FACE MORPHING AND MORPHING ATTACK DETECTION

zur Erlangung des akademischen Grades Doktor-Ingenieur (Dr.-Ing.)
genehmigte Dissertation von

ULRICH JOHANNES SCHERHAG

geboren in Mainz

1. Gutachten: Prof. Dr. Dr. eh. Dieter Fellner
2. Gutachten: Prof. Dr. Christoph Busch
3. Gutachten: Prof. Dr. Raymond N. J. Veldhuis

Tag der Einreichung: 01.10.2020
Tag der Prüfung: 16.11.2020

Fachgebiet Graphisch-Interaktive Systeme
Fachbereich Informatik
Technische Universität Darmstadt
Hochschulkennziffer D-17

November 2020

Ulrich Johannes Scherhag:
Face Morphing and Morphing Attack Detection,
Darmstadt,Technische Universität Darmstadt,

Jahr der Veröffentlichung der Dissertation auf TUprints: 2021
URN: urn:nbn:de:tuda-tuprints-145956
Tag der mündlichen Prüfung: 16.11.2020

SUPERVISORS:
Prof. Dr. Dr. eh. Dieter Fellner
Prof. Dr. Christoph Busch
Prof. Dr. Raymond N. J. Veldhuis

ERKLÄRUNGEN LAUT PROMOTIONSORDNUNG

§8 Abs. 1 lit. c PromO
Ich versichere hiermit, dass die elektronische Version meiner Dissertation mit der schriftlichen Version übereinstimmt.

§8 Abs. 1 lit. d PromO
Ich versichere hiermit, dass zu einem vorherigen Zeitpunkt noch keine Promotion versucht wurde. In diesem Fall sind nähere Angaben über Zeitpunkt, Hochschule, Dissertationsthe- ma und Ergebnis dieses Versuchs mitzuteilen.

§9 Abs. 1 PromO
Ich versichere hiermit, dass die vorliegende Dissertation selbstständig und nur unter Verwendung der angegebenen Quellen verfasst wurde.

§9 Abs. 2 PromO
Die Arbeit hat bisher noch nicht zu Prüfungszwecken gedient.

Darmstadt, November 2020

Ulrich Johannes Scherhag

ABSTRACT

In modern society, biometrics is gaining more and more importance, driven by the increase in recognition performance of the systems. In some areas, such as automatic border controls, there is no alternative to the application of biometric systems.

Despite all the advantages of biometric systems, the vulnerability of these still poses a problem. Facial recognition systems for example offer various attack points, like faces printed on paper or silicone masks. Besides the long known and well researched presentation attacks there is also the danger of the so-called morphing attack.

The research field of morphing attacks is quite young, which is why it has only been investigated to a limited extent so far. Publications proposing algorithms for the detection of morphing attacks often lack uniform databases and evaluation methods, which leads to a restricted comparability of the previously published work. Thus, the focus of this thesis is the comprehensive analysis of different features and classifiers in their suitability as algorithms for the detection of morphing attacks. In this context, evaluations are performed with uniform metrics on a realistic morphing database, allowing the simulation of various realistic scenarios.

If only the suspected morph is available, a HOG feature extraction in combination with an SVM is able to detect morphs with a D-EER ranging from 13.25% to 24.05%. If a trusted live capture image is available in addition, for example from a border gate, the deep ArcFace features in combination with an SVM can detect morphs with a D-EER ranging from 2.71% to 7.17%.

ZUSAMMENFASSUNG

In der modernen Gesellschaft gewinnt die Biometrie, insbesondere durch die Steigerung der Erkennungsleistung der Systeme, zunehmend an Bedeutung. In manchen Bereichen, zum Beispiel bei automatischen Grenzkontrollen, ist der Einsatz biometrischer Systeme alternativlos.

Trotz aller Vorteile biometrischer Systeme stellt die Angreifbarkeit dieser noch immer ein Problem dar. So bieten Gesichtserkennungssysteme verschiedene Angriffspunkte, zum Beispiel durch auf Papier gedruckte Gesichter oder Gummimasken. Neben den länger bekannten und gut erforschten Präsentationsangriffen besteht auch die Gefahr des so genannten *Morphingangriffs*.

Das Forschungsfeld im Zusammenhang mit Morphingangriffen ist noch jung, weshalb es bisher erst in einem überschaubaren Umfang bearbeitet wurde. Bei Publikationen, welche Algorithmen zur Erkennung von Morphingangriffen vorschlagen, mangelt es häufig an einheitlichen Datenbanken und Evaluationsmethoden, was zu einer begrenzten Vergleichbarkeit der bisher publizierten Arbeiten führt. Daher liegt der Fokus der vorliegenden Dissertation auf der umfassenden Analyse unterschiedlicher Merkmale und Klassifikatoren auf ihre Eignung als Algorithmen zur Erkennung von Morphingangriffen. Hierbei wird mit einheitlichen Metriken auf einer realistischen Morphing Datenbank evaluiert, sodass verschiedene realitätsnahe Szenarien abgebildet werden können.

Steht nur der mutmaßliche Morph zur Verfügung, so kann HOG in Kombination mit einer SVM Morphs mit einer D-EER zwischen 13.25% und 24.05% detektieren. Steht zusätzlich eine vertrauenswürdige Aufnahme, zum Beispiel aus der Grenzkontrolle, zur Verfügung, so kann eine Kombination aus den tiefen ArcFace Merkmalen in Kombination mit einer SVM Morphs mit einer D-EER zwischen 2.71% uns 7.17% detektieren.

PUBLICATIONS

JOURNALS

[1] U. Scherhag, L. Debiasi, C. Rathgeb, C. Busch, and A. Uhl. "Detection of Face Morphing Attacks based on PRNU Analysis." In: *IEEE Transactions on Biometrics, Behavior, and Identity Science (T-BIOM)* (2019), pp. 1–16.

[2] U. Scherhag, J. Kunze, C. Rathgeb, and C. Busch. "Face Morph Detection for Unknown Morphing Algorithms and Image Sources: A Multi-Scale Block Local Binary Pattern Fusion Approach." In: *IET-Biometrics* (2020), pp. 1–11.

[3] U. Scherhag, C. Rathgeb, J. Merkle, R. Breithaupt, and C. Busch. "Face Recognition Systems Under Morphing Attacks: A Survey." In: *IEEE Access* 7 (2019), pp. 23012–23026.

[4] U. Scherhag, C. Rathgeb, J. Merkle, and C. Busch. "Deep Face Representations for Differential Morphing Attack Detection." In: *IEEE Transactions on Information Forensics and Security (TIFS)* (2020), pp. 3625–3639.

CONFERENCES

[1] U. Scherhag, D. Budhrani, M. Gomez-Barrero, and C. Busch. "Detecting Morphed Face Images Using Facial Landmarks." In: *Proceedings of the 2018 International Conference on Image and Signal Processing (ICISP)*. Springer International Publishing, 2018, pp. 444–452.

[2] U. Scherhag, R. Ramachandra, K. B. Raja, M. Gomez-Barrero, C. Rathgeb, and C. Busch. "On the vulnerability of face recognition systems towards morphed face attacks." In: *Proceedings of the 5th International Workshop on Biometrics and Forensics (IWBF)*. IEEE, Apr. 2017.

[3] U. Scherhag, C. Rathgeb, and C. Busch. "Morph detection from single face images: a multi-algorithm fusion approach." In: *Proceedings of the 2018 International Conference on Biometrics Engineering and Application (ICBEA)*. ACM. 2018.

[4] U. Scherhag, C. Rathgeb, and C. Busch. "Performance Variation of Morphed Face Image Detection Algorithms across different Datasets." In: *Proceedings of the 6th International Workshop on Biometrics and Forensics (IWBF)*. IEEE, June 2018.

[5] U. Scherhag, C. Rathgeb, and C. Busch. "Towards detection of morphed face images in electronic travel documents." In: *Proceedings of the 13th Workshop on Document Analysis Systems (DAS)*. IAPR, 2018.

[6] U. Scherhag et al. "Biometric Systems under Morphing Attacks: Assessment of Morphing Techniques and Vulnerability Reporting." In: *Proceedings of the 2017 International Conference of the Biometrics Special Interest Group (BIOSIG)*. IEEE, Sept. 2017.

FURTHER CONTRIBUTIONS

[1] L. Debiasi, N. Damer, A. M. Saladié, C. Rathgeb, U. Scherhag, C. Busch, F. Kirchbuchner, and A. Uhl. "On the Detection of GAN-based Face Morphs using Established Morph Detectors." In: *Proceedings of the International joint conference on biometrics (IJCB)*. IEEE, 2019.

[2] L. Debiasi, C. Rathgeb, U. Scherhag, A. Uhl, and C. Busch. "PRNU Variance Analysis for Morphed Face Image Detection." In: *Proceedings of the 9th IEEE International Conference on Biometrics: Theory, Applications, and Systems (BTAS)*. IEEE, 2018.

[3] L. Debiasi, U. Scherhag, C. Rathgeb, A. Uhl, and C. Busch. "PRNU-based Detection of Morphed Face Images." In: *Proceedings of the 6th International Workshop on Biometrics and Forensics (IWBF)*. IEEE, 2018.

[4] M. Gomez-Barrero, C. Rathgeb, U. Scherhag, and C. Busch. "Is your biometric system robust to morphing attacks?" In: *Proceedings of the 5th International Workshop on Biometrics and Forensics (IWBF)*. IEEE, Apr. 2017.

[5] M. Gomez-Barrero, C. Rathgeb, U. Scherhag, and C. Busch. "Predicting the vulnerability of biometric systems to attacks based on morphed biometric information." In: *IET Biometrics* 7.4 (July 2018), pp. 333–341.

[6] J. Merkle, C. Rathgeb, U. Scherhag, and C. Busch. "Morphing-Angriffe – Ein Sicherheitsrisiko für Gesichtserkennungssysteme." In: *Datenschutz und Datensicherheit (DuD)*. Springer, 2019.

[7] J. Merkle, C. Rathgeb, U. Scherhag, C. Busch, and R. Breithaupt. "Face Morphing Detection: Issues and Challenges." In: *Proceedings of the International Confonference on Biometrics for Borders (ICBB)*. 2019.

[8] A. Röttcher, U. Scherhag, and C. Busch. "Finding the Suitable Doppelgänger for a Face Morphing Attack." In: *Proceedings of the International joint conference on biometrics (IJCB)*. IEEE, 2020.

ACKNOWLEDGMENTS

First of all I would like to thank my promoters, *Prof. Dr. Dr. eh. Dieter Fellner*, *Prof. Dr. Christoph Busch* and *Prof. Dr. Raymond Veldhuis* who have made this graduation possible. Special thanks go to *Christoph*, who led me to the interesting research field of biometrics in 2014 and has since then not only supervised my master thesis, but also my PhD with a lot of advice, interesting discussions and wisdom. Additionally I thank him for opening doors to my future professional career.

I would also like to thank all project partners who made this doctorate financially possible for me. Special acknowledgements go to *Ralf Breithaupt* from Bundesamt für Sicherheit in der Informationstechnik (BSI) and *Johannes Merkle* from Secunet.

Further thanks go to the colleagues of the da/sec research group for the pleasant working atmosphere and the good cooperation. Special thanks go to *Christian Rathgeb* for the intensive supervision of my PhD, the numerous helpful discussions and the motivation. Many thanks to my two office partners, *Pawel Drozdowski* and *Daniel Fischer*, for their pleasant, collegial atmosphere and the entertaining coffee breaks. Also, I would like to thank my colleagues from other research areas. In particular *Lorenz Liebler*, *Thomas Göbel* and *Johannes Wagner* for the interesting discussions and the views from a perspective beyond biometrics.

Thanks are also due to my former colleagues. Special thanks go to *Andreas Nautsch*, from whom I was able to learn a lot about the principles of scientific work, as well as to *Marta Gomez-Barrero* for her support, especially in the beginning of my work on the topic of morphing.

Thanks also to the research assistants who supported and accompanied me during my graduation. In particular I would like to thank *Jonas Kunze* and *Fabian Stockhardt* for their reliable preparatory work. I am looking forward to continue working on projects together with *Fabian* in the future.

I would like to thank my family, especially my parents *Anton* and *Heike* for their advice and support on my way to the doctorate. Furthermore, I would like to thank my friends who supported me, in particular *Alexander Nahrwold* for proofreading my work and for fine tuning my English.

Finally, I would like to thank my wife *Anna* for her love, support and balance during the last years.

CONTENTS

III CONCEPTS AND RELATED WORK

IV MORPHING ATTACK DETECTION PIPELINE

LIST OF FIGURES

LIST OF TABLES

ACRONYMS

ABC	Automated Border Control. 7
APCER	Attack Presentation Classification Error Rate. 76–78, 87, 166, 187, 192
BPCER	Bona Fide Presentation Classification Error Rate. 76–78, 87, 148, 150, 161, 164–168, 187, 192
BRISQUE	Blind/Referenceless Image Spatial Quality Evaluator. 125
BSI	Bundesamt für Sicherheit in der Informationstechnik. 9
BSIF	Binarised Statistical Image Features. 34, 36–38, 47, 79–81, 85, 97, 98, 152, 164, 172, 173, 175, 181, 183, 188, 190–192
CNN	Convolutional NN. 29, 83–85, 102, 105–107
CRC	Collaborative Representation Classifier. 83
D-EER	Detection Equal Error Rate. 78, 148–164, 166, 167, 169–178, 181, 182, 188, 189, 191, 192

DET	Detection Error Trade-off. 78, 164, 166, 169, 170, 172–174, 176, 178, 179
DFT	Discreet Fourier Transformation. 82, 104, 105
DNN	Deep NN. 29, 47, 55, 65, 80, 83, 84, 86, 97, 105, 107, 179
DoG	Difference of Gaussians. 42, 44
EER	Equal Error Rate. 53, 54, 78, 154
EU	European Union. 9
FAR	False Accept Rate. 53, 56
FMR	False Match Rate. 52–54, 56, 72, 77, 78, 141, 142, 144, 187
FNMR	False Non-Match Rate. 52–54, 56, 72, 74–78, 142, 168, 187
FRR	False Reject Rate. 53, 56
FRS	Face Recognition System. 4–7, 10, 13, 49, 54, 55, 61, 65, 67, 68, 71, 72, 75, 77, 86, 87, 91, 105–107, 116, 135, 141, 142, 144, 145, 168, 180, 183, 187, 188
FRVT	Face Recognition Vendor Test. 9, 10, 79, 123, 187, 191–193, 195
FTA	Failure-To-Acquire. 52
FTC	Failure-To-Capture. 51
FTE	Failure-To-Enrol. 52
FTX	Failure-To-eXtract. 52
GAN	Generative Adversarial Networks. 33, 48, 80, 82
GDPR	General Data Protection Regulation. 123
HOG	Histogram of Oriented Gradients. 38, 40–43, 80, 97, 99–101, 154, 164, 172–176, 178, 180–184, 188–190
IAPMR	Impostor Attack Presentation Match Rate. 72–74, 87, 195
ICA	Independent Component Analysis. 36
ICAO	International Civil Aviation Organization. 9, 93, 119, 123
IEC	International Electrotechnical Commission. 3, 4, 49, 50, 61, 72, 76, 83, 87, 95, 123, 129

ISO	International Organization for Standardization. 3, 4, 49, 50, 61, 72, 76, 83, 87, 95, 123, 129
LBP	Local Binary Patterns. xxi, 34–36, 38, 47, 55, 80, 81, 83, 91, 97–99, 152, 170, 172–176, 179, 181–183, 188, 191, 192
LoG	Laplacian of Gaussian. 42, 44
MA	Morphing Attack. 10, 71, 77, 80, 82, 84, 85
MAD	Morphing Attack Detection. xxii, 5, 6, 8–12, 71, 72, 76–81, 83–87, 91, 92, 100, 101, 103, 105–107, 109–111, 113–116, 119, 123, 132, 136, 137, 147–150, 152, 154, 157, 160, 161, 164, 166, 168, 170–184, 187–189, 191, 192, 195
MB-LBP	Multi-Scale Block LBP. 35, 36, 38, 98
MLP	Multi-Layer Perceptron. 28, 29
MMPMR	Mated Morph Presentation Match Rate. 72–76, 87, 143, 183, 187
MTCNN	MultiTask Cascaded convolutional Neural Network. 106
NIST	National Institute of Standards and Technology. 9, 10, 79, 123, 126, 127, 187, 191–193, 195
NN	Neural Network. xix, xx, 27–30, 47, 83
PAC	Probably Approximately Correct. 25
PAD	Presentation Attack Detection. 116, 195
PAI	Presentation Attack Instrument. 77
PDF	Probability Density Function. 52, 53, 72, 78, 141, 144
PRNU	Photo Response Non-Uniformity. 47, 48, 82, 83, 97, 103, 104, 160, 191
RBF	Radial Basis Function. 20, 21, 30, 115, 116, 148, 157, 160, 161, 164
ReLU	Rectified Linear Unit. 29
RIAPAR	Relative Impostor Attack Presentation Accept Rate. 195
RMMR	Related Morph Match Rate. 74–76, 87, 143, 183, 187, 195
RoI	Region of Interest. 101, 102

S-MAD	single image MAD. 71, 79, 80, 82, 84, 136, 148, 154, 161, 169–171, 175, 177, 180, 188
SIFT	Scale-Invariant Feature Transform. 41, 43–45, 97, 100, 101, 157
SOTAMD	State-Of-The-Art Morphing Detection. 9, 79, 123, 187, 191, 192, 195
SPN	Sensor Pattern Noise. 47, 48, 82, 83, 97, 104, 105, 160
SRKDA	Spectral Regression Kernel Discriminant Analysis. 81, 83
SURF	Speeded Up Robust Features. 41, 44, 45, 97, 100, 101, 157
SVM	Support Vector Machine. 17–21, 30, 61, 80–82, 85, 86, 101, 110, 115, 116, 125, 148, 150, 152, 154, 157, 160, 161, 164, 183, 184, 188, 189, 191
TLC	Trusted Live Capture. 71, 72, 84–86, 91–93, 101, 109, 110, 119, 123, 125–127, 129–132, 134, 137, 142, 152, 157, 160, 161, 188
TMR	True Match Rate. 75, 76, 142
WLMP	Weighted Local Magnitude Patterns. 81

Part I

OVERVIEW

1 INTRODUCTION

Biometrics describes the automated recognition of individuals based on their biological and behavioural characteristics [66].

The advantage of biometric systems over conventional authentication methods, such as password or token-based authentication is, that it is impossible to lose, forget or share biometric characteristics [68]. However, this technology bears the disadvantage, that if someone gains unauthorised possession of the Biometric Features of another person, the corresponding characteristic (and thus the extractable features) cannot be exchanged.

1.1 APPLICATIONS OF BIOMETRIC SYSTEMS

Due to the advantages mentioned above, biometric systems are gaining more and more popularity in a wide range of applications, so that biometric systems generate a market value of 24.5 billion dollars to day. Already 60% of all newly sold smartphones can be unlocked with a biometric system and half of the companies worldwide are planning to invest in biometric identity management systems [153].

In some applications, for example in smartphones or door locking systems, the use of biometric systems mainly increases the convenience of the identification process, in other applications there is no alternative to the use of biometric systems. Particularly in scenarios in which subjects itself are to be identified or verified independently of further information such as tokens or passwords, for example in the field of law enforcement and public security for the identification of criminals or suspects or for the identification or verification of civilians for example during elections, border controls or migration. Especially at airports, border controls are to a large extent automated by electronic border control systems (eGates). In 2018, more than 17 million border crossings were carried out at eGates in Germany alone; at Germany's airport with the highest throughput in Frankfurt, one third of the border controls were carried out fully automated [15].

1.2 ATTACKS ON BIOMETRIC SYSTEMS

The increasing prevalence of biometric systems also increases interest in circumventing or deceiving these systems through subversive use. The different attack vectors during identification and verification are defined as listed in International Organization for Standardization (ISO)/International Electrotechnical Commission (IEC) 30107-1 [64] and illustrated in Figure 1.1. The most common attacks considered in research are presentation attacks on the sensor (attack vector number 1 in Figure 1.1) [119]. These attacks can usually be performed without knowledge of the nature of the biometric system, are easy to implement and universally applicable. If biometric systems are properly set up and operated, the other attack vectors should not be accessible to the user and are not further considered in ISO/IEC 30107-1.

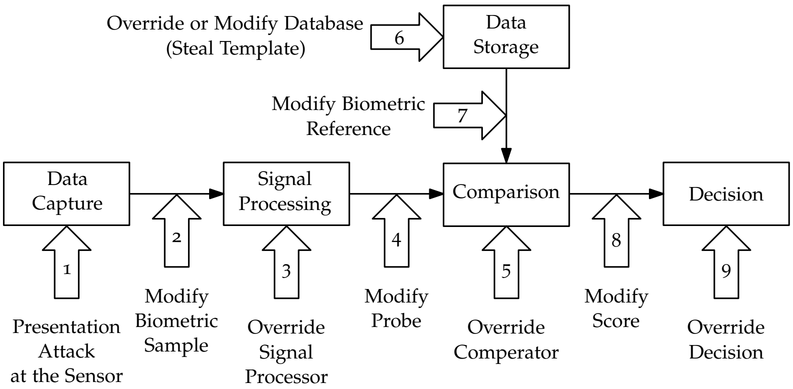


Figure 1.1: Attack points of biometric systems, inspired by [64]

Presentation attacks can be divided into two classes of subversive use: biometric impostor (or impersonation attacks) and biometric concealer (or obfuscation attacks) [64]. The objective of a biometric concealer is to conceal its own biometric characteristics in a way that it cannot be assigned to its biometric enrolment data record. This can be achieved, for example, in FRSs by obscuring the face with wigs, hats, beard, moustache, and sunglasses [84] or by changing the appearance of the face by applying makeup [80]. The objective of a biometric impostor, however, is to match to the reference of any or a specific subject stored in the database. For this attack, artefacts reflecting the biometric characteristics of the attacking subject can be presented to the biometric system's capture device. In FRSs, this can be done, for example, by presenting a sheet of paper with the facial image printed on it [3] or by wearing a silicone mask [31].

For biometric impersonator attacks, initially information about the characteristics of the subject to be impersonated is required. In case of face recognition a simple photo is usually sufficient, fingerprints can be acquired from smooth surfaces [174]. Another possibility is the theft of information stored in biometric databases, as, for example in the case of the breach of the database of the security platform *Biostar 2* from the provider Suprema, in which over one million fingerprints and face recognition information were stolen [157]. From the obtained information so-called presentation attack instruments can be constructed. For example silicone fingers to fool fingerprint readers or latex masks to fool FRSs.

In addition to the attack vectors illustrated in Figure 1.1, it is possible to attack the biometric system during the enrolment of a subject. In systems that allow the subject to hand over his or her own reference, the reference may be distorted or, in particular in the case of FRSs, manipulated by beautification [126]. If an attacker succeeds to submit a manipulated sample to the system during enrolment, which, for example, allows the verification of further subjects against the stored reference, the unique link, essential for biometric systems, between the subject and the reference is weakened. An attack based on this scheme is the face morphing attack introduced in [39]. Face morphing attacks and methods to detect those are investigated and described in detail in this thesis.

1.3 thesis organization

In this section an overview of the content of the thesis is given. It is divided into the following parts:

- Part I provides an overview of biometric systems, attacks on biometric systems in general and in particular face morphing attacks. Furthermore the structure of the thesis is described and the scope of the thesis and research questions are defined.

- Part II provides the background information needed for the understanding of the thesis. In Chapter 5 the principles of machine learning algorithms are described and in Chapter 6 different image descriptors are introduced. Chapter 7 gives an overview of the topology of biometric systems and their functioning, as well as a more detailed introduction to FRSs. Chapter 8 describes the concept of image morphing and its technical background.

- Part III describes the current techniques for the creation of morphing attacks, as well as methods used to detect these attacks and metrics to evaluate the detection performance of those. In addition, a comprehensive overview of developed algorithms for Morphing Attack Detection (MAD) is given.

- Part IV contains a description of the MAD pipeline implemented for this thesis. Chapter 14 explains the chosen design of the pipeline. The following chapters describe the preparation of the data, feature extraction, feature preparation for the machine learning algorithms and training of the machine learning algorithms.

- Part V describes the creation of the morphing database used in this thesis. The selection of the face databases used to create the morphing database is explained in Chapter 20, the protocol for creating the morphing database is described in Chapter 21.

- Part VI contains the experimental evaluation of the database created in Part V. First, in Chapter 23, the vulnerability of FRSs to the created morphs is examined. In Chapter 24 the detection performance of the MAD algorithms described in Part IV is analysed.
- Part VII concludes the findings observed in Part VI, answers the research questions defined in Part I and validates the observed detection performances with independent benchmarks.

2 MORPHING ATTACKS

As mentioned in Section 1.2, biometric systems can be compromised in their correct functioning during verification or identification by injection of manipulated samples during the enrolment process. This chapter first explains the underlying concept of attacks, followed by a discussion of the impact of these attacks on real-world application scenarios.

2.1 THE UNDERLYING CONCEPT

The precondition for using morphing attacks is the possibility to manipulate the sample prior to enrolment. The basic concept of morphing attacks is to combine the visual and biometric information of two or more subjects in one sample in such a way that both subjects are successfully verified against it. In this way, the otherwise unique link between subject and sample is loosened. The morphing attack is easy to implement for FRSs. Using known methods from the film industry, two facial images can be merged into one, containing the characteristics of both contributing subjects. Due to the fact that recording conditions of facial images are usually unconstrained, FRSs tend to offer a high robustness against changes in the image. As a consequence they are particularly susceptible to morphing attacks.

2.2 PASSPORT APPLICATION PROCESS

As, for a successful morphing attack, the sample used during enrolment has to be manipulated, it cannot be performed on every FRS. Systems in which the subject has access to the captured sample prior to the enrolment process are Automated Border Control (ABC) systems, in which the passport application process corresponds to the enrolment process.

The passport application process varies from country to country, even within the EU these processes are not standardised. In Germany, according to the passport regulations, a photograph has to be presented during the passport application process [14], the use of digital photographs is not intended. Consequently, the passport holder is able to manipulate the photograph used for enrolment, provided that it passes the visual inspection during the passport application. If a morphed passport photo is submitted and accepted, an authentic passport would be created based on this manipulated photo, which can be used to enter over 170 countries without a visa application by both subjects represented in the morph.

One possibility to overcome this weakness in the application process would be the introduction of a live enrolment, in which the passport photo would be captured directly at the application office. This would prevent access to the recorded sample by the applicant, resulting in the disabling of the attack vector necessary to launch a morphing attack. In Germany, as a political reaction to the problem of morphed passport photos, the discussion about the introduction of a live enrolment was initiated [12]. However, due to concerns of the retail sector about declining customer numbers of photographers, the creation of passport photos by photo retailers has not yet been suspended so far [94], thus the threat of attacks by morphed passport photos in German passports remains.

2.3 THREATS AGAINST THE OPERATIONAL SYSTEMS

The threat of morphing attacks, in particular on border control systems, is not only an academic problem. The number of illegal border crossings at the outer borders of the European Union is estimated by the European Commission to be 150,000 in 2018 [34]. Due to the lack of a system to record the real number of illegal border crossings, the number can only be estimated. Nevertheless, the high estimates indicate a great interest in illegal border crossings. Due to the easy application and the high chance of success, morphing attacks are suitable to simplify them considerably. The feasibility of this attack was demonstrated by the activist group *Peng!*, which in 2018 applied for a passport with a morphed picture of one of the group members and the former EU High Representative for Foreign Affairs and Security Policy, Federica Mogherini [150].

To date, only few informations about detected cases of morphed passports are available as they are not publicly announced by the respective states. As no MAD methods have been installed so far, the passport crossings with morphed passports have mostly been discovered by chance. For example, an asylum seeker who wanted to travel from Afghanistan via Belgium, Holland and Germany to Canada with a morphed Dutch passport was stopped at the German border on entry [79].

Apart from the illegal border crossing, the morphing attack can also be extended to various other scenarios, such as gym membership cards, driver licenses or insurance cards.

3 THESIS SCOPE

In this chapter the scope of the thesis is defined. First, the projects related to the topic of the thesis are presented. Subsequently, research questions are defined and research objectives are derived.

3.1 RELATED PROJECTS

In the following the projects related to the topic of this thesis are described.

3.1.1 *SOTAMD*

The objective of the State-Of-The-Art Morphing Detection (SOTAMD) project is to identify the state-of-the-art of MAD mechanisms by analysing its detection accuracy on a sequestered dataset. The partners have jointly build up this dataset including morphed and bona fide face images. This dataset serves as the basis for repeatable operational testing of morphed face image detection mechanisms. This dataset was collected in a distributed effort and subsequently a database of morphed face images was constructed, for which image quality according to International Civil Aviation Organization (ICAO) and European Union (EU) Regulation 2252/2004 is ensured.

3.1.2 *FACETRUST*

In order to improve the security of facial biometric systems, the BSI has launched the FACETRUST research project. The project FACETRUST aims to investigate the attack vectors on facial biometric systems with regard to their attack performance as well as suitable technical countermeasures for the security of the EasyPASS-eGate technology. The focus is on morphing attacks during the application and verification process, as well as presentation attacks during the verification process.

3.1.3 *NIST FRVT MORPH*

The National Institute of Standards and Technology (NIST) Face Recognition Vendor Test (FRVT) MORPH test provides ongoing independent testing of prototype face morph detection technologies. The evaluation is designed to obtain commonly measured assessment of morph detection capability to inform developers and current and prospective end-users. FRVT MORPH is open for ongoing participation worldwide, and NIST has since received multiple morph detection algorithm submissions from different academic entities, e.g. Hochschule Darmstadt University of Applied Sciences, Norwegian University of Science and Technology, and University of Bologna [103].

The test leverages a number of datasets created using different morphing methods with goals to evaluate algorithm performance over a large spectrum of morphing techniques. Testing was conducted using a tiered approach, where algorithms were evaluated on low quality morphs created with readily accessible tools available to non-experts, morphs generated using automated morphing methods based on academic research, and high quality morphs created using commercial-grade tools.

3.2 RESEARCH QUESTIONS

In the context of the thesis, six research questions are defined:

RQ1: Which metrics are applicable to the evaluation of the vulnerability of FRS and MAD algorithms?
In the research area of Morphing Attack (MA) and MAD, two types of evaluations are required. The analysis of the vulnerability of the FRS to the morphing attacks, and the evaluation of the detection performance of the MAD algorithms. In order to be able to compare different scenarios, unified metrics have to be defined. Two distinct research objectives can be identified:

- Determination of metrics and methodologies for morph vulnerability assessment.

- Determination of metrics and methodologies for morphing attack detection performance assessment.

RQ2: Under which circumstances is a system vulnerable to morphing attacks?
The general vulnerability of FRSs to morphing attacks has already been shown in several publications, for example in [39]. However, a deeper analysis of the impact of these attacks is missing. Subsequently, various morphing attacks can be tested on different FRSs. Two distinct research objectives can be identified:

- Analysis of the influence of different properties of face recognition algorithms, e.g. baseline performance, on the vulnerability of the system.

- Analysis of the influence of different properties of various morphing algorithms on the vulnerability of the system.

RQ3: Does the consideration of images from unknown data sources influence the evaluations results of MAD algorithms?
In the evaluation of machine learning algorithms it is common to divide a database into disjoint training and test sets. Face databases usually originate from a single camera, furthermore, in most publications on MAD, the morph samples in training and test sets are created using a single morphing algorithm. In a realistic scenario, however, the source of the images and the algorithm used for morphing are not uniform and unknown. The influence of this variance is to be investigated, leading to two distinct research objectives:

- Analysis of the influence of variations in the capture scenario of the images (unknown capture device, different lightning conditions, different distance to camera), which is simulated by training and test on different face databases.
- Analysis of the influence of different morphing algorithms on the evaluation performance of MAD algorithms.

RQ4: To what extent can morphed face images be reliable detected by automated algorithms?
Depending on the given scenario, different architectures of MAD algorithms are available. For each architecture a wide range of algorithms is available for extracting descriptive features, which can subsequently be used by different classifiers to detect morphing attacks. Depending on the scenario, architecture, features, classifiers and their parameters, different detection performances are achieved. It has to be investigated which combination of architecture, features and classifiers is best suited. Three distinct research objectives can be defined:

- Exploration of different MAD architectures.
- Theoretical consideration and practical investigation of which feature extractors are suitable for the detection of morphed facial images.
- Analysis of various classifiers for their suitability for the detection of morphed facial images.

RQ5: Which operational scenarios influence the detection of morphed face images?
Considering a real application scenario for MAD of passport photographs, it has to be taken into account that passport photographs may experience different processing steps prior to being stored in the passport. It has to be determined which post-processings are to be expected in passport photographs and their influence on the detection performance of MAD algorithms. Two distinct research objectives can be derived:

- Define post-processing chains the passport images have undergone depending on specific MAD scenarios.
- Evaluate the influence of different post-processing chains (e.g. resizing, print and scan, compression) to the evaluation results of MAD algorithms.

RQ6: Can information fusion be used to improve the MAD performance and robustness of the individual algorithms?
Score level fusion of different MAD algorithms may increase the performance and robustness compared to the individual classifier [135]. It has to be examined whether the improvements are observable in realistic scenarios as well. Two distinct research objectives can be defined:

- Analysis of which algorithms are contributing to an improvement of the resulting algorithm during fusion.
- Investigation whether the algorithms identified as suitable for fusion are universally applicable or depend on the specific architecture.

4 SUMMARY

Biometric systems are gaining more and more popularity as part of identity management systems. This growing prevalence in turn leads to an increased interest in attacks on these systems to deceive them in such a manner that they assign a presented characteristic to a false subject (biometric imposter or impersonation attacks) or fail to assign a presented characteristic to the respective subject (biometric concealer or obfuscation attack). The most common threat is the so-called presentation attack where a copy of a characteristic is presented to the sensor in order to deceive the system. This may be done by very simple attacks, such as a face printed on a piece of paper, or by more sophisticated attacks such as silicone masks.

A more recent and less researched attack are the so-called morphing attacks, which are particularly applicable to FRSs. Morphing is the combination of image and feature information of two subjects in one sample. In this way, both contributing subjects' samples are successfully matched with the manipulated sample. In case of an attack, this manipulated sample is stored in the database of the biometric system as a reference for one of the two subjects. Consequently, not only the subject linked to the identity is accepted by the biometric system, but also the other subject in the morph. This attack is particularly relevant for FRSs in which the user provides the reference. The most widely used systems in this context are automated border control systems, as in many countries a printed passport photo is handed over on passport application.

Part II

BACKGROUND

5 MACHINE LEARNING

Machine learning describes a sub-area of computational intelligence. In simplified terms it includes all algorithms that learn from previous observations and make decisions based on these observations [87]. Due to its universal applicability, machine learning has entered many fields of application.

In many areas of signal processing (which also includes image processing and thus, in the broadest sense, biometrics) machine learning based methods have become the standard approach for solving problems.

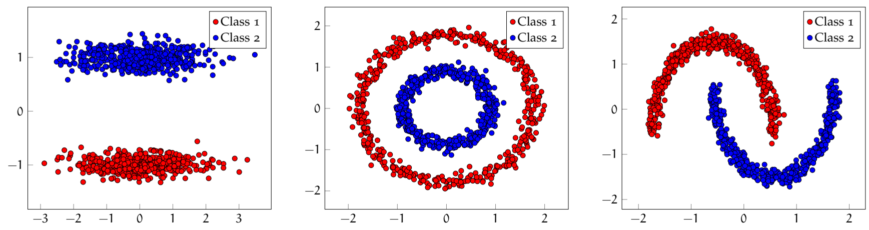
Machine learning can be divided into two categories: predictive and descriptive algorithms. Predictive algorithms, also referred to as classifier, learn dependencies between populations in order to assign new data points to a population based on this learned knowledge. Descriptive algorithms aim to describe the population, e.g. in order to cluster it. In biometrics predictive algorithms are mainly used, for example to decide whether two samples originate from one subject or from different ones. For the work presented in this dissertation only predictive algorithms were used, which are presented in this chapter.

To visualize the methodology of the different classifiers, the three two-dimensional data distributions shown in Figure 5.1 are employed. The linear separable data in Figure 5.1a represents the simplest case for classification. The examples in Figure 5.1b (circular) and Figure 5.1c (moon-shaped) are more complex to classify. The distributions shown are highly simplified. In real use cases, the data to be classified usually have much higher dimensions. Also, the data is only rarely as clearly separable as in the examples shown. Furthermore, the data could be prepared for a simpler classification (for example, transforming the circular distribution into a polar coordinate system). However, since the strengths and weaknesses of different classifiers are to be shown, this is deliberately omitted.

The classifiers described in this chapter are Support Vector Machine (SVM), Decision Tree, Random Forest, AdaBoost and Gradient Boosting.

5.1 SUPPORT VECTOR MACHINE

The SVM is the classifier most frequently used in biometrics. This is largely due to the fact that the SVM has high generalisation capabilities even with small amounts of data (if the hyperparameters are chosen correctly), as often encountered in biometrics. Simplified, the SVM



(a) Linear separable data (b) Circular separable data (c) Moon-shaped data

Figure 5.1: Exemplary two dimensional data distributions

separates space into two regions by the so-called hyperplane. The hyperplane is positioned such, that the distance to the data points of the different distributions (margin) is maximized and thus provides the best generalization capacity to unseen data [87][p. 1505].

The concept of the hyperplane is depicted in Figure 5.2. The two classes (red and blue) are to be separated by the SVM. For the positioning of the separating line, only the data points closest to the other class are relevant, which are termed support vectors (black). This concept can be extended from two-dimensional space to any number of dimensions. In the example shown, the separation can be represented by a straight line, in three-dimensional space it would be a plane, in even higher dimensions it is called a hyperplane.

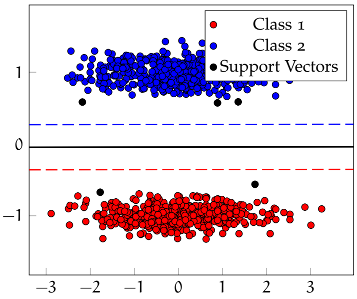


Figure 5.2: Example of the positioning of hyperplane and support-vectors of a 2-D SVM

Mathematically the hyperplane can be expressed as

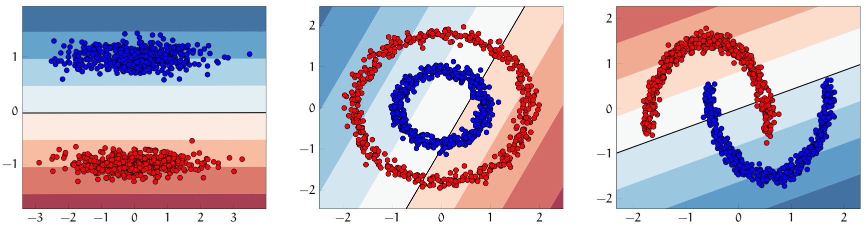
$$\boldsymbol{w}_0 \cdot \boldsymbol{x} + b_0 = 0, \tag{5.1}$$

whereas $\boldsymbol{w}_0$ is the normal-vector (weights), $\boldsymbol{x}$ the vector of training data and b_0 the offset (bias). For further information about the mathematical background of the hyperplane optimization the reader is referred to [21]. The equation for the classification of new data points is given as:

$$f(\boldsymbol{x}) = \boldsymbol{w}_0 \cdot \boldsymbol{x} + b_0. \tag{5.2}$$

For a data point which lies on the hyperplane the equation is 0, if the data point lies outside the plane, the equation takes a positive or negative value depending on which side the data point lies on.

As shown in equation 5.1 and visualized in Figure 5.2, the hyperplane is given by a linear function. Therefore, the basic SVM can only be used to separate linearly separable data. As shown in Figure 5.3, only the data distribution in Figure 5.3a can be successfully separated. The decision boundary is indicated by the black line. The darker the colour in the hatched areas, the higher the certainty with which the algorithm assigns a data point to the class with the respective colour. In case of the moon-shape distribution (Figure 5.3c) a considerable amount of data points are classified incorrectly as the hyperplane can no longer be positioned in a way that both classes are completely separated. In this case, the optimization tries to minimize the number of incorrectly classified data points. For the circular distribution (Figure 5.3c) the hyperplane can no longer be placed in a sensible way.



(a) Linear separable data separated by linear SVM

(b) Circular separable data separated by linear SVM

(c) Moon-shaped data separated by linear SVM

Figure 5.3: Classification examples of an SVM with linear kernel

To overcome this limitation of the SVM, it is possible to transform the input vector into a higher dimensional space using a transformation function ϕ:

$$\phi : \Re^n \to \Re^N, \tag{5.3}$$

whereas n is the dimension of the input vector and N the dimension of the feature space. For the classification of new data x, they are first transformed into the higher-dimensional, linearly separable space:

$$x \mapsto \phi(x). \tag{5.4}$$

Thus, the classification function given in equation 5.2 changes to

$$f(x) = \boldsymbol{w}_0 \cdot \phi(x) + b_0. \tag{5.5}$$

Furthermore, it can be shown that the normal vector $\boldsymbol{w}_0$ can be written as a linear combination of training samples [21]:

$$\boldsymbol{w}_0 = \sum_{i=1}^{l} y_i \alpha_i^0 x_i, \tag{5.6}$$

whereas l is the number of samples and $\alpha_i^0 > 0$ for support vectors. Inserted in equation 5.2, this gives the following equation:

$$f(\mathbf{x}) = \sum_{i=1}^{l} y_i \alpha_i^0 \mathbf{x}_i \cdot \mathbf{x} + b_0, \tag{5.7}$$

and for the transformed input data

$$f(\mathbf{x}) = \sum_{i=1}^{l} y_i \alpha_i^0 \phi(\mathbf{x}_i) \cdot \phi(\mathbf{x}) + b_0. \tag{5.8}$$

If ϕ is a positive definite function, the scalar product $\phi(\mathbf{x}_i) \cdot \phi(\mathbf{x})$ can be computed directly by the kernel function $K(\mathbf{x}_i, \mathbf{x}) = \langle \phi(\mathbf{x}_i), \phi(\mathbf{x}) \rangle$, which depends on the transformation function ϕ. The mathematical proof can be found in [21]. The implicit calculation is much less computationally demanding than a previous transformation of the input data. More information about finding and creating kernel functions can be found in [6] and [61]. In the following, the two most widely known and frequently used kernel functions are introduced.

5.1.1 *Polynomial Kernel*

The polynomial kernel maps the dot product to a polynomial function of arbitrary but fixed degree. The kernel function is given as:

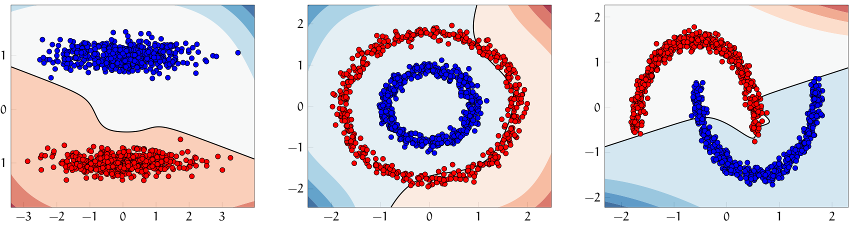
$$K(x_i, x_j) = (a x_i x_j + b)^d, \tag{5.9}$$

whereas d defines the degree of the polynomial function, a the coefficient and b is a free parameter [87][p. 1508]. The most important parameter is the degree of the polynomial. If the degree is chosen too low, the function may lack the necessary flexibility to classify the data correctly. If a too high degree is chosen, the training of the classifier becomes very complex and there is the risk of over-fitting.

Figure 5.4 shows an SVM with a polynomial kernel with $d = 5$, $a = 2$ and $b = 0$ on the example data from Figure 5.1. The linear data (Figure 5.4a) can be separated without false classifications. The circular data (Figure 5.4b) can be classified to some extent correctly, however a large percentage of the red class is falsely classified as blue. Increasing the degree of the polynomial function may give better results. The moon-shaped data (Figure 5.4c) are classified much more successfully than by the linear SVM. However, a subset of the data points (especially of the blue class) is classified incorrectly as well.

5.1.2 *RBF Kernel*

The Gaussian or Radial Basis Function (RBF) kernel implicitly maps the data points into a space of infinite dimensions. Thus the RBF



(a) Linear separable data separated by polynomial SVM

(b) Circular separable data separated by polynomial SVM

(c) Moon-shaped data separated by polynomial SVM

Figure 5.4: Classification examples of an SVM with polynomial kernel

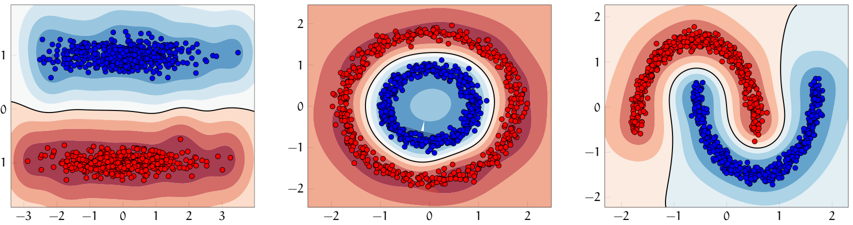
kernel contains all possible functions, or can approximate them. The kernel function is given as:

$$K(x_i, x_j) = e^{\frac{-||x_i - x_j||^2}{2\sigma^2}}, \tag{5.10}$$

whereas σ is the scalar parameter of the RBF (or *variance* of the Gaussian function) [16].

Intuitively described, the algorithm places a Gaussian function over each data point, where Sigma determines the variance of the Gaussian function. If σ is too large, the influence of the individual data points is smoothed too much, making the SVM no longer adaptable enough to model complex distributions. If the radius is too small, the SVM loses the possibility to generalize (over-fitting).

The ability of the SVM with RBF kernel to model any data distribution is shown in Figure 5.5. For the shown example, $\frac{1}{2\sigma^2}$ was set to 2. All three distribution types can be successfully separated by selecting meaningful hyperparameters. Even the circularly separable data in Figure 5.5b, which cannot be modelled by the two algorithms presented above, are correctly classified by the SVM with RBF kernel.



(a) Linear separable data separated by an SVM with RBF-kernel

(b) Circular separable data separated by an SVM with RBF-kernel

(c) Moon-shaped data separated by an SVM with RBF-kernel

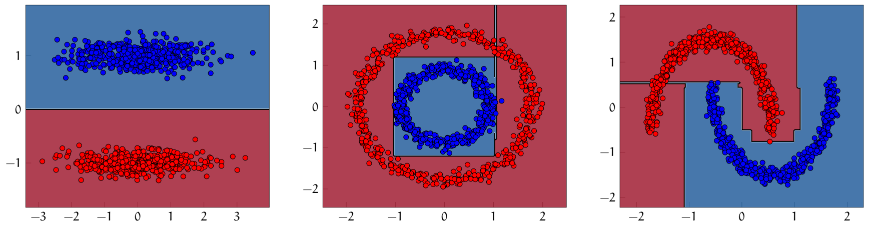
Figure 5.5: Classification examples of an SVM with RBF kernel

5.2 DECISION TREES

Decision trees have been known since the early days of the research on artificial intelligence. Decision trees can be used for example in training of expert systems [117]. However, they are less suitable for the application in this work, where classification problems usually have a high dimension of real-valued data [78]. Nevertheless, the algorithm is briefly described in this thesis, as it serves as a basis ensemble classifiers (e.g. Random Forest), which is described below.

The concept of decision trees is so intuitive that the classifier can be implemented by hand and without the need of training. A decision tree consists of nodes and leaves. The classification process starts at the root node. In each node, a decision for the further path in the tree is made based on usually only one attribute from the feature vector. From each node, at least two further nodes (or leaves) are accessible, the next node is selected depending on the previous comparison. Each path ends in a leaf, in which the result of the classification is defined [130]. The complexity of a decision tree can be controlled by a large number of parameters. These include the number of nodes, the number of branches per node, and the number of attributes considered per node.

The low-dimensional example data shown in Figure 5.1 can be successfully modelled from a decision tree. Figure 5.6 visualizes the decision of a decision tree without limitations for depth or number of attributes per node. The linear data distribution (Figure 5.6a) can be separated without errors. Circular (Figure 5.6b) and moon-shaped (Figure 5.6c) data distributions can also be separated, but training artefacts occur and round shapes cannot be modelled adequately.



(a) Linear separable data separated by a Decision Tree

(b) Circular separable data separated by a Decision Tree

(c) Moon-shaped data separated by a Decision Tree

Figure 5.6: Classification examples of a Decision Tree

5.3 ENSEMBLE CLASSIFIER

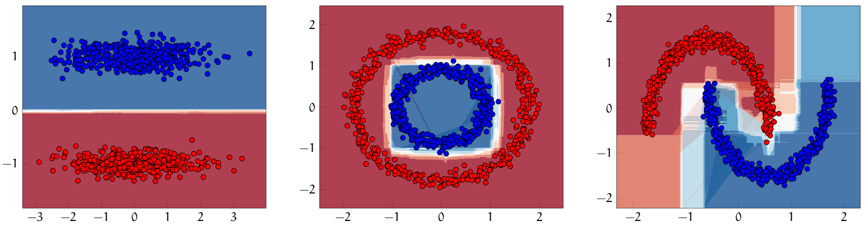
In [155], James Surowiecki states, that decisions based on the aggregation of information in groups often supersedes the decisions made by a single member of the group. This insight is the fundamental motivation for ensemble classifiers. Ensemble classifier is a collective term for all classifiers that evaluate the results of several, separate classifiers and derive a common result. At first glance, this method increases the computational effort because several classifiers have to be trained. But it is also capable of causing a significant reduction of complexity for the individual classifier. An example of this are Decision Stumps, which are explained in more detail in Section 5.3.2. Thus it is possible to obtain a higher robustness of the overall classifier. Under certain circumstances this enables solutions a single classifier might not be capable of.

In this section three common ensemble classifiers are introduced, namely Random Forest, AdaBoost and Gradient Boosting.

5.3.1 *Random Forest*

As the name indicates, this Ensemble Classifier is a collection of several of the decision trees presented in Section 5.2. The concept was described by Breimann in [13]. For the training of each individual classifier, a random subset is selected from the training data. Thus, each individual tree has different information available for training, resulting in an individual tree. In [13] the decision is made by a majority voting, which means that each individual classifier votes for a single class and the class with the most votes is chosen by the ensemble classifier as result. In state-of-the-art implementations[1], however, the probabilistic prediction of the individual classifiers is usually averaged, as this considers the certainty of the individual result.

Figure 5.7 shows the classification result of a random forest classifier with 100 trees. The parameters of each tree are equal to those of the tree visualized in Figure 5.6. For the linearly separable data (Figure 5.7a) no significant difference can be seen. For the more complex cases



(a) Linear separable data separated by a Random Forest Classifier

(b) Circular separable data separated by a Random Forest Classifier

(c) Moon-shaped data separated by a Random Forest Classifier

Figure 5.7: Classification examples of a Random Forest Classifier

[1] For example the implementation of scikit-learn used in this thesis: https://scikit-learn.org/stable/modules/ensemble.html#random-forests

(Figure 5.7b and 5.7c), however, it can be observed that the shape of the data distribution can be modeled better than with a single tree. Even if the number of misclassifications does not differ significantly, it can be concluded that the random forest has a higher generalization power.

5.3.2 *AdaBoost*

The concept of AdaBoosting was proposed by Freund in [42]. It is not a classifier, but a method to create a strong algorithm from several so-called weak learners. The choice of the weak learner is not fixed. The concept was already known before as Boosting, but AdaBoost extends it. The basic idea is that in training each weak learner, the errors of the previous weak learner are weighted higher. The process of *ada*ptive *boost*ing gives AdaBoost its name.

The boosting process is executed iteratively. The number of iterations depends on the number of weak learners to be trained. Initially, a weight vector $\boldsymbol{w}^1$ is created:

$$w_i^1 = D(i) \text{ for } i = 1, \dots, N, \tag{5.11}$$

whereas D is the distribution and N the number of training data. The following process is then carried out sequentially for all T weak learner, the current iteration is given as t. First the weighted distribution $\mathbf{p}^t$ is calculated using the weight vector $\boldsymbol{w}^t$:

$$\mathbf{p}^t = \frac{\boldsymbol{w}^t}{\sum_{i=1}^N w_i^t}. \tag{5.12}$$

Subsequently the weighted distribution $\mathbf{p}^t$ is evaluated by the weak learner, leading to the hypothesis-vector $\mathbf{h}^t$. The resulting error ϵ_t is determined by calculating the *Sum of Absolute Differences* between the hypothesis $\mathbf{h}^t$ and the ground truth of the data distribution $\mathbf{c}$:

$$\epsilon_t = \sum_{i=1}^N p_i^t |h_t(i) - c(i)|. \tag{5.13}$$

In order to weight the individual weak learners within the ensemble classifier according to their error, this error is stored in the vector $\boldsymbol{\beta}$:

$$\beta_t = \frac{\epsilon_t}{1 - \epsilon_t}. \tag{5.14}$$

The weight vector is then recalculated:

$$w_i^{t+1} = w_i^t \beta_t^{1-|h_t(i)-c(i)|}. \tag{5.15}$$

The higher the previous error ($|h_t(i) - c(i)|$) for a data point $D(i)$, the more its future weight w_i^{t+1} is increased.

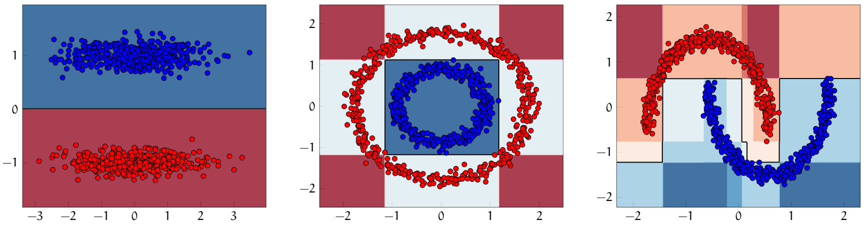
Once all weak learners have been iterated, the final ensemble classifier can be constructed. The classification of a data point $D(i)$ is calculated according to the following equation:

$$h_f(i) = \begin{cases} 1, & \sum_{t=1}^{T}(\log\frac{1}{\beta_t}h_t(i)) \geqslant \frac{1}{2}\sum_{t=1}^{T}\log\frac{1}{\beta_t} \\ 0, & otherwise \end{cases} \tag{5.16}$$

The system evaluates whether the weighted sum of the response of all weak learners is above the threshold value $\frac{1}{2}\sum_{t=1}^{T}\log\frac{1}{\beta_t}$. The derivation of the threshold formula can be found in [42].

