie elektronische Geräte mit gefälschten Gesichtern entsperrt werden können um Zugriff auf für die Strafverfolgung relevante Daten zu erhalten. Hier ist vor allem eine schnelle Datenerhebung notwendig, da es sich meist um zeitkritische Aufträge handelt. Das Entsperren des Geräts mithilfe des Nutzerpassworts kann mehrere Jahre dauern, wenn hierfür ein Brute-Force-Angriff ausgeführt werden muss. Dementsprechend kann die biometrische Entsperrung des Geräts eine schnellere Alternative darstellen.

Um aktuelle Technologien für die Gesichtserkennung zu überwinden, wurden verschiedene Ansätze untersucht. Die ersten Versuche beinhalteten das Drucken des Gesichts auf Papier, um Geräte zu entsperren, deren Gesichtserkennung auf der Darstellung des Nutzers im sichtbaren Spektrum basiert. Weitere Ansätze basierten auf dem Drucken des Infrarotbildes des Gesichts und der Erstellung von dreidimensionalen Gesichtsmasken, um Geräte zu entsperren, welche ihre Gesichtserkennung im Nahinfrarotbereich durchführen. Zusätzlich wurde die zugrunde liegende Software hinsichtlich ihrer Arbeitsweise analysiert.

Die Experimente zeigen, dass es teilweise möglich ist, die Gesichtserkennung mit gefälschten Gesichtern zu umgehen und Zugriff auf gesicherte Daten zu erlangen. Geräte, deren Gesichtserkennung auf dem sichtbaren Spektrum basiert, können mit dem gedruckten Gesicht des Nutzers entsperrt werden. Von den Geräten, welche eine Gesichtserkennung im Nahinfrarotbereich durchführen, kann ein Gerät mit einer dreidimensionalen Gesichtsmaske entsperrt werden. Weiterhin können Informationen über die Funktionsweise der Gesichtserkennung dieses Geräts in der zugrunde liegenden Software gefunden werden. Weitere Geräte bleiben gesperrt und deren Software liefert keine Anhaltspunkte bezüglich der Funktionsweise der Gesichtserkennung.

Abstract:

Biometrics has become a popular method of securing access to data as it eliminates the need for users to remember a password. Although exploiting the vulnerabilities of biometric systems increased with their usage, these could also be helpful during criminal casework.

This thesis aims to evaluate approaches to bypass electronic devices with forged faces to access data for law enforcement. Here, obtaining the necessary data in a timely manner is critical. However, unlocking the devices with a password can take years with a brute force attack. Consequently, biometrics could be a quicker alternative for unlocking.

Various approaches were examined to bypass current face recognition technologies. The first approaches included printing the user's face on regular paper and aimed to unlock devices performing face recognition in the visible spectrum. Further approaches consisted of printing the user's infrared image and creating three-dimensional face masks to bypass devices performing face recognition in the near-infrared. Additionally, the underlying software responsible for face recognition was reverse-engineered to get information about its operation mode.

The experiments demonstrate that forged faces can partly bypass face recognition and obtain secured data. Devices performing face recognition in the visible spectrum can be unlocked with a printed image of the user's face. Regarding devices with advanced near-infrared face recognition, only one could be bypassed with a three-dimensional face mask. In addition, its underlying software provided evidence about the demands of face recognition. Other devices under attack remained locked, and their software provided no clues.

Contents

List of Figures

List of Tables

1 Introduction

Using biometric systems to determine or confirm a claimed identity by relying on biometric features that uniquely identify a person, started centuries ago. In 1883, Alphonse Bertillon introduced the system for measuring the body, called "bertillonage", the first to use body measurements to identify people in criminal cases. This became the foundation for biometric systems, which then continuously developed [1, 2]. Since the beginning of the 21st century, they have been used in a wide variety of commercial applications to secure access to data or places. This type of access security has the advantage that users do not have to remember access codes as they always carry their biometric features, increasing user-friendliness [2].

Because of its uniqueness, the human face is the most visually memorable part of the human body [3]. Subsequently, it is considered a biometric trait and often serves to identify individuals, for example, in law enforcement or for user authentication [4, 5]. It provides some advantages against other biometric features as the capturing is effortless, less interaction is required, and it offers the option to perform the capturing from a distance [6].
Already in 1992, Christopher Hoogsteden mentioned that face spoofing could be a potential risk for biometric systems [7]. With the increase in their usage, the number of attack vectors against such systems, mainly targeting the sensor that records the face, also expanded. Therefore, the attacker creates an artefact of the original biometric data to fool the sensor. Due to the increasing use of social networks, finding suitable images for creating faked faces is becoming much easier [8]. Although vulnerabilities of biometric systems can be exploited for criminal purposes, creating a forged face can also help to unlock smartphones and thus access data more quickly during criminal casework.

This chapter starts with the motivation for this research topic (see Section 1.1). Next, in Section 1.2, the aim of this thesis and the applied methods are outlined. Lastly, we explain the structure of this thesis in Section 1.3.

1.1 Motivation

State-of-the-art chip technologies often contain encrypted files and programs. In particular, modern smartphones usually implement full encryption. These cryptographic methods offer a high degree of mathematical security, posing problems for digital forensics, especially in law enforcement. Without the user's password, only the encrypted data is accessible. However, testing all password possibilities by executing a brute force attack can take several years [9]. Therefore, it would be favourable to shorten the readout time.

As already mentioned, the usage of biometric systems granting access to data increased over the last decades. Accordingly, many smartphones and other electronic devices have an integrated biometric system to secure access to data, with facial recognition becoming more popular. Besides being more secure than a password that can be stolen, face recognition offers the advantages of being performed contactless, fast and automatically.

Regarding law enforcement, forcing the suspect to present their biometric feature to the device for unlocking is legally controversial according to the German Code of Criminal Procedure § 81b: *Identification measures on the suspect*. However, as the identification treatment implies the taking of photographs, creating forged faces would be possible [10].

For these reasons, new approaches are desirable to unlock devices with faked faces. As mentioned, obtaining data to generate facial artefacts is more manageable than other biometric features. Compared to password cracking, using faked faces to unlock the device could shorten the readout time tremendously.

1.2 Aim of this Thesis

This thesis researches the screen lock of devices in the context of biometric features by recreating faces to bypass facial recognition. Before the first unlock, the devices cannot be unlocked with biometric features as they always require a password; subsequently, this thesis concentrates on attacking devices after their first unlock. Some modern end devices no longer have an integrated fingerprint sensor and only offer facial recognition as a biometric variant for unlocking. Therefore, methods for unlocking the devices with facial features should be evaluated, especially for time-critical orders.

In this study, we examine approaches to bypass face authentication to unlock the device, as the forging of faces takes days, which shortens the readout time compared to executing brute force attacks, which can take several years. As the devices' face recognition relies on different technologies, several approaches must be examined to create forged faces that suit the sensor's demands.

Throughout this thesis, we conduct experiments to answer the following research questions:

1. How can state-of-the-art facial recognition be circumvented on electronic devices such as smartphones, tablets and laptops?
2. What are the sensor's limits regarding colouring and resolution, the face positioning and the required parts of the face?
3. Can the underlying face recognition software provide clues on how facial recognition works?

1.3 Structure of this Thesis

This thesis is divided into three chapters. Chapter 2 outlines the background concerning biometrics itself and biometric systems with their structure, occurring errors and their evaluation (see Section 2.1). In addition, the biometric trait face and its processing for authentication are explained in Section 2.2. The last two Sections 2.3 and 2.4 discuss the types of attacks on the system's components and outline the state-of-the-art approaches to bypass face recognition.

Next, Chapter 3 explains the approaches carried out and interprets their results. In the beginning, the experimental setup is described (see Section 3.1) by listing the devices under attack, the applied materials for the mask creation and the additional hardware and software. Afterwards, Section 3.2 examines the attacks on devices performing face recognition in the visible spectrum and their results. Then, Section 3.3 outlines the attacks on devices performing face recognition in the NIR spectrum, including the devices' requirements, the software analysis and the attack evaluation.

Lastly, Chapter 4 summarises and interprets the results of the previously explained experiments. Furthermore, it gives an outlook on approaches which can be evaluated further.

2 Biometric Systems for Face Recognition

Biometrics is the technique of recognising an individual by personal characteristics and is defined as the science of body measurement of living beings. It originates from Greek and is composed of the words *bios* (life) and *metron* (measurement) [2]. The biometric characteristics are either physiological or behavioural [11]. Furthermore, biometrics is related to biometry being the "mathematical description and measurement in biology, primarily in the area of statistics" [2]. Biometrics became increasingly crucial as large-scale solutions for user authentication, such as access to secured areas in companies, were required [2].

The history of biometrics goes back centuries, where different approaches were taken into account to identify people. The oldest records refer to the use of fingerprints. These were used, for example, by the Assyrians and in the Tang Dynasty (618-906) for signing contracts or marking clay pots to identify the artist. Furthermore, Egyptians took advantage of the body height to grant access [2].
However, the real breakthrough for biometrics was in the 19th century. In 1879, Alphonse Bertillon developed a measuring system based on physiological features to identify a person [2]. He used *Anthropometry* as the doctrine of the measurements and proportions of the human body. It incorporates the processes to measure the body and skeletal features [12]. His Bertillonage system was employed in law enforcement and is considered the forerunner of today's digital biometric identification systems [2].
Additionally, in 1892, Sir Francis Galton identified the uniqueness of fingerprints. A few years later, they led to a criminal's conviction and replaced the previously introduced Bertillonage [2].

In the 20th century, automated user recognition was introduced, providing new possibilities for user authentication based on the fusion of biometrics and machines [2, 13]. First, in the 1960s, an automated fingerprint system was developed [2]. Face recognition has been an active research field since the 1970s. In 1971, Goldstein, Harmon, and Lesk wrote an article on how a face is identified by another person and by machines [13]. This paper served as a foundation for the following research concerning "real-time man-machine interaction for computer classification and identification of multidimensional vectors specified by noisy components" [13, p. 4]. In 1973, Takeo Kanade developed the first automated system for face recognition [14]. A system for iris and retina scans was introduced in the 1980s [2]. At the end of the 20th century, the first competition for face recognition systems occurred. Since then, biometric recognition systems have been commercialised [2].

For biometric recognition, the selected feature is measured to determine the person's identity or to verify or reject a claimed identity. This process applies in biometric procedures as well as in biometric systems. A *biometric procedure* relies on biometric recognition to authenticate a person with their behavioural and physiological features. A *biometric system* goes one step further by combining software and hardware for biometric identification and verification. Furthermore, it uses biometric procedures to execute its tasks [5].
Biometric procedures can be performed because certain feature expressions can be assigned to a person, which enables user recognition. Additionally, this type of features offers several advantages. First, physiological characteristics are physically linked to the person, meaning no need for an artificial assignment exists. Second, unlike possessions, a body-based feature cannot be lost; the user always carries them and does not have to remember anything [2].

Lastly, biometric features do not have to be kept secret. They are visible but cannot be transferred or passed on [2]. Nevertheless, it is possible to forge a biometric feature. However, the effort to create an artefact can be high [5].

A biometric trait must fulfill the properties uniqueness, constancy, and distribution to be considered biometric. Thus, the feature should occur in sufficiently different variants, remain as unchanged as possible over a certain period, and be represented frequently enough within a population [5].
The biometric features can be classified based on their origin. First, *genotypic traits* are genetically determined and partly hereditary. For example, the origin-related bone structure and heredity determine the shape of the face. Furthermore, traits can be *randotypical*. They arise due to random processes in the embryonic phase. Finally, they are *conditioned*, which is behaviour-based and can be acquired [2]. However, this classification is only sometimes clear, as certain characteristics can be assigned to several categories. For example, the voice depends on anatomical conditions. In addition, the pitch of the voice can change with age due to development and can vary depending on the person's emotional state [5].
Therefore, biometric features are differentiated into the categories active and passive. Passive biometric traits are physiological and, therefore, related to the body. Those do not change over time. In contrast, active traits are behavioural traits. Those can change over time due to specific influences [5]. Table 2.1 lists the active and passive biometric features.

Table 2.1: Passive and active biometric traits [5]

Passive, physiological traits	**Active, behavioural traits**
Eyes, iris or retina	Lips, tongue
Face	Tiping behaviour
Nose, ear	Voice profile
Fingerprint	Handwriting
Hand geometry, hand line structure	Smell
DNA	Aisle
Vein pattern	
Blood	

In the following, we will concentrate on the passive trait face as it provides advantages compared to features like fingerprints or the iris. The capturing can be performed contactless and at a distance. Additionally, it is a non-intrusive process [15]. Especially for passenger documents, the face provides "the highest compatibility in a Machine Readable Travel Documents (MRTD) system" [15, p. 1] compared to other biometric features. With the increase of image capture devices and the available face data on the web, face recognition has become more important [15].

Face recognition technology comes with advantages and disadvantages. The recognition systems have a high error rate. Small changes such as a beard or a different hairstyle can be enough to make someone unrecognisable or look too similar to another person. Those changes are especially critical in law enforcement because an innocent person could be convicted. Additionally, machine learning is often used to perform face recognition. Those algorithms need huge amounts of training data to perform well. Here, it is difficult to collect enough and also representative data [16].

Nevertheless, face recognition provides some advantages. Besides a user-friendly way to unlock a mobile device, face recognition offers higher security. Criminals can be determined, and access to sensitive data can be restricted. Furthermore, the data processing is fast. Therefore, a person is identified directly. Lastly, face recognition can be integrated with other technologies, providing lower costs to extend existing software [16].

This chapter provides an overview of biometric systems. The first part explains the access control's components. Next, we analyse the structure of biometric systems and go into more detail about the individual components (see Section 2.1.1). Additionally, the errors that can occur in biometric systems are explained. Section 2.1.3 lists and discusses the criteria for evaluating a biometric system and choosing a biometric trait.
The second part deals with processing the face data and face authentication (see Section 2.2). It offers an overview of the required steps to process the captured data performing face recognition. Following, the face authentication process is outlined. Here, we explain the state-of-the-art techniques in more detail.
The third Section 2.3 discusses the types of attacks to target biometric systems. The terminology is explained here based on a standard provided by the *International Organization for Standardization and International Electrotechnical Commission* (ISO/IEC). Additionally, anti-spoofing techniques are displayed.
The last part, Section 2.4, outlines the existing approaches to bypass biometric face recognition. Furthermore, the differences between the presented approaches and the procedure of this thesis are clarified.

2.1 Biometric Systems

A biometric system works with pattern-based recognition, recognising shapes and performing a classification based on the extracted features [17, 18]. Regardless of the biometric trait, the fundamental mode of operation remains constant. First, the system captures a person's biometric trait. Next, it extracts features from the captured data. It matches those extracted features against a stored template feature set. If they match, the system confirms the right identity or rejects the person [17]. In contrast to forensic applications, biometric systems require an automated recognition based on measuring a specific biometric trait of a living person in real time [2].

Due to increased digitalisation and automation, biometric systems are now firmly anchored in IT security for user authentication. The growing use of such systems is mainly due to their intuitive usability and reliable results [19]. They are applied in commercial, governmental and forensic applications [17]. Commercial applications are, for example, online banking or smartphones, whereas governmental applications are border controls or passport controls [19]. Forensic applications use biometric systems to identify corpses, missing persons or criminals [17].

Access control consists of four steps. The first step is identification, where the user enters his ID. The system looks up the ID in the database and identifies the user. The next step is authentication. Therefore, the user enters his password or presents his biometric feature. The system compares the entry with the stored password or the biometric template of the selected ID [20].

The following step is authorisation. Here, the system authorises the user if the authentication was successful, thereby granting or rejecting access to the requested resources. The last step is accountability. It describes the storage of system logs, logging all successful and rejected login attempts [20].

A biometric system can operate in different modes. In the identification mode, the system proceeds a one-to-many comparison and searches for the matching template by going through all the templates stored in the database. The user does not have to claim an identity. Subsequently, the system answers, "Whose biometric data is this?". It identifies the user and responds with the identity if it finds a match. Otherwise, it rejects the user [11].
Identification is important in terms of negative recognition. In this case, "the system establishes whether the person is who[m] she [...] denies to be" [11, p. 6]. It is a prevention that a person cannot take more than one identity and can only be executed with a biometric trait instead of passwords [11]. For identification, the user presents his biometric trait, and special features are extracted, resulting in an input feature vector X_Q. It is used to determine the identity $I_k, k \in \{1, 2, ..., N\}$. The identities $I_1, I_2, ..., I_N$ are the enrolled users and I_{N+1} would be a non-enrolled user. A similarity function S calculates the similarity between the input features X_Q and the stored template X_{I_k} from identity I_k. Then, the identity with the maximum similarity score is chosen. Their similarity score must be greater or equal to a certain threshold t to identify the user. Otherwise, the user is rejected [17]. Therefore, the system performs a type of classification where X_Q is categorised either as a known user I_k or as a non-enrolled user I_{N+1} (see Equation 2.1) [17]. Figure 2.1 displays the identification process.

$$X_Q \mapsto \begin{cases} I_k, & if \max_k \{S(X_Q, X_{I_k})\} \geq t, k = 1, 2, ..., N \\ I_{N+1}, & otherwise \end{cases} \tag{2.1}$$

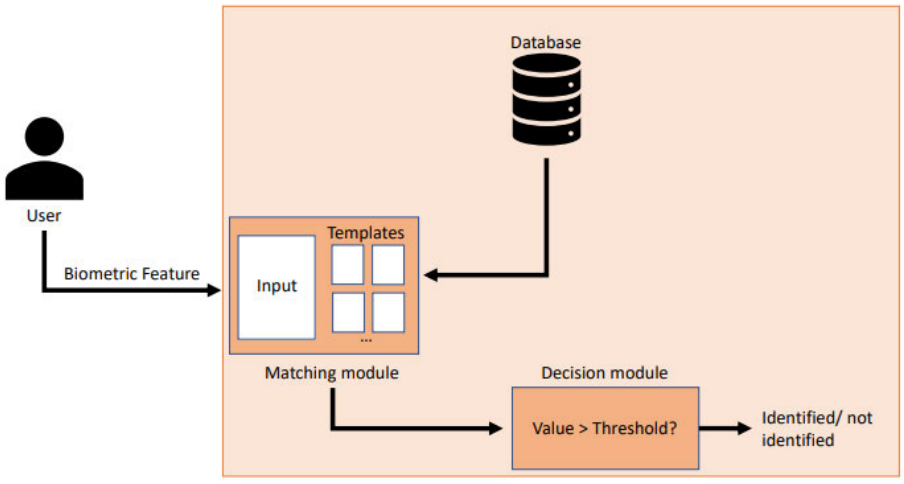

Figure 2.1: Biometric identification process [21]

Unlike identification, verification is a one-to-one comparison. The user claims an identity, and the system compares the provided data with the biometric template of the claimed identity stored in the Database. As a result, the system validates the claimed identity or rejects the user. Here, "Validation is the confirmation by providing objective evidence that the requirements for a specific intended application have been fulfilled" [21, p. 7]. Therefore, it answers the question, "Am I who I say I am?". It is a prevention that more than one person cannot take the same identity [11].

The advantage of verification is that it is less time-consuming than the identification process, as the system only executes one comparison and does not need to compare the entire database [5]. Both identification and verification work with a threshold, as even the same biometric feature of one user never gets the same result due to variating positions [17].
Authentication is similar to verification and is sometimes used as a synonym. However, it is a process to grant a user access to an information system, whereas verification is proof that a fact is true [22]. The process can be explained as follows. The user claims his identity I as an ID and then presents his biometric trait. The system extracts special features, resulting in an input feature vector X_Q. The system then classifies the combination (I, X_Q) with a categorisation function c into w_1 or w_2 where w_1 corresponds to a true claim and w_2 to a false claim. Here, the similarity function S calculates the similarity between X_Q and X_I being the stored feature template from the claimed identity. If the result is greater or equals a certain threshold t, (I, X_Q) is categorised in w_1. Otherwise, the system categorises it in w_2. Equation 2.2 represents this categorization [17]. Figure 2.2 shows the described authentication process.

$$c(I, X_Q) = \begin{cases} w_1, & if\{S(X_Q, X_I)\} \geq t \\ w_2, & otherwise \end{cases} \tag{2.2}$$

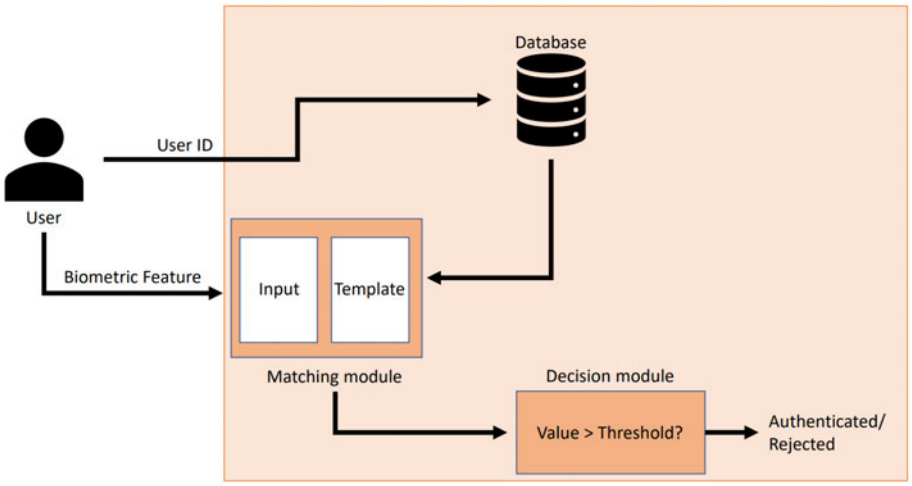

Figure 2.2: Biometric authentication process [21]

Validation is a critical aspect of identity verification that involves one of three different methods: *something you know*, *something you have*, and *something you are* [20]. The first method, *something you know*, typically involves using passwords to verify identity [20]. While this method is prevalent in commercial applications, it has its challenges [17]. Longer passwords are generally more secure, as they increase the duration of brute-force attacks. However, passwords can easily be forgotten or stolen, making the system vulnerable [20].
The second method, *something you have*, often uses smart cards to verify identity and is commonly used in governmental applications [17, 20]. Smart cards offer the possibility of two-factor authentication, typically used with a PIN. However, this method also has weaknesses, as some users write his PIN directly on the card to avoid having to remember it. When a card is stolen with the PIN written on it, it exposes vulnerabilities in the system [20].

The third and final validation option, *something you are*, involves using biometric features to verify identity [20]. This method is frequently used in forensic applications, where experts manually match biometric features [17]. While biometric features are unique to each individual, cannot be lost and are difficult to duplicate, they are also susceptible to errors [20].
Overall, each of these validation methods has its strengths and weaknesses, and it is essential to choose the appropriate method based on the application's specific requirements. Organisations can ensure secure, reliable, and effective validation processes by carefully considering these factors.

2.1.1 Structure of Biometric Systems

As already mentioned, a biometric recognition system is pattern-based. First, it captures the user's biometric trait, extracts the special features, and compares the resulting dataset to a reference dataset from the database [11]. The primary mode of operation is the same for all biometric systems. They include a module to register a new user, capture the user's biometric trait, generate the datasets and compare the newly captured one with those stored in the system. Whether the system performs enrolment or user recognition, it first captures the biometric trait. During registration, the system creates a reference dataset and later on another one for recognition [2]. Figure 2.3 displays the general structure of a biometric system.

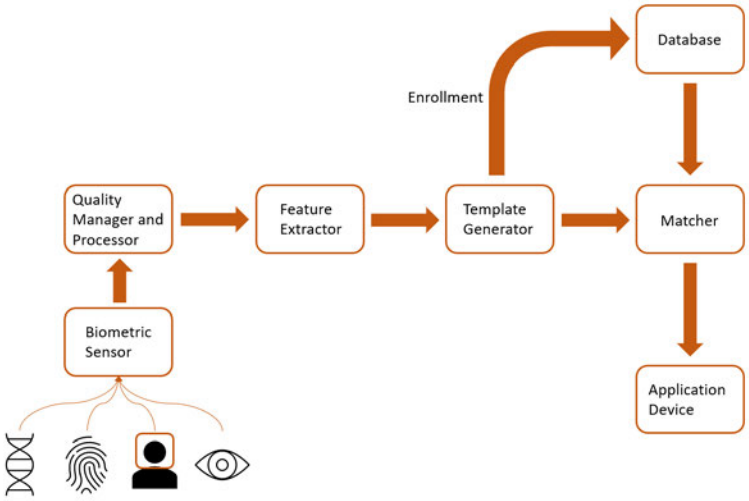

Figure 2.3: Structure of biometric systems

A biometric system consists of seven main modules. The first module is a **Biometric Sensor**, acting as an interface between the user and the machine. It can, for example, be a camera, scanner or reader capturing the biometric trait as raw data. Here, its quality is crucial for the system's performance. If it provides low-quality raw data, the system produces a high *Failure to Acquire (FTA)* rate, leading to low accuracy [11]. Later, in Section 2.1.2, these are explained in more detail.
The next module is a **Quality Manager and Processor**. It examines the quality of the captured data which must meet certain quality criteria for the next steps. Therefore, it is possible to preprocess the data with an enhancement algorithm to provide higher-quality data. Next, the captured data is processed to prepare it for feature extraction [11]. Section 2.2 explains this processing in detail.

The processed data is handed to the **Feature Extractor**. This module extracts special predefined features from the biometric trait to distinguish between users. Those features represent the biometric trait. Commonly, these features are saved in a numerical form such as a vector [11].
Next, the extracted features are stored as a dataset named template, created from the **Template Generator**. This template is handed to the **Database**, saving the reference datasets during enrolment [11]. Enrolment is the procedure of initial recording. The user's record is added to the Database, so the numerical form of the biometric data is stored for later comparison. Here, the focus lies on the Quality Manager and Processor to ensure sufficient recordings [23].
It is possible to work with multiple recordings. The system either selects the one with the highest quality or fuses the recordings to create a template. Another option would be to store multiple templates for one user [23]. This technique is especially used for face recognition due to different outer appearances or head positions. Therefore, the system achieves better results in terms of intra-class variations [11]. An *intra-class variation*, also known as intra-class correlation coefficient, is a measure of the correlation of repeated measurements of a characteristic on the same person. It indicates the reliability of a series of measurements [24]. The created template with the best quality or the template set is then stored in the Database with the user information [11].
The **Matcher** takes over the main task of the biometric system. It compares the extracted features with the stored templates from the Database and then generates a matching score. This score depends partly on the data's quality. The Matcher also evaluates the matching scores. For verification, he validates or rejects the claimed identity. For identification, the Matcher generates matching scores with every template in the Database, then ranks the templates and selects the template offering the best matching score to identify the user [11].
The last module is the **Application Device**. This module gets the final boolean result from the Matcher. Depending on the answer, the system grants or denies access to the user [11].

2.1.2 Biometric System Errors

As mentioned, two presentations of a user's biometric trait are always different due to certain influences like image properties, fluctuations in the user's body-based or behavioural characteristics or the interaction between the user and the sensor. If there is a perfect match, this could even indicate the presentation of a forgery. Such differentiating feature sets of the same user's biometric trait form the *intra-class variation* described in the section above. In contrast, the *inter-class variation* contains feature sets from the same biometric trait of different users [11]. Inter-class variation describes the fluctuation of a characteristic between users. Therefore, the goal is to provide a low intra-class variation and achieve a high inter-class variation. A low intra-class variation indicates good permanence and repeatability, whereas a high inter-class variation indicates that the chosen feature is qualified to distinguish between users [25].

As described, the matching score $S(X_Q, X_I)$ quantifies the similarity between the captured feature set X_Q and the saved template X_I for user I. The higher the score, the greater the similarity. From a mathematical point of view, two hypotheses can be formulated [17]:

1. Null hypothesis (H_0): $the\ captured\ feature\ set\ X_Q\ does\ not\ belong\ to\ the\ same\ person\ as\ template\ X_I$
2. Alternate hypotheses (H_1): $the\ captured\ feature\ set\ X_Q\ belongs\ to\ the\ same\ person\ as\ the\ template\ X_I$

Subsequently, the system's decisions can be formulated as follows [17]:

1. D_0: $The\ user\ is\ not\ who\ he\ claims\ to\ be$
2. D_1: $The\ user\ is\ who\ he\ claims\ to\ be$

The threshold t helps to decide whether both datasets belong to the same user and influences the system's decisions. If $S(X_Q, X_I)$ is greater or equals t, the system assigns both data sets to the same user and respectively decides D_1. Such datasets are defined as *genuine* and form the *genuine distribution* $p(S(X_Q, X_I)|H_1)$ which consists of scores generated from multiple images of the biometric trait of one user. It is described as the probability of the score $S(X_Q, X_I)$ under the condition that the captured feature set belongs to the same person as the template [17].
If $S(X_Q, X_I)$ is smaller than t, the system rejects the user and respectively decides D_0. Those datasets are called *impostors* and form the *impostor distribution* $p(S(X_Q, X_I)|H_0)$, meaning the probability of a score $S(X_Q, X_I)$ under the condition that the captured features do not belong to the same user as the feature set. These are scores generated from multiple images of the biometric trait of different users [17].
Figure 2.4 displays these distributions. The x-axis plots the matching score leading from $-\infty$ to ∞, and the y-axis shows the probability p leading from 0 to 1.

