

Projektarbeit

Deepfakes und Social Engineering

vorgelegt von

Julian Faigle (Matrikelnummer: 86292)
Studiengang ITS

Max Ernstschneider (Matrikelnummer: 86464)
Studiengang AIT

Semester 6



Hochschule Aalen

Hochschule für Technik und Wirtschaft

Betreut durch Prof. Roland Hellman

15.08.2024

Erklärung

Wir versichern, dass wir die Ausarbeitung mit dem Thema „Deepfakes und Social Engineering“ selbstständig verfasst haben und keine anderen Quellen und Hilfsmittel als die angegebenen benutzt haben. Die Stellen, die anderen Werken dem Wortlaut oder dem Sinn nach entnommen wurden, sind in jedem einzelnen Fall unter Angabe der Quelle als Entlehnung (Zitat) kenntlich gemacht worden. Das Gleiche gilt für beigelegte Skizzen und Darstellungen.

Aalen, den 11. August 2024

Ort, Datum

Julian Faigle

Autor

Max Ernstschneider

Autor

1 Kurzfassung

In den letzten Jahren haben technologische Fortschritte im Bereich der KI (Künstliche Intelligenz) und des maschinellen Lernens zu einer signifikanten Entwicklung in der digitalen Manipulation von Medien geführt. Insbesondere Deepfakes, bei denen KI verwendet wird, um realistisch aussehende Bilder, Videos oder Audiodateien zu erstellen, die es so nie gegeben hat, haben die Aufmerksamkeit von Forschern, Sicherheitsexperten und der breiten Öffentlichkeit gleichermaßen erregt. Diese Technologie, die ursprünglich zur Verbesserung von visuellen Effekten und für kreative Zwecke entwickelt wurde, hat sich mittlerweile zu einem potenzen Werkzeug entwickelt, das auch für bösartige Zwecke missbraucht werden kann.

Deepfakes basieren auf komplexen Algorithmen wie Generative Adversarial Networks (GANs), die es ermöglichen, Gesichter in Videos nahtlos auszutauschen, Stimmen zu imitieren oder Personen in Szenarien darzustellen, die nie stattgefunden haben. Diese Fähigkeit, die Realität auf eine Weise zu verzerren, die für das menschliche Auge oft schwer zu erkennen ist, birgt erhebliche Risiken für die Gesellschaft. Von der Verbreitung von Fehlinformationen über politische Manipulationen bis hin zu Erpressungen und Identitätsdiebstahl – die Einsatzmöglichkeiten von Deepfakes sind vielfältig und gefährlich.

Gleichzeitig hat Social Engineering, die Kunst der Manipulation von Menschen, um vertrauliche Informationen zu erhalten oder bestimmte Aktionen auszulösen, eine neue Dimension erreicht. Social Engineering nutzt gezielt psychologische Techniken, um das Vertrauen von Individuen zu gewinnen und sie dazu zu bringen, sicherheitsrelevante Fehler zu begehen. In einer zunehmend digitalisierten Welt, in der persönliche und berufliche Interaktionen häufig online stattfinden, wird die Verbindung zwischen Social Engineering und digitalen Technologien immer enger.

Die Kombination von Deepfakes und Social Engineering stellt eine besonders gefährliche Bedrohung dar. Angreifer können Deepfakes nutzen, um das Vertrauen ihrer Zielpersonen zu erschleichen oder deren Entscheidungen zu beeinflussen, indem sie manipulierte Inhalte einsetzen, die beispielsweise vertraute Gesichter oder Stimmen imitieren. Solche Angriffe sind schwer zu erkennen und zu bekämpfen, da sie sowohl technisches Wissen als auch ein tiefes Verständnis menschlicher Verhaltensmuster erfordern.

In dieser Ausarbeitung werden verschiedene Technologien im Bereich Video- und Audio-Deepfakes beleuchtet und vorgestellt. Sie werden in den Kontext von Social Engineering eingordnet und praktisch angewendet, um beispielhafte Szenarien darzustellen

Inhaltsverzeichnis

1 Kurzfassung	II
2 Stand der Technik	1
2.1 Deepfakes	1
2.1.1 Definition	1
2.1.2 Hintergrund	1
2.1.3 Theoretische Grundlagen	3
Autoencoder	3
Generative Adversarial Networks (GANs)	5
FSGAN und FSGANv2	5
Gemeinsamkeiten und Unterschiede	6
2.1.4 Arten von Deepfakes	7
2.1.5 Anwendungsgebiete	9
2.1.6 Ethik	11
2.2 Detection Methods für Video-Deepfakes	12
2.3 Social Engineering	13
2.3.1 Verschiedene Typen von Social Engineering	14
2.3.2 Überblick über gängige Angriffe	15
2.3.3 Gegenmaßnahmen gegen Social Engineering Angriffe	15
2.4 Bekannte Social Engineering Angriffe	17
3 Deepfake Varianten	19
3.1 Face Swapping und Reenactment	19
4 Erstellung von Deepfake Audios	21
4.1 Tacotron2	21
4.1.1 Motivation	21
4.1.2 Fähigkeiten	21
4.1.3 Workflow	22
4.2 Praxisbeispiel Tacotron2	22
4.2.1 Laborumgebung	23
4.2.2 Programmstruktur	23
4.2.3 Vorbereitung	24
4.2.4 Preprocessing	24
4.2.5 Extraktion	24

4.3	Praxisbeispiel Real-Time Voice Cloning	30
4.3.1	Laborumgebung	30
4.3.2	Programmstruktur	30
4.3.3	Vorbereitung	31
4.3.4	Extraktion	32
5	Erstellung von Deepfake Videos	34
5.1	DeepFaceLab	34
5.1.1	Motivation	34
5.1.2	Fähigkeiten	34
5.1.3	Workflow	34
5.2	Praxisbeispiel DeepFaceLab	36
5.2.1	Laborumgebung	37
5.2.2	Programmstruktur	37
5.2.3	Vorbereitung	37
5.2.4	Pretraining	38
5.2.5	Extraktion	41
5.2.6	Training	47
5.2.7	Conversion/Merging	53
5.3	DeepFaceLive	53
5.3.1	Motivation	53
5.3.2	Fähigkeiten	53
5.3.3	Workflow	54
6	Zusammenfassung	57

Akronyme

CLI	Command Line Interface
CNN	Convolutional Neural Network
DBIR	Data Breach Investigations Report
DFL	DeepFaceLab
DFLive	DeepFaceLive
FPS	Frames Per Second
FSGAN	Face Swapping GAN
GAN	Generative Adversarial Network
IDS	Intrusion Detection System
JIT	Just-in-Time
KI	Künstliche Intelligenz
LRCN	Long-Term Recurrent Convolutional Network
LSTM	Long Short-Term Memory
OBS	Open Broadcaster Software
OSINT	Open Source Intelligence
RDJ	Robert Downey Jr.
RTT	Real Time Transfer
SOTA	State-Of-The-Art

Glossar

AI-Upscaling	AI-Upscaling ist eine Technik, die künstliche Intelligenz verwendet, um die Auflösung von Bildern oder Videos zu erhöhen.
Enkeltrick	Ein betrügerisches Vorgehen, bei dem sich Trickbetrüger über das Telefon, neuerdings auch über Kontaktplattformen und Messengerdienste, meist gegenüber älteren und/oder hilflosen Personen, als deren nahe Verwandte ausgeben, um unter Vorspiegelung falscher Tatsachen an deren Bargeld oder Wertgegenstände zu gelangen
Magnituden-Spektrogramme	Visuelle Darstellungen der Energie von Audiosignalen über die Zeit, die für die Sprachsynthese verwendet werden.[1]
Max-Pooling	Max-Pooling ist ein auf Stichproben basierender Diskretisierungsprozess. Ziel ist es, eine Eingabedarstellung (Bild, Ausgabematrix der verborgenen Schicht usw.) zu verkleinern, um ihre Dimensionalität zu reduzieren und Annahmen über die in den gebinnten Unterregionen enthaltenen Merkmale treffen zu können.[2]
Motion Tracking	Motion Tracking ist eine Technik, die verwendet wird, um die Bewegung von Objekten oder Personen in einem Video oder einer animierten Szene zu verfolgen und zu verfolgen. Dies kann in 2D oder 3D erfolgen.
Sequenz-zu-Sequenz-Methode	Bei der Sequenzmodellierung, genauer gesagt beim Sequenz-zu-Sequenz-Lernen (Seq2Seq), handelt es sich um eine Aufgabe, bei der es darum geht, Modelle zu trainieren, um Sequenzen von einer Domäne (z.B. geschriebener Text) in eine andere Domäne (z.B. den gleichen Text, der zu Audio synthetisiert wurde) zu konvertieren.[3]

Threat Intelligence

Threat Intelligence sind Daten, die gesammelt, verarbeitet und analysiert werden, um die Motive, Ziele und das Angriffsverhalten eines Bedrohungskörpers zu verstehen. Durch Threat Intelligence können schnellere, fundiertere und datenbasierte Sicherheitsentscheidungen getroffen werden. Zudem ermöglicht es, das Verhalten im Kampf gegen Bedrohungskörper von reaktiv zu proaktiv zu ändern.[4]

Variational Auto-Encoder

VAEs sind ein Deep Learning-Modell, das verwendet wird, um das generative Modellieren von Daten zu erleichtern. VAEs versuchen, ein generisches Modell der Daten zu erstellen, indem sie ein generatives Modell aufbauen, das die Daten so gut wie möglich beschreibt.[5]

2. Stand der Technik

2.1 Deepfakes

2.1.1 Definition

Der Begriff Deepfake setzt sich aus den englischen Begriffen Deep Learning und Fake zusammen. Hierbei steht Deep Learning für eine Methode des maschinellen Lernens und Fake für eine Fälschung.

“Bei Deepfakes handelt es sich um einen Teilbereich synthetischer audiovisueller Medien: die Manipulation oder auch synthetische Erzeugung von Abbildungen, Videos und/oder Audiospuren menschlicher Gesichter, Körper oder Stimmen, zumeist mithilfe von KI.”^[6]

Deepfakes werden mit Hilfe von künstlicher Intelligenz und Deep Learning Technologien erstellt, um Personen realistische Handlungen ausführen oder Worte sagen zu lassen, in Form von Video, Bild oder Audio. Es handelt sich hierbei um gefälschte Darstellungen, die möglichst realitätsnah dargestellt werden.^[7]

2.1.2 Hintergrund

Deepfake ist eine Manipulationstechnik, die es Benutzern ermöglicht, das Gesicht einer Person mit einer anderen Person auszutauschen. Eine optimale Manipulation wird durch Verwendung mehreren Hunderten oder Tausenden Fotos der Zielperson erreicht. Das führt dazu, dass oft prominente Personen als Zielperson gewählt werden, da von ihnen viele Bilder im Internet existieren.

Bild- und Videomanipulationstechnologien bauen auf Techniken aus dem Bereich der künstlichen Intelligenz auf, welcher das Ziel verfolgt, menschliche Denkprozesse und Verhaltensweisen zu verstehen. Da maschinelles Lernen einem System ermöglicht aus Daten zu lernen, ist diese Technik wichtig für das erstellen von Deepfakes.

Deepfakes sind aus zwei Gründen beliebt: erstens wegen der Fähigkeit aus Daten wie Fotos und Videos, realistische Ergebnisse erzeugen zu können und zweitens die Verfügbarkeit der Technik, da diese für jeden leicht zu erreichen und durchzuführen ist. Es gibt Apps, welche die Schritte des Deepfakes-Algorithmus erklärt und so Personen mit wenig Kenntnissen über maschinelles Lernen oder Programmierung die Möglichkeit bietet ein Deepfake Bild oder Video zu erstellen.

Das führt zu einem Problem der heutigen Gesellschaft, da Deepfakes hauptsächlich aus Rache, Erpressung einer Person oder Verbreitung von Fake News einer höheren Person (bspw. eines Politikers) ausgenutzt werden.^[8]

Geschichte

Das Manipulieren von Bildern wurde nicht erst in den letzten Jahren bekannt. Denn auch schon früher wurden Bilder zum Beispiel von Hitler, Stalin, oder Breschnew mani-

puliert, um so die Geschichte zu ihren Gunsten verändern zu können. Damals erforderte es allerdings deutlich mehr Zeit und kompliziertere Techniken während der Fotoentwicklung in der Dunkelkammer, um ein Bild zu verfälschen. Doch durch die schnelle Entwicklung der Technologien wurde der Prozess ein Bild zu manipulieren zunehmend einfacher. Anfangs begannen ausschließlich Forscher der 1990er Jahre die Entwicklung der Deepfake-Technologie zu übernehmen, diese wurde jedoch später von Amateuren in den Online-Communities unterstützt. Die Akademiker Christoph Bregler, Michele Covell und Malcolm Slaney entwickelten 1997 ein Programm, welches vorhandenes Videomaterial einer sprechenden Person anpassen konnte, dass diese Person die Wörter von einer anderen Audiospur nachahmte. Das Programm baut auf einer älteren Technologie auf, welches bereits Gesichter interpretieren, Audio aus Texten synthetisieren und Lippen im 3D-Raum modellieren konnte. Jedoch war dieses entwickelte Programm von den drei Akademikern das erste, welches alle Komponenten zusammenfügen und überzeugend animieren konnte. So war es möglich eine neue Gesichtsanimation aus einer Audioausgabe zusammenstellen zu können.

Zu Beginn der 2000er Jahre wurde die Entwicklung der Gesichtserkennung mit dem Computer immer weiter vorangetrieben, sodass es zu großen Verbesserungen der Technologie wie Motion Trackings kam, welche die heutigen Deepfakes so überzeugend machen.

In den Jahren 2016 und 2017 gab es zwei Projekt Veröffentlichungen. Einmal das Face2Face-Projekt der Technischen Universität München und einmal das Synthesizing Obama-Projekt der University of Washington.

Das Face2Face Projekt versucht Echtzeitanimationen zu erstellen, indem es den Mundbereich des Zielvideos durch einen Schauspieler ersetzt, während das Synthesizing Obama-Projekt sich damit beschäftigte Videomaterial des ehemaligen Präsidenten Barack Obama zu modifizieren.^[9]

Im Jahr 2017 wurde das gefälschte Video des ehemaligen US-Präsidenten Barack Obama veröffentlicht und soll als Warnung der Technologie und deren potenziellen Auswirkungen gelten. Ende 2017 veröffentlichte ein Nutzer auf einer Webseite names Reddit pornografische Inhalte und behauptete, dass diese zu bekannten Personen wie zum Beispiel Taylor Swift oder Scarlett Johansson gehören. Auch wenn diese Bilder und Videos schnell wieder gelöscht wurden, erregte diese auf Deep Learning basierende Gesichtersatztechnik die Aufmerksamkeit der Medien und verbreitete sich in vielen Internetforen. Alle Inhalte, die mit der Deepfake Technik zu tun hatten, wurden am 7. Februar 2018 auf fast allen Internetforen entfernt und verboten. Trotz des Verbots hat sich die Technik dennoch weiterhin durchgesetzt und wurde weltweit verbreitet. Bei der Person, die die Deepfake-Technik entwickelt hat, soll es sich um einen Software-Ingenieur handeln, der ein Entwicklungs-Kit herausbrachte, mit dem es einem Benutzer selbst ermöglicht, eigene manipulierte Bilder oder Videos zu erstellen. Durch die Hilfe von Open Source Tools und Funktionen von großen Softwareunternehmen wie Nvidia und Google wurde die Deepfake-Technik entwickelt. Was bedeutet, dass für die Entwicklung technisches Wissen und Verständnis erforderlich sind, jedoch der Großteil der Software schon zuvor in der Öffentlichkeit zur Verfügung stand. Als klar wurde, dass selbst eine Person ohne viel Wissen in dem Gebiet, beliebig viele visuelle Medien manipulieren kann, wurde die

Bedrohung der Deepfake-Technik ernst und das US-Verteidigungsministerium schaltete sich ein. Auch im Jahr 2018 wurde ein Deepfake Video von damaligen Präsidenten Donald Trump in den Medien hochgeladen, in dem die Belgier aufgefordert wurden, aus dem Pariser Klimaschutzabkommen auszusteigen.

Durch solche Veröffentlichungen der Deepfake Videos zeigte sich, dass die Technologie sich schnell weiterentwickelt und in der Lage ist einen großen Teil der Öffentlichkeit täuschen zu können.[8]

2.1.3 Theoretische Grundlagen

Wie in fast allen Modellen der künstlichen Intelligenz, wird auch bei den Deepfakes Ausgangsmaterial benötigt. Das kann zum Beispiel Videomaterial oder Bilder von Personen sein, die man manipulieren möchte. Auch hier gilt je mehr Ausgangsmaterial vorhanden ist, desto besser erfolgt das Training des Modells. Bei einer Video Manipulation wird das Ergebnis besser, je mehr Videomaterial aus verschiedenen Perspektiven und verschiedenen Mimiken vorhanden ist.[10]

Autoencoder

Ein Ansatz zur Erstellung von Deepfakes ist die Autoencoder Variante. Dieses neuronale Netz erzeugt aus einem Eingabebild ein digitales Abbild (Encoding), indem es durch ein Verfahren wie Max-Pooling verkleinert und auf wesentliche Kernmerkmale des Bildes reduziert. Im Anschluss wird versucht, das codierte Bild dann möglichst originalgetreu wiederherzustellen (Decoding). Das Ziel des Decoding ist es also, ein identisches, künstliches Abbild des codierten Eingabebildes zu erzeugen. Um die Robustheit des Autoencoders zu erhöhen, ist es ratsam die Eingabebilder zu verzerrn, Auflösung zu variieren und die Größe zu verändern. Über längere Zeit des Trainings wird das Netz immer präziser, sodass der Unterschied zwischen Eingangs- und Ausgangsbild verschwindet gering wird. Um zwei Personen zu tauschen, wird das Autoencoder Verfahren auf beide Personen angewendet. Der Encoder bleibt hier der gleiche für beide Personen und für den Decoder gibt es 2 verschiedene. Während des Trainings des Deepfakes wird zunächst das Bildmaterial von Person A encodiert und anschließend mit Decoder A nachgestellt. Danach folgt dasselbe Training mit Person B und Decoder B. Nach dem Training beider Personen wird der Decoder A durch den Decoder B ersetzt, wodurch dann der Deepfake entsteht.[10]

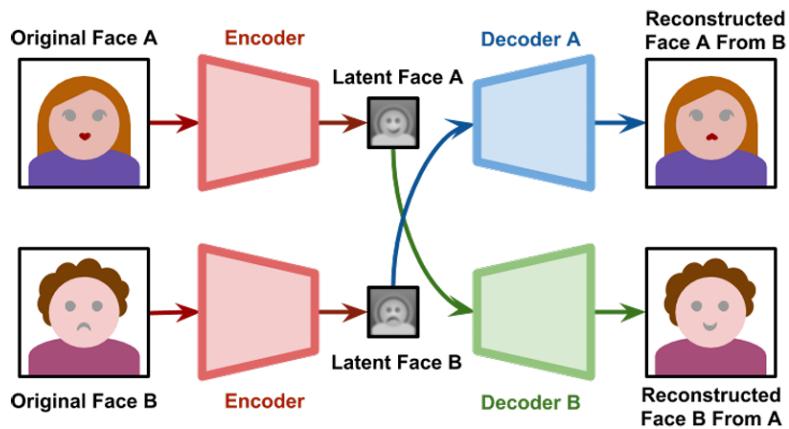


Abbildung 2.1: Darstellung des Autoencoders zur Erstellung eines Deepfakes[10]

Funktion des Autoencoder:

Autoencoder entdecken latente Variablen. Hierfür leiten sie die Eingabedaten durch einen Engpass (Code), bevor sie den Decoder erreichen. Dieser Engpass zwingt den Encoder zu lernen, dass nur die Informationen extrahiert und weitergeleitet werden, die für eine präzise Rekonstruktion der ursprünglichen Eingabe am nützlichsten sind.