As already mentioned, an arbitrary weak learner can be applied. If a learner is chosen too complex, the boosting process becomes very computationally intensive, since a new learner is trained in each iteration. Furthermore this can lead to over-fitting. According to [42] AdaBoost is able to convert "a *weak* Probably Approximately Correct (PAC) learning algorithm that performs just slightly better than random guessing into one with arbitrarily high accuracy." In state-of-the-art implementations[2], decision stumps are commonly used as weak learners. A decision stump can be described as a binary decision tree with a depth of one. Thus it is only able to binary separate the data distribution in one dimension. But by combining several classifiers of this type, it is eventually possible to solve problems of arbitrary complexity.

The classification results for such an AdaBoost classifier consisting of 100 decision stumps are shown in Figure 5.8. The linearly separable data (Figure 5.8a) can be easily classified (in this particular case one decision stump would be sufficient). The circular separable data (Figure 5.8b) can be separated to a large extent, but the algorithm is struggling to model the circular shape of the data distribution. The moon-shaped data distribution shown in Figure 5.8c is reproduced quite accurately and missclassifications are quire rare.



(a) Linear separable data separated by AdaBoost with Decision Stumps

(b) Circular separable data separated by AdaBoost with Decision Stumps

(c) Moon-shaped data separated by AdaBoot with Decision Stumps

Figure 5.8: Classification examples of AdaBoost

[2] For example the implementation of scikit-learn used in this thesis: https://scikit-learn.org/stable/modules/ensemble.html#adaboost

5.3.3 *Gradient Boosting*

Gradient Boosting is a generalised adaptation of AdaBoost proposed by Friedman in [43]. Like AdaBoost, Gradient Boosting is based on training several weak learners to create a robust, more general classifier. In contrast to AdaBoost, which aims to minimize the error given in equation 5.13, Gradient Boosting can be used to minimize any differentiable loss function $L(y, F)$. Thus, Gradient Boosting can be considered as a generalization of AdaBoost.

Gradient Boosting is optimized in an iterative process on the training data $(\mathbf{x}, \mathbf{y})$ with N samples. To do this, the constant function F_0 is initialized first (for example as the mean value of the data points). Then M iterations start, where m is the counter of the current iteration. First, the loss function $L(y, F(\mathbf{x}))$ is derived with respect to F, obtaining the so-called pseudo-residuals $\tilde{\mathbf{y}}$:

$$\tilde{y}_i = -\left[\frac{\partial L(y_i, F(x_i))}{\partial F(x_i)}\right]_{F(x)=F(m-1(x))}, \ i = 1, \dots, N. \tag{5.17}$$

A weak learner $h_m(x)$ is trained on these pseudo-residuals. This step can be interpreted as the approximation of the gradient of the steepest ascend of the loss function. Next, the step size in the direction of the previously calculated gradient is calculated by minimizing the sum of the loss function $L(y_i, F_{m-1}(x_i))$ and the weighted weak learner $\rho h_m(x_i)$:

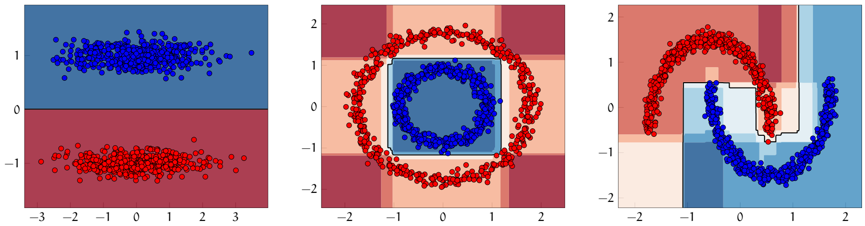
$$\rho_m = \arg\min_{\rho} \sum_{i=1}^{n} L(y_i, F_{m-1}(x_i) + \rho h_m(x_i)). \tag{5.18}$$

In the last step the gradient boosting model is updated by adding the weak learner weighted by ρ_m to the previous model:

$$F_m(\mathbf{x}) = F_{m-1}(\mathbf{x}) + \rho_m h_m(\mathbf{x}) \tag{5.19}$$

Gradient Boosting is often used with decision trees as weak learners. For this special case an adapted version of Gradient Boosting was proposed in [44], referred to as Gradient Tree Boosting or Gradient Boosted Decision Trees. The basic algorithm of gradient boosting is maintained, but the training of the weak learner is changed. The training data is divided into as many disjoint regions as there are leaves in the decision tree to be trained. For each region, a constant value is predicted by the decision tree. In addition, not one ρ_m per weak learner, but a separate ρ_m per region is calculated.

Figure 5.9 visualizes the classification results of a gradient tree boosting classifier with 100 decision trees of a maximum depth of three. The linear separable data (Figure 5.9a) can be classified without errors. The circularly arranged data in Figure 5.9b shows that the classifier tries to reproduce the round shape. The moon-shaped data (Figure 5.9c) can also be separated almost without error.



(a) Linear separable data separated by Gradient Tree Boosting

(b) Circular separable data separated by Gradient Tree Boosting

(c) Moon-shaped data separated by Gradient Tree Boosting

Figure 5.9: Classification examples of Gradient Tree Boosting

5.4 NEURAL NETWORKS

The inspiration for Neural Networks (NNs) as machine learning algorithms is the structure of the mammal brain. The first steps in this direction were already taken in the 1960s by Rosenblatt [128]. He developed the Perceptron, which simulates the signal processing in a neuron of the mammal brain in a highly simplified way. The structure of a Perceptron is shown in Figure 5.10. On the left side, binary input data x is provided. Each input x_i is weighted by the Perceptron with weight w_i. Subsequently, the weighted inputs are summed up. If the sum exceeds the threshold value τ defined in the Perceptron the output is activated (set to 1). The behaviour of the Perceptron, referred to as activation function, can be formulated as follows [104]:

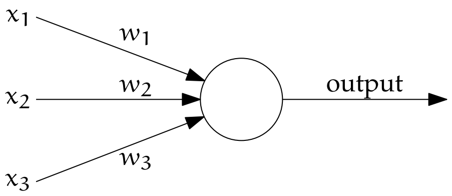


Figure 5.10: Schematic visualization of a Perceptron

$$\text{output} = \begin{cases} 0, & \sum_j w_j x_j \leqslant \tau \\ 1, & \sum_j w_j x_j > \tau \end{cases} \tag{5.20}$$

The weighting and summation of the input values can be expressed as a scalar product of the weight vector $\boldsymbol{w}$ and the input vector $\boldsymbol{x}$. Furthermore, the negative threshold value can be interpreted as a bias ($b = -\tau$), changing equation 5.20 to:

$$\text{output} = \begin{cases} 0, & \boldsymbol{w} \cdot \boldsymbol{x} + b \leqslant 0 \\ 1, & \boldsymbol{w} \cdot \boldsymbol{x} + b > 0 \end{cases} \tag{5.21}$$

Only the simplest linear correlations can be represented by the shown model. In the human brain, about 85 billion neurons are linked together [7], which can not be modelled by current computers. However, the concept can be transferred to the Perceptron. Figure 5.11 shows a so-called Multi-Layer Perceptron (MLP). This consists of a layer of input nodes, which has one node per feature of the input vector. The next layers are the so-called hidden layers. These can contain almost any number of nodes at any depth (multiple layers). At the end, the networks has an output layer, containing as many nodes as the output vector contains values. For example, if the NN is used for classification, the network must have as many output nodes as classes available. If a NN has at least 3 layers (input, hidden and output), it can model any function [22].

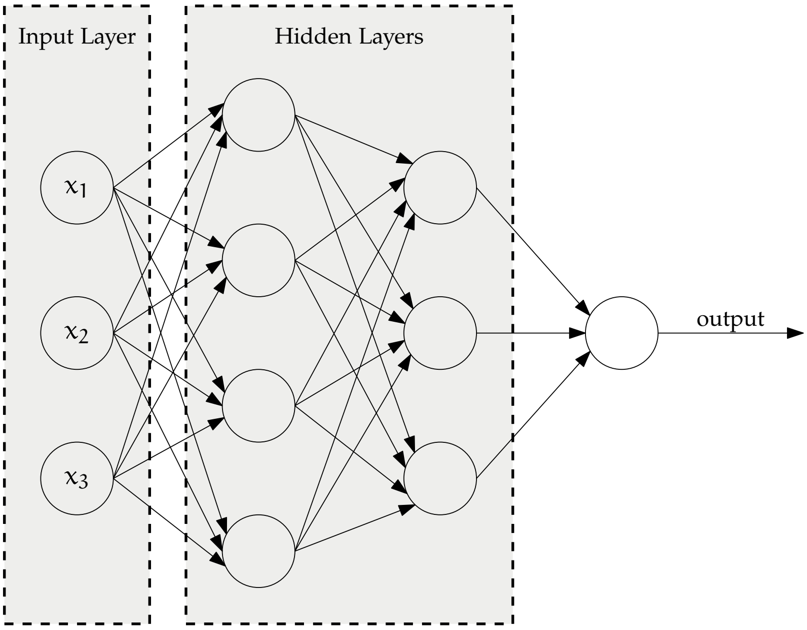


Figure 5.11: Schematic visualization of an MLP

In order to adapt a NN to a specific application, the weights of the nodes are adjusted to minimize a previously determined error (the loss function). Further information, in particular the mathematical background for the training process of NNs can be found in [104].

The aforementioned concept introduces the basics of NNs. Even though the concept is still valid today, many optimizations and adaptations to specific use cases have been proposed over time. For example, the activation functions of individual neurons have been revised. The perceptron can only produce a binary output. Thus, a small change of the input vector can produce a completely different result. To mitigate this behavior it is possible to soften the threshold by using a sigmoid function instead of the binary decision, which is defined as follows:

$$\sigma(z) = \frac{1}{1+e^{-z}} \tag{5.22}$$

Thus, the calculation of the output value given in equation 5.21 changes to

$$\sigma(\boldsymbol{w} \cdot x + b) = \frac{1}{1 + e^{-\sum_j w_j x_j - b}}. \tag{5.23}$$

This adaptation avoids an abrupt change of the output value from 0 to 1 (or vice versa) at the threshold value, but rather results in a smooth transition from one value to another. However, the sigmoid function also has disadvantages. The derivative is complicated and the calculation is time consuming, in particular for NNs with multiple layers and nodes. For these reasons, most current NNs use the *Rectifier* suggested in [53] as activation function. A node with this activation function is referred to as Rectified Linear Unit (ReLU). The *Rectifier* is a highly simplified activation function, which is defined as

$$f(x) = max(0, x), \tag{5.24}$$

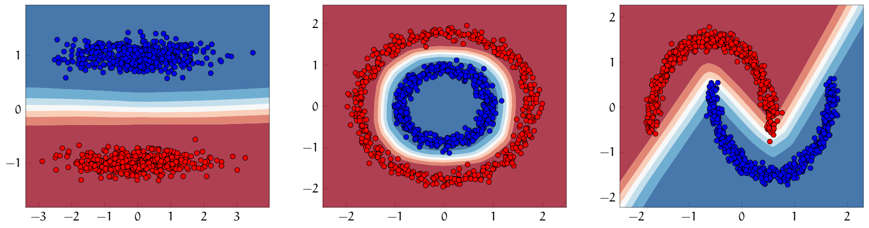
modelling a ramp function starting at 0.

In this way, different types of neurons can be designed and also combined with each other. An example are the so-called Convolutional NNs (CNNs). These networks, which are frequently used in image processing [85], are usually convoluting the input vector with a matrix in the first layers of the network. Depending on the type of matrix, different operations like smoothing or bandpass filters can be performed.

In addition to changes to the activation function, new architectures are regularly proposed for linking the neurons. Due to the increasing performance of computers a trend has developed to design Deep NNs (DNNs) (more hidden layers). These networks achieve unprecedented results in solving complex problems, e.g. speech-to-text conversion [144], object detection [32], image super resolution [76] or face recognition [164]. However, the necessity of this depth is controversial [8]. The NNs used in this thesis are introduced in more detail in the respective chapters.

Examples of the classification results of a MLP with 3 layers (1 hidden layer) are given in figure 5.12. The hidden layer consists of 100 neurons, the input layer consists of 2 neurons (x and y values of the data points) and the output layer consists of 2 neurons, one per class. The linearly separable data (Figure 5.12a) can be easily separated. For the circulating separable data (Figure 5.12b) the data distribution is modelled very precisely. For the data with moon shaped distribution (Figure 5.12c), the data is correctly separated, but the shape of the distribution is not modelled correctly.

In the example shown above, 400 weights already have to be found. With increasing complexity of the network and increasing size of the input vectors, the number of weights to be trained further increases. In order to train these algorithms robustly, large amounts of training



(a) Linear separable data separated by a NN

(b) Circular separable data separated by a NN

(c) Moon-shaped data separated by a NN

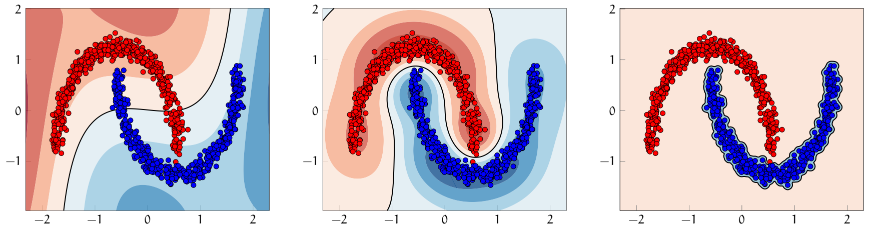
Figure 5.12: Classification examples of NNs

data are required, which are not available for the topic discussed in this thesis, thus, no NNs are trained. However, pre-trained NNs are used, e.g. to extract features from images. These are described in the respective chapters.

5.5 MACHINE LEARNING RELATED ISSUES

Machine learning is an integral part of modern information processing and thus also of biometrics. However, machine learning also has some issues. The severest ones will be described in this chapter.

One problem is over- and underfitting of classifiers. In Figure 5.13, the data distribution introduced in Figure 5.1c was made more difficult to separate by pushing the distributions together. In the shown example, an SVM with RBF kernel was applied. If a small hyperparameter $\frac{1}{2\sigma^2}$ is chosen, the flexibility of the SVM is not sufficient, as depicted in Figure 5.13a. The SVM is no longer able to model the strong bending of the data distribution. This behaviour is called underfitting. If we train with the parameters used in Subsection 5.1.2, the data can still be separated except for one outlier, as shown in Figure 5.13b. If the hyperparameter is further increased to correctly classify the last outliers of data points, this leads to the SVM visualized in Figure 5.13c.



(a) Underfitting SVM with RBF-kernel and $\frac{1}{2\sigma^2} = 0.1$

(b) Good fitting SVM with RBF-kernel and $\frac{1}{2\sigma^2} = 2$

(c) Overfitting SVM with RBF-kernel and $\frac{1}{2\sigma^2} = 500$

Figure 5.13: Examples of under- and overfitting SVMs

Here all data points are correctly classified, as the hyperplane exactly models the contour of the blue distribution. However, it must be taken into account that other data will contain a different variance. Thus, new data is likely to be misclassified due to the lack of generalisation capability of the model shown [55].

The appropriate hyperparameters differ depending on the training data and the task at hand, which means that they have to be determined according to the application. This leads to another problem of machine learning algorithms. For the example shown in Figure 5.1, an appropriate hyperparameter can be found by visual inspection. In real-world use cases, however, the training data is rarely two or three dimensional but, as shown in the following chapter, usually feature vectors with several 100 dimensions. In these cases, the trained algorithms cannot be visualized anymore. To avoid undiscovered errors in the algorithm, it is important to ensure that the data is well structured.

In general, machine learning algorithms are mainly dependent on the training data. If the variance of the data to be classified is missing in the training data, the distribution cannot be modelled correctly. Furthermore an over-representation of one class can lead to an unbalanced classifier [9]. In extreme scenarios, such as a massive over-representation of class 1, a classifier could be trained to always return class 1 as this results in the smallest error during training.

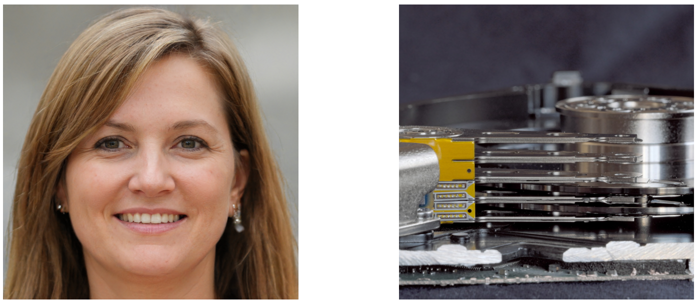
If training and evaluation are performed on the same data set, unnaturally good classification results are obtained, due to the absence of variance between training and test data. In a real scenario, however, the data to be classified is unknown, meaning a natural variance has to be expected. In order to simulate the condition of unknown data, a strict separation of training and test data has to be established, ensuring that the results of the evaluation are as close as possible to the expected classification performance on realistic data.

6 IMAGE DESCRIPTORS

The classifiers described in Chapter 5 are able to distinguish data on the basis of feature vectors. Biometrics is based on the processing of information from signals, which are usually present as images. In order to process the information contained in the images by algorithms (explicitly defined by an engineering approach or implicitly trained by machine learning), they must first be extracted from the images. Feature vectors extracted from images are collectively referred to as *Image Descriptors* [50]. In this chapter different methods are presented, which were used in this thesis to access different kinds of information from the images, which can be categorised as follows:

- Texture Descriptors
- Gradient Based Descriptors
- Keypoint Descriptors
- Landmark Extractors
- Image Noise Pattern
- Deep Features

For each feature extractor a visualization of the extracted information is shown. The two images shown in Figure 6.1 serve as a consistent example. As this thesis focuses on the processing of facial images,



(a) Face Image (b) Non-Facial Image

Figure 6.1: Example images used to visualize image descriptors

Figure 6.1a depicts a female face. In order to avoid licensing issues, the example image was generated by a Generative Adversarial Networks (GAN) [72]. To show the applicability of the feature extractors to other image contents, a natural image (macro shot of a hard disk read head, Figure 6.1b) is used in addition.

In the following, an image is represented independently of the previous image format as a three-dimensional matrix with the size: (image height × image width × color channels). Since the presented methods for feature extraction only work in the two-dimensional space, the grey values of the images (as a two-diemensional matrix) are used in the shown examples. However, by applying the feature extractors separately on each colour channel, they can also be adapted to colour images, increasing the length of the feature vector by factor of the number of colour channels.

6.1 TEXTURE DESCRIPTORS

Texture is one of the most important characteristics of images and has been analysed since the early days of computer vision [89]. Many fundamental tasks of image processing can be accomplished on the basis of texture analysis, for example object recognition [70] or edge detection [56]. A detailed overview of existing algorithms can be found in [89].

In this section, the two most frequently used texture descriptors in biometrics are described in detail, namely Local Binary Patterns (LBP) and Binarised Statistical Image Features (BSIF).

6.1.1 *Local Binary Patterns*

LBP was first proposed by Ojala et. al in [106]. The concept of feature extraction is illustrated in Figure 6.2.

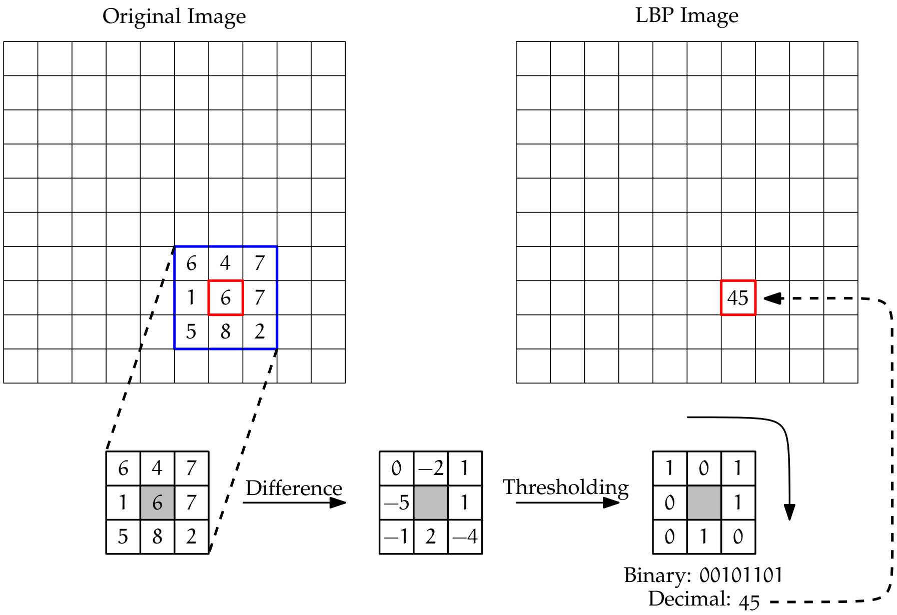
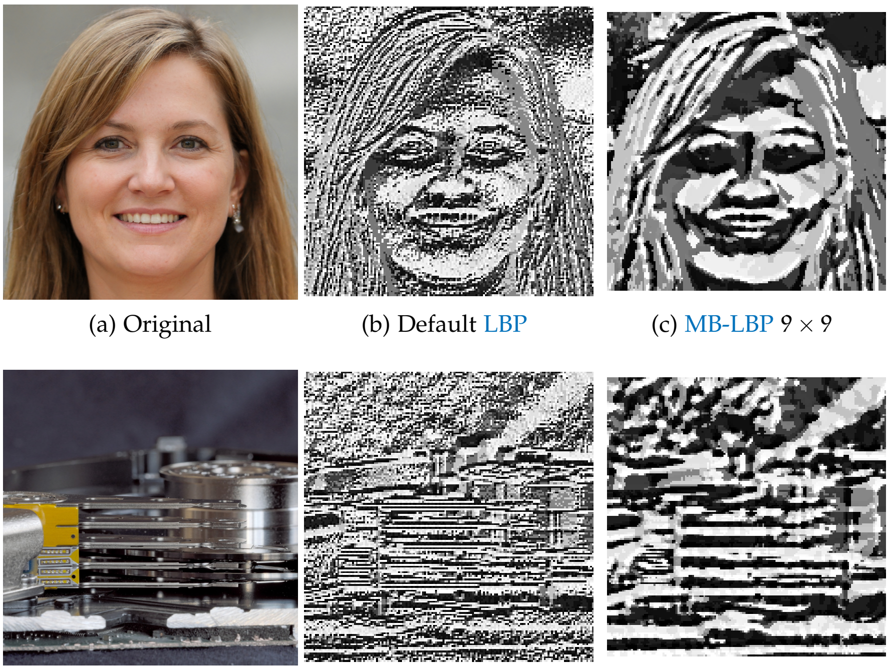


Figure 6.2: Schematic visualization of the process of LBP extraction

For each pixel of the analysed image the difference to the neighbouring pixels is calculated. The differences are then binarised, negative values are mapped to 0, positive values to 1. The resulting sequence of binary values is interpreted as a decimal value, ranging from 0 to 255 (2^8). This process is repeated for each pixel of the analysed image, whereas for pixels located on the edge, it must be taken into account that these do not have a neighboring pixel. In order to calculate an LBP value for these pixels, the lowest row of the image could be used as neighboring pixels. Examples of extracted LBP values are given in Figure 6.3b and 6.3e. In the shown example it can be seen that similar textures lead to similar LBP values and therefore similar grey values in the example image.


(a) Original

(b) Default LBP

(c) MB-LBP 9×9

(d) Original

(e) Default LBP

(f) MB-LBP 9×9

Figure 6.3: Example images of LBP

The generated LBP image is equal to the size of the original image. To compress the information into a compact feature vector, a histogram with 255 bins of LBP values is created. The feature vector thus represents the quantity of occurrences of a certain texture pattern in an image.

A major advantage of LBP is the easy to use concept and the low computational effort required to calculate the feature vector. However, LBP is not robust against rotations (in a rotated image other LBP patterns are detected) and scaling. A further disadvantage is the fixation on neighboring pixels, which can be solved by using Multi-Scale Block LBP (MB-LBP) [88].

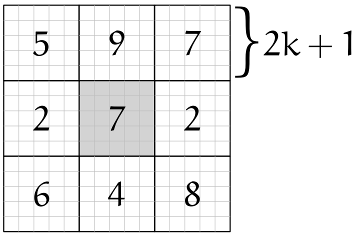


Figure 6.4: Example of a MB-LBP patch

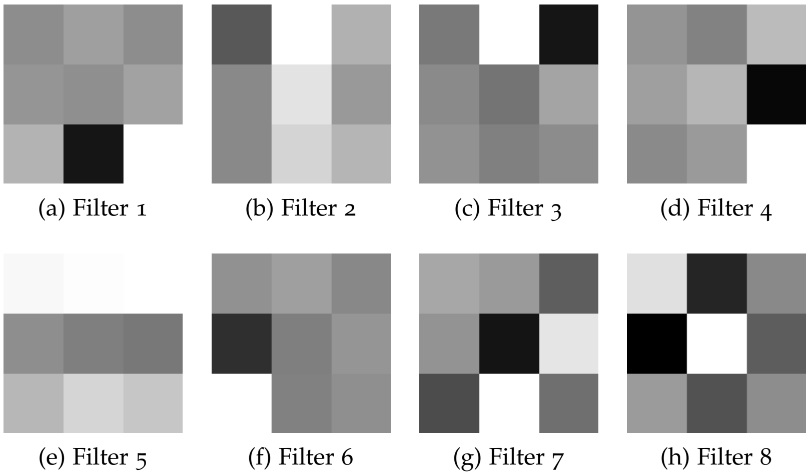
For MB-LBP, the basic concept of Basic LBP is used. However, the comparison operator for individual pixels of the 3×3 neighborhood is replaced by the comparison of averaged values of subregions. A MB-LBP patch with subregions of 5×5 pixels is shown in Figure 6.4. If the average value was calculated for each subregion the subsequent process is the same as for the base LBP. It is important to note that the MB-LBP Patch needs a central pixel to position it on the image. Thus, a subregion may only have an edge length of $2k+1$. Examples of result images of the MB-LBP with a patch size of 9×9 pixels (i.e. subregions of size 3×3) are given in Figure 6.3c and 6.3f. It can be observed that in comparison to the basic LBP in Figure 6.3b and 6.3e the regions are much more homogeneous, due to the smoothing effect of the larger MB-LBP patch. Thus, a larger patch size can lead to an increased robustness of the feature extractor, but with the loss of high frequency information.

6.1.2 *Binarized Statistical Image Features*

BSIF, proposed by Kannala et al. in [71] is based on the LBP described in Section 6.1.1. The main differences are found in the calculation of single patches. In contrast to LBP, where the calculation of the patch was constructed heuristically, the BSIF patch (or filter) is based on statistics of natural images.

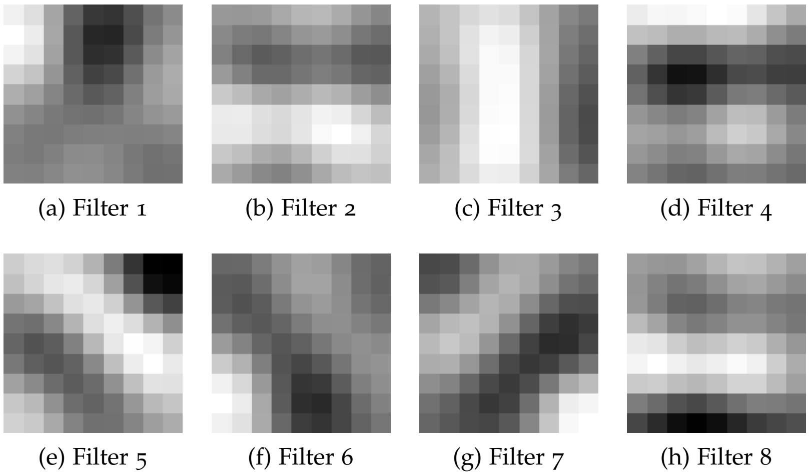
As the BSIF filter is not divided into subregions of equal size like the LBP patch, it can be created in arbitrary sizes with odd edge length. In order to create new BSIF filters, Independent Component Analysis (ICA) is used to find filters representing the differences in given training images most accurately. A detailed description of the creation of new filters is given in [71]. Training new filters using images related to the problem (in this case facial images) can lead to a loss of the generalisation ability of the filters. For this reason, the pre-trained filters provided by [71] are used in this paper.

Figure 6.5 shows a set of 8 BSIF filters of 3×3 pixels, directly comparable to the base LBP patch. An example of a set of larger filters (9×9 pixels) is given in Figure 6.6. For each pixel of the analysed image, the response of each filter in the filter set is calculated by convolution. Thus, for the filter set shown in Figure 6.5, 8 filter responses



(a) Filter 1
(b) Filter 2
(c) Filter 3
(d) Filter 4
(e) Filter 5
(f) Filter 6
(g) Filter 7
(h) Filter 8

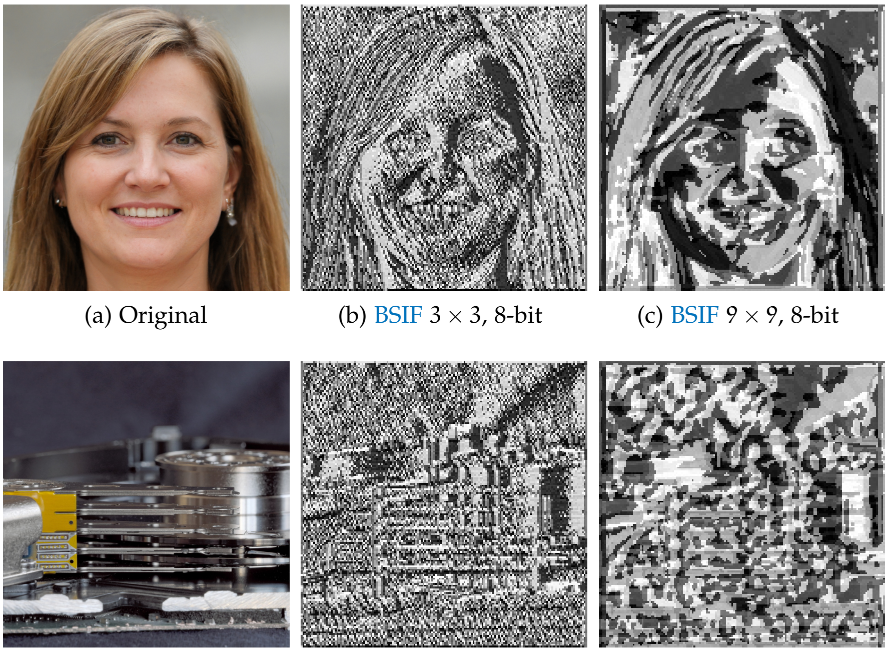
Figure 6.5: BSIF filters for 3 × 3, 8-bit



(a) Filter 1
(b) Filter 2
(c) Filter 3
(d) Filter 4
(e) Filter 5
(f) Filter 6
(g) Filter 7
(h) Filter 8

Figure 6.6: BSIF filters for 9 × 9, 8-bit

are obtained per pixel of the analysed image. In the further process, the filter responses are binarised and negative values are mapped to 0, positive values to 1. The resulting sequence of binary values per pixel is interpreted as a decimal value, in the case of the set of 8 filters it is ranging from 0 to 255 (2^8). To create the feature vector, the decimal values are transferred to a histogram with a length directly depending on the number of filters present in the filter set, in this case $2^8 = 256$. The number of filters in the filter set can be arbitrarily selected during training. As a general rule, the information represented in the feature vector increases as the number of filters increases. However, this can lead to unintentionally large feature vectors. The filter sets provided by [71] contain 5 to 12 filters, whereas the 12 filters already result in a feature vector of $2^{12} = 4096$ dimensions. Using a larger number of filters might not be beneficial in most cases.



(a) Original (b) BSIF 3×3, 8-bit (c) BSIF 9×9, 8-bit

(d) Original (e) BSIF 3×3, 8-bit (f) BSIF 9×9, 8-bit

Figure 6.7: Example images of BSIF

The BSIF images generated by the filter set shown in Figure 6.5, depicted in Figure 6.7b and 6.7e, are directly comparable to the result of the basic LBP in Figure 6.3b and 6.3e. As with LBP, areas with similar texture produce similar grey values in the BSIF image. If a filter set with larger filters is used (Figure 6.7c and 6.7f), a smoothing effect is achieved. As the filters are trained to the appropriate size, more detailed information can be extracted than with MB-LBP (see Figure 6.3c and 6.3f).

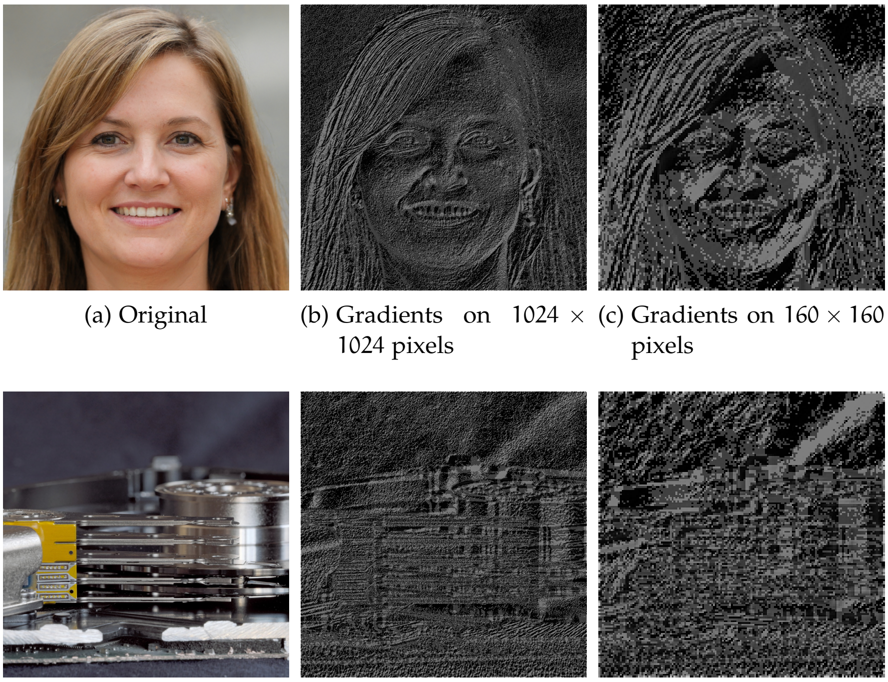
6.2 GRADIENT BASED DESCRIPTORS

The methods presented so far analyse the textures of images. Another possibility to extract information from images is the analysis of changes in the information (the frequency). This is done by calculating the gradients, the partial derivatives of the image. In this section, first the calculation of the gradient is described and then the more advanced Histogram of Oriented Gradients (HOG) feature extractor is introduced.

6.2.1 *Gradients*

The calculation of gradients (calculation of partial derivatives) is a basic mathematical operation. In signal processing the gradient represents the changes of the signal at the given position. For the processing of multidimensional matrices the gradient defined for one-dimensional signals must be adapted. In standard implementations[1] the gradients are calculated for each dimension of the analysed matrix. Thus, two gradient images are generated per grey value image.

Examples of the resulting gradient images are given in Figure 6.8. For a better representation, the values of the gradients were averaged over both gradient value images (horizontal and vertical). Figure 6.8b and 6.8e show the calculated gradients on the images in their original size (1024 × 1024 pixels and 626 × 626 respectively). If the images to be analysed are scaled to 160 × 160 pixels prior to the gradient calculation (the corresponding gradient images are shown in Figure 6.8c and 6.8f), the resulting gradient image (which in turn only consists of 160x160 pixels) shows a significantly coarser structure and the differences between areas with high and low differences are increased. Please note that for the creation of the gradient images the gradient values were stretched in order to obtain visible grey values. Thus, the grey values of the gradient images produced on the original and scaled images are not directly comparable as the stretching factors might differ.



(a) Original

(b) Gradients on 1024 × 1024 pixels

(c) Gradients on 160 × 160 pixels

(d) Original

(e) Gradients on 626 × 626 pixels

(f) Gradients on 160 × 160 pixels

Figure 6.8: Example images of Gradient

The gradient images shown provide a representation of the changes in the image signal. Strong changes result in high values, weaker

[1] for example the implementation for the gradient calculation used in this thesis: https://docs.scipy.org/doc/numpy/reference/generated/numpy.gradient.html

changes in lower values. Thus, in the more homogeneous image areas (e.g. background), considerably darker gray values (lower gradient values) can be seen in the gradient image. A smoothed image with less high frequencies will thus result in lower gradients.

6.2.2 *Histogram of Oriented Gradients*

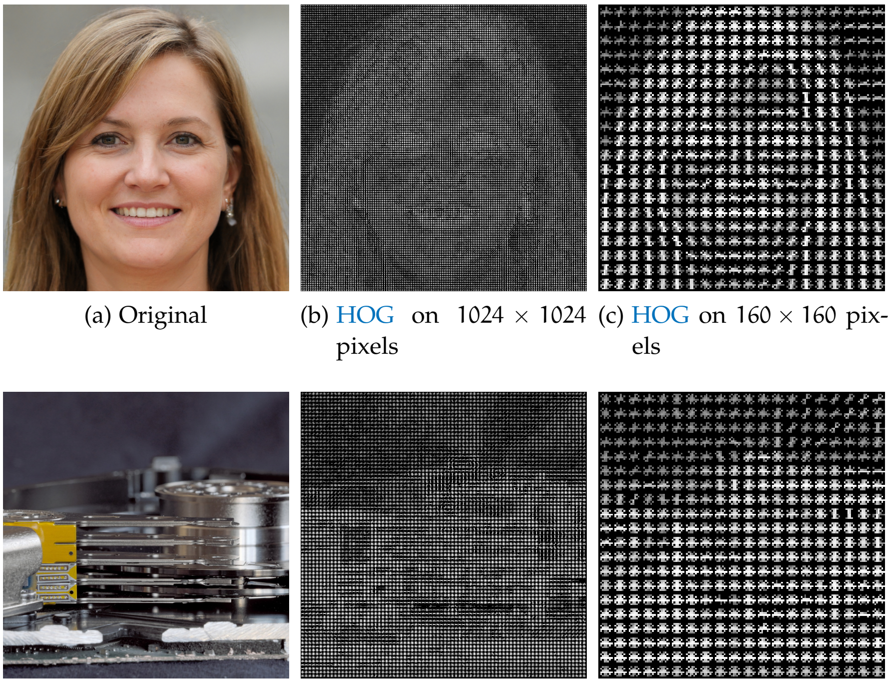
HOG is a significantly extended representation of the gradients of an image. The method was first mentioned in 1982 in an US patent [101], but the name HOG was introduced after the expiration of the patent by Dalal in [23]. By now, there are many tasks in which HOG is successfully applied, including human detection [23], object detection [99] and face recognition [30].

In order to compute HOG, the gradients for row and column of the analysed image are determined as described in Section 6.2.1. For each pixel, the direction and magnitude of a combined gradient is estimated based on the previously calculated gradients. Subsequently, the image is divided into fixed size cells, e.g. 8×8 pixels. For each cell, a HOG contained in the pixels of the cell is calculated. The histogram has a fixed width, e.g. 8, so that the direction vectors are discretised into 8 directions. The magnitude of the direction vector is used as value in the histogram. The calculated HOG values for the example images in original size are given in Figure 6.9b and 6.9e, for the down-scaled images in Figure 6.9c and 6.9f. On the smaller images the shown results are better recognisable. For each cell, the values contained in the histogram are plotted in the respective direction, resulting in a star shape (in this case comprising 8 rays). The 160×160 pixel image contains a subdivision into 20×20 cells for cells of size 8×8 pixels. With a higher pixel density of the analysed image, the number of cells and thus the number of stars displayed increases.

To achieve a higher robustness against different lighting conditions and contrasts, the histograms are normalised in blocks. A block contains a fixed number of cells, e.g. 3×3. In this block, the histograms are normalised based on the energy of the histograms in the block and subsequently stored in the feature vector. This process is repeated per cell. Depending on the selected parameters, this might lead to a cell being included and normalised in several blocks and thus being represented in the feature vector in different ways of normalisation. This is expected to provide additional stability against changes in contrast and lighting conditions.

6.3 KEYPOINT DESCRIPTORS

The two classes of feature extraction algorithms presented so far represent holistic approaches, which extract all pixel based information independent of the image content. Keypoint descriptors, in contrast,



(a) Original (b) HOG on 1024×1024 pixels (c) HOG on 160×160 pixels

(d) Original (e) HOG on 626×626 pixels (f) HOG on 160×160 pixels

Figure 6.9: Example images of HOG

first identify points of interest in the image and explicitly analyse the area around them. The advantage of this approach is that the feature vector is likely to contain mostly relevant information. However, the disadvantage is, that the feature vector may vary in length, depending on the number of points detected, which implies that no simple distance metrics can be used for comparison. This section describes the two commonly used keypoint descriptors Scale-Invariant Feature Transform (SIFT) and Speeded Up Robust Features (SURF).

6.3.1 *Scale-Invariant Feature Transform*

SIFT was proposed by Lowe in 1999 [92]. The extracted features are, within certain limits, robust against translation, rotation and scaling, as well as other variations in the image, e.g. lighting conditions [93]. Due to the robust properties of the feature extractor, it was successfully used over many years in applications like video stabilization [60] or object recognition [83], but also in the context of biometrics, for example for comparisons of fingerprints [110] or iris images [173].

The process of SIFT feature vector calculation can be divided into four steps: Scale-space extrema detection, keypoint localization, orientation assignment and keypoint descriptors estimation.

scale-space extrema detection In this step the points of interest, a.k.a salient points, are to be found. The basic idea is to search for extreme values in the second-order derivative of the image (calculable by the Laplace operator), which represents significant changes in the image, e.g. edges and corners. In order to achieve an insensitivity to noise, the image is smoothed by a Gauss operator prior to the Laplace operation.

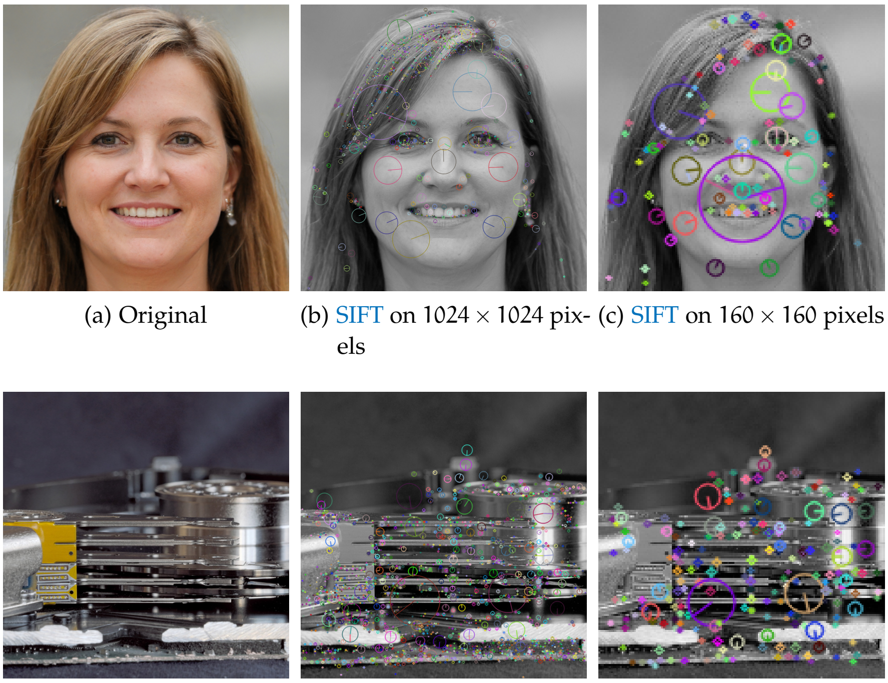
These two operations can be performed by applying the Laplacian of Gaussian (LoG) filter to the image by convolution. By varying the variance (σ) of the Gauss function, the intensity of the smoothing and thus the *size* of the corners can be adapted. To increase the robustness to image scaling, a Gaussian pyramid (the term Gaussian is not directly related to LoG) of the image is constructed. In a Gaussian pyramid, the image is reduced by half in each step (octave). Afterwards the LoG operator is applied to each octave of the pyramid. However, this calculation might be very computationally expensive, thus the LoG can be approximated by Difference of Gaussians (DoG). For this purpose, the image (per octave) is smoothed with Gauss filters with different σ resulting in multiple images per octave (in the following called scales). Subsequently, the differences between neighboring scales are calculated, resulting in an approximation of the LoG. The subtraction preserves the high-frequency information of the less smoothed image, therefore the DoG can be interpreted as a bandpass filter.

On this basis, local extreme values can be determined. To do this, each value is compared with the neighboring pixels (3×3 neighborhood), as well as with the pixels of neighboring scales ($3 \times 3 \times 3$ neighborhood). In this way, the local extreme values of the x and y coordinates of the image as well as the σ coordinate of the scales are found.

keypoint localization In order to obtain a more precise position of the extreme values in space (x and y) and scale (σ), it is approximated by a Taylor series expansion. Subsequently, an extreme value whose intensity is below a defined threshold is discarded, as well as extreme values that are detected as edges by a Harris edge detector. The latter ensures that a continuous edge does not create an unlimited number of keypoints. The remaining, significant extreme values are noted as keypoints.

orientation assignment In order to achieve rotation invariance, an orientation is assigned to each keypoint. Therefore the gradient is calculated for each pixel in the neighbourhood of the keypoint. Subsequently, gradients of a keypoint are transferred into a histogram with 36 bins. The procedure corresponds to the calculation of HOG and is described in Section 6.2.2. The bin with the largest value is defined as the direction of the keypoint. If multiple bins of high values are present in the histogram, additional keypoints with the same position but different direction can be assigned for each value above a defined threshold.

Examples of SIFT keypoints with direction and intensity are given in Figure 6.10. The diameter of the circles shows the magnitude of the extreme value, different keypoints are drawn with different colours. In the original size images (Figure 6.10b and 6.10e), considerably more keypoints are found than in the reduced size images (Figure 6.10c and 6.10f). However, it can be observed that in both images of different size several keypoints with similar direction and intensity occur, for example on the cheek or forehead in Figure 6.10b and 6.10c.



(a) Original (b) SIFT on 1024 × 1024 pixels (c) SIFT on 160 × 160 pixels

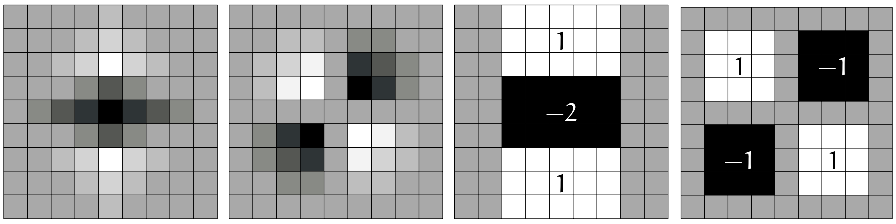
(d) Original (e) SIFT on 626 × 626 pixels (f) SIFT on 160 × 160 pixels

Figure 6.10: Example images of SIFT

keypoint descriptors For each keypoint a keypoint descriptor is calculated. This process creates a characteristic vector for each keypoint, which can be used for the comparison of keypoints. For this purpose a HOG is calculated as described in Section 6.2.2. The length of the feature vector depends on the parameters of the HOG calculation. For example, if the HOG is calculated for a neighborhood of 16 × 16 pixels around the keypoint with 16 blocks of 4 × 4 pixels and a histogram with 8 bins (8 directions), a feature vector with 128 values is obtained.

6.3.2 *Speeded Up Robust Features*

SURF is a faster calculable variant of SIFT, which is proposed by Bay et al. in [10]. For each operation an attempt is made to optimise the calculation operations in order to achieve an acceleration of the feature extraction. The major difference is the use of box filters instead of Gauss Filters. As shown in Figure 6.11, box filters can be interpreted as a rough approximation of the second order Gaussian partial derivative (LoG) used in SIFT. Box filters can be calculated extremely efficiently if the integral image of the analysed image has been calculated beforehand, as the sum of the intensities to be calculated in the box of the box filter can be mapped to three additions in the integral image [10].



(a) Discretised LoG filter in y-direction

(b) Discretised LoG filter in xy-direction

(c) Box filter approximation for LoG filter in y-direction

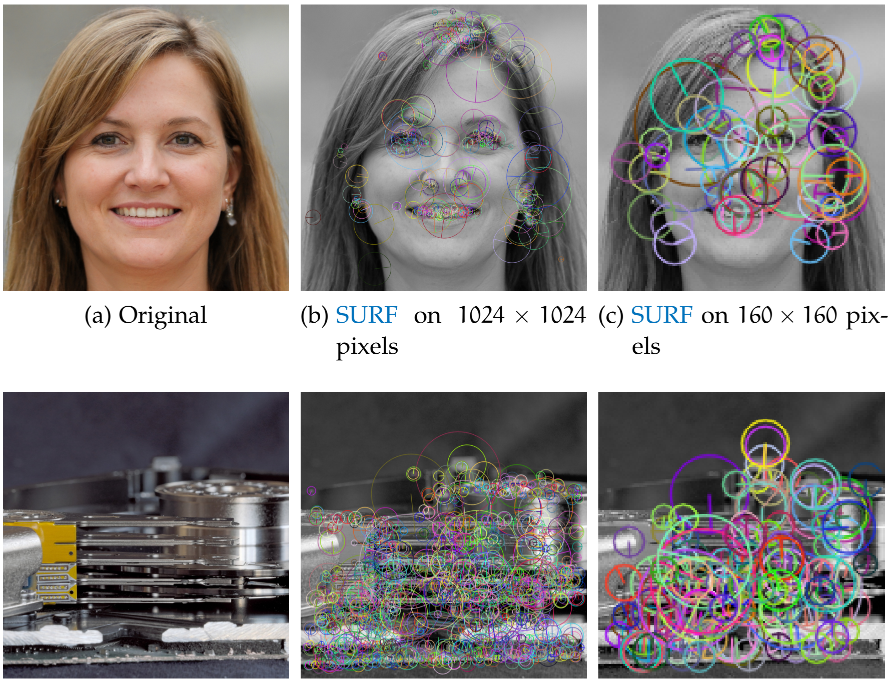
(d) Box filter approximation for LoG filter in xy-direction

Figure 6.11: Example of LoG and box filters, adapted from [10]

For SIFT the DoG are calculated on different sizes of the image (octaves). For this purpose, the image must be scaled several times and then the DoG must be calculated on each octave. For SURF, the box filter is scaled instead of the image. An image half the size corresponds to a box filter double the size. The advantage is, that the integral image has to be calculated only once. It should be noted that the high-frequency information that is lost in the scaling process of SIFT remains present and therefore might influence the result of the box filter.

The detection of keypoints is identical to the method described in Section 6.3.1. The calculation of orientation vectors is also similar to the method used in SIFT. However, instead of calculating gradients, the filter responses of Haar Wavelets are calculated (in both dimensions (d_x, d_y) of the image). The advantage of this approach is that the filter response of the Haar Wavelet can be efficiently estimated based on the already calculated integral image. It should be noted that the size of the wavelet has to be adapted to the size of the box filter responding to the related keypoint. The orientation vector is determined by the sum of the Haar Wavelet's filter responses.

Examples of the detected keypoints and the corresponding orientations and intensities are given in Figure 6.12. In the facial images (Figure 6.12b and 6.12c), less clearly correlating keypoints are found


(a) Original (b) SURF on 1024 × 1024 pixels (c) SURF on 160 × 160 pixels

(d) Original (e) SURF on 626 × 626 pixels (f) SURF on 160 × 160 pixels

Figure 6.12: Example images of SURF

for the two image sizes compared to SIFT (Figure 6.10b and 6.10c). However, keypoints with the same orientation are recognisable as well, especially in the area of the chin. In the sample images of the macro shot (Figures 6.12e and 6.12f), considerably more keypoints are found, due to the higher number of corners in the image.

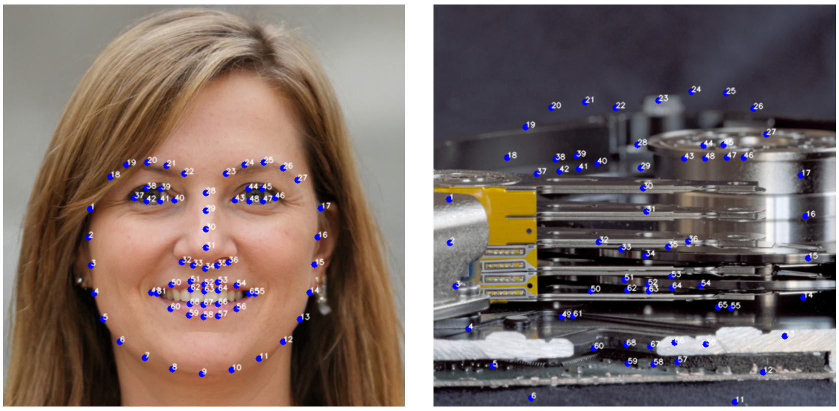
As with the determination of orientation, the descriptors are extracted using the response of Haar Wavelets. A fixed region (e.g. 20 × 20 pixels) around the keypoint is divided into 4 × 4 subregions. For each subregion, the responses to the Haar wavelet are calculated for both dimensions of the image, horizontal (d_x) and vertical (d_y). Subsequently, the feature vector v is calculated for each subregion as follows:

$$v = (\sum d_x, \sum d_y, \sum |d_x|, \sum |d_y|) \tag{6.1}$$

Subdividing the region of the keypoint into 4 × 4 subregions, the resulting feature vector of the keypoint has an overall length of 64 values.

6.4 LANDMARK EXTRACTORS

The keypoints described above are found on the basis of corners and edges in the image and are largely independent of the displayed image content. Another method to identify points of interest in images is to detect so-called landmarks. The idea is to find prominent points of a known object type. An example is given in Figure 6.13. A landmark extractor for facial landmarks[2] has been applied to the facial image (Figure 6.13a) and the macro image (Figure 6.13b). For the facial image, the landmarks were placed on the intended positions. If there is no face in the image, the algorithm attempts to arrange the landmarks in a face-like form. In contrast to the feature extractors presented so far, landmark extractors are thus only suitable for application on a specified object type, for which the landmark extractor has been trained on. Landmarks can be used for detection [20] or alignment [165] of objects, also some approaches for using landmark based systems for face recognition are available.