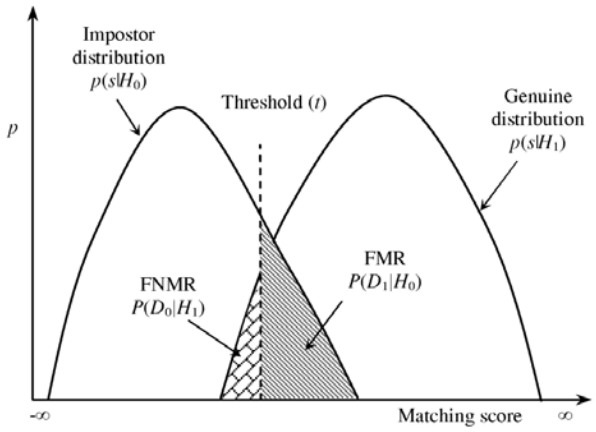

Figure 2.4: Diagram of the *genuine* and the *impostor distribution* with occurring errors [17]

Biometric systems deal with two main errors. First, the *False Acceptance Rate (FAR)*, respectively *False Match Rate (FMR)*, represents the error case where an impostor achieves a matching score that exceeds the threshold [11]. Mathematically, the system decides D_1 when H_0 is true [17]. Therefore, the system accepts an unauthorised user. In Figure 2.4, the *FMR* corresponds to the dashed part of the impostor distribution that exceeds the threshold. It is displayed as $P(D_1|H_0)$, representing the probability that the system determines that the user is who he claims to be under the condition that the captured feature set does not belong to him [17]. Therefore, the *FMR* calculates as follows [26]:

$$FMR = \frac{Number\ of\ unauthorized\ users\ that\ are\ incorrectly\ accepted}{Total\ number\ of\ unauthorized\ users} \tag{2.3}$$

Second, the *False Reject Rate (FRR)*, respectively *False Nonmatch Rate (FNMR)*, represents the error case where a genuine user achieves a matching score lower than the threshold and is therefore wrongly rejected [11]. Mathematically, the system decides D_0 when H_1 is true [17]. These errors are displayed as the tiled part of the genuine distribution that falls below the threshold (see Figure 2.4).

It is displayed as $P(D_0|H_1)$, signifying the probability of the system establishing that the user is not who he claimed to be under the condition that the captured feature set belongs to him [17]. The formula is [26]:

$$FNMR = \frac{Number\ of\ authorized\ users\ that\ are\ incorrectly\ rejected}{Total\ number\ of\ authorized\ users} \tag{2.4}$$

The *FNMR* is "inversely proportional" [26] to the *FMR*. Both errors depend on the selected threshold. If t is decreased and the system is more tolerant than before, the *FMR* increases and the *FNMR* decreases. If, on the other hand, the system is to be made more secure and the threshold increases, user-friendliness suffers, and the *FNMR* increases, whereas the *FMR* decreases. However, not every user achieves the same number of false accepts and rejects. This number depends on the distinctiveness of the user's extracted features [11].

The *National Institute of Standards and Technology (NIST)* has conducted a series of trials called the *Face Recognition Vendor Test*. In 2002, they examined the average *FNMR* and *FMR* where the test conditions include variations in lighting, in- and outdoor settings and different times of day [27]. When it comes to data sets captured indoors, the effect of the varying indoor lighting is rather small. Here, the best systems reached an *FMR* of 1% whereas the *FNMR* lays by around 10%. The system's performance decreased regarding the recognition rate with faces captured outdoors. The best system also achieved a *FMR* of 1%. However, the *FNMR* was 50% [27].
Additionally, the systems' performances were measured regarding time increases between the enrolment and the recognition. Here, the researchers examined that the recognition performance decreases around 5% per year [27].

Additional errors can occur in a biometric system. First, the *Failure to Acquire (FTA)* rate respective to the *Failure to Capture (FTC)* rate describes how often the system fails to record the presented biometric trait even if it provides sufficient quality. This error depends on the sensor's quality. A high-quality sensor has fewer problems with capturing a low-quality biometric trait than a cheaper one [11].
Second, the *Failure to Enroll (FTE)* rate can increase. This error represents the number of users that cannot be registered successfully. It also depends on the presentation's quality of the biometric trait. The system's user-friendliness should increase with helpful instructions on how the user should present his biometric trait to achieve better results [11].

A *Receiver Operating Characteristics (ROC)* curve measures the performance of a biometric system. The correct acceptance rate $(1 - FNMR)$ is typically plotted against the false acceptance rate. However, in this case, the *FMR* is plotted against the *FNMR* to display the system's performance for different application areas (see Figure 2.5) [17].

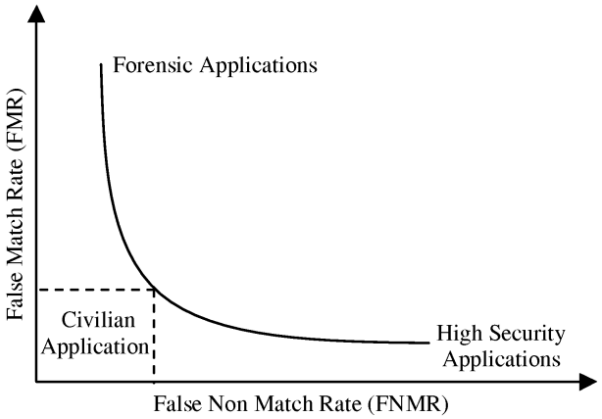

Figure 2.5: ROC curve, where the *FMR* is plotted against the *FNMR*, displaying the system's performance in different applications [17]

The focus of the system's performance depends on the underlying application. Achieving a low *FNMR* is crucial in forensic applications. For example, in terms of criminal identification, it is more important to identify every criminal. Due to the high *FMR*, false matches occur more often. However, this case is not so problematic as these can be checked and erased manually [17].
In contrast, when it comes to high-security applications, a minimal *FMR* is crucial as the goal is to detect every impostor so no unauthorised user gains access. In this case, it is considered that the system could reject some legitimate users due to the high *FNMR* [17].
Commercial applications lie between these extremes. A compromise between *FMR* and *FNMR* has to be made. For example, in banking, a falsely accepted unauthorised person would mean the loss of significant sums of money. However, falsely rejecting an authorised person would also result in customer terminations, causing the bank to lose money [17].

2.1.3 Criteria to Evaluate Biometric Systems and Features

There are several requirements for a biometric system to be usable in practice. It must fulfil certain standards regarding recognition quality, speed and resources. Additionally, the usage should be safe for the user, and the target group should accept it. Another criterion is the robustness against attacks. The biometric system should not be easily attacked and bypassed by impostors [17]. However, with the increased use of biometric systems, the attack surface on such systems is also increasing with new attack vectors. Biometric systems must be carefully evaluated and expanded to ensure security and reliability [19].

Common evaluation criteria are performance, usability and vulnerability. The **performance** aims to provide good recognition accuracy. Here, the *FNMR* is measured. Subsequently, how often a legitimate user is categorised as non-authorised is observed. Additionally, it examines the influence of environmental conditions on the system's performance [19].
Usability reflects the system's user-friendliness. Here, it is checked whether the system is easy to understand by the user and if he can use it correctly. This applicability for the target group is crucial. The more the user understands how to interact with the system, the higher the quality of the recorded data [19].

The system's **vulnerability** represents the system's resistance against different attacks. Those are explained later in Section 2.3. The sensor's functions are regarded by how well it determines between the skin and other materials. Furthermore, the system should differentiate between artefacts like masks and real biometric traits [19].

Not every biometric trait is equally suitable for every application. Each feature has advantages and disadvantages for specific applications. To evaluate which one is best suited for a particular application, there are seven criteria against which the traits are considered. First, the four general criteria of universality, distinctiveness, permanence and collectivity assess the characteristics' expression. The three remaining criteria, performance, acceptability and circumvention, are crucial for practical applicability in biometric systems [17]. The criteria and their description are listed below [11]:

Universality	Every user should have this trait.
Distinctiveness	It should be individually formed to differentiate sufficiently between two users.
Permanence	It should remain as unchanged as possible concerning its characteristics over a certain time.
Collectability	It should be quantitatively measurable.
Performance	The components of the biometric system should be adjusted to the feature in such a way that the required quality in terms of recognition and speed is achieved despite certain environmental conditions.
Acceptability	The target group should be willing to use this biometric trait.
Circumvention	The system should not be bypassed easily. Consequently, the effort to forge the biometric trait should be high.

Regarding the face, the universality is high as everybody has one. In contrast, its distinctiveness is low because each face offers the same features, with a few different expressions. The permanence is classified as medium. The facial features remain the same, but due to ageing, they slightly change over time. The collectability of a face is high as it can be appropriately measured. However, the performance could be better because it can be difficult for the system to work correctly under different lighting conditions. Furthermore, the acceptability and circumvention are high. For the user, it takes no effort to present his face. Additionally, creating a facial artefact takes time and effort [17].

2.2 Biometric Face Processing and Authentication

Face recognition is a biometric approach to identifying the user's biometric facial features. As a visual pattern problem, the device recognises the picture as a pixel matrix. The goal of the machine is to identify the parts of the matrix which correspond to the searched pattern [28].

A face recognition system consists of four modules executing face localisation, normalisation, feature extraction and matching. Face localisation and normalisation are included in the Quality Manager and Processor module [15]. The process is displayed in Figure 2.10.

Face localisation consists of two steps. First, the **Face Detector** localises the face and is subsequently responsible for the system's performance [15, 29]. It has to be capable of detecting every face in the input stream regardless of position, rotation, expression and lighting conditions [15].
To detect a face, the Face Detector orientates on key features such as the skin colour for coloured images, face motion in videos, the shape of the face, the appearance of facial features or a combination of these key features. The strongest algorithms rely on facial appearance [29].
The Face Detector searches for the face coordinates in the image. The algorithm goes over the images and shows if a face is present. Therefore, it creates a subwindow and classifies the area of the subwindow into face or non-face [29]. If it finds one, it returns its location and the approximate size [28].
During detection, it is possible to detect one face multiple times (see Figure 2.6). This multiple detection in a specific neighbourhood around a returned location can indicate a face [29]. Therefore, "a detection is confirmed if the number of multiple detections is greater than a given value" [29, p. 295]. Subsequently, the multiple hits are merged into one (see Figure 2.7) [29].

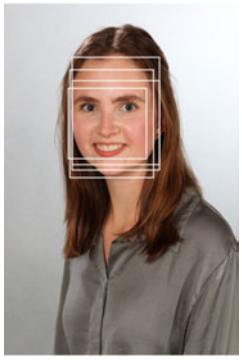

Figure 2.6: Multiple detection of a face in a specific neighbourhood around a returned location

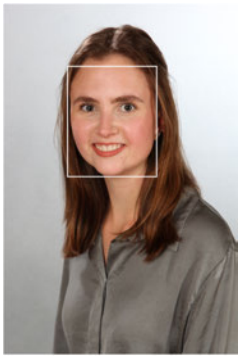

Figure 2.7: Multiple detections of a face merged into one

The Face Detector then separates the background from the face and executes the landmark localisation by identifying the essential points of a face [15]. These points are often used for enhancing face recognition or aligning face images. Usually, there are 68 landmarks in total, which are shown in Figure 2.8 [30].
First, the eyes and mouth region have to be found. Therefore, a rectangular search is executed [31]. In face verification, only a few facial landmarks like the eye's contours, the nose's tip and the mouth's outer areas are often used, or new points are calculated. For example, from the eye's shape, the eye's centre can be calculated and used as a feature [30]. This resulting selection is called facial nodal points, as shown in Figure 2.9. Here, the utilised landmarks are outlined in blue and new calculated points are displayed in blue. The face and its nodes are handed over to the next module.

Figure 2.8: 68 facial landmarks [30]

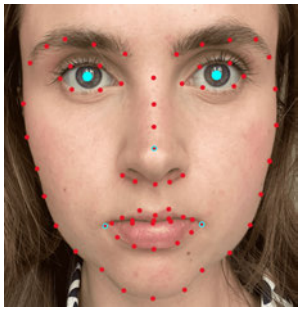

Figure 2.9: Nodal points consisting of landmarks (blue outlined) and new points (blue points) drawn with *GIMP* [32]

The **Face Normalizer** normalises the face photometrically, and geometrically. The goal is that face recognition works under different circumstances, like varying light conditions or poses. The Face Normalizer crops the face and transfers it into a standard frame for geometric normalisation. Depending on the algorithm, it may perform additional warping or morphing [15].
The photometric normalisation processes the face regarding its illumination and grey scale. As a result, the aligned face is transferred to the next module [15].
Next, the **Feature Extractor** extracts the important facial characteristics and converts them into a feature vector. These distinguish between users and must remain unchanged even under geometric or photometric modifications. The features are extracted and handed over to the last module to execute the matching [15].
Lastly, the **Face Matcher** executes the matching. Here, the recorded features are matched against features from stored templates. With a similarity metric, the Face Matcher calculates a similarity score. If the score is higher than a threshold, it returns the identity of a user and grants access, whereas a lower score leads to the user's rejection. Feature selection is crucial for high accuracy, so the system provides good results [15]. The *accuracy* measures the system's performance as it describes the proportion of correct decisions in all decisions made [33].

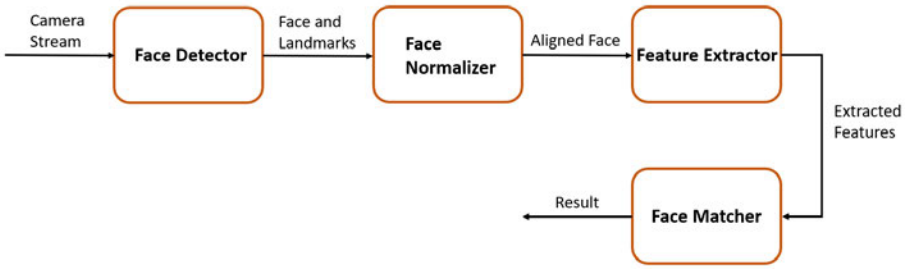

Figure 2.10: Face processing procedure

Face recognition applications can be categorised according to the user's interaction with the system. On one hand, in *cooperative* applications, the user interacts voluntarily with the system. Therefore, he presents his face according to the system's guidelines, e.g. frontal with open eyes and a neutral expression. Cooperative applications would be smartphone unlock or physical access controls [8].
On the other hand, *non-cooperative* applications deal with users not interacting properly with the system. Those can be found in surveillance applications. Here, the user does not know that he is identified and does not present his feature according to the system's guidelines [8].

In its early days, facial recognition was performed only under laboratory conditions, i.e. controlled conditions. This kind of face recognition was based on the spectrum that is visible to the human eye [34].
An RGB camera is used for *visible-spectrum-based face recognition*, providing a rectangular image matrix where each pixel is assigned a colour value. The RGB camera saves the colour values in typically 24 bits. Here, 8 bits are used for each base colour red, green and blue [35].
The authentication procedure is straightforward. First, the user has to be enrolled. Therefore, he has to present his face from different angles. Meanwhile, the software searches the facial features and extracts and saves them. When he wants to, for example, unlock his smartphone, the software compares the saved feature set with the one extracted from the live image. As this approach aims to provide fast face recognition, the goal is to find a trade-off between the system's accuracy and the recognition speed. However, if the speed is increased too much, the system's accuracy is lacking, and the system gets more vulnerable [36].

Under laboratory conditions, the systems achieved good results. Over the years, the designs have been adapted to work under actual conditions. With this change, the high recognition accuracy decreased [34]. Several factors influence the system's performance. The most known are low resolution, illumination, facial pose and expression, facial wear and motion [8, 15].
The varying lighting conditions had a significant influence on the system's performance. The lower the illumination, the higher the *signal-to-noise ratio* (SNR), which describes the balance of the power of the transmitted useful signal to the strength of the noise signal and measures the purity of a signal. It is specified in decibels (dB). A higher SNR means the signal is less noisy [34, 37]. The SNR also decreases with lower lighting, and automated data processing and face recognition get more complicated. Subsequently, night shots or capturing from a distance is challenging [34].
Additionally, the face's appearance varies under different illumination and light angles. Subsequently, a user might not be recognised. Furthermore, the variance between images from the same user with varying light conditions is higher than those of two users with the same lighting [8].

Considering these problems, researchers took different spectral bands into account. Therefore, *spectral imaging* was used as a technique where imaging devices capture the light over various spectral bands where only parts are visible to the human eye [34]. Here, a *spectrum* corresponds to the frequency distribution of the wavelengths [38]. Figure 2.11 shows the electromagnetic spectrum, where the ultraviolet (UV) spectrum leads from about 250 to 400 nm, followed by the visible spectrum, reaching from 400 to 760 nm. Next comes the infrared spectrum, divided into near- (NIR) and far-infrared (FIR). NIR leads from 760 to 3000 nm, followed by the FIR from 3000 up to 10^6 nm.

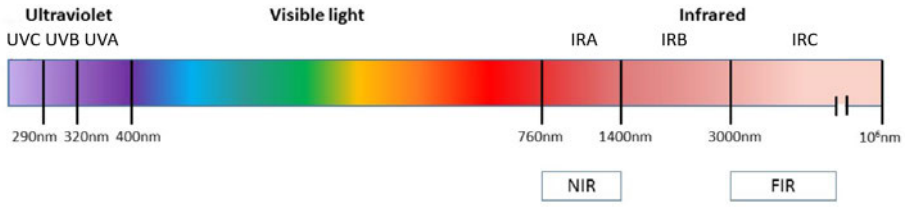

Figure 2.11: The electromagnetic spectrum [39]

The objects reflecting the light form the basis for this imaging technique. Here, light is considered as electromagnetic energy. A light source like the sun or a lamp lightens a surface which then, depending on its characteristics, transmits, absorbs, reflects or scatters the light [34]. Figure 2.12 shows the surface interactions with incoming light.

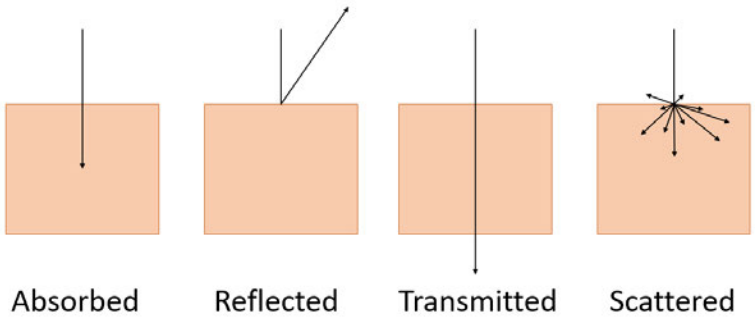

Figure 2.12: Surface interactions with incoming light

For identification in security applications, the focus was mainly on infrared imagery as this approach also works at night. When it comes to *infrared-based face recognition*, special hardware is required. Active NIR lights are installed to achieve frontal illumination. Additionally, a NIR camera captures wavelengths in the NIR spectrum [40]. This hardware setup works similarly to a camera with a flashlight, but the flash is not visible to the human eye in NIR. Especially with wavelengths of 940 nm, the light is entirely invisible [15].

There are several demands on the NIR imaging system. First, the flashlight should be non-intrusive to the human eye. Second, it has to be a fixed frontal illumination. Lastly, if the camera picks up the NIR signals, the camera should only pick up these signals and ignore those from the surrounding area [15].

NIR face recognition can be carried out two- or three-dimensionally. In terms of two-dimensional recognition, it works similarly to the *visible-spectrum-based face recognition*, but an infrared picture of the face is taken instead. This approach is harder to bypass than *visible-spectrum-based face recognition* as these cameras generate a picture with thermal energy or heat [36]. Any object warmer than zero Kelvin emits heat energy. This energy only becomes visible to the human eye when the object has a temperature of several hundred degrees Celsius, for example when metal is heated so far that it glows. Infrared imagery captures some of the heat energy invisible to the human eye and converts the invisible heat radiation into digital images. This procedure is called *thermography* [41]. With two-dimensional NIR face recognition, a regular printed or displayed picture can sometimes be invisible. Therefore, it provides higher security than two-dimensional face recognition in the visible spectrum. This technology is, for example, used in higher-end devices with "Windows Hello" [36].

The three-dimensional *infrared-based face recognition* goes even further. Besides a NIR camera, a flood illuminator and a dot projector are used to acquire the face's depth data. Therefore, the dot projector projects a point cloud on the user's face. An integrated sensor then creates a depth map of the face by measuring the layout of the projected dots. In addition, a two-dimensional NIR image of the user is recorded with the NIR camera. Another advantage is that it is even harder to bypass as the user's depth information is required. Apple's Face ID and the Google Pixel 4 use this technology [36].

With near-infrared (NIR), the illumination problem can be bypassed [8]. The design relies on the *Lambertian law* to achieve an illumination invariant face recognition. According to this law, the radiant intensity decreases as the beam angle becomes flatter. When a surface follows this law and maintains constant radiance, it results in a circular distribution of radiant intensity [42]. Regarding a point light source, the formation of an image $I(x, y)$ can be formulated as follows [40]:

$$I(x, y) = p(x, y)\vec{n}(x, y)\vec{s} \tag{2.5}$$

Here, $p(x, y)$ symbolises the albedo of the face at the point (x, y) and subsequently maps the photometric characteristics of the skin and hair. The surface normal $\vec{n} = (n_x, n_y, n_z)$ is a unit row vector in the 3D space corresponding to the face's geometric shape. The vector $\vec{s} = (s_x, s_y, s_z)$ describes the lightning direction and is a column vector. With a frontal lightning, the formula can be described as follows [40]:

$$I(x, y) = \kappa p(x, y)\vec{n_z}(x, y) \tag{2.6}$$

Here, $\vec{n_z}(x, y)$ provides the depth information acquired by the system. κ can be "monotonic to the distance between the face and the light" [40]. For illumination invariant face recognition, local binary patterns (LBP) are used [43]. *LBP* is a texture operator provided by Ojala et al. [44]. It regards the eight neighbour pixels around a centre value and creates a binary number. If the grey value of each neighbour pixel exceeds the grey value of the centre pixel, the operator assigns a zero; otherwise, a one. Subsequently, a 3x3 subwindow with a grey value in the centre and the thresholding of eight neighbour pixels is regarded [45].
To form the feature vector, the thresholding bits are concatenated anti-clockwise. The generated LBP code does not change due to monotonic transformations of the image, creating illumination invariant features [40]. Figure 2.13 describes this method. Here, the LBP string would be (0001111). Therefore, the LBP code for this 3x3 window is $0 + 0 + 0 + 8 + 16 + 32 + 64 + 128 = 248$ [40].

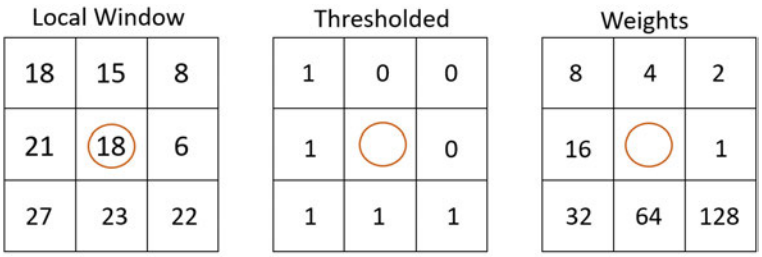

Figure 2.13: 3x3 Local Binary Pattern

When it comes to human skin in the visible spectrum, reflectance increases with the wavelength. However, in other spectral bands, it provides unique spectral imaging due to its components. Figure 2.14 shows the spectra of different skin types captured with an analytical spectral devices (ASD) field spectroradiometer. The x-axis plots the wavelength from 450 to 1800 nm. This range covers the visible spectrum and parts of the infrared spectrum separated by the orange line. The y-axis plots the skin's reflectance from 0 to 0.8 per cent. Here, Type I refers to less pigmented skin going up to Type VI being stronger pigmented skin [46].

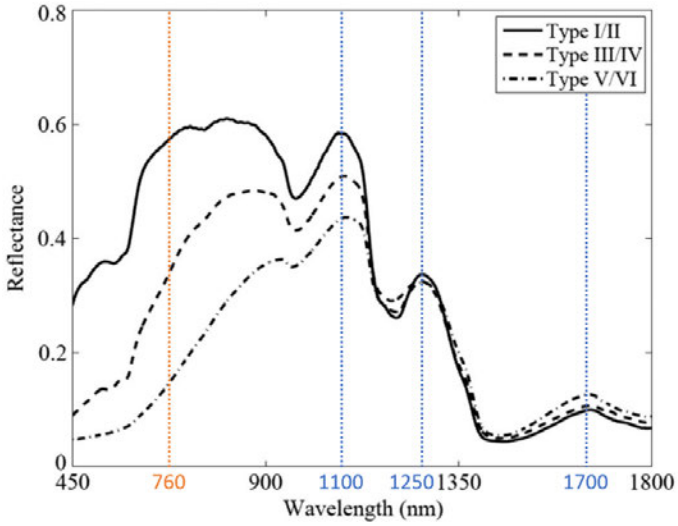

Figure 2.14: Spectra of the different skin types in the visible and NIR band (after orange line) [46]

Looking at the curve in Figure 2.14, we can see that all skin types have a similar course, but they differ in the strength of the reflection. Regarding the visible spectrum, the skin's reflectance increases with the wavelength, as mentioned. Type I and II achieve the highest reflectance, from about 0.3 % to 0.58 %. Type III and IV ranging from 0.1 % to 0.35 %, and the types V and VI provide the lowest reflectance with a reflectance rate from 0.05 % to 0.15 %.
Regarding the NIR spectrum, up to about 900 nm, the skin's reflectance also increases with the wavelength up to 0.6 % (Type I and II), 0.48 % (Type III and IV) and 0.35 % (Type V and VI). With wavelengths longer than 900 nm, the reflectance intensity lowers. Furthermore, three significant peaks are visible (blue lines), and the curves of the skin types nearly converge. The first peak is at about 1100 nm with 0.6 % for Type I/II, 0.51 % for Type III/IV and 0.43 % for Type V/VI. The second peak follows at around 1250 nm with a reflectance of 0.35 % for Type I/II and III/IV and 0.34 % for Type V/VI. The last peak is at 1700 nm, with a reflectance of 0.09 % (Type I and II), 0.1 % (Type III and IV) and 0.12 % (Type V and VI). Between the second and third peaks, it can be seen that all skin types have the same reflectance properties of 0.28 % at a wavelength of around 1300 nm [46].

Human skin provides a heterogenous structure with distinct scattering and absorption, resulting in a unique reflectance [47]. There are differences over the spectral bands due to components acting as absorbers. The skin's appearance primarily depends on proteins and amino acids regarding the UV spectral band. Figure 2.15 shows the absorption of the skin components oxy hemaglobin, water and melanin in the visible and NIR spectrum separated by the orange line. Additionally, the dermal scattering is displayed. The diagram plots wavelength in nm (x-axis) against absorption and scattering in cm^{-1} (y-axis). Here, the x-axis also leads from 450 to 1800 nm and the y-axis from 10^{-5} to $10^{4}\,cm^{-1}$.
The area until the blue line describes the skin's appearance in the visible and NIR spectrum up to 1100 nm, depending on blood and melanin. Water is the distinctive factor in wavelengths longer than 900 nm [34]. Here, the water's absorbance increases with the wavelength. Therefore, it dominates the scattering effect of the skin, whose reflectance decreases [46].

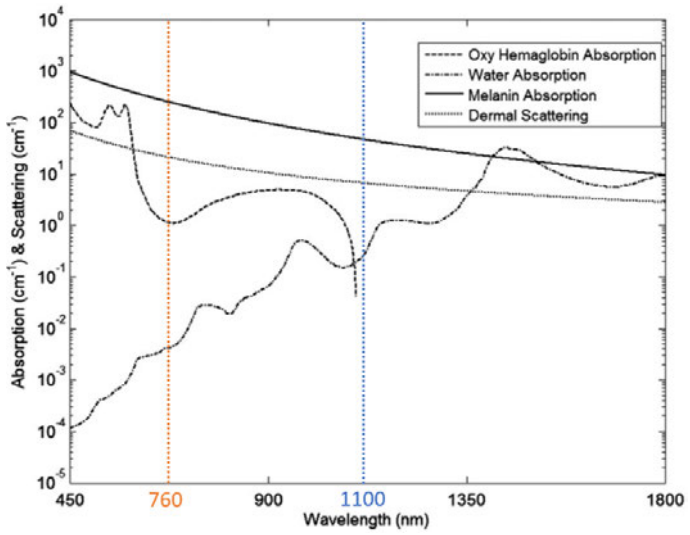

Figure 2.15: Absorption of the skin components oxy hemaglobin, water and melanin in the visible and NIR band (after orange line) [46]

Another advantage of using spectral bands other than the visible spectrum is that materials can be classified or identified by their reflective properties via the electromagnetic spectrum. Subsequently, *spectral imaging* can increase the performance of face recognition systems as it offers new possibilities. Its composition varies between individuals but also across the skin of an individual. Therefore, it can be easily distinguished between human skin and other materials, making it feasible to segment a face from the background [34]. Figure 2.16 displays the spectra of human skin, a doll and cardboard in visual and NIR bands. The x-axis represents the wavelength from 450 to 1800 nm. The reflectance is displayed on the y-axis, from 0 to 0.8 %.

Regarding the visible spectrum, it can be seen that the reflectivity of the doll and that of the skin Type I/II are similar. Both start with a reflectance of around 0.22 % and increase to around 0.5 %. Additionally, the reflectance of cardboard and that of skin Type III/IV are nearly the same. Both start with a reflectance of 0.1 % and increase to 0.3 %. Therefore, the system might not distinguish between the skin and those non-human materials.
In contrast, the reflectivity of human skin and other materials differ in NIR. The skin Types I/II and III/IV show the characteristic course with three peaks described above. Regarding the cardboard, its reflectivity rises to 0.78 % up to a wavelength of about 1300 nm, then decreases and settles at a reflectivity between 0.5 % and 0.58 %. Subsequently, cardboard provides a higher reflectivity in NIR than in human skin.
When it comes to the doll, its course resembles that of human skin. However, until around 1250 nm, its reflectivity is lower than human skin's. With wavelengths from 1250 nm to 1600 nm, the doll's reflectance is higher than that of human skin. After that, it is again lower.
These differences in reflectivity in NIR are due to the object's compositions. Both cardboard and the doll contain nearly no water, which is the distinctive factor for the skin's appearance in NIR [46]. Therefore, a distinction between human skin and other materials is possible in NIR.

