Hochschule für Angewandte Wissenschaften Hamburg
Hamburg University of Applied Sciences

Bachelorarbeit

May Al Methiab

Adversariale Angriffe auf KI-Modelle zur Tumorerkennung: Sicherheitsrisiken in der medizinischen Bildklassifikation

Fakultät Elektro-, Medien- und Informationstechnik
Department Elektro- und Informationstechnik

Faculty of Electrical Engineering, Media and Information Technology
Department of Elecrical and Information Engineering

May Al Methiab

Adversariale Angriffe auf KI-Modelle zur Tumorerkennung: Sicherheitsrisiken in der medizinischen Bildklassifikation

Bachelorarbeit eingereicht im Rahmen der Bachelorprüfung

im Studiengang Elektro- und Informationstechnik
am Department Elektro- und Informationstechnik
der Fakultät für Elektro-, Medien- und Informationstechnik
der Hochschule für Angewandte Wissenschaften Hamburg

Erstprüfer: Prof. Dr. Heike Neumann.
Zweitprüfer: Prof. Dr. Ing. Martin Lapke

Abgabedatum: 23.10.2025

Kurzreferat

May Al Methiab

Thema der Arbeit

Adversariale Angriffe auf KI-Modelle zur Tumorerkennung: Sicherheitsrisiken in der medizinischen Bildklassifikation

Stichworte

Faltungsnetz, adversariale Angriffe, White-Box-Angriff, Black-Box-Angriff, Fast Gradient Sign Method (FGSM), Basic Iterative Methode (BIM), Projected Gradient Descent (PGD), Tranfer Lernen, Generative Adversariale Netze (GANs), Rekrusive neuronale Netze (RNN), Differential Privatephäre, föderiertes Lernen, Gradienten Maskierung, Daten Vergiftung

Kurzzusammenfassung

CNN auf Kaggle-MRTs (mit/ohne Tumor) trainiert und mit Standardmetriken bewertet. Angriffe: FGSM, BIM, PGD (White-Box) + entscheidungsbasierter Black-Box; Verteidigung: adversariales Training (robusteres Modell) und Vorverarbeitungsmethode mit CLAHE-Funktion und Median-Filter.

May Al Methiab

Title of Thesis

Adversarial Attacks on AI Models for Tumor Detection: Security Risks in Medical Image Classification

Keywords

Convolutional Neural Network (CNN), adversarial attacks, white-box attack, Black-box attack, Fast Gradient Sign Method (FGSM), Basic Iterative Method (BIM), Projected Gradient Descent (PGD), Transfer Learning, Generative Adversarial Networks (GANs), Recurrent Neural Networks (RNNs),Differential Privacy, Federated Learning, gradient masking, data poisoning

Abstract

A CNN is trained on a Kaggle MRI dataset (tumor vs. non-tumor) and evaluated using standard metrics. Attacks: FGSM, BIM, PGD (white-box) plus a decision-based black-box attack; Defense: adversarial training (yielding a more robust model) and a preprocessing method using the CLAHE function and a median filter.

Inhaltsverzeichnis

Abbildungsverzeichnis

Tabellenverzeichnis

Abkürzungen

Abkürzung	**Bedeutung**
AE	Autoencoder
BIM	Basic Iterative Methode
CLAHE	Contrast Limited Adaptive Histogram Equalization
CNN	Convolutional Neural Network
CT	Computertomographie
CW	Carlini Wagner
DL	Deep learning
DP	Differential Privacy
FGSM	Fast Gradient Sign Methode
FN	False Negative
FP	False Positive
GAN	Generative Adversarial Networks
GRU	Gated Recurrent Unit
IDEs	integrated development environments
KI	Künzliche Inelligenz
LSTM	Long Short-Term Memory
MRT	Magnetresonanztomographie
PGD	Projected Gradient Descent
RBM	Restricted Boltzmann Machines
ResNet	Residual Network
RNN	Recurrent Neural Network
TN	True Negative
TP	True Posotive
VGG	Visual Geometry Group

1 Einleitung

Deep Learning, ein Teilbereich des maschinellen Lernens, hat die Analyse komplexer medizinischer Bilder stark verbessert. Besonders Convolutional Neural Networks (CNNs) erkennen, segmentieren und klassifizieren Anomalien automatisch und unterstützen so die Diagnose von Krankheiten wie Krebs, neurologischen Störungen und Herz-Kreislauf Erkrankungen. Für die binäre Klassifikation von Hirn-MRT-Bildern (Tumor vs. kein Tumor) haben CNNs bereits überzeugende Ergebnisse gezeigt. [1].