(a) Example of facial landmarks on a face image

(b) Example of facial landmarks on a non-face image

Figure 6.13: Example images of extracted Landmarks

The detection of landmarks is usually based on the concept of *Active Shape Models* as proposed by Cootes et al. [20]. The concept aims to place a predefined mash of points on an image based on certain criteria, minimizing the error caused by the individual points and the deviation from the original shape. Thus, if a sufficient number of correctly recognized landmarks is available, the other landmarks may be set correctly due to the shape of the object, allowing, for example, partially hidden objects to be correctly identified. A disadvantage of this method is that the number and rough position of the landmarks are determined during training of the landmark detector. If new landmarks are to be added at a later date, the algorithm has to be trained again.

There are many different methods and implementations for landmark detection. In addition, the individual algorithm depends on the training data used, thus even one implementation can produce different results depending on the training data. A detailed description of

[2] The dlib [77] implementation for landmark detection was used.

two implementations of landmark detection is given in Part IV Attack Detection Pipeline of the thesis, in Section 16.4.

6.5 deep features

The feature extractors presented so far aim to transform certain features of the image into a feature vector. The design of the algorithm already includes the consideration of which kind of attributes are considered relevant (e.g. texture in case of LBP and BSIF). Another possibility is to use machine learning to learn features that statistically show the highest discrepancy on the training data between the classes to be distinguished. If a new data point is presented to the trained algorithm, it is transformed into the previously learned discriminative space. In [105] it is shown that NNs are suitable to model feature extractors for images and numeric data.

The advantage of these feature extractors is that they are able to learn very complex correlations and thus (if there is enough variance in the training data) ensure a very robust feature extraction. Unfortunately, the resulting algorithms are difficult to analyse and comprehend, meaning that in case of insufficient training and test data, a possible over-fitting or a focus on insignificant artefacts of the algorithms is not apparent. Due to the abstractness and lack of illustrative capabilities of the extracted features, this section does not provide sample images of features extracted by such an algorithm. It is possible to visualize single layers of NNs or to show the influence of single areas of the image to the feature vector, but this has to be considered during the training of the algorithm and is therefore not possible for the pre-trained NNs used in this thesis. Due to the limited amount of data available, no training of own feature extractors is done for this thesis. Instead, DNN feature extractors pre-trained for other applications are applied. This allows to exclude an over-fitting to the problem and to expect a certain robustness of the feature extraction. A detailed description of the algorithms used in this thesis is given in Part IV Attack Detection Pipeline of the thesis, in Section 16.6.

6.6 image noise pattern

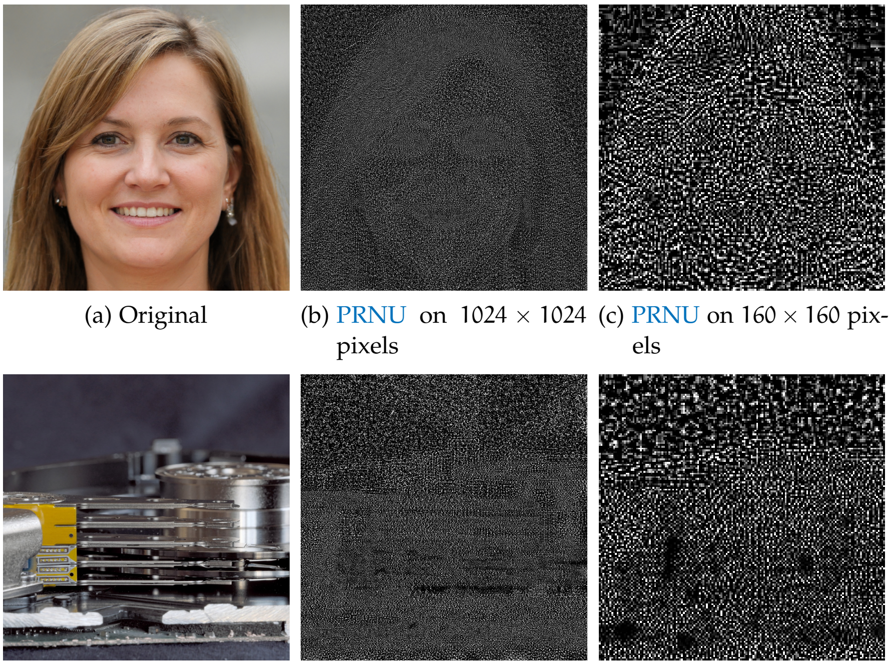
Image Forensics is a subfield of Digital Forensics. The goal of image forensics is to detect the origin [18] or manipulation [152] of images. In this section, sensor noise patterns are introduced which reflect the fingerprint of the camera used to capture the image.

In digital photography, the photons stimulating the camera chip transfer their energy to the electrons contained in the camera chip. During this process, each pixel behaves individually, and the resulting differences are transferred to the digital image as noise patterns. This noise pattern is referred to as Sensor Pattern Noise (SPN) or Photo

Response Non-Uniformity (PRNU). The extraction of these noise patterns is achieved by using high-pass filters. Technically, a smoothed version of the image is subtracted from the original. The resulting image of high-frequency information (noise) depends on the smoothing function applied. As described in [27], [28] and [132], specific features can be extracted from the resulting image. For example, a histogram of the existing frequencies can be created from which certain parameters can be extracted.

An advantage of this feature extraction is the simplicity of the implementation and, when using the frequency information, the rotational invariance. Since these features are based almost exclusively on the high frequency information, they can be changed significantly by a previous smoothing operation applied to the images.

Examples of PRNU noise patterns are given in Figure 6.14. Even though PRNU and SPN are considered to be independent of the image content, the former content is still recognisable. The example of facial images was generated by a GAN [72]. Hence no camera noise is present in this example, resulting in a very uniform noise pattern in Figures 6.14b and 6.14c. The image analysed in Figure 6.14e and 6.14f, on the other hand, originates from a digital photograph. The noise pattern extracted in this case contains much more variance. It can be seen that in lower resolution images (Figure 6.14c and 6.14f) the noise pattern appears much coarser than in the higher resolution images (Figure 6.14b and 6.14e).



(a) Original (b) PRNU on 1024 × 1024 pixels (c) PRNU on 160 × 160 pixels

(d) Original (e) PRNU on 626 × 626 pixels (f) PRNU on 160 × 160 pixels

Figure 6.14: Example images of PRNU

7

BIOMETRIC SYSTEMS

Biometric systems are a variation of identity management systems. In contrast to conventional identity management systems, which are either knowledge-based (with passwords) or token-based (with keys or ID cards), biometric systems observe the subject's biometric characteristics. Features are extracted from the captured Biometric Sample and compared with the features stored in the database. This section describes the principles of biometric systems, in particular the architecture of FRSs.

7.1 TOPOLOGY

Independent of the biometric modality, the basic structure of biometric systems can be described according to the scheme defined in ISO/IEC 19795-1 [62]. As shown in Figure 7.1, a biometric system can be divided into five mandatory components:

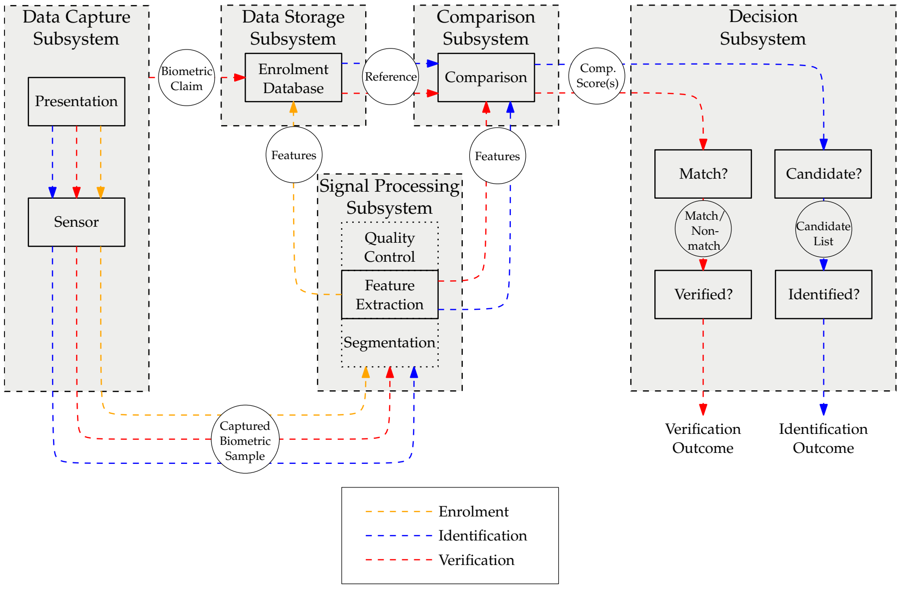


Figure 7.1: Topology of biometric systems, inspired by [62]

DATA CAPTURE SUBSYSTEM At the beginning of each process, a sensor captures the biometric characteristic of a subject and transfers it into a Biometric Sample, e.g. a camera capturing a facial image. In this digital representation, the Biometric Sample is transferred to the Signal Processing Subsystem for further processing.

SIGNAL PROCESSING SUBSYSTEM In the signal processing subsystem, characteristics are extracted from the sample. The appropriate method for feature extraction depends on the modality. Depending on requirements, the samples may need to be segmented first. Optionally, a quality control is performed at sample or feature level. The extracted characteristics are either passed on to the data storage subsystem for enrolment or to the comparison subsystem for identification or verification.

DATA STORAGE SUBSYSTEM The references of the enrolled subjects are stored in the data storage subsystem and forwarded to the comparison subsystem on request. This subsystem might be implemented as a database with several entries, for example an access control system based on fingerprints. However, it can also contain a single entry, as for example in the case of the electronic passport (ePassport), in which only the facial image and fingerprints of the passport owner are stored.

COMPARISON SUBSYSTEM In the data storage subsystem, the reference (verification) or references (identification) transferred from the data storage subsystem and the probe transferred from the signal processing subsystem are compared. The resulting comparison score is available as either a similarity score or a dissimilarity score and is passed on to the decision subsystem.

DECISION SUBSYSTEM In the last step, the comparison score is evaluated using a threshold value or a decision policy. The result is a binary decision whether the subject could be verified or identified.

7.2 OPERATION MODES

Three different operation modes are available for biometric systems. According to ISO/IEC 19795-1 [62], these are defined as follows:

BIOMETRIC ENROLMENT Regardless of the further use of the system, the enrolment is an indispensable first step. This can be interpreted as the registration of a new subject in the system, which is carried out by storing the subject's reference in the data storage subsystem. First, a sample is captured by the sensor in the data capture subsystem. In the signal processing subsystem, features are extracted from the sample, which are subsequently transferred to the data storage subsystem, where they are stored as a template. For one subject more than one template might be stored.

BIOMETRIC VERIFICATION The verification can be interpreted as a confirmation of a biometric claim, for which a one-to-one comparison is performed. First, in the data capture subsystem the sensor captures a sample, which is passed on to the signal processing subsystem. Here the features of the sample are extracted, referred to as probe, and passed on to the comparison subsystem for further processing. Simultaneously, the reference from the data storage subsystem corresponding to the biometric claim is passed on to the comparison subsystem. Subsequently, the reference is compared against the probe and the resulting comparison score is transferred to the decision subsystem, where the binary decision whether the biometric claim is true (Accept) or not (Reject) is taken based on a threshold. The operation mode is mainly used in the area of access control, e.g. for border control systems or for authorized use of protected systems.

biometric identification The aim of the identification is to find the corresponding biometric reference identifier of an individual. The biometric reference identifier is a pointer to a specific template stored in the data storage subsystem. As with verification, the process commences with the creation of the probe sample by the data capture subsystem and signal processing subsystem. However, during the identification process, not only one reference is passed from the data storage subsystem to the comparison subsystem, but all of them. The comparison subsystem compares the probe with each individual reference (a one-to-many comparison), and the resulting vector of comparison scores is passed to the decision subsystem. The decision subsystem uses this vector to decide whether a biometric reference identifier could be returned or not. This operation mode is mainly used in forensics (e.g. to identify flood victims), or for blacklists (e.g. in casinos as blacklist for gambling addicts).

7.3 performance estimation

In conventional identity management systems there are only two options: correct or incorrect (the password or key matches or not). In biometric systems, however, there is a natural variance in the capture process at the sensor, which is why probe and reference samples are never identical. In order to be able to consistently measure and thus compare the errors that arise in biometric systems, the following error types are defined in [62] and [66].

Failure-To-Capture (FTC) The proportion of failures of the biometric capture process to produce a captured Biometric Sample [66]. This describes failures that occur in the data capture subsystem. The FTC can be calculated as:

$$\mathrm{FTC} = \frac{N_{tca} + N_{nsq}}{N_{tot}}, \tag{7.1}$$

where N_{tca} is the number of terminated capture attempts, N_{nsq} the number of images with insufficient sample quality and N_{tot} the total number of capture attempts.

Failure-To-eXtract (FTX) The proportion of failures of the feature extraction process to generate a template from the captured Biometric Sample, N_{ngt}, to the number of successful captured samples, N_{sub}. This describes failures that occur in the signal processing subsystem. The FTX can be calculated as:

$$\mathrm{FTX} = \frac{N_{ngt}}{N_{sub}}. \tag{7.2}$$

Failure-To-Enrol (FTE) The proportion of a specified set of biometric enrolment transactions that resulted in a failure to create and store a biometric enrolment data record, N_{nec}, to the total number of subjects, intended to be enrolled in the biometric application, N [66]. This describes failures that occur in the data storage subsystem. The FTE can be calculated as:

$$\mathrm{FTE} = \frac{N_{nec}}{N}. \tag{7.3}$$

Failure-To-Acquire (FTA) The proportion of a specified set of biometric acquisition processes that were failure to accept for subsequent comparison of the output of a data capture process [66]. This metric summarizes the failures of the data capture subsystem and the signal processing subsystem. The FTA can be calculated as:

$$\mathrm{FTA} = \mathrm{FTC} + \mathrm{FTX} \cdot (1 - \mathrm{FTC}). \tag{7.4}$$

False Non-Match Rate (FNMR) Proportion of genuine attempt samples falsely declared not to match the template of the same biometric instance from the same subject supplying the sample [62]. This failure occurs in the algorithm of the comparison subsystem. The FNMR for a specific threshold τ can be calculated as:

$$\mathrm{FNMR}(\tau) = \int_{0}^{\tau} \Phi_g(s) ds, \tag{7.5}$$

where $\Phi_g(s)$ represents the Probability Density Function (PDF) of the genuine comparisons with s as similarity score. An example of a FNMR is given in Figure 7.2.

False Match Rate (FMR) Proportion of zero-effort impostor attempt samples falsely declared to match the compared non-self template [62]. This failures occur in the algorithm of the comparison subsystem. The FMR for a specific threshold τ can be calculated as:

$$\mathrm{FMR}(t) = \int_{\tau}^{1} \Phi_i(s)ds, \tag{7.6}$$

where $\Phi_i(s)$ represents the PDF for the imposter comparisons, with s as similarity score. An example of a FMR is given in Figure 7.2.

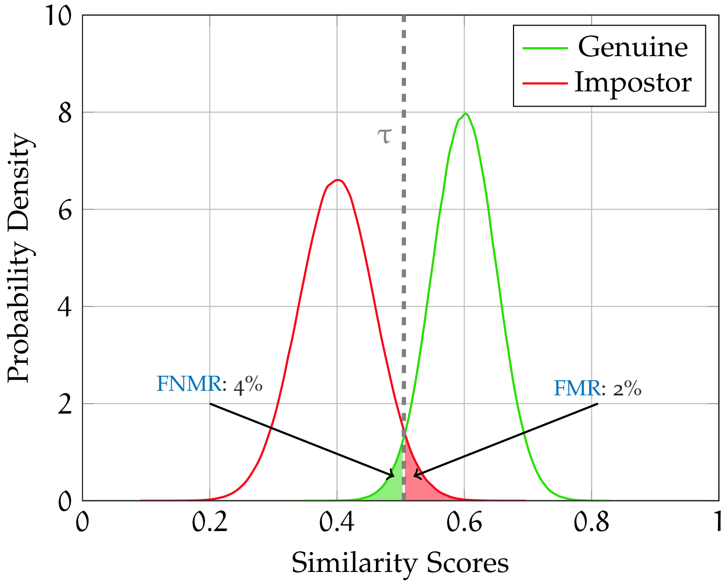


Figure 7.2: Visualization of FMR and FNMR

The two metrics, FNMR and FMR, describe comparison algorithm errors. In order to determine the overall performance of a biometric system further metrics are needed:

False Reject Rate (FRR) The proportion of verification transactions with truthful claims of identity that are incorrectly denied [62]. The FRR can be calculated as [66]:

$$\mathrm{FRR} = \mathrm{FTA} + \mathrm{FNMR} \cdot (1 - \mathrm{FTA}). \tag{7.7}$$

False Accept Rate (FAR) The proportion of verification transactions with wrongful claims of identity that are incorrectly confirmed [62]. The FAR can be calculated as [66]:

$$\mathrm{FAR} = \mathrm{FMR} \cdot (1 - \mathrm{FTA}). \tag{7.8}$$

Equal Error Rate (EER) Generally, the term EER denotes a point at which two arbitrary errors, which have to be balanced against each other, are of equal extent. However, it has become common practice in biometrics to refer the EER to the point of equal error between FMR and FNMR. In order to avoid misunderstandings, the EER in this thesis is therefore exclusively referred to FMR and FNMR. If the operating point of equal error is calculated for other error types, it will be defined separately.

7.4 FACE RECOGNITION SYSTEMS

Most FRSs can be divided into fife steps: (1) Face recognition, (2) pre-processing, (3) feature extraction, (4) comparison and (5) decision. The enumerated steps are described in more detail in this section.

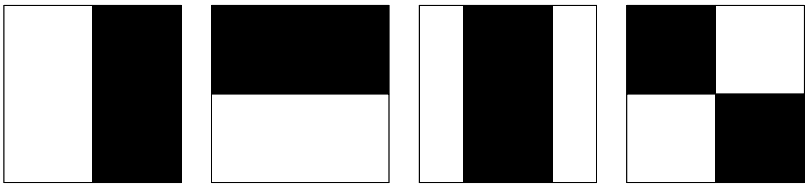
7.4.1 *Face Detection*

The first step is to determine the position of the face in the captured sample. The task of face detection is complicated by many factors. The most common difficulties encountered with biometric systems are [59]:

- The person may have different poses and facial expressions.
- The face may be covered, for example by hair or glasses.
- Various other features on the face, for example tattoos or piercings.
- Different illumination of the face.

Due to the high variance, robust face detection in real time was considered a potentially unsolvable task. Only with the Viola-Jones algrithm [162], introduced in 2001, real-time face detection became feasible. Even though many new methods for face detection have been proposed since then (an overview can be found in [54]), the Viola-Jones algorithm, which is described below, is still state-of-the-art, especially in terms of speed and accuracy.

The basic concept of the Viola-Jones algorithm is based on the Ensemble Classifiers (Section 5.3), where a strong algorithm is combined from many weak classifiers. It is based on haar-like features, which attempt to map basic features of a face, examples of those filters are given in Figure 7.3. For example, the area around the eyes is usually



(a) Example 1 (b) Example 2 (c) Example 3 (d) Example 4

Figure 7.3: Example of Haar-like filters

darker than the area of the cheeks, due to shadows. This property can be modelled using the haar-like feature shown in Figure 7.3b, which is dark in the upper area (eyes) and light in the lower area (cheeks). The haar-like features are a form of box filter and can therefore be efficiently calculated on the integral image as described in Section 6.3.2.

A filter cascade is trained from these haar-like features applying AdaBoost (see Section 5.3.2). Cascade means that not all weak learners are tested simultaneously, as in the case of the ensemble classifier, but that it depends on the result of the previous classifier whether the next one is tested or not, which can further increase efficiency. For a more detailed description of the training process the reader is referred to [162].

7.4.2 *Pre-Processing*

Once the face has been recognized, the next step is to prepare it in a way that the subsequent feature extractors can reliably extract features. For this purpose, the image is aligned, cropped and eventually enhanced. For the alignment process, landmarks are detected first, afterwards the face is aligned, e.g. by means of the eyes, in order to compensate possible pose variations. Further details on the functionality of landmark extractors are given in Chapter 8. After alignment, the face image is cropped, ensuring that only regions relevant for the feature extractor are included. Most feature extractors rely on the fact that the same facial sections are always presented to them. Finally, if necessary, the image can be improved (or unified), for example by performing a histogram normalization.

7.4.3 *Feature Extraction*

Various methods can be used to extract the facial features. Common methods for feature extraction are LBP [2] (as described in 6.1.1), in former times Eigenfaces [159] or shape based features [49]. However, the current trend is towards Deep Features. Here DNNs are trained to transform the facial image into a feature in a discriminatory space. An overview of the current deep FRSs can be found in [164].

7.4.4 *Comparison*

A comparison score is calculated by comparing two feature vectors, extracted in the previous step. The easiest way to do this is to determine the difference between the feature vectors using a distance metric, for example the euclidean or cosine distance. However, depending on the complexity of the feature output, these metrics may not be sufficient.

In this case, machine learning algorithms, see Section 5, can be used to estimate a decision score.

Regardless of the method used, the result is always a comparison score, which can either be the distance scores between the two feature vectors or the similarity score.

7.4.5 *Decision*

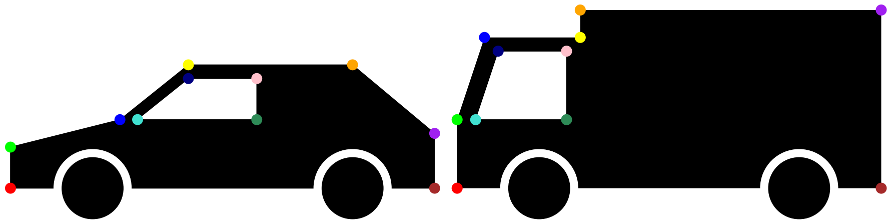
As shown in Figure 7.2, the comparison scores calculated from genuine and impostor comparisons each form a distribution curve. These curves should be clearly separable, assuming the system is working properly. In a real world situation, the system must provide a binary decision, for example, whether two samples stem from the same source or not. For this purpose, a threshold value is set in the system, which separates the genuine and the importor distribution as optimally as possible and thus minimizes the resulting FNMR and FMR (or FAR and FRR when considering the full system). The choice of the optimal threshold value depends on the situation. If the system is used in a safety-critical environment, the threshold value can be set in such a way that the FMR is decreased, if more convenience is required, the FNMR should be decreased instead.

8 IMAGE MORPHING

In image processing, morphing refers to the process of changing one image or shape into another. The research field of morphing is already decades old, the main motivation for this research is the film industry [166]. In films, morphing is regularly used as a special effect to optically transfer one image or object into the target image or object. In this section the morphing process of two dimensional images is considered. There is a wide range of different morphing algorithms, but most of them can be divided into the following three steps: correspondence determination, warping and blending. These steps are described in the following section.

8.1 CORRESPONDENCES

In order to transfer one image to another, corresponding points in both images must be determined first. A very simplified example is given in Figure 8.1. In this example a car (Figure 8.1a) is to be morphed into a truck (Figure 8.1b). The corresponding points are marked with different coloured circles. Since the two forms are very similar in type (for example, the window in both examples consists of four corners), finding the correspondence is done easily by hand. In a real application, this might be significantly more difficult. Finding the correspondence is done either manually or automatically by assigning certain points in both images to each other. The manual determination of correspondences can be arbitrarily precise, but is time-consuming. The manual process might be simplified by using line segments or curves and lead to better results. The automatic detection of correspondences is mostly done with landmark extractors (see Section 6.4). Besides the use of landmarks, there are other methods to determine correspondences, e.g. via line segments [11] or curves [86]. However, these older methods mainly aim at simplifying the manual definition



(a) Car

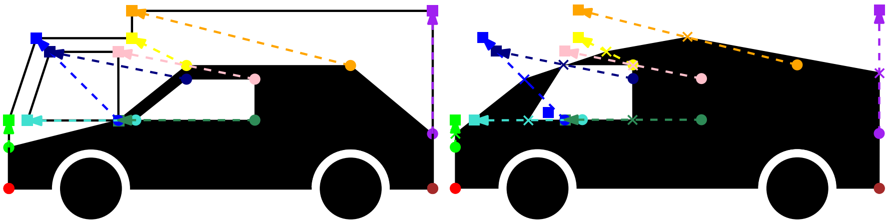
(b) Truck

Figure 8.1: Example of correspondences for image morphing

of correspondences and are not used for the automated recognition of correspondences.

8.2 WARPING

Based on the specific points of correspondence, one image can be transformed to the other. The process corresponding to the example presented is shown in Figure 8.2a. Each point is moved linearly from the position of the car to the position of the truck (visualized by the dashed lines). The translation between two corresponding pixels P_0 and P_1 can thus be expressed by the warping function $w_{P_0 \to P_1}$:



(a) Transformation lines to transform the car to a truck (b) Intermediate image ($\alpha_w = 0.5$) of transformation from car to truck

Figure 8.2: Example of the transformation of car to a truck

$$P_1 = w_{P_0 \to P_1}(P_0) \tag{8.1}$$

The goal of morphing is to calculate intermediate images of this transformation. This can be achieved by weighting the warping function with the factor α_w:

$$P_{\alpha_w} = \alpha_w \cdot w_{P_0 \to P_1}(P_0) \tag{8.2}$$

An α_w of 0 would thus correspond to the pixel positions of P_0, an $alpha_w$ of 1 to those of P_1. Figure 8.2b shows an intermediate image of the example. α_w was set to 0.5, thus the shifting of the points is only carried out to half of the distance.

The shown example has been simplified for a better explanation. Thus both objects consist of the same number of corners, which are connected by straight edges. The round shapes of the wheel cases and tires were deliberately positioned identically and not marked with landmarks. If more complex or different shapes are transformed, the pixels located between the points need to be moved individually. For the simplified example, linear interpolation can be performed between the individual points. If, however, a circle is to be warped into a square, for example, linear interpolation would only lead to the desired result with an infinite number of corresponding points. For interpolation in real applications, three main methods can be found in the literature: grid warping, mesh warping and non-local warping methods.

grid warping When using grid warping, the points marking the relevant areas (i.e. landmarks) in both source images are mapped to a virtual grid [166]. The coordinates of the landmarks within this grid can now be used to easily determine how they need to change in the transition. If a point is at $(0|0)$ in the first image and at $(100|100)$ in the second image, the point is positioned at $(25|25)$ for $\alpha_w = 0.25$. In this approach, all points are uniformly adjusted by geometrical operations.

mesh warping A further possibility is the calculation of a mesh adapted to the landmarks. The most frequently used function for determining such a mesh is Delaunay triangulation [76], whereby all triangles of the triangle mesh fulfil the so-called circumcircle condition: The circumference of a triangle of the mesh must not contain any further landmarks. Subsequently, the optimal path of the transition is determined, by which the triangles of the first output image are shifted and deformed towards the triangles of the second output image, depending on the α_w value. The warping is calculated locally for each triangle separately by transforming it into the target triangle. For this purpose an affine transformation may be applied. When calculating the transformed pixels, it may be necessary to interpolate missing pixels or pixels from non-integer pixel positions. For this interpolation the methods established for image scaling can be used, e.g. bilinear or bicubic interpolation, Sinc or Lanczos filters. The choice of the interpolation method can have a significant effect on the result. Common methods and their advantages and disadvantages are explained in [35].

non-local warping For the algorithms described so far, the images to be warped are divided into geometric shapes (a fixed grid, or triangles), which are subsequently distorted. This may result in artefacts, for example at the transitions of two triangles, as the two adjacent triangles may be distorted differently. To avoid these issues, the warping can be calculated non-locally by interpolation between the landmarks; possible methods are described in [129]. A method for global transformation, which can also be well adapted to local differences, is the use of radial basis functions [4]. A frequently used radial basis function is Thin-Plate-Spline, which aims to model the deformation of a thin metal sheet [86].

8.3 blending

So far, the warping only changed the position of the pixels. However, to create the intermediate images it is required to change the textures as well. This is done by blending. The simplest way of blending is linear blending, where the colour values of the pixels of the two original images are weighted and added. The parameter for the weighting is called α_b. For a colour image, the value I_{α_b} of the pixel at the position (x, y) is calculated for each colour channel as the weighted sum of the two source images:

$$I_{\alpha_b}(x, y) = (1 - \alpha_b) \cdot I_0(x, y) + \alpha_b \cdot I_1(x, y) \tag{8.3}$$

Thus, with $\alpha_b = 0.5$, the arithmetic mean of the values from the two output images is obtained.

According to [154], linear blending does not always yield optimal results, since it can lead to an unnatural appearance in the target images. This may result in artificially hard edges, blurred intermediate areas or similar artefacts. Alternative methods of blending have therefore been proposed, for example by minimizing a cost function (also called energy function) that captures the local variance of brightness values, resulting in a more natural appearance of colour gradients and less blur [167].

9 SUMMARY

In this part the technical basics for a deeper understanding of the following chapters are described. The basics are divided into chapters on Machine Learning, Image Descriptors, Biometric Systems and Image Morphing.

Machine learning algorithms, as a sub-area of computational intelligence, are algorithms that make decisions based on previously observed data. Machine learning algorithms can be divided into two classes, predictive and descriptive algorithms. Descriptive algorithms try to reproduce the population described by the data (for example by clustering), predictive algorithms recognise the differences between populations and aim to assign new data points to one of them. In this thesis the principles of different predictive algorithms are described. Firstly, it covers the SVM with different kernels, which, in particular in biometrics, is a frequently applied classifier. Furthermore, decision trees are described and different ensemble classifiers which can be built based on decision trees, for example Random Forest, AdaBoost or Gradient Boosting. Furthermore the basics of neural networks, which are used in the currently popular deep learning methods, are explained.

To be able to classify images with the described machine learning algorithms, it is necessary to extract features from the images, which can be used to train the classifiers and for decision making. The feature extraction methods used in this thesis, so called image descriptors, are described in chapter 6. Each class of descriptors can be used to extract a certain property of an image and describe it in a feature vector.

Chapter 7 describes the basic structure and functionality of biometric systems. According to ISO/IEC 19795-1 [62] biometric systems can be partitioned into five components: Data capture subsystem, signal processing subsystem, data storage subsystem, comparison subsystem and decision subsystem. The functionality of each subsystem is described in Section 7.1. Furthermore, the standard describes the different operation modes of biometric systems, as well as the metrics standardised in [62] and [66] which are suitable for the evaluation of biometric systems. Since FRSs play a superordinate role in this work, their function and structure are discussed in more detail.

In Chapter 8 the technical basics of the algorithms used to morph images are described. These can be divided into three components: Correspondence recognition, warping and blending. In the first step, corresponding points in both images are determined, which should be superimposed in the resulting morph. Warping distorts the images in such a way, that the corresponding pixels are positioned in the same location. The final blending combines the distorted images.

Part III

CONCEPTS AND RELATED WORK

10 MORPHING OF FACIAL IMAGES

Chapter 2 describes the problem of morphing attacks on FRSs. Chapter 8 explains the basic theory for morphing images. This chapter will go into more details about the creation of morphed facial images and the state-of-the-art. The creation of morphed facial images is done according to the concept described in Chapter 8: Detection of correspondences, warping and blending. Afterwards, optimizations tailored to morphed facial images can be applied, for example to avoid typical artefacts that might occur during the morphing process.

10.1 CORRESPONDENCES

In publications on morphing and morphing attack detection, the morphing algorithms are usually not described in detail. Especially in the beginning, the correspondences of the images to be morphed were often determined manually, e.g. in [39] and [120]. In the meantime, several algorithms with automatic landmark detection are circulating in the scientific community. These are mainly based on the open source, pre-trained landmark extractors Dlib [77] and Stasm[1], which are based on the concept of the Active Shape Model, as described in Section 6.4. In addition, there are approaches to detect landmarks by applying an DNNs [36], which, however, can only determine a limited number of landmarks so far, resulting in a significantly lower quality of the generated morphs. Despite the wide range of landmark detection algorithms available, there are no algorithms yet reliably modelling the contour of the hair. Also the detection of the iris is not provided by any algorithm so far. Although some algorithms position a landmark in the centre of the eye (e.g. those of the Apple Vision Framework[2]), they are always positioned centrally in the eye and are not adapted to the actual position of the iris.

10.2 WARPING

Most algorithms use the Delaunay triangulation as described in Chapter 8, for example the morphing method used in [39] using manual determination of correspondences or the fully automatic morphing algorithms used in [40] and [135]. An example of landmarks detected by Dlib and the resulting Delaunay triangles are depicted in Figure 10.1. If a sufficient number of correctly placed landmarks is available, high

[1] http://www.milbo.org.stasmfiles/stasm4.pdf
[2] https://developer.apple.com/documentation/vision

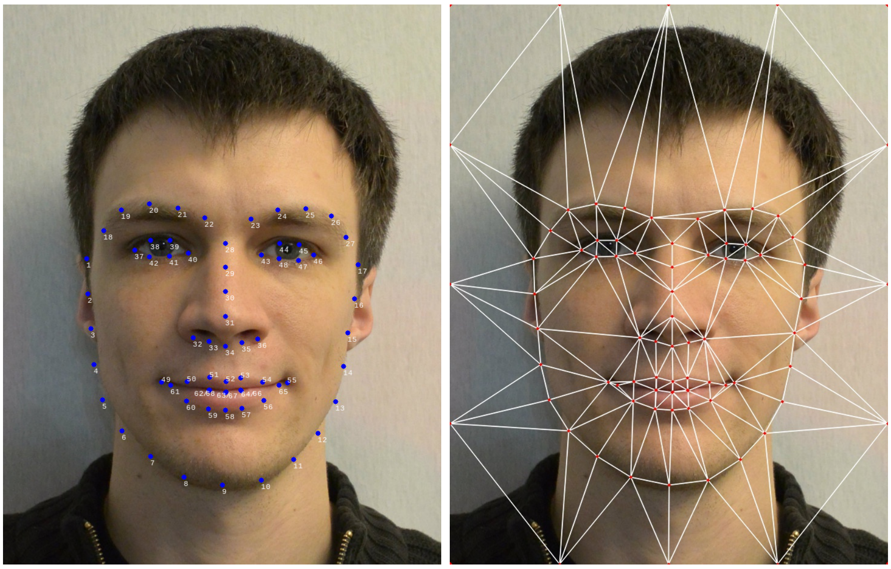

(a) Example of Dlib landmarks (b) Resulting Delaunay triangles

Figure 10.1: Example of Delaunay triangulation

quality morphs can be created in an automated manner. However, since the well-known landmark detection algorithms do not place landmarks at the hairline and outer areas of the face, these areas are prone to particularly severe artefacts. In these areas, additional landmarks would allow the creation of more precise triangles and thus more accurate morphs. Yet, using too many landmarks might also lead to negative effects on the quality of the generated morphs. For example, Dlib [77] detects separate landmarks for the upper lower lip and the lower upper lip for more general results. Thus, if images with closed mouths are morphed, these landmarks overlap, which may result in artefacts. An example of such an artefact is given in Figure 10.2.

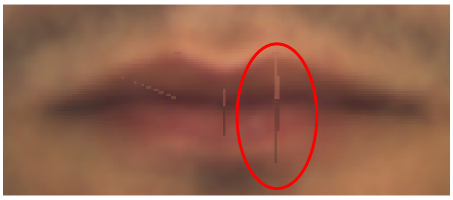


Figure 10.2: Example of morphing caused by overlapping landmarks.

10.3 BLENDING

Due to the preceding triangulation, the determination of the corresponding pixels is straightforward. The respective colour values are calculated according to the selected α_b using equation 8.3. Even though there are possibilities to refine the blending process as described in Section 8.3, there is currently no implementation known that does not calculate the blending using a weighted sum of the pixels of both source images.

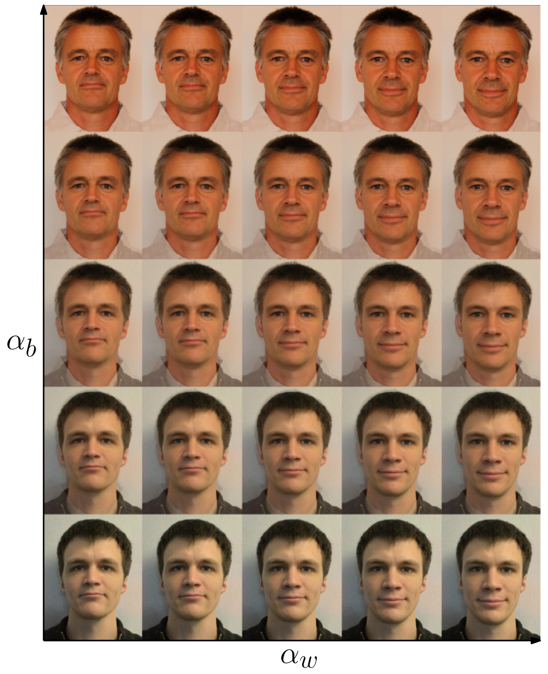


Figure 10.3: Morphed Face image with changing α_w and α_b-values

It should be noted, that the α values, for warping (α_w) and blending (α_b), can be handled independently [38]. The changes in the appearance of the morphed facial image depending on the two α values are shown in Figure 10.3. In the common case both α values are set to 0.5, which corresponds to the central frame in the transition from the start image to the target image. In this case both contributing subjects are equally represented.

Experience has shown that α values around 0.3 offer an increased chance of success in deceiving human observers [40]. However, it should be noted that the chance of success for attacks on FRS with α values unequal to 0.5 decreases, partly considerably [138]. For this reason, in this paper morphs are created with an α value of 0.5.

10.4 IMPROVEMENTS

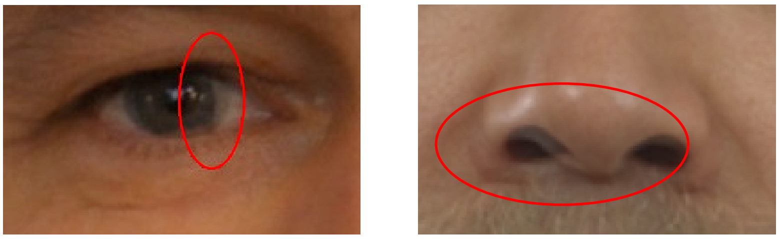
During the generation of morphed facial images by the methods described above, unavoidable artefacts might be introduced. Most of the errors are caused by too few or incorrectly positioned landmarks, which is difficult to avoid when using automated morphing algorithms [139]. In this section two methods are presented to automatically correct the most common artefact types, namely using swapping and artefact replacement.

10.4.1 *Swapping*

A significant number of artefacts are created in areas outside of the landmarks, for example on the collar of clothing or in the region of the hair. The automatic landmark detectors do not place landmarks in these areas, thus no reasonable Delaunay triangles will be formed. Since the problematic areas are of no great relevance for FRSs, a simple solution is to replace the outer areas of the morph with the artefact-free outer areas of one of the original images [97], [40]. To do this, the original image is first adjusted to the morph by distortion and eventual colour adjustments, afterwards the inner area of the morph is swapped into the original image. Optionally a smoothing of the transition line can be applied.

10.4.2 *Artefact Replacement*

There are also areas within the landmark area that are particularly susceptible to artefacts. These include particularly contrast rich areas such as the nostrils or the eyes. Examples of such artefacts are given in Figure 10.4. These areas can be replaced using a similar method as described in Section 10.4.1. For this purpose, a mask is defined according to which certain areas of the original image are blended over to the morph. An example of such a mask is given in Figure 10.5, the white areas, in the given example eyes and nostrils, are blended from the distorted original image over the morph. The fade is smooth in order to avoid hard edges. This method is used, for example, in a version of FaceFusion[3] adapted for the FACETRUST project.



(a) Example of morphing artefacts at the iris.

(b) Example of morphing artefacts at the nostrils.

Figure 10.4: Example of morphing artefacts.

10.4.3 *Manual Post-Processing*

In addition to automatic post-processing, it is possible to manually correct artefacts. However, this is very time-consuming, especially if high-quality results are desired. Basically the same concept as in

[3] http://www.wearemoment.com/FaceFusion/

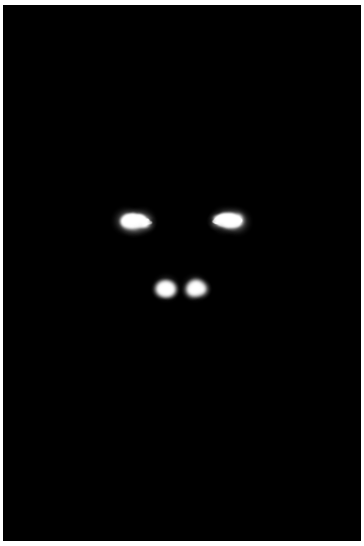


Figure 10.5: Example of a predefined mask for the replacement of critical areas

Section 10.4.1 and 10.4.2 can be applied here, but without using predefined masks. Instead, regions affected by artefacts are blended as appropriate.

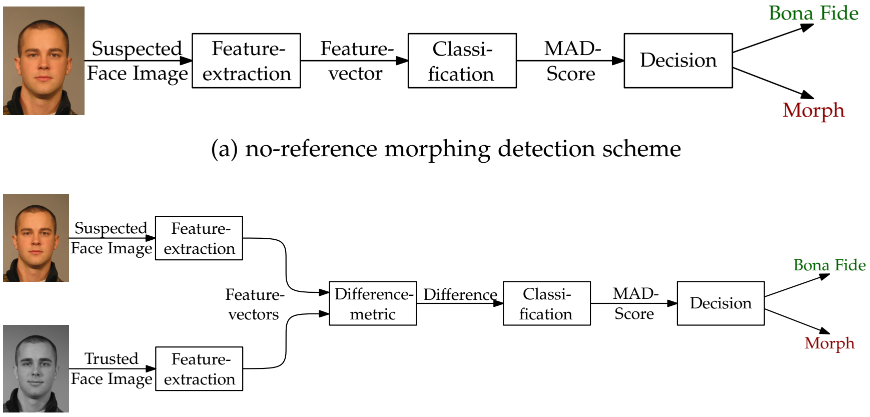
For the described improvements of the morphed face images it has to be taken into account that with each replaced region the similarity to one subject increases and the similarity to the other subject decreases. Therefore, a balanced morph ($\alpha_w = 0.5$, $\alpha_b = 0.5$) can no longer be considered balanced after such post-processing.

11 DETECTION OF MORPHED FACIAL IMAGES

As described in Chapter 2, MAs pose a serious threat to FRSs, especially in the border control scenario. In order to guarantee a secure operation of face recognition algorithms in the future, it is necessary to be able to reliably detect morphed facial images and thus be able to reject them during enrolment or verification. This chapter provides an overview of the schematic structure of MAD algorithms and metrics to measure and compare the MAD performance.

11.1 DETECTION SCHEMES

According to [139], MAD systems can be divided into two categories: no-reference or single image MAD (S-MAD) and reference based or differential MAD. The corresponding scheme for S-MAD is shown in Figure 11.1a. The image to be analysed is passed to the MAD system.



(a) no-reference morphing detection scheme

(b) differential morphing detection scheme

Figure 11.1: Categorisation to no-reference and differential morphing detection scheme

First, features are extracted, based on which the classifier decides whether the presented image is a morph or bona fide. The S-MAD scheme can be used during enrolment as well as during verification.

Differential MAD can be used in scenarios where another image, a Trusted Live Capture (TLC), is available in addition to the suspected morph. For example, during verification, when the probe image is acquired in addition to the stored reference (suspected morph). The schematic process of differential MAD is depicted in Figure 11.1b. The same features are extracted from both provided images. These are compared according to a fixed metric and the classifier uses this difference to decide if the suspected morph is a morph or bona fide.

This method has the advantage that the additional information of the TLC is used for the decision. However, it should be noted that in real scenarios TLCs are usually acquired in semi-supervised environments, e.g. border gate, and therefore show a lower quality and higher variance than the suspected images extracted from the travel document.

11.2 EVALUATION METHODOLOGY AND METRICS

To compare different algorithms with each other, uniform evaluation methods and metrics are essential. For the evaluation of the vulnerability of FRSs against morphing attacks, different metrics have been introduced in previous publications, which will not be described further in order to avoid confusion. In this thesis only the metrics proposed in [139] are described and applied. For the evaluation of the MAD algorithms the metrics defined in ISO/IEC 30107-3 [65] are applied.

11.2.1 *Face Recognition System Vulnerability*

In ISO/IEC 30107-3 [65], Impostor Attack Presentation Match Rate (IAPMR) provides a metric to evaluate the success of attacks on a biometric system:

IAPMR: in a full-system evaluation of a verification system, the proportion of impostor attack presentations using the same PAI species in which the target reference is matched [65].

Considering the PDFs depicted in Figure 11.2, the red and green curves correspond to a biometric system, as described in Section 7.3. The attacks should be separately identifiable and are represented in a further, here yellow, curve. In a scenario without attacks, the threshold τ would be set to minimize FMR and FNMR, in the given example 0.5. The exemplary attack would thus be 94% above the selected threshold, i.e. for 94% of the attacks the comparison with the subject would be successful.

For presentation attacks this metric is applicable, since one artefact is presented at a time, which is intended to circumvent the system. For morphing attacks, however, it is necessary that at least two subjects are successfully compared with the morph. In order to reflect this, the IAPMR was extended to the Mated Morph Presentation Match Rate (MMPMR) in [139]. According to [98], only uncorrelated comparisons should be carried out in evaluations. Thus, each subject is compared only once per morph. A morphing attack success is given if all subjects contained in the morph have been successfully verified against it. Therefore, only the minimum for similarity scores or the maximum

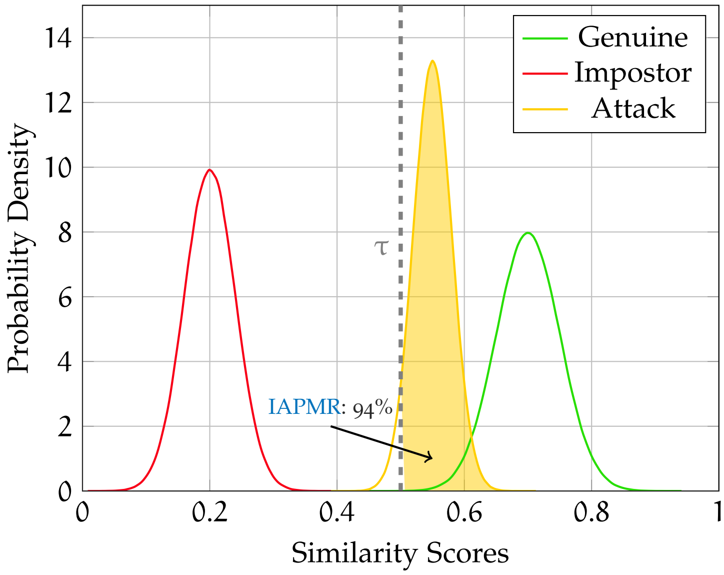


Figure 11.2: Example of IAPMR

for dissimilarity scores has to be calculated over all comparisons of a morph. For similarity scores the MMPMR is calculated as follows:

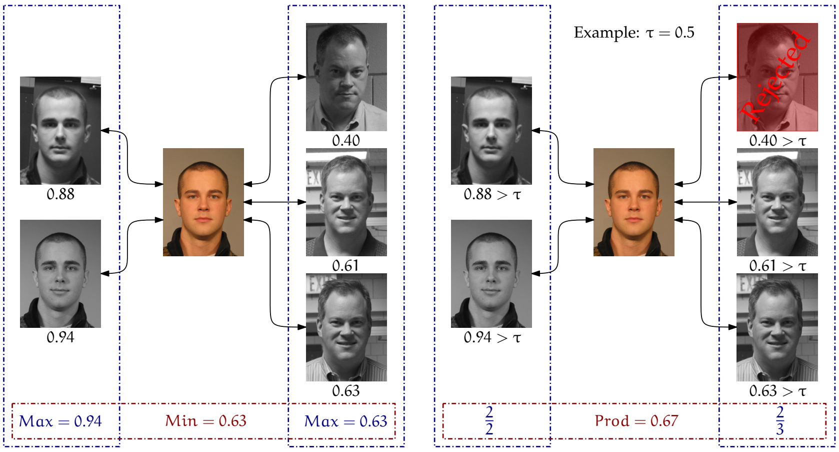
$$\text{MMPMR}(\tau) = \frac{1}{M} \cdot \sum_{m=1}^{M} \left\{ \left[\min_{n=1,\ldots,N_m} S_m^n \right] > \tau \right\}, \tag{11.1}$$

whereas τ is the verification threshold, S_m^n the mated morph comparison score of the n-th subject to morph m, M is the total number of morphed images, and N_m the total number of subjects contributing to morph m.

However, in most circumstances the requirement set out in [98] cannot be met. On the one hand, the number of possible comparisons is considerably reduced, so that, especially with a limited number of subjects, no reliable evaluation can be carried out. On the other hand, this does not necessarily reflect a realistic scenario. For example, automatic border controls compare several live images of the same subject with the passport photograph. To reflect this behaviour, the MinMax-MMPMR was defined in [139] as visualised in Figure 11.3a. If there are several images of the same subject, only the one with the highest similarity score is considered (maximum). Across the subjects, the minimum is calculated, as all subjects must be verified successfully. Equation 11.1 is thus extended as follows:

$$\textit{MinMax-MMPMR}(\tau) = \frac{1}{M} \cdot \sum_{m=1}^{M} \left\{ \left(\min_{n=1,\ldots,N_m} \left[\max_{i=1,\ldots,I_m^n} S_m^{n,i} \right] \right) > \tau \right\}, \tag{11.2}$$

whereas I_m^n is the number of samples per subject n within morph m.



(a) MinMax-MMPMR

(b) ProdAvg-MMPMR

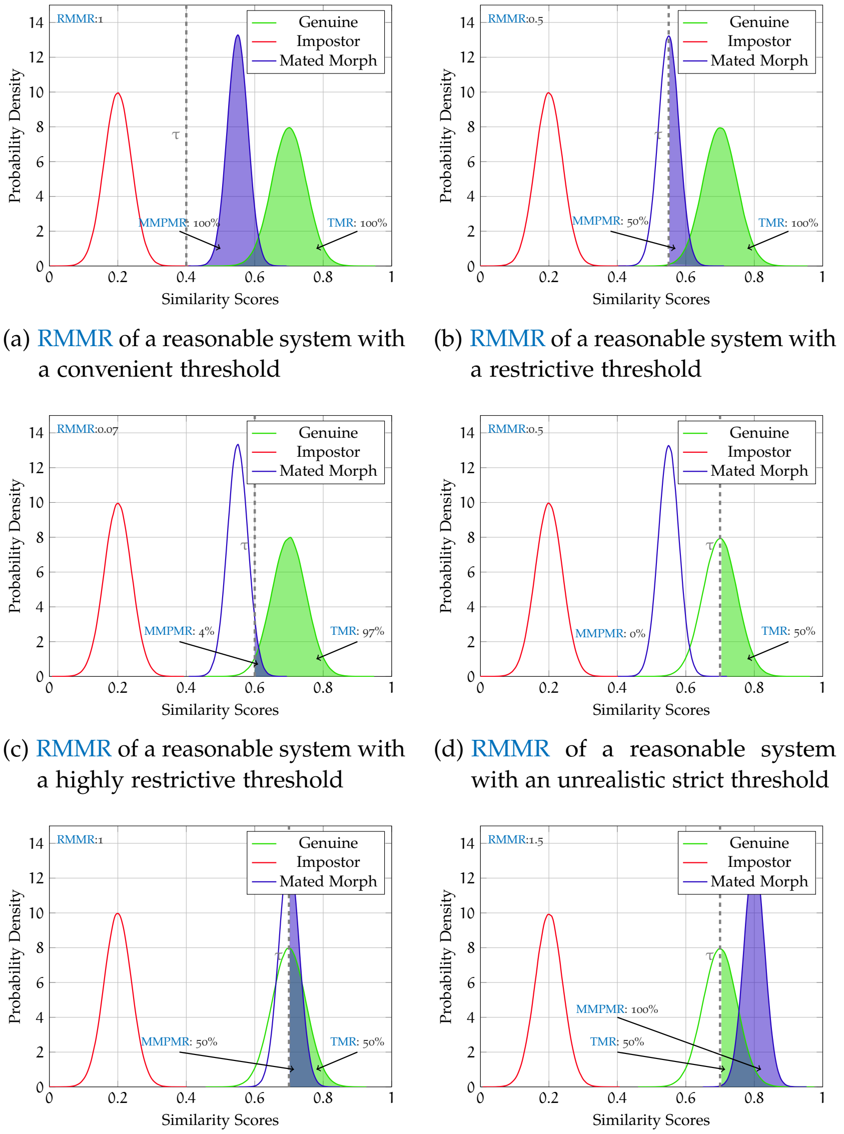
Figure 11.3: Examples of the scheme of the different MMPMR definitions

However, if an unlimited number of comparisons per subject are allowed, the MinMax-MMPMR reveals a shortcoming. The more comparisons per subject, the higher the probability that one comparison is above the threshold of the system. Since only the closest comparison is evaluated (maximum), this subject would be considered accepted. In this way, many comparisons can artificially increase the success chance of the morphing attack. To compensate for this effect, the ProdAvg-MMPMR can be used as a probabilistic interpretation for a large number of comparisons, the scheme is illustrated in Figure 11.3b. In this case, each individual comparison is checked against the threshold and the average of the successful verification of the individual comparisons is estimated for each subject. The probabilities of all subjects are then multiplied by each other as joint probabilities, resulting in the following equation for the ProdAvg-MMPMR:

$$\mathit{ProdAvg\text{-}MMPMR}(\tau) = \frac{1}{M} \cdot \sum_{m=1}^{M} \left[\prod_{n=1}^{N_m} \left(\frac{1}{I_m^n} \cdot \sum_{i=1}^{I_m^n} \{ S_m^{n,i} > \tau \} \right) \right] \tag{11.3}$$

Both IAPMR and the variations of MMPMR derived from it are directly dependent on threshold τ of the biometric system. If a highly restrictive threshold value is set in the system, fewer attacks are consequently accepted than with a less restrictive threshold value. This might lead to a system with an unrealistically restrictive threshold that takes a high FNMR and a very low MMPMR without being able to separate the attacks from bona fide attempts. In order to take the factor of the set threshold into account, the Related Morph Match Rate (RMMR) as defined in [139] can be estimated. The RMMR considers the difference between MMPMR and True Match Rate (TMR) (TMR $= 1 -$ FNMR) at the threshold and is defined as follows:

$$\begin{aligned} RMMR(\tau) &= 1 + (MMPMR(\tau) - (1 - FNMR(\tau))) \\ &= 1 + (MMPMR(\tau) - TMR(\tau)) \end{aligned} \tag{11.4}$$



(a) RMMR of a reasonable system with a convenient threshold

(b) RMMR of a reasonable system with a restrictive threshold

(c) RMMR of a reasonable system with a highly restrictive threshold

(d) RMMR of a reasonable system with an unrealistic strict threshold

(e) RMMR of a highly vulnerable system an unrealistic strict threshold

(f) RMMR of an unrealistic system preferring morphs over bona fide images samples.