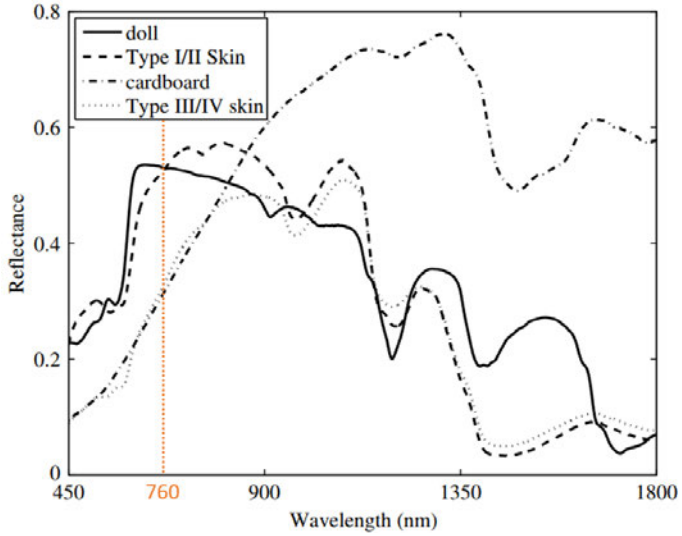

Figure 2.16: Spectra of human skin, a doll and cardboard in the visible and NIR band (after orange line)

Because of the usage of different spectral bands, a differentiation between *same-spectral band face matching* and *cross-spectral band face matching* was made. With *same-spectral band face matching*, the system compares the stored and the captured image in the same spectral band; for example, it matches an infrared image against an infrared image. In contrast, *cross-spectral band face matching* compares images captured in different spectral bands; for example, it matches an infrared image against an image from the visible spectrum [34].

2.3 Attack Types

As mentioned, the number of attack vectors also increased with face recognition technologies. Different methods to forge biometric features have been developed, and attackers use them to bypass the sensor of a biometric system. To ensure that biometric systems still secure sensitive data, the detection of these attacks is required. Subsequently, standardisations for the recognition of attacks on biometric systems are required. The standards are provided by the ISO/IEC, meaning the *International Organization for Standardization* and *International Electrotechnical Commission* being an organisation for worldwide standardisation. These include uniform terminology and measurements [48].
The *ISO/IEC 30107 Information technology - Biometric presentation attack detection* [49] was already mentioned in the previous work [21] and was revised in 2023. Here, *Presentation Attack Detection (PAD)* is the recognition of attacks on the biometric capture subsystem, respectively the sensor when a biometric trait is presented [48]. Next, some terms are defined according to this standard as they are crucial for later evaluation.

First, the proper presentation of a human trait to the sensor of the biometric system is defined as *Bona Fide Presentation*. In contrast, a *Presentation Attack* describes the presentation of an artefact or human trait to the system's sensor to gain unauthorised access. These attacks can be carried out as a single attempt or a multi-attempt transaction [49].

Furthermore, the attacker can act as an *impostor* or a *concealer*. As an *impostor*, the attacker embodies the biometric trait of a person known by the biometric system. The goal is to be recognised as an authorised user and, therefore, to get access to the secured data. Here, the attacker forges the biometric trait of a user. In contrast, as a *concealer*, he aims to hide his identity so the system would not recognise him. Here, the attacker presents a forged biometric trait or modifies his biometric trait accordingly [48].

A *Presentation Attack Instrument (PAI)* is needed to carry out the attacks. A *PAI* is an artefact that represents a biometric trait of a user to gain unauthorised access [49]. *Artefacts* can be categorized into artificial, human-based or other naturals [48].
Artificial artefacts are non-human and can be complete or partial, e.g. a whole head is modelled, or only the face is used [48]. When it comes to human-based artefacts, these can be "lifeless (e.g. cadaver), altered (e.g. mutilation), non-conformant (e.g. only fingertip), conformant (e.g. zero effort impostor attempt) and coerced (e.g. under coercion)" [21, p.13].

Regarding face recognition, there are several types of artificial artefacts. Those can be divided into two-dimensional, three-dimensional and other artefacts. In the second step, these are further divided into static and dynamic. Two-dimensional static artefacts are photos. Here, a high quality is required. Dynamic two-dimensional artefacts are videos emulating the face's shape, texture, and movements like blinking. These attacks provide a greater chance of success [50].
A static three-dimensional attack involves face masks. This approach is more difficult to implement as it requires more skills and facial data. There are multiple variants of this approach. Printing a photo on fabric to simulate deformations is the simplest way. Next, a three-dimensional model can be created from at least two photos. The variant involving the highest effort is to create a real three-dimensional model. However, this variant needs special equipment. The three-dimensional dynamic approach works with a virtual head [50].
Some attacks are, for example, executed with deep fakes. Other approaches involve make-up or surgery. However, these approaches also require a higher skill level [50].

Besides a *Presentation Attack*, a *Digital Attack* can be carried out. Here, the attacker tries infiltrating the system to modify or override the saved biometric data in the Database [50]. Due to those different attacks, multiple *Attack Types* can be defined according to the characteristics of the attack [49]. These are displayed in Figure 2.17. It shows the attack points and distinguishes between Presentation and Digital Attacks.
Type 1 is a presentation attack against the Biometric Sensor. This attack involves forging a user's biometric feature and presenting it to the Biometric Sensor. The aim is to pass oneself off as an authorised user [13].
Type 2 is a replay attack. Here, an attacker attacks the communication channel between the Biometric Sensor and the Feature Extractor. Through this channel, the attacker can intercept and store the biometric data sent between both modules. Next, the attacker can forward faked data from an authorised user to the Feature Extractor to bypass the Biometric Sensor. Thus, a man-in-the-middle attack is carried out [51].
With **Type 3**, the Feature Extractor is attacked, where the attacker installs a malicious one. Therefore, he controls the malicious Feature Extractor and can command him to send the desired feature sets to the Matcher [13].

Attack **Type 4**, like Type 2, attacks a communication channel. Here, the channel between the Feature Extractor and the Matcher is attacked. The attacker hijacks extracted features and forwards them to the Matcher to impersonate an authorised user [13].
Type 5 describes the attack on the Matcher, similar to Type 3. Here, the Matcher is replaced by a malicious one. Subsequently, the attacker commands the malicious Matcher to generate high matching scores to pose as an accepted application user and gain access [51].
Type 6 is a Database attack. Thus, the attacker gains access to it, and he can perform changes, insertions and removals [13].
Type 7 is again an attack on a communication channel, but between the Database and the Matcher. It allows the attacker to intercept and store the templates the Database sends. Next, he forwards the templates to the Matcher. Another possibility is to modify the intercepted data and then pass it on [51].
Type 8 targets a communication channel connecting the Matcher and the Application Device. Here, the attacker can tap the information sent and pass it on to it instead of the Matcher or modify the data [13].
The last attack is **Type 9**, which attacks the Application Device. Hence, it is attacked instead of the biometric system. Here, the attacker can exploit bugs to modify the program's control flow [13].

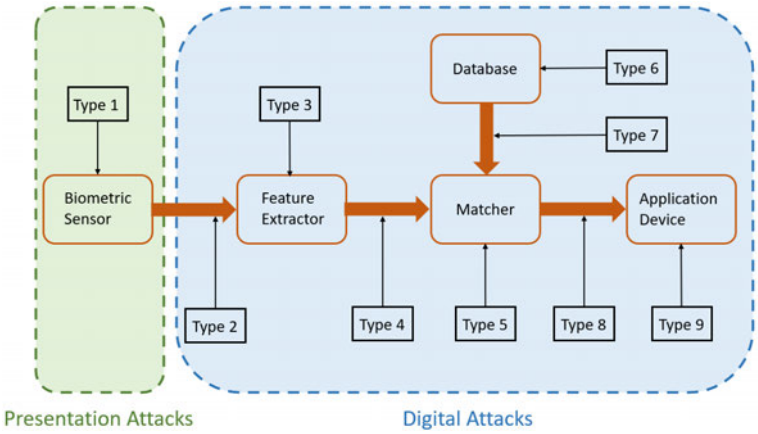

Figure 2.17: Attack Types on biometric systems

Types 2, 4, 7 and 8 are unsuitable for forensic applications as the user must interact with the system so the attacker can make man-in-the-middle attacks and hijack the captured data. However, this is not possible in a forensic environment because, on the one hand, the transmission of biometric data is encrypted, and an attacker would first have to encrypt the data to be entered with the correct key. On the other hand, the user would have to interact with the system before an attacker could intercept and replace the data.
Types 3, 5, and 6 are also unsuitable for forensic applications. The examined devices with biometric systems are often smartphones where the modules of the biometric system are integrated on a system-on-a-chip (SoC), an integrated circuit combining modules of a computer system on a single chip [52]. Subsequently, replacing one module is not possible.
Regarding the Database, in smartphones, the biometric data is stored in a secured area, the Trusted Execution Environment (TEE), which provides its own firmware and is responsible for safety-related functions [53]. Subsequently, attackers cannot change, insert or remove templates stored in the Database.

Regarding Type 9, the smartphone itself would be attacked for forensic applications. It could be a possible attack if exploits for the device are known. Another option would be to perform a brute force attack, as described in the introduction. However, as already mentioned, this is time-consuming.
Type 1 is suitable for forensic applications. If the user is known to the police, his biometric characteristics were recorded during the identification service treatment. Photographs of the user taken during the treatment are stored in the police information system INPOL. It is an electronic data network between the federal government and the states managed by the German Federal Criminal Police Office (BKA). All federal and state police agencies can enter and retrieve data here [54]. With those captured biometric traits, it is possible to create the artefacts for this attack and, therefore, impersonate an authorised user. As we concentrate on Presentation Attacks, only Type 1 is of interest.

PAD can be implemented in the data capture subsystem or through system-level monitoring. The capture subsystem performs it through anti-spoofing techniques like liveness detection. In contrast, a PAD with system-level monitoring relies on indicators like the number of false attempts [48]. Regarding face recognition, different anti-spoofing techniques can be implemented. A few are introduced in the following.

A standard method is the liveness detection. Here, the system searches for small movements like eye-blinking or mouth movements. This approach assumes that such living evidence must be visible when recording. For example, the user is supposed to blink every two to four seconds. However, this assumption can only sometimes be confirmed and only for some users. Liveness detection can prevent two-dimensional static artefacts, not mask or video attacks [55].
Three-dimensional information is also used for anti-spoofing. Here, the system captures two or more images from the user to achieve depth data. This approach can prevent two-dimensional artefacts. Nevertheless, the user has to interact with the system to present different head positions. Additionally, this approach fails regarding three-dimensional artefacts [55].
Another spoofing technique involves the visual appearance of the face. The assumption is that the face with its skin has distinctive properties regarding the light's reflection, absorption or scattering. Materials used to create artefacts provide different properties. Those visual differences are not always visible to the human eye, but with unique image processing algorithms, they can be detected. This technique senses all kinds of artefacts [55].

2.4 State of the Art

Over the last few years, facial recognition has become a popular method to unlock mobile devices or perform access control. However, due to the increased social media presence, obtaining material for faking faces and gaining unauthorised access is getting easier. In addition, many manufacturers have opted for low-cost technology, as additional sensors for improved security entail high costs. Thus, various approaches have emerged in recent years, forging faces with varying degrees of effort to unlock devices with different levels of technology.

Approaches with lower effort aim to unlock devices with a two-dimensional artefact, in most cases with a printed photo. The Dutch consumer organisation *Consumentenbond* published such an approach in 2019 [56]. They tried to unlock smartphones from various manufacturers, for example, Apple, Xiaomi, Samsung, LG or Huawei. For this purpose, they used a printed high-resolution photo of the user. This photo could unlock 26 out of 60 models [56].

One year later, *COMPUTER BILD* provided a similar approach [57]. Researchers took a photo of the user with an Apple iPhone 11, scaled it to A4 size and printed it with a colour printer. In addition, they cut around the edges of the face. This artefact opened 20 of 25 devices successfully. These again included devices from Samsung, Huawei or Xiaomi. Smartphones such as the Samsung Galaxy S9 or the Samsung Galaxy Note 9, which remained locked with the *Consumentenbond* approach, were also successfully opened with this advanced attempt [57]. Both approaches could not fool Apple iPhones or the Google Pixel 4, which utilise infrared sensors for three-dimensional face recognition.

Advanced approaches aim to unlock high-cost devices like Apple's iPhones. The first try to fool an Apple iPhone was provided by the Vietnamese cybersecurity firm *Bkav* in 2017 [58]. They developed a mask consisting of different parts to mimic a face. To begin with, *Bkav* made a face scan to print a true-to-scale shell with a 3D printer. They extended the mask with a silicone nose made by an artist and photos of the mouth and eye area [58]. They published a video showing their attempt to unlock the Apple iPhone X[1]. As can be seen, the device could be unlocked. However, this approach could not be repeated so far. Different devices were not under test [58].

In 2018, the British magazine *Forbes* presented an attempt to fool different smartphones with a three-dimensional printed head [59]. For this purpose, a setup with 50 cameras, all shooting simultaneously, was used. The photos were processed photogrammetrically and merged into a 3D model, which was then loaded into editing software to erase errors. They printed the model with a 3D printer using gypsum powder. After printing, the researchers enhanced the model by colouring it. With this approach, the researchers unlocked four devices, where some required good lighting and a certain angle. However, the Apple iPhone X remained locked [59].

One year later, security researchers provided another approach to the Black Hat hacker convention in Las Vegas [60]. They claimed to break into the Apple iPhone X in 120 seconds. FaceID only uses parts of the 3D data in the eye region if the user wears glasses. Their approach required glasses, white and black tape and a sleeping person. The researchers covered the lenses of the glasses with white tape. To mimic the pupil, they taped some black tape in the middle of the lenses and created a hole so a part of the white tape was still visible to resemble the pupil's light reflection. Now, making a sleeping person wear these glasses, the Apple iPhone X unlocked [60]. Other devices were not under test.

This thesis adapts some of the approaches mentioned above. To bypass devices where only the front camera performs face authentication, we also capture the face with a smartphone and present a printed picture with cut edges. Here, we go one step further and test the limitations of the sensors according to image resolution, colouring, presented angle and required parts of the face. Additionally, it is examined if wearing glasses affects the results.
We also create a three-dimensional face mask to bypass devices with infrared technology. In contrast to the abovementioned approaches, we use different capturing techniques and mask materials. In addition, besides the Apple iPhone, we try to bypass the Google Pixel4 and Windows Hello devices. We go one step further and look at the face recognition software to get an insight into the way of working and being able to make conclusions for the mask creation.

[1] https://www.youtube.com/watch?v=i4YQRLQVixM, visited on 04/14/2023

3 Experiments and Results

This chapter thematises the examination of this thesis, which intends to bypass multiple devices' face recognition, which implements different technologies with varying security standards. Different approaches were presented, starting with those involving minimal effort by creating two-dimensional artefacts to bypass devices carrying out their face recognition in the visible spectrum. In addition, the sensors' limitations regarding their demands concerning the image's colouring and resolution, the face's positioning and the required facial features were examined.
Then, approaches involving more effort were carried out by creating advanced two-dimensional and three-dimensional artefacts to unlock devices carrying out their face recognition with multiple sensors recording the face's depth geometry and appearance in the NIR spectrum. Here, different scanning techniques were evaluated to create face masks, which were further enhanced to adapt their appearance to a human face.

Section 3.1 explains the devices under test (DUT) and their face recognition technologies. Furthermore, the additionally used hardware and software are listed. Next, Section 3.2 examines the attacks targeting devices without additional hardware for face recognition. Moreover, the sensor's limitations are tested. Section 3.3 describes the more elaborate attacks on the devices that use additional facial recognition sensors. Here, the devices' requirements and the underlying service and firmware for face recognition are explained. In addition, the creation and evaluation of the two- and three-dimensional artefacts is outlined.

3.1 Experimental Setup

Various DUTs, materials, and additional hardware and software were required to execute the attacks. Section 3.1.1 provides an overview of the nine devices under attack and their technologies. The following Section 3.1.2 outlines the materials applied to create the two- and three-dimensional masks. Furthermore, in Section 3.1.3, the additional hardware is regarded, and its techniques are explained in more detail. Lastly, Section 3.1.4 offers an overview of the utilised software for data processing, mask evaluation and firmware analysis.

3.1.1 Devices under Attack

We chose different devices to test the generated artefacts, most of which are smartphones implementing the latest technologies. We used seven smartphones for this experiment, four of which only use their RGB camera and have no additional face recognition sensors. As mentioned in Section 2.4, most smartphones include no additional sensors to execute face recognition due to the higher costs of additional hardware. The other three smartphones contain one or more infrared sensors. Additionally, a laptop and a tablet served as DUT, implementing infrared sensors but working with a different face recognition technique. The infrared sensors provide higher security. Subsequently, more effort is required for successful spoofing attacks. Table 3.1 shows the attacked devices.

Table 3.1: Devices under attack with and without additional sensors for face recognition

Devices with no additional sensors	Devices with infrared sensors
Samsung Galaxy S9	Google Pixel 4
Huawei P30	Apple iPhone Xs Max
Xiaomi Redmi Note 7	Apple iPhone 12
Samsung Galaxy S20	Dell XPS 15 laptop
	Microsoft Surface Pro 7

Four smartphones include no additional sensors to perform face recognition; they all rely on the same technique, as only the front camera is used to capture the face. Figure 3.1 shows an example of the modules required for facial recognition on the Xiaomi Redmi Note 7. This device is described in more detail below. Initially, the devices record the user's frontal face with their RGB camera and extract the angles and distances between the facial features, which are then saved as reference data. When the user tries to unlock the smartphone, its RGB camera again captures the user's face, after which an algorithm extracts the same information and compares it with the stored data. Here, no additional information is acquired [36].

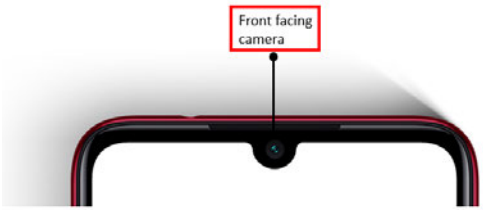

Figure 3.1: Modules involved in face recognition of devices without additional sensors [2]

Samsung Galaxy S9

The first device under attack was the Samsung Galaxy S9, working with Android version 10, released in 2018. It has a front camera resolution of eight megapixels. When enrolling a user, he first has to tick if he wears glasses, and then, he has to align his face within a rectangle. The device scans the face and displays the progress as a percentage. When it reaches 100 %, the enrolment is done.

Huawei P30

Next, the Huawei P30, working with Android Version 10, released in 2019, was attacked, which provides a front camera resolution of 32 megapixels. Here, the enrolment requires good lighting conditions like daylight or a well lit room. Additionally, the user has to align his face to a rectangle frame. When the device finds a face, a circle appears, fills, and when it is completed, the enrolment is done.

Xiaomi Redmi Note 7

The Xiaomi Redmi Note 7, working with Android version 9, was released in 2019 with a front camera resolution of 13 megapixels. As with the Samsung Galaxy S9, the user must ensure that his face is adequately illuminated for the enrolment. He has to align his face to the given oval shape on the screen, and when the face is approximately aligned, the enrolment is done. Different head positions are not required.

[2] https://www.mi.com/de/redmi-note-7/, visited on 09/12/2023

Samsung Galaxy S20

We also attacked the Samsung Galaxy S20 working with Android version 11, which was released in 2020 and provides a 10-megapixel front camera. To enrol a user, he first has to tick if he wears glasses, and then, he has to align his face in a rectangle. The device scans the face and displays the progress as a percentage. When 100% is reached, the enrolment is done.

The remaining five devices include additional hardware to perform face recognition. Therefore, the authentication process is more complex than that of the previously described devices. It is outlined in the following for each of the devices.

Google Pixel 4

Furthermore, the Google Pixel 4 with Android version 13 was attacked. It launched in 2019 and uses two infrared cameras, working with a resolution of 640x480 pixels, and a dot projector for face recognition to create a three-dimensional map of the face [61]. Additionally, a flood illuminator is used. The involved modules are displayed in Figure 3.2 and outlined in red.

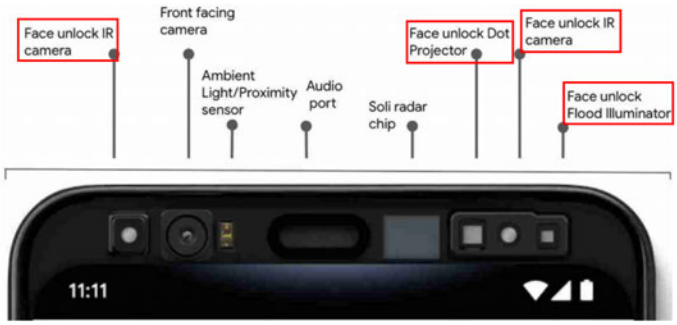

Figure 3.2: Modules involved in face recognition of the Google Pixel 4 [62]

With a spectrometer, we investigated that those infrared cameras work with wavelengths around 940 nm, subsequently NIR. Since two infrared cameras create the three-dimensional map, the process called *stereoscopy* is used. *Stereoscopy* is a technique to reproduce the spatial image impression [63].

Google also patented its array-based patterned illumination projector. Figure 3.3 shows the system description with a light emitter and a light detector as listed in the patent, and Figure 3.4 displays a flowchart from the patent specification where the emitter creates both light patterns [64].

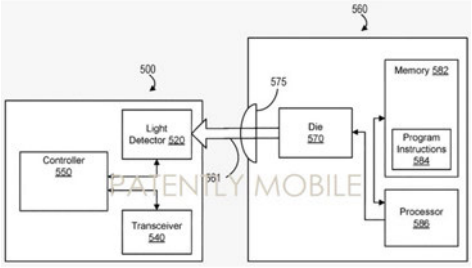

Figure 3.3: Google's light emitter and light detector [64]

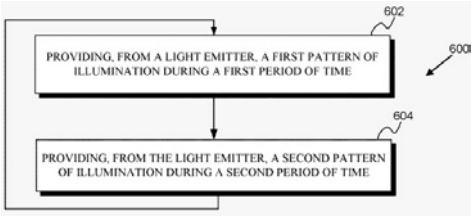

Figure 3.4: Google's process for creating light patterns [64]

To register the user, he must align his face in a predefined oval. The device executes only one scan where a sphere with individual faces appears and the user must move his head until all areas are filled to capture different face angles.
The Google Pixel 4 emits its point cloud when the lock screen is displayed. Accordingly, the presence of a face is not essential for the radiation of the point cloud. Additionally, it uses a point cloud, which is emitted twice and is made of a consistent point grid layering over the face and covering a relatively large area. Figure 3.5 shows the emitted point pattern.

Figure 3.5: Dot pattern emitted by the Google Pixel 4

Apple iPhone Xs Max and Apple iPhone 12
The Apple iPhone X, released in 2018, was the first smartphone with infrared-based face recognition. One year later, the Apple iPhone Xs Max launched, being the device under attack working with iOS 12.4.1. Both provide Apple's TrueDepth System with a seven-megapixel camera, creating a point cloud for user authentication [65]. Furthermore, we chose the Apple iPhone 12 as a device under attack, working with iOS version 14.2, released in 2020. Compared to the Apple iPhone X, it includes an improved TrueDepth Camera, providing a resolution of 12 megapixels. However, the enrolment and the general structure remain the same. Figure 3.6 displays the involved modules for face recognition outlined in red.

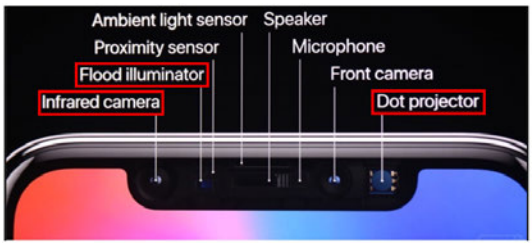

Figure 3.6: Modules involved in face recognition of the iPhone 12 [66]

To unlock a phone with Face ID, the owner has to direct his opened eyes to the device. The TrueDepth Camera then captures the geometry of the face by reading over 30,000 infrared dots from the captured depth map. In addition, it records a two-dimensional infrared image. A neural network is used for matching and provides anti-spoofing. Apple's Neural Engine (ANE) performs face authentication and anti-spoofing, implemented on the Secure Enclave Processor [67, 68]. With this technology, Apple claims that the probability that someone else unlocks the phone is 1 in 1,000,000 [65].

Apple's anti-spoofing technology is patented and is supposed to protect the device from replay or spoofing attacks. Subsequently, unlocking the phone with previously captured valid data or a mask should not be possible. Additionally, the point pattern varies as the illuminator arrays are enabled or disabled dynamically (see Figure 3.7) [69].
To make face authentication even more secure, Apple works with multiple validations during one recognition session displayed as a flow chart in Figure 3.8 [69]. The patent describes that the first step is to detect the face, after which two validations are carried out. First, it validates if the stored face data and the recorded face match, at which point, in the second step, the pattern is validated. After these validations, the face and the probing pattern are matched. The last step is to decide whether the user is accepted or not.

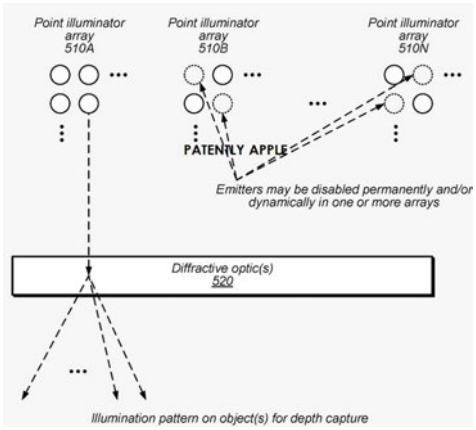

Figure 3.7: Apple's point illuminator arrays [69]

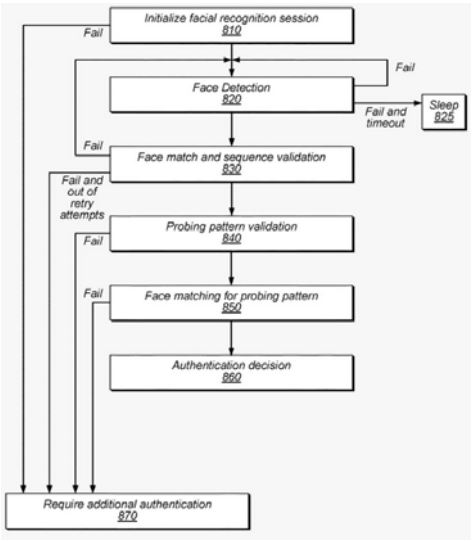

Figure 3.8: Apple's face recognition session [69]

During enrolment, a user must align his face in a frame until the device recognises a face. Next, a circle appears, and the user has to rotate his head until the circle is completed so the device can capture the head from all possible angles. In contrast to the Google Pixel 4, a second more detailed scan is carried out. A second circle appears where the user has to turn his head again until the circle is complete. After both scans are terminated successfully, the enrolment is done.
In contrast to the Google Pixel, the Apple iPhone first emits an infrared flash, and only when a face is detected, it emits the point cloud consisting of two randomised patterns (see Figure 3.9). It can be seen that the first pattern is not as dense as the second one, as the points of the second pattern partly overlap. Furthermore, the point clouds are emitted in a more centred manner and do not cover such a vast area as with the Google Pixel.

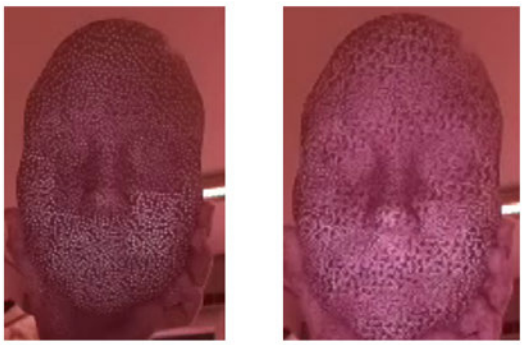

Figure 3.9: Dot patterns emitted by the Apple iPhone 12

Dell XPS 15 and Microsoft Surface Pro 7

Next, a Dell XPS 15 laptop was chosen as DUT, which was released in 2021 with a Windows Hello-compatible webcam integrating an infrared camera working with Windows 10 Pro, was chosen. Additionally, the Microsoft Surface Pro 7 with Windows 11 was tested in the same way. It was released in 2019 and also uses Windows Hello. Besides the regular webcam, this device integrates an infrared camera and a front privacy light for face authentication. These modules are outlined in red and displayed in Figure 3.10.

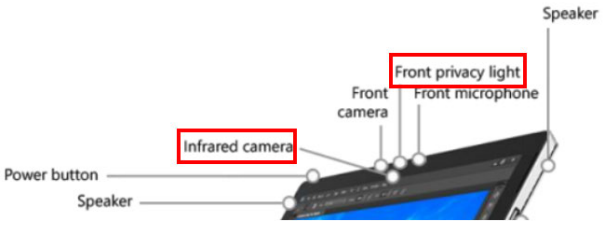

Figure 3.10: Modules involved in face recognition of the Microsoft Surface 7 Pro [3]

Windows Hello uses a NIR camera for user authentication. First, the algorithm searches for distinctive points in the user's face. The second step is to check that the user looks almost head-on at the device. If that is the case, the algorithm extracts multiple points to create a histogram to represent the differences in brightness around certain points. For matching, it compares the captured representation of the user with the saved one [70]. As Section 2.2 mentions, Windows Hello does not capture three-dimensional face information. Here, only a two-dimensional infrared image is taken into account.

Only one scan is carried out for enrolling a user. The algorithm searches the user's face. He has to look at the camera without moving his head around until the scan is complete.

[3] https://content.nexus.support.com/7dff9ef1a58548a693a4b90534e2d2aa/1f008a80d6c511e7bbeddba4889364e.png, visited on 08/23/2023

3.1.2 Materials

We used various materials to create the artefacts (see Table 3.2), which is explained in more detail in Sections 3.2, 3.3.3, 3.3.4 and 3.3.5. First, only paper and printer ink is required to generate the two-dimensional artefacts. Next, multiple materials produced the three-dimensional face casts. The plastic wrap, the plaster bandages and the modelling clay generated the first face cast. Additionally, the masks are three-dimensionally printed with filament and resin.

To enhance the masks, they were coated with liquid latex. In the second step, makeup utensils like lashes, eyeshadow, foundation, lip liner and baby powder gave the mask a more humanoid appearance. Additionally, the eyebrows are modelled with human hair. Another approach was to enhance the mask by applying acrylic paint instead of makeup to highlight the facial features.

Lastly, we used additives to enhance specific properties or to facilitate the handling during the mask generation. Dry shampoo helps brighten the dark areas during the scanning process, so the structured light scanner can capture these areas. We also worked with Vaseline to quickly remove the modelling clay from the plaster mould, and release spray helped to remove the liquid latex from the mould. Additionally, hair wax and hydrogel were applied to enhance the reflectivity of the artefacts and to simulate a higher water content of the materials. The resulting masks are displayed in Sections 3.2, 3.3.3, 3.3.4 and 3.3.5.