Der Encoder umfasst hierbei mehrere Schichten, welche eine komprimierte Darstellung der Eingabedaten durch Dimensionreduktion erzeugen. In einem typischen Autoencoder reduzieren die verborgenen Schichten die Anzahl der Knoten schrittweise im Vergleich zur Eingabeschicht. Während die Daten die Encoderschichten durchlaufen, werden sie durch einen Prozess (Strauchung) in weniger Dimensionen komprimiert.

Der Engpass (auch Code genannt) besitzt die stärkste komprimierte Darstellung der Eingabe. Er wird sowohl als Ausgangsschicht des Encodernetzes als auch die Eingabeschicht des Decodernetzes betrachtet. Eines der Ziele des Entwurfs und Trainings eines Autoencoders besteht darin, die kleinstmögliche Anzahl der relevantesten Merkmale zu ermitteln, welche für eine effektive Rekonstruktion der Eingabedaten erforderlich sind. Der austretende Code (Latent-Raum-Darstellung) dieser Schicht, wird in dem Decoder weitergeleitet und dort weiter verarbeitet.

Der Decoder besteht aus verborgenen Schichten, von dem die Anzahl der Knoten zunehmend wächst, um so die codierte Darstellung der Daten zu decodieren und in die ursprüngliche Form vor der Codierung wiederherzustellen. Die rekonstruierte Ausgabe wird mit der "Ground Truth", in den meisten Fällen die ursprünglichen Eingabe, verglichen, um die Wirksamkeit des Autoencoders zu bewerten.

In vielen Anwendungen von Autoencodern kann der Decoder nach dem Training gelöscht werden, da in solchen Fällen der Decoder nur dazu da war den Encoder zu trainieren. In einem GAN (Generative Adversarial Network) ist dieser Vorgang zu vergleichen mit der Rolle des Diskriminators, welcher dann als Komponente eines anderen neuronalen Netzes verwendet wird. Jedoch erfüllt der Decoder in vielen Autoencodern nach dem Training weiterhin einen Zweck, wie zum Beispiel das Ausgeben neuer Datenproben in Variational

Auto-Encoders.[11]

Generative Adversarial Networks (GANs)

Mit dem Aufkommen von GANs hat sich die Erstellung von Deepfakes erheblich vereinfacht und verbessert. GANs bestehen aus zwei neuronalen Netzwerken, einem Generator und einem Diskriminatator, die gegeneinander trainiert werden.[12]

Technische Details:

- Der **Generator** erzeugt Bilder, die versuchen, echte Bilder nachzuahmen.
- Der **Diskriminatator** bewertet die Bilder des Generators und unterscheidet zwischen echten und generierten Bildern.
- Durch diesen Wettbewerb verbessert sich die Qualität der erzeugten Bilder stetig.

Vor- und Nachteile:

- **Vorteile:** Hohe Qualität der erzeugten Bilder, Möglichkeit zur Erstellung realistischer und dynamischer Gesichtsausdrücke.
- **Nachteile:** Erfordert große Datenmengen und Rechenressourcen für das Training, anfällig für Artefakte und Unstimmigkeiten bei komplexen Bewegungen [13].

Einsatz im Kontext von Social Engineering: GAN-basierte Deepfakes sind effektiver und leichter zu erstellen, was sie zu einem mächtigen Werkzeug für Angreifer im Bereich Social Engineering macht. Sie können verwendet werden, um gefälschte Videos zu erstellen, die Vertrauen erwecken und die Opfer täuschen. Um ein Modell zu trainieren werden mehrere Minuten Video-Material der Zielperson und viel Rechenleistung benötigt. Ist das Modell dann trainiert, können je nach Auflösung Videos JIT (Just-in-Time), auch auf einem schwächeren PC, manipuliert werden. GANs, in Kombination mit Autoencodern sind heutzutage nach wie vor SOTA (State-Of-The-Art) und machen den Großteil der erstellten Deepfakes aus. Üblicherweise wird ein Deepfake mithilfe eines Autoencoders erstellt und anschließend wird der Output durch ein GAN verbessert und detailreicher. Auch DFL (DeepFaceLab) basiert auf dieser Technik und macht laut Entwickler ca. 95% der im Internet kursierenden Deepfakes aus[14].

FSGAN und FSGANv2

Eine Weiterentwicklung der GAN-Technologie sind die Face Swapping GANs (FSGAN (Face Swapping GAN)) und ihre verbesserte Version, FSGANv2. Diese Technologien sind speziell für den Gesichtstausch und die Gesichtsnachstellung entwickelt worden.

Technische Details:

- FSGAN nutzt GANs, um Gesichtszüge von einer Quelle auf ein Zielvideo zu übertragen, ohne dass eine explizite 3D-Modellierung erforderlich ist.
- FSGANv2 verbessert diese Methode durch bessere Algorithmen zur Anpassung der Gesichtsausdrücke und -bewegungen sowie durch die Verwendung von fortschrittlichen Netzwerktechniken, um die Konsistenz und Realismus zu erhöhen [15].

Vor- und Nachteile:

- **Vorteile:** Sehr realistische Ergebnisse, weniger Training und Daten erforderlich im Vergleich zu reinen GANs, bessere Anpassung an unterschiedliche Gesichtsausdrücke und Beleuchtungen.
- **Nachteile:** Trotz Verbesserungen immer noch anfällig für subtile Unstimmigkeiten, die bei genauer Betrachtung auffallen können [16].

Einsatz im Kontext von Social Engineering: FSGAN und FSGANv2 sind äußerst effektiv für Social Engineering-Angriffe, da sie schon mit wenigen Ausgangsbildern realistische Deepfakes erstellen können. Sie können verwendet werden, um falsche Identitäten zu erstellen und Vertrauen zu gewinnen, was das Risiko und den potenziellen Schaden solcher Angriffe erhöht [17].

Gemeinsamkeiten und Unterschiede

Gemeinsamkeiten:

- Alle Methoden zielen darauf ab, realistische Fälschungen zu erstellen, die schwer zu erkennen sind.
- Nutzung von KI und maschinellem Lernen zur Verbesserung der Qualität und Realismus der erzeugten Videos.

Unterschiede:

- Autoencoder bieten die Grundlage von Deepfakes, ihr Ergebnis ist zufriedenstellend, kann aber durch GANs verbessert werden.
- GANs und deren Weiterentwicklungen (FSGAN, FSGANv2) bieten automatisierte Lösungen mit höherer Qualität und Flexibilität.
- FSGAN und FSGANv2 sind speziell auf Gesichtsmanipulation optimiert und bieten bessere Ergebnisse bei der Anpassung an verschiedene Bedingungen[12].

Insgesamt zeigt sich, dass die Weiterentwicklung der Deepfake-Technologien, insbesondere durch GANs und deren Spezialisierungen wie FSGAN und FSGANv2, erheblich zur Verbesserung der Qualität und Realismus beigetragen hat. Dies stellt jedoch auch eine größere Bedrohung im Bereich des Social Engineering dar, da die Täuschungsabsicht hinter den erzeugten Medieninhalten immer schwerer zu durchschauen ist.

2.1.4 Arten von Deepfakes

Deepfakes können in drei Hauptarten unterteilt werden: Video Deepfakes, Audio Deepfakes und Foto Deepfakes. Diese drei Arten lassen sich zusätzlich auch noch miteinander kombinieren.[9]

Video Deepfakes

Bei Video Deepfakes wird zusätzlich zwischen 3 Arten der Manipulation unterschieden. Auf welche Art der Manipulation zurückgegriffen wird, ist davon abhängig, was der Hauptgrund der Nutzung eines Video Deepfakes ist.[9]

Face Swapping

Eine der Arten ist das Face Swapping, bei dem die Gesichter auf Bildern oder Videos durch Fake Gesichter oder Gesichter anderer Personen, wie zum Beispiel eines Promis, ersetzt wird. Dadurch ist es möglich die Person, dessen Gesicht verwendet wird, in einen anderen Kontext darstellen zu lassen, um beispielsweise in der Filmindustrie den Schauspieler mit einem Stunt Double austauschen zu können, um bestimmte Actionszenen realistischer wirken zu lassen.[9]

Face Morphing

Die zweite Art von Video Deepfakes ist das Face Morphing, welches ein Spezialeffekt ist, um ein Bild oder eine Form durch einen nahtlosen Übergang in ein anderes verändern zu können. Dieser Effekt wird oft in Filmen oder Animationen verwendet.[9]

Reenactment

Reenactment (auch face transfer or puppeteering genannt) ist eine Technik, bei der die Gesichtsausdrücke eines Source Images entsprechend eines Target Images angepasst werden. Das bedeutet, dass die Gesichtsausdrücke und (Lippen-, Augen-, usw.) Bewegungen der Ursprungsperson mit denen einer anderen Person ersetzt werden.



Abbildung 2.2: Reenactment Ergebnisse eines FSGAN[16]

Full body puppetry

Die letzte Art von Video Deepfakes ist die Full body puppetry, bei der einzelne Bewegungen bis hinzu komplette Bewegungsabläufe auf eine andere Person übertragen werden. Die meisten Deepfakes benötigen viel Zeit für die Erstellung aufgrund der Systeme, welche erst mit dem Ausgangsmaterial trainiert werden müssen, um danach Inhalte verändern zu können. Es gibt aber auch Deepfake-Methoden die in Echtzeit funktionieren, welche die Möglichkeit bietet, Mimik und Lippenbewegungen einer Person zu erkennen und diese anschließend in Echtzeit auf das Videobild einer anderen Person übertragen zu lassen.[9]

Audio Deepfakes

Eine andere Art von Deepfakes sind Audio Deepfakes, bei dem aufgenommene oder live Audio Dateien verändert werden. Wobei hier zwischen Voice Swapping und Text to Speech unterschieden wird.[9]

Voice-swapping

Bei dem Voice-swapping können Audioinhalte so verändert werden, dass ein Text von einer fremden Person gesprochen werden kann. Die Stimme kann mit verschiedenen Effekten verändert werden, sodass zum Beispiel eine Stimme jünger, älter, männlich, weiblich oder auch mit verschiedenen Dialekten versehen werden kann. Dadurch wird dem Hörer vorgespielt, dass verschiedene Personen sprechen, wobei es sich aber nur um eine Person handelt.[9]

Text to Speech

Beim Text to Speech können Audioinhalte einer Aufnahme durch Eingabe eines neuen Textes verändert werden. Dadurch können zum Beispiel falsch ausgesprochene Wörter im nachhinein ersetzt werden, ohne eine neue Aufnahme durchführen zu müssen.[9]

Foto Deepfakes

Die dritte Art der Deepfakes sind Foto Deepfakes, bei denen es sich darum handelt, Fotos zu manipulieren. Dadurch können Fotos nach belieben verändert werden, um beispielsweise eine Person auf dem Bild durch einen Alterungsfilter, den Alterungsprozess der Person dargestellt werden kann.[9]

Face and body-swapping

Mithilfe des Deepfake-Algorithmus, welcher auch bei den anderen Arten verwendet wird, können Änderungen an einem Gesicht und Körper gemacht werden, indem das Gesicht oder der Körper mit einer anderen Person ausgetauscht wird. Eine mögliche Anwendung hierfür wäre das virtuelle anprobieren einer Brille, Haarfarbe oder Kleidung.[9]

Kombination aus Audio und Video Deepfake

Zuletzt gibt es wie oben eine mögliche Kombination der verschiedenen Arten, wie zum Beispiel die Kombination aus Audio und Video Deepfake. Diese Kombination wird auch das Lip-syncing genannt, bei dem Mundbewegungen sowie die gesprochenen Wörter in einem Video verändert und synchronisiert werden. Dadurch ist es möglich eine Person in einem Video scheinbar etwas sagen zu lassen, was sie aber niemals gesagt hat. Dies kann sowohl stark Missbraucht werden, indem zum Beispiel einem Politiker eine falschaussage untergeschoben wird. Es kann aber auch für positive Sachen Verwendung finden, um beispielsweise einen Film oder Werbung in eine andere Sprache zu synchronisieren.[9]

2.1.5 Anwendungsgebiete

Positive Anwendungsgebiete

Deepfakes können in den künstlerischen, satirischen, pädagogischen und sozialen Bereich positiv verwendet werden.

So war es zum Beispiel möglich durch Deepfake-Technologien bekannte Persönlichkeiten wie Albert Einstein oder die Mona Lisa wieder zum Leben zu erwecken. Damit ist gemeint, dass so die bekannten Persönlichkeiten als Videos erstellt werden und dadurch besser in den Kontext gestellt werden können, um ein bestimmtes Thema in der Schule oder im

Studium anschaulicher darstellen zu können.

Es gibt auch zahlreiche Anwendungen von Deepfake in der Filmbranche, wie z.B. den Schauspieler mit dem Stunt double auszutauschen, verstorbene Schauspieler nochmals darstellen zu lassen oder auch bei der Synchronisation von verschiedenen Sprachen und den Mundbewegungen der Schauspieler. Auch für Modeunternehmen sind Deepfake Technologien hilfreich, so kann der potenzielle Kunde zum Beispiel online bereits die Kleidung virtuell anprobieren und dann entscheiden, ob es ihm gefällt oder nicht.

Künstlich generierte Gesichter wird auch als Alternative zur Unkenntlichmachung eingesetzt, um Privatsphäre und persönliche Daten (z.B. vor biometrischer Identifizierung) zu schützen.

Text-To-Speech als Deepfakes sind sehr hilfreich, um Personen die ihre Sprachfähigkeit verloren haben durch einer Krankheit, kann so ihre Stimme zurückgegeben werden. Hierfür wird aus gesprochenen Sätzen der Person ein digitaler Klon der Stimme erstellt. So kann ein Computer dann eingegebene Texte mit der synthetisierten Stimme der erkrankten Person ausgeben.[6]

Negative Anwendungsgebiete

Genauso wie es positive Anwendungsgebiete gibt, existieren natürlich auch einige negative Anwendungsbereiche für Deepfakes.

- Politik und Regierung:**

Gerade in der Politik und Regierung Ebene können Deepfakes als Propaganda genutzt werden oder auch um Politiker in Videos oder Bilder falsch dazustellen und so Wahlen manipuliert werden. Dadurch entsteht bei der Regierung die Angst, dass Deepfakes eine Gefahr für die Demokratie sein könnte.[9]

Auch Diplomatische Beziehungen können negativ beeinflusst werden, so wurde zum Beispiel Deepfake-Videos vom ukrainischen und russischen Präsidenten veröffentlicht, indem der ukrainische Präsident zur Kapitulation vor den Angriffen Russlands aufruft und der russische Präsident behauptet der Krieg sei beendet und es wäre Frieden zwischen beiden Ländern. Eine solche Fake Information, welche durch die Deepfakes sehr realistisch erscheint, kann zu weitreichenden Konsequenzen auf der ganzen Welt führen. Auch gefälschte Videos von Soldaten oder Polizisten die unbewaffnete Zivilisten erschießen oder eine Ankündigung einer Katastrophe, wie z.B. einen Raketenangriff, kann zum Ausbruch von Protesten, Gewalt oder Massenpanik führen.[6]

- Wirtschaft:**

Deepfakes können auch Privatpersonen und Unternehmen schädigen, indem sie erpresst werden und zum Beispiel eine Entführung vortäuschen, um so Lösegeld zu fordern, ohne dass das vermeintliche Opfer sich in der Gewalt des Erpresser befindet.

Auch CEO-Frauds, bei denen dem Mitarbeiter eines Unternehmens vorgegaukelt wird, dass eine Führungsperson eine Überweisung eines hohen Geldbetrags oder

ähnliches verlangt, werden immer realistischer. Darüber hinaus können synthetische Medien auch zur Sabotage anderer Personen eingesetzt werden, um so ihren privaten und auch beruflichen Werdegang zu schädigen und künftige Geschäftschance zu verhindern.[6]

- **Erstellung künstlicher Identitäten:**

Durch den Einsatz realistischer synthetischer Fotos können künstliche Identitäten, die real nicht existieren und komplett künstlich erzeugt wurden, so glaubhaft dargestellt werden, dass diese für Authentifizierungsverfahren oder Spear Phishing Operationen eingesetzt werden können. Dadurch werden massenhafte Fake Profile auf Social Media erstellt, um andere User in Bezug auf private oder politischen Inhalte zu täuschen.[6]

- **Mobbing:**

Auch Mobbing wird durch Deepfakes mehr Möglichkeiten gegeben, da jeder Mensch so leicht wie noch nie in lächerliche, gefährliche oder kompromittierende Szenarien gebracht werden kann.[9]

2.1.6 Ethik

Die Entwicklung von Deepfakes hat eine erhebliche Auswirkung auf die gesamte Welt. Deepfakes werden aus den verschiedensten Gründen verwendet, sei es für den Tausch von Gesichtern oder für eine Änderung von Stimmen.

Was jedoch besonders beunruhigend daran ist, ist die Erstellung und Verbreitung von realistischen Fake News. Es wurde festgestellt, dass im Hinblick auf die ethischen Verpflichtungen dieser Technologie, Deepfakes trotz der positiven Anwendungsbereiche überwiegend für bösartige Aktivitäten genutzt werden.

Ein großes ethisches Problem im Zusammenhang mit Deepfake, ist die Verletzung der Privatsphäre. Nahezu jede digitale menschliche Spur kann gefälscht werden, was zu einer Bedrohung für die Privatsphäre führt. Ein weiterer Punkt ist die Zuverlässigkeit der Nachrichtensender, da die ausgestrahlten Nachrichten verfälscht sein könnten und alle Informationen in Frage gestellt werden müssen, ob sie nun der Wahrheit entsprechen oder nicht.

Dadurch hat sich die größte ethische Herausforderung gebildet im Zusammenhang mit Deepfakes. Diese Herausforderung besteht darin, dass schädliche Aktivitäten, wie die Fälschung und Verbreitung von gefälschten Informationen, immer häufiger werden und negative Auswirkungen auf die Gesellschaft haben.

Auch Arbeitnehmer können sich in unethische Praktiken begeben, wie zum Beispiel die Entwicklung gefälschter Fotos und Videos eines Kollegen zu Unterhaltungs- oder Rachezwecken. Deshalb erfordert es eine kontinuierliche Bewertung der ethischen Aspekte des Arbeitsplatzes, um solche Taten zu umgehen. Es wurde unter anderem festgestellt, dass die Nutzer durch der Entwicklung des Internets, leichten Zugang zu den Erstellungen von Deepfakes und dessen erforderlichen Werkzeugen erhalten. Das führt dazu, dass kein System, weder technologisch noch rechtlich, die Verwendung von Deepfakes stoppen kann,

und dass aus ethischer und moralischer Sicht ein kontinuierlicher Schaden zu erwarten ist.[8]

2.2 Detection Methods für Video-Deepfakes

Die rasante Verbreitung von Deepfake-Inhalten stellt eine erhebliche Bedrohung für die Privatsphäre, die soziale Sicherheit und die Integrität des Internets dar. Um diesen Bedrohungen entgegenzuwirken, sind effektive Erkennungsmethoden unerlässlich. Verschiedene Techniken wurden entwickelt, um die Authentizität von Videos zu prüfen und Deepfakes zu identifizieren.

Temporale Sequenzanalyse

Eine der gängigsten Methoden zur Erkennung von Deepfakes ist die temporale Sequenzanalyse. Diese Technik nutzt die Fähigkeit von LSTM (Long Short-Term Memory) Netzwerken und CNNs (Convolutional Neural Networks), um zeitliche Unstimmigkeiten zwischen den Frames eines Videos zu erkennen. Durch die Analyse der Sequenzen von Frames können LSTM-Netzwerke zusammen mit CNNs Muster identifizieren, die auf Deepfake-Manipulationen hinweisen. Hierbei extrahieren die CNNs eine Vielzahl von Merkmalen aus jedem Frame und übergeben diese an die LSTM-Netzwerke, die eine temporale Sequenzbeschreibung erzeugen. Eine SoftMax-Schicht berechnet schließlich die Wahrscheinlichkeit, dass die analysierten Frames Deepfakes sind[13].