Faltungsnetze (CNNs) können durch sogenannte adversariale Angriffe leicht getäuscht werden. Dabei werden zu einem Bild sehr kleine Störungen hinzugefügt, die für das menschliche Auge unsichtbar sind. Im Modell führen diese Veränderungen jedoch zu falschen Vorhersagen. Beispielsweise kann eine kleine Änderung in einem MRT-Bild einen Fehler in den Vorhersagen des Modells verusrsachen, statt einen Hirntumor zu erkennen, stuft das Modell die Aufnahme als unauffällig ein, was potenziell lebensbedrohliche Folgen mit sich bringen kann.

Adversarielle Angriffe auf KI-Systeme lassen sich je nach dem Wissensstand des Angreifers in White-Box und Black-Box Angriffe unterscheiden. Bei einem White-Box Angriff kennt der Angreifer das Modell vollständig einschließlich aller Parameter und Hyperparameter und hat somit uneingeschränkten Zugriff auf dessen innere Struktur. Im Gegensatz dazu fehlt ihm bei einem Black-Box Angriff dieser Einblick. Der Angreifer kann lediglich über Eingaben und Ausgaben mit dem Modell interagieren, ohne dessen interne Funktionsweise zu kennen. Selbst überaus gut trainierte Deep-Learning-Modelle können dadurch in die Irre geführt werden. White-Box-Angriffe gehören zu den stärksten Angriffen und werden daher in der Forschung besonders häufig untersucht. So wird deutlich, wie relevant die Entwicklung robuster Modelle und Schutzmechanismen, die auch in solchen Situationen zuverlässig arbeiten, ist.

Diese Bachelorarbeit verfolgt zwei Hauptziele, das erste Ziel ist die Entwicklung und Bewertung von Modellen zur binären Klassifikation von Hirn-MRT-Bildern und das zweite die Untersuchung ihrer Verwundbarkeit gegenüber adversarialen Angriffen sowie die Prüfung von Verteidigungsstrategien. Als Modelle werden ein selbst entworfenes CNN und Transfer Learning mit VGG-16 eingesetzt. Grundlage ist ein öffentlich verfügbarer Kaggle-Datensatz mit MRT-Bildern mit und ohne Tumor. Die Leistungsfähigkeit des Modells wird anschließend mithilfe standardisierter Metriken sowie durch grafische Darstellungen der Ergebnisse bewertet und analysiert.

Zur Analyse der Robustheit werden drei White-Box Angriffe (FGSM, BIM, PGD) sowie Black-Box Angriffe (decision- und score-based) eingesetzt und miteinander verglichen. Betrachtet werden dabei die Störstärke (ε), die Anzahl der Iterationen, die Abfragezahl und die Sichtbarkeit der Artefakte im Bild. Als Gegenmaßnahmen prüft die Arbeit adversariales Training (vor allem gegen White-Box Angriffe) und ein Bild Vorverarbeitung wie den Median-Filter (gegen Black-Box Angriffe). Ziel ist ein ausgewogener Kompromiss zwischen Schutz, Leistung und Rechenaufwand. Dafür werden saubere Genauogkeit und robuste Genauigkeit unter Angriff systematisch gegenübergestellt.

Die Bachelorarbeit ist in mehrere Kapitel gegliedert, die aufeinander aufbauen. Zuerst werden die benötigten Grundlagen zu Deep Learning (insbesondere CNNs und Transfer Learning) bereitgestellt und kurz weitere Konzepte eingeordnet. Danach folgt die Analyse zentraler Sicherheitsrisiken in KI sowie darauf abgestimmter Verteidigungen. Anschließend wird der Stand der Forschung skizziert, bevor Versuchsaufbau und Implementierung der Angriffe beschrieben werden. Die folgenden Abschnitte präsentieren Datensatz, Modelle, Training, Angriffe und robuste Varianten sowie die Ergebnisse. Den Abschluss bilden Diskussion, Fazit und Ausblick.

2 Theoretische Grundlagen

2.1 Deep Learning

Durch maschinelles Lernen können Algorithmen selbstständig aus Daten lernen. So kann ein Modell entwickelt werden, das eine bestimmte Aufgabe erfüllt, ohne dass jeder Schritt explizit programmiert werden muss. Auf diese Weise können Modelle des maschinellen Lernens eigenständig Vorhersagen oder Entscheidungen zu treffen.