Table 3.2: Materials used for the creation of the two- and three-dimensional artefacts

Materials for 2D artefacts	3D face cast materials	3D mask materials	Aids
Paper	Plastic wrap	Liquid latex	Dry shampoo
Printer ink	Plaster bandages	Baby powder	Vaseline
	Modelling clay	Fake lashes	Release spray
	Filament (Grey, Peanut)	Foundation	Hair wax
	Resin (White)	Eyeshadow	Hydrogel
		Lipliner	
		Lash glue	
		Human hair	
		Acrylic paint	
		Contact lenses	

3.1.3 Additional Devices

In addition to the abovementioned devices themselves, the following devices were used to record the user's face two- or three-dimensionally. For the two-dimensional infrared images, the *Raspberry Pi 4 Model B*[4] in combination with the *Raspberry Pi Camera Board - Night Vision IR-CUT 5MP*[5] was used. Additionally, the device's point clouds were recorded with this camera to evaluate the face masks. The *Raspberry Pi 4 Model B* is a single-board computer, and the utilised camera is an RGB camera without IR filter providing a 1080p video resolution [71].

[4] https://www.raspberrypi.com/products/raspberry-pi-4-model-b/, visited on 09/27/2023
[5] https://thepihut.com/products/raspberry-pi-night-vision-camera-ir-cut, visited on 09/27/2023

Furthermore, we worked with several scanners offering techniques to obtain a three-dimensional model of the face. First, the *Shining 3D EinScan-SP*[6] was used. It is a fixed, structured light scanner mounted on a tripod. The handheld scanner *Artec EVA 3D*[7], also working with structured light, was additionally used.
The foundation for this technique is triangulation, meaning that by "emitting a beam of light on any single point of an object and taking three measurements of the point (including distance and angle), the exact position of the point space can be determined" [72]. Structured light scanning involves projecting line patterns onto an object with a light projector and capturing these patterns with multiple cameras which is illustrated in Figure 3.11. White light is commonly used here, but some techniques involve blue light as it can reduce reflections [72].
When the structured light hits the object's surface, it appears perfect from the point of view of the light source. However, when viewed from the angle of the cameras, the light pattern seems to be distorted as soon as it hits elevations and depressions. Multiple cameras then record these distorted light patterns. Both *Shining 3D EinScan-SP* and *Artec EVA 3D* work with two cameras. Based on the distortion of the light stripes the software can then reconstruct the three dimensional shape of the object. [72, 73].

Another utilised scanner was the handheld laser scanner *HandySCAN 3D*[8]. As with the structured light scanners, the underlying technology relies on triangulation. When it comes to laser scanning, a laser is directed at the object and its emitted location is recorded by a camera. As the laser hits different points of the object, the emitted laser points appear in different positions for the camera [74]. "By combining the geometrical data related to the position of the laser emitter and the subject and the angle of the camera concerning the laser dot as it appears on the subject, the features and dimensions of the subject can be recreated in a digital model" [74]. Picture 3.12 displays this technology. Instead of using laser points, the *HandySCAN 3D* works with multiple laser crosshairs.

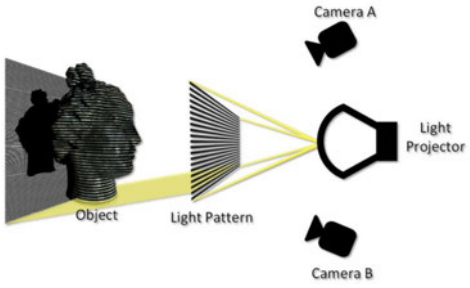

Figure 3.11: Structured light scanning [75]

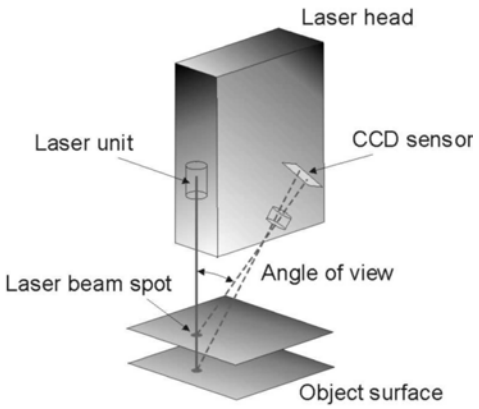

Figure 3.12: Laser triangulation [76]

Besides the scanners, another capturing method used was an array consisting of 13 single-lens reflex (SLR) cameras which recorded the face from above, below, left and right. This setup records the face's texture and creates a three-dimensional model with Structure from Motion (SfM), which generates a three-dimensional model from overlapping images taken with the same camera from different perspectives [77].

[6] https://www.einscan.com/einscan-sp/, visited on 09/27/2023
[7] https://www.artec3d.com/de/portable-3d-scanners/artec-eva, visited on 09/27/2023
[8] https://www.creaform3d.com/de/messtechnik/tragbare-3d-scanner-handyscan-3d, visited on 09/27/2023

First, the scale-invariant feature transform (SIFT) algorithm detects the same features over different pictures. Next, the relative location of the camera and the points are calculated. As a third step, more feature points are detected and his relative location calculated, resulting in a dense point cloud [77]. The principle of SfM is displayed in Figure 3.13.

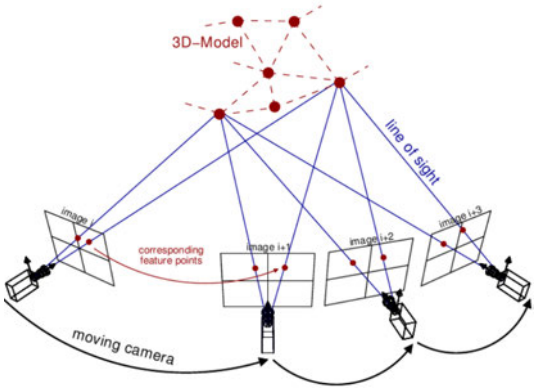

Figure 3.13: Structure from Motion [78]

Both scanning technologies and the SfM needed reference points. These served as orientation points for the algorithms later performing the three-dimensional modelling. For the scanning technologies, a white wall with a rectangle cut out in the middle was used to frame the person's head being scanned. Black dots were glued onto the wall as reference points. Figure 3.14 shows the resulting setup.
For SfM, there was a white wall behind the scanned person. As an orientation, the scanned person has to hold a plate with black and white squares and a red cross in the middle under the chin. Figure 3.15 shows the frontal picture of one of the cameras captured from the array with the orientation plate.

Figure 3.14: Scanning setup with orientation points

Figure 3.15: SfM setup with an orientation plate

Further devices were used to produce the artefacts to target the DUTs. A *HP Laserjet 500 Color M551*[9] printed the two-dimensional artefacts.
The Fused Deposition Modeling (FDM) printer *Original Prusa i3 MK3S+*[10] printed the three-dimensional objects. In this process, the filament is wound on a spool and is pushed through an extruder into the hotend, which extrudes the material from the nozzle. The warm filament is applied in layers to a preheated heatbed where the object is printed bottom-up. The layers cool down during this process, fusing slightly with the next layer [79]. Figure 3.16 shows the structure of the printer.

[9] https://www.hp.com/de-de/printers/laserjet-printers.html, visited on 09/27/2023
[10] https://www.prusa3d.com/de/, visited on 09/29/2023

Another utilised printer was the *Anycubic Photon Mono X 6k* [11] stereolithography (SLA) printer, printing the objects with resin instead of filament to test another printing method that provides a higher resolution. A UV lamp from below illuminates a plate immersed in a resin container in the object's shape. As shown in Figure 2.11, UV radiation includes wavelengths with 250 to 400 nm. Due to the illumination, the UV-active resin hardens on the plate and, therefore, in contrast to FDM printing, the model is printed upside-down [80]. This structure is presented in Figure 3.17. The printed model has to be cleaned with isopropyl alcohol to remove resin residues from the print and must fully cure under a UV lamp to form a stable object [80].

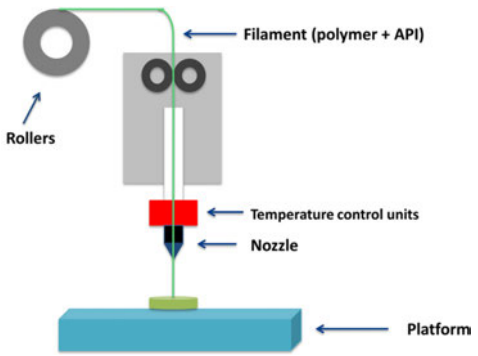

Figure 3.16: Structure of the FDM printer [81]

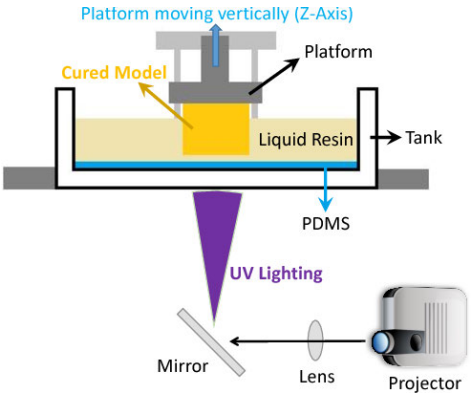

Figure 3.17: Structure of the SLA printer [82]

3.1.4 Additional Software

Different software is needed to execute the experiments described in the following. Several software applications were initially used to record the user's face two- or three-dimensionally, which are listed with their versions and functions in Table 3.3. For two-dimensional recording, *HedgeCam2* was used to capture an NIR image of the mask and evaluate its appearance in NIR by comparing it to the human face. Furthermore, emitting the device's point clouds was captured as a video with *libcamera* and then, *FFmpeg* was used to disassemble the videos into frames.
To acquire the three-dimensional face data, scanner software applications were required. They were used to transform the data captured with the scanners and SLR cameras into a three-dimensional mesh.

[11] https://www.anycubic.com/products/photon-mono-x-6k, visited on 09/29/2023

Table 3.3: Utilised software applications to record the user's face two- and three-dimensional

Software	Version	Description
HedgeCam2		Open-source software application to access the IR camera of the Google Pixel 4 [83]
libcamera		Software library supporting complex camera systems [84]
FFmpeg		Multimedia framework to process audio and video data [85]
EXScan S	3.1.2.0	Calibrates the *Shining 3D EinScan-SP* and aligns the scans to generate a mesh [86]
VXelements	9.0	Only works with Creaform products and creates a mesh from the scanned point cloud of the *HandySCAN 3D* [87]
Artec Studio Professional	17.1.2.15	Only works with Artec products and creates a mesh from the scanned point cloud of the *Artec EVA 3D* [88]
Agisoft Metashape Professional	2.0.2	Performs photogrammetric triangulation [89]

The software used to further process the two- and three-dimensional data is listed with their versions and functions in Table 3.4. To generate the two-dimensional artefacts, the *GNU Image Manipulation Program GIMP* was utilised to scale the images and create the artefacts to test the sensor's limitations. Regarding the three-dimensional artefacts, we worked with *Meshmixer* to clean, cut and smooth the meshes. Additionally, we filled some models and extruded parts of it. In addition, we created the negative mould and solidified the mask with *Fusion 360*. Moreover, the models were prepared for printing with *PrusaSlicer*, which slices the models and generates the G-code which is the content of the control file for the printer. The G-code describes a defined set of instructions that represent motions and actions building the pattern the printer has to follow to create the three-dimensional object [90].

Table 3.4: Utilised software to further process the two- and three-dimensional captured data

Software	Version	Description
GIMP	2.10.34	Open-source image editing software [32]
Meshmixer	3.5.474	Open-source software to edit triangle meshes [91]
Fusion 360	2.0.16976	Cloud-based software platform for modelling and three-dimensional designing [92]
PrusaSlicer	2.5.2	Open-source software to slice models and to generate the G-code for printing [93]

Further software to analyse the firmware and service responsible for face authentication is listed in Table 3.5. To reverse-engineer the software responsible for face authentication, *Ghidra* was used. Reverse engineering describes the process of analysing a program to identify its components and to understand its operation [94]. We also worked with *Frida* to trace the strings found with *Ghidra*. Furthermore, *HxD* displayed the hexadecimal representation of the software files to identify the file type. Additionally, *Android Debug Bridge (adb)* in combination with the tool *logcat* was used. *adb* consists of three components [95].

First, the client running on the examiner's machine sends the commands. Second, a daemon running as a background process runs the commands on the Android device. Lastly, a server manages the communication between the client and the daemon, which runs as a background process on the examiner's machine [95]. *logcat* then printed the log messages linked to the face authentication on the examiner's machine.

Table 3.5: Utilised software to analyse the firmware and service responsible for face authentication

Software	Version	Description
Ghidra	10.3.1	Open-source software reverse engineering tool provided by the NSA's Research Directorate [96]
Frida	16.1.3	Instrumentation toolkit working in the injected, embedded or preloaded mode to, for example, trace functions or instructions [97]
HxD	2.5.0.0	Editor for disk or hex editing [98]
adb		Client-server program which works as a command-line tool to communicate with an Android device and is, for example, used for debugging and installing applications [95]
logcat		Command-line tool used to print out the system logs and that can be started over adb [99]

3.2 Attacking Devices Performing Face Recognition in the Visible Spectrum

First, we targeted devices using their RGB front camera for user authentication and, therefore, selected the Samsung Galaxy S9, the Samsung Galaxy S20, the Huawei P30 and the Xiaomi Redmi Note 7, described in Section 3.1.1. As mentioned in Section 2.2, the facial features are detected, and their distances are measured in the visible spectrum. We created analogue and digital artefacts to bypass the devices and test the sensor's limitations regarding the image colouring, resolution, the head's perspective and the necessary areas of the face.

Initially, we recorded the face with an Apple iPhone 12, offering a front camera resolution of 12 Megapixels. The recording should be full-sized and executed in daylight to achieve an artefact where all facial features are displayed adequately. Furthermore, we took two pictures, with and without glasses, to test if both captures work equally well to create an artefact. When recording the images with glasses, it was attempted to achieve minimal reflection on the glasses, not to cover any eye features. The images in Figure 3.18 show the recorded photos without (left image) and with glasses (right image).

Figure 3.18: Captured images of the user without glasses (left) and with glasses (right)

The pictures' measurements were adjusted to DIN-A4 with *GIMP* while ensuring the ratio of the edges was kept so the distances between the distinct facial features would not be distorted. The scaled pictures were exported as *Portable Network Graphics (PNG)* files and printed with the original image resolution of 300 *Dots per Inch* (dpi) on paper with the *HP Laserjet 500 Color M551* printer. Here, the higher the dpi, the more detailed the image [100]. The background of the printed photos was cut off to achieve better results, as smartphones might detect the non-matching background. The scaled photos were also presented with an *iPad Pro 10.5* to provide digital artefacts and to test if presenting them digitally makes any differences regarding the recognition results. When presenting them to the DUTs, the screen brightness was increased to ensure the picture was well-illuminated, and the iPad was held such that the ceiling light did not reflect in the display. Figure 3.19 shows both printed artefacts with the background cut off (the left and second from the left image) and both digital artefacts (the second from the right and right image).

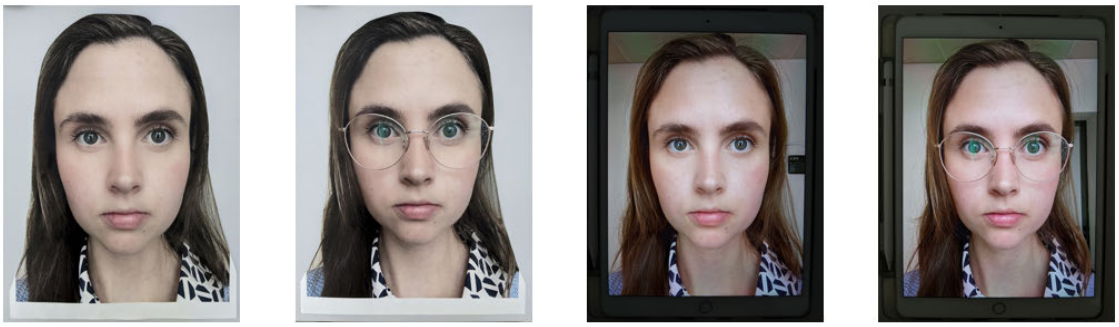

Figure 3.19: Printed artefact without glasses (left), printed artefact with glasses (second from left), digital artefact without glasses (second from right) and digital artefact with glasses (right)

When we presented the artefacts to the devices, we worked with ceiling light and held the artefacts in front of a face to get the right angle and distance between the device and the artefact. Additionally, the artefacts were presented frontally to the device. Table 3.6 shows the results of these presentation attacks.

It can be seen that the DUTs performing face recognition in the visible spectrum could be unlocked with a printed picture of a person's face. Furthermore, it made no difference if the artefact was created with a picture of the user wearing glasses.

Regarding the digital presentation attack instruments, only the Huawei P30 and the Xiaomi Redmi Note 7 could be unlocked, whereas both Samsung Galaxy devices remained locked. One reason for this could be the existing background. Some devices might identify the non-matching background and, subsequently, remain locked.

Table 3.6: Results of the two-dimensional presentation attacks

PAI	Galaxy S9	Galaxy S20	P30	Redmi Note 7
Without glasses	Yes	Yes	Yes	Yes
With glasses	Yes	Yes	Yes	Yes
Digital without glasses	No	No	Yes	Yes
Digital with glasses	No	No	Yes	Yes

However, it needed some attempts to create the right conditions to bypass all devices. As mentioned in Section 2.2, the reliability of face recognition is light-dependent. In a dark environment, even the real face is often not recognised. Therefore, the targeted devices needed special lighting conditions when presenting the artefacts. Direct light is inappropriate as it can overexpose the artefact, but a bright environment is crucial for faster recognition. The best results could be achieved with daylight or indirect lighting where the light source was not focused directly on the artefact.
The angle at which the artefact is presented was crucial as it must be held parallel to the device's screen. Additionally, the distance between the device and the artefact was essential as it should correspond to the distance a user would present his face. All vulnerable devices could be unlocked on the first attempt if all conditions are met.

Summing up, devices with no additional sensors for face recognition have a good chance to be unlocked with printed and some of them even with digital two-dimensional artefacts. Furthermore, it makes no difference if the artefact includes glasses. Printed artefacts provide more reliable results as they unlock all devices under test. Additionally, it is helpful to cut off the background, provide a bright environment with indirect lighting and present the artefacts in a position similar to how a user would present his face to the device.

We then tested the limits of the device's face recognition systems. Only the captured picture without glasses was chosen as all devices could be unlocked equally well with both images. In addition, we presented the artefacts only on paper as they provided better results than the digital artefacts. The experiments included the following:

1. The images were converted to grayscale. The aim was to determine if the sensors take care of the right colouring or if only the distances and angles between the features matter.
2. The image resolution was changed to test whether the devices needed good photo quality or low-resolution images like from surveillance cameras could also work.
3. The head position was adjusted to examine if a front-faced photo was necessary.
4. Only parts of the face were presented. The aim was to determine if a whole face is required for authentication or if only facial features are sufficient.

Colouring

We changed the image colouring with *GIMP* to convert the image colour to grayscale. The left image in Figure 3.20 shows the resulting grayscale image, which was printed with a resolution of 300 dpi with the *HP Laserjet 500 Color M551* printer, and the background was cut off as this provided more reliable results. Figure 3.20 displays the resulting artefact in the right image.

Figure 3.20: Captured image of the user converted to grayscale (left) and the resulting two-dimensional artefact with the edges cut off (right)

It stated that most DUTs working with their RGB camera for face recognition do not require a coloured image, as three of four devices could be unlocked with the printed grayscale image. Only the Samsung Galaxy S9 remained locked. However, this could also be due to unfulfilled conditions. The device can sometimes not be unlocked by the user in poor lighting conditions as it is more difficult to provide the right lighting conditions here.

In general, it seems that for face recognition, the colouring of the artefact is not considered. Therefore, only the facial features and the distances between them are required, and a photo taken without colours would also be suitable to generate a successful artefact.

Resolution

The second step was to investigate the image resolution by which the artefact is still recognised as an authorised user. Therefore, we changed the image resolution with *GIMP*. Only the dpi were adjusted to ensure the image was printed in full size while keeping the original ratio. We decreased the image resolution from 200 dpi in steps from 50 dpi and generated four artefacts with 200, 150, 100 and 50 dpi. Images providing a higher resolution than 300 dpi were not considered. The images were printed again with the *HP Laserjet 500 Color M551* printer, and the background was cut off. Figure 3.21 shows the four resulting artefacts from left to right in descending order of resolution.

Figure 3.21: Printed artefacts with 200 dpi (left), 150 dpi (second from left), 100 dpi (second from right) and 50 dpi (right)

The experimental results are displayed in Table 3.7. All tested devices could be unlocked with artefacts with 200 or 150 dpi. The Samsung Galaxy S9, the Huawei P30 and the Xiaomi Redmi Note 7 are less secure than the Samsung Galaxy S20 as they could also be bypassed with the artefact offering a resolution of 100 dpi. However, the artefact with a resolution of 50 dpi could not bypass any of the devices. A reason for this is that the facial landmarks are not well detectable as, for example, the eye's shape is barely present and therefore, their outline cannot be traced.

Table 3.7: Results of the two-dimensional presentation attacks with artefacts providing a resolution of 50, 100, 150 and 200 dpi

Resolution	Galaxy S9	Galaxy S20	P30	Redmi Note 7
200 dpi	Yes	Yes	Yes	Yes
150 dpi	Yes	Yes	Yes	Yes
100 dpi	Yes	No	Yes	Yes
50 dpi	No	No	No	No

Summing up, a higher-resolution image is unnecessary to bypass face recognition. However, artefacts with a resolution of 200 or 300 dpi need fewer attempts to bypass the device than those with a lower resolution. Furthermore, the resolution required by the devices varies slightly. Most devices can be bypassed with a minimum resolution of 100 dpi corresponding approximately to the resolution of an enlarged passport photo. In addition, capturing the user's face from a distance and scaling the image might be possible. However, since not all devices can be unlocked with this resolution, using face images with a minimum of 150 dpi is advised.

Head's position

We tested different head positions to examine whether a frontal picture is required. Therefore, we recorded additional images with the Apple iPhone 12's front camera, where the user's head was turned until the profile was mainly visible. Accordingly, it was turned about 20 (left image in Figure 3.22) and 45 degrees (second from right image in Figure 3.22), the eyes not directed at the screen. The artefacts were again printed on paper with a *HP Laserjet 500 Color M551* printer, and the background was cut off. The resulting artefacts are shown in Figure 3.22 (second from left and right).

Figure 3.22: Ten degrees angle image (left), ten degrees angle artefact (second from left), twenty degrees angle image (second from right) and twenty degrees angle artefact (right)

Table 3.8 displays the unlocking results. It can be seen that all DUTs could be bypassed with those artefacts. Additionally, the attempts to unlock the device remained the same as if a frontal artefact is presented. Furthermore, no tested device required the user's eyes to be directed to the screen. If a device needs a direct glace, it could be possible that different head positions would not work.

Table 3.8: Results of the presentation attacks with artefacts providing different head positions

Head's position	**Galaxy S9**	**Galaxy S20**	**P30**	**Redmi Note 7**
20 degrees	Yes	Yes	Yes	Yes
45 degrees	Yes	Yes	Yes	Yes

These attacks show that a frontal picture is optional to generate a successful artefact. Therefore, recording a user without active participation and using those images as an artefact might be possible. As the devices do not consider the eye's centre, no direct glance is required as face recognition might only consider the eyes' shape. However, some devices offer the possibility to perform an attention check during face recognition where the user has to look directly on the screen. If this option is enabled, a frontal picture is necessary.

Parts of the face

Lastly, we generated artefacts containing only parts of the face. The aim was to determine if the devices require a whole face for authentication. Therefore, different artefacts were generated with *GIMP*. The required regions were selected using a selection tool, and the remaining areas were removed. The selected regions were then printed on paper with the *HP Laserjet 500 Color M551* and cut out.

We split the face vertically to generate an artefact containing half a face. The aim was to test if only parts of the features were sufficient. Since face recognition only detects the facial features and their distances from each other, an artefact consisting of the area of interest was initially tested. Furthermore, artefacts were generated where one feature was erased to test if all facial features were needed for authentication. We created one without eyebrows and another without a mouth. These artefacts are displayed in Figure 3.23.

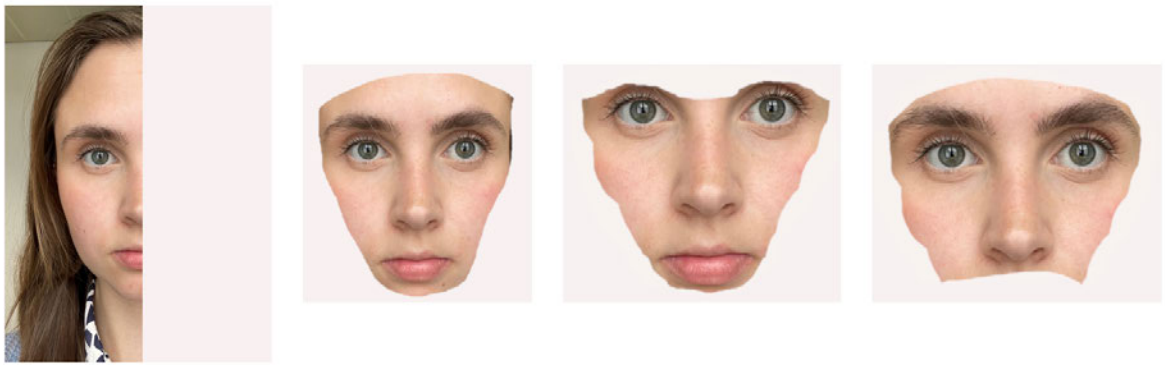

Figure 3.23: Artefacts consisting of a vertical half of the user's face (left), of only the area of interest (second from left), of the area of interest without eyebrows (second from right) and of the area of interest without the mouth (right)

We erased more features to present only distinctive regions and test whether less than three features are sufficient. Only the eye area containing the eyes and the eyebrows was presented. To go one step further, we presented only the eyes. Additionally, one artefact consisted of only the nose and the mouth to test if eyes are required. Figure 3.24 shows the resulting artefacts.

Figure 3.24: Artefacts consisting only of the user's eye area (left), only of the eyes (middle) and only of the nose and mouth (right)

The experimental results are presented in Table 3.9. The Samsung Galaxy S9 has the highest security as it only unlocked with the artefact containing the area of interest (see the second from the left image in Figure 3.23). Since the system often does not recognise a legitimate user, the sensor may require all facial features for authentication. However, it could also be possible that the other artefacts were presented incorrectly, as this device has an overall bad recognition rate.
More artefacts were able to bypass the Samsung Galaxy S20 as it accepted the artefact consisting of a vertical half of the user's face (left image in Figure 3.23), the whole area of interest and the area of interest without the mouth (second from the left and right image in Figure 3.23). The results indicate that the device's face recognition relies on the eye and nose region. Therefore, artefacts that do not contain these features cannot be recognised.
The Huawei P30 and Xiaomi Redmi Note 7 are less secure and could be unlocked with the same artefacts as the Galaxy S20. Additionally, they could be bypassed with the artefact containing only the eye area (see the left image in Figure 3.24). However, no device could be unlocked with an artefact containing only eyes or only the nose and mouth (middle and right image in Figure 3.24). This indicates that all face recognition systems might require the eyes to serve as evidence to detect a face. Furthermore, presenting only one feature seems to be insufficient as at least two were required.

Table 3.9: Results of the presentation attacks with the artefacts containing only parts of the face

Parts of the face	Galaxy S9	Galaxy S20	P30	Redmi Note 7
Half a face	No	Yes	Yes	Yes
Only area of interest	Yes	Yes	Yes	Yes
No mouth	No	Yes	Yes	Yes
No eyebrows	No	No	Yes	Yes
Only eye area	No	No	Yes	Yes
Only eyes	No	No	No	No
Only nose and mouth	No	No	No	No

In summary, devices with face recognition only relying on an RGB camera have a good chance to be bypassed with two-dimensional artefacts as all devices under attack could be unlocked, whereby printed artefacts achieved more reliable results. The artefacts should be presented under daylight or indirect light and with an angle and distance resembling the circumstances where a user would present his face.
Regarding the sensor's limitations for face recognition, the tested devices offered different levels of security concerning facial recognition. First of all, only one device required a coloured image. Unlocking the devices with artefacts with a minimum resolution of 100 dpi was also possible. Subsequently, an enlarged passport photo might be suitable for artefact creation.
In addition, no front-faced image was required as no test device needed the user's direct glance. The experiments stated that the head could be turned around 45 degrees to unlock the devices.

Furthermore, none of the devices required a complete head, so hair or ears seem irrelevant. In addition, not all landmark points are considered as the outline of the face was unnecessary, and the systems only relied on facial features. In some cases, the devices only needed two features to perform authentication, one of which must be the eyes. However, all distinctive areas, including the eyes, eyebrows, nose and mouth, should be present to ensure all devices can be unlocked with the artefact.

3.3 Attacking Devices Performing Face Recognition in the NIR Spectrum

As mentioned in Section 2.2, some devices include additional NIR sensors to execute face recognition. They require the face's appearance in NIR and partly work with depth data to perform the authentication. In this section, we created face masks to unlock the Dell XPS 15, the Microsoft Surface Pro 7, the Google Pixel 4, the iPhone Xs Max and the iPhone 12. As described in Section 3.1.1, both Windows Hello devices do not require depth data, whereas the three smartphones need additional depth information.

First, Section 3.3.1 examines the requirements of the devices under attack regarding the utilised sensors and the necessary facial features. In Section 3.3.2, we describe the results of the firmware and service reverse engineering of the Google Pixel 4 and iPhone 12. Furthermore, the logging process of the face authentication session with *adb* is explained.

Then, Section 3.3.3 explains the attacks with two-dimensional infrared images only targeting the Windows Hello devices. In Sections 3.3.4 and 3.3.5, the attacks with a face cast and three-dimensionally printed masks are outlined, describing their creation and evaluation with *HedgeCam2* and an additional infrared camera.