Figure 11.4: Examples of RMMR values in different systems with different threshold configurations

Examples of MMPMR and RMMR for different systems with differently set thresholds are given in Figure 11.4. The first example (Figure 11.4a) shows the PDFs for a reliable FRS. The genuine and impostor distributions are clearly separable, the distribution of attacks (mated morphs) is situated slightly in the genuine distribution. Since the threshold is set to a very convenient level (low FNMR), not only all genuine comparisons, but also all mated morph comparisons are accepted, resulting in an RMMR of 1. If the threshold is more stringent (Figure 11.4b), all genuine comparisons are still accepted, but half of the mated morphs are rejected, resulting in an RMMR of 0.5. If the threshold is set even stricter (Figure 11.4c), almost all mated morph comparisons are rejected, but also some of the genuine comparisons, the resulting RMMR is 0.07. If all mated morph comparisons are to be rejected, the threshold has to be set even stricter (Figure 11.4d). However, this will also cause half of the genuine comparisons to be incorrectly rejected. Thus, the RMMR increases to 0.5 again. If the genuine and mated morph comparisons are inseparable, as shown in Figure 11.4e, approximately the same number of mated morph comparisons as genuine comparisons are accepted almost independently of the threshold, resulting in an RMMR around 1. If the mated morph comparisons are accepted more often than genuine comparisons (Figure 11.4f), RMMR values greater than 1 are possible. It is important to note that in all cases the MMPMR should never be higher than the TMR, otherwise the intra-subject similarity would be increased by the morphing process, which is an unrealistic pre-requirement.

11.2.2 *Theoretical System Vulnerability Assessment*

If no concrete values for the evaluation of the system's vulnerability are available, it is possible to evaluate the system using the framework described in [47] and [48]. The main assumptions are, firstly, that the impostor comparisons are Gaussian distributed and, secondly, that the morph comparison score is exactly midway between the mean of the impostor distribution and the genuine comparison score of the sample used for morphing. Thus the distribution of the morph comparisons is approximated by averaging the Genuine distribution and the mean value of the Impostor distribution. For a specific threshold value, the approximated distribution can be used to predict the susceptibility of the system to morphing attacks.

11.2.3 *Morphing Attack Detection Performance*

To evaluate the performance of MAD algorithms, each comparison is considered individually, since each morph has to be detected separately. For this reason, the metrics defined in ISO/IEC 30107-3 [65] for the performance reporting of presentation attacks can be used here, namely Attack Presentation Classification Error Rate (APCER) and Bona Fide Presentation Classification Error Rate (BPCER), which are defined as follows:

APCER: proportion of attack presentations using the same Presentation Attack Instrument (PAI) species incorrectly classified as bona fide presentations in a specific scenario [65].

As an equation, the APCER can be expressed as follows:

$$\mathrm{APCER} = 1 - \left(\frac{1}{N_{PAIS}}\right) \sum_{i=1}^{N_{PAIS}} \mathrm{Res}_i, \tag{11.5}$$

whereas N_{PAIS} is the number of attack presentations for the given presentation attack instrument species. This can be replaced by the number of MA samples. Res_i is equal 1 if the i^{th} presentation is classified as an attack presentation, and 0 if classified as a bona fide presentation.

BPCER: proportion of bona fide presentations incorrectly classified as presentation attacks in a specific scenario [65].

As an equation, the BPCER can be expressed as follows:

$$\mathrm{BPCER} = \frac{\sum_{i=1}^{N_{BF}}}{N_{BF}}, \tag{11.6}$$

whereas N_{BF} is the number of bona fide presentations. Again Res_i is equal 1 if the i^{th} presentation is classified as an attack presentation, and 0 if classified as a bona fide presentation.

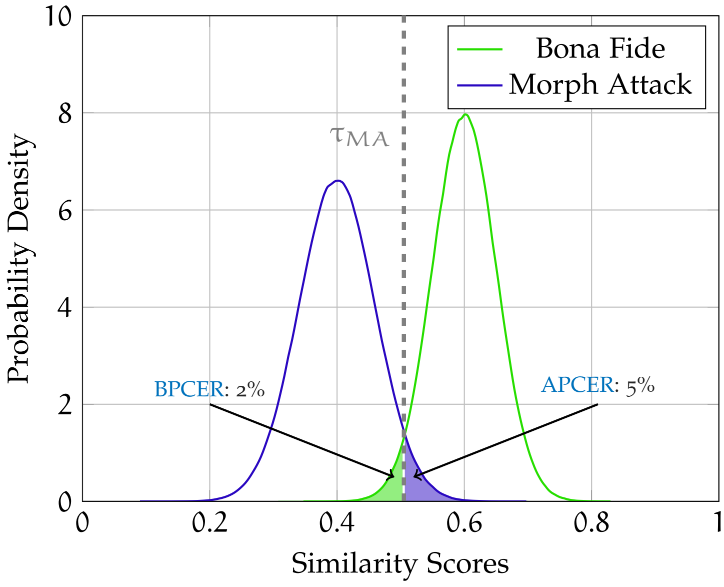


Figure 11.5: Visualization of ACPER and BPCER

As depicted in Figure 11.5, APCER and BPCER of a MAD system are comparable to FMR and FNMR of a FRS. In a proper MAD system, the resulting MAD scores of MA and bona fide samples should be clearly separable. For overlapping PDF curves, a trade-off between security (low APCER) and high throughput (low glsbpcer) has to be found by setting the threshold τ_{MA}.

11.2.4 *Equal Error Rate*

As described in Section 7.3, the EER is not standardized. In Biometrics it is usually interpreted as the operating point of equal FMR and FNMR, thus, in order to avoid confusion, the operating point of equal error between APCER and BPCER will be referred to as Detection Equal Error Rate (D-EER).

11.2.5 *Detection Error Trade-off Plots*

The D-EER merely reflects the error rates in a single operating point. However, if algorithms are to be compared independently of the operating point, the Detection Error Trade-off (DET) plot is recommended. A DET plot corresponding to the PDFs shown in Figure 11.5 is given in Figure 11.6. In the case of MAD systems, the APCER is plotted in relation to the BPCER. If a low APCER is required (very strict threshold), it leads to a higher BPCER and vice versa. In an optimal system, the DET plot would therefore be located in the lower left corner of the graphic. For a fixed APCER, this will result in a fixed BPCER for the system. These BPCER values can be reported, e.g. as BPCER-10 (for an APCER of 10%) or BPCER-20 (for an APCER of 5%). In the example shown, the DET plot consists of a straight line. This is not common and is due to the fact that the PDFs shown in Figure 11.5 are pure Gaussian distributions.

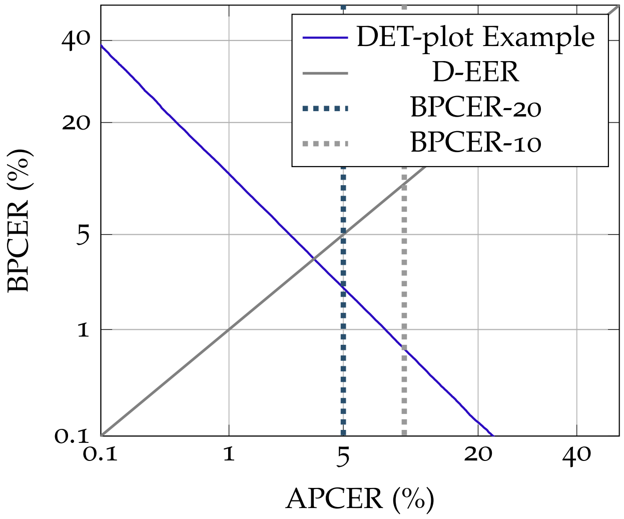


Figure 11.6: Example of an DET-plot of PDF-plot shown in Figure 11.5

12 CURRENT STATE-OF-THE-ART IN MORPHING ATTACK DETECTION

In recent years, since the first description of the problems of face morphing attacks in [39], many articles proposing new MAD techniques have been published. In this chapter the existing publications are listed in Table 12.1, 12.2, 12.3 and 12.4. If more than one approach was proposed in a publication, only the algorithm performing best according to the publication will be listed in the table. The proposed algorithms can be divided into two classes according to the scheme described in Section 11.1: Single image and differential MAD algorithms.

Many of the algorithms show promising results on the databases used for testing, but it has to be considered that mostly small databases with a lack of variance were used for training and testing, thus it is difficult to give a clear statement about the generalisation capability and robustness of the algorithms. Further, since the morphing databases are mainly in-house databases, no direct comparison between the algorithms is possible. For these reasons, the reported performance of the individual algorithms from the literature will not be discussed. For comparable performance evaluations, algorithms can be tested on common platforms under the same conditions, for example the NIST FRVT MORPH (see Section 3.1.3) and the Face Morphing Challenge of the SOTAMD project (see Section 3.1.1).

At time of the commencement of the work for this thesis there were exclusively two publications on the topic of morphing attacks, namely the first description of the problem [39] and a first approach to the detection of morphing attacks using BSIF [120]. During the work on the topic many new papers were published. Some of them build on the work presented in this thesis, for some it is the other way round. Due to this interdependence, a separation into own work and related work would disturb the logical and chronological connection between the publications. For this reason, the works are presented collectively. A list of the publications published in the context of this thesis can be found on page vii.

12.1 SINGLE IMAGE MORPHING ATTACK DETECTION

In this section, the existing S-MAD algorithms are subdivided according to the features used for classification, as introduced in Chapter 6. Features used for MAD are Texture Descriptors, Image Forensics (including Image Noise Pattern) and Deep Features.

PUB.	APPROACH	MA	SOURCE DATABASE	POST-PROC.	REMARKS
[120]	BSIF + SVM	GIMP/GAP	in-house	-	-
[134]	BSIF + SVM	GIMP/GAP	in-house	print and scan	adapted DB of [120]
[137]	BSIF + SVM	triangulation + blending	FRGCv2 [114]	-	-
[136]	HOG + SVM	triangulation + blending	FRGCv2 [114], FERET [115], ARface [100]	-	cross database performance evaluation
[151]	LBP + SVM	triangulation + blending	FRGCv2 [114], FERET [115]	-	cross database performance evaluation
[25]	LBP + SVM	MorGan [25]	CelebA [91]	-	-
[123]	multi-channel-LBP + Pro-CRC	OpenCV	FRGCv2 [114]	print and scan	-
[124]	multi-channel-LBP + SRKDA	[125]	[125]	print and scan	-
[163]	high-dim. LBP + SVM	triangulation + blending + swapping	Multi-PIE [52]	-	-
[5, 69]	ULBP + RIPS + KNN	[97]	Utrecht [160]	-	-
[133]	MB-LBP + SVM	triangulation + blending + swapping	FRGCv2 [114], FERET [115]	print and scan, rescaling, JP2000 compression	cross database performance evaluation
[1]	WLMP + SVM	Snapchat	in-house	-	-
[135]	general purpose image descriptors + score-level fusion	triangulation + blending	FRGCv2 [114]	-	-

Table 12.1: Relevant S-MAD algorithms based on texture descriptors

TEXTURE DESCRIPTORS Relevant algorithms based on texture descriptors are listed in Table 12.1. In the first publication on MAD [120], the use of BSIF and an SVM for the detection of morphed images is proposed. In previous publications, the combination of BSIF features with an SVM has shown to be a suitable method for similar problems, for example for the detection of presentation attacks [118]. However, [134] proved that with a realistic separation of training and test data, the detection performance of BSIF and SVM decreases significantly for MAD especially if the images were first printed and scanned (as is expected for passport photos). The same approach was analysed again in [137] and it is shown that if the filter parameters of the BSIF and hyperparameters of the SVM are chosen properly and a database of sufficient size is available, good detection rates can be achieved. However, this was only evaluated on a single database (with clear separation of training and test set) and without post-processing.

If strong differences in the quality of the databases exist, the detection performance drops considerably. A low but stable performance is shown for HOG features in combination with an SVM in [136]. Furthermore, LBP is able to achieve a good detection performance of morphed facial images [151], even across databases. In [25] LBP is not only tested against conventionally generated morphs (as described in Chapter 10), but also against morphs generated by a DNN (a so called GAN). These morphs can also be detected as well, the detection performance increases if the morph type to be detected is included in the training data. Furthermore, there are several suggestions for the extension of the classical LBP aiming to improve the detection performance of the resulting algorithm. For instance, in [123] it is proposed to perform the LBP extraction not on the greyscale image but separately on each colour channel of the colour image, to fuse the channels at feature level and subsequently process the resulting feature vector with an Spectral Regression Kernel Discriminant Analysis (SRKDA) classifier. Unfortunately, this approach is not compared to an SVM based approach, thus no conclusions about the advantages of feature extraction on individual colour channels and the advantage of using a different classifier can be drawn. In [163] it is proposed to extract a high-dimensional LBP as described in [17]. For this purpose, LBP is extracted in different scales and merged into a high-dimensional feature vector, subsequently it is classified employing an SVM. A fusion of several LBP extractors using a Vietoris-Rips complex is proposed in [5] and [69]. The Vietoris-Rips complex is built using the response of the uniform LBP (a simplified version of the LBP using a reduced number of predefined LBP filers). The ideas from [123] and [163] are combined and extended in [124]. In this paper it is proposed to extract LBP features with three different scales for the two colour spaces (*HSV* and *YCbCr*) on all three colour channels. One SRKDA classifier is trained for each scale and channel, finally a score level fusion is performed. Although the proposed algorithms were partially evaluated on printed and scanned images, the robustness against database changes and further alterations in the testing environment are not analysed. The robustness of LBP based MAD algorithms against these changes is investigated in [133]. Different combinations of LBP scaling and cell subdivisions in combination with an SVM are merged in a score level fusion, in order to compensate for the sensitivity of LBP to variations in the testing environment. It is shown that the proposed algorithm is very robust against variances occurring in a natural scenario. Experiments are performed on four different morphing algorithms, different post-processing methods (print and scan, rescaling, JP2000 compression), and two different databases. Furthermore, in [1] a variant of LBP is proposed as a MAD algorithm, called Weighted Local Magnitude Patterns (WLMP). Instead of the binarization performed in the standard LBP, the individual fields of the LBP are weighted according to the previously calculated differences.

The advantage of the fusion of several algorithms based on different feature extractors is investigated in [135]. It is shown that especially the fusion of LBP or BSIF with additional feature extractors can improve the robustness of the resulting algorithm.

IMAGE FORENSICS Relevant algorithms based on image forensics are listed in Table 12.2. During the creation of morphed facial images,

PUB.	APPROACH	MA	SOURCE DATABASE	POST-PROC.	REMARKS
[81]	image degradation	triangulation + blending (+ swapping)	in-house, Utrecht [160]	-	-
[27, 28]	PRNU analysis	triangulation + blending	FRGCv2 [114]	hist. equalization, scaling, sharpening	-
[171]	SPN analysis	triangulation + blending (+ swapping)	Utrecht [160], FEI [158]	-	-
[132]	PRNU analysis	triangulation + blending (+ swapping)	FRGCv2 [114]	print and scan	further analysis of behaviour of PRNU on Dresden image database [46]
[26]	PRNU analysis	MorGan [25]	CelebA [91]	-	-
[161]	Denoise-CNN + Pyramid LBP + SRKDA	[123], [151]	FRGCv2 [114], PUT [73], in-house	-	-
[125]	luminance component +steerable pyramid + ProCRC	unclear	[123] extended	print and scan	-
[97]	double-compression artefacts	triangulation + blending (+ swapping)	Utrecht [160], FEI [158]	-	-
[58]	double-compression artefacts	[97]	Utrecht [160], FEI [158]	-	-
[141]	reflection analysis	triangulation + blending (+ swapping)	in-house	-	-

Table 12.2: Relevant S-MAD algorithms based on image forensics

at least two facial images are manipulated (warping) and merged (blending). Hence, it is reasonable to attempt to detect these manipulations by means of image forensic techniques. One possibility is to try to detect the morphed face images on the basis of a reduced image quality of the morphed images, which is investigated in [81]. However, this approach is only applicable if a lower quality of morphs compared to bona fide images can be assumed. However, in reality this assumption is mostly incorrect.

Another possibility is to analyse the noise pattern of the images. For this purpose PRNU or SPN, as described in Section 6.6, can be applied, which are usually applied to detect image manipulations or to determine the sources of images. The information extracted by PRNU or SPN can be used in explicit algorithms, for example by analysing individual parameters such as position or value of the maximum of the Discreet Fourier Transformation (DFT) magnitude histogram of the noise pattern [28] or by comparing the DFT magnitude histograms of different areas in the image [27]. Furthermore, [171] introduces an algorithm which implicitly processes the information contained in the noise pattern by means of an SVM. In [132] the effect of morphing on the noise pattern is investigated in more detail and the generalisability of PRNU based algorithms across different camera types is shown. In addition, it was observed that PRNU is able to robustly detect morphs generated with a GAN [26].

The algorithm proposed in [161] follows a similar concept as PRNU and SPN. The method used to extract the noise pattern is the denoising CNN presented in [169]. The noise pattern is extracted per *HSV* color channel. Subsequently, an LBP pyramid in combination with an SRKDA classifier is applied to the extracted noise pattern, which, according to the authors, captures the noise patterns.

In [124] another algorithm for the analysis of high-frequency image information is proposed. Even though, the algorithm analyses texture features, it is categorized as an image forensic algorithm due to its focus on high-frequency information. First, the image is converted into a greyscale image based on luminance, subsequently the high-frequency information is extracted from this image. A Collaborative Representation Classifier (CRC) classifier [172] is trained on the resulting high-frequency image.

An approach presented in [97] and [58] proposes to detect morphed facial images by means of double compression artefacts. This approach is based on the assumption that by repeatedly saving with lossy compression during the morphing process, detectable artefacts are introduced into the image. However, since it can be assumed that an attacker will always choose the technically best option and thus be saving the images in raw format or with lossless compression, this assumption does not hold in a real scenario.

Furthermore, it is possible to detect morphed facial images on basis of inconsistencies in the facial image. In [141] it is proposed to analyse the reflections in the face, as it can be assumed that they change unnaturally as a result of the morphing process. Using a digital 3D reconstruction, the expected reflections are approximated and compared to the reflections in the image to be checked. Although this concept is very interesting, it will not be used for morph detection in passport photos, as the ISO/IEC standard requires the absence of hot spots and reflections in facial images used in electronic travel documents. In particular, diffuse lighting, multiple symmetrical sources or other lighting methods should be used, i.e. a single bright "point" light source such as a camera-internal flash is not acceptable for imaging [63].

DEEP FEATURES As described in Section 6.5, any feature extractors can also be modelled by NN, extracting so called deep features. Relevant algorithms based on deep features are listed in Table 12.3.

Since large amounts of natural training data are required for the training of DNNs, which are usually not available in the area of MAD, the proposed algorithms use pre-trained networks or adapt them by transfer learning. In [122] it is proposed to modify two CNNs, namely VGG19 [146] and AlexNet [82], by transfer learning in order to adapt them to the MAD problem and apply the intermediate features of both algorithms to train a CRC. A deeper analysis of the use of CNN features for the detection of morphed facial images in

PUB.	APPROACH	MA	SOURCE DATABASE	POST-PROC.	REMARKS
[122]	VGG19 + AlexNet + ProCRC	[134]	in-house	print and scan	-
[143]	VGG19	triangulation + blending (+ swapping)	BU-4DFE [168], CFD [95], FEI [158], FERET [115], PUT [73], scFace [51], Utrecht [160], in-house	motion blur, Gaussian blur, salt-and-pepper noise, Gaussian noise	trained on all combinations (no unseen attack classes)
[142]	VGG19 + GoogLeNet + AlexNet	triangulation + blending (+ swapping)	in-house	-	-

Table 12.3: Relevant S-MAD algorithms based on deep features

general, is presented in [142]. For this purpose, three different CNNs are benchmarked, each pre-trained and trained from scratch, namely VGG19 [146], AlexNet [82] and GoogLeNet [156].

The publications described above mostly ignore the issue of over-fitting, as pointed out in Section 5.5. Due to the fact that morphing attacks for the training and test set database are usually generated by one algorithm from one database, there is a high risk that the algorithms will over-fit on the particular data and that this over-fitting will not be detected during evaluation. In [143] an attempt is made to analyse this problem in more detail. For this purpose the images are processed with various post-processing (motion blur, Gaussian blur, salt-and-pepper noise, Gaussian noise). In addition, in some images certain regions are occluded (eyes, nose, mouth) to prevent over-fitting to artefacts occurring in these areas. Since training and testing is carried out on all combinations, a statement about the influence of the individual factors is not possible.

The training of the DNNs is usually performed on the entire image, thus it is difficult to determine which areas of the facial image are considered by the classifier. A rough idea can be given by visualizing the weights of the input pixels in a heat-map, showing which areas tend to have a higher weight and therefore a stronger presence in the feature vector [143].

12.2 DIFFERENTIAL MORPHING ATTACK DETECTION

Differential MAD algorithms have the conceptual advantage over S-MAD algorithms that they have access to the information of the TLC in addition to the suspected morph. Thus, these algorithms are potentially able to work more robust [137]. For processing this additional information, two types of algorithms are known, in this thesis divided into *feature comparison* and *morphing reversion*. The most important, existing algorithms are listed in Table 12.4.

PUB.	APPROACH	MA	SOURCE DATABASE	POST-PROC.	REMARKS
		FEATURE COMPARISON			
[137]	differential BSIF + SVM	triangulation + blending	FRGCv2 [114]	-	-
[131]	landmark angles	OpenCV	ARface [100]	-	-
[24]	directed distances of landmarks	triangulation + blending (+ swapping)	FERET [115]	-	-
[148]	SfSNet [145] + AlexNet [82] + SVM	[123]	in-house	printed and scanned	-
[138]	ArcFace [100] + SVM	triangulation + blending + swapping	FRGCv2 [114], FERET [115]	print and scan, rescaling, JP2000 compression	cross database performance evaluation
		MORPHING REVERSION			
[40]	Demorphing	GIMP/GAP	ARface [100]	-	-
[41]	Demorphing	GIMP/GAP	ARface [100], CAS-PEAL-R1 [45]	-	CAS-PEAL-R1 contains images with pose variations
[112]	DNN-Demorphing	triangulation + blending	in-house	-	-
[109]	DNN-Demorphing	triangulation + blending + swapping	in-house	-	-

Table 12.4: Differential algorithms

FEATURE COMPARISON An intuitive approach to incorporate the features of the TLC is a subtraction with the feature vector of the suspected morph. It is expected that features of the subject contained in both, the suspected morph and the TLC, will be reduced and any differences between the morph and the subject will be amplified, resulting in a more robust MAD. In [137] it was shown that this may improve, e.g. the detection performance of BSIF based MAD algorithms.

Another approach is to observe differences in facial geometry between the two images. During the warping process, the geometry of the morph is changed due to the warping process. In [131] an attempt is made to detect these changes by measuring the angles between the landmarks, but unfortunately no stable detection performance can be achieved. Damer et al. are refining the approach in [24] and propose the use of directed distances between the landmarks.

A further approach is proposed in [148]. Using an CNN (SfSNet [145]) the image is decomposed into a normal image, an albedo image (which represents reflectance) and a shading image (which represents illuminance). From shading and the albedo image a reconstruction of the original image is generated. The reconstruction of the suspected morph and the TLC are combined in a further CNN (AlexNet [82]), which generates a feature vector, which is classified using a linear SVM. In addition, feature vectors are extracted from both normal images using AlexNet and classified using a linear SVM. The results of both classifiers are combined in a weighted score level fusion.

Although DNNs offer a great risk of undetected over-fitting, they can still be used to extract robust feature vectors. In [138] it is shown that for example the feature vectors of the ArcFace FRS [100] can be used to perform robust MAD. For this purpose the ArcFace features of the suspected morph and the TLC are extracted and subtracted from each other. The difference vector is used to train an SVM with a polynomial kernel. The advantage of this procedure is that the feature extractor is not adapted. I.e. it was trained for face recognition, so no morphed face images were present in the training set, thus it can be excluded that the feature extractor is over-fitted for certain morphing artefacts. The robustness of the resulting overall algorithm is demonstrated on different databases with various post-processings (print and scan, rescaling, JP2000 compression).

MORPHING REVERSION The objective of morphing reversion is to use the TLC to invert the morphing process, such that, in the case of a morphed image, the TLC does not match the demorphed image during a comparison. This concept was first proposed in [40]. However, this approach encountered the limitation, that even a slight pose variance, which can occur especially with TLCs recorded under semi-controlled conditions, strongly influences the result of the demorphing process. Therefore, [41] proposed an improved version of demorphing, which normalizes the images previous to the demorphing process. In [112] and [109] it is proposed to map the process of demorphing by a DNN.

13 SUMMARY

In this part, the publications related to the topic of the thesis are discussed. First, the state-of-the-art in face morphing is summarised. The technique is based on the steps described in Chapter 8: Determining correspondence, warping and blending. The different methods for determining the correspondences are described. Those are mainly implemented by landmark detection based on active shape models. For the warping and morphing process the images are usually divided into Delaunay triangles, which are subsequently distorted and blended. Since this simple method of morphing is prone to produce artefacts, especially in regions of the image where not enough or too imprecise landmarks are located, Section 10.4 introduces several techniques to improve the resulting morph, namely swapping, artefact replacement and manual post-processing.

Furthermore, the basics of algorithms for the detection of morphed facial images, so-called MAD algorithms, are discussed in this part. Basically, these algorithms can be separated into two classes: single image and differential detection schemes. In the single image scheme only the information of the morph to be evaluated is available, in the differential scheme a live image (which can be assumed to be a bona fide) is available in addition to the potential morph.

In order to allow an objective evaluation of the morphing attacks and MAD algorithms, two classes of metrics are presented. On the one hand, with the MMPMR, an adapted variant of the IAPMR standardised in ISO/IEC 30107-3 [65] is presented for the evaluation of FRS vulnerability, as well as with the RMMR a variant of the metric depending on the performance of the FRS, on the other hand, metrics for measuring the performance of MAD systems are given with the APCER and BPCER standardised in ISO/IEC 30107-3 [65].

Finally, Chapter 12 provides a comprehensive overview of the current state-of-the-art of MAD algorithms. Divided into single image and differential MAD algorithms, as well as subdivided according to the analysed features, the approaches presented in the publications are listed and evaluated. Since the proposed algorithms have been tested on inconsistent databases with partially inconsistent metrics, a direct comparison of the algorithms is not available.

Part IV

MORPHING ATTACK DETECTION PIPELINE

14 DESIGN DECISIONS

In the context of this thesis, MAD algorithms based on different features were created, in this part the structure of the individual algorithms is described in more detail. In order to minimize the development effort, the algorithms are modular in design, which allows to adapt the algorithms by exchanging single modules. The individual modules of the pipeline for the creation of new algorithms are illustrated in Figure 14.1. It consists of the following 4 steps: data preparation, feature extraction, feature preparation, and classifier training.

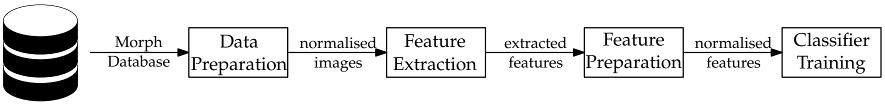


Figure 14.1: Design of MAD pipeline

DATA PREPARATION For most feature extractors applied, it is necessary to prepare the image data to be processed beforehand, similar to the pre-processing in FRS. In Chapter 6 it has already been shown, that the result of most feature extractors depends on the resolution of the analysed image, requiring a normalisation of the image size. Furthermore, especially with the TLCs, variances in position and pose may occur, which can be corrected by the data preparation. In addition, it is useful, for example for texture-based feature extractors, to crop the image to the relevant facial area, ensuring that no information from the background influences the feature vector. The different *data preparation* methods of the MAD pipeline are described in Chapter 15.

FEATURE EXTRACTION The next module is the *feature extraction*, controlling which information of the image is used in the further process. Depending on the feature extractor selected and the configuration, the feature vector will contain different information, information not contained in the feature vector is not available to the algorithms in the further process. For example, if a basic LBP histogram is calculated as described in Section 6.1.1, the feature vector will not contain any spatial information. If, despite the use of LBP histograms, spatial information is to be included in the feature vector, the image to be analysed can be divided into cells, a histogram can be calculated for each cell and the resulting histograms can be concatenated. Thus, spatial information in resolution of the cells can be preserved, however, the length of the feature vector increases accordingly. The concepts of the feature extractors examined in this thesis are described in Chapter 6, details on the implementation and configuration of the individual feature extractor are given in Chapter 16.

FEATURE PREPARATION Once the feature vectors have been created, they have to be prepared for the training of the classifier. For example, many classifiers only accept one-dimensional input data, requiring multi-dimensional characteristics to be prepared accordingly. Further, for differential MAD algorithms, this module combines the feature vectors of the suspected morph and the TLC. The choice of the combination method is arbitrary, but determines the length of the resulting feature vector as well as the contained information. A description of the *feature preparation* for differential and single image MAD is given in Chapter 17. Most classifiers require normalized data for optimal training, thus the feature vectors are zero-centred and calibrated to a scale between -1 and 1. The feature normalization used in this processing pipeline is described in Section 17.3.

CLASSIFIER TRAINING In the last module classifiers are trained on basis of the previously prepared feature vectors. In order to achieve the best possible separation of the feature vectors into classes, appropriate classifiers and parameters have to be chosen. The optimal classifier and parameters depend on the information in the respective feature vectors. The functional principle of the classifiers applied in this work is described in Chapter 5, the *classifier training* module is described in Chapter 18. The extracted features can usually be visualized and thus the proper functioning of the algorithms can be verified. The classifiers, on the other hand, can hardly be visualized and an analysis of the trained classifier is, if at all, only possible with great difficulty, meaning that errors in the training may not be noticed, even during the evaluation, and thus the outcome of the evaluation may be inconclusive. In order to minimize this risk, basic principles during training should be considered, which are described in Section 18.1.

15 DATA PREPARATION

The training data of the classifiers, should sufficiently represent the variance of the data to be classified, in order to obtain a robust model. At the same time, however, unnecessary variances should be reduced in order to keep the training process and the classifier model as simple as possible.

In the case of facial images, the natural variance includes, for example, the pose of the face, the position in the image or lighting and shadows. To minimize this variance in passport photographs, ICAO recommends following the standard for capturing facial images defined in [63]. Even if, consequently, a certain level of image quality can be assumed in passports, and thus for the suspected morphs to be analysed, a certain variance remains. For the TLCs, a significantly higher variance is to be expected, as the capture process is semi-controlled, thus no constant quality can be expected.

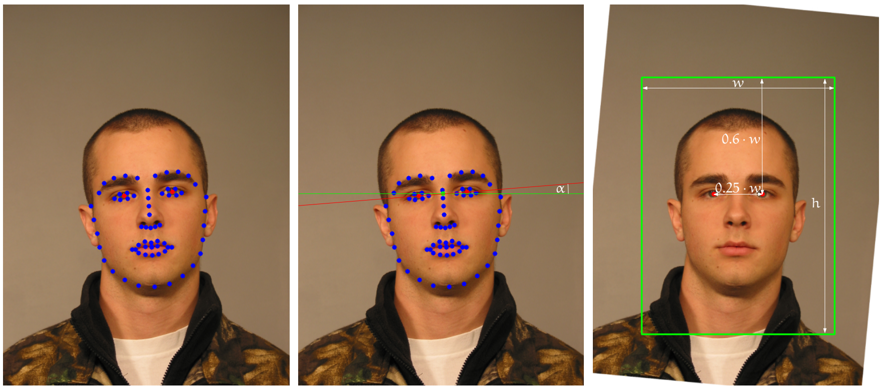
To reduce the variance of the facial images, a normalisation of the positioning of the face in the image prior to the feature extraction can be performed. If identical normalisation procedures are carried out during training and in the later operation, the variance of the training data as well as the data to be classified is reduced simultaneous, which simplifies the training of the algorithm and increases its robustness. The following section describes the technical implementation developed in this work for normalisation of position, slight pose variations and image size.

15.1 IMAGE NORMALISATION

In the normalisation applied in this work, the image is changed exclusively by rotation (horizontal alignment) and scaling (adjustment of the size of the image), which largely avoids changes in texture and color values. In rare situations, pixels have to be approximated during the operations, which may result in a slight change of the image at pixel level. However, this effect is negligible compared to the benefits of normalisation.

The first step of normalisation is the horizontal alignment of the image by rotation. Since faces seldom exhibit perfect symmetry, it is necessary to determine fixed points for orientation during alignment. The images in this work are normalised according to the centre of the eyes. On the one hand, they are automatically detectable with good accuracy, on the other hand, the definition for biometric passport images described in [63] refers to the position of the eyes. Furthermore, the minimum size of passport photos is also defined by the inter-eye distance.

Technically, the detection of the eyes is based on landmarks. As described in Chapter 6.4, most landmark detection algorithms do not detect the centre of the eye, but only the pixels surrounding the eye. Thus, the average of all eye-surrounding landmarks is calculated, providing a suitable approximation of the centre of the eye. In Figure 15.1a, an example of the approximated eye centre is given. The detected landmarks are coloured in blue, the approximated landmarks are coloured in red. Once the eye centres have been estimated, the horizontal alignment of the image can be commenced. For this purpose, the eye centres are brought to a horizontal line by rotating the image. First, the angle α between the line through both eye centres and the horizontal, as visualised in Figure 15.1b, is calculated:



(a) Example of approximated eye centres (coloured in red)

(b) Visualization of angle calculation

(c) Visualization of cropping calculation

Figure 15.1: Example of face normalisation

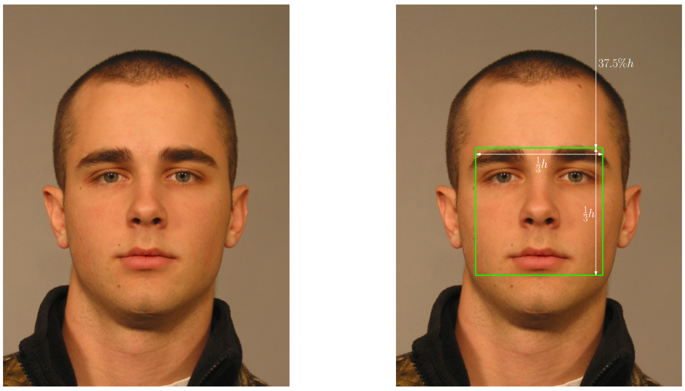
$$\alpha = \tan^{-1}\left(\frac{eye^r_x - eye^l_x}{eye^r_y - eye^l_y}\right), \tag{15.1}$$

whereas eye^r_x refers to the x-coordinate of the right eye, eye^l_y to the y-coordinate of the left eye and vice versa. Based on the calculated angle, a rotation matrix is determined, according to which the image is rotated by affine transformation around the centre between both eyes. As a result of the rotation the position of the eye centre point in the image changes, thus the new position has to be determined. The y value for the position of both eye centres is the average of the y values of the eye centres prior to rotation. The x value corresponds to the previous distance to the centre of both eye centres (the centre of rotation), which can be estimated by the Pythagorean theorem:

$$\begin{aligned} rotatedEye^l_x &= c_x - \sqrt{(eye^l_x - c_x)^2 + (eye^l_y - c_y)^2} \\ rotatedEye^r_x &= c_x + \sqrt{(eye^r_x - c_x)^2 + (eye^r_y - c_y)^2}, \end{aligned} \tag{15.2}$$

whereas c refers to the coordinates of the rotation point.

Once the horizontal alignment is applied, the face images should be cropped, such that the faces are aligned at a constant size in a constant position of the image. For this purpose, the guidelines of ISO/IEC 19794-5 [63] are followed. For biometric passport photographs, a minimum face width and height in the image is requested. Since these two parameters are difficult to measure in an automatic manner, the standard for token images is applied instead in this thesis. According to the standard, the facial images will be cropped according to the centre of the eyes, which are given, due to the previous horizontal alignment. It is specified that the inter-eye distance should be 25% of the image width, and the eyes should be horizontally centred in the image. The distance of the eyes to the upper edge of the image should be 60% of the image width. An example of the cropping of a facial image is depicted in Figure 15.1c, the facial image would be cropped to the dimensions displayed in green. In addition to the alignment of the face in the image, [63] requires a minimum resolution for biometric passport images. The minimum resolution is defined to correspond to at least 180 pixel width of the head, which, according to the standard, is equivalent to an inter eye distance of at least 90 pixels, resulting in a minimum image resolution of 360 × 480 pixels, higher resolutions are admitted as well. An example of a cropped full face image is given in Figure 15.2a.



(a) Example of a cropped full face image

(b) Example of the cropping of the facial area

Figure 15.2: Example of close crop of the facial area

15.2 IMAGE CROPPING

As described in Chapter 14, some feature extractors require that only the relevant areas of the image are passed. In this work, the cropping of the facial area has been chosen to ensure that no background is present on the cropped areas and that artefact-threatened areas, such as the neck or hair, are omitted. For full face images normalised according to Section 15.1, a suitable cropping was determined to be a square with edge length of $\frac{1}{3}$ of the image height, centred horizontally and at a distance of 37.5% of the image height from the top edge of the image. An example of this cropping is shown in Figure 15.2b.

16 FEATURE EXTRACTION

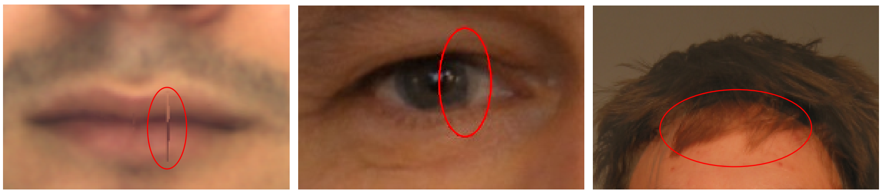
After normalising the images, the feature extraction module extracts the features relevant for classification. Each feature vector transfers the image, which is represented as a matrix, into a typically lower dimensional space, which promises a higher discriminant power for the respective problem, making it easier to separate the data. In this thesis different categories of features are investigated, which contain mostly different information. The following classes of feature extractors are implemented in this framework: Texture descriptors essentially describe the surface structure of the image. Gradient information is extracted either globally, for example via HOG, or locally via SIFT and SURF. An extraction of pure spatial information is provided by landmark detection algorithms. For the analysis of noise patterns PRNU or SPN based approaches can be used. If the features are to be determined in a purely probabilistic way, DNNs can be used to extract so-called deep features. This chapter motivates the use of the selected algorithms whose concepts were presented in Chapter 6 and describes them in more detail regarding implementation and configuration.

16.1 TEXTURE DESCRIPTORS

During the creation of morphed facial images, the morphing process introduces changes into the image that can be used to detect said images. In particular, these changes are reflected by faulty regions, such as overlapping landmarks, which result in incorrectly distorted triangles, as shown in Figure 16.1a. Another error common to automated morphing algorithms are artefacts in the eye region, which is particularly prone to errors due to the high contrast provided by shadows and wrinkles, and the difficult detection of the iris as described in Section 10.1. An example of artefacts in the eye region is given in Figure 16.1b. Furthermore, ghost artefacts can be caused by landmarks that are too few or too poorly positioned. This happens frequently in the area of the neck or hair, as visualized in Figure 16.1c. In order to be able to map this kind of image changes in feature vectors, texture descriptors can be used. In this thesis the suitability of LBP and BSIF for describing these artefacts is investigated.

16.1.1 *LBP*

The basic concept of the applied LBP is described in 6.1.1. The LBP implementation uses the image normalized and cropped according to



(a) Example of errors introduced by incorrectly distorted triangles (b) Example of errors in eye region (c) Example of errors in hair region

Figure 16.1: Example of errors introduced by incorrect morphing

Section 15.2 as input image. Prior to processing, the regions are scaled to a fixed size of 160 × 160 pixels to achieve an independence from the size of the analysed image. Although it was shown in [136] that the use of MB-LBP patches may contribute to the stability of the system, the performance of the classical LBP is far ahead of the MB-LBP in other scenarios [137], which can be explained by the high relevance of the detailed information, which cannot be mapped by MB-LBP, due the smoothing effect of the bigger patch. Thus, the classical LBP with a 3 LBP patch, as well as an MB-LBP with a 9 × 9 patch are implemented.

By calculating the LBP histogram, any local information contained in the image is discarded. To preserve local information, the LBP image can be divided into cells, subsequently a histogram is calculated for each cell. As a result, the length of the feature vector multiplies by the number of cells, but spatial information is obtained in resolution of the cell division. An inevitable correlation exists between cell division, patch size, image size and the resulting histogram. The finer the cell division and the larger the patch, the fewer values can be calculated per cell and the sparser the histogram. As the resolution increases, the number of values per cell increases as well. For the applied patch sizes and the region of 160 × 160 pixels, a subdivision into 4 × 4 cells has shown to be appropriate, thus it is implemented in addition to the LBP calculation without cell division.

16.1.2 *BSIF*

As a further texture descriptor, BSIF is implemented as described in Section 6.1.2. The implementation receives pre-cropped face regions, which are scaled to 160 × 160 pixels prior to the processing step. As for LBP, it has been shown that the use of larger BSIF patches results in more robust systems [136], but using smaller BSIF patches results in significantly higher performance [137]. In order to allow a better comparison to LBP, BSIF with a patch size of 3 × 3 and 9 × 9 pixels with 8 filters are used. The resulting feature vector of a length of 256 is directly comparable to that of the LBP.

As for LBP, the spatial information is lost during the calculation of the histogram, an effect which can be prevented by the previous division into cells. The behaviour of the feature vector is identical to that of LBP, which is why, also to ensure comparability, the same configuration as for LBP of no cell division and division into 4×4 cells is implemented.

16.2 GRADIENT BASED DESCRIPTORS

Morphing can be simplified as an averaging of two images. This process reduces the probability of the occurrence of extreme values, resulting in a smoothing of the image. This smoothing, which is hardly visible in the image, can be captured by gradient extraction. A basic approach is the calculation of the Mean of Gradients, a more elaborate method is the extraction of HOG features.

16.2.1 *Mean of Gradients*

This approach to extract gradient features is straight-forward. In a first step, the previously cropped face region is scaled to 160×160 pixels, afterwards a gradient image is calculated for x and y dimension, as described in Section 6.2.1. Finally, the mean value per gradient image is estimated, resulting in a feature vector of length 2.

Due to the calculation of the mean, any spatial information is dropped. Thus, in order to avoid said loss, a subdivision into cells can be applied. For comparability with previous algorithms, a subdivision into 4×4 cells is implemented in addition to the mean of gradients without cell division, resulting in a feature vector of length 32.

16.2.2 *HOG*

A much more elaborate extraction of the gradient information is achieved by calculating the HOG on the previously cropped face region, scaled to 160×160 pixels. As indicated in the description of the technical background of HOG in Section 6.2.2, considerably more parameters have to be configured, compared to the previous presented feature extractors. First, the region is already divided into cells by default, second, the number of discrete directions and thus the length of the histogram per cell has to be defined, finally the number of cells to be combined into a block has to be specified as well. The definition of the parameters influences the result of the histogram calculation, as well as the length and content of the feature vector. In order to achieve a robust and general applicable HOG extraction, recommended standard parameters[1] are applied, namely 9 orientations, 8 × 8 pixels per cell (which corresponds to 20 × 20 cells for regions of 160 × 160 pixels), and 3 × 3 cells per block, resulting in a feature vector of length 26,244.

16.3 KEYPOINT DESCRIPTORS

The feature extractors described in Section 16.2 transfer the entire provided image evenly into a feature vector. As described in Section 6.3, keypoint descriptors, such as SIFT and SURF, extract the features in a similar way to HOG, but with a focus on prominent regions. Thus, the resulting feature vectors do not provide a constant length, but depending on the number of detected prominent regions, meaning that the resulting feature vectors are not suitable for subsequent use in classifiers. For that reason, a modified variant of the keypoint descriptors is applied in this thesis. Under the assumption stated in Section 16.2, that the morphing process smoothes the generated image, the number of prominent points (e.g. edges) in the image decreases. Thus, the number of detected keypoints should allow a statement about whether an image has been morphed or not.

16.3.1 *SIFT*

Since it is assumed that morphed facial images can be identified by the number of detected keypoints, this implementation, as described in Section 6.3.1, initially computes the SIFT keypoints on the 160 × 160 pixels facial region, however, subsequently only the number of keypoints is assessed, discarding any further information extracted by SIFT. The parameters relevant to the number of keypoints, namely the number of octave layers, as well as the thresholds for contrast (to filter weak keypoints) and edge detection (to filter edges), are set according to the recommended default parameters[2]: 3 octvave layers, contrast threshold of 0.04 and edge threshold of 10.

Since the generated result is a scalar, it can directly be used as an MAD score, meaning that the training of classifiers can be omitted at this point. In order to avoid discarding the spatial information during the calculation, the facial region can be divided into cells prior to the counting of keypoints, determining the number of keypoints per cell. In this implementation, a subdivision of 4 × 4 cells was chosen, resulting in a feature vector of length 16.

[1] The standard parameters are derived from the documentation of the used HOG implementation: https://scikit-image.org/docs/dev/api/skimage.feature.html

[2] The standard parameters are derived from the documentation of the used SIFT implementation: https://docs.opencv.org/3.4.9/d5/d3c/classcv_1_1xfeatures2d_1_1SIFT.html

16.3.2 *SURF*

As described in Section 6.3.2, the functionality of SURF is very similar to that of SIFT, the major difference lies in the implementation of SURF optimised on the computational time. For this reason, the concept for extracting feature vectors described in 16.3.1 can be fully adopted, with a difference in the definition of the parameters. In the employed library[3], the number of octave layers and a threshold value for the determination of keypoints has to be defined. In this implementation, both values are set according to the recommendations of the employed library, namely 3 octave layers and a threshold value of 100 for the keypoints.

16.4 LANDMARK EXTRACTORS

During the morphing process, the two images to be morphed are distorted in a way, that corresponding landmarks overlap. Hence, it can be assumed that morphed images may be detected by measuring the landmark shift compared to the original. Due to the absence of the original image, the potentially morphed image has to be compared with an unaltered image of the subject, the TLC. Thus, landmark based methods are among those that can only be applied in the differential scenario. In [131] it is proposed to determine angles and distances between the landmarks and use these as a feature vector for classification, however, the resulting detection performance is not convincing. In [24] a refined approach is presented in which the distances between corresponding landmarks are considered separately for the x and y dimension. Since considerably better results are reported for the second approach (even if a direct comparison is not feasible, due to different databases and classifiers), the approach proposed in [24] is implemented in the applied MAD pipeline. For landmark extraction, the two algorithms presented in the following sections are employed.

16.4.1 *Dlib*

Dlib is a comprehensive library for machine learning and image processing [77]. This library contains an implementation for training shape predictors according to [74] and provides a pre-trained model for the detection of 68 landmarks in facial images.

Initially, a Region of Interest (RoI) is determined in which the landmarks should be located. For this purpose, a face detection is carried out applying HOG features and a linear SVM on an image pyramid. Based on the RoI, the landmark detection is conducted on the basis of an ensemble of regression trees. A pre-trained classifier is included

[3] The Implementation for SURF of the OpenCV library was applied: https://docs.opencv.org/3.4/d5/df7/classcv_1_1xfeatures2d_1_1SURF.html

in the library. The training is realized with the gradient tree boosting, described in Section 5.3.3. The constant function F_0, which is required for training, is described by the average position of each landmark in the RoI on the training data. This ensures that the detected landmarks are roughly face shaped. As a feature for landmark detection the algorithm utilises the difference of intensity values of pixel pairs. A detailed explanation for the choice of algorithms, parameters and features can be found in [74].

16.4.2 *WING*

A further approach for the extraction of facial landmarks is described in [36], where a CNN with a customised loss function is proposed. The network accepts a three-dimensional matrix of $64 \times 64 \times 3$ (x, y, and color channels) as input vector and returns a vector with the 2D coordinates of the landmarks. The proposed network is a basic CNN consisting of five convolutional layers of size 3×3, one fully connected layer and the output layer. The special characteristic is the use of the so called Wing-Loss instead of the L2-Loss, commonly used for landmark extraction. Wing-Loss was designed to provide the non-linear behaviour of a log function for small errors, compensating the influence of errors of different sizes. For large errors, however, the function behaves like an L2-error, in order to be able to adapt to large errors, e.g. strong pose variations. The resulting function is given as follows:

$$wing(x) = \begin{cases} w \ln(1 + \frac{|x|}{\epsilon}) & \text{if } |x| < w \\ |x| - C & otherwise \end{cases} \tag{16.1}$$

w limits the non-linear (upper) part of the equation to the interval $[-w, w]$ and ϵ defines the curvature of the non-linear function. $C = w - w \ln(1 + \frac{w}{\epsilon})$ is a constant depending on w and ϵ, which allows a smooth transition between the linear (lower) and the non-linear (upper) part of the equation. An example of Wing-Loss with $w = 5$ and $\epsilon = 0.5$ in comparison to L2-Loss (scaled by 0.1) is depicted in Figure 16.2.

The implementation[4], applied in this thesis, provides a pre-trained CNN which determines the position of 19 landmarks.

16.5 IMAGE NOISE PATTERN

The origin of images can be determined on the basis of the image noise pattern, a high frequency feature unique to each camera chip. Bona fide images are captured by a single camera mostly without

[4] The Matlab implementation for Wing-Loss landmark detection provided by the authors can be found at: https://github.com/FengZhenhua/Wing-Loss.

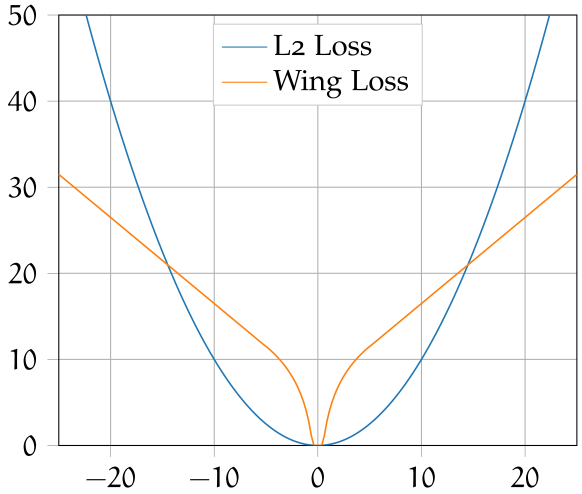


Figure 16.2: Example of L2-Loss and Wing-Loss Function

further processing, thus a single image noise pattern of one camera can be detected in the image. Morphed images, on the other hand, consist of a merge of at least two images, which may originate from different cameras resulting in a superposition of two different image noise patterns. In addition, the images to be morphed are distorted, which alters the image noise pattern. Based on these effects, it can be assumed that morphed images can be detected by the analysis of the image noise pattern. In the framework used for this work, two different interpretations of the image noise pattern are implemented, which are described in the following sections.

16.5.1 *PRNU*

Likewise the previous algorithms, this algorithm operates on the cropped facial regions of normalized images, initially scaling them to 160×160 pixels to produce results independent of the image size. The extraction of the PRNU is basically only a high pass filtering of the image. For this purpose the image I is denoised with a denoising function $F(I)$, the PRNU corresponds to the residuals W_I of the difference between the image and the smoothed image:

$$W_I = I - F(I). \tag{16.2}$$

The resulting PRNU depends on the choice of the filter function. For the PRNU extraction in this work, the filter suggested in [102] was applied.

For further analysis of the extracted PRNU, the system proposed in [132] is utilized. Different concepts have been proposed in several publications [28], [27], [132], however, the MAD pipeline implemented for this work is limited to two approaches, which have proven to offer a high performance [132], both in testing on the database and in robustness across different camera types.

For both concepts the PRNU signal is initially divided into 10×10 cells. The first approach calculates a histogram H_p of the distribution of the PRNU values per cell. Per histogram the variance is calculated, resulting in a scalar value per cell:

$$P_{var} = \frac{1}{B} \sum_{n=1}^{B} (H_P(n) - \overline{H}_P)^2, \tag{16.3}$$

whereas B represents the number of bins and $\overline{H}_P$ the average of the frequencies of the Bins of histogram H_p. In [132] different methods for the aggregation over all cells of the scalar SV_n of the single cell n are examined. The most effective aggregation for P_{var} was found to be the aggregation by calculating the maximum, referred to as A_{max}:

$$A_{max} = \max_{\forall n \in 1 \dots N} SV_n. \tag{16.4}$$

For the sake of simplicity, in the following this approach will be referred to as PRNU-1.

The second approach analyses the spectral characteristics of the PRNU signal. First the DFT transformed of the PRNU is calculated, subsequently, the energy of the DFT magnitudes per cell are estimated:

$$D_{en} = \sum_{x \in M} |x|^2, \tag{16.5}$$

where M represents the DFT magnitudes and x the respective values. The aggregation of the scalars of the cells is carried out by calculating the minimum over all cells:

$$A_{max} = \min_{\forall n \in 1 \dots N} SV_n. \tag{16.6}$$

For the sake of simplicity, in the following this approach will be referred to as PRNU-2.

Since the extracted characteristics are scalars, they can directly be used for classification, avoiding the need to train a classifier.

16.5.2 *SPN*

A further method for the analysis of the image noise pattern was proposed in [171] in parallel to the algorithm used in Section 16.5.1. As with the calculation of PRNU, the high-frequency information of the facial region, scaled to 160×160, is extracted according to equation 16.2, however, the guided image filtering suggested in [57] is used as filter function $F(I)$. Further, an adaptive Wiener filtering is performed on the residuals, providing the SPN. Under the assumption that differences in the frequency response of the SPN for bona fide and morphed images are detectable, the 2D Fourier transform of the SPN signal is computed by means of DFT. The Fourier transform is divided radially into 18 areas and each area is further divided into 7 axial sub-areas, creating 144 individual regions. Due to the symmetry of the Fourier spectrum, however, the upper and lower halves are identical, thus the lower half can be discarded. For each remaining region, the mean and variance of the contained values is computed, resulting in a feature vector with length of 144.

16.6 DEEP FEATURES

As described in Section 6.5, machine learning algorithms, especially DNNs, can be used to extract statistically significant features from images in addition to hand-crafted feature extractors. The difficulty of this approach is the dependence of the information represented in the extracted features on the nature of the training data used to train the feature extractor. If the wrong training data is chosen, this might cause an over-fitting of the feature extractor, resulting in very good results on known data, which, however, cannot be reproduced in a real use case. In order to avoid this effect, only DNNs pre-trained for face recognition are applied in this thesis. These networks have been trained to extract representative features from facial images, without containing morphed facial images in the training process, thus implicitly preventing an over-fitting to artefacts of a specific morphing algorithm. In the implemented MAD pipeline the feature extractors of three different FRSs are used, which are described in more detail in the following sections.