Blinzelmuster Erkennung

Eine weitere Methode basiert auf der Analyse der Augenblinzelmuster in Videos. Deepfake-Videos weisen oft unnatürliche Blinzelraten auf, da das Blinzeln in den synthetisierten Videos schlecht dargestellt wird. Hierfür wird das Video in einzelne Frames konvertiert und die Augenbereiche werden extrahiert. Diese Augenbereich-Sequenzen werden durch LRCNs (Long-Term Recurrent Convolutional Networks) verarbeitet, um die Wahrscheinlichkeit der Augenöffnungs oder -schließzustände vorherzusagen. Diese Methode ist besonders effektiv, da menschliche Blinzelmuster schwer nachzuahmen und synthetisierbar sind[13].

Physiologisch basierte Erkennungsmethoden

Physiologisch basierte Erkennungsmethoden nutzen Unterschiede zwischen computer-generierten Gesichtern und realen menschlichen Gesichtern. Ein Beispiel hierfür ist die Analyse von Blutflussmustern im Gesicht, die in Deepfake-Videos oft fehlen oder unnatürlich dargestellt werden. Solche physiologischen Signale können mit spezieller Software

erfasst und analysiert werden, was eine hohe Genauigkeit bei der Erkennung von subtilen Unterschieden zwischen echten und gefälschten Gesichtern ermöglicht[18].

Digitale Wasserzeichen und Blockchains

Digitale Wasserzeichen und Blockchain-Technologien bieten ebenfalls effektive Möglichkeiten zur Authentifizierung von Videoinhalten. Digitale Wasserzeichen können in Videos eingebettet werden und gehen bei Manipulationen verloren. Die Blockchain-Technologie kann verwendet werden, um digitale Signaturen zu speichern und die Verbreitung von Videos zu verfolgen. Dies bietet eine hohe Sicherheit und Transparenz bei der Verifizierung der Authentizität von Videoinhalten, erfordert jedoch einen hohen Implementierungsaufwand[18].

Echtzeiterkennung durch Augenspiegelung

Aktive forensische Methoden sind besonders nützlich für die Echtzeit-Erkennung von Deepfakes, beispielsweise bei Videoanrufen. Diese Methoden nutzen spezifische Muster oder Reize, um Deepfakes in Echtzeit zu erkennen. Ein Beispiel ist die Anzeige eines unverwechselbaren Musters auf dem Bildschirm und die Analyse der kornealen Reflexion im Auge des Gesprächspartners. Solche biometriebasierten Ansätze sind effektiv für die Echtzeit-Erkennung und bieten eine robuste Lösung zur Verhinderung von Deepfake-Angriffen in Videokonferenzen[19].

Insgesamt bieten die verschiedenen Erkennungsmethoden für Video-Deepfakes eine breite Palette von Ansätzen zur Identifizierung und Authentifizierung von Inhalten. Die kontinuierliche Weiterentwicklung dieser Technologien ist entscheidend, um den immer leistungsfähigen Deepfake-Techniken entgegenzuwirken.

2.3 Social Engineering

Unter Social Engineering werden alle Angriffe zusammengefasst, die die Schwachstelle Mensch ausnutzen. Es wird durch verschiedene Techniken versucht, an private oder sensible Inhalte von Personen zu gelangen. Social Engineering ist heutzutage eine der größten Gefahren im digitalen Raum. Kryptografische Verfahren und Protokolle wurden über die Jahre immer weniger anfällig für Exploits. Ist ein System bzw. eine digitale Infrastruktur richtig gehärtet, sind Angriffe wie Brute-Force- oder Dictionary-Attacken wirkungslos. Außerdem werden durch moderne IDSs (Intrusion Detection Systems), sowie Threat Intelligence technische Angriffe immer schneller erkannt und blockiert. Social Engineering Angriffe eine höhere Erfolgschance als herkömmliche Methoden. Laut des DBIR (Data Breach Investigations Report) von Verizon waren 2024 45% der erfassten

Cyberangriffe auf Social Engineering zurückzuführen. Bei 83% der 3661 reporteten Incidents wurden Daten extrahiert.[20]

2.3.1 Verschiedene Typen von Social Engineering

Social Engineering lässt sich in verschiedene Bereiche untergliedern, die folgenden Beschreibungen richten sich nach dem Artikel: "Social Engineering Attacks: A Survey"[21]

Hier werden Social Engineering Angriffe in zwei Kategorien unterteilt.

Human-based: Diese Angriffe werden manuell von einem Menschen ausgeführt. Sie sind in der Regel spezifisch auf das Opfer angepasst und mit höherem Aufwand verbunden. Dafür sind die Erfolgsschancen höher als bei automatisierten Angriffen.

Computer-based: Diese Angriffe werden automatisiert durchgeführt. Sie sind qualitativ deutlich schlechter als ihr Gegenstück, dafür werden sie in hoher Quantität durchgeführt. Hierzu zählen Phishing-Mails oder SMS. Es gibt verschiedene Tools, um solche Angriffe durchzuführen, ein bekanntes ist das "[Social Engineering Toolkit](#)".

Des Weiteren können Social Engineering Angriffe in drei weitere Kategorien unterteilt werden.

Social-based: Diese Form von Social Engineering Angriffen besteht aus zwischenmenschlicher Interaktion. Dabei spielt sie mit der Psychologie und den Emotionen der Zielperson. Diese Form von Social Engineering birgt ein hohes Risiko, hat aber ebenfalls eine hohe Erfolgsschance, da der Angreifer im direkten oder indirekten Kontakt mit dem Opfer steht. Beispiele hierfür wären: Baiting, Spear-Phishing, aber auch nicht technische Methoden wie der Enkeltrick.

Technical-based: Hier werden Angriffe über Internet remote ausgeführt. Dafür werden Social-Media Plattformen und Online-Dienste verwendet, um Passwörter, Kreditkarteninformationen oder personenbezogene Daten zu stehlen. Hierzu zählen zum Beispiel Phishing-Kampagnen oder gefälschte Webseiten.

Physical-based: Physical-based Angriffe geschehen abseits des Internets in der realen Welt. Dabei werden durch physisches Handeln Informationen erschlossen. Ein Beispiel wäre das Durchsuchen von Müllcontainern (auch Dumpster-Diving genannt) nach sensiblen Dokumenten.

Je nachdem, aus welchem Blickwinkel die verschiedenen Techniken des Social Engineerings betrachtet werden, können diese in noch mehr verschiedene Kategorien eingeteilt werden. Neben Human-, Computer-, Social-, Technical- und Physical-based Social Engineering ist die zusätzliche Unterscheidung in **direkt** und **indirekt** sinnvoll. Ersteres benötigt direkten Kontakt zwischen Angreifer und Opfer, dabei zählen physischer Kontakt sowie Telefonate. Beispiele sind: physical access, shoulder surfing, dumpster diving, phone social engineering, pretexting und impersonation on help desk call. Indirekte Angriffe sind entsprechend analog dazu. Hierzu zählen: phishing, fake software, Pop-Up windows, ransomware und SMSishing.

2.3.2 Überblick über gängige Angriffe

Phishing ist eine der am weitesten verbreiteten Social Engineering-Techniken. Ziel dieser Angriffe ist es, private oder vertrauliche Daten der Opfer zu stehlen. Dabei werden hauptsächlich E-Mails, SMS, Anrufe oder gefälschte Webseiten eingesetzt, um die Opfer zur Preisgabe ihrer Informationen zu verleiten. Phishing lässt sich grob in folgende Kategorien unterteilen[22]:

- **Spear Phishing:** Diese Methode ist zielspezifisch und verwendet oft durch OSINT (Open Source Intelligence) gesammelte Informationen, um maßgeschneiderte E-Mails zu erstellen. Die Nachrichten wirken dadurch besonders glaubwürdig und erhöhen die Erfolgsschancen des Angriffs.
- **Whaling Phishing:** Hierbei handelt es sich um Angriffe auf hochrangige Ziele, wie Führungskräfte oder Personen in Schlüsselpositionen. Diese Angriffe sind oft sehr aufwendig und spezifisch auf das Ziel zugeschnitten, um wertvolle Informationen zu erlangen.
- **Vishing:** Voice Phishing, bei dem Telefonanrufe oder Sprachdienste wie Teams genutzt werden, um sensible Informationen zu erlangen. Die Angreifer geben sich häufig als vertrauenswürdige Institutionen oder Personen aus, um das Opfers zu verleiten Informationen zu teilen.

Eine weitere Technik ist **Baiting**. Baiting, auch als Road Apples bekannt, verleitet Personen dazu, auf etwas zu klicken oder ein Gerät zu benutzen, um vermeintlich etwas gratis zu erhalten. Ein bekanntes Beispiel hierfür sind E-Mails mit einem Gewinn, für den man sich nur noch registrieren braucht, um ihn zu erhalten. Außerdem gehören auch infizierte USB-Sticks, die in der Hoffnung verteilt werden, dass jemand sie benutzt, zu Baiting dazu. Bei Bad-USBs wird auf die Neugierde des Menschen gesetzt. Durch das Einsticken des USB-Sticks in einen Computer kann Schadsoftware installiert werden, die es den Angreifern ermöglicht, auf das System zuzugreifen.

Tailgating Attacks beziehen sich auf das unerlaubte Verschaffen von Zutritt zu gesicherten Bereichen, indem zum Beispiel einer autorisierten Person gefolgt wird. Auch Angriffe auf die Sicherheitsmechanismen, wie z.B. das Kopieren eines NFC- oder RFID-Tags gehören in diese Kategorie. Solche Angriffe ermöglichen es dem Angreifer, physisch gesicherte Bereiche zu betreten und dort Informationen zu stehlen oder Schaden anzurichten.

2.3.3 Gegenmaßnahmen gegen Social Engineering Angriffe

Die Abwehr von Social Engineering Angriffen erfordert eine Kombination aus präventiven und reaktiven Maßnahmen. Eine der effektivsten Präventionsstrategien ist die **Schulung der Mitarbeiter**. Durch regelmäßige Schulungen und Sensibilisierungsprogramme können Mitarbeiter lernen, die Anzeichen von Social Engineering Angriffen zu erkennen und

angemessen darauf zu reagieren. Dies umfasst das Überprüfen der Authentizität und Integrität von Nachrichten, sei es per E-Mail, SMS oder Telefon.

Überprüfung der Authentizität und Integrität von Nachrichten: Es sollte stets darauf geachtet werden, ungewöhnliche oder verdächtige Nachrichten sorgfältig zu prüfen. Dazu gehört, den Absender zu überprüfen, auf Rechtschreib- und Grammatikfehler zu achten und Links sowie Anhänge nicht ohne weiteres zu öffnen. Wenn Zweifel bestehen, sollte die Nachricht direkt beim vermeintlichen Absender verifiziert werden.

Da human-basierte Social Engineering Angriffe schwer oder gar nicht automatisiert zu erkennen sind, ist die **Schadensbegrenzung** von entscheidender Bedeutung. Hier kommen verschiedene technische und organisatorische Maßnahmen ins Spiel:

- **Domain-Tiering:** Diese Technik hilft, die Auswirkungen eines erfolgreichen Angriffs zu minimieren, indem unterschiedliche Sicherheitsstufen für verschiedene Domänen innerhalb eines Unternehmens festgelegt werden. Dadurch kann ein kompromittierter Bereich isoliert und der Schaden begrenzt werden.
- **Notfallmanagement:** Ein effektives Notfallmanagement umfasst klare Protokolle und Verantwortlichkeiten für den Fall eines Angriffs. Regelmäßige Schulungen und Übungen stellen sicher, dass alle Mitarbeiter wissen, wie sie im Ernstfall reagieren müssen. Dies beinhaltet auch die schnelle Identifikation und Isolation kompromittierter Systeme sowie die Benachrichtigung betroffener Personen und Behörden.

Zusätzlich zu diesen Maßnahmen können technische Hilfsmittel den Schutz vor Social Engineering Angriffen verbessern:

- **E-Mail-Sicherheitslösungen:** Tools wie E-Mail-Filter und Anti-Phishing-Software können verdächtige Nachrichten erkennen und blockieren, bevor sie die Mitarbeiter erreichen.
- **Zwei-Faktor-Authentifizierung (2FA):** Durch die Implementierung von 2FA wird ein zusätzlicher Schutzlayer hinzugefügt, der es Angreifern erschwert, Zugang zu sensiblen Systemen und Daten zu erlangen, selbst wenn sie bereits Anmelddaten gestohlen haben.
- **Netzwerküberwachung und IDSs:** Diese Systeme überwachen den Netzwerksverkehr auf verdächtige Aktivitäten und können Angriffe frühzeitig erkennen und abwehren.

Ein ganzheitlicher Ansatz, der sowohl präventive als auch reaktive Maßnahmen umfasst, ist unerlässlich, um die Widerstandsfähigkeit eines Unternehmens gegenüber Social Engineering Angriffen zu erhöhen. Durch die Kombination aus regelmäßiger Mitarbeiterschulung, technischer Absicherung und einem robusten Notfallmanagement kann das Risiko solcher Angriffe erheblich reduziert werden[21], [23]: Auch im Privaten, können und sollen dieser Techniken angewandt werden.

2.4 Bekannte Social Engineering Angriffe

Wie bereits beschrieben sind Social Engineering Angriffe weit verbreitet. Oft werden nur Anmeldedaten gestohlen oder PCs in einen Bot verwandelt. Es gibt allerdings auch regelmäßige Vorfälle die große und mittelständische Unternehmen betreffen.

Reenacted Video Call in Hong Kong

Ein besonders eindrucksvolles Beispiel für die Nutzung von Social Engineering in Verbindung mit Deepfake-Technologie ereignete sich Anfang des Jahres bei einem Unternehmen in Hongkong. Ein Finanzmitarbeiter wurde von Betrügern dazu gebracht, \$25 Millionen zu überweisen. Wie die Hongkonger Polizei berichtet, wurde der Mitarbeiter zu einem Videoanruf eingeladen, jede Person in diesem Meeting war jedoch eine Deepfake. Mittels Echtzeit Face- und Voice-Reenactment wurde das gesamte Meeting gefälscht.

Der Betrug begann mit einer Nachricht, die angeblich vom Finanzchef des Unternehmens in Großbritannien stammte und von einer geheimen Transaktion sprach. Obwohl der Mitarbeiter zunächst misstrauisch war und einen Phishing-Versuch vermutete, ließ er sich durch die anschließende Videokonferenz überzeugen, da die anwesenden Personen wie seine tatsächlichen Kollegen aussahen und klangen.

Dieser Fall zeigt eindrucksvoll, wie fortschrittlich und gefährlich Deepfake-Technologie inzwischen ist. Sie wurde hier nicht nur verwendet, um ein realistisches Video der Zielperson zu erstellen, sondern auch, um mehrere scheinbar authentische Teilnehmer an einem Videoanruf zu simulieren. Die Täuschung flog erst auf, als der Mitarbeiter nachträglich beim Hauptsitz des Unternehmens nachfragte[24].

Marriott-Hotel Databreach

Ein weiteres Beispiel für einen Social Engineering Angriff ereignete sich im Juni 2022 im Marriott-Hotel am Flughafen von Baltimore im US-Bundesstaat Maryland. Kriminelle hatten sich mittels Social Engineering durch einen Mitarbeiter des Hotels Zugang zum Netzwerk verschafft und 20 GB an Daten abgeschöpft, darunter auch Kreditkartendaten von Gästen und interne Geschäftsdaten des Hotels. Marriott hat die Strafverfolgungsbehörden eingeschaltet und wird nach eigenen Angaben etwa 400 Personen benachrichtigen. Eine Lösegeldforderung der Erpresser lehnte die Hotelkette ab[25].

Phishing-Kampanie United States Department of Labor

Ein weiteres Beispiel für einen Phishing-Angriff ist eine Kampagne aus dem Jahre 2022, die die United States Department of Labor (DoL) imitiert und Empfänger auffordert, Angebote einzureichen, um Office 365-Anmeldeinformationen zu stehlen. Die E-Mails

passierten gekaperte Server, die gemeinnützigen Organisationen gehören, um E-Mail-Sicherheitsblöcke zu umgehen. Außerdem wurde die Sender Adresse gespooft um den tatsächlichen Domains des DoL zu entsprechen.

Die Angreifer geben sich als leitender DoL-Mitarbeiter aus, der den Empfänger einlädt, sein Angebot für ein laufendes Regierungsprojekt einzureichen. Die E-Mails enthalten einen gültigen Briefkopf, professionell gestalteten Inhalt und einen dreiseitigen PDF-Anhang mit einem scheinbar legitimen Formular.

Das PDF enthält eine „BID“-Schaltfläche, die die Opfer auf eine Phishing-Seite weiterleitet. Die gefälschte Seite sieht überzeugend aus und enthält identisches HTML und CSS wie die echte. Die Bedrohungsakteure haben sogar eine Anweisungs-Pop-up-Nachricht hinzugefügt, um die Opfer durch (Phishing-)Prozess zu führen.

Wer für ein Projekt bietet, wird zu einem Formular zur Erfassung von Anmeldeinformationen weitergeleitet, das die E-Mail-Adresse und das Passwort von Microsoft Office 365 der Opfer abfragt[26].

3. Deepfake Varianten

3.1 Face Swapping und Reenactment

Es gibt verschiedene Techniken von Video Deepfakes, im Folgenden werden Face-Swapping, sowie Reenactment näher betrachtet. Beide Varianten können mit den oben vorgestellten Möglichkeiten realisiert werden. Es gibt Modelle die für einen von beiden Anwendungsfällen besser geeignet sind, grundlegend basieren aber heutige Modelle immer auf GANs.

Face Swapping

Face Swapping, eine weit verbreitete Technik innerhalb der Deepfake-Technologie, beinhaltet das Austauschen eines Gesichts in einem Bild oder Video durch das Gesicht einer anderen Person. Diese Methode hat insbesondere in den letzten Jahren erhebliche Fortschritte gemacht, vor allem durch die Entwicklung von GANs. Bei Face Swapping wird das Gesicht der Zielperson durch ein anderes Gesicht ersetzt, wobei Merkmale wie Hautfarbe, Beleuchtung und Gesichtsausdrücke so angepasst werden, dass das Ergebnis möglichst realistisch wirkt. Diese Technik findet vor allem Anwendung in der Gestaltungs- und Medienbranche. Es können z.B. Gesichter von Schauspielern auf ihre Stunt doubles gesetzt werden, um realistischere Stunt Szenen zu erzeugen. Eine besondere Form des Face Swapping ersetzt speziell den Mund eines Schauspielers, um die Synchronisation in anderen Sprachen zu vereinfachen[13]. Da nur das Gesicht ersetzt wird, müssen Dinge wie Hintergrund, Frisur und Kleidung selbst an die Zielperson angepasst werden. Dieser Aufwand ist heutzutage nicht mehr nötig, da Reenactmentmodelle ähnlich gute Ergebnisse erzielen. Bei hochrangigen Zielen lohnt sich dieser Mehraufwand, da die Ergebnisse von Face Swapping zum heutigen Stand der Technik noch unübertroffen sind.