3.3.1 Requirements

Before the attacks, the requirements of the devices under attack were examined. The first step was determining whether only the infrared sensors are used for face recognition or if the RGB camera is also involved. Subsequently, the goal was to resolve if the devices work with *same-* or *cross-spectral band* face matching as described in Section 2.2.
In this regard, we identified and taped off the front camera module of the Google Pixel 4, both iPhones and the Microsoft Surface 7 Pro. As mentioned in Section 3.1.1, the face recognition on the Dell device works with an infrared camera integrated within the RGB camera. Subsequently, it was impossible to tape off this specific RGB camera. Nevertheless, it was possible to unlock all devices with taped off cameras via face recognition. To verify that the device's IR sensors and the Dell device's webcam work only in the infrared spectrum, we held an infrared bandpass filter in front of the camera, which blocks the frequencies of the visible spectrum and only allows the frequencies of the infrared range to pass. Even with this filter, all devices could be unlocked by face recognition. Therefore, the devices under attack execute a *same-spectral band* face matching and only rely on infrared imagery.

In this process, the proximity sensor was identified as another sensor required for face recognition being integrated into the iPhones and the Google Pixel 4 (see Figures 3.2 and 3.6). If this sensor is taped off, the device displays an error, stating that the sensor had to be cleaned. A *proximity sensor* detects objects without touching them by emitting infrared light to generate information about an object's presence and changing it into an electrical signal [101]. Here, this sensor is used to detect the distance of the face to distinguish if it is too far or too close to perform face recognition and possibly also to adapt the distances between features depending on the corresponding distance of the user.

Google Pixel 4
First, the Google Pixel 4's requirements were investigated. As it can be seen in Figure 3.2, besides the proximity sensor, four modules are involved in performing face recognition, including two IR cameras, a dot projector and a flood illuminator which we successively taped off to check whether all of them are needed. It turned out that the flood illuminator was only required in a dark environment. The remaining three modules are indispensable for face recognition.

Furthermore, we investigated the demands to enrol a user successfully. Here, we recognised that opened eyes are necessary to enrol a user. Afterwards, it was tested if the user must present his face frontally to the device. It stated that he could turn his head up to 10 degrees to unlock the device.
It was also recognised that the device's demands depend on whether the eyes are opened, covered with glasses or closed. In addition, the device's requirements partly differ in terms of the features needed when covering areas with human and non-human materials. When the user unlocks his phone with opened eyes, and covers areas with something humanoid like a hand, only the eyes and eyebrows are required to recognise the user. Subsequently, the mouth and nose can be covered.

In contrast, only the mouth can be covered when the user covers areas with something non-humanoid, like tissue paper. Subsequently, the eyes, eyebrows and nose must be visible to recognise the user successfully. More features could be required here because non-human materials look differently in infrared than skin. Accordingly, the device is still more likely to count human covers as the user's face.
Furthermore, the device does not always recognise the user if the eyebrows are hidden. As parts of the eyelid are covered, and as the eye region is the most important, the user is not recognised. Additionally, when the face is covered vertically, and only half a face is presented, the nose must be complete to recognise the user. The nose might be required here since only half of the eye and eyebrow region are present, and other features must be added accordingly to provide enough clues concerning the user's identity.

The device behaves differently when the user's eyes are closed or he wears glasses. This could be because not all eye region features are visible. Therefore, this region is not as distinctive. As with opened eyes, the device's requirements partly differ in terms of the features needed when covering areas with human and non-human materials. To still successfully unlock the device, the user can only cover his mouth with something human. When covering areas with no-humanoid materials, the device cannot be unlocked. If the user covers the nose and mouth with something non-human, the device does not even recognise a face.
Furthermore, the user is not recognised when the eyebrows are covered, whether with something human or non-human. The same goes for the vertically covered face. Here, the user is also not recognised even if the nose and mouth are evident. A reason could be that the eye area is not as distinctive as with opened eyes, and therefore, all features are required.

Summing up, apart from the fact that open or closed eyes make a difference, it is also crucial whether areas are covered by something human or non-human. If the user's eyes are open, the other features are not necessary to unlock the device, whereas with closed eyes, all remaining features are required. When the user wears glasses, some additional features are necessary.

iPhone
We investigated the demands of both Apple iPhones. As with the Google Pixel 4, we examined which of the three modules, despite the proximity sensor, displayed in Figure 3.6, are required for face recognition. Therefore, we also successively taped off the modules. Similar to the Google Pixel 4, the flood illuminator is only necessary in a dark environment. The other two modules are indispensable for face recognition.

To successfully enrol a user to the system or to unlock the devices, opened eyes directly directed on the screen with a maximum head rotation of 10 degrees are required. In contrast to the Google Pixel 4, the device's demands do not depend on whether the user wears glasses, and they do not distinguish between human and non-human covers.
To unlock the devices, the eyes, nose and one additional feature have to be visible. Subsequently, the user can either cover his eyebrows or his mouth. Additionally, when the user covers half his face vertically, the nose must be completely visible to unlock the phone. One explanation could be that Apple's facial recognition does not look at whether human skin is present but uses other clues to determine between a fake and a real user.

For Apple's iPhones, it is insignificant if parts of the face are covered with something human or non-human. However, it is more strict than the Google Pixel 4 as at least three features are required whereby the eyes and nose must be visible. Additionally, there are no significant differences in whether the user wears glasses.

Windows Hello

Lastly, we investigated the demands of both Windows Hello devices. As with the other devices, we first examined which modules of the Windows Surface Pro 7, displayed in Figure 3.10, are necessary for face recognition and, therefore, taped them successively off. Again, the front privacy light is not necessarily required for face authentication and is only used for dark environments.

Additionally, we researched the requirements for the user enrolment of both devices, and it stated that the user can be successfully enrolled with closed eyes. Moreover, the device's demands are looser than those of the other devices. Both devices can be unlocked with opened and closed eyes and glasses, which does not lead to different requirements. First, the eye area cannot be completely covered, and the nose must be visible; subsequently, only the mouth or the eyebrows can be covered. Furthermore, the user can cover half his face vertically, and, in contrast to the other devices, the nose does not have to be completely visible. Regarding the head's position, the user's head can only be turned up to 10 degrees.
As the devices only regard the two-dimensional face's appearance in IR, the areas that have to be visible could be similar to those of the two-dimensional attacks in the visible spectrum described in Section 3.2.

Further investigations

Afterwards, it was also investigated if all DUTs require the exact shape of the facial features. The aim was to generate demands for the mask creation as some features, such as eyebrows or lips, may have to be painted on the modelled masks. First, the eyebrow's shape was changed. Therefore, the user's eyebrows were covered by flatly glueing them onto the skin and applying baby powder and three layers of foundation. Then, the eyebrows were painted on using dark brown eyeshadow to draw the single hairs. All devices could be unlocked with drawn eyebrows.
Next, the lip contour was changed. Therefore, the lips were overlined with lipliner and filled with lipstick in dark brown. As with the eyebrows, all devices could be unlocked. Subsequently, the exact shape of the lips and eyebrows is not required, so it should be possible to paint these facial features.

3.3.2 Firmware and Service Reversing

To investigate what happens when the masks are presented to the Google Pixel 4, we examined the system logs of the device to find information about the face authentication service. Therefore, we worked with *adb* and *logcat*. We enabled USB-debugging on the Google Pixel 4. Next, we connected the device to a laptop on which *adb* is installed and started the command-line tool to display the system messages with *logcat*.

While the device was locked, we started *logcat* without filters to not exclude possible important information in advance. Then, we unlocked the device with an authorised user and closely examined the upcoming log messages. Here, it could be seen that there are two services named *BiometricsFace@1.0* and *afl*, which seem to be essential for face authentication as they show up when the device tries to recognise a face. Therefore, in the following, we filtered the system message output to only display log messages concerning either of the services.

Afterwards, the differences in the log messages in an authorised and an unauthorised attempt were considered. Figure 3.25 shows a section of the log messages that occur when an authorised user unlocks the device. It can be seen that the service *BiometricsFace@1.0* starts the authentication process with the log message "`Authenticating via airbrush...`" and a few messages later, the *afl* service messages "`Airbrush Faceauth Result: 1`". This is followed by two messages from the "`TrustZone app`", reporting 1 for two commands. These might be the responses from the security chip performing the data comparison.
Figure 3.26 shows a section of the log messages coming up when an unauthorised user tries to unlock the device. As with the previous log, the *BiometricsFace@1.0* service starts the authentication session, and moments later, the *afl* service reports 21 as the authentication result. Additionally, at the end of the section, it can be seen that the *BiometricsFace@1.0* reports that it identified a non-enrolled face for user 0.
Results for unlocking attempts with created masks are described in Section 3.3.5. It could be examined that the *afl* service displays an "`Airbrush Faceauth Result`" in the form of an integer between one and 24, where the result is one when the device recognises the authorised user and 21 if an unauthorised user tries to unlock the device.

```
09-04 11:52:09.472  1686  2517 I BiometricsFace@1.0: Authenticating via airbrush...
09-04 11:52:09.472  1686  2517 I afl     : Authenticate
09-04 11:52:09.472  1686  2517 I afl     : IOC_START operation: 4
09-04 11:52:09.618  1686  2517 I afl     : FW Version: 0x15800
09-04 11:52:09.618  1686  2517 I afl     : Airbrush Faceauth Result: 1
09-04 11:52:09.621  1686 10463 I BiometricsFace@1.0: TrustZone app reported 1 for command 101
09-04 11:52:09.624  1686 10465 I BiometricsFace@1.0: TrustZone app reported 1 for command 102
09-04 11:52:09.625  1686  2517 I BiometricsFace@1.0: Not running autocal. (Last run: 3815 seconds ago.)
09-04 11:52:09.625  1686  2517 I BiometricsFace@1.0: Operation finished with success: authenticate(0)
09-04 11:52:09.625  1686  2517 E BiometricsFace@1.0: Total dropped frames: 3136/564802
09-04 11:52:09.625  1686  2517 I BiometricsFace@1.0: CameraHelper::Close started: logical
09-04 11:52:09.625  1686  3076 I afl     : Release lambda
09-04 11:52:09.625  1686  2517 I BiometricsFace@1.0: CameraHelper::ACameraDevice_close started: logical
09-04 11:52:09.744  1686  2517 I BiometricsFace@1.0: CameraHelper::ACameraDevice_close ended: logical
09-04 11:52:09.751  1686  2517 I BiometricsFace@1.0: CameraHelper::Close finished: logical
09-04 11:52:09.751  1686  2517 W BiometricsFace@1.0: No unlock result to report!
```

Figure 3.25: Section of the log messages when an authorised user unlocks the device with important messages highlighted in red

```
09-04 13:16:35.216  1686  2517 I BiometricsFace@1.0: Authenticating via airbrush...
09-04 13:16:35.216  1686  2517 I afl     : Authenticate
09-04 13:16:35.216  1686  2517 I afl     : IOC_START operation: 4
09-04 13:16:35.260  1686 16402 I BiometricsFace@1.0: CameraHelper::OnCaptureCompleted observed partial frame drop
09-04 13:16:35.338  1686  2517 I afl     : FW Version: 0x15800
09-04 13:16:35.823  1686  2517 I afl     : Airbrush Faceauth Result: 21
09-04 13:16:35.833  1686 16429 I BiometricsFace@1.0: TrustZone app reported 1 for command 101
09-04 13:16:35.838  1686 16432 I BiometricsFace@1.0: TrustZone app reported 1 for command 102
09-04 13:16:35.838  1686  2517 I BiometricsFace@1.0: Triggering autocal run. (Last run: 7594 seconds ago.)
09-04 13:16:35.838  1686  2517 I BiometricsFace@1.0: Autocal using repair calibration: false
09-04 13:16:35.840  1686 16434 I BiometricsFace@1.0: Run split pipeline in autocal TZ app with multiple commands:
09-04 13:16:35.840  1686 16433 E BiometricsFace@1.0: Dumping autocal
09-04 13:16:35.844  1686 16434 I BiometricsFace@1.0: Autocal command = 2, status = -1
09-04 13:16:35.847  1686 16434 I BiometricsFace@1.0: Autocal command = 3, status = -1
09-04 13:16:35.850  1686 16434 I BiometricsFace@1.0: Autocal command = 4, status = -1
09-04 13:16:35.853  1686 16434 I BiometricsFace@1.0: Autocal command = 5, status = -1
09-04 13:16:35.872  1686 16434 I BiometricsFace@1.0: Autocal command = 6, status = -1
09-04 13:16:35.889  1686 16434 I BiometricsFace@1.0: Autocal command = 7, status = -1
09-04 13:16:35.892  1686 16434 I BiometricsFace@1.0: Autocal command = 8, status = -1
09-04 13:16:35.895  1686 16434 I BiometricsFace@1.0: Autocal command = 9, status = -1
09-04 13:16:35.917  1686 16434 I BiometricsFace@1.0: Autocal command = 10, status = -1
09-04 13:16:35.951  1686 16434 I BiometricsFace@1.0: Autocal command = 11, status = -1
09-04 13:16:35.955  1686 16434 I BiometricsFace@1.0: Autocal command = 12, status = -1
09-04 13:16:35.976  1686 16434 I BiometricsFace@1.0: Autocal command = 13, status = -1
09-04 13:16:35.979  1686 16434 I BiometricsFace@1.0: Autocal command = 14, status = -1
09-04 13:16:35.990  1686 16434 I BiometricsFace@1.0: Autocal command = 15, status = 1
09-04 13:16:35.993  1686 16434 I BiometricsFace@1.0: Autocal command = 100, status = -1
09-04 13:16:35.995  1686 16434 I BiometricsFace@1.0: Autocal command = 100, status = 1
09-04 13:16:36.079  1686 16434 I BiometricsFace@1.0: TrustZone app reported 1 for command 0
09-04 13:16:36.079  1686 16433 E BiometricsFace@1.0: Dumping autocal
09-04 13:16:36.080  1686  2517 E BiometricsFace@1.0: Autocal succeeded, calibration updated.
09-04 13:16:36.087  1686  2517 E BiometricsFace@1.0: Identified a non-enrolled face for user: 0
09-04 13:16:36.087  1686  2517 I BiometricsFace@1.0: Operation finished with success: authenticate(0)
```

Figure 3.26: Section of the log messages when an unauthorised user tries to unlock the device with important messages highlighted in red

To research the meaning of the different authentication results displayed by the *afl* service and to gain more information about the underlying software, the firmware and the service of the face authentication on the Google Pixel 4 were partly reverse-engineered. Therefore, a rooted device was used, meaning we could access and edit system files. Subsequently, the firmware files and the face recognition service were searched and extracted by copying them onto the analysis device. It stated that there were four versions of the face authentication firmware (*v1* to *v4*), each consisting of four files *b00* to *b03*.

First, the service named *android.hardware.biometrics.face@1.0-service.google* was analysed, which might be responsible for the communication between the central processing unit (CPU) and the coprocessor, which is a processor that relieves the CPU and adds functions to it to accelerate the system [102]. Therefore, the service.google file was loaded into *HxD* to get an overview and identify the file format. Figure 3.27 displays the beginning of the file in the hex editor.
At the beginning of the file, the magic bytes `0x7F454C46` indicating an Executable and Linking Format (ELF) file can be seen, which is a standard binary format for executable files of UNIX systems that is also often used for firmware files for the ARM microprocessor architecture [103]. Furthermore, it was examined that the service is Little-Endian-based and designed for a 64-bit system. Also of interest is the Instruction Set architecture, which is entered in the header at position `0x12`. The value `0xB7` stands for aarch64, describing the 64-bit extension for ARM. Furthermore, the Address of Entry Point is saved in the ELF-Header on position `0x18`, which is essential for further analysis as it contains the address where the first instruction of the program is located; in this case at the address `0x028000`.

```
Offset(h)  00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F  Dekodierter Text
00000000   7F 45 4C 46 02 01 01 00 00 00 00 00 00 00 00 00  .ELF............
00000010   03 00 B7 00 01 00 00 00 00 80 02 00 00 00 00 00  ..·.....€......
00000020   40 00 00 00 00 00 00 00 F0 5C 0A 00 00 00 00 00  @.......ð\......
00000030   00 00 00 00 40 00 38 00 0B 00 40 00 1C 00 1A 00  ....@.8...@.....
```

Figure 3.27: Beginning of the service file *android.hardware.biometrics.face@1.0-service.google* with the magic bytes, the Instruction Set architecture and the Address of Entry Point marked in red

After importing the file into *Ghidra*, we analysed the file and its functionality, starting at the main function located at `0x00128000`. We could identify the function responsible for printing "Airbrush Faceauth Result: " but no internal calls to that function could be found. This leads to the assumption that it is called externally by the face authentication firmware itself.

Another approach was to analyse the firmware of the face authentication, which files were designed for a 32-bit system. However, the system is 64-bit based, so it was assumed that the face authentication is carried out on a coprocessor. As mentioned, there are four firmware versions, each containing four files. However, only the *b03* files are interesting as they provide the greatest file sizes with around 30 MB, whereas the other files are only 1 KB. One reason for several firmware versions could be that they have been adapted more often, and a new version has been released each time. We chose to analyse the *faceauth_v4.b03* file, which is a raw binary file not having a standardised structure like, for example, ELF. However, based on the datasheet of the Google Pixel 4, the chip type was identified as a Qualcomm SM8150 Snapdragon 855. Therefore, we chose ARM Cortex 32-bit Little Endian as the language to analyse the file with *Ghidra*.

As many references within the firmware point to a region starting at `0x20000000`, it indicates the RAM location. The binary's entry point was identified through the reset function referenced by the Vector Table located at the beginning of the file. In the reset function, the BSS and DATA regions are initialised, where the BSS region contains only uninitialised and the DATA region initialised data. The aim was to generate more information about the data structures processed during face authentication. However, it provided no further evidence as the DATA area is only written at runtime.

A string search identified some strings, including the term *TPU* (Tensor Processing Unit), which indicates the usage of neural networks or machine learning as this is a processor to accelerate applications that process data with artificial neural networks [104]. This could lead to the assumption that Google's Neural Core, whose task is not described in more detail, involves face authentication and acts as the coprocessor with which the service communicates. Subsequently, it would be possible for the device to improve with each presentation, which was also noticed in the test series described in Section 3.2.
Another chip involved in the authentication might be the Titan M security chip, as it includes the Trusted Execution Environment (TEE), mentioned in Section 2.3, which stores the biometric data. As the biometric data remains in the chip, the Titan M might internally compare the stored data with the recorded data and only forward the authentication result [105]. This might lead to three face authentication components: the CPU, the Neural Core and the Titan M security chip.

When analysing the strings, including *TPU*, it was recognised that the skin and the facial depth are important for face authentication. Table 3.10 lists the corresponding strings.
Regarding the strings concerning the skin, the point pattern on the skin ("`TPU_DOT_SKIN_IN`" and "`DOT_TPUSKIN_OUT`") and some type of skin flood ("`TPU_FLOOD_SKIN_IN`" and "`FLOOD_TPUSKIN_OUT`") are interesting, which might indicate the use of a flood fill algorithm. Additionally, it can be seen that the flood skin result could lead to a reject indicated by the string "`TPU_FLD_SKIN_REJECT_TASK`".

Next, we searched for further strings containing "SKIN" but not "TPU" and found "FLOOD_SKIN_SCORE" and "DOT_SKIN_SCORE". As both strings include the term "SCORE", this might imply that the software calculates a score for the skin flood and the dot pattern on the skin. The string "Skinclassifier" was also found, which might work with the resulting scores to classify the face's surface as skin or no skin.
Furthermore, the last two strings containing skin ("TPU_SKIN_INPUT_PAD" and "TPU_SKIN_OUTPUT_PAD") enclose the string "PAD", which might imply that the Google Pixel 4 performs a type of Presentation Attack Detection based on the skin's appearance. As mentioned in Section 2.3, PAD is the recognition of attacks on the Biometric Sensor when a biometric feature is presented [48].

Besides these strings, some contain "TPU" and "depth"; therefore, the depth data might also be processed with neural networks. Next, we searched for further strings containing "depth" and found, for example, "Detected depth score:". This indicates that the device not only takes care that a three-dimensional object is presented but also measures the user's depth data, which must match those of the presented object. However, it can be seen that no string contains the term "reject", implying that the skin might be more important than the depth to distinguish between a mask and a natural face.

Table 3.10: Important TPU strings found in the firmware file *faceauth_v4.b03* of the Google Pixel 4

Strings containing "skin"	**Strings containing "depth"**
TPU_FLOOD_SKIN_IN	TPU_DEPTH_IN
FLOOD_TPUSKIN_OUT	TPU_ULTRADEPTH_TASK
TPU_DOT_SKIN_IN	TPU_DEPTHID_TASK
DOT_TPUSKIN_OUT	
TPU_FLD_SKIN_REJECT_TASK	
TPU_DOT_SKIN_ACCEPT_TASK	
TPU_SKIN_INPUT_PAD	
TPU_SKIN_OUTPUT_PAD	

Furthermore, strings beginning with "REJECT", listed below, might help understand why masks are rejected. These strings are not directly used but are saved in an array called by different functions and describe why the device remains locked at certain authentication attempts. As can be seen, most reasons are based on the face's presentation, like the face's position or the distance between the device and the face. However, there are also reasons indicating a type of PAD like "REJECT_SKIN" or "REJECT_SPOOF".
To examine where this array is used, we worked with *FRIDA* and its tracing engine named *Stalker* to follow the references on this array. It turned out that the Airbrush Faceauth Results displayed in the log messages are mapped to this array, where the result corresponds to the position in the array. However, the numeration starts with two and ends with 24; number one is not listed here as it indicates a successful attempt. Here, it can be seen that the 21, which occurs when a non-legitimate user presents his face, is a complete rejection (HARD_REJECT).

2. REJECT_NO_FACE
3. REJECT_MAX_FACE_CAPACITY
4. REJECT_NO_ATTENTION
5. REJECT_TOO_SMALL
6. REJECT_CLIPPED_LEFT
7. REJECT_CLIPPED_RIGHT
8. REJECT_CLIPPED_TOP
9. REJECT_CLIPPED_BOTTOM
10. REJECT_TOO_DARK
11. REJECT_TOO_BRIGHT
12. REJECT_TILT_ANGLE
13. REJECT_PAN_ANGLE
14. REJECT_TOO_CLOSE
15. REJECT_TOO_FAR
16. REJECT_USERS_FULL
17. REJECT_PROFILES_FULL
18. REJECT_INVALID_DEPTH
19. REJECT_SKIN
20. REJECT_SPOOF
21. HARD_REJECT
22. ERROR
23. REJECT_FRAME_SELECT
24. REJECT_FEATURE_FLAGS

Concluding, three processors might be involved in the face authentication process, including the CPU, the TPU on Google's Neural Core and the Titan M security chip. Besides the distances between the facial features, the face authentication is based on the user's skin, the dot pattern and the depth data. Furthermore, especially the skin might lead to a mask's rejection. In addition, there are several reject reasons, including the false positioning of the face, the skin or spoofing attempts. Subsequently, the Google Pixel 4 executes a type of spoofing detection, which might be based on the skin score.

As with the Google Pixel 4, the iPhone 12's software was also analysed, starting with the system log files. Therefore, the iPhone 12 was connected to a Mac Book Pro, and the logging process was started via the console. We logged the sessions when an authorised or unauthorised user tried to unlock the device. However, these log files were less informative than those from the Google Pixel 4. Some strings indicated the authentication process but did not yield new insights into why someone was rejected.
Another approach was to regard the log messages of a rooted device, as this might lead to more detailed system messages. However, when rooting the device, Face ID is no longer available as, for example, the sensor's serial number is wiped, and the device does not recognise it any longer. Therefore, this approach was not further investigated.

To gain further information about the underlying software, the service of the iPhone 12 named *facemetricsd* was analysed. This file is in Mac OS X Mach-O format that uses the aarch64 Instruction Set Architecture working on a Little Endian-based 64-bit system. It was analysed with *Ghidra*, but the service did not provide helpful findings regarding Apple's FaceID because only strings concerning emotion tracking, like "`angry mouth`", were visible.

Furthermore, we extracted the firmware of the iPhone XR. We could not work with the iPhone 12 as a jailbreak is required to extract the software but all known exploits only work up to the iPhone X at the time of writing. The software file named *sep-firmware.n841.RELEASE.im4p.dec* is a binary file, and the ARM Cortex 32-bit Little Endian architecture was chosen as a language to analyse the file with *Ghidra*. After the analysis, we performed the string search and found many strings, some related to biometrics. However, no strings provided further information about the rejection of authentication attempts. Interestingly, some strings refer to TouchID, although this device no longer offers fingerprint unlocking. A reason could be that the previous firmware for biometric authentication was further used and adapted, but no new firmware was written.

The Google Pixel 4 software provides evidence concerning the rejection of authentication attempts, whereas the iPhone software offers hardly any clues. Subsequently, when it comes to the mask's evaluation in Section 3.3.5, only the log messages of the Google Pixel 4 are considered.

3.3.3 Evaluation of Attacks with Two-dimensional Artefacts

Two-dimensional attacks were again carried out to target the Windows Hello devices. As mentioned in Section 3.3.1, they perform face recognition only with the two-dimensional NIR appearance of the user and do not capture the facial depth. Subsequently, as a first approach, two-dimensional artefacts were created to bypass the devices.

As the Windows Hello devices work in the NIR spectrum, a printed NIR photo of the user might be suitable to bypass the devices. Therefore, the user's face was initially captured with two infrared cameras. *HedgeCam2*, described in Section 3.1.4, was used to access the infrared camera of the Google Pixel 4 to capture the user's face with a resolution of 640x480 pixels.
In addition, the user's face was also captured with the IR camera of the Raspberry Pi, described in Section 3.1.3. We worked with *libcamera* (see Section 3.1.4) to record the user's face with a full resolution of five megapixels and saved it as a JPEG file. Figure 3.28 displays the captured images of the user with *HedgeCam2* (left) and with the IR camera of the Raspberry Pi (right).

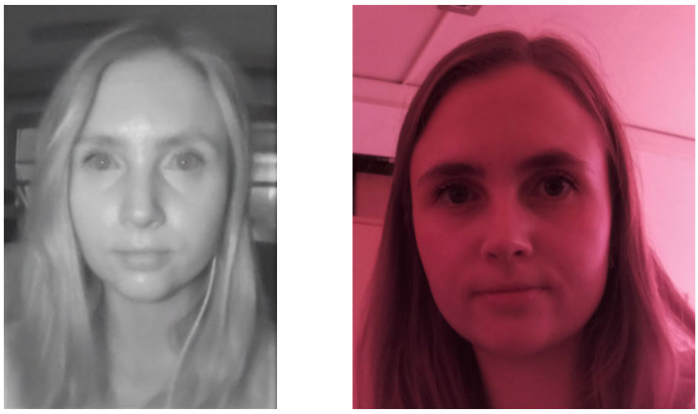

Figure 3.28: Captured images of the user with *HedgeCam2* (left) and with the IR camera of the Raspberry Pi (right)

As with the previous two-dimensional attacks described in Section 3.2, the images were scaled to DIN-A4 with *GIMP* by ensuring the ratio of the edges was kept so the distances between the facial features would not be distorted. The resulting images were exported as PNG files and printed with 300 dpi on regular paper with the *HP Laserjet 500 M551* printer. The image captured with the IR camera of the Raspberry Pi was printed with and without colour to examine if these artefacts led to different results. Furthermore, as with the artefacts described in Section 3.2, the background was cut off as the devices might detect the non-matching background. In addition, the artefact generated from a photo captured with an RGB camera was also chosen to compare it to the NIR ones. Figure 3.29 displays the resulting artefacts.

Figure 3.29: Printed artefacts captured with the RGB camera (left), captured with *HedgeCam2* (second from left) and captured with the IR camera of the Raspberry Pi printed without (second from right) and with colour (right)

These artefacts were presented to the Windows Surface 7 Pro and the Dell XPS 15, but could not unlock the devices. The artefacts created from the images captured with the RGB camera (left image in Figure 3.29) and with *HedgeCam2* (second from the left image in Figure 3.29) were directly rejected. In contrast, both artefacts captured with the IR camera of the Raspberry Pi (second from right and the right image in Figure 3.29) were not directly rejected as the devices responded that they had to make sure that it is a legitimate user. Only after multiple attempts, these artefacts were eventually rejected, too. However, enrolling all artefacts was possible, but the devices remained locked when again presenting the artefacts.

We evaluated their appearance in *HedgeCam2*, displayed in Figure 3.30. The left image shows the NIR appearance of the artefact captured with an RBG camera, and it can be seen that the facial features are visible. However, the area of the user's skin is strongly reflective and appears white. Subsequently, the artefact looks unnatural. In contrast, the facial features of the artefact recorded with *HedgeCam2*, displayed in the second from the left picture of Figure 3.30, are barely visible, but its reflectance is muted. However, the face is barely recognisable. When it comes to the grayscale artefact created from the photo captured with the IR camera of the Raspberry Pi (see the second from the right image in Figure 3.30), the features are more visible than those of the previous artefact; subsequently, the face is better detectable. The appearance of the coloured artefact is the best as it looks the most natural, which can be seen in the right image in Figure 3.30. The user's facial features are detectable, and the skin's area is less reflective. However, the NIR light is strongly reflected from the paper's surface.

Figure 3.30: Appearance of the two-dimensional artefact captured with the RGB camera (left), captured with *HedgeCam2* (second from left) and captured with the IR camera of the Raspberry Pi printed without (second from right) and with colour (right) in NIR with *HedgeCam2*

Concluding, the artefacts captured with the IR camera of the Raspberry Pi provided the best results as both devices did not reject them directly. However, as seen in NIR, the NIR light is reflected strongly on the paper's surface, and therefore, it seems unnatural. Furthermore, considering the unlocking results, printing the artefact captured with the IR camera of the Raspberry Pi in grayscale did not lead to different results, but its appearance differs in NIR. In general, those artefacts are unsuitable for bypassing the Windows Hello devices; subsequently, different approaches must be considered.

3.3.4 Evaluation of Attacks with Face Cast Masks

We created three-dimensional artefacts where the first masks were generated with a face cast, a method to replicate a face. No additional technology was required as the face cast was formed directly from the user's face. It is used to create the masks that are later supposed to fit on another person's face to impersonate the legitimate user. This section describes the modelling of the face cast, the creation of the masks and their evaluation.