Zu den wichtigsten Methoden des maschinellen Lernens gehören das überwachte, das unüberwachte und das bestärkende Lernen. Deep Learning (DL) stellt einen spezialisierten Bereich innerhalb des maschinellen Lernens dar. Basierend auf künstlichen neuronalen Netzen mit zahlreichen Schichten können insbesondere zur Analyse großer und komplexer Datensätze eingesetzt werden. Im Unterschied zu klassischen Verfahren des maschinellen Lernens, die häufig eine manuelle Extraktion relevanter Merkmale erfordern, sind Deep-Learning-Modelle in der Lage, Merkmalsrepräsentationen direkt aus den Rohdaten zu erlernen. So wird unter anderem eine besonders effektive Bewältigung komplexer Aufgaben wie der Bild- und Spracherkennung sowie der Verarbeitung natürlicher Sprache möglich. [1]

Aufgrund der Erfolge der Faltungsnetze, wächst das Interesse am Deep Learning im Bereich der medizinischen Bildgebung. Durch diesen werden Muster und Merkmale in Bildern erkannt. Dadruch wird bestimmt, welche Objekte wahrgenommen werden. Genau auf diesem Prinzip basieren die Faltungsnetze. [2]

Im Folgenden wird einen Überblick über häufig eingesetzte Deep-Learning-Modelle in der medizinischen Bildanalyse gegeben. Dabei wird grundsätzlich zwischen überwachtem und unüberwachten Lernen unterschieden. Beim überwachten Lernen wird ein Modell mit beschrifteten Daten trainiert. Jede Eingabe ist ein bekanntes Ziel (Label) zugeordnet, sodass das Modell die Abbildung von Eingaben auf Ausgaben erlernt. Beim unüberwachten Lernen stehen keine Labels zur Verfügung; das Modell identifiziert selbstständig Strukturen wie Muster oder Cluster, die als Grundlage für weitere Analysen und Vorhersagen dienen können.
Im medizinischen Bereich spielen sowohl überwachte als auch unüberwachte DL-Modelle eine entscheidende Rolle bei diversen Aufgaben wie Klassifizierung, Segmentierung und Anomalieerkennung, wobei jeder Ansatz je nach Art der Daten und dem zu lösenden klinischen Problem unterschiedliche Vorteile bietet.

2.2 Supervised Learning

Supersived Learning oder auf Deutsch Überwachtes Lernen beschreibt ein Verfahren des maschinellen Lernens, im Zuge dessen ein Modell anhand von beschrifteten Daten trainiert wird. Zu jeder Eingabe (z. B. ein Bild) gibt es somit die passende Ausgabe (z. B. Tumor oder kein Tumor). Ziel ist es, eine Funktion zu entwickeln, die in der Lage ist, neue und bisher unbekannte Daten zuverlässig vorherzusagen. Diese Methode ist in medizinischen Anwendungen, bei denen annotierte Daten vorliegen, besonders wirkungsvoll. [4]

2.2.1 Convolutional neural Network

Convolutional Neural Networks (CNNs) oder Faltungsnetze auf Deutsch sind moderne Methoden zur Bildverarbeitung. Sie können Bilddaten so verkleinern, dass weniger Informationen

Abbildung 2.1: DL Basis-Kategorien [3]

verarbeitet werden müssen, ohne dass wichtige Merkmale verloren gehen. Dadurch müssen weniger Parameter gelernt werden, was die Recheneffizienz erheblich steigert.

Ein Zentraler Vorteil von CNNs ist ihre Vielseitigkeit. Sie können sowohl zweidimensionale als auch dreidimensionale Bilder verarbeiten. Das macht sie besonders nützlich in der Medizin zum Beispiel bei Röntgenbildern (2D) oder bei MRT- und CT-Scans (3D). [3]
CNNs sind besonders dazu geeignet, aus markierten Daten zu lernen. Sie eignen sich daher sehr gut für überwachte Lernaufgaben, vor allem in der Medizin, wo genaue Bildanalysen wichtig sind. Im Folgenden werden die einzelnen Bausteine von CNNs genauer erklärt und verdeutlicht, wie sie medizinische Bilder verarbeiten, daraus lernen und diese interpretieren.

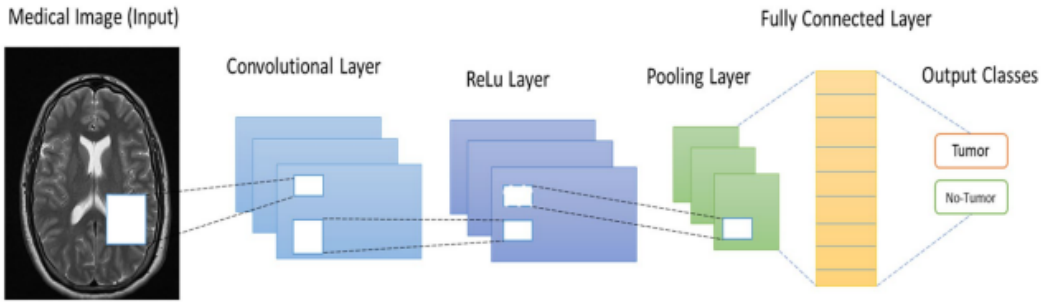

Abbildung 2.2: Allgemeine CNN-Architektur [3]

Auf der Grundlage von Abbildung 2.2 wird die Grundstruktur von CNNs wie folgt definiert:

- **Input layer:** oder Eingabeschicht auf Deutsch verarbeitet medizinische Bilddaten, meist als Graustufen oder RGB-Bilder mit einer einheitlichen Auflösung von 128x128, 224×224 oder 256×256 Pixeln.