Reenactment

Reenactment, auch als Face Transfer oder Puppeteering bekannt, ist eine Technik, bei der die Gesichtsausdrücke und -bewegungen eines Ausgangsbildes oder -videos auf ein Zielbild oder -video übertragen werden. Dies ermöglicht es, das Gesicht der Zielperson so zu manipulieren, dass es die gleichen Bewegungen und Ausdrücke wie das Ausgangsgesicht zeigt. Diese Technik findet sich ebenfalls in der Filmbranche wieder, indem z.B. verstorbene oder anderweitig verhinderte Schauspieler trotzdem noch in Filmen oder Serien zu sehen sind. Das vermutlich bekannteste Beispiel hierfür ist die Nutzung von Deepfake-Technologie, um die Charaktere Grand Moff Tarkin und Prinzessin Leia in "Rogue One: A Star Wars Story" realistischer darzustellen. In der originalen Filmproduktion wurden von beiden Charakteren alte 3D-Modelle bzw. Facescans verwendet um mit herkömmlichen Animations- und Rendertechniken realisiert. Durch den Einsatz von Face-Swapping wurde das Reenactment dieser Charaktere von Fans so verbessert, dass sie natürlicher und lebensechter wirken als die ursprünglichen Effekte. Dieses Beispiel zeigt die Vorteile vom Einsatz von Deepfakes in der Filmproduktion und erweitern die Möglichkeiten der

Branche erheblich[27].

Im Kontext von Cyber-Security sind die Anwendungsfälle offensichtlich. Es können durch Reenactment Videos von einflussreichen Personen innerhalb von Firmen erstellt werden, um Phishing noch effektiver zu gestalten. Außerdem können Videos von Personen des öffentlichen Lebens angefertigt werden in denen diese kontroverse Aussagen tätigen, um die öffentliche Meinung ins Negative zu ziehen. Ein gutes Beispiel hierfür ist “[You Won’t Believe What Obama Say In This Video](#)”.

Vor allem durch den Einsatz von auf Performance spezialisierter GANs können Videos nahezu in Echtzeit gefaked werden. Für Social Engineering relevante Videomedien sind ohnehin Video-Calls – dies hat zur Folge, dass ein Delay von wenigen Sekunden, sowie kleine Artefakte oder Bildrauschen nicht ins Gewicht fallen, da diese auch von der Streamingplattform ausgehen könnten. Die Qualität der Deepfakes braucht also nicht auf filmreifen Niveau zu sein, um für Social Engineering brauchbar zu sein. Dies hat zur Folge, dass schon mit wenig Aufwand und Wissen, eine Großzahl von Personen solche Fakes erstellen können.

4. Erstellung von Deepfake Audios

Um Audio Deepfakes erstellen zu können gibt es verschiedene Tools, für die verschiedene Arten des Audio Deepfakes. In dieser Arbeit wird auf 2 unterschiedlichen Audio Deepfake Tools eingegangen, um die Vielfältigkeit der Deepfakes besser demonstrieren zu können. Hierfür wird das Tool Tacotron2, für einen Text to Speech Deepfake und das Tool Real-Time Voice Cloning, um eine Echtzeit Sprachklonung durchzuführen, verwendet.

4.1 Tacotron2

Tacotron ist eine Architektur für Sprachsynthesen, die eine Sequenz-zu-Sequenz-Methode verwendet, um Magnituden-Spektrogramme direkt aus einer Eingabesequenz von Zeichen zu erzeugen.

4.1.1 Motivation

Die Motivation hinter Tacotron2 ist es bei der Erstellung von Text-to-Speech Deepfakes die Sprachqualität deutlich zu verbessern, sodass die synthetisch generierte Sprache so natürlich wie möglich klingt. Außerdem reduziert Tacotron2 die Komplexität des Prozesses, welcher normalerweise viel Fachkenntnisse und manuelle Anpassungen benötigt.[28],[29]

4.1.2 Fähigkeiten

Tacotron2 zeichnet sich durch mehrere Hauptmerkmale aus:

- **Sequenz-zu-Sequenz Modell:** Dieses Modell wird verwendet, um die Eingabesequenz (Text) in eine Ausgabesequenz (Sprachspektrogramm) zu konvertieren. Das Modell verwendet außerdem Aufmerksamkeitsparadigmen, um dem Modell zu helfen, sich auf relevante Teile des Textes zu konzentrieren, während es die Sprache generiert.[29]
- **Mel-Spektrogramm Generierung:** Diese Spektrogramme, bieten die Möglichkeit als Eingabe für ein Vocoder verwendet zu werden, welches die endgültige Audiosynthese durchführen kann, um noch bessere Audioqualität zu erreichen.[28]
- **Flexibilität:** Tacotron2 ist in der Lage, verschiedene sprachliche Eigenschaften und Stile zu erlernen und wiederzugeben, wodurch es eine breite Auswahl zur Generierung von Stimmen und Ausdrucksweisen hat.[28]
- **Integration mit Vocoder:** Tacotron2 übernimmt die Generierung der Spektrogramme, welche dann optimal in ein Vocoder, wie z.B. WaveNet, eingegeben werden kann, um so die finale Sprachsynthese durchführen zu können. Zudem führt die Kombination aus Tacotron2 und einem Vocoder zu einer deutlich verbesserten Audioqualität, weshalb die Integration mit einem guten Vocoder von hoher Bedeutung ist. [28]

4.1.3 Workflow

Der Workflow von Tacotron2 ist in einer Pipeline und besteht aus drei Hauptphasen: Extraction, Training und Conversion.

Preprocessing

Für die Erstellung eines Deepfakes wird eine Sammlung von Daten benötigt, um das Modell trainieren zu können. Die Sammlung beinhaltet das Zusammenstellen eines Datensatzes aus Text- und Sprachaufnahmen. Hierbei wird darauf geachtet dass die Daten für das Training des Modells geeignet sind.[28]

Extraction

In der Extraktionsphase werden dann die relevanten Abschnitte aus den Sprachaufnahmen extrahiert. Tacotron2 wandelt hierbei die Sprachaufnahmen in Mel-Spektrogramme um, die das Modell anschließend dann während des Trainings verwendet.[28]

Training

Durch das Training passiert dann der eigentliche Prozess, in der ein Modell trainiert wird, um realistische Text-To-Speech Ausgaben zu erzeugen. Dabei wird das Modell darauf trainiert, aus den Eingabetexten Mel-Spektrogramme zu erzeugen. Parallel oder auch anschließend dazu kann ein Vocoder, wie WaveNet, trainiert werden, um aus den Mel-Spektrogrammen die endgültige Audiodaten zu erzeugen.[28]

Conversion

In der letzten Phase, die Konvertierungsphase, werden dann die Spektrogramme in eine Wellenform umgewandelt. Der Prozess von der Erzeugung von Sprache aus einem Spektrogramm wird Vocoder genannt. Dadurch wird dann die tatsächliche Sprache generiert.[28],[30]

4.2 Praxisbeispiel Tacotron2

Im Folgenden wird der Workflow zum Erstellen von Audio (Text-To-Speech) Deepfakes mit Tacotron näher betrachtet. Ziel dieses Kapitels ist es eine menschliche Audiodatei von einer Texteingabe zu erzeugen.

4.2.1 Laborumgebung

Die Tacotron2 Audio Deepfakes werden auf folgender Hardware erstellt.

CPU:	AMD Ryzen 7 2700X
RAM:	16GB DDR4 3000MHz
GPU:	NVIDIA GTX 1070 (8GB GDDR5)
OS:	Windows 10
Mikrofon:	Auna Mic CM900
Aufnahmeprogramm:	Audacity

4.2.2 Programmstruktur

Für Vorbereitung und Preprocessing des Trainingsmaterials wurden folgende 3 Skripts verwendet:

- **transcribe_wav2vec.py:** Das Skript [transcribe_wav2vec.py](#) wird verwendet, um grobe Transkripte von den Aufnahmen zu erstellen. Dieses Transkript kann auch händisch geschrieben werden, jedoch erspart dieses Skript eine Menge Zeit.
- **tacotron2_preprocessor_wav_files.py:** Das Skript [tacotron2_preprocessor_wav_files.py](#) wird verwendet, um die Audiodateien in das von Tacotron2 benötigte Format zu konvertieren.
- **audio_metadata_updater.py:** Das Skript [audio_metadata_updater.py](#) wird verwendet, um den Titel jeder WAV Datei zu aktualisieren, sodass er der entsprechenden Nummer entspricht. Auch dieses Skript wird nicht dringen benötigt, da der Vorgang auch händisch getan werden kann. Jedoch spart es auch hier eine Menge an Zeit.

Für das anschließende Training und Synthesieren der Modelle werden folgende 2 Ipynb Dateien verwendet:

- **FakeYou_Tacotron_2_Training.ipynb:** Hier wird das Training der Modelle konfiguriert und ausgeführt.
- **FakeYou_Tacotron2_Hi_Fi_GAN_(CPU).ipynb:** Hier werden die trainierten Modelle synthetisiert und ausgegeben.

4.2.3 Vorbereitung

Bevor der Deepfake erstellt werden kann, müssen zuvor einige Schritte durchgeführt werden. Zuerst werden Audiospuren einer Person benötigt, um genug Ausgangsmaterial für das Training zur Verfügung zu haben. Hierzu müssen mehrere verschiedene Sätze, Satz für Satz, aufgenommen und in separate .wav Audiodateien gespeichert werden. Hierbei gilt natürlich, je mehr Audiodateien zur Verfügung stehen, desto höher wird die finale Qualität des trainierten Modell. Die Audiodateien sollen dann in eine gemeinsame Ordner mit einer Zahl als Dateinamen, aufsteigend, abgelegt werden (z.B. 1.wav, 2.wav, 3.wav, ...).

Danach wird ein [Skript](#) benötigt, um grobe Transkripte der Aufnahmen zu erstellen. Hierbei wird eine [Liste.txt](#) Datei erstellt, in der die aufgenommenen Sätze stehen. Diese Liste muss auf Richtigkeit geprüft werden. Es sollte außerdem darauf geachtet werden, dass jede Zeile mit einem Punkt endet und weder Großbuchstaben noch Kommas im Satz vorkommen.

4.2.4 Preprocessing

Um das Training durchführen zu können, müssen zunächst die zuvor erstellten .wav Audiodateien durch ein preprocessing Skript durchlaufen. Das [Skript](#) konvertiert die Audiodateien in das von Tacotron2 benötigte Format. Es ändert das Audioformat, einschließlich der Abtastrate und Kanäle.

Nachdem das preprocessing erfolgreich durchlaufen ist, muss in jeder .wav Audiodatei die entsprechende Zahl als Titel eingefügt werden. Hierzu gibt es ein weiteres [Skript](#), dass die Zahlen aufsteigend in den Titel der jeweiligen Audiodateien schreibt.

Danach muss der Ordner, in den die .wav Audiodateien liegen, in einen komprimierten Zip-Ordner erstellt und dann in Google Drive hochladen werden.

4.2.5 Extraktion

Training

Um das Modell zu trainieren wird [Google Colab](#) verwendet. Dort müssen zunächst mehrere Schritte getan werden, bevor es zum Training des Modells geht. Zuerst wird die GPU überprüft, danach muss Google Drive verbunden werden und anschließend Tacotron2 installiert werden.

Check the GPU

It is not recommended to use the **K80** card (although it works correctly, it is **time consuming**). Restart the runtime to **get another card**.

[Code anzeigen](#)

```
GPU 0: Tesla T4 (UUID: GPU-8a14a279-97c7-511c-e612-b1a2f4b8729c)
```

Training

1. Mount your Google Drive.

[Code anzeigen](#)

```
Mounted at drive
```

2. Install Tacotron2 (w/ARPAbet).

[Code anzeigen](#)

```
Requirement already satisfied: contourpy>=1.0.1 in /usr/local/lib/python3.10/dist-packages (from matplotlib) (1.2.1)
Requirement already satisfied: cycler>=0.10 in /usr/local/lib/python3.10/dist-packages (from matplotlib) (0.12.1)
Requirement already satisfied: fonttools>=4.22.0 in /usr/local/lib/python3.10/dist-packages (from matplotlib) (4.53.1)
Requirement already satisfied: kiwisolver>=1.0.1 in /usr/local/lib/python3.10/dist-packages (from matplotlib) (1.4.5)
Requirement already satisfied: packaging>=20.0 in /usr/local/lib/python3.10/dist-packages (from matplotlib) (24.1)
Requirement already satisfied: pyparsing>=2.3.1 in /usr/local/lib/python3.10/dist-packages (from matplotlib) (3.1.2)
Requirement already satisfied: python-dateutil>=2.7 in /usr/local/lib/python3.10/dist-packages (from matplotlib) (2.8.2)
Requirement already satisfied: more-itertools>8.5.0 in /usr/local/lib/python3.10/dist-packages (from inflect) (10.3.0)
Requirement already satisfied: typeguard>=4.0.1 in /usr/local/lib/python3.10/dist-packages (from inflect) (4.3.0)
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.10/dist-packages (from python-dateutil>=2.7->matplotlib) (1.16.0)
Requirement already satisfied: typing-extensions>=4.10.0 in /usr/local/lib/python3.10/dist-packages (from typeguard>=4.0.1->inflect) (4.12.2)
```

Abbildung 4.1: GPU Prüfung, Google Drive Verbindung und Tacotron2 Installation

Nach der Installation von Tacotron2 muss die wavs.zip Datei ausgewählt werden, sowie das Transkript (list.txt), welches am Anfang erstellt wurde.

3. Load dataset.

The audios can be compressed in a **ZIP** file (recommended) or loose.
You can also manually upload the ZIP/folder and enter the path in the field below, or
itself, leave the field empty.

Enable audio processing?

Remember that they must be in a compatible format, i.e., sample rate **22050, 16 bit, mono**. If you

`audio_processing:`

`drive_path: "text hier einfügen"`

[Code anzeigen](#)

```
/content/TTS-TT2/wavs
```

Upload your dataset(audios)...

wavs.zip

• **wavs.zip**(application/x-zip-compressed) - 3590380 bytes, last modified: 3.8.2024 - 100% done

Saving wavs.zip to wavs.zip

rm: cannot remove '/content/TTS-TT2/wavs/list.txt': No such file or directory

Metadata removal and audio verification...

mkdir: cannot create directory '/content/tempwav': File exists

50 processed audios. total duration: 0:01:28

All set, please proceed.

4. Upload the transcript.

The transcript must be a `.txt` file formatted in **UTF-8 without BOM**.

Abbildung 4.2: Datensatz laden für Modell Training

Nach dem Hochladen des Datensatzes kann das Modell konfiguriert werden, mit welchen Parametern das Training ausgeführt werden soll.

5. Configure the model parameters.

Your desired model name:

`model_filename: "max-50"`

Upload your transcription / text to TTS-TT2/filelists and right click -> copy path:

`Training_file: "filelists/list.txt"`

Lower learning rates will take more time but will lead to more accurate results:

`hparams.A_: 3e-4`

Your batch size, lower if you don't have enough ram:

`hparams.batch_size: 5`

`use_cmudict: true`

Your total epochs to train to. Not recommended to change:

`hparams.epochs: 250`

Where to save your model when training:

`output_directory: "/content/drive/MyDrive/colab/outdir"`

Abbildung 4.3: Konfiguration der Modell Parameter

Danach werden die .wav Dateien in die sogenannten Mel Spektrogramme konvertiert.

6. Convert the .WAV files to Mel spectrograms and check the files.

[Code anzeigen](#)

```
Generating Mels  
100% [██████████] 50/50 [00:00<00:00, 84.38it/s]  
Checking for missing files  
Checking Training Files  
Checking Validation Files  
Finished Checking
```

7. Check the working cmudict patch

[Code anzeigen](#)

```
/content/TTS-TT2  
{W IY1} {M AH1 S T} {K AE1 P CH ER0} {AE1 N} {ER1 TH} {K R IY1 CH ER0} , {K EY1} nine , {AH0 N D} {R IH0 T ER1 N} {IH1 T} {B AE1 K} {W IH1 DH} {AH1 S} {T UW1} {M AA1 R Z} .
```

Abbildung 4.4: Mel Spektrogramme Konvertierung

Sobald die Vorbereitungen abgeschlossen sind, kann das Modell trainiert werden. Das

trainierte Modell befindet sich nach dem Training in dem angegebenen Google Drive Ordner.

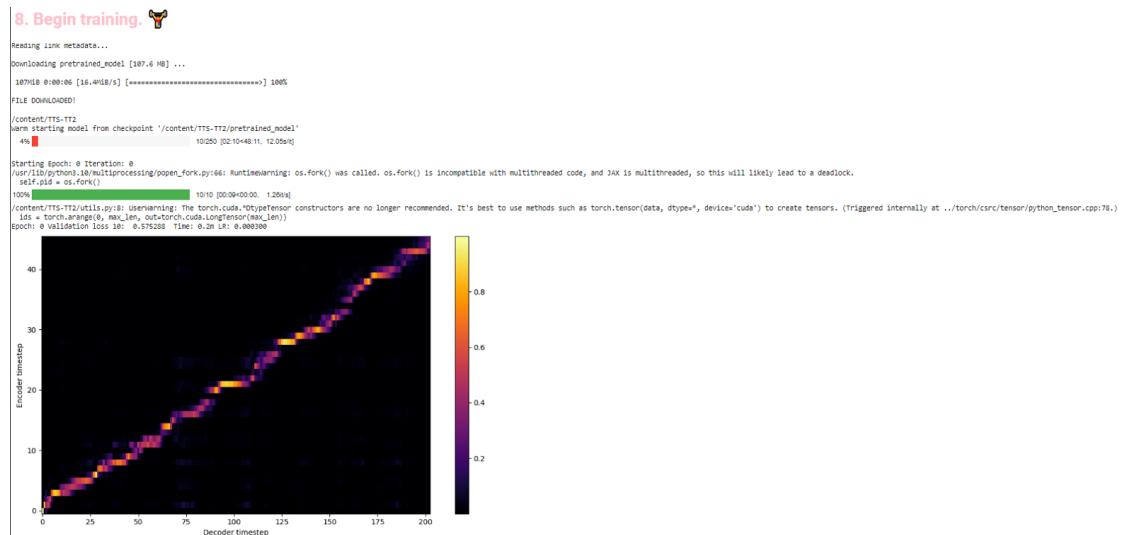


Abbildung 4.5: Modell Training

Synthetisierung von Sprache

Nach dem Training des Modells, muss die Sprache noch synthetisiert werden. Hierfür muss zunächst das Modell in Google Drive auf öffentlich gestellt und danach die Tacotron ID aus dem Google Drive Link kopiert werden. Die Tacotron ID ist immer unterschiedlich von Modell zu Modell.

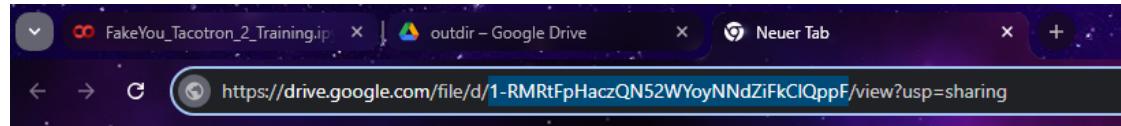


Abbildung 4.6: Tacotron ID aus dem Link des Modells

Danach kann die Tacotron ID in den **Vocoder** eingegeben werden, welcher dann das Modell verwendet, um einen Text zu einer Audio zu synthetisieren. Nach dem Laden, kann anschließend die Audio angehört werden.