Face Cast creation

To model the face cast, the user's face must be prepared to apply the materials. We used plastic wrap to cover the hair to prevent it from being plastered. We cut the protruding ends of the plastic wrap to fit the face and then glued the edges to the face with eyelash glue. Additionally, the face, especially the eyebrows, was covered with Vaseline to prevent the facial hair from sticking to the plaster.

We worked with plaster bandages cut into 3x5 cm pieces, which were soaked with water, then applied to the face and smoothed out to generate the plaster mould. Here, the bandage holes must be closed to create a homogenous surface. We plastered the entire face past the hairline, leaving out the nostrils and eyes. Overall, three plaster bandage layers were applied to create a stable form. Each layer had to be fully cured before the next layer was applied. The mask had to cure for around 30 minutes before being taken off the face. The left image in Figure 3.31 displays the user's face covered with three layers of plaster bandages. After the mask was taken off the face, thin areas were subsequently reinforced with further pieces of plaster, and the eyes and nostrils were covered with three layers of plaster bandages (see the second from left and second from right image in Figure 3.31).

The inside of the plaster mould was stitched with Vaseline before the form was filled so that the actual face cast could be removed more easily later. A plaster-based modelling clay was poured into the mould and cured for 24 hours. The finished face cast could be removed from the mould and was sealed with a thin layer of latex to erase the furrows (see the right image in Figure 3.31).

Figure 3.31: The user's face covered with plaster bandages (left), the plaster form with filled eyes and nostrils (second from left), the inside of the plaster form (second from right) and resulting face cast (right)

Mask creation and evaluation

Figure 3.32 displays the three masks. Liquid latex was applied to the face cast, forming the mask displayed in the left image, which was presented with the face cast underneath. This mask was not recognised as a person by the devices as the facial features are not visible, especially the eye's shape cannot be detected.

The mask was removed from the face cast with baby powder to prevent it from sticking together (see the middle image of Figure 3.32), but it was not recognised as a person either. Besides the missing facial features, the shape of the face was also lost because the mask was not stable enough.

We tried to mimic the facial features and gave the mask a more humanoid look by enhancing it with makeup by drawing eyebrows with dark brown and lips with red eyeshadow, which can be seen in the right image of Figure 3.32. However, the depth data was off again. The Google Pixel 4 and Windows Hello devices partly recognised this mask as a person. Since the Apple iPhones require open eyes, it was unsurprising that these masks could not unlock the devices.

Figure 3.32: Mask with the face cast underneath (left), the mask lifted from the face cast (middle) and the enhanced mask with makeup (right)

Concluding, the mask's eye region needs to be more distinctive, as the eye sockets are not as deep and do not resemble the shape of the user's eye. Additionally, the mouth region is problematic as the lip structure could not be precisely modelled, so the lips' outline could not be drawn well. Those masks were not evaluated with *HedgeCam2* or *libcamera* as they are not similar enough to the user's face.

We tried to overcome the problems described by deepening the eye sockets to mimic the depth of the real sockets. Therefore, we measured the depth of the eye sockets in the outer and inner eye corners and the middle of the crease with a calliper. The modelling clay was worked on by hand with a thin slotted screwdriver, removing material until the desired depth and shape were achieved (see Figure 3.33 left).
As the enhanced mask provided the best results, we again applied liquid latex to the face cast and drew eyebrows and lips with dark brown and red eyeshadow. This mask was removed from the face cast but could not bypass any of the devices. Regarding the shaped eyes, the crease area might be deeper, but it still does not look like humanoid eyes.
As with the other masks, the eye region might not be as detailed, and the mask is not similar enough to the user's face. To test if the eye region is insufficient, we cut off the eyes (see Figure 3.33 right) and held the mask before the user's face to provide an exact eye region. However, this mask also did not unlock any device, as the problem might be the still deformed face. When removing the mask from the face cast, it cannot hold the correct shape, and therefore, the depth information is not adequately represented. In addition, the user's features are displayed distorted. These masks were not evaluated with *HedgeCam2* or *libcamera* as the overall appearance is not similar enough to a human face.

Figure 3.33: Face cast with deepened eye region (left), enhanced mask with makeup (middle) and enhanced mask with eyes cut out (right)

All masks created with a face cast are not detailed enough to mimic the user's real face, and therefore, no device could be unlocked with those masks. The shape and dimensions of the face are adapted, but the individual features such as eyebrows, lips or eyes cannot be represented in detail. Especially the eyes' specific shape and the eye sockets' depth are lost, which are the most important features. Regarding the two-dimensional attacks in Section 3.2 or the specified device's requirements in Section 3.3.1, the eyes are a distinctive factor for face recognition. Therefore, this region must be recreated in detail to recognise the mask as an authorised user.

In addition, removing the latex mask from the mould does not bring any advantages. On the contrary, it is rather disadvantageous because the latex mask is not stable enough to hold the exact shape. However, it would also not be possible to apply more layers of latex milk, as otherwise, the dimensions of the features would be lost. Accordingly, methods other than face casting must be evaluated to generate the masks, which show the user's face, especially the eye area, in more detail and provide a flat surface.

3.3.5 Evaluation of Attacks with Three-dimensionally Printed Masks

We worked with the three scanners *Shining 3D EinScan-SP*, *Handyscan 3D* and *Artec EVA 3D* as well as with SfM to generate a three-dimensional model of the user's face. The scanning, processing and printing of the models captured with these devices are outlined in this section.
If not mentioned otherwise, when covering the masks with liquid latex, 20 layers are applied to provide a surface for the additives and to mimic human skin. It was applied in thin layers; each layer had to cure before the next one was applied.

The mask's appearance in NIR was evaluated to make it look more similar to the user. Therefore, *HedgeCam2* was used to compare the mask's appearance to the user's face in NIR. Here, the focus was primarily on the appearance of the skin, as this is presumably an essential factor for face recognition or forgery detection, as can be seen from the strings described in Section 3.3.2. Additionally, the feature presentation in NIR was examined. The left and the second from the left image in Figure 3.34 display the user's appearance in NIR, later used as a reference to evaluate the masks.

Furthermore, the device's point clouds were made visible to compare the pattern on the mask with those on a real face. As mentioned in Section 3.3.2, facial depth is also a decisive factor in the authentication process. Therefore, regarding the mask's evaluation, another focus was laid on the dot's appearance.
The IR camera with the Raspberry Pi, described in Section 3.1.3, was used over *libcamera* to capture a ten-second long video of the point clouds of the Google Pixel 4 and the Apple iPhone 12 with a resolution of 1920x1080 pixels. The video stream was disassembled into frames with *FFmpeg*, and the resulting frames were saved as PNG files.
The middle image in Figure 3.34 displays the dot pattern on the user's face emitted by the Google Pixel 4, whereas the second from right and the right image show both dot patterns on the user's face emitted by the Apple iPhone 12. These images are used as references to evaluate the mask's appearance in the following. We do not display the point cloud of the Apple iPhone Xs Max as it works with the same technique as the Apple iPhone 12.

Figure 3.34: User's appearance in *HedgeCam2* with closed (left) and opened eyes (second from left), dot pattern from the Google Pixel 4 on the user's face (middle) and both dot patterns from the Apple iPhone 12 on the user's face (second from right and right)

Initially, we concentrated on creating masks with closed eyes, as the Google Pixel 4 and both Windows Hello devices can be unlocked with closed eyes. It might lead to better results when the eye area is not imitated. These masks were nevertheless presented to the Apple iPhones for the sake of completeness. To evaluate the suitability of the mask's material, they are also presented without additives.

Shining 3D EinScan-SP

The *Shining 3D EinScan-SP*, described in Section 3.1.3, working with the *EXScan S* software (see Section 3.1.4) was used to capture the user's face. Before capturing the face, the scanner must be calibrated with a turntable and a blackboard with black dots. Furthermore, a white balance must be executed with white paper.
The scanner was mounted on a tripod standing on a desk. The scanned person was positioned with a chair in front of the scanner to enable a full-size face record to provide a more detailed scan. Additionally, the face was aligned according to the specifications in the software, and High Dynamic Range (HDR) was enabled so that the scanner independently adjusts the illumination of the face.

For scanning, the user's face was sprayed with dry shampoo, and the lips were covered with baby powder to brighten up dark areas and minimise the skin's shine, increasing the quality of the scanning results. We performed 13 scans from different perspectives (frontal, left, right, top, bottom), with at least one-third of the scans overlapping. Hence, the scanner recognises matching points in different scans and uses them as an orientation to merge the scans. The single scans are displayed in *EXScan S*, and if all scans are sufficient, meaning they contain few false scanned areas, a global optimisation is carried out. The scans were meshed with to create a waterproof model, meaning there are no holes in the resulting mesh, and it exported as a *Standard Triangle Language* (STL) file, saving the three-dimensional data as triangles to create the object's surface [106].

To further process the mesh, it was loaded into *Meshmixer* described in Section 3.1.4. Figure 3.35 displays the unprocessed STL file on the left. We aligned the model with the instructions `Align` and `Transform` to the point of origin and then cleaned up the model. Here, we used the instruction `Plane Cut` to cut off the backside of the scan, which does not belong to the face displayed in the second picture from left in Figure 3.35. Therefore, we achieved a straight edge, which later served as a support surface for printing the model.

Additionally, imperfect areas were erased manually by selecting and removing them. The resulting holes were filled with `Smooth Fill` and smoothed with the `Smooth Brush`. The processed model, which is then again exported as an STL file, is displayed on the right and the second from the right picture in Figure 3.35.

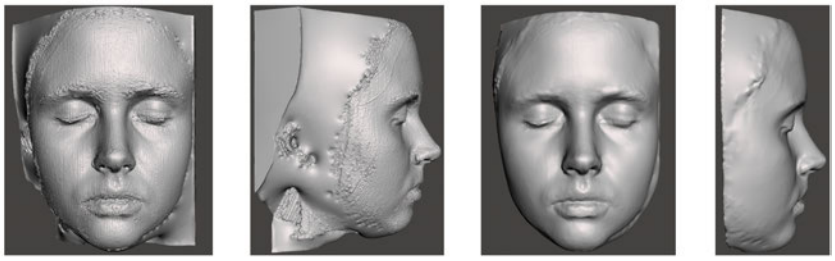

Figure 3.35: Frontal view (left) and side view (second from left) of the STL file before further processing and frontal view (second from right) and side view (right) of the STL file after processing with *Meshmixer*

To prepare the model for printing, the model was sliced with *PrusaSlicer*, described in Section 3.1.4. The settings were adapted to choose the utilised printer *Original Prusa i3 MK3S+* with the 0.2 mm nozzle and the filament type *Polylactic Acid* (PLA). We imported the processed STL file and aligned it with the support surface described above on the printing bed. Furthermore, we chose an infill of 5% as the model was not required to be filled on the inside.
The model was adaptively sliced, meaning that the thickness is adjusted according to the required level of detail so that more detailed areas are printed with a lower layer thickness. Therefore, the eyes, mouth and nose were printed with a lower layer thickness of around 0.07 mm, whereas the remaining areas with fewer details were printed with a layer thickness of around 0.23 mm.
The resulting G-code explained in Section 3.1.4 was exported and forwarded to the printer. As mentioned, we printed the model with the *Original Prusa i3 MK3S+* with grey PLA, leading to a printing duration of around ten hours. The left picture in Figure 3.36 shows the printed model.

The mask was presented without liquid latex, and both Windows Hello devices could not be bypassed as the systems did not react when the mask was presented. Subsequently, they did not recognise a face, so the mask could not be enrolled as a user. We were also unable to unlock the Google Pixel 4 as the device responded after a few moments that it did not recognise the face. When regarding the log messages from the Google Pixel 4, it can be seen that the device did not detect a face as the authentication result states 23 being `REJECT_FRAME_SELECT` (see Section 3.3.2) and the warning `NOT_DETECTED` is stated. Another reject reason is the mask's brightness as it gets the result 11 being `REJECT_TOO_BRIGHT`. Ultimately, the device rejected the mask with a 21, meaning `HARD_REJECT`. We also attacked both Apple iPhones, which remain locked. As with the Windows Hello device, they did not react when the mask was presented and, therefore, did not detect a face.

We examined the mask appearance in NIR with *HedgeCam2* to verify the mask's brightness. The result is displayed in Figure 3.36 as the second picture from left. It can be seen that the PLA is more reflective than human skin (left image in Figure 3.34), and the mask's facial features are hardly visible. This high reflectance could be the reason for the Airbrush Faceauth Result 11. Additionally, the material seems way darker than human skin in NIR.

Regarding the attempt to capture the point cloud from the Google Pixel 4 (second from right image in Figure 3.36), the point cloud is barely visible, and the NIR light is almost transmitted through the mask and therefore, compared to the user's point cloud (middle image in Figure 3.34), it is less visible. When it comes to the Apple iPhone 12 (see the right image of Figure 3.36), it can be seen that the mask is not categorised as a face because only the infrared flashlight is visible, and the device did not send out the point clouds.

Figure 3.36: Printed mask with grey PLA (left), mask's appearance in *HedgeCam2* (second from left), mask's point cloud from the Google Pixel 4 (second from right) and mask's point cloud from the Apple iPhone 12 (right)

The overall problem with this mask could be that the facial features are barely visible; subsequently, all devices could not recognise a face. Additionally, this PLA might not be suitable for mask creation as it is highly reflective and partly transparent. As mentioned in Section 3.3.2, the face authentication of the Google Pixel 4 attaches great importance to the appearance of the skin, but this mask's surface is way darker, and the point cloud is less visible and not as strongly reflected as with a human face.

Liquid latex was applied to provide a more skin-like surface, minimise the mask's reflectance, and brighten the mask in NIR. In addition, foundation and powder were applied to mimic the skin structure and to reduce the reflectance further. Makeup was applied, drawing the eyebrows, lips and lash lines with dark brown and red eyeshadow to highlight the facial features (see Figure 3.37 on the left).

Both Windows Hello devices recognised this mask as a face, and it could be enrolled as a user. However, both devices remained locked. Even if they categorise it as a face, the two-dimensional image of the mask looks not like the legitimate user as described in the following. The Google Pixel 4 could not be unlocked. However, the device rejects the mask more quickly as it might detect a face but categorises it as an unidentified user. We again logged the session when we presented the mask to the device. Some logging sections resemble the previous mask except for result 11 (`REJECT_TOO_BRIGHT`). Here, the mask gets the result 23 (`NOT_DETECTED`) multiple times and is then rejected with the 21 (`HARD_REJECT`). However, when the device responds quickly that the user is not recognised, it can be seen in the log messages that the mask directly gets 21, like a humanoid unauthorised person trying to unlock the device. The Apple iPhones do not categorise the mask as a face. As with the previous mask, they did not react when it was presented and remained locked.

The overall appearance of the mask with *HedgeCam2*, displayed in the second picture from the left in Figure 3.37, is slightly darker than human skin (left image in Figure 3.34). Due to the made-up facial features, the mask looks more humanoid and now resembles the natural person more. However, the facial features are still not present enough, as the eyebrows are barely visible, and the lips are not visible. In addition, the surface is less reflective than the previous mask but still darker than human skin.
When we captured the device's point clouds with *libcamera*, it can be seen that the point cloud of the Google Pixel 4, shown in the picture second from right of Figure 3.37, is more intense than before; it might not be transmitted through the mask due to the appliances. Therefore, it resembles the point cloud of the user's face (middle image in Figure 3.34), but its edges are sharper than those on the user's skin. It seems as if the infrared light penetrates the skin, whereas the dots on the mask provide precise edges, and it does not seem like the light penetrates the mask. However, the Apple iPhone 12 again only sent the flashlight and no point cloud (see the right image in Figure 3.37).

Figure 3.37: Mask with applied makeup (left), its appearance in *HedgeCam2* (second from left), the Google Pixel 4's point cloud (second from right) and the Apple iPhone 12's point cloud (right)

In summary, applying makeup improves the mask's humanoid appearance. However, the chosen pigments must be amplified as they are barely visible. Furthermore, the surface structure and its brightness do not reflect the properties of human skin.
Other scanners that produce higher quality scans were considered so that the model's surface requires less smoothing and thus keeps more information. Additionally, a scan capturing opened eyes should be carried out. To fit the properties of human skin, a different PLA should be tested, which is not partly transparent and has a different base colour.

Handyscan 3D

As a second scanner, the *Handyscan 3D* described in Section 3.1.1 was used. The setup to perform the scans is displayed in Figure 3.14, and the scan was carried out with closed eyes because the scanner works with a laser. The scanner was manually guided around the face for three minutes at a roughly constant distance of 30 cm, whereby the distance was partially reduced to generate more detailed images. In addition, photos from different perspectives were captured with an SLR camera to obtain the person's surface texture.

The resulting point cloud was captured with the scanner, and the photos were loaded into *VXelements* described in Section 3.1.4, which meshed the point cloud and generated the surface texture. The meshed file was exported as an *object* (OBJ) file, a format storing the three-dimensional information, the colour and texture of an object [107]. The model was also saved as an STL file, which does not contain textural information. The OBJ model is displayed in Figure 3.38 (left and second from left).

To further process the model, the STL file was loaded into *Meshmixer*, scaled with `Units` to its original size and aligned with `Transform` and `Align`. As with the other model, the `Plane Cut` was used to cut off the areas not belonging to the face to provide an even support surface.
As it can be seen on the left and second from the left picture in Figure 3.38, the surface has some holes, which were selected with `Select`, removed and filled with `Smooth Fill`. Additionally, the surface's irregularities were smoothed with the `Smooth Brush`, and it was made waterproof with `Make Solid Model` to ensure the surface has no additional holes.
The file was exported as STL, displayed in Figure 3.38 (second from right and right). However, this scan was not used due to its lower quality than the others explained in the following.

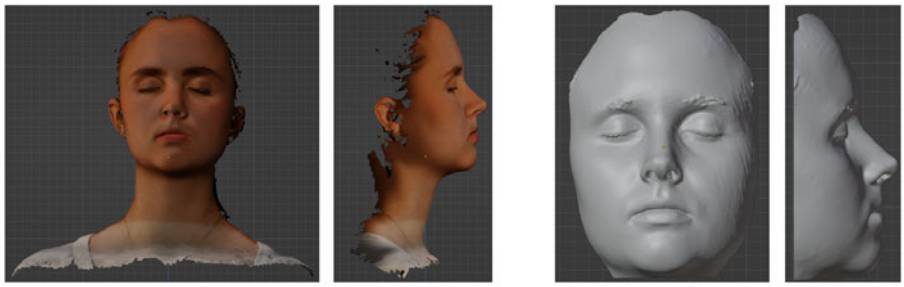

Figure 3.38: Frontal view (left) and side view (second from left) of the *Handyscan 3D* OBJ file before further processing and frontal view (second from right) and side view (right) of the STL file after processing with *Meshmixer*

Artec EVA 3D

Another utilised structured light scanner was the *Artec EVA 3D* described in Section 3.1.3, which captured the user's face with the setting displayed in Figure 3.14. To scan the face, the user had to close his eyes while the scanner was guided around the face for about two minutes at a constant distance of 30 cm, which was partly reduced to record certain areas of the face like the eyes or the lips in more detail. In contrast to the *Handyscan 3D*, no additional images were taken as this scanner directly captures the object's texture.
The scans were loaded into *Artec Studio Professional* (see Section 3.1.4) to convert the captured data into a mesh being exported with texture as OBJ and without as an STL file. The corresponding images are displayed in Figure 3.39.

As with the other scans, the STL file was loaded into *Meshmixer* to process the mesh further. The measuring units were adapted with `Units` to scale the mesh to its original size, and the scan was aligned to the displayed XY-level with `Transform` and `Align`. With `Plane Cut`, the areas not belonging to the face were cut off, and the backside was straightened to provide an even support surface.

Additionally, the model was made waterproof with `Make Solid` to get a filled model, which was to print. In contrast to the previous scans, this model was not smoothed as the surface was already even, and no holes had to be filled. We tried to minimise the mesh processing to keep the face's geometry untouched. The processed model was exported as an STL file (see the second from right and right image in Figure 3.39).

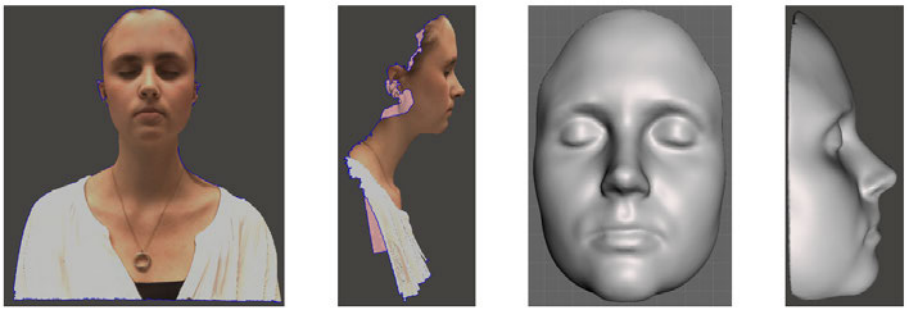

Figure 3.39: Frontal view (left) and side view (second from left) of the *Artec EVA 3D* OBJ file before further processing and frontal view (second from right) and side view (right) of the STL file after processing with *Meshmixer*

The model was sliced the same way as the one created with the *Shining 3D EinScan-SP*, resulting in layers from 0.07 to 0.25 mm thickness. Its resulting G-code was forwarded to the *Original Prusa i3 MK3S+*, which printed the model in ten hours with a 0.2mm nozzle. For this model, a PLA in the colour "peanut" was used, a non-transparent filament with an optical brightness similar to the user's skin. The printed model is displayed in the left image in Figure 3.40.

This mask was presented without any additives to evaluate the PLA's suitability, and both Windows Hello devices recognised it as a face but could not be bypassed. However, enrolling the mask was possible. Therefore, the mask's facial features are better preserved as with the *Shining 3D EinScan-SP*. The mask was presented to the Google Pixel 4, and after a few moments, the device responded that it did not recognise the face. Regarding the recorded logging section, this mask achieves a 23 or a two as the Airbrush Faceauth Result, meaning the device did not detect a face. As usual, the log section ends with a `HARD_REJECT` and reports that a non-identified user is detected. Compared to the mask without additives from the *Shining 3D EinScan-SP*, this mask seems better as it did not get the error `REJECT_TOO_BRIGHT`. When presenting the mask to the Apple iPhones, they responded "not recognised" and remained locked. However, as they rejected the mask, they might recognise it as a face.

We examined the mask's appearance in NIR with *HedgeCam2* (see the second from the left image in Figure 3.40). Its base brightness is slightly darker than the user's skin (see the left image in Figure 3.40) but more skin-like than the previous PLA. Additionally, its reflectance is muted and, therefore, resembles more that of human skin (middle image in Figure 3.34). However, the facial features have to be enhanced as they provide the same brightness as the remaining surface; subsequently, they are barely visible.
When it comes to the dots projected from the Google Pixel 4 (see the image in the middle in Figure 3.40), it can be seen that the dots are reflected more strongly from the mask surface than from human skin (see the middle image in Figure 3.34). Again, the edges are too sharp, so the IR does not penetrate the surface.

When it comes to the point clouds sent out from the Apple iPhone 12, this is the first mask which is categorised as a face, and therefore, not only the flashlight is visible but also both point clouds (see the second from right and right image in Figure 3.40). As with the point cloud from the Google Pixel 4, here, the dots do not penetrate the mask as with the human skin (see the second from right and right image in Figure 3.34).

Concluding, the mask printed with a non-transparent PLA with a colour similar to the user's skin achieves better results than the previous PLA as all devices recognised the mask as a human face. However, as previously outlined, this mask should also be enhanced by colouring the facial features to make the mask more humanoid and distinctive.

Figure 3.40: Printed mask (left), the mask's appearance in *HedgeCam2* (second from left), the Google Pixel 4's point cloud (middle) and the Apple iPhone 12's point clouds (second from right and right)

The mask was enhanced with ten layers of liquid latex. As these layers will not be removed from the underlying PLA, ten layers should be enough to support the makeup and provide a more skin-like surface. Additionally, the mask was enhanced with makeup by covering the surface with foundation, drawing on the eyebrows with dark brown eyeshadow and the lips with red lipliner, glueing on lashes and applying baby powder to the mask's surface to make it less reflective. We also enhanced deeper areas like the nostrils, the crease, the cupid's bow and the lip's middle line by shadowing these areas with dark brown eyeshadow to create optical depth for the two-dimensional appearance. The left image in Figure 3.41 displays the enhanced mask.

We presented the enhanced mask to both Windows Hello devices, which recognised it as a face but could not be unlocked. However, the mask could be enrolled as a user for both devices. When this mask is enrolled, the devices could be unlocked with the enhanced mask created with the *Shining 3D EinScan-SP*. An explanation could be that the enhanced masks are more similar to the enhanced mask than to the user.
When presenting the mask to the Google Pixel 4, the device responds "not recognised" immediately, which is also reflected in the device's log messages. Here, the mask directly achieves a 21 (`HARD_REJECT`) as Airbrush Faceauth Result, and therefore, the mask is rejected as if a human unauthorised user tries to unlock the device.
As with the first mask created with the *Artec EVA 3D*, both Apple iPhones recognise a face but not an authorised user; subsequently, the mask is rejected. Summing up, this mask could also not bypass the devices under attack.

We evaluated the mask's appearance in NIR and the appearance of the emitted dot pattern. When it comes to the appearance in NIR (see the second from the left image in Figure 3.41), the mask looks more similar to the user's face (left image in Figure 3.34) than the previous masks. However, its surface is less reflective than human skin and, therefore, seems darker. The shaded areas helped create visible depth so the mask looked not as flat. However, the brightness of the eyebrows and lips are different, and it looks unnatural as it is visible that they are painted on the mask.
Regarding the point clouds, first of all, it can be seen that all three devices emit a point cloud. The dots sent from the Google Pixel 4 (see the image in the middle in Figure 3.41) appear brighter as they are more strongly reflected from the surface as from the user's face (middle image in Figure 3.34). The same goes for the point clouds emitted by the Apple iPhone 12 (see the second from right and the right image in Figure 3.41). However, as the Apple iPhone 12 emits a point cloud, the device categorises this mask as a face. The dot's edges are also sharp, as the NIR might not penetrate the mask's material.

Figure 3.41: Enhanced mask (left), the mask's appearance in *HedgeCam2* (second from left), the Google Pixel 4's point cloud (middle) and the Apple iPhone 12's point clouds (second from right and right)

Concluding, the mask's enhancement provides better results than the previous mask as it is directly detected by the Google Pixel 4 and rejected as a human unauthorised user. The makeup enhances the humanoid appearance as this mask is the most similar to the user's face. However, the mask's surface is not similar enough to human skin as it appears to be darker, and the NIR cannot penetrate the material. Therefore, other materials that resemble the optical characteristics of human skin must be investigated. Furthermore, drawing the eyebrows and lips looks unnatural; different techniques should be investigated to provide a more humanoid look.

In this course, further tests were carried out. The same products (foundation, powder, eyeshadow and lipliner) were applied to the user's face to adapt its appearance to the mask, which was then presented to the devices. It was observed that the Google Pixel 4 could no longer be unlocked, regardless of whether the user's eyes were open or closed. Additionally, the user could also not be enrolled when he wears makeup. The other devices under attack could be unlocked.

The applied makeup was removed gradually to investigate why the Google Pixel 4 could not be unlocked. First, the lip and eyebrow makeup was removed, but the device remained locked. Next, the foundation was removed from one eye, after which the device could be unlocked again with the eyes open but not with closed eyes. When the second eye was also freed from the foundation, the Google Pixel 4 could be unlocked with closed eyes.

We again applied foundation to the eye area and removed it from the nose to test whether the skin around the eyes should not be covered or if other features could provide the real skin. However, the device could be unlocked partly with open eyes but not closed ones.
The user's face was regarded in *HedgeCam2* to examine these results further. First, no differences between the areas with and without makeup could be detected. Therefore, in the second step, the flood illuminator was taped off to reduce the overexposure of the face. Then, the areas without makeup were lighter than those with makeup.
Subsequently, as seen in the strings mentioned in Section 3.3.2, the surface's brightness is crucial for face recognition. However, it is sufficient if the area around the eyes resembles the skin. Therefore, for the following masks, no foundation was applied anymore.

As mentioned, the PLA in peanut was more suitable for brightness. However, the NIR might not penetrate the material. Therefore, further materials had to be examined. It was researched that light with a wavelength over 800 nm provides a penetration depth of around five to ten millimetres, meaning it reaches the skin's subcutaneous tissues [108]. We further researched materials adapting the absorption properties of human skin concerning NIR. According to Dabrowska et al. [109], epoxy resin can be used to simulate the properties of human skin for IR applications. Therefore, another approach was to work with an SLA printer described in Section 3.1.3, which prints objects with resin.

Before printing the model, it was further processed to fit onto the printing plate by cutting the mask's backside in *Meshmixer* with `Plane Cut`. Additionally, the nostrils were extruded to provide more depth information by selecting the surfaces where the nostrils should be with the command `Select`, and these were extruded four millimetres inwards with `Edit -> Extrude -4 mm`. The further processed model was exported as an STL file, sliced with a layer height of 0.05 mm, and a lighting duration of six seconds was chosen to print the model. Supports were also added to prevent the model from ripping off the plate and to make it more stable. The sliced file was sent to the *Anycubic Photon Mono X 6k* and was printed with white resin for ten hours. The left image in Figure 3.42 displays the resulting mask.

The printed mask was presented without enhancement to the devices but could not unlock any of them. Both Windows Hello Devices did not recognise a face and remained locked. Subsequently, this mask achieved worse results than the previous mask. When presenting the mask to the Google Pixel 4, it responded after a few seconds that it did not recognise the user and remained locked. When examining the log messages, it can be seen that the device does not categorise this mask as a face as it achieves the Airbrush Faceauth Results 2 (REJECT_NO_FACE) and 23 (REJECT_FRAME_-SELECT), both meaning NOT_DETECTED, before the device executes the HARD_REJECT. Both Apple iPhones remained locked but recognised a face as they responded that they did not recognise the user.