- **Convolutional layer:** Faltungsschichten auf Deutsch wurden verwendet, um spezifische Merkmale wie Ecken, Kanten, und Rauschen aus Bildern zu extrahieren. Dies wird möglich, wenn ein Filter oder Kernel auf das Bild angewendet wird, der sich in einer gleitenden Fenstertechnik über das Bild bewegt, bis es vollständig abgedeckt ist.

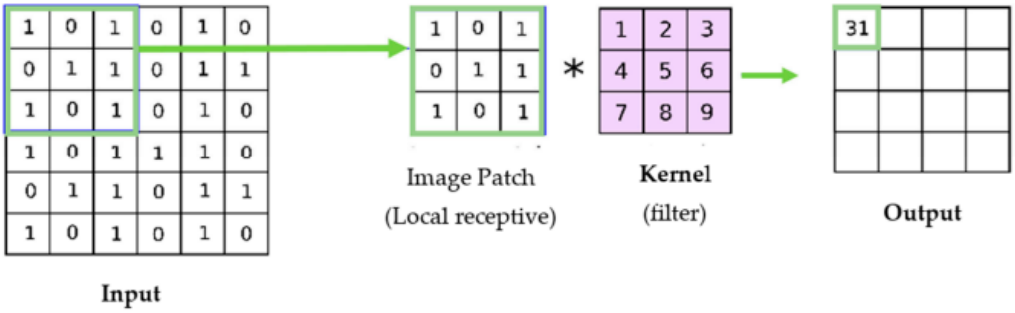

Abbildung 2.3: Faltungsprozess [5]

Gemäß Gleichung(2.1) wird der Kernel mit h und das Eingabebild mit f bezeichnet. Die Zeilen- und Spaltenindizes der Ergebnismatrix werden mit m und n bezeichnet und $\sum_j \sum_k$ die Summe über alle Pixel des Filters

$$(f\,x)[m,n] = (h * f)(m,n) = \sum_j \sum_k h[j,k] f[j+m, k+n] \tag{2.1}$$

- **Pooling layer:** Die Pooling-Schicht verkleinert die Größe der Feature-Maps aus der vorherigen Schicht. Dadurch entstehen neue Feature-Maps mit geringerer Auflösung. Das bringt diverse Vorteile mit sich. Es liegen weniger Parameter vor, die Berechnungen werden schneller, und das Risiko von Overfitting sinkt. Außerdem ermöglicht Pooling nur die wichtigen Informationen erhalten bleiben, während unwichtige Details herausgefiltert werden.

  Medizinische Bilder wie MRT- oder CT-Aufnahmen zeigen viele feine Details. Schon kleine Veränderungen wie Drehung, Verschiebung oder Vergrößerung können das Bild stark verändern. Die Pooling-Schicht hilft, die wichtigsten Merkmale zu behalten und unwichtige Unterschiede zu ignorieren. Dadurch wird das Modell stabiler und weniger anfällig für solche Veränderungen in den medizinischen Bildern.

  Basierend auf Abbildungen 2.4 und 2.5 gibt es zwei Arten von Pooling:

  – Max-Pooling: Dabei wird pro Patch der maximale Wert übernommen. So lassen sich dominante Strukturen identifizieren, beispielsweise besonders helle Tumorregionen im Gehirn.

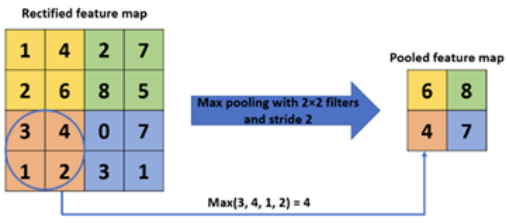

Abbildung 2.4: Max-Pooling [6]

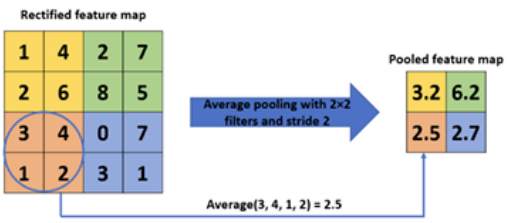

Abbildung 2.5: Mittelwert-Pooling [6]

  - Average Pooling: oder Mittelwert-Pooling auf Deutsch ermittelt pro Patch den Mittelwert und betont so die Gesamtintensität eines Bereichs, statt einzelne besonders helle Pixel hervorzuheben.

- **Fully Connected layer (Dense):** (Deutsch: Vollständig verbundene Schicht) ist die letzte Schicht in einem CNN. Sie verarbeitet die zuvor extrahierten und gefilterten Merkmale und entscheidet schließlich, zu welcher Klasse ein Bild gehört.