Config:

Restart the runtime to apply any changes.

```
tacotron_id: "1-RMRTFpHaczQN52WYoyNNdZiFkCIQppF"
hifigan_id: "universal"
```

leave blank or enter "universal" for universal model

pronunciation_dictionary:

show_graphs:

max_duration: 20

stop_threshold: 0.5

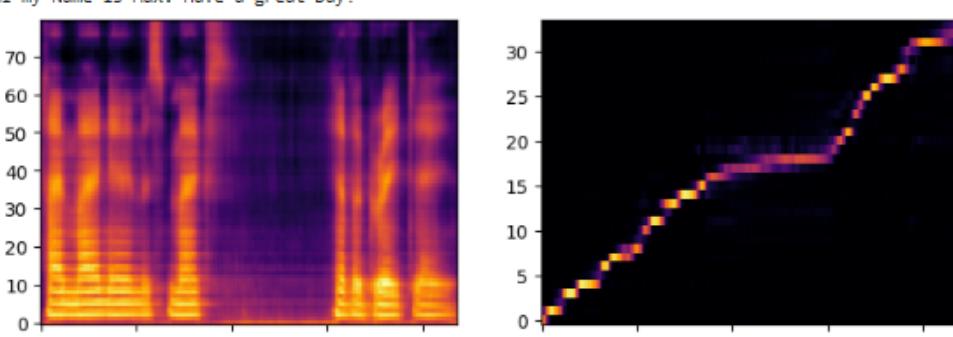
superres_strength: 5

[Code anzeigen](#)

```
*** Current Config:
pronunciation_dictionary: False
show_graphs: True
max_duration (in seconds): 20
stop_threshold: 0.5
superres_strength: 5
```

Enter/Paste your text.

Hi my Name is Max. Have a great Day!



0:02 / 0:02

Abbildung 4.7: Synthesisierung des Modells

4.3 Praxisbeispiel Real-Time Voice Cloning

Real-Time Voice Cloning ist ein Bereich der Sprachsynthese und des maschinellen Lernens, welcher darauf abzielt, die Stimme einer Person in Echtzeit zu klonen, sodass der synthetisierte Klang kaum vom Original zu unterscheiden ist.

Die Motivation, Fähigkeiten und Workflow sind vergleichbar mit denen von Tacotron2. Im Folgenden wird ein der Workflow zum Erstellen von Real-Time Voice Cloning Audios demonstriert. Ziel dieses Kapitels ist es, eine Stimme in Echtzeit zu klonen und das mit nur einer 5 sekündigen Audioeingabe.

4.3.1 Laborumgebung

Die Real-Time Voice Cloning Audio Deepfakes werden auf folgender Hardware erstellt.

CPU:	AMD Ryzen 7 5700
RAM:	16GB
GPU:	AMD Radeon(TM) Graphics
OS:	Windows 11
Mikrofon:	Auna Mic CM900
Aufnahmeprogramm:	Audacity

4.3.2 Programmstruktur

In dem [Github Repository](#) befinden sich verschiedene Ordner und Skripte. Unsere 2 Hauptskripte die demo_cli.py und demo_toolbox.py führen die anderen Skripte in dem Repository mit aus, um so zum Beispiel den Encoder zu Trainieren oder Testen. Die drei Dateien, mit denen wir uns hauptsächlich befassen, sind folgende:

- **demo_cli.py:** Das demo_cli.py Skript prüft ob der Encoder, Vocoder und Synthesizer in der richtigen Stelle abgelegt ist und ob diese funktionieren sowie auch miteinander kommunizieren.
- **demo_toolbox.py:** Das demo_toolbox.py Skript öffnet die eigentliche Toolbox, mit der wir die Stimme klonen können.
- **requirements.txt:** In der requirements.txt Datei stehen die benötigten Packages mit dementsprechender Version, welche installiert werden müssen.

4.3.3 Vorbereitung

Das Voice Cloning erfolgt über eine Real-Time Voice Cloning Toolbox. Hierfür muss zunächst das [Github Repository](#) auf der Laborumgebung geklont werden. Anschließend wird Python 3.9 benötigt, um die benötigte Packages installieren zu können. Danach werden die Komponenten PyTorch und ffmpeg benötigt, um Audio Dateien einlesen zu können. Nach der Installation der zwei Komponenten müssen die Packages, welche in dem Github Repository in einer Text Datei stehen, installiert werden. Wenn alles Erfolgreich installiert wurde, kann zunächst das Python Skript demo_cli.py in dem Repository ausgeführt werden. Das Skript testet ob der Encoder, Vocoder und Synthesizer vorhanden sind und richtig funktionieren.

```
Loaded encoder "encoder.pt" trained to step 1564501
Synthesizer using device: cpu
Building Wave-RNN
Trainable Parameters: 4.481M
Loading model weights at saved_models\default\vocoder.pt
C:\Users\ernst\OneDrive\Desktop\RealTime\Real-Time-Voice-Cloning\vocoder\inference.py:36: FutureWarning: pickle module implicitly. It is possible to construct malicious pickle data which will execute code. In a future release, the default value for `weights_only` will be flipped to `True` to be loaded via this mode unless they are explicitly allowlisted by the user via `torch.serialization.load_lua`. Please open an issue on GitHub for any issues related to this.
Testing your configuration with small inputs.
    Testing the encoder...
        Testing the synthesizer... (loading the model will output a lot of text)
Trainable Parameters: 30.870M
C:\Users\ernst\OneDrive\Desktop\RealTime\Real-Time-Voice-Cloning\synthesizer\models\tacotron2: FutureWarning: pickle module implicitly. It is possible to construct malicious pickle data which will execute code. In a future release, the default value for `weights_only` will be flipped to `True` to be loaded via this mode unless they are explicitly allowlisted by the user via `torch.serialization.load_lua`. Please open an issue on GitHub for any issues related to this.
    checkpoint = torch.load(str(path), map_location=device)
Loaded synthesizer "synthesizer.pt" trained to step 295000

| Generating 1/1

Done.

    Testing the vocoder...
All test passed! You can now synthesize speech.

This is a GUI-less example of interface to SV2TTS. The purpose of this script is to show how
Interactive generation loop
Reference voice: enter an audio filepath of a voice to be cloned (mp3, wav, m4a, flac, ...):
```

Abbildung 4.8: Encoder, Vocoder und Synthesizer Test

Waren die Tests erfolgreich, kann das Skript demo_toolbox.py ausgeführt werden, um

die Toolbox zu öffnen.

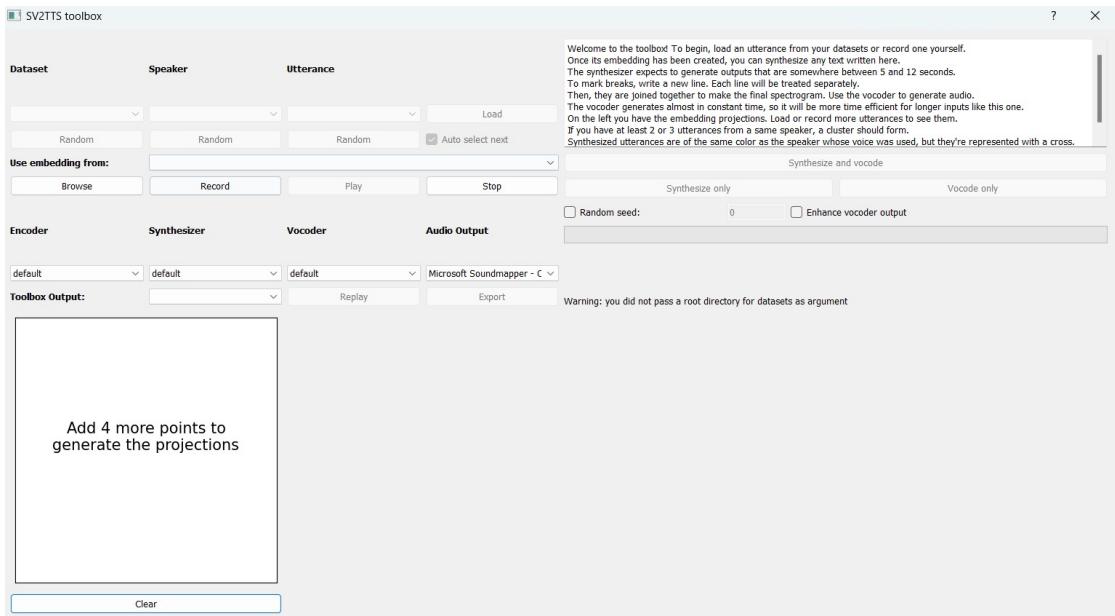


Abbildung 4.9: Real-Time Voice Cloning Toolbox

4.3.4 Extraktion

Angekommen in der Toolbox, kann entweder eine bereits vorhandene Audiodatei eingegeben werden oder eine Echtzeit Aufnahme verwendet werden, um so das Trainingsmaterial der Toolbox bereitzustellen. Nachdem auswählen des Trainingsmaterial, kann die Stimme synthetisiert und vocoded werden. Zuerst wird der Text eingegeben, der von der geklonten Stimme wieder geben werden soll. Die Toolbox erstellt dann ein Mel Spektrogramm von dem Trainingmaterial und lässt dieses durch den Synthesizer durchlaufen, um an eine Audioausgabe zu erstellen. Nach einer kurzen Wartezeit wurde die Stimme geklont.

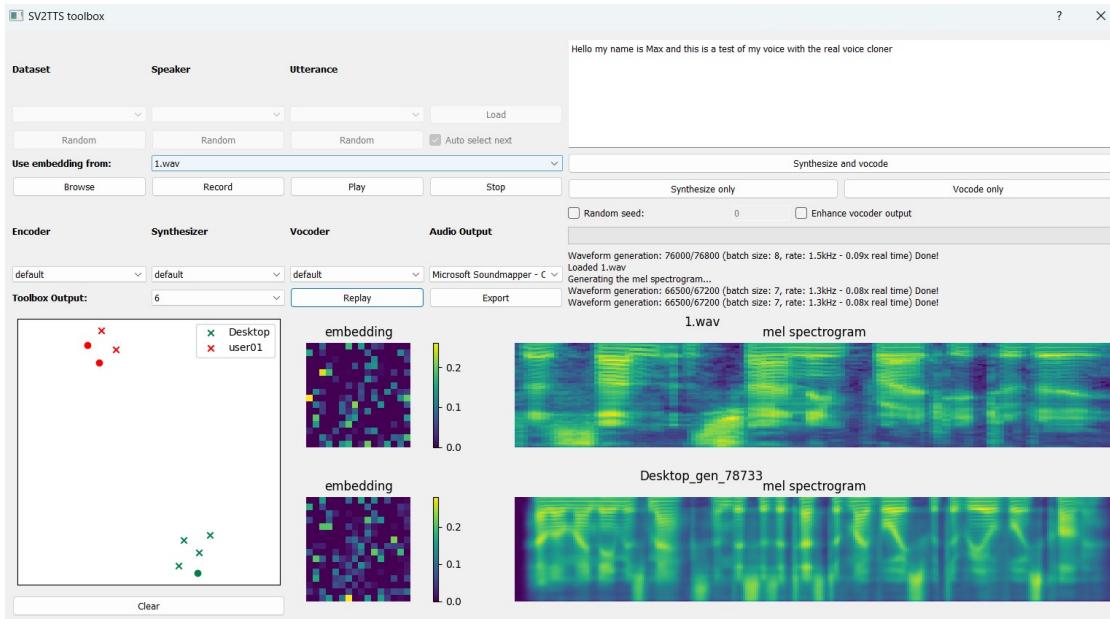


Abbildung 4.10: Real-Time Voice Cloning Toolbox nach Erzeugung der geklonten Stimme

5. Erstellung von Deepfake Videos

Um Deepfakes erstellen zu können, gibt es einige verschiedene Tools. Die verbreitetsten Tools sind DeepFaceLab (DFL) für Video-Deepfakes und DeepFaceLive (DFLive) für JIT-Anwendungen. Im Folgenden werden beide Tools mit ihren verschiedenen Möglichkeiten sowie Best-Practices vorgestellt.

5.1 DeepFaceLab

DFL ist ein Open-Source Framework zur Erstellung von Face-Swapping Videos. DFL pipelined den Prozess der fotorealistischen Videomanipulation.

5.1.1 Motivation

Die Motivation hinter DFL ist es, sowohl die Erstellung als auch die Erkennung von Deepfakes zu verbessern. Durch die Bereitstellung eines leistungsfähigen und flexiblen Werkzeugs für Gesichtsmanipulationen trägt es zur Weiterentwicklung der Forschung im Bereich der Medienfälschungserkennung bei. Es hilft dabei, qualitativ hochwertige Fälschungsdaten zu erzeugen, die für die Entwicklung robuster Erkennungsmodelle unerlässlich sind[17], [31].

5.1.2 Fähigkeiten

DeepFaceLab zeichnet sich durch mehrere Hauptmerkmale aus:

- **Usability:** Der gesamte Workflow von DeepFaceLab, einschließlich Datenverarbeitung, Modelltraining und Nachbearbeitung, ist darauf ausgelegt, so benutzerfreundlich und effizient wie möglich zu sein. Es bietet ein vollständiges CLI (Command Line Interface), das eine flexible Implementierung ermöglicht.
- **Breite technische Unterstützung:** Das Tool unterstützt Multi-GPU-Konfigurationen und die Verwendung von mehreren Threads zur Beschleunigung grafischer Operationen und Datenverarbeitung. Laut Paper können selbst auf einem Rechner mit nur 2GB VRAM erfolgreiche Gesichtsmanipulationen durchgeführt werden[31].
- **Erweiterbarkeit:** Die Architektur von DeepFaceLab ist modular aufgebaut, sodass einzelne Komponenten einfach ausgetauscht werden können[31].
- **Skalierbarkeit:** DFL unterstützt durch verschiedene Tools die Verarbeitung von großen Datenmengen.[31].

5.1.3 Workflow

Der Workflow von DeepFaceLab ist in der Form einer Pipeline und besteht aus drei Hauptphasen: Extraction, Training und Conversion.

Pretraining

Da das Trainieren von Deepfake Modellen viel Zeit und Rechenleistung in Anspruch nimmt, ist es ratsam ein vortrainiertes Modell (engl. pretrained Model) als Ausgangspunkt zu verwenden. **Pretrained Models** sind Modelle die mit vielen unterschiedlichen Gesichtern trainiert wurden. Sie liefern zwar nach gleich vielen Iterationen schlechtere Ergebnisse als Modelle, die speziell auf ein Gesichterpaar trainiert wurden. Allerdings können diese generischen Modelle schnell auf ein Gesichterpaar spezialisiert werden. Es ist also ratsam ein generisches Modell vorzutrainieren und abzuspeichern. Künftige spezifische Modelle können dieses als Ausgangspunkt verwenden und so schneller trainiert werden. Oft können auch **Pretrained Models** bereits im Internet gefunden werden.

Extraction

In der Extraktionsphase werden Gesichter aus den Quell- und Zielvideos extrahiert. Diese Phase umfasst mehrere Algorithmen und Verarbeitungsschritte, dazu gehören face detection, face alignment und face segmentation. DFL bietet verschiedene Extraktionsmodi (z.B. half-face, full-face, whole-face), um den unterschiedlichen Anforderungen gerecht zu werden[17], [31].

- **Face Detection:** Hierbei wird ein CNN verwendet, um Gesichter in den Video-frames zu erkennen. Diese Detektion ist entscheidend, um die Position und Größe des Gesichts für die weiteren Verarbeitungsschritte zu bestimmen.
- **Face Alignment:** Nachdem die Gesichter erkannt wurden, werden sie durch Algorithmen zur Gesichtsangleichung normalisiert. Dies bedeutet, dass die Gesichter in eine einheitliche Position und Größe gebracht werden, was die Genauigkeit der späteren Schritte erhöht.
- **Face Segmentation:** In diesem Schritt werden die Gesichter von den restlichen Bildinformationen getrennt. Dies ermöglicht eine gezielte Bearbeitung der Gesichter ohne Beeinträchtigung des restlichen Bildes.

Training

Das Training ist die entscheidende Phase, in der ein Modell trainiert wird, um realistische Gesichtsmanipulationen zu erzeugen. DeepFaceLab verwendet zwei Hauptstrukturen: die DF-Struktur und die LIAE-Struktur. Dabei wird eine Mischung aus DSSIM- und MSE-Verlusten verwendet, um sowohl eine schnelle Generalisierung als auch eine hohe Präzision zu erreichen[17], [31].

- **DF-Struktur (DeepFakes):** Diese Struktur basiert auf GANs und nutzt zwei Netzwerke - ein Generator und ein Diskriminatator. Der Generator versucht, realistische Gesichter zu erzeugen, während der Diskriminatator versucht, zwischen echten und generierten Gesichtern zu unterscheiden. Durch diesen Wettbewerb lernen beide Netzwerke, immer realistischere Ergebnisse zu produzieren. Diese Struktur erzielt die realistische Darstellung des Quellgesichts.
- **LIAE-Struktur (Lenient Interpolation AutoEncoder):** Diese Struktur verwendet ebenfalls GANs. Es wird ein Encoder-Decoder-Ansatz verwendet, bei dem das Gesicht in einen latenten Raum kodiert und anschließend in das Zielgesicht dekodiert wird. Diese Struktur erlaubt das morphen des Quellgesichts. Dies führt zu einer besseren Integration in das Zielbild mit der Möglichkeit einer Realitätsabweichung.
- **Verlustfunktionen:** Der DSSIM (Structural Dissimilarity) Verlust wird verwendet, um strukturelle Unterschiede zwischen dem generierten und dem echten Bild zu minimieren, während der MSE (Mean Squared Error) Verlust die pixelweisen Unterschiede minimiert.

Conversion

In der Konvertierungsphase werden die erzeugten Gesichter wieder in die ursprünglichen Zielbilder eingefügt. Dieser Schritt umfasst Farbanpassungen, um den Hautton und die Beleuchtung anzugeleichen, sowie das Schärfen der Bilder, um Details hervorzuheben. DeepFaceLab bietet mehrere Farbanpassungsalgorithmen und nutzt ein vortrainiertes Super-Resolution-Netzwerk, um die endgültigen Bilder zu verbessern[17], [31].

- **Farbanpassung:** Hierbei werden Algorithmen verwendet, die den Hautton und die Beleuchtung des generierten Gesichts an das Zielbild anpassen, um einen nahtlosen Übergang zu gewährleisten.
- **Super-Resolution:** Ein vortrainiertes Super-Resolution-Netzwerk wird eingesetzt, um die Auflösung der generierten Gesichter zu erhöhen und feinere Details hervorzuheben.

5.2 Praxisbeispiel DeepFaceLab

Im Folgenden wird der Workflow zum Erstellen von Video Deepfakes mit DFL näher betrachtet. Ziel dieses Kapitels ist das Tauschen zweier Gesichter in zwei Videos. Für das Ersetzen in Echtzeit wird DFLive (DeepFaceLive) verwendet, dessen Workflow in einem anderen Kapitel (5.3) beschrieben wird.

5.2.1 Laborumgebung

Alle Video-Deepfakes (DFL und DFLive), wurden auf folgender Hardware erstellt.

CPU: AMD Ryzen 5 2600X
RAM: 16GB DDR4 3000MHz
GPU: NVIDIA RTX 2070 (8GB GDDR6 VRAM)
OS: Windows 11

Ziel des Deepfakes ist das Gesicht von RDJ (Robert Downey Jr.) auf Prof. Volker Knoblauch in diesem [Video](#) zu swappen, sodass die Hochschule Aalen amerikanische Prominente auf ihrem Youtube-Kanal zeigen kann.

5.2.2 Programmstruktur

Nach der Installation von DFL finden sich im entsprechenden Ordner eine Vielzahl von .bat-Dateien, sowie ein _internal- und workspace-Ordner. Die .bat-Dateien führen die im _internal-Ordner abgelegten Scripte mit den entsprechenden Parametern aus. Es wäre also möglich auf diese Dateien zu verzichten und DFL lediglich über eine Konsole auszuführen. Alle Dateien, die während der Erstellung eines Deepfakes erstellt oder benötigt werden, werden im workspace-Ordner abgelegt. Bei der Installation von DFL werden Beispieldaten mitgeladen.