16.6.1 *FaceNet*

The first FRS utilised for the extraction of deep features is a reimplementation[5] of FaceNet, a deep CNN FRS proposed in [140]. FaceNet is built following the topology of *Inception-Resnet-v1* (also known as *GoogLeNet*) proposed in [156]. The distinctive element of this topology is the use of so-called inception modules. The determination of the optimal kernel size for the convolution step in CNNs is not trivial, due to the non-constant size of the objects to be described. For this reason, several kernel sizes are used in parallel in the inception modules, promising an increased robustness of the algorithm. In the case of the *Inception-Resnet-v1*, 9 inception modules are concatenated, resulting in a network with a depth of 27 layers.

Different pre-trained models are available for the employed implementation, in this work, the most recent model was chosen. The

[5] The utilized implementation can be found at:
https://github.com/davidsandberg/facenet.

model was trained on the VGGFace2 database [111], normalised by *MTCNN* [170] and resized to 160×160 pixels. Thus, all images whose features are to be extracted are also preprocessed as full face images in the same manner and subsequently passed to FaceNet.

The network extracts feature vectors of length 512.

16.6.2 *ArcFace*

As described for example in [111], an optimization of deep CNNs as FRSs is often carried out based on the soft-max loss, which can be represented in a simplified way according to [90] as follows:

$$L = -\frac{1}{N}\sum_{i=1}^{N} \log \frac{e^{W_{y_i}^T x_i}}{\sum_{y=1}^{n} e^{W_j^T x_i}}, \tag{16.7}$$

whereas x_i represents the feature of the i-th sample, belonging to class y_i. W_j represents the j-th column of weights W, N is the number of samples and n is the number of classes (in the described application $n = 2$). However, according to [29], the soft-max loss has the disadvantage of not explicitly optimizing the features by increasing the similarity of the features for intra-class comparisons and decreasing it for inter-class comparisons. Thus, trained CNNs do not provide optimal robustness against high intra-class appearance variations, e.g. due to different poses or different age at the time of recording. For that reason, Deng et al. [29] suggest using the ArcFace loss instead. The main difference to the soft-max loss given in equation 16.7 is that $W_j^T x_i$ is replaced by $\|W_j\|\|x_i\|\cos\theta_j$, where θ_j is the angle between weight W_j and feature x_j. In order to achieve a higher inter-class discrepancy and intra-class compactness of the features, a margin penalty m is introduced in addition, resulting in the following ArcFace loss:

$$L = -\frac{1}{N}\sum_{i=1}^{N} \log \frac{e^{s(\cos(\theta_{y_i}+m))}}{e^{s(\cos(\theta_{y_i}+m))} + \sum_{j=1, j\neq y_i}^{n} e^{s(\cos\theta_j)}}, \tag{16.8}$$

where s represents a re-scaling factor for $\|x\|$. A detailed deduction of the ArcFace loss, as well as the mathematical background and reasoning behind the design can be found in [29].

In the MAD pipeline the existing implementation[6] of the authors of [29] is utilized. As with the FaceNet, described in Section 16.6.1, the images are normalised using MultiTask Cascaded convolutional Neural Network (MTCNN) and scaled to 112×112 pixels, prior to training or feature extraction. The authors offer several pre-trained models, in this pipeline the model *LResNet50E-IR,ArcFace@ms1m-refine-v1* is chosen, since, according to the authors, it achieves the most stable performance on the tested databases. The architecture of the selected

[6] The corresponding source code can be found at: https://github.com/deepinsight/insightface.

network is, as the name suggests, a residual network comprised of 50 layers. A residual network is characterised by shortcut connections between different layers, allowing the output of a previous layer (residuals) to be processed as input on subsequent layers, simplifying the computationally expensive training of very deep CNNs.

The network extracts feature vectors of length 512.

16.6.3 *Eyedea*

The third type of Deep Features is provided by the commercial FRS *EyeDentity*[7] of the company *eyedea*. Unlike for the other FRSs utilised, the implementation is a company secret and thus not openly available, however, in contrast to most other commercial products, the software allows separate extraction of feature vectors, allowing their use for MAD. According eyedea the feature extraction is done by an DNN. However, little information is available about the extraction process. Full facial images in gray scale or colour are accepted as input, the extracted feature vector has a length of 256.

[7] Further information regarding the software and the company can be found at: https://www.eyedea.cz/eyedentity/

17 FEATURE PREPARATION

The feature vectors extracted by the algorithms described in Chapter 16 could, in theory, directly be used as an input for classification by machine learning algorithms. However, there are a couple of reasons for a prior preparation of the feature data. First, for most machine learning algorithms, it is important that the training data is zero-centred and has a variance of 1 (as a consequence, all other data has to be normalized according to the same scheme in order to enable classification). Furthermore, a feature level fusion can be performed at this point, enabling, for example, in the differential MAD scenario, the features of the suspected morph and the TLC to be merged into a single feature vector. In the MAD pipeline applied in this work, both, single image algorithms and differential algorithms, are implemented, the construction of the respective feature vectors is described in the following sections.

17.1 SINGLE IMAGE FEATURES

In order to implement an algorithm according to the scheme described in Section 11.1 and illustrated in Figure 11.1a, no further action is required at this point. In principle, all features not requiring a comparison with an additional feature can be used. This includes all multidimensional features extracted by the algorithms described in Section 16.4. However, in some scenarios the features do not contain information relevant for MAD without the information of the TLC. For example, landmark features are not considered in the single image scenario, as only by the comparison of the landmarks of the suspected morph with those of a TLC a statement an be made about whether or not the considered image has been morphed.

17.2 DIFFERENTIAL FEATURES

In the differential scenario a TLC is available in addition to the suspected morph. As described in Chapter 12, the TLC can be either utilized for attempted morph reversal, so called de-morphing, or for a comparison of the features of the suspected morph and the TLC. Due to the fundamental differences of de-morphing, it is regarded outside the MAD pipeline and is solely used for comparisons in the evaluation as a black box. There are unlimited possibilities for the combination of features for comparative algorithms, the characteristic of the resulting, fused feature vector and the contained information, varies accordingly.

An obvious combination is the concatenation of the feature vectors. In this case, no information is discarded, meaning all information from both feature vectors is available in a subsequent training. However, this procedure doubles the number of dimensions in the feature vector. The increasing length of the feature vectors has a high impact on the training of the classifiers, as a higher dimension of the input data leads to a larger number of free variables in the classifier (e.g. more dimensions of the hyperplane of an SVM). From a mathematical point of view, the training of a machine learning algorithm is the solving of an equation. If the number of free variables (dimensions in feature space) doubles, twice the number of data points (training samples) is needed in order to solve the equation. Another possibility is to capture the differences between the two feature vectors. This causes the loss of the information about the absolute of the individual values in the feature vector, which is considered to be acceptable, as the relevant information is assumed to lie in the differences. An advantage of this representation is that the dimensions of the feature vector remains the same as for the single image scenario. For this reason, the calculation of the difference between the feature vectors of the potential morphs and the TLCs for the creation of the merged features was chosen in this MAD pipeline.

17.3 FEATURE NORMALISATION

Regardless of whether single image features or differential features are used, the machine learning algorithms to be trained benefit from a normalisation of the features [147]. The goal of the normalisation is to zero-centre the data and to set the variance to 1. The normalisation hereby is applied to the dimension of the feature vectors. I.e. every feature of the feature vector is normalised across all data points. If the data is not zero-centred, it might lead to the effect that during the first iterations of the machine learning algorithms only the offset of the data is compensated. For example, with a linear SVM the hyperplane would initially be shifted in one direction only. This might cause a considerable extension of the training process. If the variance is much greater than 1, the errors occurring during training are significantly larger, since the absolute distances between the data points are bigger, changing the ratio of the errors to the fixed variables of the loss function, which in turn changes the training behaviour of the algorithms.

Different algorithms can be used for normalisation. An intuitive approach is the min-max normalisation. In this approach, the value range of each feature across all data points is first transformed by subtracting the minimum, such that the value range starts at 0, afterwards the value range is mapped to 1 by dividing with the value range. The corresponding equation is given as:

$$y_i = \frac{x_i - \min(x_i)}{\max(x_i) - \min(x_i)}, \tag{17.1}$$

whereas x_i represents the vector of the i-th feature over all data points and y_i the normalised values of the i-th feature of all data points. The min-max normalisation bears the disadvantage that outliers in the value range can have a huge influence on the result of the normalisation. To reduce the influence of outliers, the Z-score normalisation can be applied. Instead of $\min()$ and $\max() - \min()$, the mean value and the standard deviation are used, as they react more robustly to outliers:

$$y_i = \frac{x_i - \mathrm{mean}(x_i)}{\mathrm{std}(x_i)}. \tag{17.2}$$

Due to the more robust behaviour, the Z-score normalisation is used in this MAD pipeline. It is important to note that the exact same mean and standard deviation of the data used for training have to be used to normalise the data to be evaluated as well, making the normalisation function identical.

18 TRAINING OF CLASSIFIERS

In order to be able to make a statement about whether a sample has been morphed or not, based on the feature vectors extracted as described in Chapter 16 and prepared as described in Chapter 17, a classification of the feature vectors is necessary, reducing the high-dimensional space of the feature vectors to a one-dimensional score, which can be used to derive a binary decision based on a threshold value.

As described in Chapter 5, the result of training the machine learning algorithms is hard to comprehend, thus a possible over-fitting can only be proven by appropriate testing. For this reason, it is recommended to minimize the risk of errors by adhering to training principles, which are described in Section 18.1.

Various classifiers are implemented in the MAD pipeline used in this work. Section 18.2 describes the unification of the usage of the different classifiers, in order to keep them interchangeable and to allow a good comparability of obtained results.

As described in Chapter 5, the choice of the hyperparameters of a classifier can significantly influence the result of the training process. Hence, Section 18.3 discusses the finding of optimal hyperparameters, describes the problems associated to this and explains the choice of the hyperparameters implemented in this pipeline. Finally, Section 18.4 gives an overview of the classifiers available in this pipeline.

18.1 TRAINING PRINCIPLES

For the training and evaluation of machine learning algorithms for the classification of suspected morphs, the general principles for the training of machine learning algorithms as well as for the training and evaluation of biometric systems have to be considered. Furthermore, specific effects resulting from the self-generated morphing databases have to be considered.

In general, it should be ensured that any feature vector used in the context of the algorithm (regardless of whether it is used for training, evaluation or in a real-world deployment scenario) exhibits the same dimensions. This requirement refers not only to the length of the feature vectors (it is mathematically and logically impossible to process vectors of different lengths in the presented machine learning algorithms), but also to the nature of the dimensions. If the feature vectors have been normalized for training, the same must be applied to all other feature vectors used with this algorithm.

A further prerequisite is a clear separation of the data sets [98]. The data used for training must not be present in the evaluation. In the case of morphed facial images, it should be noted that the images used for the creation of the morph are directly related to the morph itself, consequently the training and test sets should be separated based on the subjects, prior to the creation of the morphs.

In addition to the clear separation of subjects, it should be ensured that, as described in Chapter 5.5, the number of samples per class is balanced. In the case of MAD, it is much easier to increase the number of morphed samples than the number of bona fide samples, since, in theory, all subjects in a subset can be morphed among each other, resulting in an exponentially increasing number of morphed samples in relation to the number of subjects. Thus, a pre-selection of the morphs to be created has to be made, in order to create a balanced dataset.

Furthermore, there are no public morphing databases that can be used to train the algorithms. Thus, the morphs in the databases used in most of the publications are created by the authors themselves, usually using only one morphing algorithm. As a result, the training and test data comprise morphs that contain the same artefacts and inconsistencies. Over-fitting of machine learning algorithms to artefacts specific to a morphing algorithm can only be detected by evaluating morphs created by an algorithm with different characteristics. Otherwise, no statement can be made about the generalizability of the MAD algorithm.

18.2 TRAINING FRAMEWORK

In order to enable a fair benchmark of the applied classifiers, the machine learning algorithms used as classifiers in the MAD pipeline are unified. Initially, the feature vectors, previously extracted according to Chapter 16 and prepared according to Chapter 17, are loaded. At classification stage it is irrelevant whether the features are for the differential or single image use-case. The parameters of the normalisation applied in Section 17.3 have to be stored, as it is mandatory to normalise all further data with identical parameters. The preparation is performed independently of the machine learning algorithm to be trained.

The model to be trained, depending on the selected algorithm, is initialised with fixed parameters. The choice of parameters is described in more detail in Section 18.3. Due to the choice of fixed parameters, the different machine learning algorithms can be considered as black boxes, receiving the feature vectors as input and generating a predictive model. The generated models are stored together with the parameters for normalisation for the evaluation of further data.

18.3 PARAMETERS FOR CLASSIFIERS

The choice of hyperparameters of machine learning algorithms has an influence on the training and subsequent structure of the generated model. The selectable parameters of the individual machine learning algorithms and their influence on the model are described in Chapter 5. The determination of the optimal hyper parameters is a complex issue [19]. On the one hand, the search spaces for the optimal hyperparameter set are often very large, on the other hand, the optimal hyperparameters also depend on the nature of the training data, meaning the optimal hyperparameter set has to be determined for each dataset individually.

Different approaches exist to detect the optimal hyperparameters. The easiest approach is a grid search. The optimal value is searched in a defined range for each definable hyperparameter. Since the hyperparameters might influence each other, the optimal parameter set must be determined throughout all dimensions, leading to an explosion of possibilities, especially for algorithms with multiple hyperparameters. One model has to be trained and evaluated for each parameter set, meaning that this method is only feasible for a small number of parameters or a small range of possible values per parameter. More elaborate is the search for hyperparameters via optimisation algorithms, so-called auto-tuning, for example by Bayesian optimisation [149]. The model itself is considered as a function to be optimised, which allows a stepwise tuning of the parameters in certain directions, reducing the number of models to be trained.

For the algorithms trained in this thesis, auto-tuning of the hyperparameters was omitted. Due to the limited size of the databases and the limited computing resources, the optimisation of the hyperparameters did not prove useful. To achieve robust algorithms the recommended standard parameters are applied. It should be noted that an optimisation of the hyperparameters might bring a further performance gain, but this has to be analysed in more detail on larger databases for a smaller number of algorithms, in order to avoid an over-fitting.

18.4 CHOSEN CLASSIFIERS

In the MAD pipeline used, four different machine learning algorithms are implemented, which are listed including the defined hyperparameters in Table 18.1. The freely available implementations of SciKit-Learn[1] have been used for the respective algorithms.

SVM The first classifier used is an SVM with RBF kernel. SVMs are a widely used and often applied classifier in biometrics, for example

[1] The documentation for the used implementations can be found at: https://scikit-learn.org/

ALGORITHM	HYPERPARAMETER
SVM with RBF kernel	$C = 1.0, \gamma = \frac{1}{n_{features}}$
Random Forest	$n_{estimators} = 100$
AdaBoost	$n_{estimators} = 50, learningrate = 1$
Gradient Boosting	$n_{estimators} = 100, learningrate = 0.1$

Table 18.1: Machine learning algorithms and respective parameter sets implemented in the MAD pipeline

for FRSs [116] or Presentation Attack Detection (PAD) for FRSs [121]. As described in Section 5.1, the RBF kernel provides the possibility to robustly separate data distributions in arbitrary constellations. In the used implementation two hyperparameters have to be set, C and γ. C is the regulatory parameter for the training of the algorithm. A lower C reduces the *flexibility* of the training process. The default parameter is $C = 1$. The parameter γ corresponds to $\frac{1}{2\sigma^2}$, as described in Section 5.1.2, defining the radius of influence for a single data point. By default γ is set based on the number of features ($n_{features}$) and the variance of the training data ($var(X)$):

$$\gamma = \frac{1}{2\sigma^2} = \frac{1}{n_{features} \cdot var(X)} \tag{18.1}$$

Due to the fact that the training data in this MAD pipeline is normalised according to Section 17.3, setting the standard deviation to 1, a value of 1 can be assumed for the variance as well, resulting in a γ in a reverse relationship to the size of the feature vector.

RANDOM FOREST In [37] different classifiers are evaluated on different data. It is shown, that in addition to SVM with RBF kernel, decision tree based ensemble classifiers are very universally applicable. Thus, 3 different decision tree based ensemble classifiers are implemented in the MAD pipeline. Based on the results from [37], a random forest classifier is used, a classifier which has already proven to be suitable for other fields of biometrics, for example fingerprint quality assessment [107, 108]. The used random forest classifier is implemented according to the concept described in Section 5.3.1. The only hyperparameter affecting the ensemble itself is the number of decision trees to train ($n_{estimators}$), which is set to 100 by default. All other parameters determined the character of the individual trees. The trees themselves are almost not restricted, no maximum depth is defined and no specifications are made about the number of samples per node. This configuration would result in a high probability of over-fitting for a single decision tree, but when used in an ensemble, the possible over-fitting of a single tree is balanced by the repeated training on different subsets.

ADABOOST A further ensemble classifier employed is AbaBoost with decision stumps. As described in Section 5.3.2, decision stumps can be interpreted as binary decision trees of depth one, so that the AdaBoost used can be understood as an ensemble classifier based on decision trees. The hyperparameters concerning the ensemble are the number of weak learner ($n_{estimators}$) to be trained, which is set to 50 by default, and the $learningrate$, which will control the influence of each weak learner in the final algorithm. A lower learning rate reduces over-fitting, but requires more weak learners to achieve the same algorithm performance. The parameter is set to 1 by default, resulting in full influence of each weak learner. Since the weak learner is a binary decision tree of depth one, no further parameters need to be defined.

GRADIENT BOOSTING The last machine learning algorithm employed is gradient boosting with decision trees. An implementation according to Gradient Tree Boosting, as described in Section 5.3.3, is used. For this implementation, significantly more ensemble-related hyperparameters are definable, compared to the ensemble classifiers described above. In addition to the number of decision trees ($n_{estimators}$) and $learningrate$, which, by default, are set to 100 and 0.1, respectively, the number of subsamples, which, by default, is set to 1, can be specified, meaning that no subsampling is performed. The decision trees themselves are mainly not restricted, only the maximum depth is limited to 3 .

19 SUMMARY

Part IV describes the design decisions and the resulting structure of the MAD pipeline used in this thesis. In order to ensure the greatest possible flexibility and to reduce the implementation effort, the pipeline is designed as a modular system. The modules are *Data Preparation*, *Feature Extraction*, *Feature Preparation*, and *Training of Classifiers*.

Depending on the scenario, only passport photos or passport photos and TLCs are available to the pipeline. The variance of the passport photographs is limited by the standards for passport photographs defined by ICAO, see [63]. However, the variance of TLCs can be much higher as they are captured in a semi supervised environment. In order to ensure a robust feature extraction, all images are prepared in advance in a uniform way. For this purpose, the images are rotated, ensuring that the eye centers are on a horizontal line. The rotated image is subsequently cropped according to the proportions defined in [63]. Some feature extractors are dependent on obtaining a fixed facial section of the same size in order to return proper results. For these feature extractors, a fixed region is cut out of the previously normalised facial image.

After the normalisation of the images, the features can be extracted. For this purpose, different types of feature extractors are implemented in the pipeline: texture descriptors, gradient based descriptors, keypoint descriptors, landmark extractors, image noise pattern, and deep features. The exact implementation, selected parameters, and a motivation for the use of the individual feature extractors is described in the respective sections of Chapter 15.

Depending on whether the images are to be evaluated in a single image scenario, or whether a TLC is available for differential evaluation, the data has to be prepared differently for the use in training a machine learning algorithm. In the differential scenario, the information of the feature vectors of the suspected morph and the TLC need to be merged. In this pipeline the difference between the two feature vectors is calculated. Most machine learning algorithms require zero-centered data with a variance of 1 for proper training. In order to ensure this, all feature vectors, whether single image or differential, are scaled using z-score normalisation.

In the final step, the classifiers are trained using the previously prepared data. Due to the uniform implementation of the pipeline, the training of the individual classifiers hardly differs. In order to keep the subsequent evaluation clear and comparable, the optimisation of the hyper parameters was deliberately omitted. The classifiers implemented in the pipeline are listed in Section 18.4 including the selected parameters and a motivation for the use of the respective classifier.

Part V

EXPERIMENTAL DATA

20 FACE IMAGE DATABASE SELECTION

Despite numerous publications concerning MAD, no viable public databases for the training of MAD algorithms are available. The main reason for this are legal restrictions. Especially in Europe, the acquisition of biometric databases is accompanied by a high legal requirement of data protection, resulting in the fact that acquired databases cannot be distributed, as the acquired subject has the right to withdraw his or her consent to contribute to the database at any time, in accordance with General Data Protection Regulation (GDPR) Article 7 - "Conditions for consent". The issue with the use of publicly available databases is that the respective license agreements usually prohibit further processing and redistribution.

With the NIST FRVT MORPH, described in Section 3.1.3, and SOTAMD, described in Section 3.1.1, projects are initiated to create databases as a basis for a uniform evaluation, however, these databases will only be used for independent third party reconfirmation of the performance of MAD algorithms and are not accessible for the creation of algorithms, thus they are not considered in this Chapter. For training and for in-house testing the need for dedicated morphing databases is given.

Thus, a dedicated MAD database was constructed for this thesis. The decision criteria for the selection of the used face databases are described in Section 20.1, the available face databases are listed and evaluated in Section 20.2.

20.1 PREREQUISITES FOR REALISTIC DATABASES

In order to create a database that realistically reflects the nature of expected passport photographs (morph and bona fide), as well as TLC images, for example from the border gates, the underlying face database has to meet certain criteria.

No standardised specifications regarding the nature of the image are available for the TLC sample. Since the capturing process is semi-supervised and in a less controlled environment, the TLC images may exhibit a high degree of variance. A prerequisite for the creation of a realistic database is thus the presence of facial images reflecting this variance.

In order to create morphs, two facial images are needed that meet the requirements of a passport photo, which are defined by ICAO 9303 [67], which refers to ISO/IEC 19794-5 [63]. The properties of images contained in passports in actual use may deviate from the defined standard, as the standard might not be correctly applied during capturing and transfer of the photograph to the passport. However, the database to be created is assumed to comply with the standards as far as possible. The most important properties of a passport photograph defined in the standard are defined below.

20.1.1 *Pose*

The first set of rules concerns the subject represented in the image. First of all, the pose of the subject in the image is specified. It should be as neutral as possible, meaning that deviations from the frontal pose should be kept to a minimum. For the morphing process, an identical pose of both contributing subjects is required. Slight variances, which are covered by the standard, can be normalised by the process described in Section 15.1, which, however, may lead to a shoulder pose which is not permitted according to the standard.

Furthermore, the standard requires a neutral facial expression. This requirement cannot be met by subsequent corrections, thus it must be ensured that, for example, no closed eyes or exposed teeth are present in the data used as passport photograph.

Furthermore, the position of the face in the picture is of importance. The standard defines a frame in which the face has to be located. However, this requirement can be met by subsequent normalisation, given that there is a sufficient margin and the resolution of the facial area is sufficient.

20.1.2 *Artefacts*

In general, all types of face-covering artefacts, such as scarves or headgear, should be avoided in the image. However, some artefacts are part of the appearance of the subject. For example, a subject wearing glasses will also wear them when crossing the border, meaning that they may also be present in the passport photo. In this case, the glasses have to be transparent, the frame must not cover the iris and there must be no reflections in the glasses. In the case of headgear that cannot be removed for the picture, e.g. due to religious reasons, it must be placed such, that the face is not covered.

20.1.3 *Image Quality*

An important criterion for the suitability of a picture as a passport photo is the quality, which is a very broad term and thus summarises many picture characteristics. An essential factor is the resolution of the image. In [63] it is specified, that the width of the face should consist of at least 180 pixels, which, according to the parameters described in Section 15.1, corresponds to an overall resolution of at least $360 \times$

480 pixels. If the images are not of sufficient size, upscaling is not recommended, as the missing information cannot be reconstructed.

Other factors, such as the contrast or focus of the image, are also described in the standard, however no objective metrics are defined, leaving some leeway for implementation. The methods available for a quantitative quality assessment can, according to [75], be divided into three categories:

- Full-reference methods
- Reduced-reference methods
- No-reference methods

The full-reference and reduced-reference methods are easier to implement and designed to describe quality degradation [113]. For the full-reference method an image in original quality is available in addition to the image to be examined, enabling a direct comparison of both images. For the reduced-reference methods, only features extracted from the original image are available for comparison. However, the quality evaluation of passport photographs is not relative, but an absolute value is required, meaning that the reference-based methods are not applicable. The no-reference methods are more complicated to implement than the reference-based methods, since the direct comparison of values is not available. An algorithm designed for referenceless quality evaluation of images is, for example, Blind/Referenceless Image Spatial Quality Evaluator (BRISQUE) [96], extracting statistical features of natural images, and subsequently evaluating them with an SVM, in order to obtain a measure of the quality of the image. In the quality evaluation, factors as compression, noise and sharpness are evaluated.

Despite existing methods for the quantitative quality determination of images, the actual quality of accepted passport photographs may vary. Consequently, the passport photographs contained in a realistic database cannot be selected according to quantitative quality characteristics, but should exhibit a certain variance in quality.

20.1.4 *Passport and TLC Images*

A morphing database consists of three different types of images. The suspected or passport images can be both bona fide and morphed, the TLCs or eGate images are always bona fide. Therefore, besides the two quality classes described above, the number of images of each class is of importance. In an optimal case, at least two passport quality images per subject are available (one for morphing, the other as a bona fide reference), as well as at least one TLC image in a quality expectable from an eGate. A higher number of reference images increases the number of comparisons in the differential scenario. For the single

name	provider	resolution	passport quality	TLC quality	subjects
ARFace[a]	Ohio State University	768×576	No	Yes	126
AT&T Database of Faces[b]	AT&T	92×112 pixels	No	Yes	50
BioID Face[c]	BioID	384×286 pixels	No	Yes	23
CelebA[d]	University of Hong Kong	178×218 pixels	No	Yes	10.177
Color FERET[e]	NIST	512×768 pixels	Yes	Yes	856
Face in Action[f]	Carnegie Mellon University	640×480 pixels	No	Yes	180
FRGCv2[g]	NIST	average 250 pixels inter-eye distance	Yes	Yes	570
Labelled Faces in the Wild[h]	University of Massachusetts	150×150 pixels	No	Yes	1680 with two or more images
Multi-PIE[i]	Carnegie Mellon University	$\sim 2180 \times 2884$ pixels	Yes	No	337
Yale Face[j]	Yale University	320×243 pixels	No	Yes	15

Table 20.1: Available face databases

[a] http://www2.ece.ohio-state.edu/~aleix/ARdatabase.html
[b] No direct link available
[c] https://www.bioid.com/facedb/
[d] http://mmlab.ie.cuhk.edu.hk/projects/CelebA.html
[e] https://www.nist.gov/itl/products-and-services/color-feret-database
[f] No direct link available
[g] https://www.nist.gov/programs-projects/face-recognition-grand-challenge-frgc
[h] http://vis-www.cs.umass.edu/lfw/
[i] http://www.cs.cmu.edu/afs/cs/project/PIE/MultiPie/Multi-Pie/Home.html
[j] http://cvc.cs.yale.edu/cvc/projects/yalefaces/yalefaces.html

image scenario, the number of comparisons can only be extended by increasing the number of available passport quality images. If the database lacks passport quality images, images used for morphing can be additionally included as bona fide images. Whilst this violates the independence of the individual comparisons among each other, as demanded in [98], no influence on the expected result of an evaluation is known, due to the fact that the evaluations are performed independently of each other.

20.2 Existing face image databases

A broad range of face databases is available for research, however, only a few of them meet the requirements to be used for the creation of a morphing database. The databases considered in this thesis and their properties are listed in Table 20.1. The databases can be divided into three classes: Web-scraped databases, recorded databases in low quality and recorded databases in high quality.

The *CelebA* and *Labelled Faces in the Wild* are databases of the first category. Both contain a large number of subjects, but without quality standards for the images. The resolution does not meet the minimum requirements for passport photos. Thus, databases of this category can be excluded.

The second category mostly consists of older databases, which, due to their age, do not meet the required quality standards. The *AT&T Database of Faces* was recorded in the AT&T Labs between 1992 and 1994. The images contain a variance in pose and gestures, but are only available as low-resolution greyscale images and are therefore, like the images of the *BioID Face* or *Yale Face DB*, not suitable for use as passport photos. The *Face in Action* is much more up-to-date with pictures taken between 2004 and 2005, but the recording scenario was deliberately chosen to correspond to the pictures taken at the border crossing. Thus only images satisfying the requirements for TLC images are available.

Face databases suitable for the creation of morphing databases can be found in the third cathegory. The *ARFace* offers color images of 136 subjects in neutral and non-neutral poses, with different lighting and occlusions of the face (for example by scarf and cap) in a sufficient resolution. Unfortunately, the images are blurry, making them unsuitable as bona fide passport images or as a basis for the creation of morphs. The *Multi-PIE* offers very high resolution images of 337 subjects, however, the images were recorded in a highly constrained scenario, leaving not enough variations for TLC images. The *FERET* was taken at NIST between 1993 and 1996 with an analogue camera and digitalised in 2003. Due to the high quality scans, enough images satisfying the requirements for passport photos are available. It has to be noted, the scanned analogue photos exhibit a different camera noise properties as digital captured images. The database partly contains grey scale images, which are not considered as passport photos. Since the subjects were taken with a non-neutral facial expression in addition to the neutral facial expression, sufficient images are available which can be used as TLC images. The *FRGCv2* was recorded by NIST at a more recent date. In addition to portraits, which are suitable as passport photos, the subjects were captured in different locations with different lightings and backgrounds, resulting in a variance, allowing them to be used as TLC images.

Due to the described properties, the morphing database of this thesis is built on the basis of *color FERET* (referred to as FERET) and *FRGCv2* (referred to as FRGC).

21 MORPH DATABASE CREATION

This chapter describes the creation of the morphing database used in this thesis, based on the face databases selected in Section 20.2. Not all images contained in the face databases are suitable for the creation of the morphing database, therefore the images are pre-selected according to the scheme described in Section 21.1. The image pairs to be morphed are determined according to the scheme described in Section 21.2. Finally, the images are post-processed as described in Section 21.3.

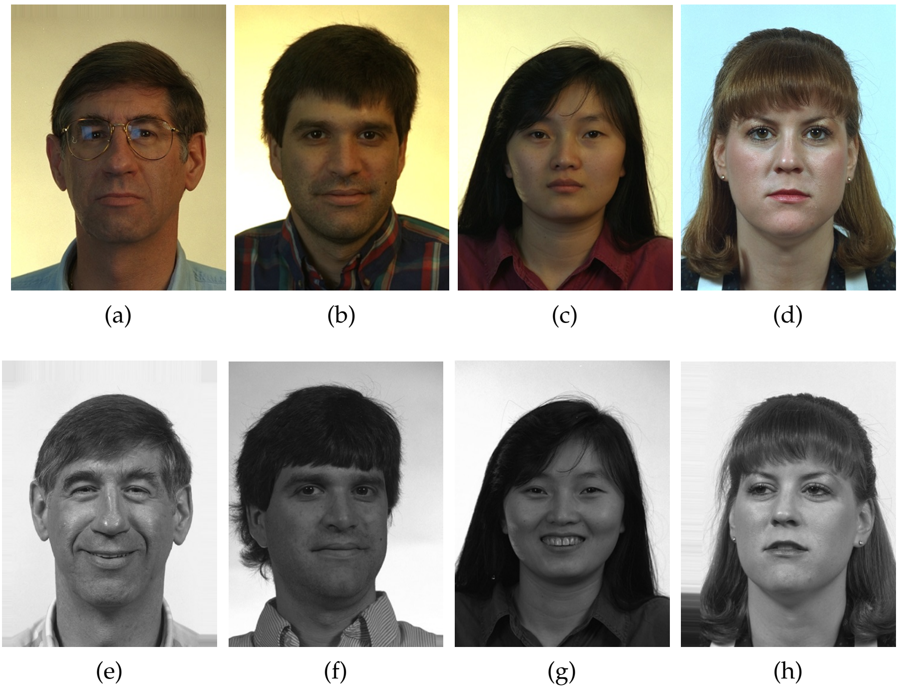
21.1 IMAGE PRE-SELECTION

The creation of the morphing database requires 3 categories of images: Bona fide reference images, morph input images and TLC images. The Bona fide reference images correspond to an unaltered passport image and should meet the corresponding quality criteria described in Section 20.1. The morph input images are used in pairs for the morphing process. These should be of passport image quality as well. For the selection of the images in passport image quality, the guidelines standardised in ISO/IEC 19794-5 [63] are followed. Consequently, only images with a closed or minimally opened mouth and a neutral facial expression or a slight smile are included. Images with reflecting glasses are discarded. The class of TLC images corresponds to live recordings, for example at the eGate. Therefore, the images should not be of a controlled, high quality, as this cannot be expected from semi-supervised capturing. For this class, all images not classified as suitable for passport photos in the above pre-selection can be considered. Thus, the images contain unsharpness, uneven lighting, non-neutral facial expressions, pose variations, etc. The partitioning of the images into the classes *passport image quality* and *TLC quality* was carried out manually. The result of the division of the databases is listed in Table 21.1. Besides the number of subjects, the biggest difference between the databases is the variance of the images in the category

DATABASE	SUBJECTS	QUALITY	SAMPLES	SUBJECTS PER # SAMPLES				
				1	2	3	4	>5
FERET	529	Passport image	761	443	78	54	-	-
		TLC Image	1389	267	88	64	42	68
FRGC	533	Passport Image	984	24	58	20	27	404
		TLC Image	2967	121	86	68	61	197

Table 21.1: Categories of images in both face databases

TLC image quality. In the FERET database, mainly different facial expressions and slight rotations in the pose are included, examples are given in Figure 21.1. In the FRGC database the variances are more significant. In addition to different facial expressions, different backgrounds, illuminations and focuses of the images can be observed, examples are shown in Figure 21.2.


(a) (b) (c) (d)

(e) (f) (g) (h)

Figure 21.1: Examples of reference and grey scale TLC images for FERET

Based on the two pre-sorted classes, the images are divided into three categories (bona fide reference images, morph input images and TLC images). In order to create realistic scenarios, the time of capture between the passport images and the probe images is maximized as far as possible on the basis of the databases. Due to the large differences in the number of images per subject between the databases, different protocols are used for both databases.

For FERET the images are selected per subject according to the following scheme:

1. The chronologically last image of the class *TLC image quality* is used as TLC image.

2. If more than one recording session is available, all images of the *passport image quality* class from the same recording session as the TLC image selected in step 1 are discarded.

3. The chronologically first sample of the class *passport image quality* is used as bona fide sample.

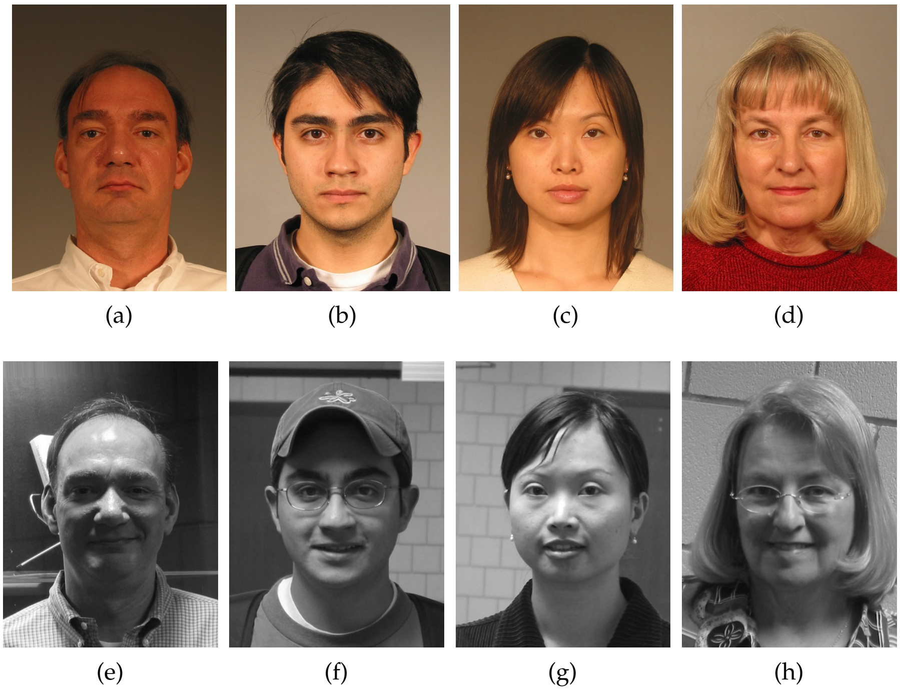


Figure 21.2: Examples of reference and grey scale TLC images for FRGC

4. The chronologically last sample of the class *passport image quality* is used as morph input image.[1] If, following step 2, only one sample remains in class *passport image quality*, the sample used for bona fide and morph input is identical.

5. If there is more than one sample in class *TLC image quality*, a second sample image is selected, preferably with a different pose than the first sample image and with a maximum time difference to the selected images of class *passport image quality*.

For FRGC the images are selected per subject according to the following scheme:

1. Up to 4 samples are selected from class *passport quality* in chronologically equal intervals. Even ones serve as bona fide, odd ones as morph input images. If only one image is available, it will be used as both, bona fide and morph input image.

2. From the class *TLC image quality* up to 5 chronologically equidistant samples are selected as probe images.

The composition of the resulting database is described in Table 21.2.

[1] This method has the side effect that the samples used for morphing are closer in time to the sample than the bona fide. However, since there is rarely more than one recording session, this effect is negligible.

DATABASE	SUBJECTS	MALE	FEMALE	BONA FIDE	MORPH INPUT	TLC
FERET	530	330	200	530	530	791
FRGC	533	231	302	984	964	1726

Table 21.2: Composition of the database resulting from the image preselection

21.2 IMAGE MORPHING

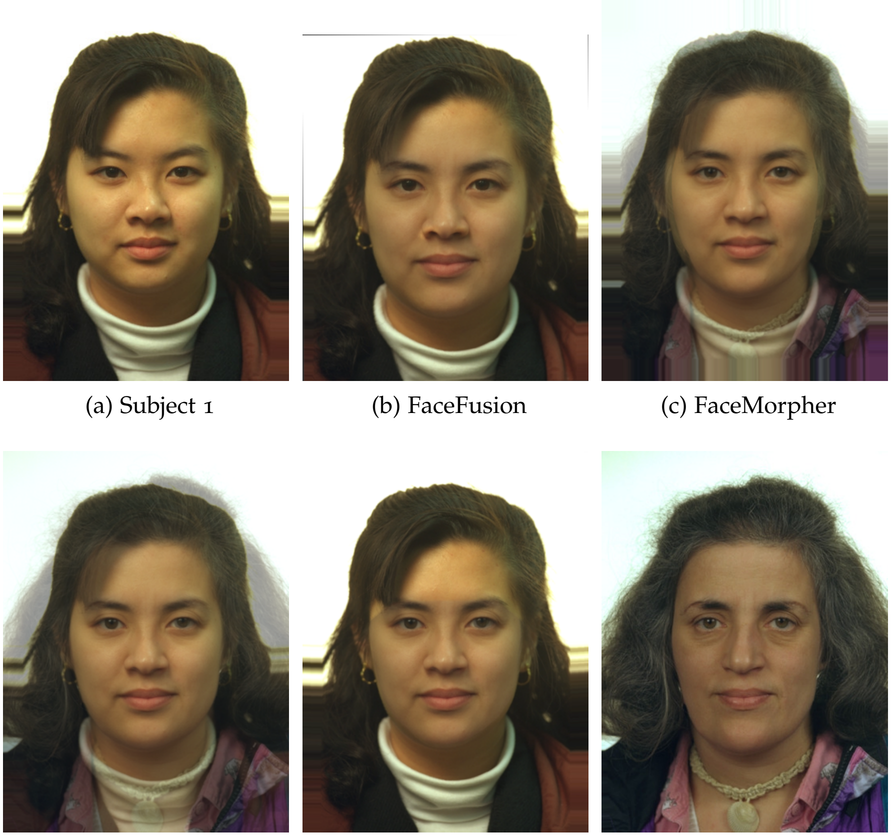
In order to enable the morphing database to be used for evaluating the generalisability of MAD algorithms towards differing morphing algorithms, four different morphing algorithms are applied to construct the database, hereafter referred to as *FaceFusion*[2], *FaceMorpher*[3], *OpenCV* and *UBO Morpher*.

FACEFUSION FaceFusion is a proprietary morphing algorithm. Originally being an iOS app, an adaptation for Windows which uses the 68 landmarks of Dlib and Delaunay triangles was applied. After the morphing process, certain regions (eyes, nostrils, hair) of the first face image are blended over the morph to hide artefacts. Optionally, the corresponding landmarks of upper and lower lips can be reduced as described in [97] to avoid artefacts at closed mouths. The created morphs have a high quality and low to no visible artefacts. An example is shown in Figure 21.3b.

FACEMORPHER FaceMorpher is an open-source implementation using Python, realising the morphing concept described in Chapter 10. In the version applied for this work, the algorithm uses STASM for landmark localisation. Delaunay triangles, which are formed from the landmarks, are warped and blended. The area outside the landmarks is averaged. The generated morphs show strong artefacts in particular in the area of neck and hair. An example is shown in Figure 21.3c.

OPENCV The OpenCV based algorithms is a self implemented morphing algorithm derived from "Face Morph Using OpenCV"[4]. This algorithm works similar to FaceMorpher. Important differences between the algorithms are that for landmark detection Dlib is used instead of STASM and that for this algorithm landmarks are positioned at the edge of the image, which are also used to create morphs. Thus, in contrast to FaceMorpher, the edge does not consist of an averaged image, but like the rest of the image, of morphed triangles. However, strong artefacts outside the face area can be observed, which is mainly due to missing landmarks. An example is shown in Figure 21.3d.

[2] www.wearemoment.com/FaceFusion
[3] github.com/alyssaq/face_morpher
[4] www.learnopencv.com/face-morph-using-opencv-cpp-python


(a) Subject 1 (b) FaceFusion (c) FaceMorpher

(d) OpenCV (e) UBO Morpher (f) Subject 2

Figure 21.3: Examples of morphed face images from all four algorithms

UBO MORPHER UBO Morpher is the morphing tool of University of Bologna, as used, e.g., in [40]. This algorithm receives two input images as well as the corresponding landmarks. Dlib landmarks were used in this work. The morphs are generated by triangulation, warping and blending. To avoid artefacts in the area outside the face, the morphed face is copied to the background of one of the original images. Even if the colors are adjusted, visible edges may appear at borderline of the blended areas. An example is shown in Figure 21.3e.

The morph input images, pre-selected in Section 21.1, are used to create the morphs. A blind morphing of each morph candidate with each other would result in over 100,000 morphs for the morph input images included in FERET alone. The massive imbalance between morphs and bona fide passport images would unilaterally affect the training of the classifiers and cause a bias in the evaluation. Hence, morph pairs are formed in a meaningful manner, in order to keep the ratio between morphs and bona fide images in balance. Two parameters, namely sex and whether the subject wears glasses, are taken into account for the construction of the morph pairs. Morphing subjects of different sexes usually results in morphs with unnatural appearance. The creation of morphs with subjects of different sex are not to be expected in the real scenario, thus they are excluded from the database. Furthermore, it has been found, that if two subjects wearing glasses are morphed, the resulting morph contains double glasses. To avoid this kind of artefacts, morph pairs are formed with at most one subject wearing glasses.

The morph pairs are formed within one face database, in order to enable a clear separation of datasets during training and evaluation. Due to the different number of morph input images per subject in both databases, different protocols are defined.

FERET contains one morph input image per subject. Per sex, the images are sorted such that subjects with and without glasses are alternating. Since the database contains more subjects without glasses, the end of the list contains only subjects without glasses. From the list, a morph pair is formed from each subject with its successor (the last subject forms a morph pair with the first one), resulting in the same number of morph pairs as morph input images.

In FRGC, up to two morph input images are available per subject. Based on the concept of the image pair creation of FERET, for each sex a list is created consisting of the first morph input images, from which the morph pairs are created. A second list, containing the available second morph input images, is constructed in the same manner, meaning that no two subjects wearing glasses are adjacent to each other. Since not all subjects available in the database provide two morph input images, the order of subjects in both lists might differ. From the second list, the morphing pairs are created by keeping a distance of two subjects between the paired images. This procedure ensures that the morphing pairs are not containing the same subjects as those created from the first list and that no two subjects wearing glasses are combined.

With each morphing tool morphs are created from all available morph pairs. The morphs are created with an α_b and α_w of 0.5, however, due to the automatic improvement processes of FaceFusion and UBO Morpher, the morphs created by these algorithms are not symmetrical.

21.3 IMAGE POST-PROCESSING

The passport images (morph and bona fide) and the TLC images are post-processed in a different way. The TLC images are converted to greyscale, as some camera systems used at the border are only providing monochrome images. Since the morphing algorithms produce different, and sometimes recognisable, outputs, for example, by partially normalising the images, all passport images (including the bona fides) are normalised according to the procedure described in Section 15.1, in order to prevent over-fitting to artefacts not present

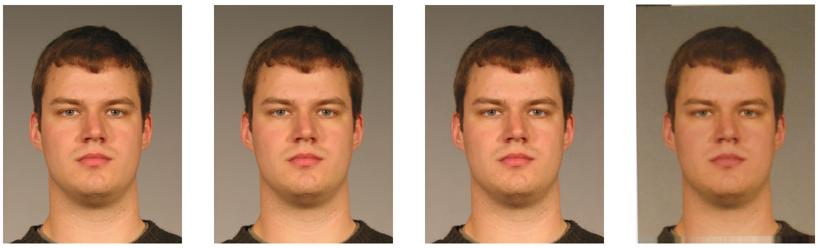

(a) NPP (b) RS (c) JP (d) PS

Figure 21.4: Examples of an original image and the three post-processing types

in a real scenario, such as different image sizes between morphs and bona fides. During the normalisation process, images are scaled to 960 × 720 pixels, resulting in a face region of 160 × 160 pixels.

Depending on the process by which the facial image is inserted into the passport, various post-processing steps are performed on the image. To reflect the realistic scenarios, the database contains four different post-processing chains for all passport photographs (Examples are shown in Figure 21.4):

UNPROCESSED The images are not further processed. In the text below referred to as *NPP* (no post-processing). This serves as baseline.

RESIZED The resolution of the images is reduced by half, reflecting the average size of a passport image. In the text below referred to as *RS*. This pre-processing corresponds to the scenario that an image is submitted digitally by the applicant.

JPEG2000 The images are resized by half and then compressed using JPEG2000, a wavelet-based image compression method that is recommended for EU passports [33]. The setting is selected in a way that a target file size of 15KB is achieved. This scenario reflects the post-processing path of passport images if handed over digitally at the application desk. In the text below referred to as *JP*.

PRINT/SCAN - JPEG2000 The original images (uncompressed and not resized) are first printed with a high quality laser printer (*Fujifilm Frontier 5700R Minlab* on *Fujicolor Crystal Archive Paper Supreme HD Lustre photo paper*) and then scanned with a premium flatbed scanner (*Epson DS-50000*) with 300 dpi. A dust and scratch filter is then applied in order to reduce image noise. Subsequently, the images are resized by half and then compressed to 15 KB using JPEG2000.[5] This scenario

[5] Due to the lustre print, the scans exhibit a visible pattern of the paper surface, which is only partly removed by the dust and scratch filter and results in stronger compression artefacts than for scans of glossy prints.

DATABASE	GENUINE COMP. BONA FIDE COMP.	IMPOSTOR COMPARISONS	MORPH COMPARISONS	BONA FIDE SAMPLES	MORPH SAMPLES
FERET	791	418,966	791	530	529
FRGC	3,298	1,695,086	3,246	984	964

Table 21.3: Number of comparisons per post-processing in the resulting database

reflects the post-processing path of passport images if handed over at the application desk as a printed photograph. In the text below referred to as *PS*.

The properties of the resulting database are listed in Table 21.3. The number of Genuine and Impostor comparisons is relevant for the analysis of the recognition performance of FRSs. For the evaluation of differential MAD algorithms the number of bona fide comparisons and morph comparisons is relevant, for S-MAD algorithms the number of bona fide samples and morph samples. The values given are per post-processing, quadrupling the actual number of passport images contained in the database.

22 SUMMARY

Part V describes the database used for training and evaluation of the MAD algorithms in this thesis. At present no database with morphed images suitable for the training of MAD algorithms is available, hence databases for this thesis were generated by the author.

In order to create a database with morphed images, a face database with suitable images is required. The images have to meet certain prerequisites in order to be applicable for the morphing process. These requirements include pose, artefacts and image quality. More details about the required characteristics of suitable facial images are given in Section 20.1. If the database should be usable for testing differential MAD algorithms as well, then, in addition to the morphed and bona fide images in passport image quality, TLCs are required as well, which, in the real scenario, may exhibit a significantly lower quality due to the uncontrolled recording conditions and should therefore not exhibit passport image quality in the database either. In Section 20.2 an overview of face databases available for research is given along with an assessment of their suitability for the creation of a database for training and evaluation of MAD algorithms. Eventually only two databases were found to be suitable: *color FERET* and *FRGCv2*.

The selected databases are initially divided into images in passport quality and TLC image quality. Morph pairs are built from the images in passport photo quality. Four different morphing algorithms are used to create the morphs, namely FaceFusion, FaceMorpher, an OpenCV based algorithm and UBO Morpher. The properties and functionality of the different algorithms are described in Section 21.2.

Depending on the process chain during the passport application process, the passport images may undergo various post-processing steps. In order to reproduce these, the passport images in the database (both morphs and bona fide) are subjected to various post-processing operations. *NPP* are the not post-processed images and serve as baseline, *RS* corresponds to the post-processing of a digitally transmitted passport image. *JP* corresponds to the post-processing of a digitally transmitted image stored in the passport and *PS* corresponds to an analogously transmitted, scanned and subsequently stored passport image.

Part VI

EXPERIMENTAL EVALUATION

23 VULNERABILITY ASSESSMENT

First, it is investigated whether the morphs generated for the database pose a threat to state-of-the-art FRSs. For this purpose the metrics described in Section 11.2.1 are applied. Four FRSs (two open source and two commercial) are examined, which are described in detail in Section 23.1.

23.1 FACIAL RECOGNITION SYSTEMS

As open source algorithms *FaceNet* and *ArcFace* are employed, whose feature extractors are described in Section 16.6.1 and Section 16.6.2 respectively. In the original implementation, the distances between the feature vectors (the distance score of the FRS) are calculated as L2-norm in case of *FaceNet* and as squared L2-norm in case of *ArcFace*. In this thesis, the L2-norm is used for both classifiers, resulting in an altered value range for *ArcFace*, but no changes in the error rates.

The commercial algorithms used are *Eyedea*, which is described in Section 16.6.3, and another commercial system, which, due to terms of use, is referred to *COTS* in the remainder of this work. For both commercial systems the internally used algorithms are unknown, thus only the results can be evaluated.

23.2 RESULTS

First, the face recognition performance of the algorithms is evaluated on the database created in Chapter 21. The thresholds of the face recognition algorithms are set to an FMR of 0.1%, as recommended by Frontex in [127] for border control systems. Based on this threshold, the vulnerability of the face recognition algorithms for morphing attacks is evaluated in Section 23.2.2.

23.2.1 *Recognition Performance*

For the evaluation of the recognition performance of the algorithms, the databases arranged in Chapter 21, namely FERET and FRGC, are evaluated independently. For the evaluation, only the unprocessed passport images (*NPP*) are considered. As indicated in Table 21.3, a significant imbalance between the number of genuine and impostor comparisons exists, which, however, does not influence the evaluation of the algorithms negatively, but rather results in a more meaningful impostor distribution. The PDFs of the examined algorithms on both

DATABASE	ALGORITHM	FMR	FNMR	TMR	THRESHOLD
FERET	FaceNet	0.1%	6.3%	93.7%	0.76
	ArcFace	0.1%	0%	100%	1.12
	Eyedea	0.1%	0%	100%	0.42
	COTS	0.1%	0%	100%	0.63
FRGC	FaceNet	0.1%	71.7%	28.3%	0.76
	ArcFace	0.1%	5.3%	94.7%	1.09
	Eyedea	0.1%	5.4%	94.6%	0.39
	COTS	0.1%	0%	100%	0.62

Table 23.1: Performance of face recognition algorithms

databases are shown in Figure 23.1 along with the determined threshold values. The open source FRSs provide a distance (dissimilarity score) as result, whereas the commercial FRSs return a similarity score. In order to achieve a uniform presentation, the results of the open source FRSs are displayed with inverted x-axis. In general, it can be observed that the genuine and the impostor comparisons of the FERET database are significantly more distinctive than those of the FRGC, regardless of the face recognition algorithm applied. This behaviour can be attributed to the nature of the TLC images. As described in Section 21.1, the TLC images of FERET usually only contain variations in pose and facial expression, whereas the TLC images of FRGC, with various backgrounds, lighting and sharpness, are of less stable quality, rendering the face recognition process considerably harder.

The detailed values for the examined algorithms are listed in Table 23.1. It can be observed, that the threshold value for each algorithm behaves similar for both databases, suggesting an almost identical impostor distribution. In contrast, the significant change in FNMR from FERET to FRGC for *FaceNet* shows that the genuine distribution of FRGC is much closer to the impostor distribution. Even though the FNMR of the other FRSs is not measurable for either database, it can be concluded from Figure 23.1 that the genuine distributions of FRGC are considerably closer to the impostor distribution than for FERET. Furthermore, it can be observed that face recognition algorithms clearly separating the comparisons of the FERET database (e.g. *COTS* in Figure 23.1c) are more capable of separating the comparisons of FRGC (see Figure 23.1h) as well, thus, a better generalisation capability of these classifiers can be concluded.