  Im medizinischen Bereich kann diese Schicht beispielsweise dazu genutzt werden, um festzustellen, ob ein MRT-Bild einen Tumor zeigt oder wie stark eine Lungeninfektion in einem Röntgenbild ausgeprägt ist.

- **Activation function:** (Deutsch: Aktivierungsfunktion) Auf die vollständig verbundenen Schichten und Faltungsschichten folgen meist Aktivierungsfunktionen. Anhand dieser wird das Modell dazu befähigt selbst komplexe Zusammenhänge zu lernen und nicht nur einfache lineare Entscheidungen zu treffen. Zwei häufig verwendete Aktivierungen sind dabei:

  - ReLU (Rectified Linear Unit): Nach jeder Berechnung in einer Schicht setzt ReLU alle negativen Werte auf 0 und lässt positive Werte unverändert. Dadurch kann das Netzwerk auch komplexe Muster lernen.
  - Softmax: Diese Funktion wird oft in der letzten Schicht bei Klassifikationsaufgaben verwendet. Sie wandelt die Ausgaben in Wahrscheinlichkeiten um, die zusammen eins ergeben. So kann das Modell entscheiden, zu welcher Klasse ein Bild am wahrscheinlichsten gehört.

- **Output layer:** (Deutsch: Ausgabeschicht) In dieser Schicht wird die Klassenbezeichnung vorhergesagt. Bei mehreren Klassen wird dafür die Softmax-Funktion verwendet, bei zwei Klassen die Sigmoid-Funktion.

Wichtige Parameter und Hyperparameter für den Aufbau von CNNs :
Der Aufbau eines qualitativ hochwertigen CNN-Modells erfordert die sorgfältige Einstellung verschiedener Parameter und Hyperparameter. Diese bestimmen, wie das Netzwerk Eingaben verarbeitet und wichtige Muster erkennt. Insbesondere in sensiblen Bereichen wie der medizinischen Bildanalyse ist das von großer Bedeutung. [7]
Die wichtigsten Hyperparameter und ihre Funktionen werden im Folgenden erläutert:

- Kernels (Filters): Ein Kernel ist eine kleine Matrix, die über das Eingabebild gleitet, um durch eine Faltungsoperation lokale Merkmale zu extrahieren. Während dieses Prozesses führt der Kernel eine elementweise Multiplikation (Skalarprodukt) mit jedem Teil des Bildes durch, den er abdeckt. So entstehen neue Werte, die anzeigen, ob bestimmte Muster im Bild vorhanden sind, wie zum Beispiel Kanten oder Linien.

  In der medizinischen Bildgebung können Kernel Tumorgrenzen, Organformen oder Texturanomalien erkennen. Die Bewegung des Kernels wird durch den Stride-Wert gesteuert. [8]

- Bias: (Deutsch: Verzerrung) ist ein trainierbarer Parameter, der vor der Aktivierungsfunktion zum Ausgang eines Neurons addiert wird. Dadurch kann das Netzwerk die Aktivierungsschwelle verschieben und sich besser an die Datenverteilung anpassen. Mathe-

matisch gesehen wird der Output, wenn ein Neuron den Input x erhält und das Gewicht w hat, zu (w*x + b),erst danach wird die Aktivierungsfunktion angewendet.

- Padding: Beim Padding werden vor dem Faltungsprozess zusätzliche Zeilen und Spalten, in der Regel Nullen, um die Ränder des Eingabebildes hinzugefügt. [9]

  Ohne Padding könnten die Filter die Bildränder nicht vollständig berücksichtigen, und das Ergebnisbild würde kleiner werden. Mit Padding bleibt die ursprüngliche Größe des Bildes erhalten, und auch die Randbereiche fließen in die Verarbeitung ein.

- Stride: (Deutsch: Schrittweite) gibt an, um wie viele Pixel sich der Kernel beim Durchlauf über das Bild bewegt. Eine Schrittweite von 1 bedeutet, der Filter wird Pixel für Pixel verschoben. Eine Schrittweite von 2 bedeutet, der Filter springt immer zwei Pixel weiter, wodurch die Ausgabe kleiner wird. [10]

  In medizinischen CNNs wählt man oft kleinere Schrittweiten, um die Auflösung zu erhalten und sehr feine Strukturen zu erkennen.

- Dropout: Dropout ist eine Methode gegen Overfitting in neuronalen Netzen. Dabei werden während des Trainings zufällig einige Neuronen vorübergehend ausgeschaltet. So entstehen viele kleine Teilnetzwerke. Beim Testen wird das ganze Netz wieder genutzt und das Netzwerk kombiniert das Wissen dieser vielen kleinen Netze. was die Generalisierungsfähigkeit des Modells verbessert. [11]

  Bei der Klassifizierung medizinischer Bilder verbessert Dropout die Generalisierung und verringert das Risiko von Fehlalarmen oder zu sichere, aber falsche Vorhersagen.