Immer wieder finden sich die Bezeichnungen `src` und `dst`. Diese referenzieren in DFL:

- **src (Source):** Das Gesicht, bzw. das zugehörige Videomaterial, das über das Zielgesicht gelegt werden soll.
- **dst (Destination):** Das Gesicht, bzw. das zugehörige Videomaterial, das ersetzt werden soll.

5.2.3 Vorbereitung

Bevor ein Deepfake erstellt werden kann, muss zuerst einmal geeignetes Ausgangsmaterial gesammelt werden. Generell gilt, je besser das Trainingsmaterial, desto besser werden die Deepfakes. Die Qualität des Ausgangsmaterials hängt von der Auflösung, der Belichtung, der Vielseitigkeit der Ausdrücke und der verschiedenen Aufnahmewinkeln (Fig. 5.1) ab. Dabei gilt, je ähnlicher Source- und Destination-Material sind, desto überzeugender wird das Ergebnis.

Ein Trainingsdatensatz sollte mehrere Hundert Bilder umfassen. Je nach Resolution des trainierten Models sollten die Zahlen sogar in die Tausende gehen.

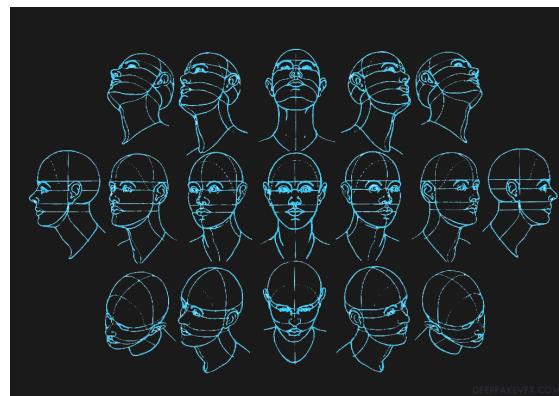


Abbildung 5.1: Head Angles Diagram

Für den exemplarischen Deepfake dieser Arbeit, werden das mitgelieferte Standardvideo von RDJ als **Source** und das Interview von Prof. Dr. Harald Rieger und Prof. Volker Knoblauch als **Destination** verwendet.

5.2.4 Pretraining

Aus Gründen die in 5.1.3 beschrieben sind, wird ein Modell vortrainiert. DFL liefert bereits einen Datensatz zum Vortrainieren mit. Das Pretraining kann also ohne weitere Vorbereitung beginnen.

Pretrain XSeg

```
5.XSeg) train.bat
```

Öffnet die Konsole für die Konfiguration.

- **Face type:** Bestimmt wie viel von einem Gesicht ersetzt werden soll. Z.B. das Gesicht mit oder ohne Stirn usw.
- **Batch_size:** Je höher die Batch size, desto größer die Hardwareanforderungen und desto besser die Ergebnisse
- **Enable pretraining mode:** Selbsterklärend

Nach einem Test ob die Hardware ausreichend für die Konfiguration ist, startet das Training (Abbildung 5.2). Das Pretraining für XSeg kommt schon mit einigen Zehntausend Iterationen aus.

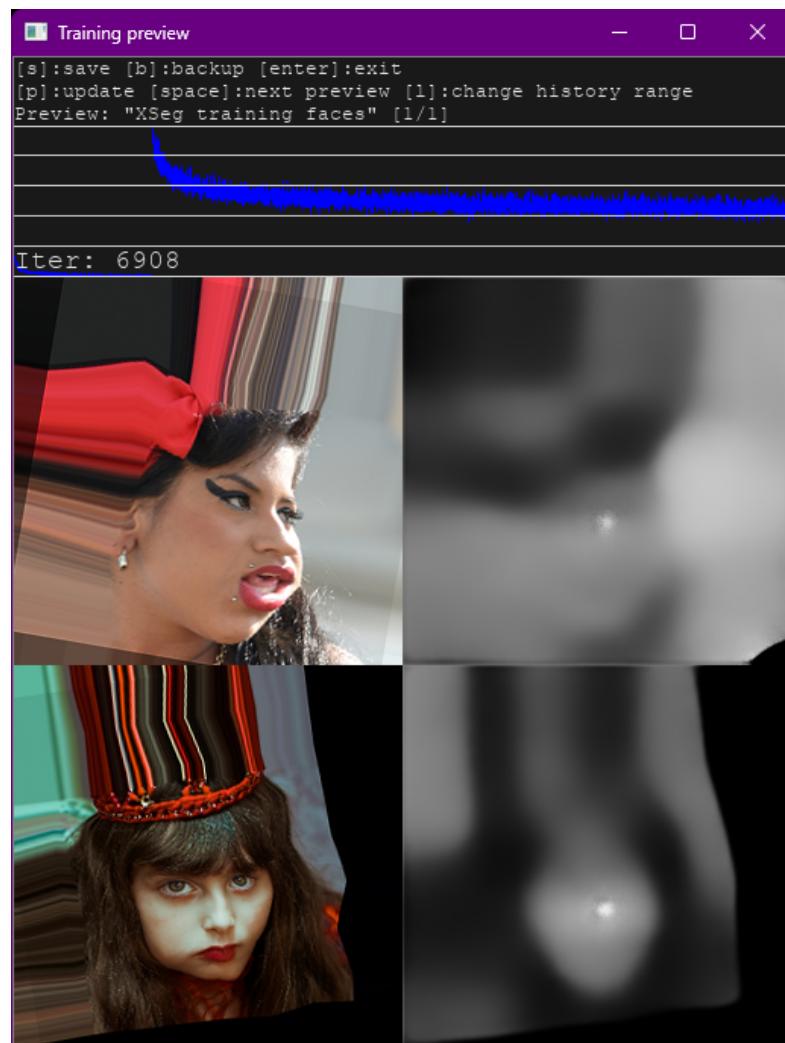


Abbildung 5.2: XSeg Pretraining im *head* Modus

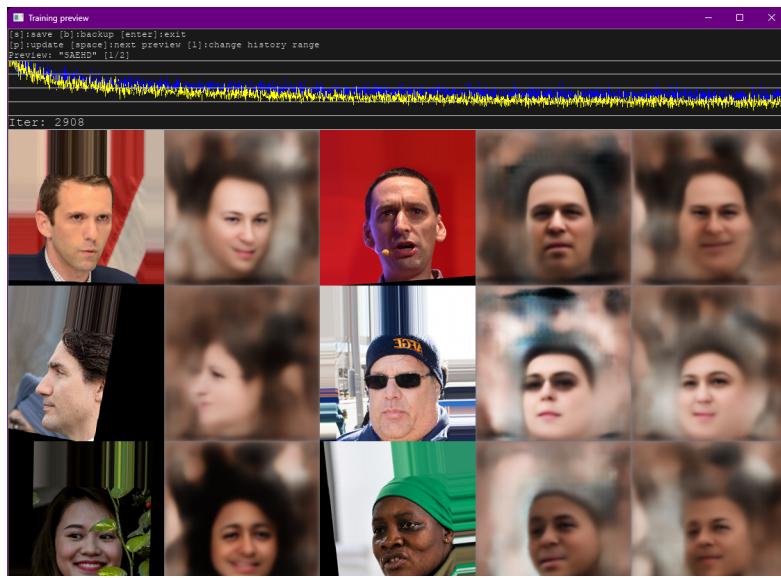


Abbildung 5.3: SAEHD Pretraining mit 224 Resolution im *head* Modus

Pretrain SAEHD

6) train SAEHD.bat

Dies öffnet die Konsole um ein neues Modell zu konfigurieren. Für das Pretraining können alle Einstellungen, außer wenige Ausnahmen, auf Standard gelassen werden. Geändert werden sollten die Folgenden:

- **Autobackup every N hour:** Selbsterklärend
- **Target iteration:** Sinnvolle Werte für Pretraining sind 500 Tausend bis 1 Million
- **Resolution:** Je höher der Wert, desto hochauflöster das Ergebnis. Für **face** und **whole-face** sind 128 ausreichend. Für **head** mindestens 224. Für das bestmögliche Ergebnis so hoch setzen wie die Hardware mithalten kann.
- **Face type:** Bestimmt wie viel von einem Gesicht ersetzt werden soll. Z.B. das Gesicht mit oder ohne Stirn usw.
- **Batch_size:** Analog zu Resolution. Nachdem sich für eine Resolution entschieden wurde, so hoch setzen bis die Hardware ausgelastet ist.
- **Enable pretraining mode:** Selbsterklärend

Nun wird ebenfalls die Hardware getestet und das Training gestartet (Abbildung 5.3).

5.2.5 Extraktion

Im ersten tatsächlichen Schritt werden die Videos zu einem Trainingsdatensatz verarbeitet.

- 2) extract images from video data_src.bat
- 3) extract images from video data_dst FULL FPS.bat

Diese Skripte zerlegen mithilfe von FFmpeg das **src**- bzw. **dst**-Video in ihre einzelnen Frames. Diese sind im **workspace** unter **data_src** bzw. **data_dst** zu finden. Bei dem **src**-Video kann in der Konsole zusätzlich angegeben werden, wie viele FPS (Frames Per Second) extrahiert werden sollen. Bei langen **src**-Videos kann dies sinnvoll sein. Werden 5 FPS aus einem 4-minütigen Video extrahiert, ist die Variation der Bilder größer als bei 10 FPS aus einem 2-minütigen Video. Natürlich können immer auch alle Frames extrahiert werden, hier muss der größere Speicheraufwand und die längere Trainingsdauer abgewogen werden. Da das exemplarische Video gerade einmal 655 Frames lang ist, können unbedenklich alle Frames genutzt werden. Des Weiteren kann zwischen **PNG** und **JPG** entschieden werden. Die Entscheidung bringt die üblichen Vor- und Nachteile der beiden Formate.

- **PNG:** Verlustfreie Komprimierung → Größere Dateien → Kein Qualitätsverlust
- **JPG:** Verlustbehaftete Komprimierung → Kleinere Dateien → Qualitätsverlust

Da Speicherplatz für Videos dieser Länge nicht der entscheidende Faktor ist, kann hier die höhere Qualität von **PNG** genutzt werden.

Das **dst**-Video wird immer komplett extrahiert, da die Frames am Ende der Pipeline wieder zu einem Video zusammengesetzt werden. Es müssen alle Frames vorhanden sein, um ein flüssiges Endergebnis zu gewährleisten. Es kann ebenfalls das Bildformat ausgewählt werden, dieses sollte gleich gewählt werden wie beim ersten Video.

Face Extraction

Nun müssen die Gesichter aus den Bildern extrahiert werden. Im Folgenden wird der Prozess für das **src**-Material (in DFL Schritt 4.X) beschrieben. Für das **dst**-Material verläuft der Prozess (als Schritt 5.X) analog.

Es gibt zwei Varianten, die Gesichter aus den Frames zu extrahieren.

- 4) data_src faceset extract MANUAL.bat
- 4) data_src faceset extract.bat

Wie die Namen vermuten lassen, werden die Gesichter einmal händisch und einmal durch ein vortrainiertes CNN extrahiert. Bei der automatischen Extraktion müssen im Nachhinein ggf. falsch erkannte Gesichter manuell gelöscht werden. Allerdings ist dieser Arbeitsaufwand bei weitem geringer als die manuelle Extraktion. Bei der Ausführung

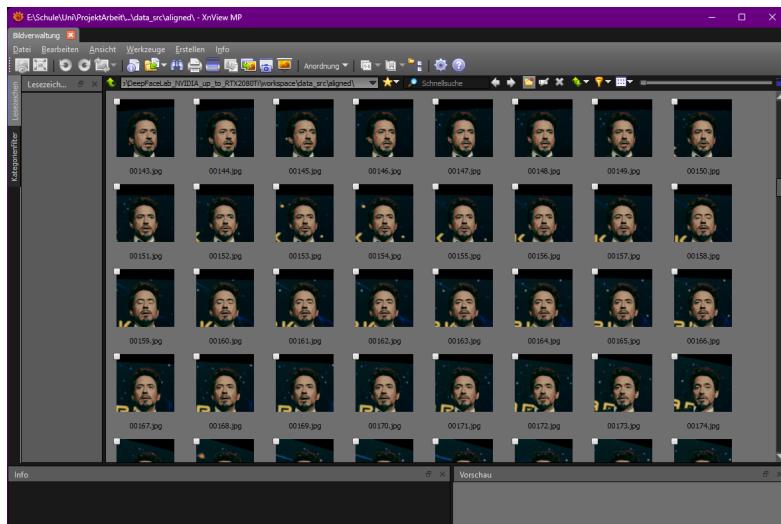


Abbildung 5.4: XnView (Rust Explorer)

können verschiedene Punkte konfiguriert werden. Der **face type** gibt an, wie viel vom Gesicht extrahiert werden soll.

- **f (face):** Nur das Gesicht
- **wf (whole face):** Das ganze Gesicht inklusive Stirn und Kinn
- **h (head):** Der gesamte Kopf inklusive Haare

Max numbers of faces gibt an, wie viele Gesichter pro Frame extrahiert werden sollen. Dieser Wert sollte auf 0 (alle) gesetzt werden. Ist mehr als ein Gesicht in den Videos zu sehen, werden alle Gesichter extrahiert; nicht benötigte Bilder können anschließend wieder gelöscht werden. Sind es zu viele Gesichter bzw. wird die Verarbeitungszeit zu hoch, muss entweder manuell oder mit einer Obergrenze extrahiert werden. Im ersten Fall fällt erhöhter Arbeitsaufwand an, im zweiten Fall fällt der Datensatz kleiner aus, da das gewünschte Gesicht ggf. übersprungen wird. Die Beispielvideos bestehen nur aus einem bzw. zwei Gesichtern und können unproblematisch automatisiert extrahiert werden. Anschließend kann entweder im Standard Windows Explorer unter `workspace/data_src/aligned` oder mithilfe des mitgelieferten Explorers die Daten gesichtet werden.

4.1) data_src view aligned result.bat

Dieses Skript öffnet den in Rust implementierten Explorer **XnView** (Abbildung 5.4), welcher auf das schnelle Anzeigen von Bildern optimiert wurde. Nun müssen alle Bilder, die nicht richtig erkannt wurden, gelöscht werden. Dabei stellt DFL einige Sortierungsmöglichkeiten zur Verfügung, die den Prozess erleichtern. Diese sind zu einem Skript zusammengefasst.

4.2) data_src sort.bat

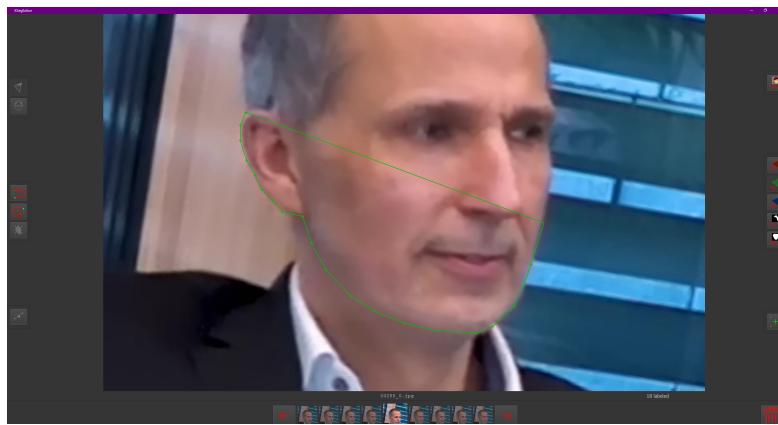


Abbildung 5.5: Polygon zeichnen im XSeg-Editor

Durch das Sortieren nach [0] `blur` und [1] `motion_blur` können schnell unscharfe Bilder ausfindig gemacht werden. Durch die Sortierung nach [5] `histogram similarity` werden ähnliche Gesichter zusammen gruppiert. So können Bilder einer nicht erwünschten zweiten Person einfach entfernt werden.

Ist die Auswahl der Gesichter abgeschlossen, können diese noch bei Bedarf mit AI-Upscaling vergrößert werden. Dies sollte nur gemacht werden, wenn die Bilder sonst unscharf oder zu klein sind. Besser ist es, direkt scharfe, hoch aufgelöste Bilder bzw. Ausgangsvideos zu verwenden. Das gesamte bisher beschriebene Vorgehen ist für das `dst`-Material identisch.

XSeg Mask

Sind alle Bilder entsprechend gesichtet und aussortiert, muss eine `XSeg Mask` angewandt werden. Diese Maske erfasst das ganze Gesicht mit seinen genauen Umrissen. Wenn Deepfakes im *face* oder *whole-face* Modus gemacht werden, kann eine pretrained generische Maske angewandt werden.

```
5.XSeg Generic) data_src whole_face mask - apply.bat
5.XSeg Generic) data_dst whole_face mask - apply.bat
```

Für den *head* Modus muss ein eigenes `XSeg-Model` trainiert werden. Dafür werden mehrere Bilder benötigt, in die händisch der gewünschte Umriss gezeichnet wird. Im Beispiel wäre das der gesamte Kopf, einschließlich Haaren.

```
5.XSeg) data_dst mask - edit.bat
5.XSeg) data_src mask - edit.bat
```

Diese Skripte öffnen den XSeg-Editor, in dem die Polygone gezeichnet (Abbildung 5.5) werden können. Es sollten je nach gewünschter Genauigkeit mehrere Dutzend bis wenige

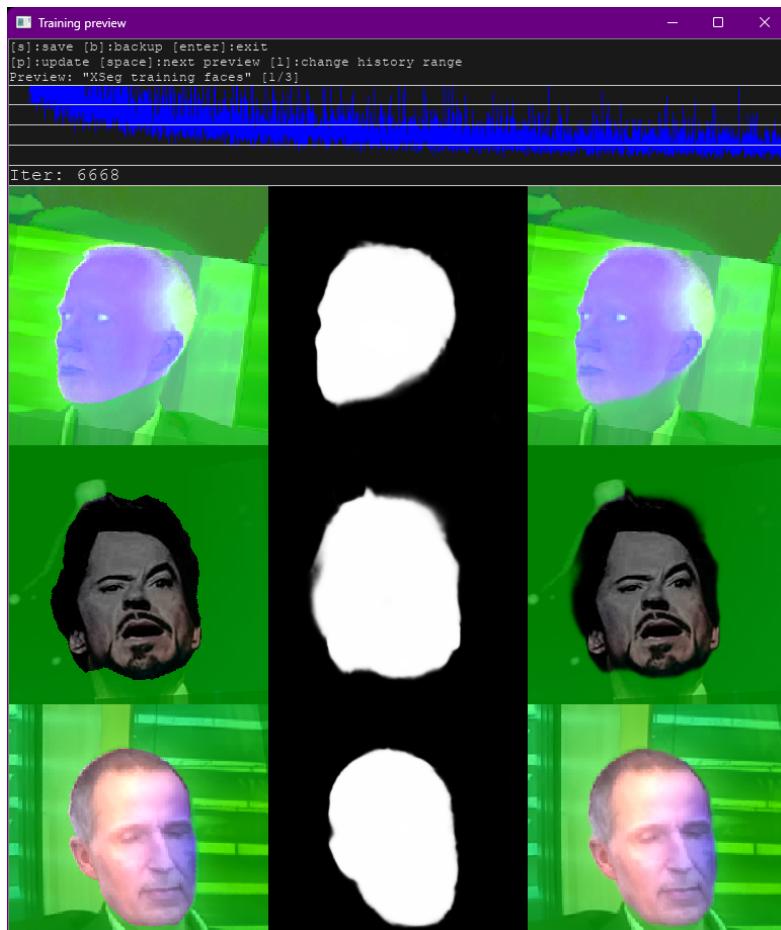


Abbildung 5.6: Training von XSeg im *head* Modus

hundert Bilder markiert werden. Dabei ist es entscheidend, möglichst viele verschiedene Ausdrücke und Kopfstellungen zu markieren und die Größe der Markierung konsistent zu halten. Z.B. sollte die Kieferkontur immer gleich gezeichnet werden oder die Menge der Stirn, die maskiert werden soll. Wurde der Prozess für `src` und `dst` durchgeführt, können die markierten Bilder in einen anderen Ordner verschoben werden und die Maske trainiert werden. Dabei werden die maskierten Bilder verwendet und zusätzlich automatisch verzerrt und eingefärbt (Abbildung 5.6).

```
5.XSeg) data_src mask - fetch.bat
5.XSeg) data_dst mask - fetch.bat
5.XSeg) train.bat
```

Das Training sollte so lange fortgesetzt werden, bis die Konturen der Gesichter klar erkennbar sind. Dies sollte i.d.R. nach einigen Tausend Iterationen der Fall sein (Abbildung 5.7).