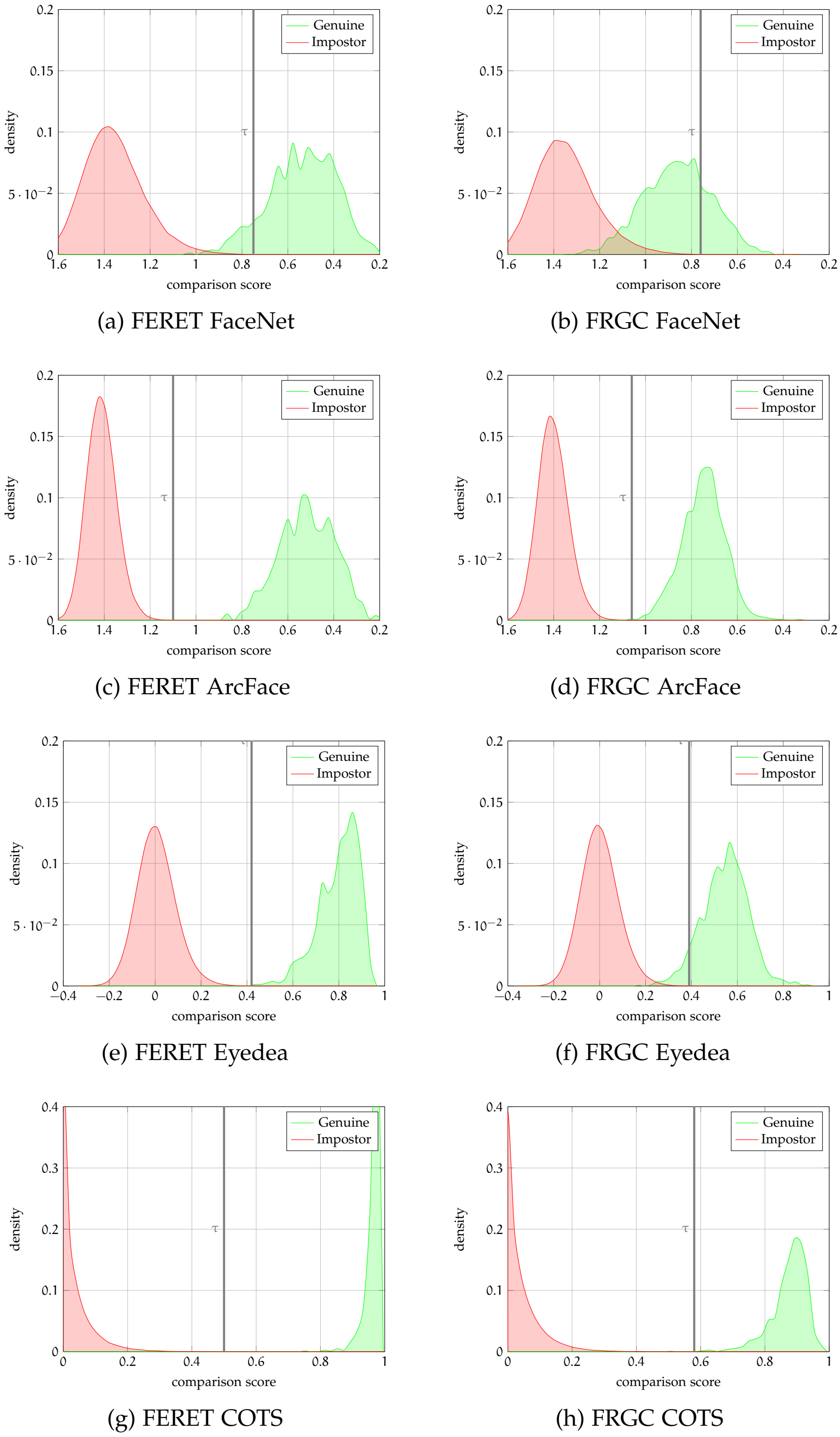
23.2.2 *Vulnerability to Morphing Attacks*

The vulnerability of the FRSs to morphing attacks can be analysed on the basis of the thresholds defined in Section 23.2.1. Figure 23.2 depicts, in addition to the genuine and impostor distributions and

DATABASE	ALGORITHM	MMPMR/RMMR			
		FACEFUSION	FACEMORPHER	OPENCV	UBO
FERET	FaceNet	31.4%/37.3%	17.6%/24.3%	18.0%/24.3%	29.3%/35.6%
	ArcFace	96.2%/96.2%	81.5%/81.5%	85.0%/85.0%	95.2%/95.2%
	Eyedea	71.6%/71.6%	90.0%/90.0%	87.5%/87.5%	72.6%/72.6%
	COTS	97.7%/97.7%	90.7%/90.7%	92.4%/92.4%	99.1%/99.1%
FRGC	FaceNet	10.7%/82.4%	5.2%/76.9%	4.6%/76.3%	8.0%/79.3%
	ArcFace	92.0%/97.3%	71.3%/76.6%	74.5%/79.8%	87.9%/93.4%
	Eyedea	64.7%/70.2%	64.6%/70.1%	46.7%/52.1%	47.3%/52.7%
	COTS	98.3%/98.3%	87.2%/87.2%	90.4%/90.4%	97.8%/97.8%

Table 23.2: Vulnerability of face recognition algorithms to morphing attacks

the threshold τ, the distribution for comparisons with the morphs generated by the morphing algorithms described in Section 21.2. In general, it can be stated that the distributions of morphing attacks are situated between the impostor and genuine distribution. The distributions of the different morphing algorithms are situated close to each other, however the distributions of the more complex morphing algorithms (FaceFusion and UBO Morpher) are consistently closer to the genuine distribution than the distributions of the more basic morphing algorithms (FaceMorpher and OpenCV). Furthermore, it can be observed that with more robust face recognition algorithms, which are able to separate the impostor and genuine distributions more effectively (for example, COTS on the FERET database in Figure 23.2g), the distribution of morphing attacks is closer to the genuine distribution than with less robust algorithms achieving a less clear separation of genuine and impostor distribution (for example FaceNet on the FRGC database in Figure 23.2b). Detailed error values are given in Table 23.2. The error metrics MMPMR and RMMR described in Section 11.2.1 are reported. Due to the limited number of morph comparisons per subject, the MinMax-MMPMR defined in equation 11.2 is applied. The error metrics are estimated per morphing algorithm.



(a) FERET FaceNet

(b) FRGC FaceNet

(c) FERET ArcFace

(d) FRGC ArcFace

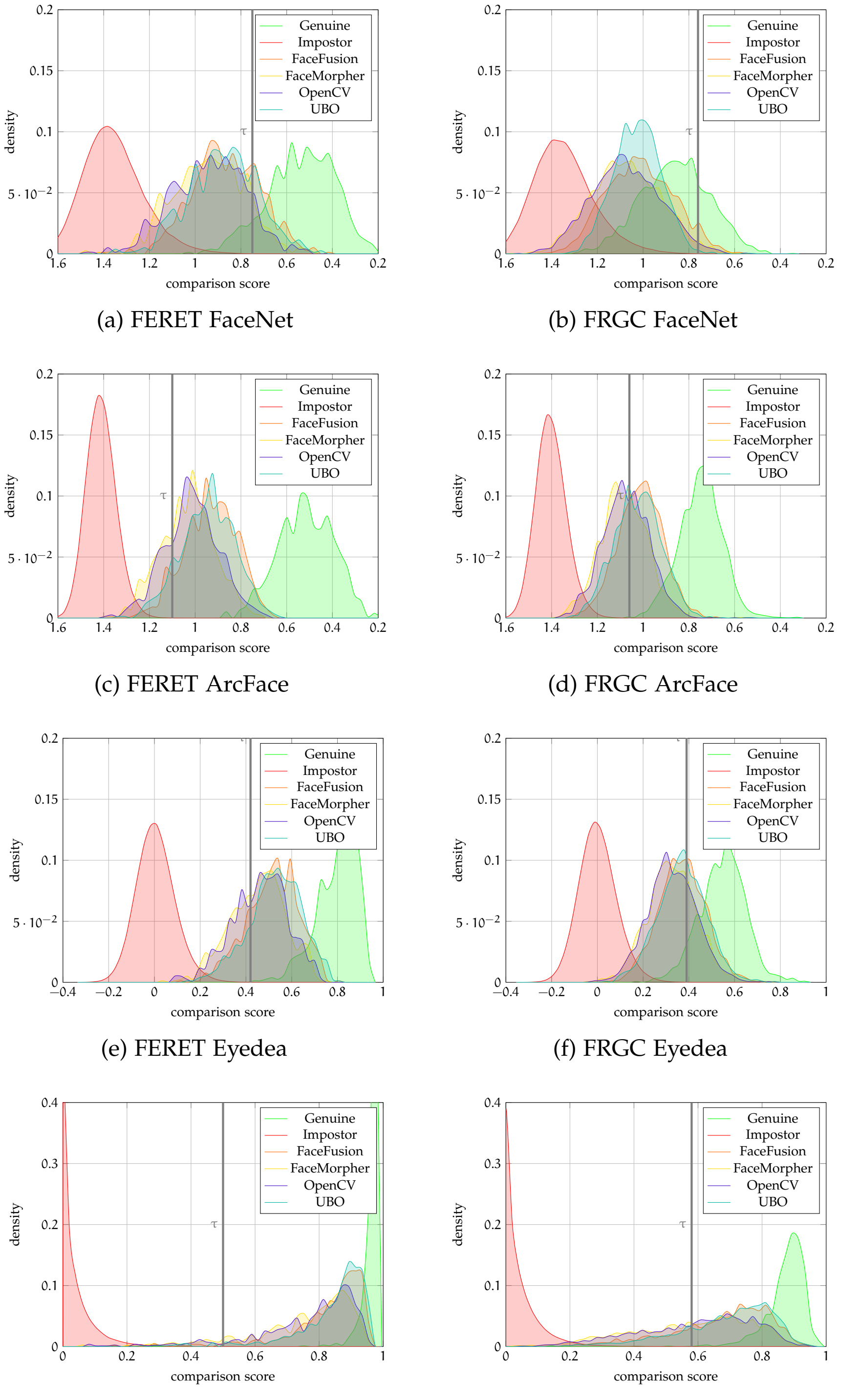
(e) FERET Eyedea

(f) FRGC Eyedea

(g) FERET COTS

(h) FRGC COTS

Figure 23.1: PDFs of comparison scores for the evaluated FRSs. The estimated threshold for an FMR of 0.1% is depicted by τ.



(a) FERET FaceNet

(b) FRGC FaceNet

(c) FERET ArcFace

(d) FRGC ArcFace

(e) FERET Eyedea

(f) FRGC Eyedea

(g) FERET COTS

(h) FRGC COTS

Figure 23.2: Susceptibility of the evaluated FRSs to morphing attacks

24 MORPHING ATTACK DETECTION PERFORMANCE ASSESSMENT

Different combinations of the features described in Chapter 16 and Chapter 18 are evaluated on the database described in Chapter 21. Due to the large number of possible configurations regarding databases, morphing algorithms, post-processings, feature extractors and classifiers, the experiments are systematically designed, in order to reduce the number of results to a comparable set, without lowering the significance of the evaluation. In experiment 1 the influence of a database shift is investigated. In experiment 2 suitable combinations of feature extractors and classifiers for the single image and differential scenario are determined. In experiment 3 these combinations are evaluated with respect to different post-processing scenarios. In experiment 4 the performance of a fusion of the algorithms selected in experiment 2 is analysed.

24.1 EXPERIMENT 1 - DATABASE SHIFT

In most evaluations of machine learning algorithms a database is divided into a training- and a test-set. This procedure was followed in the initial publications regarding MAD, for example in [135] and [137]. However, this methodology of evaluation bears the risk, that the database may contain properties simplifying the classification, which cannot be expected from realistic data. This danger is aggravated by the fact, that, due to the lack of morphing databases, the second class (the morphing attacks) has to be generated individually. If only one morphing algorithm is used in the database, it is likely that the MAD algorithm will overfit to artefacts specific to that particular morphing algorithm. The general influence of database shifts to the performance of MAD algorithms was prior proven in [136].

24.1.1 *Experimental Setup*

In order to allow for a quantitative statement about the influence of a change of database or morphing algorithm on the results of the MAD evaluation, the most successful algorithms determined in [137] on a single database and morphing algorithm are compared to the results obtained on the database described in Part V.

The database used in [137] is based on the facial images of FRGC, which were manually filtered for passport format images. In [137], texture, gradient bases and keypoint image descriptors in different

Training		Test		Feature $\left(\begin{smallmatrix}\text{Cells}\\ \text{Filtersize}\end{smallmatrix}\right)$							
Database	Morphing Algorithm	Database	Morphing Algorithm	LBP				BSIF			
				1×1 3×3	4×4 3×3	1×1 9×9	4×4 9×9	1×1 3×3	4×4 3×3	1×1 9×9	4×4 9×9
FRGC-Train	*OpenCV*	*FRGC-Test*	*OpenCV*	*5.1%*	*5.2%*	*13.7%*	*11.9%*	*2.9%*	*3.5%*	*16.5%*	*10.9%*
FRGC	OpenCV	FERET	OpenCV	24.4%	22.4%	32.9%	27.3%	25.6%	20.1%	27.8%	27.1%
FRGC	OpenCV	FERET	FaceMorpher	21.6%	17.7%	28.1%	25.7%	20.1%	16.3%	29.7%	28.2%
FRGC	OpenCV	FERET	FaceFusion	32.5%	24.4%	32.8%	31.0%	31.1%	26.2%	34.1%	30.3%
FRGC	OpenCV	FERET	UBO Morpher	27.0%	21.4%	29.2%	28.6%	27.3%	24.1%	32.9%	29.4%

Table 24.1: Performance difference introduced by evaluating on different databases and morphing algorithms for S-MAD algorithms

configurations are combined with an SVM with RBF kernel for single image and differential scenarios. Texture-based feature extractors have been found to achieve the highest performance, therefore the comparison below is limited to those. The presented percentage values are D-EERs. As no further findings are to be expected from a comparison over different operating points, the analysis of further operating points, e.g. BPCER-10 or BPCER-20, will be omitted in this experiment.

24.1.2 *Evaluation*

In Table 24.1 the comparison of the single image algorithms is shown. The first row of results shows the error rates obtained in [137], where the MAD algorithms are trained on one subset of the FRGC (FRGC-Train) and tested on another subset (FRGC-Test). The successive error rates are determined on the database described in Part V. If training and evaluation is performed on databases with different characteristics, a significant increase in the error rate, up to four times higher depending on the feature extractor, can be observed. This emphasizes the importance of independent databases for a robust evaluation. Furthermore, it can be observed that morphs generated by some algorithms are more difficult to detect than others. For example, the morphs generated by FaceFusion consistently produce higher error rates than those generated by OpenCV and FaceMorpher. The higher quality morphs, which are automatically post-processed, thus significantly reducing the number of artefacts, are more difficult to detect by the tested MAD algorithms. It is noticeable, that despite the fact that training is done exclusively on morphs created by OpenCV, the morphs created by FaceMorpher are easier to detect than morphs created by OpenCV. This leads to the conclusion that the origin of the morphs has an influence on how successful they can be detected, but the training itself is not influenced by the origin of the morphs.

For the differential scenario similar effects can be observed as for the single image scenario. The corresponding error values are listed in Table 24.2. First, it can be observed that training and evaluation on

Training		Test		Feature $\begin{pmatrix}\text{Cells}\\ \text{Filtersize}\end{pmatrix}$							
Database	Morphing Algorithm	Database	Morphing Algorithm	LBP				BSIF			
				1×1 3×3	4×4 3×3	1×1 9×9	4×4 9×9	1×1 3×3	4×4 3×3	1×1 9×9	4×4 9×9
FRGC-Train	*OpenCV*	*FRGC-Test*	*OpenCV*	*3.9%*	*3.9%*	*7.3%*	*7.4%*	*4.4%*	*4.7%*	*9.3%*	*9.8%*
FRGC	OpenCV	FERET	OpenCV	21.9%	28.8%	37.5%	38.7%	15.4%	18.1%	20.0%	20.1%
FRGC	OpenCV	FERET	FaceMorpher	18.6%	25.5%	35.9%	38.2%	14.1%	15.3%	19.8%	20.3%
FRGC	OpenCV	FERET	FaceFusion	23.9%	30.7%	39.2%	40.2%	18.4%	19.3%	21.6%	22.6%
FRGC	OpenCV	FERET	UBO Morpher	21.7%	29.3%	37.3%	39.8%	17.4%	18.1%	21.7%	21.7%

Table 24.2: Performance difference introduced by evaluating on different databases and morphing algorithms for differential MAD algorithms

entirely independent datasets yields significantly higher error values than training and testing on disjoint subsets of a single database. Furthermore, the observation made in the single image scenario, that morphs with lower quality (e.g. generated by FaceMorpher) are more likely to be detected than those with higher quality (e.g. generated by FaceFusion), can also be confirmed for the differential scenario.

24.1.3 *Discussion*

In a real world operation of a MAD system, no assumptions about the images' origin or eventual applied morphing algorithms can be made. Thus, the general statement can be formulated, that, regardless of the scenario, evaluation should be performed on datasets as independent as possible, in order to obtain results predicting the impact ot be expected in real operation. The image source (for both, bona fide or morphed images) has an major impact on the evaluation. Depending on the algorithm, the difference in D-EER between the usage of images from the same or different sources can exceed 20 percentage points. A further factor is the quality of the morphs to be detected. If the morphs were generated by an algorithm capable of producing high quality morphs, the morphs are more difficult to detect than morphs generated by less sophisticated morphing algorithms. However, the differences in D-EER for the tested algorithms are only in the range of lower one-digit percentage points. Nevertheless, MAD algorithms should be tested primarily on high-quality morphs, since it is likely that an attacker will produce the highest possible morph in a real scenario, and furthermore, since high-quality morphs represent the higher obstacle, it can be assumed that if an algorithm succeeds in detecting high-quality morphs, lower-quality morphs will also be detected. In contrast, the origin of the morphs used for training is less relevant. The tested MAD algorithms have been found to generalise well across different morphing algorithms, meaning the effect of the quality of the morphs to be tested has a higher impact on the evaluation than the fact whether the algorithms used for training were generated by the same morphing algorithm.

24.2 EXPERIMENT 2 - GENERAL SUITABILITY

In Chapter 16 different feature extractors are introduced, describing different image properties. In addition hypotheses are formulated, explaining why the extracted features should be suitable for the detection of morphed facial images. This experiment will investigate, to what extent the different features in combination with the classifiers described in Section 18.4 are in principle suitable to detect morphed facial images in a basic scenario.

24.2.1 *Experimental Setup*

The features described in Chapter 16 combined with the classifiers described in Section 18.4 are tested in both, the single image and the differential scenario, on the non post-processed images of the databases described in Part V. The experimental setup applies the conclusions drawn in Section 24.1. Training is performed on one of the two database, evaluation is performed on the other and vice versa. For training, the morphs generated by either OpenCV or FaceMorpher are used separately; evaluation is performed on the morphs generated by FaceFusion or UBO Morpher, as they are more difficult to detect.

24.2.2 *Evaluation*

Due to the large number of possible MAD algorithms, the algorithms are examined separately by category of the applied feature extractor. In the initial phase, only the D-EER is analysed. A summary of the best performing MAD algorithms, with an analysis of the operating points BPCER-10 and BPCER-20, is given in Section 24.2.3.

TEXTURE DESCRIPTORS In this paragraph, the morphing detection capabilities of texture descriptors in different configurations, as described in Section 16.1, are analysed.

The D-EERs of the respective algorithms in a single image scenario are listed in Table 24.3. In the following tables only the database used for training is indicated, the evaluation was performed on the respective other database. In general it can be stated, that, across all algorithms, morphs are more difficult to detect for FERET than for FRGC. Furthermore, it can be observed that SVM and gradient boosting based algorithms tend to provide a better detection performance compared to algorithms based on AdaBoost or Random Forest. Regardless of the applied feature type, algorithms dividing images into cells prior to feature extraction usually achieve a higher

Classifier	Training		Test	Feature $\left(\begin{smallmatrix}\text{Cells}\\\text{Filtersize}\end{smallmatrix}\right)$							
	Database	Morphing Algorithm	Morphing Algorithm	LBP				BSIF			
				1 × 1 3 × 3	4 × 4 3 × 3	1 × 1 9 × 9	4 × 4 9 × 9	1 × 1 3 × 3	4 × 4 3 × 3	1 × 1 9 × 9	4 × 4 9 × 9
SVM	FERET	FaceMorpher	FaceFusion	27.86%	22.00%	24.41%	19.38%	25.08%	19.51%	16.76%	14.33%
			UBO Morpher	20.65%	16.55%	22.53%	17.72%	19.88%	14.48%	15.04%	12.54%
		OpenCV	FaceFusion	24.99%	19.69%	22.10%	20.49%	21.73%	17.81%	16.09%	15.96%
			UBO Morpher	19.51%	15.87%	20.99%	18.06%	19.14%	13.84%	15.16%	13.34%
	FRGC	FaceMorpher	FaceFusion	32.91%	31.01%	35.19%	30.51%	32.41%	31.65%	33.16%	30.76%
			UBO Morpher	28.23%	26.71%	34.81%	30.38%	27.97%	26.71%	29.87%	28.99%
		OpenCV	FaceFusion	31.14%	26.20%	34.05%	30.25%	32.53%	24.43%	32.78%	31.01%
			UBO Morpher	27.34%	24.05%	32.91%	29.37%	26.96%	21.39%	29.24%	28.61%
Random Forest	FERET	FaceMorpher	FaceFusion	30.54%	31.53%	32.73%	30.72%	24.59%	31.06%	20.15%	28.47%
			UBO Morpher	19.69%	24.81%	30.94%	23.91%	24.62%	21.14%	30.76%	24.41%
		OpenCV	FaceFusion	28.38%	26.38%	27.33%	31.99%	29.86%	23.98%	18.67%	30.76%
			UBO Morpher	23.14%	19.48%	25.15%	28.91%	20.34%	17.41%	28.38%	26.13%
	FRGC	FaceMorpher	FaceFusion	32.41%	34.94%	38.99%	41.90%	35.70%	38.73%	41.27%	38.86%
			UBO Morpher	30.00%	28.61%	41.01%	38.35%	32.78%	25.57%	37.09%	39.75%
		OpenCV	FaceFusion	31.90%	31.14%	38.35%	36.46%	41.01%	34.68%	36.20%	37.97%
			UBO Morpher	31.14%	32.66%	37.72%	33.29%	33.29%	26.33%	30.51%	33.04%
AdaBoost	FERET	FaceMorpher	FaceFusion	24.99%	22.62%	32.82%	25.73%	29.18%	24.93%	23.91%	26.47%
			UBO Morpher	22.28%	19.41%	31.25%	25.39%	25.49%	24.28%	23.05%	25.02%
		OpenCV	FaceFusion	24.87%	19.48%	30.11%	25.67%	22.87%	23.64%	24.59%	23.76%
			UBO Morpher	22.59%	17.75%	29.83%	24.25%	22.99%	21.63%	23.88%	21.94%
	FRGC	FaceMorpher	FaceFusion	32.78%	32.78%	35.95%	33.80%	38.73%	34.56%	35.06%	31.65%
			UBO Morpher	29.49%	28.48%	35.44%	33.42%	32.03%	29.11%	32.78%	29.87%
		OpenCV	FaceFusion	33.04%	30.63%	36.20%	37.34%	30.51%	30.13%	34.30%	33.16%
			UBO Morpher	29.24%	28.99%	34.05%	34.94%	27.72%	27.34%	32.66%	29.75%
Gradient Boosting	FERET	FaceMorpher	FaceFusion	22.87%	22.03%	26.10%	27.06%	29.92%	25.18%	23.45%	24.01%
			UBO Morpher	19.91%	18.15%	24.68%	23.82%	24.93%	22.25%	22.40%	21.73%
		OpenCV	FaceFusion	22.22%	21.14%	27.30%	25.30%	25.76%	22.43%	23.45%	24.04%
			UBO Morpher	19.32%	17.97%	26.19%	23.91%	22.53%	20.34%	21.60%	22.28%
	FRGC	FaceMorpher	FaceFusion	34.56%	33.42%	34.30%	34.18%	37.85%	36.96%	35.32%	30.63%
			UBO Morpher	29.87%	30.00%	33.54%	33.54%	31.65%	31.77%	33.29%	30.00%
		OpenCV	FaceFusion	32.53%	29.49%	35.32%	36.58%	35.57%	31.52%	33.92%	32.53%
			UBO Morpher	29.11%	27.85%	35.70%	35.06%	29.11%	26.58%	32.28%	32.03%

Table 24.3: Detection performance (D-EER) of texture descriptors with different configurations in single image scenario

performance. Furthermore, the algorithms can generally achieve better individual performance using a larger filter size, but, in particular with the morphs of the FERET database, a smaller filter size results in more robust results. When comparing the two feature types, it is noticeable that BSIF tends to perform better than LBP, however, in particular LBP with 4 × 4 cells and a filter size of 3 × 3 stands out due to its consistent performance, especially on the morphs of the FERET database, most other algorithms are struggling to detect. Regarding the choice of training data, no significant difference between the morphs created by FaceMorpher or OpenCV can be found. The difference in the evaluation of the separately trained algorithms is marginal and not uniform. However, there is a clear trend in terms of the choice of data to be evaluated. Morphs created by FaceFusion are generally more difficult to detect than morphs created by the UBO Morpher.

The best performing algorithm is BSIF with 4 × 4 cells and a filter size of 9 × 9 in combination with an SVM, achieving an average D-EER of 14% on FRGC and 30% on FERET, as well as LBP with 4 × 4 cells and a filter size of 3 × 3 in combination with an SVM, achieving an average D-EER of 18.5% on FRGC and 27% on FERET.

The D-EERs of the texture descriptor based algorithms in a differential scenario are listed in Table 24.4. In contrast to the single image scenario, the morphs of the FERET database are usually easier to detect than those of the FRGC database. This effect is due to the fact, that the TLC images of the FRGC contain a much higher variance in illumination and sharpness. Furthermore, it can be observed that depending on the database and the used feature extractor, the choice of the optimal classifier varies, however, the applied random forest classifier is not suitable and rarely achieves D-EERs below 30%. Independent of the feature type, smaller filter sizes achieve better performances in easier scenarios, but larger filter sizes prove to be more robust across all scenarios. Cell subdivision prior to feature extraction can improve the detection performance. As for the single-image scenario it can be observed that morphs generated by FaceFusion are more difficult to detect than morphs generated by UBO Morpher. The choice of morphs used for training may influence the evaluation results, however, no definite scheme can be observed.

Many of the tested algorithms reach D-EERs around 30%, rendering texture descriptors more suitable for creating single image algorithms. Thus it can be concluded that the information relevant to MAD is contained in the discrete values of the features of the references and not in the difference to the TLC considered in the differential scenario. The best performing algorithm is LBP with a cell division into 4 × 4 cells and a filter size of 3 × 3, achieving an average D-EER of 26.2% on FRGC and 23.4% on FERET. Even though larger filters are more

Classifier	Training		Test	Feature $\begin{pmatrix}\text{Cells}\\\text{Filtersize}\end{pmatrix}$							
	Database	Morphing Algorithm	Morphing Algorithm	LBP				BSIF			
				1×1 3×3	4×4 3×3	1×1 9×9	4×4 9×9	1×1 3×3	4×4 3×3	1×1 9×9	4×4 9×9
SVM	FERET	FaceMorpher	FaceFusion	48.17%	46.32%	43.08%	36.06%	49.31%	46.72%	38.89%	30.91%
			UBO Morpher	48.10%	47.40%	42.00%	34.58%	51.12%	48.78%	38.43%	29.55%
		OpenCV	FaceFusion	44.99%	43.76%	41.33%	35.41%	46.93%	43.39%	37.04%	29.40%
			UBO Morpher	44.84%	44.81%	40.52%	33.78%	48.32%	45.36%	36.73%	27.83%
	FRGC	FaceMorpher	FaceFusion	33.04%	37.72%	42.53%	39.75%	24.94%	23.16%	22.28%	21.77%
			UBO Morpher	30.38%	35.44%	41.77%	40.63%	21.65%	20.76%	22.41%	21.77%
		OpenCV	FaceFusion	23.92%	30.76%	39.24%	40.25%	18.48%	19.37%	21.65%	22.66%
			UBO Morpher	21.77%	29.37%	37.34%	39.87%	17.47%	18.10%	21.77%	21.77%
Random Forest	FERET	FaceMorpher	FaceFusion	45.12%	44.53%	35.47%	46.50%	36.12%	54.24%	41.54%	38.15%
			UBO Morpher	39.97%	40.83%	32.82%	44.01%	34.85%	50.66%	38.21%	38.06%
		OpenCV	FaceFusion	60.46%	51.06%	34.05%	36.09%	55.56%	37.04%	28.84%	46.53%
			UBO Morpher	57.38%	47.18%	31.80%	32.91%	55.90%	39.23%	47.80%	45.79%
	FRGC	FaceMorpher	FaceFusion	21.01%	37.97%	39.11%	41.65%	21.01%	40.00%	36.46%	30.76%
			UBO Morpher	37.47%	36.08%	41.65%	43.42%	29.75%	35.44%	36.20%	32.03%
		OpenCV	FaceFusion	26.71%	23.92%	28.35%	46.08%	20.76%	35.82%	27.09%	32.03%
			UBO Morpher	22.15%	25.70%	26.46%	47.47%	32.66%	36.96%	27.47%	33.67%
AdaBoost	FERET	FaceMorpher	FaceFusion	32.57%	27.92%	38.74%	34.30%	31.46%	28.78%	33.19%	31.74%
			UBO Morpher	29.12%	24.56%	36.70%	34.02%	30.11%	29.55%	31.71%	32.63%
		OpenCV	FaceFusion	33.25%	27.33%	36.15%	35.04%	33.87%	30.54%	31.86%	30.76%
			UBO Morpher	30.76%	25.08%	35.84%	33.90%	34.45%	29.89%	31.09%	30.02%
	FRGC	FaceMorpher	FaceFusion	21.14%	24.30%	31.01%	29.75%	26.08%	27.72%	27.34%	30.76%
			UBO Morpher	17.72%	19.62%	29.75%	29.87%	22.03%	25.32%	26.71%	30.25%
		OpenCV	FaceFusion	19.87%	22.41%	32.53%	34.81%	23.80%	23.29%	30.00%	31.52%
			UBO Morpher	17.09%	19.11%	32.28%	33.92%	20.25%	22.91%	26.96%	30.00%
Gradient Boosting	FERET	FaceMorpher	FaceFusion	45.02%	31.62%	36.83%	38.15%	39.04%	36.15%	30.94%	31.74%
			UBO Morpher	42.13%	28.66%	35.56%	37.87%	38.64%	34.67%	30.45%	30.94%
		OpenCV	FaceFusion	33.96%	31.31%	36.02%	36.52%	36.24%	31.16%	29.61%	31.77%
			UBO Morpher	32.30%	28.20%	34.98%	36.18%	35.25%	31.56%	28.60%	29.58%
	FRGC	FaceMorpher	FaceFusion	20.63%	25.32%	26.84%	29.49%	27.47%	25.70%	27.34%	31.01%
			UBO Morpher	18.35%	22.15%	24.81%	28.99%	22.66%	24.05%	24.94%	30.89%
		OpenCV	FaceFusion	17.85%	18.23%	26.58%	32.15%	20.89%	20.89%	26.08%	29.87%
			UBO Morpher	17.22%	17.34%	24.94%	31.65%	18.61%	20.51%	25.32%	28.35%

Table 24.4: Detection performance (D-EER) of texture descriptors with different configurations in differential scenario

robust, the errors of this algorithm are consistently lower than the error values of the other algorithms.

GRADIENT BASED DESCRIPTORS In this paragraph, the morphing detection capabilities of gradient based descriptors in different configurations, as described in Section 16.2, are analysed.

The D-EERs of the respective algorithms in a single image scenario are listed in Table 24.5. For the HOG based algorithms, cell subdivision was omitted, since HOG subdivides the image to be analysed into cells during feature extraction. Mean of gradient based approaches do not achieve a detection performance below 30% D-EER, thus the suitability of these algorithms for S-MAD algorithms can be excluded. A closer examination of the gradient images and a more refined extraction of the contained information may yield different results. For example, the HOG based algorithms demonstrate the general suitability of gradient based features for MAD. In particular, when combined with an SVM, average EERs of 14% for FRGC and 22.1% for FERET can be achieved. As with the previous single image algorithms, the detection of FRGC morphs and the detection of morphs created by the UBO Morpher results in a lower error range. The choice of morphs used for training has almost no influence on the result.

The D-EERs of the respective algorithms in a differential scenario are listed in Table 24.6. Similar to the single image scenario, the mean of gradient based MAD algorithms are barely able to detect morphs. Regardless of whether a cell division is applied or not, usually only D-EER above 30% are obtained. HOG based MAD, however, exhibits an acceptable detection performance even in the differential scenario. In combination with an SVM, an average D-EERs of 25.9% for FRGC and 17.4% for FERET can be achieved, confirming, as with other differential algorithms, that, in the differential scenario, FERET morphs are easier to detect than FRGC morphs.

It can be summarised, that out of the category of tested gradient based descriptors, only HOG is applicable for the creation of MAD algorithms, whereas it is applicable for both, the single image and the differential scenario.

KEYPOINT DESCRIPTORS In this paragraph, the morphing detection capabilities of keypoint descriptors in different configurations, as described in Section 16.3, are analysed. In contrast to the evaluation of the previous feature extractors, the table of results are not split according the scenario (single image or differential), but according cell division. Without cell division the feature extraction defined in Section 16.3 returns a scalar value, which can be directly applied for decision making. Thus, in this case, no training database is required. If, however, a cell division is applied prior to the feature extraction, a feature vector is obtained, requiring a subsequent classification.

Classifier	Training		Test	Feature (Cells)		
	Database	Morphing Algorithm	Morphing Algorithm	Mean of Gradients		HOG
				1 × 1	4 × 4	1 × 1
SVM	FERET	FaceMorpher	FaceFusion	47.06%	36.83%	*14.73%*
			UBO Morpher	43.73%	33.99%	*13.25%*
		OpenCV	FaceFusion	47.06%	37.72%	*14.48%*
			UBO Morpher	45.24%	34.88%	*13.78%*
	FRGC	FaceMorpher	FaceFusion	47.47%	42.78%	*24.05%*
			UBO Morpher	46.71%	41.01%	*19.75%*
		OpenCV	FaceFusion	47.09%	43.80%	*23.92%*
			UBO Morpher	45.82%	40.76%	*20.63%*
Random Forest	FERET	FaceMorpher	FaceFusion	56.98%	38.92%	28.54%
			UBO Morpher	51.16%	33.96%	28.38%
		OpenCV	FaceFusion	56.02%	38.92%	24.87%
			UBO Morpher	51.34%	35.07%	24.84%
	FRGC	FaceMorpher	FaceFusion	45.57%	43.04%	42.28%
			UBO Morpher	45.57%	39.11%	38.99%
		OpenCV	FaceFusion	52.03%	42.28%	32.53%
			UBO Morpher	46.96%	34.94%	30.76%
AdaBoost	FERET	FaceMorpher	FaceFusion	43.05%	38.71%	21.48%
			UBO Morpher	48.44%	38.64%	20.59%
		OpenCV	FaceFusion	47.64%	39.45%	25.15%
			UBO Morpher	45.24%	37.66%	24.35%
	FRGC	FaceMorpher	FaceFusion	45.95%	43.67%	29.49%
			UBO Morpher	45.82%	41.27%	27.34%
		OpenCV	FaceFusion	47.72%	42.28%	32.41%
			UBO Morpher	47.85%	40.63%	27.97%
Gradient Boosting	FERET	FaceMorpher	FaceFusion	51.77%	40.80%	25.76%
			UBO Morpher	49.86%	37.38%	27.12%
		OpenCV	FaceFusion	49.80%	40.55%	25.08%
			UBO Morpher	49.28%	38.46%	25.76%
	FRGC	FaceMorpher	FaceFusion	46.71%	41.77%	29.87%
			UBO Morpher	45.06%	39.24%	27.09%
		OpenCV	FaceFusion	48.86%	43.80%	32.15%
			UBO Morpher	47.97%	42.41%	30.13%

Table 24.5: Detection performance (D-EER) of gradient based descriptors with different configurations in single image scenario

Classifier	Training		Test	Feature (Cells)		
	Database	Morphing Algorithm	Morphing Algorithm	Mean of Gradients		HOG
				1 × 1	4 × 4	1 × 1
SVM	FERET	FaceMorpher	FaceFusion	47.61%	48.01%	*27.52%*
			UBO Morpher	46.32%	46.50%	*25.24%*
		OpenCV	FaceFusion	47.61%	48.17%	*26.41%*
			UBO Morpher	46.63%	47.24%	*24.31%*
	FRGC	FaceMorpher	FaceFusion	37.85%	34.81%	*19.37%*
			UBO Morpher	36.46%	29.37%	*15.70%*
		OpenCV	FaceFusion	37.72%	34.18%	*18.73%*
			UBO Morpher	35.19%	28.86%	*15.70%*
Random Forest	FERET	FaceMorpher	FaceFusion	46.87%	42.77%	35.04%
			UBO Morpher	46.16%	39.72%	33.68%
		OpenCV	FaceFusion	47.18%	41.88%	34.36%
			UBO Morpher	44.19%	38.98%	32.60%
	FRGC	FaceMorpher	FaceFusion	42.66%	46.20%	22.91%
			UBO Morpher	43.42%	41.14%	23.92%
		OpenCV	FaceFusion	45.95%	48.10%	27.59%
			UBO Morpher	43.67%	40.25%	27.72%
AdaBoost	FERET	FaceMorpher	FaceFusion	46.10%	43.30%	27.43%
			UBO Morpher	48.32%	39.23%	28.63%
		OpenCV	FaceFusion	47.12%	43.61%	26.41%
			UBO Morpher	45.70%	39.94%	26.32%
	FRGC	FaceMorpher	FaceFusion	32.41%	37.85%	30.13%
			UBO Morpher	25.95%	28.73%	28.61%
		OpenCV	FaceFusion	56.96%	39.49%	31.39%
			UBO Morpher	55.19%	31.39%	28.10%
Gradient Boosting	FERET	FaceMorpher	FaceFusion	47.24%	46.47%	30.35%
			UBO Morpher	45.86%	45.02%	30.60%
		OpenCV	FaceFusion	47.40%	45.92%	29.18%
			UBO Morpher	45.82%	43.48%	29.77%
	FRGC	FaceMorpher	FaceFusion	38.35%	34.05%	30.38%
			UBO Morpher	36.96%	30.51%	28.73%
		OpenCV	FaceFusion	37.85%	35.95%	28.23%
			UBO Morpher	36.71%	28.99%	25.32%

Table 24.6: Detection performance (D-EER) of gradient based descriptors with different configurations in differential scenario

Test		Feature (no cell division)			
Database	Morphing Algorithm	Single Image		Differential	
		SIFT	SURF	SIFT	SURF
FERT	FaceFusion	40.51%	39.24%	36.84%	33.92%
	UBO Morpher	36.84%	38.86%	34.68%	31.90%
FRGC	FaceFusion	32.11%	32.70%	39.94%	38.03%
	UBO Morpher	25.89%	27.21%	35.38%	35.81%

Table 24.7: Detection performance (D-EER) of keypoint descriptors with different configurations in single image and differential scenario without cell division

The D-EERs of keypoint descriptor based algorithms without cell division are listed in Table 24.7. Due to the fact, that the number of possible combinations is greatly reduced by the lack of training parameters, the evaluation of the results is shortened. It can be seen that in both scenarios the keypoint extractors without cell division reach D-EERs above the 30%, which is why they are basically not suitable for creating MAD algorithms.

The D-EERs of keypoint descriptor based algorithms with cell division into 4×4 cells are listed in Table 24.8. In the case of SIFT and SURF with cell division, the evaluation has more parameters than the evaluation without cell division, as the extracted feature vectors are to be classified, resulting in multiple combination options. The single image algorithms tend to achieve slightly better results than the differential algorithms, although D-EERs below 30% are rarely achieved. Therefore, these applied keypoint algorithms can generally be considered unsuitable for MAD. A more sophisticated method for extracting keypoint characteristics might have a higher potential and should be considered in future work.

LANDMARK DESCRIPTORS In this paragraph, the morphing detection capabilities of two different landmark descriptors, as described in Section 16.4, are analysed.

The basic idea behind the use of landmarks for MAD is to detect the distortion of the face induced by the morphing process on the basis of the offset of the facial landmarks. For this concept the comparison to a TLC is mandatory, which is why only the differential scenario is considered in the evaluation. Table 24.9 shows the determined D-EERs for the examined MAD algorithms. The feature extraction is, as mentioned in Section 16.4, based on the concept described in [24]. To the author's knowledge, the only difference to the recommended procedure is the omission of an automatic hyperparameter optimisation of the employed classifiers (in [24] only an SVM with RBF kernel was used). Still, the determined error rates considerably deviate from the nearly optimal detection rates stated in [24]. Regardless of the choice

Classifier	Training		Test	Feature (4 × 4 Cells)			
	Database	Morphing Algorithm	Morphing Algorithm	Single Image		Differential	
				SIFT	SURF	SIFT	SURF
SVM	FERET	FaceMorpher	FaceFusion	31.65%	34.70%	48.20%	47.46%
			UBO Morpher	25.58%	31.59%	47.43%	45.95%
		OpenCV	FaceFusion	31.03%	31.59%	49.37%	46.75%
			UBO Morpher	26.78%	29.92%	48.04%	45.89%
	FRGC	FaceMorpher	FaceFusion	37.59%	40.25%	44.30%	44.56%
			UBO Morpher	36.58%	38.48%	44.30%	43.04%
		OpenCV	FaceFusion	38.10%	40.63%	45.44%	45.57%
			UBO Morpher	36.58%	37.59%	45.82%	44.81%
Random Forest	FERET	FaceMorpher	FaceFusion	34.24%	37.66%	42.31%	43.88%
			UBO Morpher	37.78%	34.61%	36.98%	42.00%
		OpenCV	FaceFusion	41.14%	41.73%	50.54%	52.63%
			UBO Morpher	35.72%	35.25%	46.13%	52.97%
	FRGC	FaceMorpher	FaceFusion	46.20%	41.39%	49.11%	44.68%
			UBO Morpher	42.03%	36.96%	50.00%	40.63%
		OpenCV	FaceFusion	38.86%	42.66%	40.38%	46.33%
			UBO Morpher	41.90%	37.34%	39.62%	40.38%
AdaBoost	FERET	FaceMorpher	FaceFusion	29.31%	32.33%	38.61%	43.51%
			UBO Morpher	34.98%	27.92%	33.41%	39.60%
		OpenCV	FaceFusion	31.96%	33.13%	37.01%	46.47%
			UBO Morpher	27.09%	32.76%	41.63%	43.08%
	FRGC	FaceMorpher	FaceFusion	41.01%	38.86%	32.91%	39.75%
			UBO Morpher	37.85%	37.22%	29.87%	36.08%
		OpenCV	FaceFusion	50.63%	40.38%	44.30%	28.99%
			UBO Morpher	42.41%	38.61%	41.27%	49.37%
Gradient Boosting	FERET	FaceMorpher	FaceFusion	33.13%	35.62%	39.38%	44.50%
			UBO Morpher	28.04%	33.34%	35.07%	44.07%
		OpenCV	FaceFusion	35.22%	36.70%	42.25%	44.68%
			UBO Morpher	29.86%	33.37%	41.11%	43.91%
	FRGC	FaceMorpher	FaceFusion	41.39%	40.25%	43.16%	40.51%
			UBO Morpher	38.48%	37.22%	40.25%	36.96%
		OpenCV	FaceFusion	40.63%	39.75%	43.92%	37.97%
			UBO Morpher	38.10%	37.85%	42.03%	36.20%

Table 24.8: Detection performance (D-EER) of keypoint descriptors with different configurations in single image and differential scenario with cell division

Classifier	Training		Test	Feature	
	Database	Morphing Algorithm	Morphing Algorithm	Dlib	WING
SVM	FERET	FaceMorpher	FaceFusion	43.81%	43.94%
			UBO Morpher	42.92%	43.79%
		OpenCV	FaceFusion	45.51%	43.17%
			UBO Morpher	45.14%	42.84%
	FRGC	FaceMorpher	FaceFusion	39.85%	40.63%
			UBO Morpher	42.13%	41.01%
		OpenCV	FaceFusion	38.20%	40.89%
			UBO Morpher	38.83%	41.90%
Random Forest	FERET	FaceMorpher	FaceFusion	40.20%	36.92%
			UBO Morpher	37.40%	35.84%
		OpenCV	FaceFusion	36.10%	40.74%
			UBO Morpher	35.64%	38.61%
	FRGC	FaceMorpher	FaceFusion	34.90%	43.29%
			UBO Morpher	35.41%	44.81%
		OpenCV	FaceFusion	33.88%	43.54%
			UBO Morpher	33.38%	45.44%
AdaBoost	FERET	FaceMorpher	FaceFusion	46.78%	44.47%
			UBO Morpher	44.96%	43.85%
		OpenCV	FaceFusion	45.11%	43.61%
			UBO Morpher	45.29%	43.88%
	FRGC	FaceMorpher	FaceFusion	42.64%	38.48%
			UBO Morpher	44.92%	36.96%
		OpenCV	FaceFusion	44.16%	38.86%
			UBO Morpher	44.04%	38.61%
Gradient Boosting	FERET	FaceMorpher	FaceFusion	43.01%	44.96%
			UBO Morpher	42.30%	43.94%
		OpenCV	FaceFusion	43.04%	44.93%
			UBO Morpher	42.58%	42.71%
	FRGC	FaceMorpher	FaceFusion	38.07%	38.48%
			UBO Morpher	40.74%	38.73%
		OpenCV	FaceFusion	40.10%	38.86%
			UBO Morpher	40.48%	36.58%

Table 24.9: Detection performance (D-EER) of landmark descriptors with different configurations in differential scenario

Database	Training	Test	Feature		
	Morphing Algorithm	Morphing Algorithm	PRNU-1	PRNU-2	SPN
FERET	FaceMorpher	FaceFusion	39.58%	47.54%	42.13%
		UBO Morpher	36.93%	46.40%	33.68%
	OpenCV	FaceFusion	-	-	42.93%
		UBO Morpher	-	-	33.56%
FRGC	FaceMorpher	FaceFusion	31.98%	42.37%	59.24%
		UBO Morpher	27.10%	38.73%	51.14%
	OpenCV	FaceFusion	-	-	59.75%
		UBO Morpher	-	-	44.94%

Table 24.10: Detection performance (D-EER) of image noise pattern with different configurations in single image scenario

of classifier and landmark extractor, no D-EER below 33% can be obtained. This discrepancy in detection results is presumably due to the higher realism and resulting variance of the TLC images available in the used database. This assumption is supported by the fact that the D-EERs on the morphs of FRGC are higher than on the morph of FERET. Consequently, the investigated landmark based algorithms can be considered as not suitable for MAD in realistic scenarios.

IMAGE NOISE PATTERN In this paragraph, the morphing detection capabilities of two different image noise pattern extractors, as described in Section 16.5, are analysed.

The objective of image noise pattern analysis is to detect a change in the image induced by the morphing process based on the noise pattern. Thus, a comparison of the suspected morph to a TLC image is not reasonable, hence only the single image scenario is considered. The D-EERs of the examined algorithms are given in Table 24.10. The two PRNU based approaches described in Section 16.5.1 require no previous training. In order to enable a direct comparison with the SPN based approach, the resulting D-EER of the PRNU based algorithms are listed in a row with the SPN based algorithm trained on morphs generated by FaceMorpher. To avoid repeating the values, the corresponding fields in the row for morphs generated by OpenCV have been left blank. The choice of the classifier used for training the SPN based algorithm was limited to an SVM with RBF kernel, as proposed in [171].

Both, [132] and [171] reported promising detection performances for image noise pattern based MAD algorithms. However, these cannot be reproduced on the realistic data set applied for this evaluation, meaning that none of the applied algorithms is capable of achieving a D-EER lower than 27%, for most constellations the error rates are higher than 40%. Thus, image noise pattern based features may be considered unsuitable for detecting MAD algorithms.

DEEP FEATURES In this paragraph, the morphing detection capabilities of three different deep feature extractors, as described in Section 16.6, are analysed.

The D-EERs of the respective algorithms in a single image scenario are listed in Table 24.11. It is noticeable, that, compared to the MAD algorithms investigated so far, the difference in D-EER between FERET and FRGC is considerably lower. The difference in the detection performance of morphs created with FaceFusion and the UBO Morpher is marginal. It can be concluded, that the extracted features are abstracting from the image source, leading to a higher robustness of the resulting MAD algorithms. It can be observed that ArcFace and Eyedea feature based MAD algorithms exhibit higher performance than FaceNet based algorithms, with best results achieved in combination with an SVM. However, even those algorithms are not able to achieve D-EERs less than 25%, therefore the application of the tested deep features for S-MAD is not recommended.

The D-EERs of the respective algorithms in a differential scenario are listed in Table 24.12. It can be observed that, in contrast to the single image scenario, a greater difference between the detection performance of morphs from FERET and FRGC occurs. Despite the robust feature extraction, the greater variance of the TLC images of FRGC significantly reduces the detection performance. Morphs generated by the UBO Morpher tend to be better detectable than morphs generated by FaceFusion, no clear pattern emerges from the differences in D-EER in dependence of the choice of morphs used for training. In the differential scenario, the ArcFace feature-based MAD algorithms stand out due to their low D-EER, in combination with an SVM, average D-EERs of 2.7% for FRGC and 6.7% for FERET are achieved, representing the lowest error rates of this experiment. With 7.7% average D-EER for FRGC and 16.9% average D-EER for FERET, the Eyedea features in combination with an SVM can be considered suitable as well.

It can be summarized that deep features are particularly suitable for the implementation of differential MAD algorithms. In particular ArcFace and Eyedea features in combination with an SVM with RBF kernel achieve promising D-EERs.

24.2.3 *Discussion*

The MAD algorithms that have been shown to perform best on non post-processed images are listed in the following tables, divided according to single image and differential scenario. In addition to the

Classifier	Training		Test	Feature		
	Database	Morphing Algorithm	Morphing Algorithm	FaceNet	ArcFace	Eyedea
SVM	FERET	FaceMorpher	FaceFusion	32.82%	29.71%	26.35%
			UBO Morpher	33.71%	29.83%	27.64%
		OpenCV	FaceFusion	31.80%	24.84%	24.96%
			UBO Morpher	32.76%	24.87%	25.55%
	FRGC	FaceMorpher	FaceFusion	33.67%	28.23%	27.59%
			UBO Morpher	33.67%	26.46%	30.25%
		OpenCV	FaceFusion	35.06%	26.84%	27.09%
			UBO Morpher	35.32%	25.82%	26.96%
Random Forest	FERET	FaceMorpher	FaceFusion	46.29%	39.38%	31.80%
			UBO Morpher	46.59%	40.37%	32.88%
		OpenCV	FaceFusion	35.56%	39.78%	30.94%
			UBO Morpher	36.86%	36.86%	31.34%
	FRGC	FaceMorpher	FaceFusion	47.09%	42.66%	35.70%
			UBO Morpher	49.62%	38.35%	34.81%
		OpenCV	FaceFusion	33.16%	36.33%	31.01%
			UBO Morpher	28.73%	37.22%	32.66%
AdaBoost	FERET	FaceMorpher	FaceFusion	37.50%	34.45%	29.43%
			UBO Morpher	37.78%	35.50%	30.54%
		OpenCV	FaceFusion	35.99%	31.71%	27.92%
			UBO Morpher	36.80%	31.77%	28.72%
	FRGC	FaceMorpher	FaceFusion	37.09%	37.09%	30.25%
			UBO Morpher	36.71%	36.46%	31.90%
		OpenCV	FaceFusion	36.71%	36.71%	33.54%
			UBO Morpher	36.96%	34.68%	31.90%
Gradient Boosting	FERET	FaceMorpher	FaceFusion	34.14%	34.73%	31.28%
			UBO Morpher	35.38%	35.65%	31.65%
		OpenCV	FaceFusion	32.27%	34.14%	30.45%
			UBO Morpher	35.16%	33.47%	30.42%
	FRGC	FaceMorpher	FaceFusion	36.08%	36.58%	32.78%
			UBO Morpher	36.71%	34.94%	32.41%
		OpenCV	FaceFusion	35.82%	36.20%	32.53%
			UBO Morpher	35.32%	35.19%	30.13%

Table 24.11: Detection performance (D-EER) of deep features in single image scenario

Classifier	Training		Test	Feature		
	Database	Morphing Algorithm	Morphing Algorithm	FaceNet	ArcFace	Eyedea
SVM	FERET	FaceMorpher	FaceFusion	26.32%	7.17%	16.27%
			UBO Morpher	25.24%	6.65%	17.23%
		OpenCV	FaceFusion	26.84%	6.80%	16.39%
			UBO Morpher	24.90%	6.31%	16.83%
	FRGC	FaceMorpher	FaceFusion	14.43%	2.71%	7.34%
			UBO Morpher	13.67%	2.58%	8.35%
		OpenCV	FaceFusion	14.05%	2.71%	7.34%
			UBO Morpher	13.16%	2.71%	7.59%
Random Forest	FERET	FaceMorpher	FaceFusion	30.63%	15.61%	40.15%
			UBO Morpher	30.57%	14.96%	42.16%
		OpenCV	FaceFusion	24.10%	25.94%	37.07%
			UBO Morpher	25.58%	25.26%	38.40%
	FRGC	FaceMorpher	FaceFusion	19.49%	5.41%	11.27%
			UBO Morpher	20.00%	7.09%	11.52%
		OpenCV	FaceFusion	13.67%	6.57%	17.34%
			UBO Morpher	13.42%	5.03%	18.48%
AdaBoost	FERET	FaceMorpher	FaceFusion	32.23%	20.37%	28.44%
			UBO Morpher	32.36%	20.28%	29.40%
		OpenCV	FaceFusion	31.12%	20.31%	27.40%
			UBO Morpher	30.57%	21.08%	28.41%
	FRGC	FaceMorpher	FaceFusion	19.87%	10.70%	19.11%
			UBO Morpher	18.23%	9.66%	17.09%
		OpenCV	FaceFusion	21.27%	10.82%	20.89%
			UBO Morpher	19.24%	10.95%	18.86%
Gradient Boosting	FERET	FaceMorpher	FaceFusion	31.46%	19.10%	28.97%
			UBO Morpher	31.49%	18.24%	30.94%
		OpenCV	FaceFusion	30.97%	18.18%	28.26%
			UBO Morpher	30.66%	17.56%	29.86%
	FRGC	FaceMorpher	FaceFusion	17.59%	8.89%	22.78%
			UBO Morpher	17.34%	9.66%	21.14%
		OpenCV	FaceFusion	16.96%	8.51%	23.67%
			UBO Morpher	16.58%	8.12%	21.27%

Table 24.12: Detection performance (D-EER) of deep features in differential scenario

Training		Test	EER		
Database	Morphing Algorithm	Morphing Algorithm	LBP 4 × 4 3 × 3	BSIF 4 × 4 9 × 9	HOG
			SVM	SVM	SVM
FERET	FaceMorpher	FaceFusion	22.00%	14.33%	14.73%
		UBO Morpher	16.55%	12.54%	13.25%
	OpenCV	FaceFusion	19.69%	15.96%	14.48%
		UBO Morpher	15.87%	13.34%	13.78%
FRGC	FaceMorpher	FaceFusion	31.01%	30.76%	24.05%
		UBO Morpher	26.71%	28.99%	19.75%
	OpenCV	FaceFusion	26.20%	31.01%	23.92%
		UBO Morpher	24.05%	28.61%	20.63%

Table 24.13: Detection performance (D-EER) of selected features in the single image scenario

previously analysed D-EER, the more security-concerned operating points BPCER-10 and BPCER-20 are considered.