- Learn rate μ: (Deutsch: Lernrate) Lernrate bestimmt, wie stark die Gewichte eines Modells nach jedem Backpropagation-Schritt angepasst werden. Eine kleine Lernrate sorgt dafür, dass das Training langsam, aber stabil abläuft. Eine große Lernrate macht das Training schneller, kann aber dazu führen, dass das Modell keine gute Lösung findet. Häufig werden die Lernraten dynamisch angepasst, mit Techniken wie adaptiven Optimierern. So erreicht man ein gutes Gleichgewicht zwischen Geschwindigkeit und Genauigkeit, was besonders bei sensiblen Aufgaben wie der Krankheitsdiagnose wichtig ist. [12]

Anwendungen im medizinischen Bereich:

- Organ- und Läsionssegmentierung (in CT-/MRT-Bildern). [13]

- Tumorerkennung (bei MRT-Untersuchungen des Gehirns). [14]

2.2.2 Transfer Learning

Beim Transferlernen nutzt man ein bereits vortrainiertes Deep-Learning-Modell beispielsweise ResNet und VGG16, das auf vielen Daten gelernt hat, und passt es an eine neue Aufgabe mit wenig Daten an. Die unteren Netzwerkschichten erfassen allgemeine Muster (Kanten, Texturen, Formen, bei Texten auch Syntax und Bedeutung), die für viele Aufgaben nützlich sind und deshalb wiederverwendet werden können.
Mit Transferlernen braucht man weniger manuell beschriftete Bilder. Das Modell konvergiert schneller und wird genauer, vor allem bei seltenen Krankheiten für die nur wenige beschriftete Proben verfügbar sind. [15]
Anwendungen im medizinischen Bereich:

- COVID-19 Erkennung mittels Röntgenaufnahmen des Brustkorbs. [15]

- Erkennung der diabetischen Retinopathie in Fundusbildern [16]

2.2.3 Recurrent Neural Networks(RNN)

Rekurrente Neuronale Netze (RNN) sind Modelle für Daten, die eine Reihenfolge haben, z. B. Wörter in einem Satz. Durch Rückkopplungen können sie sich Informationen aus früheren Schritten merken. Dieses Gedächtnis steckt in sogenannten verborgenen Zuständen, die zusammenfassen, was das Netz bisher gesehen hat.
Normale RNNs haben beim Lernen oft das Problem, dass die Gradienten sehr klein oder sehr groß werden (Verschwinden/Explodieren). Dann lassen sich die Gewichte schlecht anpassen, und weit zurückliegende Informationen gehen verloren. Deshalb gibt es Varianten wie LSTM und GRU. Sie nutzen Gatter, die den Informationsfluss regeln und die Gradienten stabil halten, sodass auch lange Abhängigkeiten gelernt werden können.
RNNs haben den Nachteil, dass sie eine Sequenz Schritt für Schritt verarbeiten. Das kostet viel Rechenzeit besonders bei langen Sequenzen und großen Datensätzen. [17]
Anwendungen im medizinischen Bereich:

- Analyse von Elektrokardiogramm-Signalen zur Erkennung von Herzanomalien [18]

- Automatisierte Erstellung medizinischer Berichte anhand von Röntgenaufnahmen des Brustkorbs [19]

- Überwachung von Intensivpatienten über einen längeren Zeitraum (z.B. Vorhersage einer Sepsis). [20]

Tabelle 2.1: Zusammenfassung der Typen des überwachten Lernens

Modell	**Typische Daten**	**Typische Anwendungen**	**Vorteile**	**Nachteile**
CNN	medizinische Bilder	Klassifikation	gute Bildleistung	Schwach beim langen Kontext
Transfer Learning	medizinische Bilder	Klassifikation	Kleine medizinische Datensätze	Abhängigkeit vom Quellmodell
RNN/LSTM	sequentielle Daten	Vorhersage	Zeitreihen oder Erstellung klinischer Berichte	viel Rechenzeit

2.3 Unsupervised Learning

(Deutsch: Unüberwachtes Lernen) Ein Algorithmus sucht ohne vorgegebene Labels selbstständig Muster und Strukturen in Daten. Das ist besonders nützlich in der medizinischen Bildgebung, wo gelabelte Datensätze oft knapp oder teuer sind. [21]
In diesem Abschnitt werden drei wichtige unüberwachte Ansätze vorgestellt: Autoencoder, Generative Adversarial Networks und Restricted Boltzmann Machines