Considering the mask's evaluation, only the surface was considered to rate the resin's effect. The mask was examined with *HedgeCam2* (see the second from the left image in Figure 3.42). Compared to the user's face (see the left image in figure 3.34), the surface is lighter than the user's skin, and therefore, it has a higher reflectance than human skin.
Regarding the point clouds, both the Google Pixel 4's (see the middle picture in Figure 3.42) and the Apple iPhone 12's (see the second from right and the right image in Figure 3.42) are barely visible as the NIR is transmitted through the surface. Subsequently, the devices might not detect the mask's geometrical shape.

In conclusion, this mask is not further enhanced or used as the resin provided no advantages compared to the PLA regarding the NIR penetrating the mask. Additionally, regarding the log messages, its results were worse than those of the previous masks.

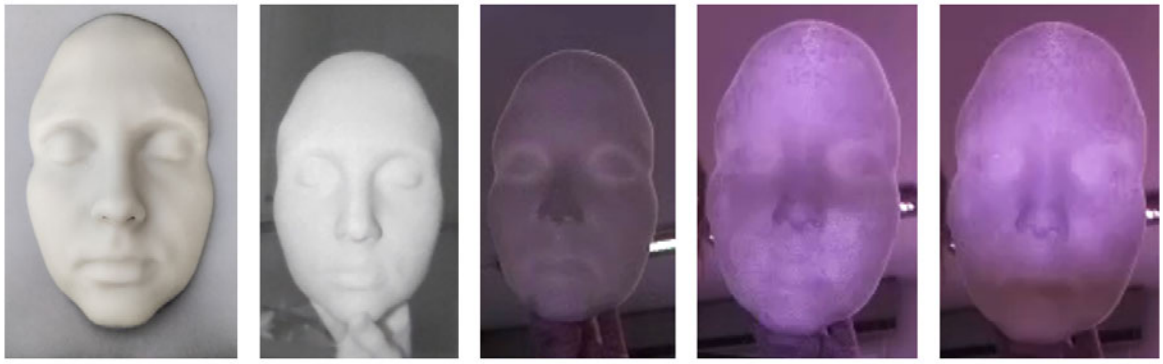

Figure 3.42: Resin mask (left), the mask's appearance in *HedgeCam2* (second from left), the Google Pixel 4's point cloud (middle) and the Apple iPhone 12's point clouds (second from right and right)

As another material adapting the properties of human skin in NIR, we worked with pigskin to cover the mask. Regarding it in *HedgeCam2*, the brightness matches those of human skin. Furthermore, the dots emitted by the devices penetrated this material. However, it was too stiff to cover the mask in such a way that the geometric information was preserved. Subsequently, pigskin was not further used.

To better mimic human skin, a mask purely made from latex was created. This mask should not have the PLA underneath but be stable enough to hold its shape. Hence, a negative was produced with *Autodesk Fusion 360* by importing the STL file and converting it to a solid body with `Convert Mesh to Solid body`. The mask was turned around with `Move 180 degrees` so the nose tip points downwards. To create a stable form, a rectangular sketch was drawn under the mask, slightly bigger than the mask and then extruded with `Extrude` until the extruded surface was aligned with the back of the mask. With `Combine`, both bodies were combined using the block as the target body and the mask as the tool body to cut the mask's shape out of the block. The resulting mould was exported as an STL file.

The mould was sliced with *PrusaSlicer*, resulting in layer thicknesses between 0.07 and 0.25 mm and printed with PLA with the *Original Prusa i3 MK3S+* for 18 hours. The finished mould was sprayed with release spray to make it easier for the liquid latex to come off later. The left image in Figure 3.43 shows the finished mould. It was filled with liquid latex in layers, each layer having to dry completely before the next could be applied, which took about two and a half weeks. The dried liquid latex could be removed from the negative, resulting in a positive mask (see the second from the left image in Figure 3.43).

This mask was then presented to the devices under attack but could not bypass them. Windows Hello devices recognise it as a face, and the mask could be enrolled as a user. However, it could not unlock the devices when the actual user is enrolled. The mask was presented to the Google Pixel 4, and after a few moments, the device responded that the user was not recognised. However, regarding the log messages, it can be seen that the face is first not detected (Airbrush Faceauth Result 2), but then, the mask achieves a one as a result; subsequently, the device recognises the mask as a person, but

the service reports that no features were found. Hence, the mask's dot score might resemble the legitimate user's, but the facial features are not visible enough for the final authentication. In the end, the device prints the HARD_REJECT and remains locked. Both Apple iPhones remained locked but responded that they did not recognise a legitimate user. Subsequently, they categorised the mask as a face.

Here, only the point clouds were evaluated as we searched for materials where the NIR light penetrates the surface. The point clouds from both devices' dot patterns look similar to those on human skin (see the middle, second from the right and right image in Figure 3.34). The Google Pixel 4's dot pattern is displayed in the middle image of Figure 3.43 whereas the Apple iPhone 12's point clouds are shown in the second from the right and right image of Figure 3.43. It can be seen that the NIR partly penetrates the mask, and therefore, the dots do not have sharp edges as with the previous masks. Additionally, the dots are less reflective than before. Therefore, this material seems more suitable to mimic human skin.

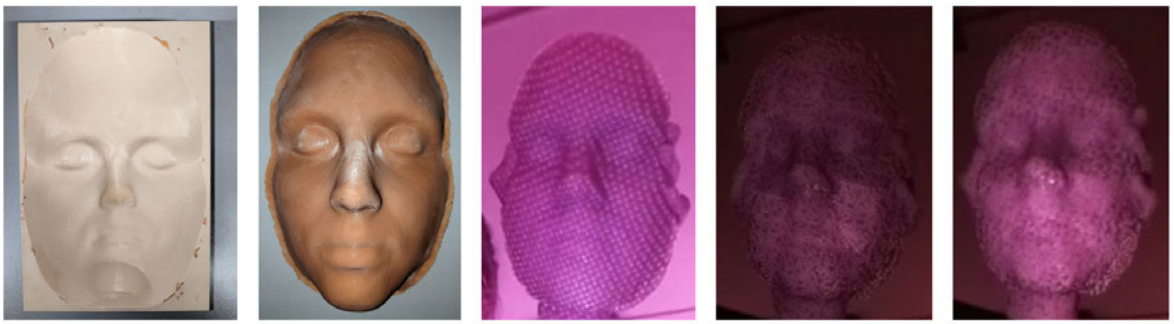

Figure 3.43: Printed mould (left), resulting mask consisting of liquid latex (second from left), the Google Pixel 4's point cloud (middle) and the Apple iPhone 12's point clouds (second from right and right)

In conclusion, this mask provides the best results as the Google Pixel recognises its dot pattern as that of a legitimate user. Therefore, creating a mask only out of liquid latex with no additional material underneath seems suitable to mimic the skin's texture but also provides a stable mask. However, the features must be further enhanced to be visible.

Since the dots on the masks emitted by both devices look like those on human skin, we focused on enhancing the facial features and only regarded the following mask's look in *HedgeCam2*. The eyebrows and lips were first drawn on with dark brown and red eyeshadow to make the features visible, as they might look different because of the underlying material. Additional lashes were then glued onto the surface. The left image in Figure 3.44 shows the enhanced mask consisting of liquid latex.

All devices rejected this mask. As with the previous masks, it could be enrolled in the Windows Hello Devices, and they recognise it as a human face. Additionally, the mask is not directly rejected, but the system displays that it has to ensure that it is the legitimate user and, in the end, rejects the mask. Regarding the Google Pixel 4, the device responded after a few moments that the user was not recognised. Furthermore, the log messages displayed that the mask's enhancement led to a deterioration in terms of facial recognition.

Now, the mask was directly rejected with a HARD_REJECT (Airbrush Faceauth Result 21) and was not recognised as a legitimate user. The mask was also presented to the Apple iPhones, but these could not be bypassed either. However, they might recognise a face as both showed a reaction and rejected the mask.

The mask was regarded in *HedgeCam2* (see the second image from the left in Figure 3.44) to evaluate the mask's enhancement. It can be seen that the facial features are more visible than before. However, the lips and eyebrows' brightness does not resemble those of humanoid features. The lips are also different as the characteristic darker line in the middle of the lips is not visible because the mask's lips are not as deep in this area. Furthermore, the lips seem flat, as the humanoid lips are lighter in the centre and darker on the outside. Only the lashes are similar.

To further enhance the mask and achieve better results, hydrogel was used. As mentioned in Section 2.2, the skin's appearance in NIR depends primarily on the water and Oxy Hemaglobin absorption. Therefore, we tried to increase the mask's water content to change the feature look in NIR. The hydrogel was applied all over the mask and the previously drawn features. However, this mask led to the same results as without the hydrogel.
With *HedgeCam2*, it could be examined that the mask's surface looks more glossy but darker; therefore, its brightness is more similar to human skin. In addition, the features also appear darker than before and, therefore, look more similar to human features.

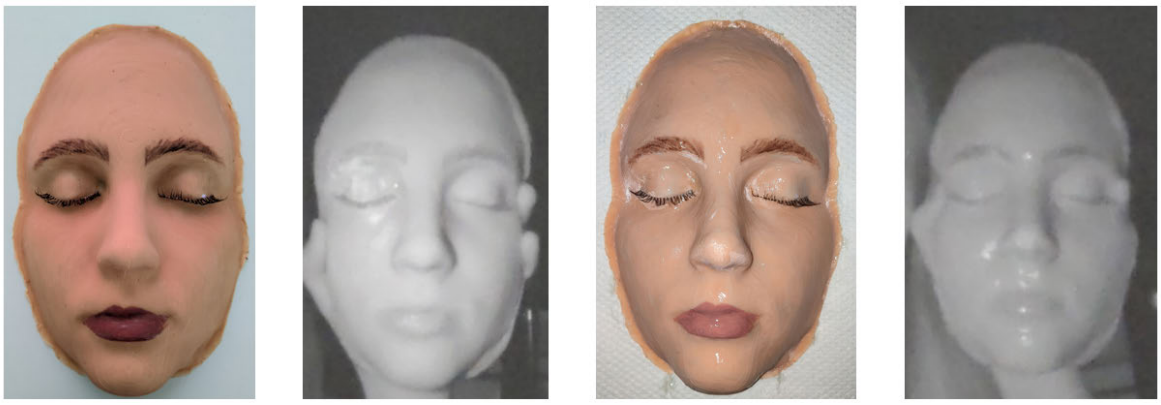

Figure 3.44: Enhanced liquid latex mask (left), its appearance in *HedgeCam2* (second from left), the enhanced liquid latex mask with hydrogel (second from right) and its appearance in *HedgeCam2* (right)

As previously mentioned, the creation of the facial features must be examined in more detail as the makeup application downgrades the mask's result and leads to a rejection. Especially the eyebrows and lips must be adapted. Since the application of hydrogel brought at least a visual improvement, it will be partially used further in the following.

The mask was cleaned to provide more realistic features, and the eyebrows were modelled with human hair. The hair was cut into small pieces and glued onto the surface with the help of tweezers and lash glue to make the eyebrows look more realistic than with drawn hair strokes. The lips were again drawn onto the mask with eyeshadow, but the middle line was darkened, and hydrogel was applied as the overall brightness of the lips was previously more similar to human lips with hydrogel.

The enhanced mask is displayed in the left image in Figure 3.45. This mask achieves the same results for all devices as the previous masks. According to the log messages, it is also directly rejected with a HARD_REJECT, and the device did not recognise it as a legitimate user.

When it comes to the mask's appearance in *HedgeCam2* displayed in the second from the left image in Figure 3.45, the eyebrows consisting of human hair look similar to those of the legitimate user as the overall brightness matches (see the left image in Figure 3.34). The lips are the most similar to human lips; however, they still look very different from the user's lips in terms of brightness.

The added facial features were removed step by step to examine which features disturb face recognition. Only the eyebrows consisting of human hair did not disturb the face recognition as all other added features had to be removed until the mask achieved the Airbrush Faceauth Result one.
The mask is shown in the image second from right in Figure 3.45. Its appearance in *HedgeCam2* is displayed in the right image in Figure 3.45. This mask is again categorised as a legitimate user, but the device still found no features. Here, the eyebrows and the shape of the eyes might not be enough. As mentioned in Section 3.3.1, more features are required when the eyes are closed. Therefore, we considered investigating further materials to mimic the user's lips and create a mask with opened eyes to provide more features.

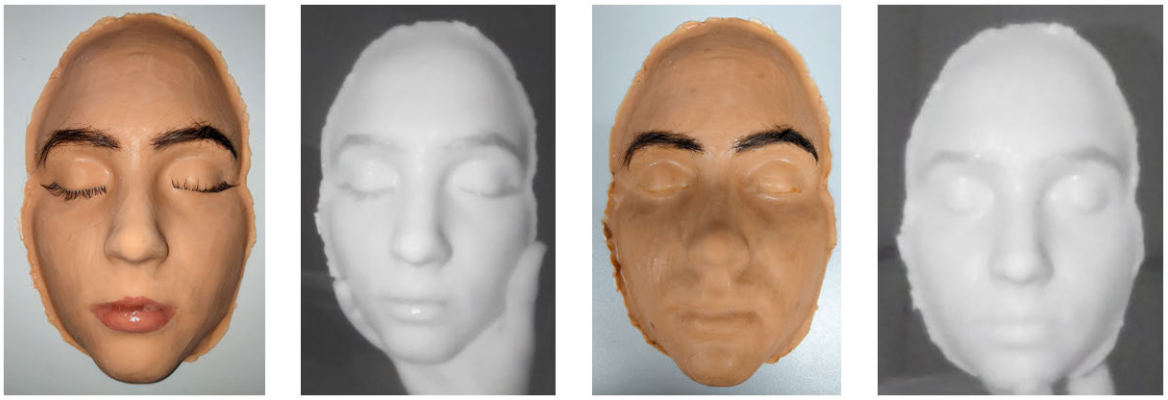

Figure 3.45: Enhanced liquid latex mask with eyebrows consisting of human hair, drawn lips enhanced with hydrogel and applied lashes (left), its appearance in *HedgeCam2* (second from left), the liquid latex mask where only eyebrows consisting of human hair were applied (second from right) and its appearance in *HedgeCam2* (right)

The masks based on the scan from the *Artec EVA 3D* provide better results than those based on the *Shining EinScan-SP* as all devices recognised them as a face. Additionally, they could be enrolled as a user in the Windows Hello devices. One reason could be the utilised PLA, whose brightness is similar to that of human skin in NIR. Another reason might be that the face's geometry was better preserved as the scan was not smoothed.
Liquid latex is more suitable as mask material than regular PLA as the NIR dots can penetrate the surface, and therefore, the mask is categorised as a legitimate user. Furthermore, enhancing the mask with additives provides better results than presenting the mask plain. Humanoid additives like human hair for modelling eyebrows are more suitable than artificial additives like makeup. However, enhancers that look more humanoid must be researched to further adapt the lip's appearance in NIR. In addition, a scan with opened eyes should be considered to provide more distinctive features for face recognition, as the eyes are the most crucial part of the face.

SfM

SfM, described in Section 3.1.3, was used as another technique to record the face and to capture the opened eyes. The user was placed on a chair with enough space around it and a white background with black dots as an orientation for meshing the point cloud. The user had to sit still with opened eyes and to ensure that the view was permanently directed to the same spot, another orientation point was placed opposite the user. An assistant guided the SLR camera around the user three times, once each at the user's eye level, from a slight top and a bottom view. In each round, about 30 overlapping photos were taken, which were then imported into *Meshlab Professional* (see Section 3.1.4) to create a textured mesh. The left image in Figure 3.46 shows the resulting mesh with its texture. Here, the opened eyes are visible; however, regarding the mesh without the texture on top (the right image of Figure 3.46), it can be seen that the mesh is highly inaccurate, and the eye's shape is also not visible. Therefore, another method was tested as this model was unsuitable.

Figure 3.46: Model with texture generated with SfM (left) and model without texture captured with SfM (right)

As another method, the setup described in Section 3.1.3 was used with 13 SLR cameras capturing the user simultaneously. The photos were also imported into *Meshlab Professional* and were meshed together. The resulting mesh with its texture is displayed in the left image in Figure 3.47. As this model was more detailed than the previous ones, it was further processed in *Blender* by extruding the surfaces by two millimetres on the backside to provide a stable mask. The area of interest was cut out by selecting the areas not belonging to the face and removing them with the command `Remove Faces and Edges`. The resulting mesh without texture from the front and the side view can be seen in the second from the left and second from the right image in Figure 3.47.

This mask served as a fundament for the negative mould as the mask consisting only of liquid latex provided the best results. We worked with *Blender* and used the `path` tool to trace the mask's outer edges and extrude them. A rectangle was drawn around the mask, which is about 5 millimetres longer on each side than the mask and extruded until its edges were about the same height as the mask. The mask was then joined with the edges of the box, and the resulting mould was solidified to create a stable cast without holes.
The mould was exported as an STL file and forwarded to *PrusaSlicer* for slicing with the same settings described earlier, resulting in layer thicknesses between 0.07 mm and 0.25 mm. As with the other masks, it was printed with the *Original Prusa i3 MK3S+* with PLA. The resulting mould can be seen in the right image in Figure 3.47.

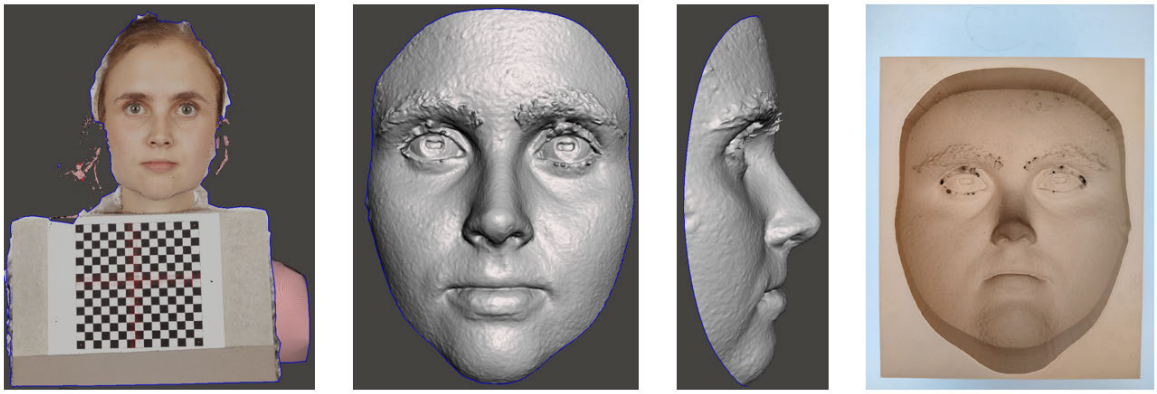

Figure 3.47: Model with texture captured with SfM (left), the front of the processed model as STL file generated with SfM (second from left), the side view of the model as STL file captured with SfM (second from right) and the resulting mould printed with PLA (right)

The mould was coated with release spray so that the liquid latex was easier to remove from the mould later, and it was poured in layer by layer, where each layer had to be fully cured before the next was applied. The liquid latex was then carefully released from the mould, and powder was applied to the freed areas to prevent them from rolling up or sticking to the mould again. The resulting mask consisting only of liquid latex is displayed in the left image of Figure 3.48.

When presenting the mask without additives, all devices recognised it as a face; however, it could not unlock any of the devices. This mask could also be enrolled as a legitimate user for both Windows Hello devices and is recognised as a face as the devices respond that they had to make sure it is the correct user. We examined the log messages the Google Pixel 4 provided, and it achieved the best results as it got the Airbrush Faceauth Result multiple times in a single session before the device reported the HARD_REJECT. Both Apple iPhones recognised the mask as a face but could not be unlocked.
As the devices' point clouds on the liquid latex mask were evaluated previously, here again, only the mask's appearance in *HedgeCam2* was considered. Regarding the mask's appearance in NIR (see the second from the left image in Figure 3.48), the non-enhanced mask looks more detailed than the previous plain masks as the facial features are partly visible, and the eye region is more distinctive than with closed eyes. Subsequently, the Google Pixel 4 directly detects more humanoid features, which makes recognising the user easier, as the characteristics are more distinctive.

We searched for a makeup substitute to draw the facial features to enhance the mask further. We tested acrylic paint, but many pigments are too reflective in NIR and appear white. Therefore, we tried several colours to search for less reflective ones that appear darker in NIR and analysed the pigments with *HedgeCam2*. As the devices do not consider the RGB appearance of the face, the pigment's colour in the visible spectrum was irrelevant; subsequently, only the pigment's appearance in NIR was examined.

The colour being the closest in NIR to the brightness of the facial features was the "Citadel base" in the colour "Kantor Blue", which was mixed with the "Vallejo Game Air" in "Dead White" and "Pale Flesh" to mimic the brightness of the vitreous and the iris. The "Citadel base" in the colour "Rhinox Hide" was used to shade the pupil as it was the darkest pigment in NIR. Furthermore, the eye's reflectance was mimicked by using the "Vallejo Game Air" in "Dead White". The eyebrows were

again modelled with human hair, which looked more natural than drawing lines onto the surface. In addition, the eye circles and laugh lines were deepened to generate optical depth. Here, no acrylic paint was used as they are more difficult to blend, and the exact brightness of these areas might not be required. The resulting mask is shown in the second from the right image of Figure 3.48.

This enhanced mask was presented to the devices but could not unlock them and provided the same results for both Windows Hello devices and Apple iPhones. However, looking at the Google Pixel 4, it took longer for the device to report that the face was not recognised. When examining the log messages, this enhanced mask is better categorised as a legitimate user than the mask without additives, as it achieved the Airbrush Faceauth Result one in a single session even more often. In contrast to the applied makeup, the acrylic paint did not disturb the face recognition but increased the recognition results.

The mask's appearance in NIR was evaluated with *HedgeCam2* and is displayed in the right image of Figure 3.48. It can be seen that this mask is the most similar to the user's face (see the second from the left image of Figure 3.34). The mask consists only of liquid latex, so the overall brightness is similar to human skin's. Furthermore, the human hair eyebrows also appear similar regarding their brightness. However, when it comes to the eyes, the mask's eyes look dilated, and it seems as if further details like the lashes or the shadows provided by the upper lid are missing. Additionally, due to its wetness, the humanoid eye looks more shiny and not as flat. The lips could be further enhanced to make them more visible.

Figure 3.48: Resulting mask generated with SfM consisting only of liquid latex (left), its appearance in *HedgeCam2* (second from left), the mask with enhanced facial features with acrylic paint (second from right) and its appearance in *HedgeCam2*

The eyes were adapted to enhance the mask by drawing more details and creating optical depth. Here, the upper and lower lash lines and the lacrimal sac were coloured with the "Kantor Blue", but this time, less "Dead White" and "Pale Flesh" were mixed in to make it look darker than the iris. Additionally, the lashes were coloured with "Rhinox Hide" to appear as dark as the pupil. Contact lenses consisting of hydrogel were considered to mimic the eye's shine and were trimmed to fit onto the drawn pupil. In addition, the lips were coloured with a mixture of "Kantor Blue", "Pale Flesh", and "Dead White", whereby the middle line and the outer corners were coloured darker. The colour was thinned with water for the centre of the lips to appear lighter and mimic the user's lip brightness. The finished mask is displayed in the left image of Figure 3.49.

The further enhanced mask was presented to the devices under attack. The results for both Windows Hello devices and the Apple iPhones remained the same. However, this mask could unlock the Google Pixel 4. Regarding its log messages, it can be seen that a few adjustments are required until the mask is recognised as a legitimate user. The device sends out a combination of the Airbrush Faceauth Result one and other results concerning the face's position, like the Airbrush Faceauth Result seven (REJECT_CLIPPED_RIGHT). When we moved the mask to the left, the device sent a Faceauth Result one with the following Trustzone app messages explained in Section 3.3.2 and unlocked the device.

Furthermore, the mask's appearance in NIR was examined (see the right image of Figure 3.49). Here, it can be seen that the further enhancements described above made the mask's eyes look alive as they created depth, and the pupil's shine is visible. In addition, the lips are now detectable as the line is darker, and the brightness of the lips stands out slightly from that of the general surface so that they resemble natural lips. Due to the darkened edges, they seem more realistic as it creates optical depth. Overall, the brightnesses of the facial features match those of the legitimised user (see the second from the left image of Figure 3.34).

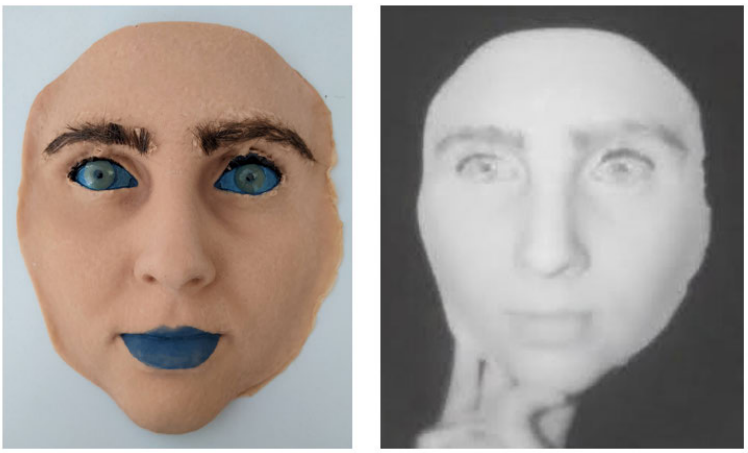

Figure 3.49: Resulting enhanced mask generated with SfM consisting only of liquid latex (left) and its appearance in *HedgeCam2* (right)

The artefact creation for devices with face recognition using additional infrared sensors requires more effort than for devices using only their RGB camera. A three-dimensional model must be created to generate an artefact resembling the user's face. In addition, the facial depth and the face's infrared appearance must be considered.
The model captured with SfM provides the best results, as the resulting mask could unlock the Google Pixel 4. Compared to the scans captured with the *Shining 3D EinScan-SP* and the *Artec EVA 3D*, this model is the most detailed one as it captured the opened eyes, eyebrows and nostrils. Furthermore, as previously mentioned, the dot pattern on the mask consisting only of liquid latex is the most similar to the dot pattern on the user's face as the NIR can penetrate the material. In addition, enhancing the mask provides better results as the devices can detect facial features. In contrast, acrylic paint is more suitable than makeup as it can be adapted to match the feature's appearance. Furthermore, contact lenses enhance the eye's appearance by making the drawn iris glossy. Hence, the shine of a human eye can be resembled.

However, as mentioned, Windows Hello devices and Apple iPhones cannot be fooled with the described experiments. The Windows Hello devices might remain locked as they take a two-dimensional face image and do not consider depth information. However, the created masks are primarily modelled to give the mask depth information. Additionally, they are adapted to the requirements of the Google Pixel 4; therefore, the mask's two-dimensional image differs from the user's as the goal was to adapt the brightness of the facial features and not to create an exact copy of the user's face.

Both Apple iPhones might remain locked as they provide anti-spoofing techniques in the form of a neural network trained with face masks. Hence, the devices may recognise the mask as a spoofing attempt due to certain unknown factors and remain locked even if the dot pattern and the facial feature's brightness are similar to those of the legitimate user. In addition, when trying to enrol the mask, FaceID directly reports that it is not available and shuts down.
Furthermore, Apple's TrueDepth camera offers a resolution of seven megapixels (3072×2304 pixels), about five times the resolution of the Google Pixel 4. Hence, this camera captures more details of the user's face. Subsequently, the mask might look too different from the user's face.
Moreover, Apple also takes a two-dimensional image of the user's NIR appearance. As with both Windows Hello devices, an NIR picture of the mask might not be similar enough to the user's.
Additionally, the number of attempts to unlock the device with a face before a password is required is more strict than that of the Google Pixel 4. The mask can only be presented twice before the device requires a password. Therefore, it is harder to test the mask as it has to be directly positioned perfectly, whereas the Google Pixel 4 allows to adapt the positioning before the mask is rejected.

4 Conclusion and Future Work

In this thesis, the different techniques for face authentication on electronic devices were analysed in Chapter 2. The structure of biometric systems and the types of face authentication are outlined, which are crucial for examining the research questions. Furthermore, the approaches to bypass the devices under attack are discussed in Chapter 3. The attacks are described and evaluated, the sensor's limitations are researched, and the underlying software is analysed.

This study aimed to find answers to several research questions described in Section 1.2. First, it should be examined how state-of-the-art facial recognition can be circumvented on electronic devices. Therefore, approaches considering bypassing the devices using their RGB camera for face authentication as well as the creation of two-dimensional artefacts are described in Section 3.2. Hence, an image of the user was printed on regular paper with a printer, which could bypass the targeted devices. The background was cut off to create more reliable artefacts, as some devices might recognise the non-matching background. The artefacts had to be presented with indirect light or daylight, and the angle and distance to the device were crucial as they must resemble the position a regular user would present his face to the device. Furthermore, it made no difference if the artefacts were created using a user's picture with or without glasses.

In Section 3.3.3, two-dimensional attacks with artefacts created from infrared images of the user were carried out to circumvent the Windows Hello devices, which only require the user's appearance in NIR. For the capturing, the IR camera from the Raspberry Pi achieved the best results as the facial features of these artefacts were visible in NIR, and the skin area was not too reflective. Therefore, they are the most similar to the appearance of the user's face. However, none of these artefacts could bypass the Windows Hello devices, although the focus of the work was not on these devices. Moreover, three-dimensional masks were created (see Section 3.3.4 and 3.3.5) to bypass the devices with face recognition which requires the user's appearance in NIR and the facial depth data. Face casts are not suitable for this task of mask creation as the results are too inaccurate. SfM is the most suitable technique to capture the user's face as this method provides the most detailed surface, and opened eyes could be reconstructed. In addition, masks consisting only of liquid latex achieved the best results as they simulate the surface properties of human skin. Subsequently, the dot pattern on the mask's surface emitted by the Google Pixel 4 and both Apple iPhones is similar to that on human skin. Acrylic paint, human hair and contact lenses enhanced the mask's facial features best. Therefore, the mask's two-dimensional NIR appearance was similar to the legitimate user's. This mask achieved the best results as it unlocked the Google Pixel 4.

Concluding, for the first research question, it was found that devices with face recognition relying on the visible spectrum can be bypassed with two-dimensional artefacts printed on regular paper. Regarding devices whose face recognition is based on the near-infrared spectrum, each has its own requirements. While all systems are based on NIR, each device has a different focus. Accordingly, the Google Pixel 4 requires a three-dimensional mask of latex milk as a skin substitute and facial features generated from hair, acrylic paint and contact lenses. Masks bypassing the Apple iPhones and the Windows Hello devices could not be generated with the presented approaches.