SINGLE IMAGE ALGORITHMS The D-EERs of the best single image algorithms are listed in Table 24.13, the corresponding BPCER-10 and BPCER-20 are given in Table 24.14 and 24.15 respectively. Among the best performing single image algorithms only those using an SVM with RBF kernel for classification can be found.

In general, it can be observed that the morphs of FERET are much more difficult to detect than the morphs of FRGC. HOG is the most robust algorithm, meaning the D-EER is consistently below 25%, regardless of the database and morphing algorithm. Even in the security-concerned BPCER-20 scenario, 70% of the bona fide comparisons are accepted for FRGC and almost half for FERET. Across all single image algorithms, it can also be observed that the error rates for detecting morphs of the FaceFusion morphing algorithm are higher than those for detecting the UBO Morpher morphs.

The corresponding DET plots are shown in Figure 24.1. In order to enable an organised visualisation, only DET curves for algorithms trained on OpenCV are depicted. It can be observed, that the algorithm with the lowest D-EER performs best over all other operating points. Only in the evaluation of morphs generated by the UBO Morpher providers on the FRGC database there are overlaps between BSIF and HOG.

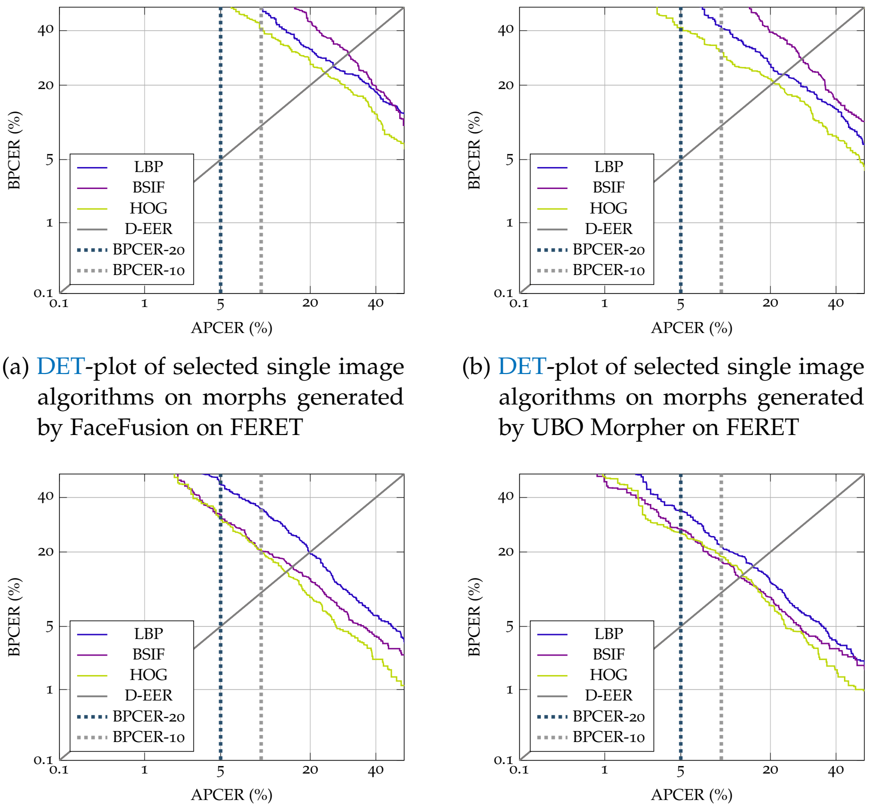
DIFFERENTIAL ALGORITHMS The D-EERs of the best performing differential algorithms are listed in Table 24.16, the corresponding BPCER-10 and BPCER-20 are given in Table 24.17 and 24.18 respectively.

Training		Test	BPCER-10		
Database	Morphing Algorithm	Morphing Algorithm	LBP 4×4 3×3	BSIF 4×4 9×9	HOG
			SVM	SVM	SVM
FERET	FaceMorpher	FaceFusion	41.34%	20.20%	21.38%
		UBO Morpher	25.30%	14.19%	17.50%
	OpenCV	FaceFusion	35.46%	20.44%	20.84%
		UBO Morpher	21.96%	17.29%	18.74%
FRGC	FaceMorpher	FaceFusion	52.91%	62.53%	42.91%
		UBO Morpher	44.68%	61.27%	33.80%
	OpenCV	FaceFusion	49.11%	62.91%	41.01%
		UBO Morpher	41.14%	58.23%	31.52%

Table 24.14: Detection performance (BPCER.10) of selected features in the single image scenario

Training		Test	BPCER-20		
Database	Morphing Algorithm	Morphing Algorithm	LBP 4×4 3×3	BSIF 4×4 9×9	HOG
			SVM	SVM	SVM
FERET	FaceMorpher	FaceFusion	52.17%	28.33%	31.48%
		UBO Morpher	37.85%	23.32%	25.66%
	OpenCV	FaceFusion	45.62%	33.00%	31.48%
		UBO Morpher	34.67%	27.60%	26.51%
FRGC	FaceMorpher	FaceFusion	69.11%	79.24%	56.20%
		UBO Morpher	55.70%	72.66%	46.71%
	OpenCV	FaceFusion	66.08%	75.44%	54.81%
		UBO Morpher	55.95%	70.89%	41.27%

Table 24.15: Detection performance (BPCER-20) of selected features in the single image scenario



(a) DET-plot of selected single image algorithms on morphs generated by FaceFusion on FERET

(b) DET-plot of selected single image algorithms on morphs generated by UBO Morpher on FERET

(c) DET-plot of selected single image algorithms on morphs generated by FaceFusion on FRGC

(d) DET-plot of selected single image algorithms on morphs generated by UBO Morpher on FRGC

Figure 24.1: DET-plots of selected single image algorithms

Most MAD algorithms based on handcrafted features, for example texture or gradients, achieve better results in the single image scenario than in the differential scenario. This is caused by the fact, that the probe images used for comparison contain such a high variance, that the additional information available does not contribute to a stabilisation of the detection. This also provides an explanation, why the results obtained on FERET on a differential scenario are usually better than those obtained on FRGC, since the probe images contained in the FRGC exhibit a significantly higher variance than those of FERET. For the evaluation of more security-conscious operating points, it should be noted that if an algorithm achieves a D-EER lower than the required APCER, the corresponding BPCER will also drop below the D-EER. Consequently, especially for the ArcFace based algorithm, the BPCER-10 and BPCER-20 for the evaluation of the FERET data are significantly reduced.

The corresponding DET plots are shown in Figure 24.2. The massive performance advantage of the ArcFace based MAD algorithm over the other MAD algorithms is clearly visible in these examples. Regardless of the scenario and operating point, it achieves the lowest error rates.

Training		Test	EER			
Database	Morphing Algorithm	Morphing Algorithm	LBP 4 × 4 3 × 3	HOG	ArcFace	Eyedea
			AdaBoost	SVM	SVM	SVM
FERET	FaceMorpher	FaceFusion	27.92%	27.52%	7.17%	16.27%
		UBO Morpher	24.56%	25.24%	6.65%	17.23%
	OpenCV	FaceFusion	27.33%	26.41%	6.80%	16.39%
		UBO Morpher	25.08%	24.31%	6.31%	16.83%
FRGC	FaceMorpher	FaceFusion	24.30%	19.37%	2.71%	7.34%
		UBO Morpher	19.62%	15.70%	2.58%	8.35%
	OpenCV	FaceFusion	22.41%	18.73%	2.71%	7.34%
		UBO Morpher	19.11%	15.70%	2.71%	7.59%

Table 24.16: Detection performance (D-EER) of selected features in the differential scenario

Training		Test	BPCER-10			
Database	Morphing Algorithm	Morphing Algorithm	LBP 4 × 4 3 × 3	HOG	ArcFace	Eyedea
			AdaBoost	SVM	SVM	SVM
FERET	FaceMorpher	FaceFusion	57.54%	57.11%	4.59%	24.99%
		UBO Morpher	50.17%	51.84%	4.29%	26.30%
	OpenCV	FaceFusion	54.26%	55.08%	4.02%	23.96%
		UBO Morpher	51.93%	50.08%	3.68%	24.20%
FRGC	FaceMorpher	FaceFusion	54.43%	28.35%	1.16%	5.70%
		UBO Morpher	41.14%	23.29%	1.16%	7.34%
	OpenCV	FaceFusion	40.38%	30.38%	0.90%	5.32%
		UBO Morpher	31.90%	22.28%	0.90%	6.71%

Table 24.17: Detection performance (BPCER-10) of selected features in the differential scenario

Training		Test	BPCER-20			
Database	Morphing Algorithm	Morphing Algorithm	LBP 4 × 4 3 × 3	HOG	ArcFace	Eyedea
			AdaBoost	SVM	SVM	SVM
FERET	FaceMorpher	FaceFusion	74.55%	73.92%	10.83%	38.91%
		UBO Morpher	66.67%	68.30%	9.19%	39.52%
	OpenCV	FaceFusion	68.52%	72.52%	8.82%	37.22%
		UBO Morpher	68.24%	66.33%	8.28%	37.58%
FRGC	FaceMorpher	FaceFusion	70.25%	40.89%	1.80%	9.87%
		UBO Morpher	59.49%	34.56%	1.80%	11.27%
	OpenCV	FaceFusion	56.08%	43.29%	1.68%	9.11%
		UBO Morpher	51.01%	36.84%	1.42%	10.63%

Table 24.18: Detection performance (BPCER-20) of selected features in the differential scenario

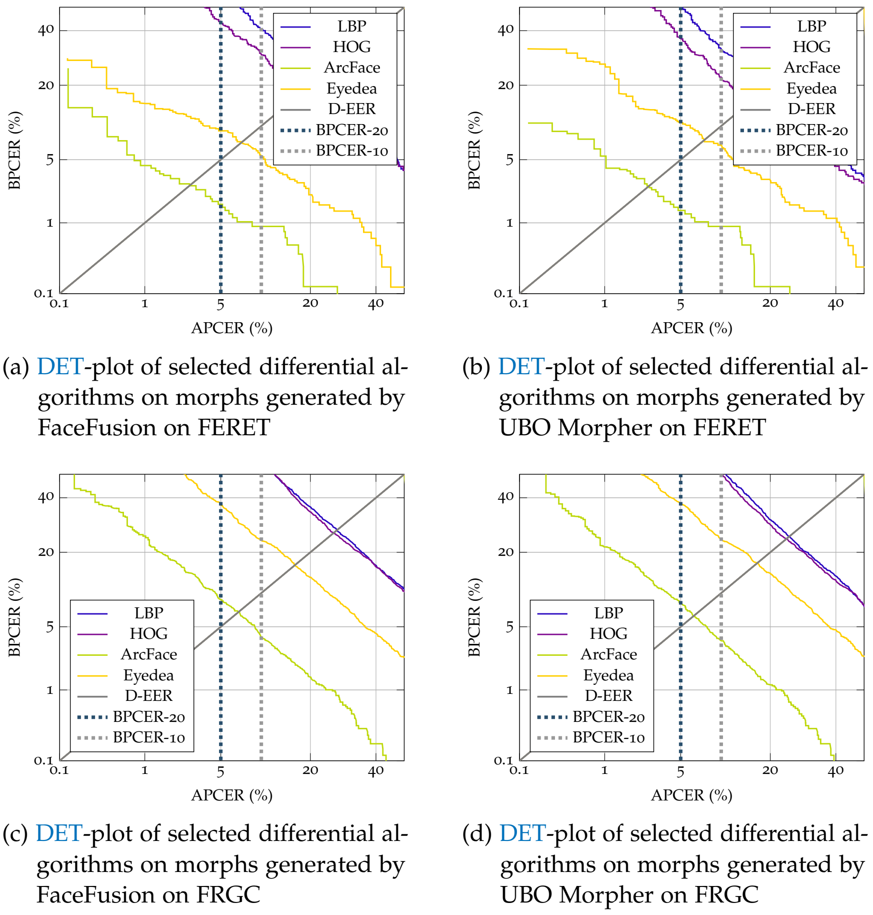
In particular considering the graphs of the FERET database, it should be noted that already at an APCER of 20% almost no BPCER is measurable. This behaviour can be interpreted in a way, that the system can be configured such that when deployed in a FRS the FNMR of the overall system would not increase, but 80% of the morphs could be recognized.

24.3 EXPERIMENT 3 - POST-PROCESSING

The selection of suitable feature extractors and classifiers in Section 24.2 is carried out on a database with realistic variance regarding the acquisition parameters of the images, for example pose and illumination. However, in a real scenario, passport photographs may be processed differently prior to being included in the passport. The impact of different post-processing steps on the detection performance of the MAD algorithms has already been shown, for example, for print and scan [134]. In this experiment the influence of three different post-processing chains on the detection performance of the MAD algorithms determined in Section 24.2 will be investigated.

24.3.1 *Experimental Setup*

The three scenarios considered in this thesis and the associated post-processing chains are described in Section 21.3. The MAD algorithms for the single image and differential scenario, as determined in Section 24.2, are tested and evaluated with respect to their generalisation capability against these post-processing chains. For this purpose, the algorithms are trained on non post-processing images, morphs created



(a) DET-plot of selected differential algorithms on morphs generated by FaceFusion on FERET

(b) DET-plot of selected differential algorithms on morphs generated by UBO Morpher on FERET

(c) DET-plot of selected differential algorithms on morphs generated by FaceFusion on FRGC

(d) DET-plot of selected differential algorithms on morphs generated by UBO Morpher on FRGC

Figure 24.2: DET-plots of selected differential algorithms

by OpenCV and FaceMorpher are used. The evaluation on morphs created by FaceFusion and UBO Morpher is reported in the following.

24.3.2 *Evaluation*

The evaluation is divided into the three post-processing scenarios *RS*, *JP* and *PS*. Per scenario the D-EER is reported, and a visualisation of the performance over all operating points is given as DET-plot. In order to achieve a comparability to the previous DET-plots, the performance of algorithms trained on morphs generated by OpenCV are depicted.

RESIZED The post-processing in the *RS* scenario corresponds to the digital image transmission in the passport application. The image has been resized to a size that complies with the minimum resolution requirements for passport photos, as described in Section 20.1.3.

The error rates of the selected S-MAD algorithms are listed as D-EER in Table 24.19. Resizing to half the image size has almost no effect on

Database	Morphing Algorithm		Algorithm/Classifier		
	Training	Test	LBP 4 × 4 3 × 3	BSIF 4 × 4 9 × 9	HOG
			SVM	SVM	SVM
FRGC	FaceMorpher	FaceFusion	23.17%	14.36%	14.79%
		UBO Morpher	16.86%	12.76%	13.16%
	OpenCV	FaceFusion	20.71%	15.50%	14.45%
		UBO Morpher	15.29%	13.41%	14.02%
FERET	FaceMorpher	FaceFusion	31.52%	30.76%	23.92%
		UBO Morpher	26.84%	29.11%	19.87%
	OpenCV	FaceFusion	25.44%	31.39%	24.05%
		UBO Morpher	23.42%	28.48%	20.63%

Table 24.19: Detection performance (D-EER) of selected S-MAD algorithms on images post-processed according *RS*

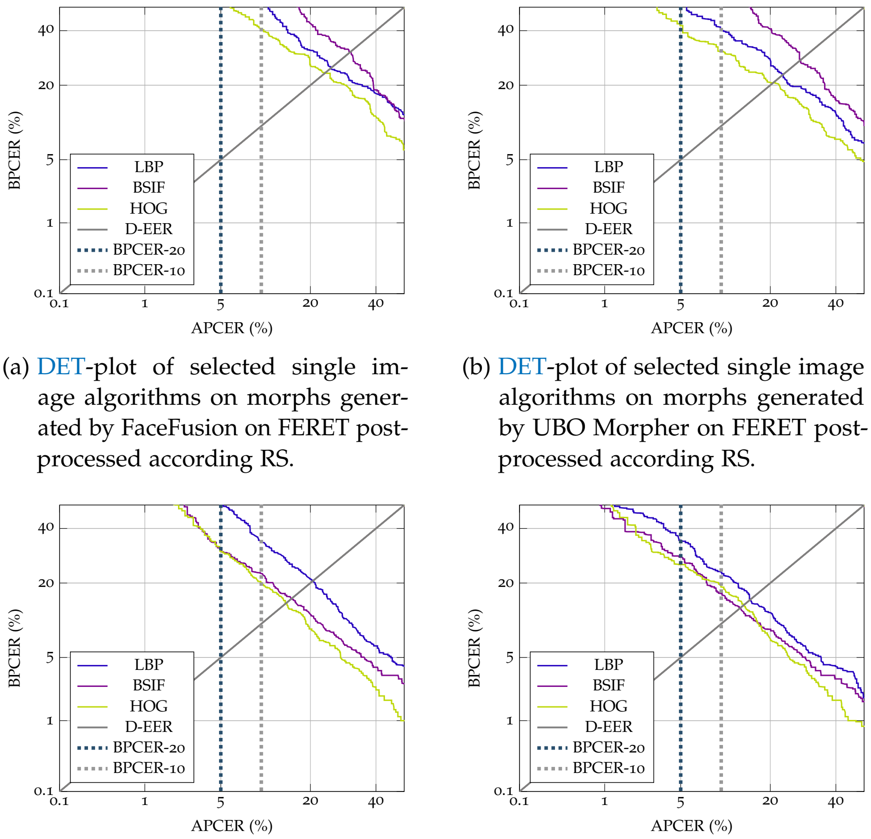
the performance of all three algorithms, the D-EERs only change in the second decimal place.

The corresponding DET plots are depicted in Figure 24.3. It can be observed, that post-processing according the *RS* scenario has no effect on the evaluation of further operating points. Thus, it can be concluded, that the selected S-MAD algorithms are able to generalize very well in case of a resizing of the image to a size permitted for passport images, meaning, no performance decrease of the algorithms is to be expected.

The D-EERs for the differential scenario are listed in Table 24.20. Again, it can be observed, that post-processing according the *RS* scenario barely influences the detection performance of the selected MAD systems. In cases of particularly low D-EER, e.g. ArcFace on morphs of the FERET database generated by FaceFusion, the influence on the detection performance tends to be slightly higher than in cases of higher baseline D-EER.

The corresponding DET-plots are depicted in Figure 24.4. It can be observed here that the detection performance over all operating points is not influenced by the applied post-processing. Consequently, the selected differential MAD algorithms can also be considered robust against post-processing according *RS*.

jpeg2000 The post-processing in the following scenario corresponds to a face image, digitally transferred to the application office and stored into the passport. The images resized to half the size are subsequently compressed with JPEG2000 to 15kb, resulting in a loss of sharpness and high-frequency information. As described in Section 21.3 this post-processing is abbreviated as *JP*.



(a) DET-plot of selected single image algorithms on morphs generated by FaceFusion on FERET post-processed according RS.

(b) DET-plot of selected single image algorithms on morphs generated by UBO Morpher on FERET post-processed according RS.

(c) DET-plot of selected single image algorithms on morphs generated by FaceFusion on FRGC post-processed according RS.

(d) DET-plot of selected single image algorithms on morphs generated by UBO Morpher on FRGC post-processed according RS.g

Figure 24.3: DET-plots of selected single image algorithms post-processed according *RS*

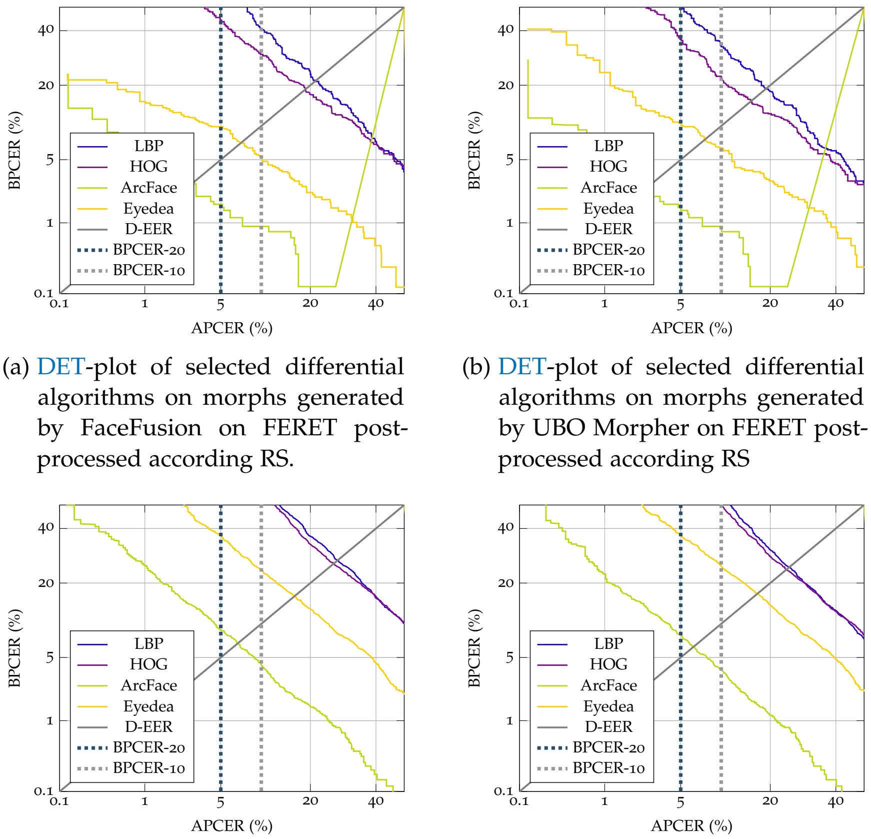
The error rates of the selected S-MAD algorithms on image post-processed according *JP* are listed as D-EER in Table 24.19. The loss of information caused by the *JP* process has a significant impact on the detection performance of the LBP based MAD algorithm. Regardless of the scenario, no D-EER below 39% is obtained, whereas the error rates for the morphs of the FERET database, which are harder to detect, with almost 50% D-EER are close to a random guess. The BSIF based MAD algorithm, also processing texture information, shows a significantly higher robustness regarding *JP* post-processing, with an increase of roughly 4 percent points D-EER for FRGC. It is noticeable that the D-EERs for FERET, which was already significantly higher on the non post-processed images, hardly changes for the post-processed images. This leads to the conclusion that the features employed for classification of FERET by the BSIF based algorithm are not affected by JPEG2000 compression. The different behaviour of the LBP and BSIF based algorithms can be attributed to the fact, that the LBP

Database	Morphing Algorithm		Algorithm/Classifier			
	Training	Test	LBP 4×4 3×3	HOG	ArcFace	Eyedea
			AdaBoost	SVM	SVM	SVM
FRGC	FaceMorpher	FaceFusion	28.04%	27.40%	7.20%	16.24%
		UBO Morpher	24.25%	25.08%	6.71%	17.07%
	OpenCV	FaceFusion	27.43%	26.41%	6.77%	16.33%
		UBO Morpher	25.05%	24.25%	6.37%	17.01%
FERET	FaceMorpher	FaceFusion	24.56%	18.35%	2.84%	7.47%
		UBO Morpher	21.27%	15.06%	2.84%	8.35%
	OpenCV	FaceFusion	21.39%	18.61%	2.96%	7.22%
		UBO Morpher	18.86%	15.44%	2.58%	7.72%

Table 24.20: Detection performance (D-EER) of selected differential MAD algorithms on images post-processed according *RS*

Database	Morphing Algorithm		Algorithm/Classifier		
	Training	Test	LBP 4×4 3×3	BSIF 4×4 9×9	HOG
			SVM	SVM	SVM
FRGC	FaceMorpher	FaceFusion	41.14%	18.06%	19.41%
		UBO Morpher	40.22%	16.33%	19.11%
	OpenCV	FaceFusion	40.71%	18.18%	20.18%
		UBO Morpher	39.82%	16.83%	19.32%
FERET	FaceMorpher	FaceFusion	49.24%	31.14%	28.23%
		UBO Morpher	48.86%	30.51%	27.34%
	OpenCV	FaceFusion	43.42%	30.76%	31.90%
		UBO Morpher	43.16%	29.75%	31.01%

Table 24.21: Detection performance (D-EER) of selected S-MAD algorithms on images post-processed according *JP*



(a) DET-plot of selected differential algorithms on morphs generated by FaceFusion on FERET post-processed according RS.

(b) DET-plot of selected differential algorithms on morphs generated by UBO Morpher on FERET post-processed according RS

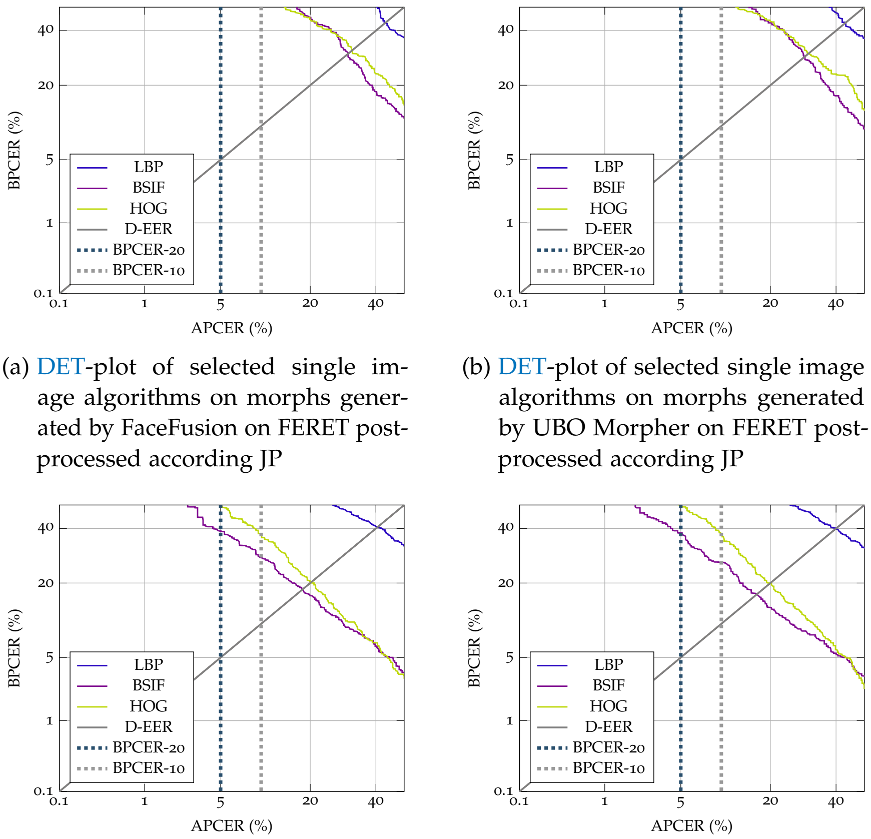
(c) DET-plot of selected differential algorithms on morphs generated by FaceFusion on FRGC post-processed according RS

(d) DET-plot of selected differential algorithms on morphs generated by UBO Morpher on FRGC post-processed according RS

Figure 24.4: DET-plots of selected differential algorithms post-processed according *RS*

based algorithm applies smaller filters. This means, that mainly high-frequency information is processed, which is particularly influenced by the lossy compression of JPEG2000. The HOG based algorithm, which, for FERET images, provides a significantly better performance on non post-processed images than the BSIF based algorithm, is more affected by the compression, causing a close of the gap between the D-EERs and the BSIF based algorithm.

The corresponding DET-plots are shown in Figure 24.5. It can be observed that the DET-plot for the LBP based algorithm is consistently far to the upper right edge of the plot. Regardless of the constellation, no meaningful classification can be expected from this algorithm. For the HOG and the BSIF based algorithm, it should be noted, that the DET plots are shifted uniformly to the upper right corner, meaning that all operating points are affected more or less equally by the post-processing.



(a) DET-plot of selected single image algorithms on morphs generated by FaceFusion on FERET post-processed according JP

(b) DET-plot of selected single image algorithms on morphs generated by UBO Morpher on FERET post-processed according JP

(c) DET-plot of selected single image algorithms on morphs generated by FaceFusion on FRGC post-processed according JP

(d) DET-plot of selected single image algorithms on morphs generated by UBO Morpher on FRGC post-processed according JP

Figure 24.5: DET-plots of selected single image algorithms post-processed according *JP*

The error rates of the selected differential MAD algorithms on image post-processed according *JP* are listed as D-EER in Table 24.19. In the differential scenario the LBP based algorithm is significantly influenced by the applied post-processing, but considerably less than in the single image scenario. However, no D-EER below 30% is obtained on the post-processed images. It should be noted that, in contrast to the previous behaviour in the differential scenario, the detection of morphs of the FERET database is harder compared to those of the FRGC. A similar effect is observed for the HOG based MAD algorithm. The performance degradation, caused by the post-processing, on images of the FERET database is significantly higher, resulting in a reduction of the performance difference in the detection of both databases to one percentage point D-EER. The performance of the deep features based MAD algorithms is nearly not affected by the post-processing, emphasising the robustness of the applied feature extraction against compression of the analysed images.

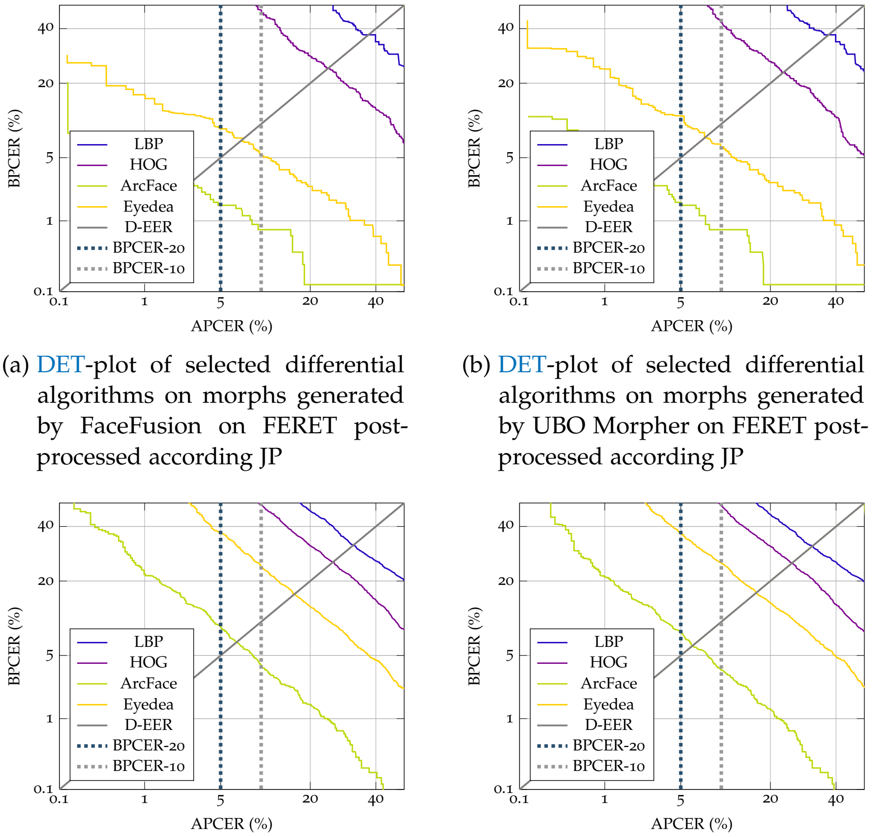
Database	Morphing Algorithm		Algorithm/Classifier			
	Training	Test	LBP 4×4 3×3	HOG	ArcFace	Eyedea
			AdaBoost	SVM	SVM	SVM
FRGC	FaceMorpher	FaceFusion	33.59%	26.59%	7.30%	16.24%
		UBO Morpher	33.07%	26.04%	6.65%	17.29%
	OpenCV	FaceFusion	32.60%	26.16%	6.68%	16.27%
		UBO Morpher	32.08%	25.92%	6.18%	16.61%
FERET	FaceMorpher	FaceFusion	39.75%	22.03%	2.96%	7.59%
		UBO Morpher	38.23%	21.52%	3.09%	8.23%
	OpenCV	FaceFusion	36.08%	24.68%	2.71%	7.34%
		UBO Morpher	35.95%	23.42%	2.71%	7.59%

Table 24.22: Detection performance (D-EER) of selected differential MAD algorithms on images post-processed according *JP*

The corresponding DET-plots are shown in Figure 24.6. It can be observed, that the DET plots of the LBP and the HOG based algorithm are shifted linearly over all operating points. As indicated by the D-EER, the plots of the LBP based algorithm are impacted more than those of the HOG based algorithm. The DET-plots of the deep feature based algorithms are almost unaffected, indicating robustness across all operating points.

print/scan - jpeg2000 The last post-processing scenario investigated, namely *Print/Scan - JPEG2000*, abbreviated *PS*, is intended to reflect the scenario of an analogous delivered passport image stored in the passport. Thus, the post-processing corresponds to the quality that can be expected, e.g. from German passports at the passport check. The difference to the previous scenario is that the passport image is printed and scanned with 300dpi prior to resizing and compression, resulting in a further reduction of the contained information. Details about the process are provided in Section 21.3.

The error rates of the selected S-MAD algorithms on image post-processed according *PS* are listed as D-EER in Table 24.23. Compared to *JP* the error rates of the LBP based algorithm do not change significantly. However, since the LBP algorithm already performs in many situations close to a random guess, this behaviour could be expected. The BSIF based algorithm, on the other hand, is massively influenced by the further post-processing step. Compared to the performance on images post-processed with *JP*, the performance degradation for images of the FRGC database is doubled. Furthermore, the D-EERs determined on the FERET database, which are barely influenced by *JP* post-processing, increase by 10 percentage points. For FRGC, the HOG based algorithm is as nearly as performant as for the *JP* post-



(a) DET-plot of selected differential algorithms on morphs generated by FaceFusion on FERET post-processed according JP

(b) DET-plot of selected differential algorithms on morphs generated by UBO Morpher on FERET post-processed according JP

(c) DET-plot of selected differential algorithms on morphs generated by FaceFusion on FRGC post-processed according JP

(d) DET-plot of selected differential algorithms on morphs generated by UBO Morpher on FRGC post-processed according JP

Figure 24.6: DET-plots of selected differential algorithms post-processed according *JP*

processing, for FERET, however, the impact increases. In contrast to the other evaluation scenarios, the morphing algorithm utilised for the generation of the Morphs gains in importance, resulting in a difference of up to 15 percent points D-EER. However, as these scenarios are at D-EERs above 30% this effect can be neglected for the evaluation.

The corresponding DET-plots are depicted in Figure 24.7. In particular in Figure 24.7a, the strong post-processing induced performance degradation across all algorithms is evident. The HOG based algorithm returns poorer results than a random guess, which is why the plot is out of scale and no longer displayed. From the poor overall performance of the algorithms it can be concluded that the examined single image algorithms are not suitable for MAD of images post-processed according *PS*.

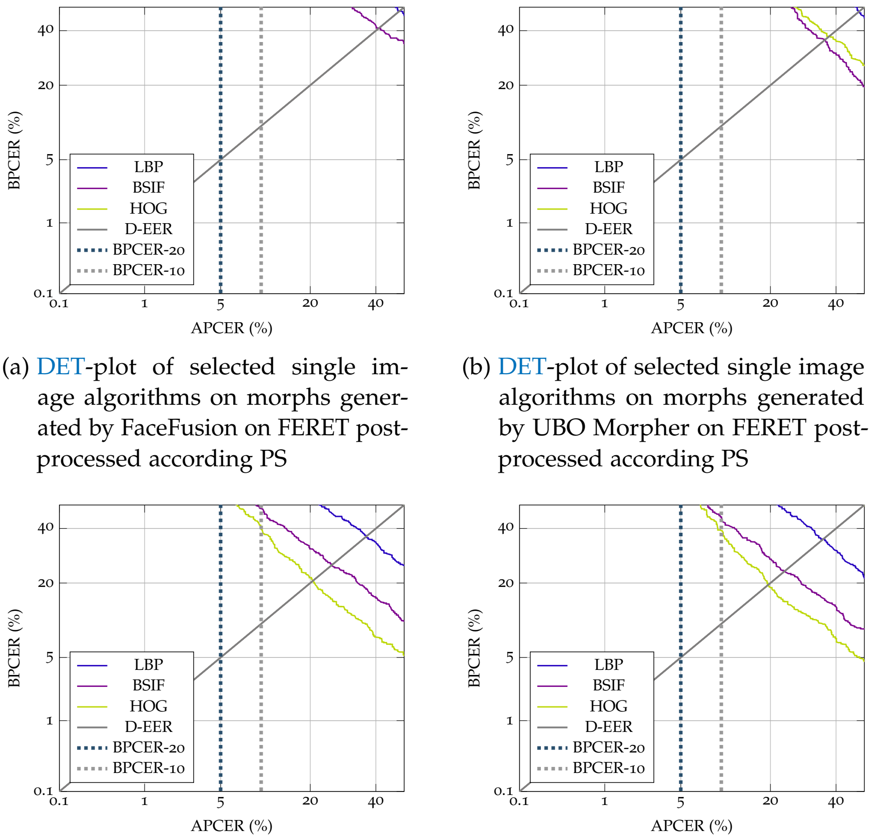
The error rates of the selected differential MAD algorithms on image post-processed according *PS* are listed as D-EER in Table 24.24. The performance of the LBP based algorithm is decreased compared to the

Database	Morphing Algorithm		Algorithm/Classifier		
	Training	Test	LBP 4 × 4 3 × 3	BSIF 4 × 4 9 × 9	HOG
			SVM	SVM	SVM
FRGC	FaceMorpher	FaceFusion	39.35%	26.66%	20.86%
		UBO Morpher	37.13%	23.42%	19.97%
	OpenCV	FaceFusion	36.67%	25.92%	20.55%
		UBO Morpher	35.65%	23.82%	19.29%
FERET	FaceMorpher	FaceFusion	56.96%	40.25%	34.05%
		UBO Morpher	53.16%	35.44%	36.58%
	OpenCV	FaceFusion	48.23%	41.14%	53.42%
		UBO Morpher	47.85%	36.20%	37.72%

Table 24.23: Detection performance (D-EER) of selected S-MAD algorithms on images post-processed according *PS*

Database	Morphing Algorithm		Algorithm/Classifier			
	Training	Test	LBP 4 × 4 3 × 3	HOG	ArcFace	Eyedea
			AdaBoost	SVM	SVM	SVM
FRGC	FaceMorpher	FaceFusion	40.09%	30.14%	7.76%	16.89%
		UBO Morpher	40.46%	29.58%	7.05%	17.60%
	OpenCV	FaceFusion	38.21%	29.52%	7.20%	16.83%
		UBO Morpher	37.13%	28.72%	6.71%	17.32%
FERET	FaceMorpher	FaceFusion	39.37%	42.28%	1.29%	8.35%
		UBO Morpher	43.54%	33.92%	3.22%	9.75%
	OpenCV	FaceFusion	38.35%	53.29%	1.42%	7.59%
		UBO Morpher	41.14%	34.05%	3.09%	8.61%

Table 24.24: Detection performance (D-EER) of selected differential MAD algorithms on images post-processed according *PS*



(a) DET-plot of selected single image algorithms on morphs generated by FaceFusion on FERET post-processed according PS

(b) DET-plot of selected single image algorithms on morphs generated by UBO Morpher on FERET post-processed according PS

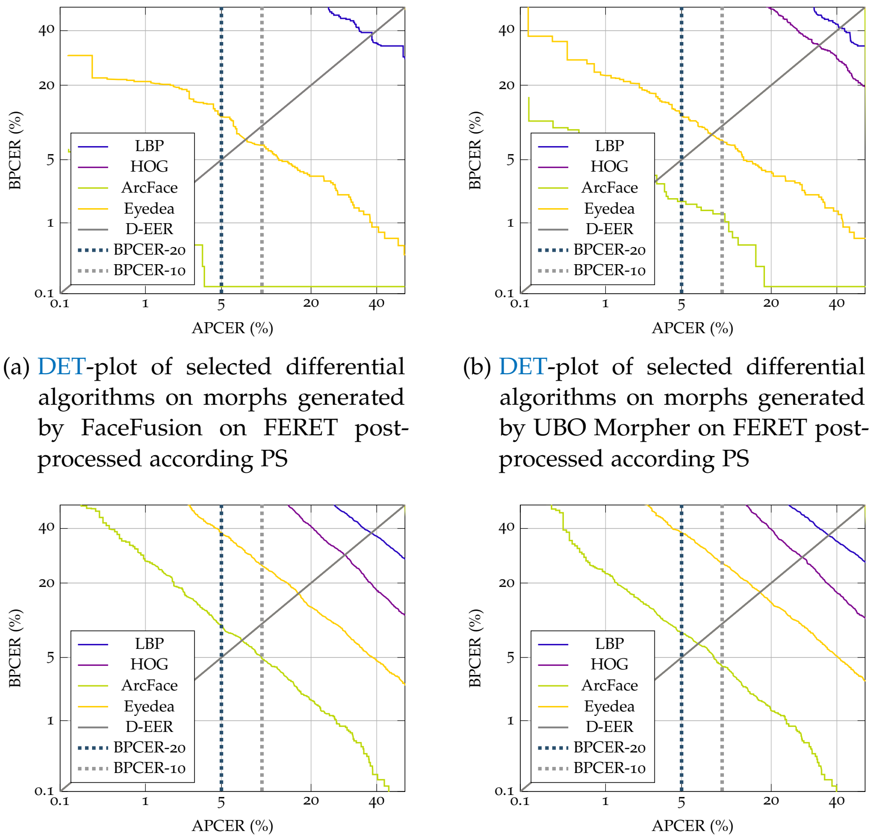
(c) DET-plot of selected single image algorithms on morphs generated by FaceFusion on FRGC post-processed according PS

(d) DET-plot of selected single image algorithms on morphs generated by UBO Morpher on FRGC post-processed according PS

Figure 24.7: DET-plots of selected single on images post-processed according *PS*

JP post-processing, resulting in D-EERs above 40%. This leads to the conclusion, that most relevant texture information is lost due to the *PS* post-processing. The performance of the HOG based algorithm is also impacted significantly. Whereas a D-EER of roughly 30% can still be achieved on the images of the FRGC database, the D-EER of the evaluation on the FRGC are partly above 50%. All the more remarkable is the fact, that the deep features based algorithms deliver consistently good detection rates despite the striking *PS* post-processing. In most constellations the change in D-EER is below one percent point, partially the performance is even increased.

The corresponding DET-plots are shown in Figure 24.8. As for the single image scenario, the plot of the HOG based algorithm is not displayed in Figure 24.8a, as the error exceeds the scale. In addition, in all other constellations, the plots of the hand crafted features are in the upper right corner, indicating that these algorithms are not suitable for MAD on images post-processed according to *PS*. The



(a) DET-plot of selected differential algorithms on morphs generated by FaceFusion on FERET post-processed according PS

(b) DET-plot of selected differential algorithms on morphs generated by UBO Morpher on FERET post-processed according PS

(c) DET-plot of selected differential algorithms on morphs generated by FaceFusion on FRGC post-processed according PS

(d) DET-plot of selected differential algorithms on morphs generated by UBO Morpher on FRGC post-processed according PS

Figure 24.8: DET-plots of selected differential algorithms on images post-processed according *PS*

deep features based algorithms show a consistent DET curve, allowing the algorithms to be used in a wide range of operating points. It is noticeable, that the performance of ArcFace on morphs generated by FaceFusion within the FERET database is improved by the post-processing. It can be assumed that the post-processing enhances the distinguishability of the extracted features, but as the functioning of the DNN is not comprehensible, no explicit conclusion can be drawn.

24.3.3 *Discussion*

The resizing of the images has no influence on the detection performance of the examined MAD algorithms. The selected algorithms can therefore be considered independent of the image size in the range of allowed sizes for passport images. This is due to the fact, that only less high-frequency information is lost by resizing. Since especially the algorithms based on hand crafted feature extractors are trained on down-scaled image sections, the influence of resizing is minimized. Regarding the post-processings reducing more information, different behaviour of the algorithms can be observed. For example, the LBP based algorithms cannot reliably separate morphs and bona fide images after *JP* post-processing. The performance of the HOG based S-MAD algorithm, for example, is reduced by the *JP* post-processing on images of the FERET database, but is hardly affected by the *PS* post-processing. The HOG based differential MAD algorithm, on the other hand, gets progressively weaker the more destructive the post-processing is.

In general it can be observed, that hand crafted features perform well in the single image scenario on non post-processed images, however, post-processing (especially *PS*) usually reduces the performance to such an extent, that they can no longer be applied for MAD.

Among the examined algorithms, only the deep feature based ones demonstrate a constant performance over all operating points and post-processings. This can be attributed to the fact, that those feature extractors have been trained to operate in a robust FRSs. In this case, the features of the face should be extracted independently from other image properties. Thus, influences like artefacts added by compression or printing and scanning are isolated. The outstanding performance of the ArcFace based algorithm can be attributed to the modified loss function used during the training of the ArcFace feature extractor. As described in Section 16.6.2 it implements a margin penalty in order to increase inter-class discrepancy and intra-class compactness.

24.4 EXPERIMENT 4 - ALGORITHM FUSION

In [135] it was shown that a score level fusion of different MAD algorithms may improve performance and robustness of the resulting algorithm. In this experiment we investigate if this observation is applicable to the algorithms determined in Section 24.2.

24.4.1 *Experimental Setup*

The selected algorithms allow for an enormous number of combinations for the implementation of a score level fusion. The experimental setup aims to reduce these combinations to a reasonable number. For the single image and the differential scenario all possible combinations of two of the algorithms selected in Section 24.2 are created. The weights of the fusion are determined on non post-processed images, a grid search is performed with an accuracy of 5%, in order to obtain the optimal weights. Subsequently, the gain in robustness and performance on images post-processed according to *JP* and *PS* is evaluated. The *RS* scenario is deliberately excluded, since it was shown

Database	Post-Processing	Algorithms		
		LBP 4 × 4 3 × 3	LBP 4 × 4 3 × 3	BSIF 4 × 4 9 × 9
		BSIF 4 × 4 9 × 9	HOG	HOG
Weight		10/90	10/90	0/100
FERET	NPP	24.94%	23.42%	23.92%
	JP	31.51%	32.53%	31.90%
	PS	42.03%	50.25%	53.41%
Weight		40/60	35/65	35/65
FRGC	NPP	12.63%	12.51%	11.62%
	JP	24.50%	20.62%	17.41%
	PS	26.78%	20.25%	19.51%

Table 24.25: Detection performance (D-EER) and robustness of fused single image algorithms

in Section 24.3 that it has no significant influence on the detection performance of the algorithms.

As it has already been shown, that the images used for training only take a subordinate role, the training of the algorithms is limited to the morphs created with OpenCV. The evaluation is carried out on morphs created with FaceFusion, as it has been demonstrated that these are the most difficult to detect, thus representing the upper limit for the expected error rate.

24.4.2 *Evaluation*

The evaluation is split into two tables, single image and differential scenario. Table 24.25 shows the fusion results for the single image scenario. In the header of the table the pairs of the fused MAD algorithms are listed. For each database the fusion weights determined on the non post-processed data are stated. It should be noted that the weights determined on FRGC assign more equal importance to both algorithms than those of FERET. In this case, the HOG based algorithm is preferred, the BSIF based algorithm outweighs the LBP based one. Due to this imbalance, the fusion does not show a great influence on robustness, resulting in a significant performance degradation on post-processed images, as observed for single algorithms. The results obtained on FRGC show an improvement of performance on non post-processed images caused by the fusion. The algorithm based on HOG is the best single algorithm in this constellation and achieves a D-EER of 14.48%. A fusion with the BSIF based algorithm can reduce the D-EER to 11.62%. For post-processed images with *PS*,

Database	Post-Processing	Algorithms					
		LBP 4 × 4 3 × 3	LBP 4 × 4 3 × 3	LBP 4 × 4 3 × 3	HOG	HOG	ArcFace
		HOG	ArcFace	Eyedea	ArcFace	Eyedea	Eyedea
Weight		90/10	10/90	35/65	0/100	5/95	100/0
FERET	NPP	15.94%	1.93%	5.95%	2.70%	6.84%	2.70%
	JP	24.94%	2.58%	7.22%	2.71%	5.70%	2.71%
	PS	51.65%	15.18%	7.47%	1.42%	9.49%	1.42%
Weight		95/5	75/25	65/35	5/95	5/95	95/5
FRGC	NPP	22.13%	5.47%	13.34%	6.67%	16.15%	6.80%
	JP	29.80%	5.90%	15.04%	6.65%	16.02%	6.74%
	PS	34.95%	7.26%	16.43%	7.17%	16.89%	7.33%

Table 24.26: Detection performance (D-EER) and robustness of fused differential algorithms

however, the synergy resulting from the fusion is reduced, the single algorithm reaches a D-EER of 20.55%, the fusion algorithm 19.51%.

The results for the fusion of the differential algorithms are given in Table 24.26. The choice of weights indicates a clear trend towards deep feature based algorithms, with the ArcFace based algorithm being preferred to the Eyedea based one. On the FERET database the only combination capable of achieving a better result on the non post-processed data than the single ArcFace based algorithm is a 10% fusion with the LBP based algorithm. The D-EER can be reduced from 2.7% to 1.93%. However, if the fused algorithm is evaluated on post-processed images with *PS*, the D-EER increases to 15.15%, whereas the D-EER of the single ArcFace based algorithm remains at a low value of 1.42%. The weights determined on the FRGC show a slightly lower preference for the ArcFace based algorithm. In this case, a fusion with the LBP or HOG based algorithm, as well as a fusion with the Eyedea based algorithm, allow for a slight improvement of the D-EER on the non post-processed images. However, for most fusions, the D-EER on post-process images falls below that of the simple ArcFace based algorithm. An exception is the 5% fusion with the HOG based algorithm, which shows a minimal better performance than the single algorithm over all scenarios.

24.4.3 *Discussion*

Generally it can be stated, that a fusion can lead to a performance improvement of the individual algorithms on non post-processed images. However, this performance gain is usually not observable for evaluation on post-process images. In most cases the best individual algorithms achieve better results than the fusion. In addition, no scheme that applies to both databases can be identified for any fusion.

Thus, for the investigated MAD algorithms, a score level fusion is not considered useful. In the single image scenario the HOG based algorithm dominates, in the differential scenario the ArcFace based algorithm.

25 SUMMARY

The evaluation can be divided into two chapters: the evaluation of the vulnerability of existing FRS and the evaluation of the detection performance of MAD systems.

In the initial step, the threat potential of the created database is determined. For this purpose, in Chapter 23 the performance of two commercial (ArcFace and COTS), as well as two open source (FaceNet and ArcFace) FRSs is determined on the bona fide images of the database. Afterwards, the number of falsely accepted attacks is measured. The vulnerability of the systems is reported by means of MMPMR and RMMR. It can be observed that FRS, which obtain a higher generalisation capability and thus achieve a higher performance on bona fide data, are more vulnerable to morphs than FRS with a lower recognition performance on bona fide images.

The second step is to evaluate the detection performance of the MAD algorithms described in Chapter 16 and Chapter 18. Due to the large number of possible combinations, an all-encompassing evaluation is not feasible. Thus, four experiments are performed in order to determine suitable and robust MAD algorithms.

In the first experiment, the influence of unknown morphing algorithms and data sources is analysed. For this purpose, the detection performance on known morphing algorithms and data sources is compared to the detection performance on unknown data sources and morphing algorithms. It can be seen that changing the data source has a serious influence, whereas changing the morphing algorithm does not generally lead to a deterioration of the detection performance. However, it can be observed, that morphed images with higher quality (e.g. generated by FaceFusion or the UBO Morpher) are harder to detect than those with a lower qualtiy (e.g. generated by the OpenCV based algorithm or FaceMorpher).

In the second experiment, the best combinations of feature extractor and classifier in differential and single image scenarios are examined in a broad analysis on the non post-processed images. It is shown that in the single image scenario, SVM based algorithms generally provide the best detection performance. Furthermore, it can be observed that the morphs of the FERET database are more difficult to detect than those of the FRGC. The best feature extractors in the single image scenario are LBP with a patch size of 3×3 and a cell division into 4×4 cells, BSIF with a filter size of 9×9 and a cell division into 4×4 cells, and HOG. In the differential scenario it is shown that the morph of the FRGC is more difficult to detect than that of the FERET. Furthermore, the deep feature based algorithms provide by far the best results, led by the ArcFace feature extractor in combination with an SVM.

In the third experiment, the influence of different post-processings, expected during the processing of the passport photographs, are examined. For this purpose, the algorithms recognised as best performing in experiment 2 are evaluated on post-processed images according to the scheme described in Section 21.3. It turns out that the resizing of the images has no noticeable impact on the detection performance of the MAD algorithms. However, the remaining post-processings (*JP* and *PS*) significantly reduce the detection performance, in particular for algorithms utilizing hand-crafted features. The deep features based algorithms exhibit a constant detection performance across all post-processings.

The last experiment investigates whether a score level fusion of different MAD systems might lead to an improvement of the overall system. For this purpose the optimal weights for a fusion of two algorithms are determined for the algorithms determined as best performing in experiment 2. It can be observed, that on non post-process images a fusion might result in an improvement of the detection performance. On post-process images, however, the performance of the non-fused algorithms is superior. Thus, it can be generally concluded that a fusion might be particularly detrimental to the robustness of the algorithms and does not provide an additional benefit in the constellations tested. Finally, it can be stated that in the single image scenario HOG based algorithms show the best performance and in the differential scenario ArfFace based algorithms.

Part VII

CONCLUSIONS

26 SUMMARY OF RESULTS

In the scope of this thesis, morphing attacks are thoroughly investigated and various algorithms are tested for their suitability as MAD systems. In this chapter, the findings gathered in the previous chapters are summarised in answers to the research questions defined in Section 3.2. Finally, in order to strengthen the validity of the evaluations, the results obtained on the database described in Part V are compared with those obtained on the independent databases of the SOTAMD and NIST FRVT MORPH projects described in Section 3.1.1 and 3.1.3 respectively.