2.3.1 Autoencoders (AEs)

Autoencoder (AEs) sind unüberwachte neuronale Netzwerke, die entwickelt wurden, um Eingabedaten zu komprimieren und anschließend möglichst genau wiederherzustellen.
Sie bestehen aus einem Encoder, der die Daten in eine kompakte latente Darstellung (Code) komprimiert, und einem Decoder, der aus diesem Code die Daten wieder rekonstruiert. In der Dekodierungsphase (Decoder) werden diese kompakten Repräsentationen wieder in die ursprüngliche Form der Eingabedaten zurückgeführt.
Autoencoder nutzen Faltungsschichten (Convolutional Layers) und Pooling-Methoden, um lokale Merkmale wie Kanten oder Muster zu erfassen und gleichzeitig die Datenmenge zu verringern.
Beim Training wird eine Rekonstruktionsfunktion (z. B. mittlere quadratische Abweichung oder Kreuzentropie) minimiert, Dabei lernt der Autoencoder, wichtige Merkmale zu erkennen und Störrauschen zu verwerfen.
Die Leistungsfähigkeit eines Autoencoders zeigt sich daran, wie präzise die rekonstruierten Daten dem Original ähneln. [22]

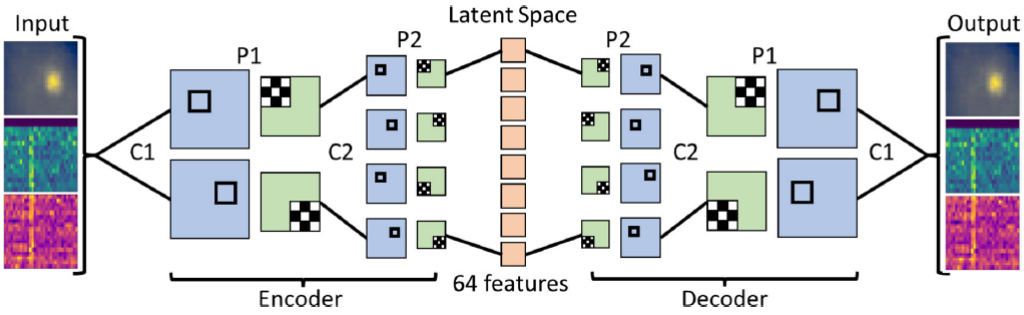

Abbildung 2.6: AE Struktur [22]

Wie in Abbildung 2.6 gezeigt, setzt sich der Autoencoder aus einem Encoder zusammen, der die Eingabedaten in eine niedrigdimensionale latente Repräsentation komprimiert, sowie einem Decoder, der die ursprünglichen Daten aus dieser codierten Darstellung wiederherstellt.
Anwendungen im medizinischen Bereich:

- Perfekt geeignet zur Reduzierung von Daten dimensionen und zur unüberwachten Extraktion relevanter Merkmale und bei der Erkennung von Anomalien.

2.3.2 Generative Adversarial Networks (GANs)

(Deutsch: Generative gegnerische Netzwerke) Das Funktionsprinzip von GANs in der Künstlichen Intelligenz basiert auf einem Wettbewerb zwischen zwei Komponenten. Der Generator erstellt künstliche Bilder oder Daten, die den echten Daten stark ähneln, während der Diskriminator dafür zuständig ist, zwischen echten und künstlichen Daten zu unterscheiden. Der Diskriminator wird darauf trainiert, eine binäre Klassifikation vorzunehmen („echt“ oder „falsch“), während der Generator darauf trainiert wird, den Diskriminator zu täuschen, indem er immer realistischere Daten erzeugt. [23]

Anwendungen im medizinischen Bereich:

- Wird eingesetzt, um künstliche medizinische Bilder zu erzeugen, Datensätze zu vergrößern und die Genauigkeit bei Segmentierungsaufgaben zu steigern.

2.3.3 Restricted Boltzmann Machines

Eine Restricted Boltzmann Machine (RBM) hat zwei Schichten von Knoten: eine sichtbare und eine versteckte. Knoten aus der sichtbaren Schicht können mit Knoten aus der versteckten Schicht verbunden sein aber nie Knoten innerhalb derselben Schicht miteinander.