Abbildung 5.7: Training Ergebnis von XSeg im *head* Modus



Abbildung 5.8: Vergleich: Angewandtes XSeg-Model (highlight) und manuelle Markierung (grün)

Anschließend muss die trainierte Maske auf `src` und `dst` angewandt werden.

```
5 .XSeg) data_src trained mask – apply.bat
5 .XSeg) data_dst trained mask – apply.bat
```

Danach kann die trainierte Maske im XSeg-Editor überprüft werden (Abbildung 5.8). Falls die Maske an manchen Stellen noch nicht passt, sollten diese Frames manuell markiert werden und danach das Training fortgeführt werden. Anschließend muss die Maske neu angewandt werden.

Facepack Erstellung

Der letzte (optionale) Schritt der Extraktionsphase ist das Erstellen eines Facepacks. Dies ist eine komprimierte Version der bisher geleisteten Arbeit. Die bisherigen Schritte erforderten vergleichsweise wenig Rechenleistung; durch Facepacks lassen sich die Daten einfacher auf z.B. einen stärkeren Computer übermitteln, außerdem verkürzen Facepacks die Initialisierungszeit für die folgenden Schritte, da nicht tausende Einzeldateien geladen werden müssen. DFL speichert alle Informationen in den Bilddateien, daher könnten die Bilder auch einfach in einem Archiv verschickt werden. Eine `.pak`-Datei ist allerdings der von DFL bevorzugte Weg. Ein Facepack beinhaltet immer nur die Informationen zu einem Gesicht, nicht einem Gesichterpaar. Es müssen also zwei Pakete erstellt werden.

```
4 .2) data_src util faceset pack.bat
5 .2) data_dst util faceset pack.bat
```

5.2.6 Training

Es gibt verschiedene Modelle, die für das Training genutzt werden können. Die wesentlichen Unterschiede sind:

- **SAEHD (6GB+):** Sparse Auto Encoder HD – geeignet für GPUs mit mindestens 6GB VRAM. Einstellbar und für die meisten Benutzer empfohlen.
- **AMP (6GB+):** Neues Modell mit unterschiedlicher Architektur, das versucht, die Form des `src`-gesichtes beizubehalten. Für GPUs mit mindestens 6GB VRAM. Das AMP-Modell befindet sich noch in der Entwicklung. Es wird empfohlen, zunächst mit SAEHD zu arbeiten.
- **Quick96 (2-4GB):** Einfacher Modus für GPUs mit 2-4GB VRAM. Feste Parameter: 96x96 Pixel Auflösung, `whole-face`, Batchgröße 4, DF-UD-Architektur. Wird hauptsächlich für schnelle Tests verwendet.

. Für das exemplarische Training wurde zuerst ein Testmodell mit Quick96 trainiert. Und anschließend das richtige Modell mit der SAEHD-Architektur erstellt.

Für die Konfiguration der Modelle sind ebenfalls einige Einstellungsmöglichkeiten vorhanden. Einige davon können nach der Initialisierung nicht mehr geändert werden, da diese großen strukturellen Einfluss auf das Modell haben. Dazu gehören:

- Model resolution (Oft abgekürzt mit: "res")
- Model architecture ("archi")
- Models dimensions ("dims")
- Face type
- Morph factor (nur bei AMP training)

Trainingskonfiguration

Die Konfiguration von DeepFake-Modell-Trainingsparameter bietet eine Vielzahl von Optionen, um die Qualität und Effizienz des Trainings zu optimieren. Nachfolgend wird auf die verschiedenen Parameter und deren empfohlene Einstellungen ausführlich eingegangen:

Autobackup every N hour (0-24): Diese Option ermöglicht die automatische Sicherung des Modells in regelmäßigen Abständen. Ein Wert von 0 deaktiviert diese Funktion. Die automatische Sicherung ist besonders wichtig, um den Verlust von Trainingsfortschritten zu vermeiden. Im Falle eines unerwarteten Systemabsturzes oder anderer Probleme kann das Modell von der letzten Sicherung wiederhergestellt werden. Die Backups bewegen sich je nach `Resolution` zwischen mehreren Hundert MB und einigen GB. Ein

Modell mit 224 `res` war im Test ca. 500MB groß, ein 384 `res` schon ca. 2GB.

Write preview history (y/n): Diese Einstellung speichert Vorschaubilder während des Trainings in regelmäßigen Abständen. Wenn aktiviert, wird im Weiteren abgefragt ob die Bilder zufällig oder manuell ausgewählt werden sollen. Im zweiten Fall wird ein weiteres Fenster geöffnet, in dem die zu speichernden Bilder manuell ausgewählt werden können. Das Speichern von Vorschaubildern ist nützlich, um den Fortschritt des Trainings zu überwachen und frühzeitig Probleme zu erkennen. Beispielsweise können Artefakte oder andere unerwünschte Effekte im Trainingsprozess sofort bemerkt und behoben werden.

Target iteration: Diese Einstellung bestimmt, nach wie vielen Iterationen das Training beendet wird.

Flip SRC faces randomly (y/n): Das zufällige horizontale Spiegeln der `src`-Gesichter kann hilfreich sein, um alle Winkel im `dst`-Datensatz abzudecken. Durch das Spiegeln können mehr Variationen des Gesichts erzeugt werden, was die Generalisierungsfähigkeit des Modells verbessern kann. Allerdings kann es zu unnatürlichen Ergebnissen führen, da Gesichter nie perfekt symmetrisch sind und spezifische Merkmale von einer Seite zur anderen übertragen werden können. Die Funktion sollte nur in frühen Phasen des Trainings oder gar nicht aktiviert werden. Durch das Auswählen guter Ausgangsvideos werden die Vorteile dieser Option überflüssig.

Flip DST faces randomly (y/n): Diese Option verbessert die Generalisierung, wenn das zufällige Spiegeln der `src`-Gesichter deaktiviert ist. Durch das Spiegeln der `dst`-Gesichter können ähnliche Vorteile wie bei den `src`-Gesichtern erzielt werden, insbesondere wenn das Spiegeln der `src`-Gesichter deaktiviert ist. Dies kann die Vielfalt der Trainingsdaten erhöhen und das Modell robuster machen.

Batch_size: Die Batch-Größe beeinflusst die Anzahl der Gesichter, die in jeder Iteration verglichen werden. Eine höhere Batch-Größe liefert bessere Ergebnisse, da mehr Daten pro Iteration verarbeitet werden, benötigt jedoch mehr VRAM und verlängert die Trainingszeit. Eine niedrige Batch-Größe kann die Trainingsgeschwindigkeit erhöhen, führt jedoch zu weniger genauen Ergebnissen. Empfohlene Werte liegen zwischen 6 und 12, um ein gutes Gleichgewicht zwischen Trainingszeit und Ergebnisqualität zu erreichen.

Resolution (64-640): Die Auflösung des Modells beeinflusst die Detailgenauigkeit der trainierten Gesichter. Höhere Auflösungen führen zu detaillierteren Gesichtern, erfordern jedoch mehr Rechenleistung und verlängern die Trainingszeit erheblich. Diese Einstellung kann während des Trainings nicht geändert werden, daher sollte sie sorgfältig gewählt werden. Eine höhere Auflösung ist vorteilhaft, wenn die Gesichter im endgültigen Video

sehr detailliert sein sollen.

Face type (h/mf/f/wf/head): Diese Option legt den zu trainierenden Gesichtsbereich fest:

- **HF (Half Face):** Nur der Bereich vom Mund bis zu den Augenbrauen.
- **MHF (Mid Half Face):** Deckt 30% mehr des Gesichts ab als HF und reduziert das Risiko, dass wichtige Gesichtsteile abgeschnitten werden.
- **FF (Full Face):** Deckt den größten Teil des Gesichts ab, schließt jedoch die Stirn aus.
- **WF (Whole Face):** Deckt das gesamte Gesicht einschließlich der Stirn ab und sorgt so für eine vollständigere Gesichtsabdeckung.
- **HEAD (Head):** Tauscht den gesamten Kopf aus (nicht geeignet ist für Personen mit langen Haaren).

AE architecture (df/liae - Varianten): Diese Option ermöglicht die Auswahl zwischen zwei Hauptarchitekturen des SAEHD-Modells: DF und LIAE sowie deren Varianten. Jede Variante hat spezifische Vor- und Nachteile:

- **DF:** Bietet eine bessere Ähnlichkeit zum Quellgesicht auf Kosten schlechterer Licht- und Farbanpassung. Diese Architektur erfordert, dass das Quellset besser an die Winkel und Lichtverhältnisse des Zielsets angepasst ist.
- **LIAE:** Bietet eine bessere Anpassung an Licht und Farbe und ist toleranter gegenüber unterschiedlichen Gesichtsproportionen. Diese Architektur benötigt mehr VRAM und GPU Leistung.

AutoEncoder dimensions (32-2048): Bestimmt die allgemeine Fähigkeit des Modells, Gesichter zu lernen.

Encoder dimensions (16-256): Beeinflusst die Fähigkeit des Encoders, Gesichter zu lernen.

Decoder dimensions (16-256): Beeinflusst die Fähigkeit des Decoders, Gesichter wiederherzustellen.

Decoder mask dimensions (16-256): Beeinflusst die Qualität der gelernten Masken.

Morph factor (0.1-0.5) (Nur bei AMP): Beeinflusst, wie stark das Modell die vorhergesagten Gesichter an die Quellgesichter anpasst. Ein höherer Wert kann zu einer höheren Ähnlichkeit führen, jedoch auf Kosten der Realismus des Zielgesichts. Empfohlener Wert ist 0.5.

Masked training (y/n) (Nur bei AMP): Priorisiert das Training der maskierten Bereiche, um sicherzustellen, dass der Fokus des Modells auf den relevanten Teilen des Gesichts liegt.

Eyes and mouth priority (y/n): Verbessert die Schärfe und Detailgenauigkeit der

Augen und des Mundes, indem diese Bereiche während des Trainings stärker gewichtet werden.

Uniform yaw distribution of samples (y/n): Hilft beim Training von Profilgesichtern, indem es das Modell zwingt, gleichmäßig auf alle Gesichtsrichtungen zu trainieren. Dies kann besonders nützlich sein, wenn das Quellset nicht viele Profilaufnahmen enthält.

Blur out mask (y/n): Macht den Bereich außerhalb der Maske weicher, um eine glattere Übergangszone zu schaffen und Artefakte zu reduzieren.

Place models and optimizer on GPU (y/n): Verbessert die Leistung, indem alle Berechnungen auf der GPU durchgeführt werden. Dies erhöht jedoch den VRAM-Verbrauch erheblich.

Use AdaBelief optimizer? (y/n): Erhöht die Genauigkeit und Qualität der trainierten Gesichter durch einen besseren Optimierungsalgorithmus, erhöht jedoch den VRAM-Verbrauch.

Use learning rate dropout (y/n/cpu): Beschleunigt das Training und reduziert Fluktuation. Kann auf der CPU ausgeführt werden, um VRAM zu sparen, was jedoch die Trainingszeit verlängert.

Enable random warp of samples (y/n): Wird verwendet, um das Modell zu generalisieren, indem es zufällige Verzerrungen auf die Trainingsbilder anwendet.

Random hue/saturation/light intensity: Verbessert die Farbstabilität der Quell-Daten während des Trainings, indem zufällige Änderungen von Farbton, Sättigung und Helligkeit angewendet werden. Empfohlener Wert ist 0.05.

GAN power (0.0-5.0): Wird zur Erzielung schärferer und detaillierterer Gesichter verwendet.

Face style power (0.0-100.0): Kontrolliert die Stilübertragung des Gesichts, um die Beleuchtung und Farben des Zielgesichts besser anzupassen.

Background style Power (0.0-100.0): Kontrolliert die Stilübertragung des Hintergrunds.

Color transfer for src faceset (none/rct/lct/mkl/idt/sot): Methoden zur Anpassung der Farben der Quell-Daten an die Ziel-Daten, um Farbabweichungen zu minimieren:

- **None:** Keine Farbanpassung, kann in manchen Fällen bessere Ergebnisse liefern.
- **RCT (Reinhard Color Transfer):** Basierend auf der Reinhard-Farbübertragung.
- **LCT (Linear Color Transfer):** Passt die Farbdarstellung des Zielbildes an die des Quellbildes an.
- **MKL (Monge-Kantorovich Linear):** Basierend auf der Monge-Kantorovich-Theorie.
- **IDT (Iterative Distribution Transfer):** Iterative Verteilungstransfermethode.
- **SOT (Sliced Optimal Transfer):** Optimale Transfertyp, die Leistungseinbußen während des Trainings und der Zusammenführung verursachen kann.

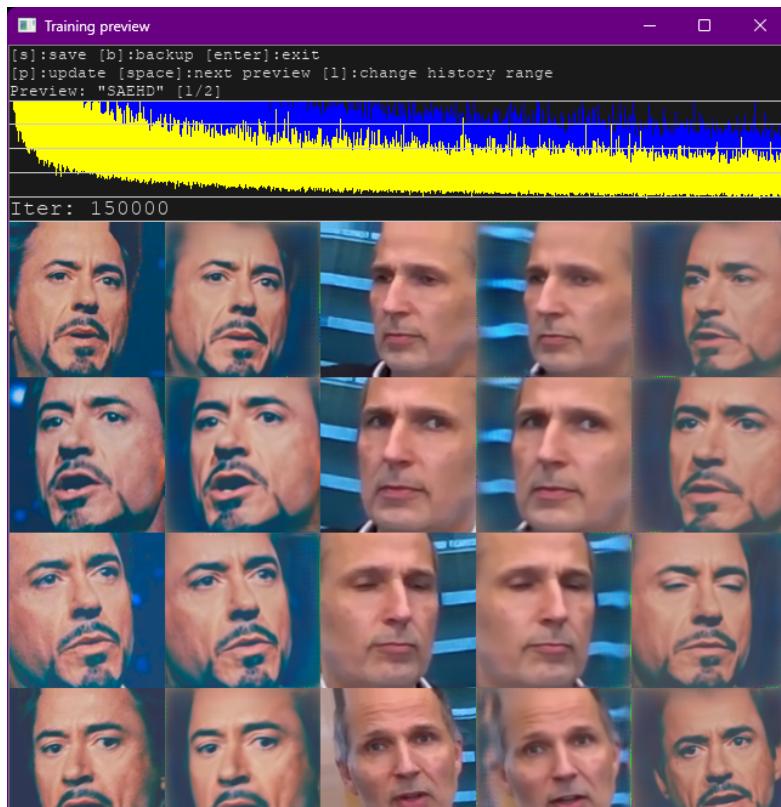


Abbildung 5.9: SAEHD Training 128 res

Welche der Farbtransfer-Algorithmen am besten geeignet ist, muss abhängig vom Datensatz ausprobiert werden. **Enable gradient clipping (y/n):** Verhindert den Modellkollaps, der durch die Verwendung verschiedener Funktionen verursacht werden kann.

Enable pretraining mode (y/n): Wie bereits in früheren Kapiteln beschrieben. Sehr empfehlenswert für Modelle mit -D Architekturvarianten.

Nach der Konfiguration des Modells und Training, wird überprüft, ob die genügenden Hardware (insbesondere RAM und VRAM) verfügbar ist. Anschließend startet das Training. Im neu geöffneten Fenster (Abbildung 5.9) sind 5 Spalten zu erkennen. Diese beinhalten (von links nach rechts):

1. Ein Original **src**-Bild
2. Die Rekonstruktion des **src**-Gesichts
3. Ein Original **dst**-Bild
4. Die Rekonstruktion des **dst**-Gesichts

5. Das `src`-Gesicht mit den Ausdrücken des `dst`-Gesichts

Das Konsolenfenster zeigt nochmals die gewählten Einstellungen, sowie einige anderen Informationen.

1. Die aktuelle Zeit
2. Die aktuelle Iteration
3. Die Dauer der letzten Iteration
4. `src loss value`
5. `dst loss value`

Loss values sind Werte die angeben, wie verschieden das rekonstruierte Gesicht vom tatsächlichen Gesicht ist. Über die Zeit werden diese Werte immer niedriger und gehen gegen Null. Bei einem unzureichend vorbereiteten Datensatz konvergieren die Werte nicht gegen Null.

Trainings Workflow und Ende

In den Tests hat sich ein Vorgehen herauskristallisiert, dass zu weitestgehend zufriedenstellenden Ergebnissen geführt hat. Dies bestand aus:

1. Pretraining
2. Training mit `Uniform yaw distribution of samples` (sonst default)
3. Training mit `Eyes and mouth priority`
4. Default training
5. Training mit `GAN Power`

Es lässt sich kein universelles Setup formulieren das über verschiedenes `src`- und `dst`-Material hinweg gute Ergebnisse liefert. Außerdem ist auch bei gut gewählten Einstellungen, das Ausgangsmaterial maßgeblich entscheidend für die Qualität der Deepfakes.

Werden die **Loss values** nicht mehr kleiner, kann das Training beendet werden. Außerdem kann in der Preview Ansicht der aktuelle Stand des Modells überprüft werden. Ein Training kann auch jederzeit wieder fortgesetzt werden, falls die Ergebnisse des exportierten Materials nicht zufriedenstellend sind.

Ist ein Modell trainiert, kann dieses als eine `.dfm`-Datei exportiert werden. Diese Dateien können in **DeepFaceLive** importiert und verwendet werden, um Echtzeit Face swapping durchzuführen. Der Trainingsprozess für ein DeepFaceLive Modell unterscheidet sich in einigen Punkten von dem bisher beschriebenen Prozess und wird in Kapitel 5.3 näher betrachtet.



Abbildung 5.10: Unterschiede: `overlay`, `hist-match`, `seamless`, `seamless histmatch`

5.2.7 Conversion/Merging

Nach abgeschlossenem Training muss das trainierte Modell noch auf das Zielvideo angewandt werden. Dafür bietet DFL zahlreiche Konfigurationsmöglichkeiten. Diese sind jedoch von Fall zu Fall verschieden und müssen individuell angepasst werden. Ein Vergleich einer dieser Konfigurationsmöglichkeiten zeigt Abbildung 5.10.

Ist das Ergebnis zufriedenstellend, können die neu generierten Bilder wiederrum mit FFmpeg zu einem Video zusammengefügt werden. DFL bietet hierfür wiederrum einen Wrapper.

```
8) merged to mp4.bat
```

5.3 DeepFaceLive

DFLive ist ein weiteres Open-Source-Tool zur Erstellung von Deepfakes, das sich auf Echtzeitanwendungen konzentriert. Es ermöglicht die nahtlose Integration von Gesichtsaustausch in Video-Streams, was die Software für Social Engineering interessant macht.

5.3.1 Motivation

Die Hauptmotivation hinter DFLive ist es, die Möglichkeiten der Echtzeit-Gesichtsmanipulation zu erforschen und zu erweitern. Dadurch wird es möglich, Gesichter in Videos während des Streamings in Echtzeit auszutauschen, was neue Anwendungen und Herausforderungen sowohl im Bereich der Unterhaltung als auch der Sicherheit mit sich bringt.