26.1 RQ1: EVALUATION METRICS

> *Which metrics are applicable for the evaluation of the vulnerability of FRSs and MAD algorithms?*

In the context of the analysis of morphing attacks, metrics are needed for two types of evaluations, to describe the vulnerability of FRS and to describe the detection performance of MAD algorithms. To evaluate the vulnerability of FRSs, the baseline performance can be determined using the FMR and FNMR defined in ISO/IEC JTC1 SC37 and described in Section 7.3. The vulnerability of the FRS itself can be reported either independently of the baseline performance in form of MMPMR, or as a function of the baseline performance in form of RMMR. Both metrics are defined in Section 11.2.1. To evaluate the detection performance of the MAD algorithms, the APCER and BPCER defined in ISO/IEC 30107-3 and described in Section 11.2.3 can be used.

26.2 RQ2: SYSTEM VULNERABILITY

> *Under which circumstances is a system vulnerable to morphing attacks?*

The vulnerability of biometric systems can be determined on a theoretical level, as described in Section 11.2.2. This procedure can be applied to any system, but due to the fact that it is based on assumptions about the distribution of the comparison scores, the real vulnerability of the systems may differ. In Chapter 23, the vulnerability of various FRSs is empirically determined using a realistic database and reported by means of MMPMR and RMMR. From the obtained resuslts it can be concluded that FRS showing a high baseline performance are more vulnerable to morphing attacks. The quality of the morphs used for the attack is of subordinate importance. The quality of the TLCs contained in the database, on the other hand, has a great effect not only on the scores obtained with the morphs, but also on the baseline performance of the FRS.

26.3 RQ3: INFLUENCE OF UNKNOWN DATA SOURCES

> *Does the consideration of images from unknown data sources influence the evaluations results of MAD algorithms?*

In the first experiment in Section 24.1 the influence of unknown data sources on the detection performance of two conventional MAD algorithms, namely LBP and BSIF in combination with an SVM, is investigated. It can be observed that the error rates in the detection of morphed images more than doubles if tested on data from a different database than the training data. It can be concluded that the algorithms tend to overfit for database-specific features, which emphasises the need for cross-database evaluations. A change of the morphing algorithms used to create the training data does not necessarily lead to a deterioration of the detection performance of the MAD algorithms. The quality of the morphs used for testing, however, has a higher impact on the detection performance of the MAD algorithms. As a consequence, the morphs created with more complex morphing algorithms, e.g. FaceFusion and UBO Morpher, are usually more difficult to detect.

26.4 RQ4: DETECTION OF MORPHED IMAGES

> *To what extent can morphed face images be reliable detected by automated algorithms?*

MAD algorithms can be constructed according to two distinct schemes depending on the scenario, which are described in Section 11.1. If only the image to be analysed is available, the information contained in the image can be evaluated by S-MAD algorithms. If a TLC is available in addition to the potential morph, the additional information of the further image can be evaluated by differential MAD algorithms. The feature extractors investigated in this thesis are described in Chapter 16. In addition, the use of each feature extractor is motivated. Six different types of feature extractors are considered, namely texture descriptors, gradient based descriptors, keypoint descriptors, landmark descriptors, image noise pattern and deep features. These are tested in combination with four different classifiers: SVM, Random Forest, AdaBoost, Gradient Boosting. In the single image scenario, morphs can be detected by HOG in combination with an SVM with a D-EER between 13.25% and 24.05%, depending on the database and the morphing algorithm used to create training and test data. The detection performance in the differential scenario for ArcFace features in combination with an SVM ranges from 2.71% to 7.17% D-EER.

26.5 RQ5: INFLUENCE OF OPERATIONAL SCENARIOS

> *Which operational scenarios influence the detection of morphed face images?*

Various post-processings expected to be applied to the passport image during the passport creation process are described in Section 21.3, namely *NPP*, *resized*, *JP2* and *PS*. In experiment 3, in Section 24.3, the effect of these post-processings on the detection performance of the MAD algorithms identified as best performing in experiment 2 is investigated. It is shown that post-processing of images after *RS* has no discernible effect on the detection performance of the algorithms. In the single image scenario, the detection performance for HOG with SVM is from 13.16% to 24.05% D-EER and from 2.58% to 7.20% for ArcFace features with SVM in the differential scenario. The post-processing according to *JP2* significantly influences the detection performance of the single image algorithms, thus increasing the D-EER to from 19.32% to 31.90%. It should be noted that constellations which previously showed a higher D-EER usually show a higher D-EER after post-processing as well. In the differential scenario, no change can be observed for the ArcFace based algorithm. It achieves error rates of 2.71% to 7.30% D-EER. Other differential algorithms, especially those not based on deep features, are partly significantly influenced by the post-processing. The post-processing after *PS* again reduces the detection performance of the single image algorithms. Thus error rates of 19.29% to 53.42% are achieved. In the differential scenario the ArcFace based algorithm shows a high robustness and achieves error rates of 1.29% to 7.76% D-EER.

26.6 RQ6: INFORMATION FUSION

> *Can information fusion be used to improve the MAD performance and robustness of the individual algorithms?*

In experiment 4, in Section 24.4, the algorithms determined to be best performing in experiment 2 are merged at the score level to evaluate a possible improvement of the overall system performance. The weights for the fusion are determined exclusively on the images of the *NPP* scenarios. In general, the weights for optimal overall system performance vary depending on the database used. Furthermore, in some constellations a fusion does not provide any advantage, thus the optimal weights for one of the two algorithms to be fused is set to 100%, for example in the single image scenario for the fusion of HOG and BSIF, and in the differential scenario for the fusion of ArcFace with HOG or Eyedea. Furthermore, it can be observed that in most constellations the non-fused algorithms are superior to the fused algorithms in terms of robustness against post-processing. Thus, it can be stated that a fusion of the algorithms in the tested constellations does not provide an additional value.

27 VALIDATION OF RESULTS

When creating the database described in Part V, care was taken to use as many morphing algorithms and post-processings as possible in order to guarantee a high reliability of the results. However, despite all these measures, there are still certain connections between training and test database. For example, both databases have been normalized according to the same scheme and the post-processing steps are identical (for the same scenario). In order to check the results obtained in Chapter 24 for undetected database-dependent errors, part of the high-performance algorithms were tested on the independent databases of the SOTAMD project described in Section 3.1.1 and the NIST FRVT MORPH project described in Section 3.1.3. In the following, the tests performed are described and the obtained results are presented.

27.1 SOTAMD

The test database of the SOTAMD project was created in a cooperation of the University of Applied Sciences Darmstadt, University of Bologna, University of Twente and NTNU. Each institution has acquired a face database, created morphs with different morph factors and post-processed images (including manual post-processing). On the resulting database of 10960 bona fides and 37000 morphs (per factor) MAD algorithms adapted to the test framework can be tested against digital, as well as printed and scanned images and evaluated according to different data subsets (e.g. only male or female).

A comparison of the D-EER determined in this thesis with the performance determined on the SOTAMD data is given in Table 27.1. The comparison is conducted in the digital and print-scan scenarios. The D-EERs given for this thesis are determined on the FRGC and averaged across all morphing algorithms. Please note that only one algorithm, the PRNU based, has been tested in the single image scenario so far. The configuration corresponds to PRNU-1 described in Section 16.5.1. In the differential scenario, LBP were previously tested with a patch size of 9×9 and a cell partitioning of 4×4, BSIF with a filter size of 3×3 with 8 bits and without cell division, dlib landmarks, and ArcFace feature were tested in combination with an SVM. Except for the LBP algorithm, the implementation and functionality is identical to the description in Section 16, allowing a direct comparison of these algorithms. When comparing the results for LBP, the configuration differs slightly (cell division of 4×4 in this work and 3×3 on the SOTAMD data).

Algorithm	Scheme	D-EER			
		Thesis		SOTAMD	
		Digital	Print-Scan	Digital	Print-Scan
PRNU	single image	39.44%	39.20%	44.81%	48.04%
LBP	differential	33.78%	40.28%	33.47%	29.28%
BSIF	differential	46.63%	53.01%	45.93%	51.36%
Landmarks	differential	44.15%	43.17%	37.13%	36.17%
ArcFace	differential	5.03%	5.44%	4.54%	4.62%

Table 27.1: Performance of tested algorithms compared to SOTAMD evaluation

It can be observed that the D-EERs are in the same order of magnitude, which indicates the reliability of the results presented in this thesis. Of particular note is the excellent performance of the ArcFace based algorithm, which achieves even better results on the SOTAMD data than on the database used in this work.

27.2 NIST FRVT MORPH

The NIST FRVT MORPH is a challenge organized by NIST for a consistent evaluation of MAD algorithms. Single and differential MAD algorithms are tested on three different classes of morphs: Low quality morphs (LQ), automated morph (Autom.) and high quality morphs (HQ). A detailed description of the individual data sets can be found in [103].

In this test environment, the algorithms submitted for the SOTAMD evaluation were tested as well. In addition to the differential versions of the LBP and BSIF algorithms, single image versions have been submitted for evaluation. The filter and patch size of the differential versions were applied.

The results of the tested algorithms are listed in Table 27.2. Within the evaluation of the NIST FRVT MORPH, no D-EER is calculated, but APCER and BPCER at a fixed threshold, as well as BPCER10 and BPCER100. Since in this thesis the algorithms are evaluated independently of the threshold, BPCER10 is most suitable for comparison. The comparison is made with the average BPCER over all morphing algorithms of the *NPP* images of the FRGC database.

It can be observed that single image algorithms consistently exhibit a BPCER10 of close-to 100%. It is interesting that the automatically generated morphs are more difficult to detect for the differential algorithms than the high quality morphs. Again, the ArcFace features based algorithm stands out in this evaluation due to its excellent detection performance.

Algorithm		Scheme	BPCER10			
in Thesis	In [103]		Thesis	LQ	Autom.	HQ
LBP + SVM	hdalbp-006	single image	59%	99%	93%	0.99
BSIF + SVM	hdabsif-004	single image	47%	98%	96%	100%
PRNU-1	hdaprnu-004	single image	53%	98%	100%	99%
LBP + SVM	hdalbp-006	differential	73%	82%	95%	91%
BSIF + SVM	hdabsif-004	differential	84%	60%	95%	66%
Dlib Landmarks + SVM	hdawl-002	differential	85%	90%	90%	84%
ArcFace Features + SVM	hdaarcface-001	differential	2%	2%	13%	9%

Table 27.2: Performance of tested algorithms compared to NIST FRVT MORPH evaluation

28 FUTURE WORK

In this thesis morphing attacks and the respective MAD algorithms are thoroughly tested and investigated. During the work on the topic, subjects emerged which could not be covered in detail anymore, but which offer interesting tasks for future work. The most relevant aspects are listed and described in this chapter.

28.1 STANDARDISATION

In order to create a common basis for discussions in research, it is beneficial to work in accordance with standardised methods. This can be facilitated, for example, by using a uniform vocabulary [66]. By standardising metrics, it is possible to create a common basis for the comparison of evaluations. For evaluations in biometrics there are already various standards for different use cases, such as those standardised in [62] and [66] and described in Section 7.3 for the evaluation of the recognition performance of biometric systems, or the metrics standardised in [65] and described in Section 11.2.3 for the evaluation of the detection performance of MAD or PAD systems. However, the metrics for assessing the vulnerability of biometric systems to morphing attacks have not yet been standardised. Although in [65] with IAPMR, a metric for assessing the vulnerability of biometric systems against presentation attacks is introduced, and with the Relative Impostor Attack Presentation Accept Rate (RIAPAR) an adaptation for presentation attacks of the RMMR presented in Section 11.2.1 will be included in the revision of the ISO/IEC 30107-3 standard, but these metrics, as described in Section 11.2.1, are not directly applicable for vulnerability analysis by morphing attacks. Thus, further standardisation work is required.

28.2 REALISTIC DATABASES

Another factor essential for the comparability of results is the availability of uniform databases. The offer of a uniform evaluation by the SOTAMD project and the NIST FRVT MORPH provides a platform for direct comparison of different algorithms. However, the databases used for evaluation are kept confidential, thus it is not possible to use these data for training algorithms. So far, no public databases are available for the training of MAD algorithms. The distribution of such databases is hampered by the licensing structures of face databases and by the applicable data protection laws. As a consequence, in-house morphing databases have to be created. These databases often lack realism, which means that evaluations on the own databases usually yield promising results, which cannot be validated on other test data. The creation of an open accessible morphing database could provide a common understanding of the quality of image data, reduce effort and promote the reproducibility of research. For this purpose, however, the problems of licenses and data protection need to be settled.

28.3 REPRODUCIBLE RESULTS

In general, a reproducibility of the previously published results is barely given. This problem is closely related to the lack of standards for evaluation and the absence of open accessible databases. This issue is further aggravated by the fact that some of the implementations used in the publications are inadequately described, making a correct reimplementation of the algorithms impossible. This leads to the problem that published results cannot be reproduced, making it difficult to link to the work of other researchers or to compare results.

28.4 FURTHER ANALYSIS OF DEEP FEATURES

Within the scope of this thesis it has been shown that algorithms which evaluate deep features in the differential scenario can offer a high and stable detection performance across different scenarios. Due to the high complexity of the processes during the ArcFace features extraction, the underlying operations are not easy to comprehend. Through deeper analysis of the network and training, it may be possible to learn which information is being represented in the feature vector. In the long run, it may be possible to create a hand-crafted feature extractor representing the same information. This information could also be helpful in the manual detection of morphs, in order to be able to make robust decisions.

GLOSSARY

APCER
Proportion of attack presentations using the same PAI species incorrectly classified as bona fide presentations in a specific scenario [65]. xix, 77, 78, 87, 166, 187, 192

Biometric Feature
Numbers or labels extracted from Biometric Samples and used for comparison [66]. 3, 52

Biometric Sample
Analog or digital representation of biometric characteristics prior to biometric feature extraction [66]. 5, 7, 8, 13, 49–52, 54, 56, 73, 75–77, 113, 123, 130, 131, 136

Biometric Verification
Process of confirming a biometric claim through biometric comparison [66]. 53

Bona Fide Presentation
Interaction of the biometric capture subject and the biometric data capture subsystem in the fashion intended by the policy of the biometric system [65]. 9, 71, 74, 75, 77, 82, 102, 105, 114, 123, 125–127, 129–131, 133–137, 149, 164, 179, 183, 191

BPCER
Proportion of bona fide presentations that cause no response at the PAD subsystem or data capture [65]. xix, 77, 78, 87, 148, 150, 161, 164, 166, 187, 192

D-EER
The error rate at the operating point at which both, APCER and BPCER, are equal. xix, 78, 148–150, 152, 154, 157, 160, 161, 164, 166, 169, 170, 172–176, 178, 181, 182, 188, 189, 191, 192

EER
The error rate at the operating point at which both, acceptance and rejection errors, are equal. xx, 53, 54, 78, 154

FAR
Proportion of verification transactions with wrongful claims of identity that are incorrectly confirmed [62]. xx, 53, 56

FMR
The false match rate is the proportion of samples, acquired from zero-effort impostor attempts, that are falsely declared to match the compared non-self template [62]. xx, 53, 54, 56, 72, 77, 78, 141, 142, 187

FNMR The false non-match rate is the proportion of samples, acquired from genuine attempts, that are falsely declared not to match the template of the same characteristic from the same user supplying the sample [62]. xx, 52–54, 56, 72, 74–78, 142, 168, 187

FRR Proportion of verification transactions with truthful claims of identity that are incorrectly denied [62]. xx, 53, 56

FTA The failure-to-acquire rate is the proportion of verification or identification attempts for which the system fails to capture or locate a sample of sufficient quality [62]. xx, 52

FTC Failure of the biometric capture process to produce a captured Biometric Sample of the biometric characteristic of interest [66]. xx, 51

FTE The failure-to-enrol rate is the proportion of the population for whom the system fails to complete the enrolment process [62]. xx, 52

IAPMR In a full-system evaluation of a verification system, the proportion of impostor attack presentations using the same PAI species in which the target reference is matched [65]. xx, 72–74, 87, 195

MAD Detection of morphed face images during an Biometric Enrolment or Biometric Verification attempt. xxi, 5, 6, 8–12, 71, 72, 76–81, 83–87, 91, 92, 100, 101, 103, 105–107, 109–111, 113–116, 119, 123, 132, 136, 137, 147–150, 152, 154, 157, 160, 161, 164, 166, 168, 170, 172–174, 176, 178–184, 187–189, 191, 192, 195

MMPMR In a full-system evaluation of a verification system, the proportion of morphing attack presentations in which the target reference is matched. xxi, 73–76, 87, 143, 183, 187

RIAPAR Ratio of the IAPMR to the FRR of the system. xxi

RMMR In a full-system evaluation of a verification system, the MMPMR in relation to the TMR. xxi, 74–76, 87, 143, 183, 187, 195

TMR Proportion of verification transactions with truthful claims of identity that are correctly confirmed. xxii, 75, 76, 142

BIBLIOGRAPHY

[1] A. Agarwal, R. Singh, M. Vatsa, and A. Noore. "SWAPPED! Digital face presentation attack detection via weighted local magnitude pattern." In: *Proceedings of the 2017 International Joint Conference on Biometrics (IJCB)*. IEEE, 2017.

[2] T. Ahonen, A. Hadid, and M. Pietikäinen. "Face Recognition with Local Binary Patterns." In: *Lecture Notes in Computer Science*. Springer Berlin Heidelberg, 2004, pp. 469–481.

[3] A. Anjos, M. M. Chakka, and S. Marcel. "Motion-based countermeasures to photo attacks in face recognition." In: *IET Biometrics* 3.3 (2014), pp. 147–158.

[4] N. Arad, N. Dyn, D. Reisfeld, and Y. Yeshurun. "Image Warping by Radial Basis Functions: Application to Facial Expressions." In: *CVGIP: Graphical Models and Image Processing* 56.2 (1994), pp. 161–172.

[5] A. Asaad and S. Jassim. "Topological Data Analysis for Image Tampering Detection." In: *Proceedings of 2017 International Workshop on Digital Watermarking (IWDW): Digital Forensics and Watermarking*. Springer International Publishing, 2017, pp. 136–146.

[6] N. Ayat, M. Cheriet, and C. Suen. "Automatic model selection for the optimization of SVM kernels." In: *Pattern Recognition* 38.10 (2005), pp. 1733–1745.

[7] F. A. C. Azevedo, L. R. B. Carvalho, L. T. Grinberg, J. M. Farfel, R. E. L. Ferretti, R. E. P. Leite, W. J. Filho, R. Lent, and S. Herculano-Houzel. "Equal numbers of neuronal and nonneuronal cells make the human brain an isometrically scaled-up primate brain." In: *The Journal of Comparative Neurology* 513.5 (2009), pp. 532–541.

[8] J. Ba and R. Caruana. "Do Deep Nets Really Need to be Deep?" In: *Proceedings of the 27th Advances in Neural Information Processing Systems (NIPS)*. MIT Press, 2014, pp. 2654–2662.

[9] G. E. A. P. A. Batista, R. C. Prati, and M. C. Monard. "A study of the behavior of several methods for balancing machine learning training data." In: *ACM SIGKDD Explorations Newsletter* 6.1 (2004), p. 20.

[10] H. Bay, A. Ess, T. Tuytelaars, and L. V. Gool. "Speeded-Up Robust Features (SURF)." In: *Computer Vision and Image Understanding* 110.3 (2008), pp. 346–359.

[11] T. Beier and S. Neely. “Feature-based image metamorphosis.” In: *Proceedings of the 19th annual conference on Computer graphics and interactive techniques (SIGGRAPH)*. ACM Press, 1992.

[12] D. Borchers. *Gesetzentwurf: Passfotos nur noch unter behördlicher Aufsicht anfertigen*. Mar. 13, 2020. URL: https://www.heise.de/newsticker/meldung/Gesetzentwurf-Passfotos-nur-noch-unter-behoerdlicher-Aufsicht-anfertigen-4616624.html.

[13] L. Breiman. “Random Forests.” In: *Machine Learning* 45.1 (2001), pp. 5–32.

[14] Bundesministerium des Innern, ed. *Verordnung zur Durchführung des Passgesetzes*. 2017.

[15] Bundespolizeipräsidium. *Jahresbericht 2018 der Bundespolizei*. Tech. rep. Bundesministerium des Innern, für Bau und Heimat, Mar. 10, 2020.

[16] O. Chapelle and V. Vapnik. “Model Selection for Support Vector Machines.” In: *Proceedings of the 1999 Conference on Neural Information Processing Systems (NIPS)*. 1999.

[17] D. Chen, X. Cao, F. Wen, and J. Sun. “Blessing of Dimensionality: High-Dimensional Feature and Its Efficient Compression for Face Verification.” In: *Proceedings of the 2013 Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2013.

[18] M. Chen, J. Fridrich, M. Goljan, and J. Lukas. “Determining Image Origin and Integrity Using Sensor Noise.” In: *IEEE Transactions on Information Forensics and Security* 3.1 (2008), pp. 74–90.

[19] M. Claesen and B. D. Moor. “Hyperparameter Search in Machine Learning.” In: *arXiv e-prints* (Feb. 7, 2015).

[20] T. F. Cootes, C. J. Taylor, D. Cooper, and J. Graham. “Active Shape Models-Their Training and Application.” In: *Computer Vision and Image Understanding* 61.1 (1995), pp. 38–59.

[21] C. Cortes and V. Vapnik. “Support-Vector Networks.” In: *Machine Learning* 20.3 (1995), pp. 273–297.

[22] G. Cybenko. “Approximation by superpositions of a sigmoidal function.” In: *Mathematics of Control, Signals, and Systems* 2.4 (1989), pp. 303–314.

[23] N. Dalal and B. Triggs. “Histograms of Oriented Gradients for Human Detection.” In: *Proceedings of the 2005 Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2005.

[24] N. Damer, V. Boller, Y. Wainakh, F. Boutros, P. Terhörst, A. Braun, and A. Kuijper. "Detecting Face Morphing Attacks by Analyzing the Directed Distances of Facial Landmarks Shifts." In: *Proceedings of the 40th German Conference of Pattern Recognition (GCPR)*. 2018.

[25] N. Damer, A. M. Saladie, A. Braun, and A. Kuijper. "MorGAN: Recognition Vulnerability and Attack Detectability of Face Morphing Attacks Created by Generative Adversarial Network." In: *Proceedings of the 9th International Conference on Biometrics: Theory, Applications, and Systems (BTAS)*. IEEE, 2018.

[26] L. Debiasi, N. Damer, A. M. Saladié, C. Rathgeb, U. Scherhag, C. Busch, F. Kirchbuchner, and A. Uhl. "On the Detection of GAN-based Face Morphs using Established Morph Detectors." In: *Proceedings of the 20th International Conference on Image Analysis and Processing (ICIAP)*. 2019.

[27] L. Debiasi, C. Rathgeb, U. Scherhag, A. Uhl, and C. Busch. "PRNU Variance Analysis for Morphed Face Image Detection." In: *Proceedings of the 9th International Conference on Biometrics: Theory, Applications, and Systems (BTAS)*. IEEE. 2018.

[28] L. Debiasi, U. Scherhag, C. Rathgeb, A. Uhl, and C. Busch. "PRNU-based Detection of Morphed Face Images." In: *Proceedings of the 6th International Workshop on Biometrics and Forensics (IWBF)*. IEEE. 2018.

[29] J. Deng, J. Guo, N. Xue, and S. Zafeiriou. "ArcFace: Additive Angular Margin Loss for Deep Face Recognition." In: *Proceedings of the 2019 Computer Vision and Pattern Recognition (CVPR)*. 2019.

[30] O. Déniz, G. Bueno, J. Salido, and F. D. la Torre. "Face recognition using Histograms of Oriented Gradients." In: *Pattern Recognition Letters* 32.12 (2011), pp. 1598–1603.

[31] N. Erdogmus and S. Marcel. "Spoofing 2D face recognition systems with 3D masks." In: *Proceedings of the 2013 International Conference of the Biometrics Special Interest Group (BIOSIG)*. Darmstadt, 2013, pp. 1–8.

[32] D. Erhan, C. Szegedy, A. Toshev, and D. Anguelov. "Scalable Object Detection using Deep Neural Networks." In: *Proceedings of the 2014 Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2014.

[33] European Commission. *EU-eMRTD Specification*. Tech. rep. European Commission, 2018.

[34] European Commission, ed. *Detections of illegal border-crossings; monthly statistics*. Mar. 13, 2020. URL: https://ec.europa.eu/knowledge4policy/dataset/ds00032_en.

[35] S. Fadnavis. “Image interpolation techniques in digital image processing: an overview.” In: *International Journal of Engineering Research and Applications* 4.10 (2014), pp. 70–73.

[36] Z.-H. Feng, J. Kittler, M. Awais, P. Huber, and X.-J. Wu. “Wing Loss for Robust Facial Landmark Localisation with Convolutional Neural Networks.” In: *Proceedings of the 2018 Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2018.

[37] M. Fernández-Delgado, E. Cernadas, S. Barro, and D. Amorim. “Do we Need Hundreds of Classifiers to Solve Real World Classification Problems?” In: *Journal of Machine Learning Research* 15 (2014), pp. 3133–3181.

[38] M. Ferrara, A. Franco, and D. Maltoni. “Decoupling texture blending and shape warping in face morphing.” In: *Proceedings of the 2019 International Conference of the Biometrics Special Interest Group (BIOSIG)*. 2019, pp. 1–5.

[39] M. Ferrara, A. Franco, and D. Maltoni. “The magic passport.” In: *Proceedings of the 2014 International Joint Conference on Biometrics (IJCB)*. IEEE, 2014.

[40] M. Ferrara, A. Franco, and D. Maltoni. “Face Demorphing.” In: *IEEE Transactions on Information Forensics and Security* 13.4 (2018), pp. 1008–1017.

[41] M. Ferrara, A. Franco, and D. Maltoni. “Face demorphing in the presence of facial appearance variations.” In: *Proceedings of the 26th European Signal Processing Conference (EUSIPCO)*. 2018.

[42] Y. Freund and R. E. Schapire. “A desicion-theoretic generalization of on-line learning and an application to boosting.” In: *Lecture Notes in Computer Science*. Springer Berlin Heidelberg, 1995, pp. 23–37.

[43] J. H. Friedman. “Greedy function approximation: A gradient boosting machine.” In: *The Annals of Statistics* 29.5 (2001), pp. 1189–1232.

[44] J. H. Friedman. “Stochastic gradient boosting.” In: *Computational Statistics & Data Analysis* 38.4 (2002), pp. 367–378.

[45] W. Gao, B. Cao, S. Shan, D. Zhou, X. Zhang, and D. Zhao. *The CAS-PEAL Large-Scale Chinese Face Database and Baseline Evaluations*. Tech. rep. JDL-TR-04-FR-001. Chinese Academy of Sciences, May 2004.

[46] T. Gloe, S. Pfennig, and M. Kirchner. “Unexpected artefacts in PRNU-based camera identification.” In: *Proceedings of the 2012 Conference on Multimedia and Security (MM&Sec)*. ACM Press, 2012.

[47] M. Gomez-Barrero, C. Rathgeb, U. Scherhag, and C. Busch. "Is your biometric system robust to morphing attacks?" In: *Proceedings of the 5th International Workshop on Biometrics and Forensics (IWBF)*. IEEE, Apr. 2017.

[48] M. Gomez-Barrero, C. Rathgeb, U. Scherhag, and C. Busch. "Predicting the vulnerability of biometric systems to attacks based on morphed biometric information." In: *IET Biometrics* 7.4 (July 2018), pp. 333–341.

[49] G. Gordon. "Face recognition based on depth and curvature features." In: *Proceedings of the 1992 Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 1992.

[50] A. A. Goshtasby. "Image Descriptors." In: *Image Registration*. Springer London, 2012, pp. 219–246.

[51] M. Grgic, K. Delac, and S. Grgic. "SCface surveillance cameras face database." In: *Multimedia Tools and Applications* 51.3 (2009), pp. 863–879.

[52] R. Gross, I. Matthews, J. Cohn, T. Kanade, and S. Baker. "Multi-PIE." In: *Proceedings of the 8th International Conference on Automatic Face & Gesture Recognition (FG2008)*. IEEE, 2008.

[53] R. H. R. Hahnloser, R. Sarpeshkar, M. A. Mahowald, R. J. Douglas, and H. S. Seung. "Digital selection and analogue amplification coexist in a cortex-inspired silicon circuit." In: *Nature* 405.6789 (2000), pp. 947–951.

[54] H. Hatem, Z. Beiji, and R. Majeed. "A Survey of Feature Base Methods for Human Face Detection." In: *International Journal of Control and Automation* 8.5 (2015), pp. 61–78.

[55] D. M. Hawkins. "The Problem of Overfitting." In: *Journal of Chemical Information and Computer Sciences* 44.1 (2004), pp. 1–12.

[56] D.-C. He and L. Wang. "Detecting texture edges from images." In: *Pattern Recognition* 25.6 (1992), pp. 595–600.

[57] K. He, J. Sun, and X. Tang. "Guided Image Filtering." In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 35.6 (2013), pp. 1397–1409.

[58] M. Hildebrandt, T. Neubert, A. Makrushin, and J. Dittmann. "Benchmarking face morphing forgery detection: Application of StirTrace for impact simulation of different processing steps." In: *Proceedings of the 5th International Workshop on Biometrics and Forensics (IWBF)*. IEEE, 2017.

[59] E. Hjelmås and B. K. Low. "Face Detection: A Survey." In: *Computer Vision and Image Understanding* 83.3 (2015), pp. 236–274.

[60] R. Hu, R. Shi, I fan Shen, and W. Chen. "Video Stabilization Using Scale-Invariant Features." In: *Proceedings of the 11th International Conference Information Visualization (IV07)*. IEEE, 2007.

[61] C.-M. Huang, Y.-J. Lee, D. K. Lin, and S.-Y. Huang. "Model selection for support vector machines via uniform design." In: *Computational Statistics & Data Analysis* 52.1 (2007), pp. 335–346.

[62] ISO/IEC JTC1 SC37 Biometrics. *Information technology – Biometric performance testing and reporting – Part 1: Principles and framework*. ISO ISO/IEC 19795-1:2006. Geneva, Switzerland: International Organization for Standardization, 2006.

[63] ISO/IEC JTC1 SC37 Biometrics. *Information technology – Biometric data interchange formats – Part 5: Face image data*. ISO/IEC 19794-5:2005. 2011.

[64] ISO/IEC JTC1 SC37 Biometrics. *Information technology – Biometric presentation attack detection – Part 1: Framework*. ISO ISO/IEC IS 30107-3:2017. Geneva, Switzerland: International Organization for Standardization, 2016.

[65] ISO/IEC JTC1 SC37 Biometrics. *Information technology – Biometric presentation attack detection – Part 3: Testing and Reporting*. ISO ISO/IEC IS 30107-3:2017. Geneva, Switzerland: International Organization for Standardization, 2017.

[66] ISO/IEC JTC1 SC37 Biometrics. *Information technology – Vocabulary – Part 37: Biometrics*. ISO ISO/IEC 2382-37:2017. Geneva, Switzerland: International Organization for Standardization, 2017.

[67] International Civil Aviation Organization. *ICAO Doc 9303, Machine Readable Travel Documents – Part 9: Deployment of Biometric Identification and Electronic Storage of Data in MRTDs (7th edition)*. Tech. rep. ICAO, 2015.

[68] A. Jain, A. Ross, and S. Prabhakar. "An Introduction to Biometric Recognition." In: *IEEE Transactions on Circuits and Systems for Video Technology* 14.1 (2004), pp. 4–20.

[69] S. Jassim and A. Asaad. "Automatic Detection of Image Morphing by Topology-based Analysis." In: *Proceedings of the 26th European Signal Processing Conference (EUSIPCO)*. 2018.

[70] T. Kalinke, C. Tzomakas, and W. von Seelen. "A texture-based object detection and an adaptive model-based classification." In: *Proceedings of the 1998 Intelligent Vehicles Symposium*. Vol. 98. Citeseer. IEEE, 1998, pp. 341–346.

[71] J. Kannala and E. Rahtu. "BSIF: Binarized statistical image features." In: *Proceedings of the 21st International Conference on Pattern Recognition (ICPR)*. 2012, pp. 1363–1366.

[72] T. Karras, S. Laine, M. Aittala, J. Hellsten, J. Lehtinen, and T. Aila. “Analyzing and Improving the Image Quality of StyleGAN.” In: *Proceedings of the 2020 Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2020.

[73] A. Kasinski, A. Florek, and A. Schmidt. “The PUT face database.” In: *Image Processing & Communications* (Jan. 2008).

[74] V. Kazemi and J. Sullivan. “One millisecond face alignment with an ensemble of regression trees.” In: *Proceedings of the 2014 Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2014.

[75] B. Keelan. *Handbook of Image Quality: Characterization and Prediction (Optical Science and Engineering)*. CRC Press, 2002. ISBN: 978-0-8247-0770-5.

[76] J. Kim, J. Kwon Lee, and K. Mu Lee. “Accurate Image Super-Resolution Using Very Deep Convolutional Networks.” In: *Proceedings of the 2016 Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2016.

[77] D. E. King. “Dlib-ml: A Machine Learning Toolkit.” In: *Journal of Machine Learning Research* 10 (Dec. 2009), pp. 1755–1758. ISSN: 1532-4435.

[78] R. D. King, C. Feng, and A. Sutherland. “STATLOG: Comparison of Classification Algorithms on Large Real-World Problems.” In: *Applied Artificial Intelligence* 9.3 (1995), pp. 289–333.

[79] F. Knopjes. *State of the art of Morphing Detection*. 2019. URL: https://www.icao.int/Meetings/TRIP-Symposium-2019/PublishingImages/Pages/Presentations/State%20of%20the%20art%20of%20Morphing%20Detection.pdf.

[80] K. Kotwal, Z. Mostaani, and S. Marcel. “Detection of Age-Induced Makeup Attacks on Face Recognition Systems Using Multi-Layer Deep Features.” In: *IEEE Transactions on Biometrics, Behavior, and Identity Science* 2.1 (2020), pp. 15–25.

[81] C. Kraetzer, A. Makrushin, T. Neubert, M. Hildebrandt, and J. Dittmann. “Modeling Attacks on Photo-ID Documents and Applying Media Forensics for the Detection of Facial Morphing.” In: *Proceedings of the 5th Workshop on Information Hiding and Multimedia Security (IHMMSec)*. ACM Press, 2017.

[82] A. Krizhevsky, I. Sutskever, and G. E. Hinton. “ImageNet classification with deep convolutional neural networks.” In: *Communications of the ACM* 60.6 (2017), pp. 84–90.

[83] M. C. Kus, M. Gokmen, and S. Etaner-Uyar. “Traffic sign recognition using Scale Invariant Feature Transform and color classification.” In: *Proceedings of the 23rd International Symposium on Computer and Information Sciences (ISCIS)*. IEEE, 2008.

[84] V. Kushwaha, M. Singh, R. Singh, M. Vatsa, N. Ratha, and R. Chellappa. “Disguised Faces in the Wild.” In: *Proceedings of the 2018 Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. IEEE, 2018.

[85] Y. LeCun, K. Kavukcuoglu, and C. Farabet. “Convolutional networks and applications in vision.” In: *Proceedings of the 2010 International Symposium on Circuits and Systems (ISCAS)*. IEEE, 2010.

[86] S.-Y. Lee, K.-Y. Chwa, J. Hahn, and S. Y. Shin. “Image morphing using deformable surfaces.” In: *Proceedings of the 1994 Computer Animation*. IEEE, 1994.

[87] S. Z. Li and A. K. Jain. *Encyclopedia of Biometrics*. Springer US, 2015.

[88] S. Liao, X. Zhu, Z. Lei, L. Zhang, and S. Z. Li. “Learning Multi-scale Block Local Binary Patterns for Face Recognition.” In: *Advances in Biometrics*. Springer Berlin Heidelberg, 2007, pp. 828–837.

[89] L. Liu, J. Chen, P. Fieguth, G. Zhao, R. Chellappa, and M. Pietikäinen. “From BoW to CNN: Two Decades of Texture Representation for Texture Classification.” In: *International Journal of Computer Vision* (2018).

[90] W. Liu, Y. Wen, Z. Yu, M. Li, B. Raj, and L. Song. “SphereFace: Deep Hypersphere Embedding for Face Recognition.” In: *Proceedings of the 2017 Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2017.

[91] Z. Liu, P. Luo, X. Wang, and X. Tang. “Deep Learning Face Attributes in the Wild.” In: *Proceedings of the 2015 International Conference on Computer Vision (ICCV)*. CelebA. IEEE, 2015.

[92] D. Lowe. “Object recognition from local scale-invariant features.” In: *Proceedings of the 7th International Conference on Computer Vision (ICCV)*. IEEE, 1999.

[93] D. G. Lowe. “Distinctive Image Features from Scale-Invariant Keypoints.” In: *International Journal of Computer Vision* 60.2 (2004), pp. 91–110.

[94] J. Löhr. *Passfotos von Fotohändlern bleiben erlaubt*. Mar. 13, 2020. URL: https://www.faz.net/aktuell/wirtschaft/plan-von-seehofer-passfotos-nur-noch-unter-staatlicher-aufsicht-16569744.html.

[95] D. S. Ma, J. Correll, and B. Wittenbrink. “The Chicago face database: A free stimulus set of faces and norming data.” In: *Behavior Research Methods* 47.4 (2015), pp. 1122–1135.

[96] A. Makrushin, C. Kraetzer, T. Neubert, and J. Dittmann. "Generalized Benfords Law for Blind Detection of Morphed Face Images." In: *Proceedings of the 6th Workshop on Information Hiding and Multimedia Security (IH&MMSec)*. ACM Press, 2018.

[97] A. Makrushin, T. Neubert, and J. Dittmann. "Automatic Generation and Detection of Visually Faultless Facial Morphs." In: *Proceedings of the 12th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISIGRAPP)*. SCITEPRESS - Science and Technology Publications, 2017.

[98] A. J. Mansfield and J. L. Wayman. *Best practices in testing and reporting performance of biometric devices*. Tech. rep. Centre for Mathematics and Scientific Computing, 2002.

[99] L. Mao, M. Xie, Y. Huang, and Y. Zhang. "Preceding vehicle detection using Histograms of Oriented Gradients." In: *Proceedings of the 2010 International Conference on Communications, Circuits and Systems (ICCCAS)*. IEEE, 2010.

[100] A. Martinez and R. Benavente. *The AR Face Database*. Tech. rep. 24. Computer Vision Center (CVC), June 1998.

[101] R. K. McConnell. "Method of and apparatus for pattern recognition." Pat. US4567610A. 1986.

[102] M. K. Mihcak, I. Kozintsev, and K. Ramchandran. "Spatially adaptive statistical modeling of wavelet image coefficients and its application to denoising." In: *Proceedings of the 1999 International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*. IEEE, 1999.

[103] M. Ngan, P. Grother, K. Hanaoka, and J. Kuo. *Face Recognition Vendor Test (FRVT) part 4: MORPH-Performance of Automated Face Morph Detection*. Tech. rep. National Institute of Standards and Technology (NIST), 2020.

[104] M. A. Nielsen. *Neural networks and deep learning*. Vol. 25. Determination press San Francisco, CA, USA: 2015.

[105] S. Notley and M. Magdon-Ismail. "Examining the Use of Neural Networks for Feature Extraction: A Comparative Analysis using Deep Learning, Support Vector Machines, and K-Nearest Neighbor Classifiers." In: *arXiv e-prints* (2018).

[106] T. Ojala, M. Pietikäinen, and D. Harwood. "A comparative study of texture measures with classification based on featured distributions." In: *Pattern Recognition* 29.1 (1996), pp. 51–59.

[107] M. A. Olsen, V. Šmida, and C. Busch. "Finger image quality assessment features – definitions and evaluation." In: *IET Biometrics* 5.2 (2016), pp. 47–64.

[108] M. A. Olsen, H. Xu, and C. Busch. "Gabor filters as candidate quality measure for NFIQ 2.0." In: *Proceedings of the 5th IAPR International Conference on Biometrics (ICB)*. IEEE, 2012.

[109] D. Ortega-Delcampo, C. Conde, D. Palacios-Alonso, and E. Cabello. "Border Control Morphing Attack Detection with a Convolutional Neural Network De-morphing Approach." In: *IEEE Access* (2020), pp. 1–1.

[110] U. Park, S. Pankanti, and A. K. Jain. "Fingerprint verification using SIFT features." In: *Biometric Technology for Human Identification V*. Ed. by B. V. Kumar, S. Prabhakar, and A. A. Ross. SPIE, 2008.

[111] O. M. Parkhi, A. Vedaldi, and A. Zisserman. "Deep Face Recognition." In: *Proceedings of the 2015 British Machine Vision Conference (BMVC)*. British Machine Vision Association, 2015.

[112] F. Peng, L.-B. Zhang, and M. Long. "FD-GAN: Face De-Morphing Generative Adversarial Network for Restoring Accomplice's Facial Image." In: *IEEE Access* 7 (2019), pp. 75122–75131.

[113] B. S. Phadikar, G. K. Maity, and A. Phadikar. "Full Reference Image Quality Assessment: A Survey." In: *Lecture Notes in Networks and Systems*. Springer Singapore, 2017, pp. 197–208.

[114] P. Phillips, P. Flynn, T. Scruggs, K. Bowyer, J. Chang, K. Hoffman, J. Marques, J. Min, and W. Worek. "Overview of the Face Recognition Grand Challenge." In: *Proceedings of the 2005 Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2005.

[115] P. Phillips, H. Wechsler, J. Huang, and P. J. Rauss. "The FERET database and evaluation procedure for face-recognition algorithms." In: *Image and Vision Computing* 16.5 (1998), pp. 295–306.

[116] J. Qin and Z.-S. He. "A SVM face recognition method based on Gabor-featured key points." In: *Proceedings of the 2005 International Conference on Machine Learning and Cybernetics (ICMLC)*. IEEE, 2005.

[117] J. R. Quinlan. "Induction of decision trees." In: *Machine Learning* 1.1 (1986), pp. 81–106.

[118] R. Raghavendra and C. Busch. "Presentation attack detection algorithm for face and iris biometrics." In: *Proceedings of the 22nd European Signal Processing Conference (EUSIPCO)*. 2014, pp. 1387–1391.

[119] R. Raghavendra and C. Busch. "Presentation Attack Detection Methods for Face Recognition Systems." In: *ACM Computing Surveys* 50.1 (2017), pp. 1–37.

[120] R. Raghavendra, K. B. Raja, and C. Busch. "Detecting morphed face images." In: *Proceedings of the 8th International Conference on Biometrics Theory, Applications and Systems (BTAS)*. IEEE, 2016.

[121] R. Raghavendra, K. B. Raja, S. Marcel, and C. Busch. "Face presentation attack detection across spectrum using time-frequency descriptors of maximal response in Laplacian scale-space." In: *Proceedings of the 6th International Conference on Image Processing Theory, Tools and Applications (IPTA)*. IEEE, 2016.

[122] R. Raghavendra, K. B. Raja, S. Venkatesh, and C. Busch. "Transferable Deep-CNN Features for Detecting Digital and Print-Scanned Morphed Face Images." In: *Proceedings of the 2017 Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. IEEE, 2017.

[123] R. Raghavendra, K. Raja, S. Venkatesh, and C. Busch. "Face morphing versus face averaging: Vulnerability and detection." In: *Proceedings of the 2017 International Joint Conference on Biometrics (IJCB)*. IEEE, 2017.

[124] R. Raghavendra, S. Venkatesh, K. Raja, and C. Busch. "Towards making Morphing Attack Detection robust using hybrid Scale-Space Colour Texture Features." In: *Proceedings of 5th International Conference on Identity, Security and Behaviour Analysis (ISBA)*. 2019, pp. 22–24.

[125] R. Raghavendra, S. Venkatesh, K. Raja, and C. Busch. "Detecting Face Morphing Attacks with Collaborative Representation of Steerable Features." In: *Proceedings of the 3rd Computer Vision and Image Processing (CVIP2018)*. 2018, pp. 1–11.

[126] C. Rathgeb, A. Dantcheva, and C. Busch. "Impact and Detection of Facial Beautification in Face Recognition: An Overview." In: *IEEE Access* 7 (2019), pp. 152667–152678.

[127] Research and Development Unit. *Best Practice Technical Guidelines for Automated Border Control (ABC) Systems*. Tech. rep. FRONTEX, 2012.

[128] F. Rosenblatt. *Principles of neurodynamics. perceptrons and the theory of brain mechanisms*. Tech. rep. Cornell Aeronautical Lab Inc Buffalo NY, 1961.

[129] D. Ruprecht and H. Muller. "Image warping with scattered data interpolation." In: *IEEE Computer Graphics and Applications* 15.2 (1995), pp. 37–43.

[130] S. R. Safavian and D. A. Landgrebe. "A survey of decision tree classifier methodology." In: *IEEE Trans. Systems, Man, and Cybernetics* 21 (1991), pp. 660–674.

[131] U. Scherhag, D. Budhrani, M. Gomez-Barrero, and C. Busch. "Detecting Morphed Face Images Using Facial Landmarks." In: *Proceedings of the 2018 International Conference on Image and Signal Processing (ICISP)*. Springer International Publishing, 2018, pp. 444–452.

[132] U. Scherhag, L. Debiasi, C. Rathgeb, C. Busch, and A. Uhl. "Detection of Face Morphing Attacks based on PRNU Analysis." In: *IEEE Transactions on Biometrics, Behavior, and Identity Science* (2019), pp. 1–1.

[133] U. Scherhag, J. Kunze, C. Rathgeb, and C. Busch. "Face Morph Detection for Unknown Morphing Algorithms and Image Sources: A Multi-Scale Block Local Binary Pattern Fusion Approach." In: *IET-Biometrics* (2020), pp. 1–11.

[134] U. Scherhag, R. Raghavendra, K. B. Raja, M. Gomez-Barrero, C. Rathgeb, and C. Busch. "On the vulnerability of face recognition systems towards morphed face attacks." In: *Proceedings of the 5th International Workshop on Biometrics and Forensics (IWBF)*. IEEE, Apr. 2017.

[135] U. Scherhag, C. Rathgeb, and C. Busch. "Morph detection from single face images: a multi-algorithm fusion approach." In: *Proceedings of the 2018 International Conference on Biometrics Engineering and Application (ICBEA)*. ACM Press, 2018.

[136] U. Scherhag, C. Rathgeb, and C. Busch. "Performance Variation of Morphed Face Image Detection Algorithms across different Datasets." In: *Proceedings of the 6th International Workshop on Biometrics and Forensics (IWBF)*. IEEE, June 2018.

[137] U. Scherhag, C. Rathgeb, and C. Busch. "Towards detection of morphed face images in electronic travel documents." In: *Proceedings of the 13th IAPR Workshop on Document Analysis Systems (DAS)*. 2018.

[138] U. Scherhag, C. Rathgeb, J. Merkle, and C. Busch. "Deep Face Representations for Differential Morphing Attack Detection." In: *IEEE Transactions on Information Forensics and Security* (2020), pp. 1–1.

[139] U. Scherhag et al. "Biometric Systems under Morphing Attacks: Assessment of Morphing Techniques and Vulnerability Reporting." In: *Proceedings of the 2017 International Conference of the Biometrics Special Interest Group (BIOSIG)*. IEEE, Sept. 2017.

[140] F. Schroff, D. Kalenichenko, and J. Philbin. "FaceNet: A unified embedding for face recognition and clustering." In: *Proceedings of the 2015 Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2015.

[141] C. Seibold, A. Hilsmann, and P. Eisert. “Reflection Analysis for Face Morphing Attack Detection.” In: *Proceedings of the 26th European Signal Processing Conference (EUSIPCO)*. 2018.

[142] C. Seibold, W. Samek, A. Hilsmann, and P. Eisert. “Detection of Face Morphing Attacks by Deep Learning.” In: *Proceedings of 2017 International Workshop on Digital Watermarking (IWDW): Digital Forensics and Watermarking*. Springer International Publishing, 2017, pp. 107–120.

[143] C. Seibold, W. Samek, A. Hilsmann, and P. Eisert. “Accurate and Robust Neural Networks for Security Related Applications Exampled by Face Morphing Attacks.” In: *arXiv e-prints* (June 11, 2018).

[144] F. Seide, G. Li, and D. Yu. “Conversational speech transcription using context-dependent deep neural networks.” In: *Proceedings of the 2011 Conference of the International Speech Communication Association (Interspeech)*. 2011, pp. 437–440.

[145] S. Sengupta, A. Kanazawa, C. D. Castillo, and D. W. Jacobs. “SfSNet: Learning Shape, Reflectance and Illuminance of Faces ‘in the Wild’.” In: *Proceedings of the 2018 Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2018.

[146] K. Simonyan and A. Zisserman. “Very Deep Convolutional Networks for Large-Scale Image Recognition.” In: *arXiv e-prints* (2014).

[147] D. Singh and B. Singh. “Investigating the impact of data normalization on classification performance.” In: *Applied Soft Computing* (2019), p. 105524.

[148] J. M. Singh, R. Raghavendra, K. B. Raja, and C. Busch. “Robust Morph-Detection at Automated Border Control Gate using Deep Decomposed 3D Shape and Diffuse Reflectance.” In: *Proceedings of the 15th International Conference on Signal-Image Technology & Internet-Based Systems (SITIS)*. Dec. 3, 2019.

[149] J. Snoek, H. Larochelle, and R. P. Adams. “Practical Bayesian Optimization of Machine Learning Algorithms.” In: *Proceedings of the 25th Advances in Neural Information Processing Systems (NIPS)*. Ed. by F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger. Curran Associates, Inc., 2012, pp. 2951–2959.

[150] S. Spelsberg. *Zwei Gesichter, ein Dokument*. Mar. 13, 2020. URL: https://taz.de/Peng-Kollektiv-faelscht-Passbilder/!5534868/.

[151] L. Spreeuwers, M. Schils, and R. Veldhuis. “Towards Robust Evaluation of Face Morphing Detection.” In: *Proceedings of the 26th European Signal Processing Conference (EUSIPCO)*. 2018.

[152] M. C. Stamm and K. J. R. Liu. “Forensic detection of image manipulation using statistical intrinsic fingerprints.” In: *IEEE Transactions on Information Forensics and Security* 5.3 (2010), pp. 492–506.

[153] Statista. *Biometric technologies - Statistics & Facts*. Mar. 10, 2020. URL: https://www.statista.com/topics/4989/biometric-technologies/.

[154] T. Stich, C. Linz, G. Albuquerque, and M. Magnor. “View and Time Interpolation in Image Space.” In: *Computer Graphics Forum* 27.7 (2008), pp. 1781–1787.

[155] J. Surowiecki. *The Wisdom of Crowds*. Anchor, 2005. ISBN: 8601300236001.

[156] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. “Going deeper with convolutions.” In: *Proceedings of the 2015 Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2015.

[157] J. Taylor. *Major breach found in biometrics system used by banks, UK police and defence firms*. Mar. 10, 2020. URL: https://www.theguardian.com/technology/2019/aug/14/major-breach-found-in-biometrics-system-used-by-banks-uk-police-and-defence-firms.

[158] C. E. Thomaz and G. A. Giraldi. “A new ranking method for principal components analysis and its application to face image analysis.” In: *Image and Vision Computing* 28.6 (2010), pp. 902–913.

[159] M. Turk and A. Pentland. “Face recognition using eigenfaces.” In: *Proceedings of the 1991 Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 1991.

[160] Utrecht University. *Utrecht ECVP*. European Conference on Visual Perception. 2008.

[161] S. Venkatesh, R. Raghavendra, K. Raja, L. Spreeuwers, R. Veldhuis, and C. Busch. “Morphed Face Detection Based on Deep Color Residual Noise.” In: *Proceedings of the 9th International Conference on Image Processing Theory, Tools and Applications (IPTA)*. IEEE, 2019.

[162] P. Viola, M. Jones, et al. “Rapid object detection using a boosted cascade of simple features.” In: *Proceedings of the 2001 Conference on Computer Vision and Pattern Recognition (CVPR)*. Vol. 1. 511-518. IEEE, 2001, p. 3.

[163] L. Wandzik, G. Kaeding, and R. V. Garcia. “Morphing Detection Using a General-Purpose Face Recognition System.” In: *Proceedings of the 26th European Signal Processing Conference (EUSIPCO)*. 2018.

[164] M. Wang and W. Deng. "Deep Face Recognition: A Survey." In: *arXiv e-prints* (2018).

[165] W. Wang, S. Shan, W. Gao, B. Cao, and B. Yin. "An improved active shape model for face alignment." In: *Proceedings of the 4th International Conference on Multimodal Interfaces (ICMI)*. IEEE, 2002.

[166] G. Wolberg. "Image morphing: a survey." In: *The Visual Computer* 14.8-9 (1998), pp. 360–372.

[167] E. Wu and F. Liu. "Robust image metamorphosis immune from ghost and blur." In: *The Visual Computer* 29.4 (2012), pp. 311–321.

[168] L. Yin, X. Wei, Y. Sun, J. Wang, and M. Rosato. "A 3D Facial Expression Database For Facial Behavior Research." In: *Proceedings of the 7th International Conference on Automatic Face and Gesture Recognition (FGR)*. IEEE, 2006.

[169] K. Zhang, W. Zuo, Y. Chen, D. Meng, and L. Zhang. "Beyond a Gaussian Denoiser: Residual Learning of Deep CNN for Image Denoising." In: *IEEE Transactions on Image Processing* 26.7 (2017), pp. 3142–3155.

[170] K. Zhang, Z. Zhang, Z. Li, and Y. Qiao. "Joint Face Detection and Alignment Using Multitask Cascaded Convolutional Networks." In: *IEEE Signal Processing Letters* 23.10 (2016), pp. 1499–1503.

[171] L.-B. Zhang, F. Peng, and M. Long. "Face Morphing Detection Using Fourier Spectrum of Sensor Pattern Noise." In: *Proceedings of the 2018 International Conference on Multimedia and Expo (ICME)*. IEEE, 2018.

[172] L. Zhang, M. Yang, and X. Feng. "Sparse representation or collaborative representation: Which helps face recognition?" In: *Proceedings of the 2011 International Conference on Computer Vision (ICCV)*. IEEE, 2011.

[173] R. Zhu, J. Yang, and R. Wu. "Iris Recognition Based on Local Feature Point Matching." In: *Proceedings of the 2006 International Symposium on Communications and Information Technologies (ISCIS)*. IEEE, 2006.

[174] A. Zwiesele, A. Munde, C. Busch, and H. Daum. "BioIS study. Comparative study of biometric identification systems." In: *Proceedings of the 34th Annual International Carnahan Conference on Security Technology*. IEEE, 2000.