The sensor's limitations regarding the image colouring and resolution, the face positioning and the required parts of the face should also be analysed. When it comes to devices performing face recognition in the visible spectrum (see Section 3.2), none considered the image colouring; subsequently, only the distances and angles between the facial features are important.
Next, the image resolution was varied, and it stated that, except for the Samsung Galaxy S20, all devices under attack working in the visible spectrum could be bypassed with 100 dpi images. Hence, scaling a passport photo to create an artefact or capturing the user's face from a distance without his participation can be possible as the sensor's boundaries are loose. However, higher-resolution images (around 300 dpi) achieve better results as fewer attempts are needed to bypass the devices.
Furthermore, a front-faced image of the user is unnecessary as all devices performing face recognition with an RGB camera could be unlocked with artefacts consisting of a turned head, but more than the user's profile is needed. Hence, capturing the user's face without his knowledge is possible as he does not have to look right into the camera. However, if the attention check is enabled, a front-faced image is required; subsequently, it is advisable to create artefacts with a frontal image of the user.
It could be stated that no device under attack working in the visible spectrum required a complete head; therefore, the face's outline is not considered a facial feature. In addition, it turned out that these devices could be unlocked with artefacts consisting of only two facial features, one of which must be the eyes; thus, only the eyes and eyebrows or the eyes and nose are needed to unlock the DUTs performing face recognition with an RGB camera. Subsequently, the eyes are necessary for face authentication; otherwise, no face is detected.

When examining the sensor's limitations for devices performing NIR face recognition (see Section 3.3.1), no device needs the user's appearance in the visible spectrum; subsequently, only the NIR appearance is considered.
Regarding the head's position, all devices can be unlocked when the user's head is turned up to 10 degrees, but for Apple iPhones, the eyes must be directed on the screen. However, the masks were always presented frontally to the devices to achieve the best results.
Furthermore, considering the necessary parts of the face, the requirements depend on the specific device under attack. Regarding the Google Pixel 4, only the eyes and eyebrows are needed when the user's eyes are open, whereas the mouth is also necessary with closed eyes or glasses. In contrast to the Google Pixel 4, the Apple iPhones can only be unlocked with open eyes. Besides the eyes, the nose and either the eyebrows or the mouth must be visible. Windows Hello devices do not require opened eyes; besides the eye region, the nose must be visible to unlock the device.
In addition, further tests provided clues about Apple's anti-spoofing technique. In contrast to the Google Pixel 4, these devices could also be unlocked with applied foundation. Accordingly, Apple's mask recognition is probably not based on the skin's appearance, as it appears darker with a foundation in NIR.

In conclusion, devices with face recognition based on the visible spectrum have nearly the same limitations as it seems as no device needs a coloured image. Most devices can be unlocked with images providing a minimum resolution of 100 dpi, and no frontal picture is required. In addition, these devices require only two features, one of which must be the eye region. Devices performing face recognition in the NIR spectrum only rely on the NIR appearance, and the user can turn his head to a maximum of 10 degrees. Regarding the required parts of the face, similar to the devices with no additional sensors, all devices require the eye area, although these can be closed on the Google Pixel and Windows Hello devices. In addition, most devices also need the nose to authenticate a user.

Lastly, it should be examined if the underlying face recognition software provides clues on how face recognition works. Hence, the service and firmware responsible for face recognition were reverse-engineered, as outlined in Section 3.3.2. It could be stated that the face recognition service and firmware of the Google Pixel 4 provided more clues about its operating principle than the Apple iPhones. Considering the Google Pixel 4's components included in the face authentication, the firmware offered evidence that three chips are used: the main processor, Google's Neural Core and the Titan M security chip. As the Neural Core is involved, Google might also work with neural networks or machine learning to determine the results of the different features.
In addition, strings could be found indicating that Google also implements a type of anti-spoofing technique which relies on the appearance of the dot pattern on the skin and their brightness. Furthermore, an array of strings outlined the device's reject reasons, indicating that an object is rejected because of its positioning, because of the skin or because it is categorised as a spoofing attempt. Those appear in the system's log messages as an "Airbush Faceauth Result".
Regarding the Apple iPhones, no clues could be found in the firmware or the service concerning its face recognition operation mode. In addition, there were no strings indicating the foundation of Apple's anti-spoofing technique.

Concluding, evidence could be found in the software of the Google Pixel 4 that its face recognition relies on the skin's appearance in NIR, the facial depth and the dot pattern on the skin. In addition, Google also implements a type of anti-spoofing based on these priorities. Concerning the Apple iPhones, no evidence could be found in the underlying software.

In general, generating two-dimensional artefacts is relatively fast, requiring less than 15 minutes to print a photo of the user's face. Generating three-dimensional masks is more time-consuming as the liquid latex mask requires about two weeks to cure. In addition, the mask enhancement takes about a day. However, compared to brute force attacks, both methods are faster as these attacks might take years [9]. Therefore, unlocking devices for reading out data should preferably be done with biometric features. Regarding the Google Pixel 4, the mask should be created in advance as it cannot be ready within a day.

Further approaches could be examined to create artefacts bypassing the Windows Hello devices and the Apple iPhones. Concerning Windows Hello, only the user's two-dimensional appearance is considered. Subsequently, a three-dimensional mask providing depth information is not required. Therefore, two-dimensional approaches could be more successful. Instead of printing the user's infrared image, printing an RGB image with ink visible in the NIR spectrum might be possible. This printed artefact might look more similar to the user than the generated mask as the facial features do not have to be painted on the mask and, subsequently, look exactly like the user's features. In addition, they might be better resembled than those of the artefact consisting of an NIR image.

Regarding the Apple iPhones, another approach could be disturbing the neural network. As a neural network acts like a black box, meaning its operation mode remains unknown, and Apple's underlying software provides no clues about the system's priorities, it is not easy to reconstruct the foundation of Apple's face recognition. In addition, as the neural network is trained with masks, it will categorise each attempt with a three-dimensional printed mask as a spoof.

Therefore, disturbing the neural network in a way that does not categorise the mask as a spoofing attempt but as a legitimate user might work. For example, stickers could be glued onto the surface to distract the device. As the focus might not lie on the user's skin, other trigger points have to be examined and could then be exploited.

Moreover, the capturing technique can be evaluated further to create the user's three-dimensional model. Since an exact scan of the suspect cannot be performed later, other approaches must be investigated to create a model. As already described in Section 1.2, § 81b of the German Code of Criminal Procedure covers the identification measures on the suspect, whereby photographs are also taken. Hence, it should be evaluated if those images could be used to create the face model. For example, there is an add-on for *Blender* named *FaceBuilder*, which generates a three-dimensional model of a head based on photographs [110]. This course would have to evaluate how accurate models based only on photos are concerning the distances between the features.

Bibliography

[1] U.S. National Library of Medicine, *Visible Proofs: Forensic Views of the Body: Galleries: Biographies: Alphonse Bertillon (1853–1914)*. [Online]. Available: https://www.nlm.nih.gov/exhibition/visibleproofs/galleries/biographies/bertillon.html (visited on 06/29/2023).

[2] Bundesamt für Sicherheit in der Informationstechnik, *Grundsätzliche Funktionsweise biometrischer Verfahren*. [Online]. Available: https://www.bsi.bund.de/DE/Themen/Verbraucherinnen-und-Verbraucher/Informationen-und-Empfehlungen/Technologien_sicher_gestalten/Biometrie/AllgemeineEinfuehrung/einfuehrung.html?nn=452592 (visited on 06/29/2023).

[3] R. S. Lacruz *et al.*, "The evolutionary history of the human face", *Nature Ecology & Evolution*, vol. 3, pp. 726–736, Apr. 15, 2019.

[4] H. Baqeel and S. Saeed, "Face detection authentication on smartphones: End users usability assessment experiences", Imam Abdulrahman Bin Faisal University, 2019.

[5] D. Labudde and M. Spranger, *Forensik in der digitalen Welt, Moderne Methoden der forensischen Fallarbeit in der digitalen und digitalisierten realen Welt*. Berlin: Springer Spektrum, 2017.

[6] T. Bourlai and B. Cukic, "Multi-spectral face recognition: Identification of people in difficult environments", Lane Department of Computer Science and Electronical Engineering, research rep., 2012.

[7] C. Hoogsteden and P. Cross, "Public access to gps - government duty, economic rationality or international philanthropy?", *CISM journal*, vol. 46, no. 1, pp. 41–53, 1992.

[8] S. Marcel, J. Fierrez, and N. Evans, *Handbook of Biometric Anti-Spoofing - Presentation Attack Detection and Vulnerability Assessment*, Third Edition. Springer Nature Singapore, 2023.

[9] P. Schmitz and S. Luber, "Was ist ein Brute-Force-Angriff?", *Security Insider*, Jan. 2018. [Online]. Available: https://www.security-insider.de/was-ist-ein-brute-force-angriff-a-677192/ (visited on 06/09/2023).

[10] D. M. Grzesiek, "Die entschlüsselung von smartphones gegen den willen des beschuldigten", *StV-Strafverteidiger*, no. 3, pp. 117–124, Mar. 2021.

[11] A. K. Jain, P. Flynn, and A. A. Ross, *Handbook Of Biometrics*. Springer, 2008.

[12] D. F. Antwerpes and S. A. Bröse, *Anthropometrie*, Nov. 22, 2022. [Online]. Available: https://flexikon.doccheck.com/de/Anthropometrie (visited on 07/25/2023).

[13] Y. Levalle, "Bypassing biometric systems with 3d printing and 'enhanced' grease attacks", Dreamlab Technologies, 2020.

[14] T. Kanade, "Picture processing system by computer complex and recognition of human faces", Ph.D. dissertation, Department of Information Science Kyoto University, Nov. 1973.

[15] S. Z. Li and A. K. Jain, *Handbook of Face Recognition*, 2nd ed. London: Springer, 2011.

[16] Kaspersky, *Die Gesichtserkennung – Definition und Erläuterung*, Apr. 2023. [Online]. Available: https://www.kaspersky.de/resource-center/definitions/what-is-facial-recognition (visited on 06/29/2023).

[17] A. K. Jain, A. Ross, and S. Prabhakar, "An introduction to biometric recognition", *Transactions on Circuits and Systems for Video Technology*, vol. 14, no. 1, Jan. 2004.

[18] G. Boesch, "What is Pattern Recognition? A Gentle Introduction", *viso.ai*, Feb. 2023. [Online]. Available: https://viso.ai/deep-learning/pattern-recognition/ (visited on 08/15/2023).

[19] Hochschule Bonn-Rhein-Sieg, *BIOLAB*. [Online]. Available: https://h-brs.de/de/isf/biolab (visited on 07/26/2023).

[20] N. A. Lal, S. Prasad, and M. Farik, "A reivew of authentication methods", *International Journal of Scientific & Technology Research*, vol. 5, 11 Nov. 2016.

[21] N. Halbe, "Analysis of biometric features for user authentication on mobile devices for forensic purposes", University of applied science Mittweida, 2021.

[22] B. Haluschak, "Sperren sie ihr rechenzentrum ab", *Computerwoche*, Mar. 10, 2009. [Online]. Available: https://www.computerwoche.de/a/sperren-sie-ihr-rechenzentrum-ab,1889056 (visited on 07/26/2023).

[23] D. Maltoni, D. Maio, A. K. Jain, and S. Prabhakar, *Handbook Of Fingerprint Recognition*, 2nd ed. London: Springer, 2009.

[24] R. Hilgers, N. Heussen, and S. Stanzel, "Korrelationskoeffizient", in *Lexikon der Medizinischen Laboratoriumsdiagnostik*, A. M. Gressner and T. Arndt, Eds. Berlin, Heidelberg: Springer, 2018.

[25] D. A. Reid, S. Samangooei, C. Chen, M. S. Nixon, and A. Ross, "Soft biometrics for surveillance: An overview", in *Handbook of Statistics*. 2013, vol. 31, ch. 13, pp. 327–352.

[26] R. M. Devi *et al.*, "Retina biometrics for personal authentication", in *Machine Learning for Biometrics*. 2022, ch. 5, pp. 87–104.

[27] P. J. Phillips, P. Grother, R. J. Micheals, D. M. Blackburn, E. Tabassi, and M. Bone, "Face recognition vendor test 2002 overview and summary", National Institute of Standards and Technology, Tech. Rep., Mar. 2003.

[28] L. Li, X. Mu, S. Li, and H. Peng, "A review of face recognition technology", *IEEE*, vol. 8, Aug. 2020.

[29] S. Z. Li and J. Wu, "Face detection", in *Handbook of Face Recognition*, 2nd ed. London: Springer, 2011.

[30] G. Amato, C. Gennaro, F. Falchi, and C. Vairo, "A comparison and of face and verification with facial and landmarks and deep and features", Institute of Information Science and Technologies of the National Research Council of Italy, Tech. Rep., Dec. 19, 2019.

[31] X. Ding and L. Wang, "Facial landmark localization", in *Handbook of Face Recognition*, 2nd ed. London: Springer, 2011.

[32] GIMP, *GIMP - gnu image manipulation program*. [Online]. Available: https://www.gimp.org/ (visited on 06/12/2023).

[33] J. Landwehr, "Machine Learning: Was bedeutet Accuracy und Precision?", *IT-Talents*, May 2020. [Online]. Available: https://it-talents.de/it-wissen/machine-learning-accuracy-und-precision/ (visited on 07/26/2023).

[34] T. Bourlai, *Face Recognition Across the Imaging Spectrum*. Switzerland: Springer, 2016.

[35] P. G. Pomaska, *RGB-Kameras*, 2013. [Online]. Available: https://www.scanner.imagefact.de/de/rgbcam.html (visited on 08/21/2023).

[36] C. Wankhede, "Facial recognition on smartphones: Is it secure and should you use it?", *Android Authority*, Aug. 2022. [Online]. Available: https://www.androidauthority.com/face-unlock-smartphones-3043993/ (visited on 08/08/2023).

[37] ITWissen, "Signal-rausch-verhältnis (s/n)", *ITWissen.info*, Jan. 22, 2014. [Online]. Available: https://www.itwissen.info/Signal-Rausch-Verhaeltnis-S-N-signal-to-noise-ratio-SNR.html (visited on 08/04/2023).

[38] Spektrum, "Spektrum", in *Lexikon der Physik*. Heidelberg: Spektrum, 1998. [Online]. Available: https://www.spektrum.de/lexikon/physik/spektrum/13565 (visited on 08/04/2023).

[39] D. Barolet, F. Christiaens, and M. R. Hamblin, "Infrared and skin: Friend or foe", *Journal of photochemistry and photobiology*, vol. 155, pp. 78–85, Feb. 2016.

[40] S. Z. Li and D. Yi, "Face recognition, near-infrared", in *Encyclopedia of Biometrics*. Boston: Springer, 2009, pp. 352–355.

[41] mobiLLab-Team, "Theorie wärmebildkamera und ir-thermometer", mobiLLab. [Online]. Available: https://www.mobillab.ch/Articulate/Waermebildkamera/story_content/external_files/Theorie_Waermebildkamera.pdf (visited on 08/06/2023).

[42] S. Z. Li and A. Jain, "Lambertian law", in *Encyclopedia of Biometrics*. Boston: Springer, 2009, p. 883.

[43] S. Z. Li, R. Chu, S. Liao, and L. Zhang, "Illumination invariant face recognition using near-infrared images", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 4, pp. 627–639, 2007.

[44] T. Ojala, M. Pietikainen, and D. Harwood, "A comparative study of texture measures with classification based on feature distributions", in *Pattern Recognition*. 1996.

[45] T. Ahonen, A. Hadid, and M. Pietikainen, "Face recognition with local binary patterns", in *Computer Vision - ECCV 2004*. Heidelberg: Springer, 2004.

[46] M. J. Mendenhall, A. S. Nunez, and R. K. Martin, "Human skin detection in the visible and near infrared", *Applied Optics*, vol. 54, no. 35, pp. 10 559–10 570, Dec. 15, 2015.

[47] G. Lu and B. Fei, "Medical hyperspectral imaging: A review", *Journal of biomedical optics*, 2014.

[48] ISO, "Information technology - biometric presentation attack detection", ISO/IEC, 2017. [Online]. Available: https://www.iso.org/standard/67381.html (visited on 08/15/2023).

[49] ISO, "ISO/IEC 30107-3:2023(en), Information technology — Biometric presentation attack detection — Part 3: Testing and reporting", ISO/IEC, 2023. [Online]. Available: https://www.iso.org/obp/ui/#iso:std:iso-iec:30107:-3:ed-2:v1:en (visited on 04/12/2023).

[50] Z. Zheng, Q. Wang, and C. Wang, "Spoofing attacks and anti-spoofing methods for face authentication over smartphones", *IEEE Communications Magazine*, pp. 1–7, Mar. 2023.

[51] A. Obied, "How to attack biometric systems in your spare time", Department of Computer Science University of Calgary, Tech. Rep., 2006.

[52] V. S. Chakravarthi and S. R. Koteshwar, “Introduction to design of system on chips and future trends in vlsi”, in *SoC Physical Design : A Comprehensive Guide*. Cham: Springer International Publishing, 2022, pp. 1–20.

[53] D. S. Luber and P. Schmitz, “Was ist eine trusted execution environment (tee)?”, *Security Insider*, Nov. 23, 2020. [Online]. Available: https://www.security-insider.de/was-ist-ein-trusted-execution-environment-tee-a-984015/ (visited on 08/22/2023).

[54] Bundesbeauftragte für den Datenschtutz und die Informationssicherheit, *Polizeiliches informationssystem - inpol*. [Online]. Available: https://www.bfdi.bund.de/DE/Buerger/Inhalte/Polizei-Strafjustiz/National/DasPolizeilicheInformationssystem-INPOL.html (visited on 08/22/2023).

[55] I. Chingovska, N. Erdogmus, A. Anjos, and S. Marcel, “Face recognition systems under spoofing attacks”, in *Face Recognition Across The Imaging Spectrum*. Switzerland: Springer, 2016.

[56] P. Kulche, “Gezichtsherkenning op smartphone niet altijd veilig”, Apr. 2019. [Online]. Available: https://www.consumentenbond.nl/veilig-internetten/gezichtsherkenning-te-hacken (visited on 04/10/2023).

[57] T. Stürmer, “Schockierend einfach: Gesichtserkennung auch bei s20, p40 pro & co. geknackt”, *COMPUTER BILD*, Apr. 2020. [Online]. Available: https://www.computerbild.de/artikel/cb-Tests-Handy-Gesichtserkennung-mit-Foto-geknackt-24910985.html (visited on 04/03/2023).

[58] D. Oberhaus, “Iphone x’s face id can be fooled with a 3d-printed mask”, Nov. 2017. [Online]. Available: https://www.vice.com/en/article/qv3n77/iphone-x-face-id-mask-spoof (visited on 04/05/2023).

[59] T. Brewster, “We broke into a bunch of android phones with a 3d-printed head”, *Forbes*, Dec. 2018. [Online]. Available: https://www.forbes.com/sites/thomasbrewster/2018/12/13/we-broke-into-a-bunch-of-android-phones-with-a-3d-printed-head/?sh=4a16aea41330 (visited on 04/07/2023).

[60] D. Winder, “Apple’s iphone faceid hacked in less than 120 seconds”, *Forbes*, Aug. 2019. [Online]. Available: https://www.forbes.com/sites/daveywinder/2019/08/10/apples-iphone-faceid-hacked-in-less-than-120-seconds/?sh=785c8ac821bc (visited on 04/07/2023).

[61] S. Lindow-Zechmeister, “Pixel 4 - Radar Soli-Chip und Face Unlock bestätigt”, *Android User*, Jul. 2019. [Online]. Available: https://www.android-user.de/pixel-4-radar-soli-chip-und-face-unlock-bestaetigt/ (visited on 06/09/2023).

[62] A. Seeger, “Pixel 4: Google enthüllt face unlock und radarsteuerung”, *connect*, Jul. 30, 2019. [Online]. Available: https://www.connect.de/news/google-pixel-4-face-unlock-soli-3199756.html (visited on 08/25/2023).

[63] Spektrum, “Stereoskopie”, *Lexikon der Biologie*, [Online]. Available: https://www.spektrum.de/lexikon/biologie/stereoskopie/63752 (visited on 08/16/2023).

[64] J. Purcher, “Google Wins a Patent for Face ID-like Technology that could be used in a Future Pixel Phone to interpret hand Gesturing”, *Patently Extra News - Industry News+*, Nov. 27, 2018. [Online]. Available: https://www.patentlyapple.com/patently-apple/2018/11/google-wins-a-patent-for-face-id-like-technology-that-could-be-used-in-a-future-pixel-phone-to-interpret-hand-gesturing.html (visited on 08/17/2023).

[65] Apple, “Face id security”, Nov. 2017. [Online]. Available: https://www.apple.com/business-docs/FaceID_Security_Guide.pdf (visited on 04/07/2023).

[66] G. Ng, “Apple’s iphone 12 face id notch detailed by relible leaker”, *iPhone in Canada*, Apr. 19, 2020. [Online]. Available: https://www.iphoneincanada.ca/2020/04/19/iphone-12-face-id-notch-leaker/ (visited on 08/25/2023).

[67] W. Wu and T. Qiong, “Apple neural engine internal”, Ant Security Lab, May 2021.

[68] A. Orhon, A. Wadhwa, Y. Kim, F. Rossi, and V. Jagadeesh, “Deploying transformers on the apple neural engine”, *Machine Learning Research*, Jun. 2022. [Online]. Available: https://machinelearning.apple.com/research/neural-engine-transformers (visited on 09/04/2023).

[69] J. Purcher, “Apple wins a Patent for Face ID and more specifically for techniques that Prevent Spoofing of Biometric Data”, *Biometrics & Health*, Oct. 19, 2021. [Online]. Available: https://www.patentlyapple.com/2021/10/apple-wins-a-patent-for-face-id-and-more-specifically-for-techniques-that-prevent-spoofing-of-biometric-data.html (visited on 08/15/2023).

[70] Microsoft, “Windows Hello-Gesichtsauthentifizierung”, Dec. 2022. [Online]. Available: https://learn.microsoft.com/de-de/windows-hardware/design/device-experiences/windows-hello-face-authentication (visited on 06/09/2023).

[71] Raspberry Pi, “The official raspberry pi beginner’s guide”, 2020.

[72] J. Flynt, “Structured Light 3D Scanning: What Is It and How Does It Work?”, *3D Insider*, Jan. 2020. [Online]. Available: https://3dinsider.com/structured-light-3d-scanning/ (visited on 06/13/2023).

[73] M. McMillion, “Wie funktioniert das 3D-Scannen mit strukturiertem Licht?”, Nov. 25, 2022. [Online]. Available: https://www.artec3d.com/de/learning-center/structured-light-3d-scanning (visited on 06/13/2023).

[74] J. Flynt, “What is 3D Scanning?”, en-US, *3D Insider*, Apr. 2019. [Online]. Available: https://3dinsider.com/what-is-3d-scanning/ (visited on 06/13/2023).

[75] J. Raychev, G. Hristov, D. Kyuchukova, and P. Zahariev, “Workflow for development of a virtual museum that will provide better way for learning the cultural heritage”, Jul. 2017.

[76] T. Giesko, A. Zbrowski, and P. Czajka, “Laser profilometers for surface inspection and profile measurement”, *Problemy Eksploatacji*, Jan. 2007.

[77] A. Knudby, “Photogrammetry and structure-from-motion”, in *REMOTE SENSING*, 2023.

[78] S. van Riel, “Exploring the use of 3d gis as an analytical took in archaeological excavation practice”, Ph.D. dissertation, Jun. 2016.

[79] S. Kahraman, “Fused Deposition Modeling (FDM) Alles über das FDM Verfahren”, *three-dom.de - 3D Druck Blog*, Apr. 2023. [Online]. Available: https://threedom.de/fused-deposition-modeling-fdm (visited on 06/13/2023).

[80] Formlabs, *Der ultimative Leitfaden für den 3d-druck im stereolithographieverfahren (sla)*. [Online]. Available: https://formlabs.com/de/blog/leitfaden-stereolithografie-sla-3d-druck/ (visited on 07/05/2023).

[81] A. A. Konta, M. García, and D. R. Serrano, "Personalised 3d printed medicines: Which techniques and polymers are more successful?", Sep. 2017.

[82] C. Wu, R. Yi, Yong-Jin Liu, Y. He, and C. C. L. Wang, "Delta dlp 3d printing with large size", in *International Conference on Intelligent Robots and Systems (IROS)*, Daejon, Korea: IEEE, Oct. 2016, pp. 9–14.

[83] C. Hedgehog, *Hedgecam 2: Advanced camera*. [Online]. Available: https://play.google.com/store/apps/details?id=com.caddish_hedgehog.hedgecam2 (visited on 06/12/2023).

[84] Libcamera, *Libcamera*. [Online]. Available: https://libcamera.org/ (visited on 08/17/2023).

[85] FFmpeg, *Ffmpeg*. [Online]. Available: https://ffmpeg.org/about.html (visited on 08/25/2023).

[86] Shining 3D, *EinScan S Software*, en, 2023. [Online]. Available: https://www.einscan.com/einscan-software/exscan-software-download/ (visited on 06/12/2023).

[87] Creaform, *3D-Messoftwareplattform und Anwendungssuite*. [Online]. Available: https://www.creaform3d.com/de/messtechnik/softwareplattformen-fuer-3d-anwendungen (visited on 06/12/2023).

[88] Artec Studio, *Artec studio 17*. [Online]. Available: https://www.artec3d.com/de/3d-software/artec-studio (visited on 06/12/2023).

[89] Agisoft, *Agisoft metashape: Professional*. [Online]. Available: https://www.agisoft.com/features/professional-edition/ (visited on 08/17/2023).

[90] P. Merkert, "G-code verstehen und generieren", Hochschule Augsburg, 2018.

[91] Meshmixer, *Autodesk meshmixer*. [Online]. Available: https://meshmixer.com/ (visited on 06/12/2023).

[92] Autodesk, *Fusion 360*. [Online]. Available: https://www.autodesk.de/products/fusion-360/overview?term=1-YEAR&tab=subscription (visited on 06/12/2023).

[93] prusa3d, *PrusaSlicer 2.5.2*. [Online]. Available: https://www.prusa3d.com/de/page/prusaslicer_424/ (visited on 06/12/2023).

[94] Elliot J. Chikofsky and James Henry Cross, "Reverse engineering and design recovery: A taxonomy", 1990.

[95] Android Studio, *Android Debug Bridge (adb)*. [Online]. Available: https://developer.android.com/tools/adb (visited on 08/17/2023).

[96] Ghidra, *Ghidra*. [Online]. Available: https://ghidra-sre.org/ (visited on 08/17/2023).

[97] FRIDA, *Frida - dynamic instrumentation toolkit for developers, reverse-engineers, and security researchers.* [Online]. Available: https://frida.re/ (visited on 09/05/2023).

[98] mh-nexus, *Hxd - freeware hex editor and disk editor*. [Online]. Available: https://mh-nexus.de/en/hxd/ (visited on 09/04/2023).

[99] Android Studio, *Logcat command-line tool*. [Online]. Available: https://developer.android.com/tools/logcat (visited on 08/17/2023).

[100] P. Christensson, *Dpi*, 2006. [Online]. Available: https://techterms.com/definition/dpi (visited on 08/25/2023).

[101] OMRON, "Technical guide proximity sensors", Tech. Rep. [Online]. Available: https://www.ia.omron.com/data_pdf/guide/41/proximity_tg_e_6_2.pdf (visited on 09/04/2023).

[102] P. Schnabel, *Computertechnik-Fibel*, 5th ed. 2020.

[103] Yale University, "Executable and linkable and format (elf)", Yale University, Portable Format Specification, version 1.1.

[104] D. Luber and N. Litzel, "Was ist eine tensor processing unit (tpu)?", *Bigdata Insider*, Sep. 4, 2018. [Online]. Available: https://www.bigdata-insider.de/was-ist-eine-tensor-processing-unit-tpu-a-750292/ (visited on 09/04/2023).

[105] *Reversing and Fuzzing the Google Titan M Chip*, presented at the ROOTS'21: Reversing and Offensive-oriented Trends Symposium, New York: Association for Computing Machinery, Nov. 2021.

[106] sculpteo, *Was ist eine stl datei?* [Online]. Available: https://www.sculpteo.com/de/3d-lernzentrum/erstellung-einer-druckbaren-3d-datei/was-ist-eine-stl-datei/ (visited on 08/29/2023).

[107] D. Chakravorty, "The obj file format - simply explained", *All3DP*, Mar. 30, 2023. [Online]. Available: https://all3dp.com/1/obj-file-format-3d-printing-cad/ (visited on 08/31/2023).

[108] N. Teraphongphom, C. Kong, J. Warram, and E. L. Rosenthal, "Specimen mapping in head and neck cancer using fluorescence imaging: Specimen mapping in hnc", *Laryngoscope Investigative Otolaryngology*, 2017.

[109] A. K. Dabrowska *et al.*, "Materials used to simulate physical properties of human skin", Swiss Federal Laboratories for Materials Science, Technology, Laboratory for Protection, and Physiology, 2015.

[110] KeenTools, *Facebuilder for blender 2022.2: From photos to metahuman*, Aug. 13, 2022. [Online]. Available: https://www.blendernation.com/2022/08/13/facebuilder-for-blender-2022-2-from-photos-to-metahuman/ (visited on 09/25/2023).

Eidesstattliche Erklärung

Hiermit versichere ich – Navina Halbe – an Eides statt, dass ich die vorliegende Arbeit selbstständig und nur unter Verwendung der angegebenen Quellen und Hilfsmittel angefertigt habe.

Sämtliche Stellen der Arbeit, die im Wortlaut oder dem Sinn nach Publikationen oder Vorträgen anderer Autoren entnommen sind, habe ich als solche kenntlich gemacht.

Diese Arbeit wurde in gleicher oder ähnlicher Form noch keiner anderen Prüfungsbehörde vorgelegt oder anderweitig veröffentlicht.

Wiesbaden, 06. October 2023 | Navina Halbe, B.Sc.

Ort, Datum