5.3.2 Fähigkeiten

- **Reenactment von Bildern:** Schneller und einfacher, bei gutem Ausgangsbild akzeptable Ergebnisse

- **Face Swapping mit Bildern:** Schneller und einfacher, bei gutem Ausgangsbild akzeptable Ergebnisse
- **Face Swapping eines DFM Modells:** Erfordert Modelltraining in DFL, dafür bessere Ergebnisse

5.3.3 Workflow

Der Workflow von DeepFaceLive umfasst mehrere Schritte, um eine nahtlose Echtzeit-Gesichtsmanipulation zu gewährleisten:

Initialisierung

Die Initialisierung umfasst das Einrichten der Umgebung und das Laden der notwendigen Modelle. Nutzer können Modelle aus dem Internet verwenden oder eigene Modelle trainieren, um spezifische Anforderungen zu erfüllen. Die Pipeline von DFLive besteht aus mehreren Schritten.

Data Source

Das `dst`-Video kann entweder per Datei oder als Kamera-Stream bereitgestellt werden. Das Verwenden einer Datei hat im Kontext von Social Engineering allerdings keine Relevanz.

Detection, Alignment and Marking

Der nächste Schritt ist das Erkennen von Gesichtern im Ausgangsmaterial. Hierfür stehen verschiedene Algorithmen bzw. Neuronalen Netze zur Verfügung: CenterFace, S3FD oder YoloV5. Das erkannte Gesicht wird anschließend mittig im Bild ausgerichtet und mit Landmarks markiert. Auch für diesen Schritt stehen verschiedene Möglichkeiten zur Verfügung: OpenCV LBF, Google FaceMesh oder InsightFace_2D106. Die Standardeinstellungen sind in den meisten Fällen ausreichend, ggf. können auch andere Modelle getestet werden.

Face Swapping/Reenactment

Nachdem das dst-Material nun für den Tausch vorbereitet wurde kann zwischen drei Möglichkeiten gewählt werden. **Face animator** bietet die Möglichkeit des Reenactments eines Bildes. Dafür ist es empfehlenswert ein gut belichtetes, neutrales Bild zu verwenden. Diese Variante ist für Social Engineering Angriffe bedingt geeignet. Nutzt man die Technik und legt im Nachhinein flasche Störsignale über das Video, werden die Nachteile dieses einfachen Modells überdeckt. Diese Art kann also nur in Video-Chats verwendet werden.

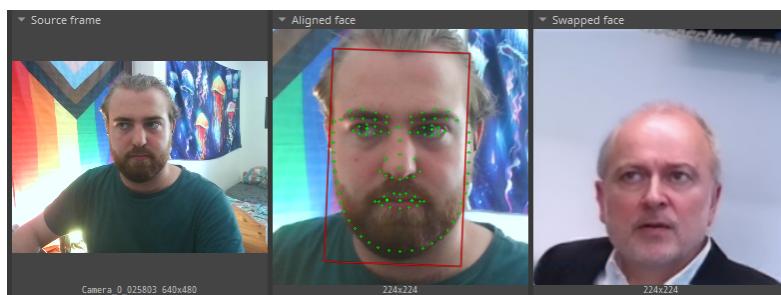


Abbildung 5.11: Face Animator Pipeline

Face swap (Insight) bietet die Möglichkeit ein Gesicht aus einem einzelnen Bild auf ein Zielgesicht zu swappen. Hier ist die Qualität des Bildes ebenfalls entscheidend.

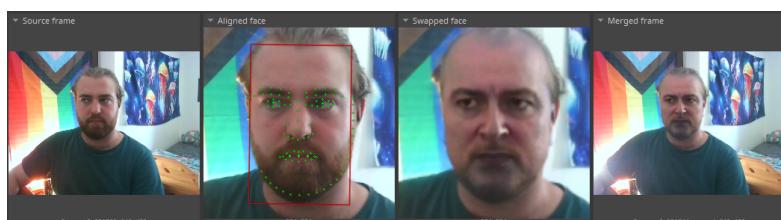


Abbildung 5.12: Face Swap (Insight)

Face swap (DFM) benutzt in DeepFaceLab (DFL) trainierte Modelle zum Ersetzen der Gesichter. Dies bietet den Vorteil, dass das Gesicht (bei richtigem Training) von verschiedenen Blickwinkeln und Expressionen richtig dargestellt wird. Die Ergebnisse sind natürlicher als die, die nur auf einem Bild basieren. Allerdings fordert diese Variante einen deutlich höheren Mehraufwand. Es müssen zuerst genügend Bilder zusammengetragen werden und ein Modell trainiert werden. Außerdem muss beim Face swap eine Person, die ähnlich wie die Zielperson aussieht den Deepfake bzw. Social Engineering Angriff durchführen. Das in Abbildung 5.13 gezeigt Modell wurde, auf Grundlage eines 10.000.000 Iterationen vorgebildeten Modells, ca. 1.000.000 Iterationen auf Harald Riegel trainiert. Dies bedeutete beim genannten Testsetup ca. 10 Tage Rechenaufwand.

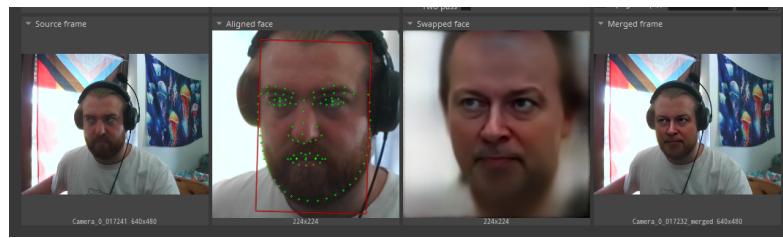


Abbildung 5.13: Face Swap (DFM)

Face Merger und Output

Wird ein Gesicht getauscht, muss das Zielgesicht wieder mit dem Originalbild zusammengefügt werden. Diese Aufgabe übernimmt der **Face Merger**, hier können ebenfalls verschiedene Einstellungen geändert werden, die aus DFL bekannt sind. Für Reenactment ist dies nicht nötig. Anschließend kann der modifizierte Video-Stream entweder in einer Datei gespeichert werden, per `mpegts udp` gestreamt oder in einem neuen Fenster angezeigt werden. Durch Tools wie die OBS (Open Broadcaster Software) kann dieses Fenster wiederrum als virtuelle Kamera in anderen Anwendungen verwendet werden.

RTT Model Training

Das Trainieren eines RTT (Real Time Transfer) Models, weicht von der in Kapitel 5.2 beschriebenen Methode ab. Im wesentlichen unterscheiden sich die zum Training verwendeten Datensätze. Für ein RTT Model muss das Gesicht das später als `.dfm`-Datei in DFLive verwendet werden soll gewohnt als `src` verwendet werden. Dieses Mal wird das Modell allerdings nicht auf ein Zielgesicht, sondern auf möglichst viele verschiedene Gesichter trainiert. Dafür sind entsprechende RTT Facepacks im Internet zu finden. Pretrained Models können allerdings trotzdem verwendet werden. Sonst folgt der Workflow dem in Abschnitt 5.2 beschriebenen.

6. Zusammenfassung

Social Engineering

Social Engineering ist eine Methode, die gezielt menschliches Verhalten manipuliert, um unautorisierten Zugang zu Informationen oder Systemen zu erlangen. Die Ausarbeitung hebt hervor, dass Angriffe wie Phishing, Pretexting und Baiting besonders effektiv sind, da sie auf psychologische Tricks setzen, um Benutzer zur Preisgabe sensibler Informationen zu verleiten. Phishing beispielsweise wird häufig durch E-Mails oder Nachrichten ausgeführt, die vorgeben, von vertrauenswürdigen Quellen zu stammen. Pretexting wiederum nutzt falsche Identitäten, um gezielt Informationen zu erlangen. Diese Angriffe sind besonders gefährlich, da sie unabhängig von technischen Sicherheitsmaßnahmen erfolgreich sein können und auf die Schwachstelle Mensch abzielen.

Deepfakes

Deepfakes stellen eine moderne Bedrohung dar, die durch den Einsatz von KI und Maschinellem Lernen (ML) ermöglicht wird. Durch die Verwendung von GANs können täuschend echte Videos und Bilder erstellt werden, die kaum von echten Inhalten zu unterscheiden sind. In der Ausarbeitung werden auf Risiken hingewiesen, die mit Deepfakes einhergehen, da sie zur Verbreitung von Desinformation, zur Rufschädigung und zur Manipulation der öffentlichen Meinung genutzt werden können. Die Fähigkeit, visuelle Inhalte auf diese Weise zu fälschen, stellt eine erhebliche Herausforderung für die Wahrnehmung und das Vertrauen in digitale Medien dar.

Erkennungsmechanismen von Deepfakes

Die Erkennung von Deepfakes ist ein komplexes Unterfangen, da die Technologie stetig fortschreitet und die Ergebnisse immer realistischer werden. In der Ausarbeitung werden verschiedene Erkennungsmethoden beschrieben, wie die Analyse von Blinkmustern, die Überprüfung von Unregelmäßigkeiten in Augenbewegungen und die Untersuchung von Beleuchtungsunterschieden. Besonders fortschrittlich ist die Methode der Corneal Reflections, bei der die Reflexionen in den Augen einer Person analysiert werden, um Anomalien zu entdecken.

Auswirkungen und Gegenmaßnahmen

Die potenziellen Gefahren, die sowohl von Social Engineering als auch von Deepfakes ausgehen, werden in der Ausarbeitung detailliert beschrieben. Im Falle von Social Engineering wird betont, dass Aufklärung und Schulung der Nutzer eine der wirksamsten

Maßnahmen darstellen, um solche Angriffe abzuwehren. Für den Umgang mit Deepfakes werden technologische Erkennungsmethoden sowie rechtliche Maßnahmen vorgeschlagen, um Missbrauch vorzubeugen. Gleichzeitig wird die Notwendigkeit unterstrichen, das Bewusstsein der Öffentlichkeit für die Existenz und die Gefahren von Deepfakes zu schärfen, um deren negativen Einfluss auf die Gesellschaft zu minimieren.

Schlussfolgerung

Beide Themen verdeutlichen, wie moderne Technologien und psychologische Manipulationen genutzt werden können, um Schaden anzurichten. Um diesen Bedrohungen effektiv zu begegnen, ist es unerlässlich, die Erkennungstechnologien und präventiven Maßnahmen kontinuierlich weiterzuentwickeln. Nur durch eine Kombination aus technischer Innovation und umfassender Aufklärung können diese Gefahren eingedämmt werden.

Literaturverzeichnis

- [1] S. Direct. (). „Magnitude Spectrum“ [Online]. Verfügbar: <https://www.sciencedirect.com/topics/computer-science/magnitude-spectrum>
- [2] ComputerScienceWiki. (27. Feb. 2018). „Max-pooling / Pooling“ [Online]. Verfügbar: https://computersciencewiki.org/index.php/Max-pooling_-_Pooling
- [3] J. Richter. (6. Jan. 2020). „Temporal Convolutional Networks für die Sequenz-Modellierung“ [Online]. Verfügbar: <https://dida.do/de/blog/temporal-convolutional-networks-fuer-sequenzmodellierung>
- [4] CrowdStrike. (4. Juli 2024). „What is Cyber Threat Intelligence? [Beginner’s Guide]“ [Online]. Verfügbar: <https://www.crowdstrike.com/cybersecurity-101/threat-intelligence/>
- [5] appmeisterei. (). „Variational Auto-Encoder (VAE)“ [Online]. Verfügbar: <https://www.appmeisterei.de/app-entwicklung-glossar/Variational-Auto-Encoder-VAE>
- [6] M. Block. (29. Aug. 2023). „Definition und Anwendungsbereiche“ [Online]. Verfügbar: https://link.springer.com/chapter/10.1007/978-3-662-67427-7_2
- [7] L. Whittaker. (Juli 2023). „Mapping the deepfake landscape for innovation: A multidisciplinary systematic review and future research agenda“ [Online]. Verfügbar: <https://www.sciencedirect.com/science/article/pii/S0166497223000950#abs0015>
- [8] J. A. Marwan Albahar, „DEEPFAKES: THREATS AND COUNTERMEASURES SYSTEMATIC REVIEW“, *Journal of Theoretical and Applied Information Technology*, Jg. 97, Nr. 22, 2019. [Online]. Verfügbar: <https://www.jatit.org/volumes/Vol97No22/7Vol97No22.pdf>.
- [9] J.-T. Kötke. (Feb. 2021). „DEEPFAKE -EINE KURZE EINLEITUNG Deepfake -Eine kurze Einleitung“ [Online]. Verfügbar: https://www.researchgate.net/profile/Jennifer-Tia-Koetke/publication/373041489_DEEPFAKE_-EINE_KURZE_EINLEITUNG_Deepfake_-_Eine_kurze_Einleitung/links/64d4ffddd3e680065aac7ee3/DEEPFAKE-EINE-KURZE-EINLEITUNG-Deepfake-Eine-kurze-Einleitung.pdf
- [10] A. Köcher. (). „Whitepaper zum Thema ”Deepfakes““ [Online]. Verfügbar: <https://ai.hdm-stuttgart.de/downloads/student-white-paper/Sommer-2020/Deepfakes.pdf>
- [11] C. S. Dave Bermann. (23. Nov. 2023). „Was ist ein Autoencoder?“ [Online]. Verfügbar: <https://www.ibm.com/de-de/topics/autoencoder>

- [12] D. Cavedon-Taylor, „Deepfakes: a survey and introduction to the topical collection“, *Synthese*, Jg. 204, Nr. 1, S. 14, 2024, ISSN: 1573-0964. DOI: [10.1007/s11229-024-04634-8](https://doi.org/10.1007/s11229-024-04634-8). [Online]. Verfügbar: <https://doi.org/10.1007/s11229-024-04634-8>.
- [13] A. Chadha, V. Kumar, S. Kashyap und M. Gupta, *Deepfake: An Overview*, P. K. Singh, S. T. Wierzchoń, S. Tanwar, M. Ganzha und J. J. P. C. Rodrigues, Hrsg. Singapore: Springer Singapore, 2021, S. 557–566, ISBN: 978-981-16-0733-2.
- [14] iperov. (25. Juli 2024). „DeepFaceLab - Github repository“
- [15] Y. Nirkin, Y. Keller und T. Hassner, „FSGANv2: Improved Subject Agnostic Face Swapping and Reenactment“, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Jg. 45, Nr. 1, S. 560–575, Jan. 2023, PubMed-not-MEDLINE, PMID: 35471874, ISSN: 1939-3539, 0098-5589. DOI: [10.1109/TPAMI.2022.3155571](https://doi.org/10.1109/TPAMI.2022.3155571). [Online]. Verfügbar: <https://doi.org/10.1109/TPAMI.2022.3155571>.
- [16] Y. Nirkin, Y. Keller und T. Hassner, „FSGAN: Subject Agnostic Face Swapping and Reenactment“, in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, Okt. 2019.
- [17] K. Liu, I. Perov, D. Gao, N. Chervonyi, W. Zhou und W. Zhang, „Deepfacelab: Integrated, flexible and extensible face-swapping framework“, *Pattern Recognition*, Jg. 141, S. 109 628, 2023, ISSN: 0031-3203. DOI: <https://doi.org/10.1016/j.patcog.2023.109628>. [Online]. Verfügbar: <https://www.sciencedirect.com/science/article/pii/S0031320323003291>.
- [18] M. Westerlund, „The Emergence of Deepfake Technology: A Review“, *Technology Innovation Management Review*, Jg. 9, S. 40–53, Nov. 2019, ISSN: 1927-0321. DOI: <http://doi.org/10.22215/timreview/1282>. [Online]. Verfügbar: timreview.ca/article/1282.
- [19] H. Guo, X. Wang und S. Lyu, „Detection of Real-Time Deepfakes in Video Conferencing with Active Probing and Corneal Reflection“, in *ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2023, S. 1–5. DOI: [10.1109/ICASSP49357.2023.10094720](https://doi.org/10.1109/ICASSP49357.2023.10094720).
- [20] C. D. Hylander, P. Langlois, A. Pinto und S. Widup, „2024 Data Breach Investigations Report“, Verizon, Apr. 2024, Available at <https://www.verizon.com/business/resources/reports/dbir/>.
- [21] F. Salahdine und N. Kaabouch, „Social Engineering Attacks: A Survey“, *Future Internet*, Jg. 11, Nr. 4, 2019, ISSN: 1999-5903. DOI: [10.3390/fi11040089](https://doi.org/10.3390/fi11040089). [Online]. Verfügbar: <https://www.mdpi.com/1999-5903/11/4/89>.
- [22] K. Krombholtz, H. Hobel, M. Huber und E. Weippl, „Advanced social engineering attacks“, *Journal of Information Security and Applications*, Jg. 22, S. 113–122, 2015, Special Issue on Security of Information and Networks, ISSN: 2214-2126. DOI: <https://doi.org/10.1016/j.jisa.2014.09.005>. [Online]. Verfügbar: <https://www.sciencedirect.com/science/article/pii/S2214212614001343>.

- [23] BSI. (5. Juli 2024). „Social Engineering - der Mensch als Schwachstelle“ [Online]. Verfügbar: https://www.bsi.bund.de/DE/Themen/Verbraucherinnen-und-Verbraucher/Cyber-Sicherheitslage/Methoden-der-Cyber-Kriminalitaet/Social-Engineering/social-engineering_node.html
- [24] CNN. (5. Juli 2024). „Finance worker pays out \$25 million after video call with deepfake ‘chief financial officer’“ [Online]. Verfügbar: <https://edition.cnn.com/2024/02/04/asia/deepfake-cfo-scam-hong-kong-intl-hnk/index.html>
- [25] DSGVO-Portal. (5. Juli 2024). „Datenpanne bei Marriott International, Inc. | Sicherheitsvorfalls-Datenbank“ [Online]. Verfügbar: https://www.dsgvo-portal.de/sicherheitsvorfaelle/datenpanne_bei_marriott-international-inc.-1069.php
- [26] B. Computer. (5. Juli 2024). „Office 365 phishing attack impersonates the US Department of Labor“ [Online]. Verfügbar: <https://www.bleepingcomputer.com/news/security/office-365-phishing-attack-impersonates-the-us-department-of-labor/>
- [27] C. Blend. (7. Juli 2024). „Rogue One Deepfake Makes Star Wars’ Leia And Grand Moff Tarkin Look Even More Lifelike“ [Online]. Verfügbar: <https://www.cinemablend.com/news/2559935/rogue-one-deepfake-makes-star-wars-leia-and-grand-moff-tarkin-look-even-more-lifelike>
- [28] Y. W. Jonathan Shen Ruoming Pang, „NATURAL TTS SYNTHESIS BY CONDITIONING WAVENET ON MEL SPECTROGRAM PREDICTIONS“, *University of California, Berkeley*, Jg. 2, Nr. 1712.05884, 2018. [Online]. Verfügbar: <https://arxiv.org/pdf/1712.05884.pdf>.
- [29] R. A. S. Yuxuan Wang RJ Skerry-Ryan, „TACOTRON: TOWARDS END-TO-END SPEECH SYNTHESIS“, *University of California, Berkeley*, Jg. 2, Nr. 1703.10135, 2017. [Online]. Verfügbar: <https://arxiv.org/pdf/1703.10135.pdf>.
- [30] M. H. Yao-Yuan Yang. (). „Text-to-Speech with Tacotron2“ [Online]. Verfügbar: https://pytorch.org/audio/stable/tutorials/tacotron2_pipeline_tutorial.html
- [31] I. Perov, D. Gao, N. Chervoniy u. a., *DeepFaceLab: Integrated, flexible and extensible face-swapping framework*, 2021. arXiv: [2005.05535 \[cs.CV\]](https://arxiv.org/abs/2005.05535). [Online]. Verfügbar: <https://arxiv.org/abs/2005.05535>.