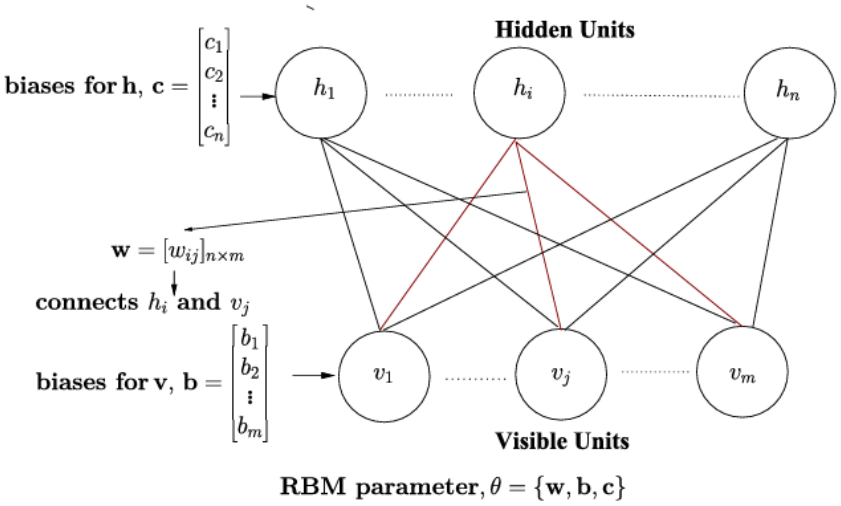

Abbildung 2.7: Restricted Boltzmann Machine [24]

RBMs sind vor allem generative Modelle. Sie kann die Eingaben automatisch ordnen und passt dabei ständig ihre Gewichte und Biases an. Auf diese Weise erkennt die RBM wichtige Merkmale selbstständig und ist in der Lage, die ursprünglichen Daten möglichst genau wiederherzustellen. Sie lernen die Struktur von Daten und können neue, ähnliche Beispiele erzeugen. Sie haben eine wichtige Rolle bei der Entwicklung moderner neuronaler Netze und des Deep Learnings gespielt. Eine RBM braucht keine gelabelten Daten, was besonders nützlich für echte Datensätze wie Bilder, Videos oder Audiodateien ist. [24]
Anwendungen im medizinischen Bereich: [24]

- Analyse der Konnektivitätsstruktur des Gehirns anhand von MRI-Bildern.

- Erstellung von Themenmodellen aus unstrukturierten Textdaten

Tabelle 2.2: Zusammenfassung der Modelle für unüberwachtes Lernen in der medizinischen Bildgebung.

Modell	**Typische Daten**	**Typische Anwendungen**
Autoencoder (AE)	Unbeschriftete Bilddaten	Denoising, Kompression
Generative Adversarial Networks (GAN)	hochdimensionale Medien	Hochrealistische synthetische Bilder
Restricted Boltzmann Machines (RBM)	Binäre oder normalisierte Daten	Modellierung einfacher Strukturen

3 Sicherheitsrisiken in der Radiologie

Obwohl Deep Learning die Diagnosemöglichkeiten in der Radiologie durch die schnelle und genaue Interpretation medizinischer Bilder revolutioniert hat, bringt es auch eine Reihe neuer kritischer Risiken mit sich. Diese Risiken können die Patientensicherheit gefährden, gegen Datenschutzgesetze verstoßen und das Vertrauen der Ärzte in die Diagnose auf Basis von KI-Tools mindern. In diesem Abschnitt werden daher die wichtigsten Schwachstellen im Zusammenhang mit DL-Modellen im Bereich der Radiologie aufgezeigt, wobei der Schwerpunkt auf adversarialen Angriffen, Datenschutzverletzungen und Bedrohungen der Datenintegrität liegt.

3.1 Adversariale Angriffe

Adversariale Angriffe sind ein wachsendes Problem beim Einsatz von Deep Learning in der medizinischen Bildgebung, besonders in der Radiologie. Dabei werden die Eingabedaten absichtlich so verändert, dass das Modell falsche Ergebnisse liefert. Das kann durch sehr kleine, gezielte Änderungen an den Originalbildern passieren oder durch die Erstellung realistischer gefälschter Bilder um Deep-Learning-Algorithmen zu täuschen. Obwohl diese Veränderungen für das menschliche Auge oft unsichtbar sind, nutzen sie gezielt Schwächen im Modell aus, um Klassifizierungs- oder Diagnosefehler zu verursachen, was in der Medizin besonders kritisch ist. [25]

Adversariale Angriffe werden üblicherweise anhand ihrer Absicht, eine falsche Klassifizierung zu erzwingen, in zwei Kategorien eingeteilt:

- **Untargeted Attacks:** (Deutsch: Ungezielte Angriffe) zielen darauf ab, dass das Modell irgendeine falsche Kategorie wählt, nicht eine bestimmte. So kann zum Beispiel ein normales Hirnbild fälschlich als Hirntumor eingestuft werden oder ein Bild mit Tumor als unauffällig. Das kann zu ungenauen und ungeeigneten medizinischen Entscheidungen führen, etwa zu unnötigen Untersuchungen oder Behandlungen.

- **Targeted Attacks:** (Deutsch: gezielte Angriffe) Bei einem gezielten Angriff zielt der Angreifer darauf ab, das Modell gezielt zu einer bestimmten falschen Vorhersage zu bringen.

Angreifertechniken werden auch auf der Grundlage des Wissens des Angreifers über das Ziel-DL-Modell unterteilt:

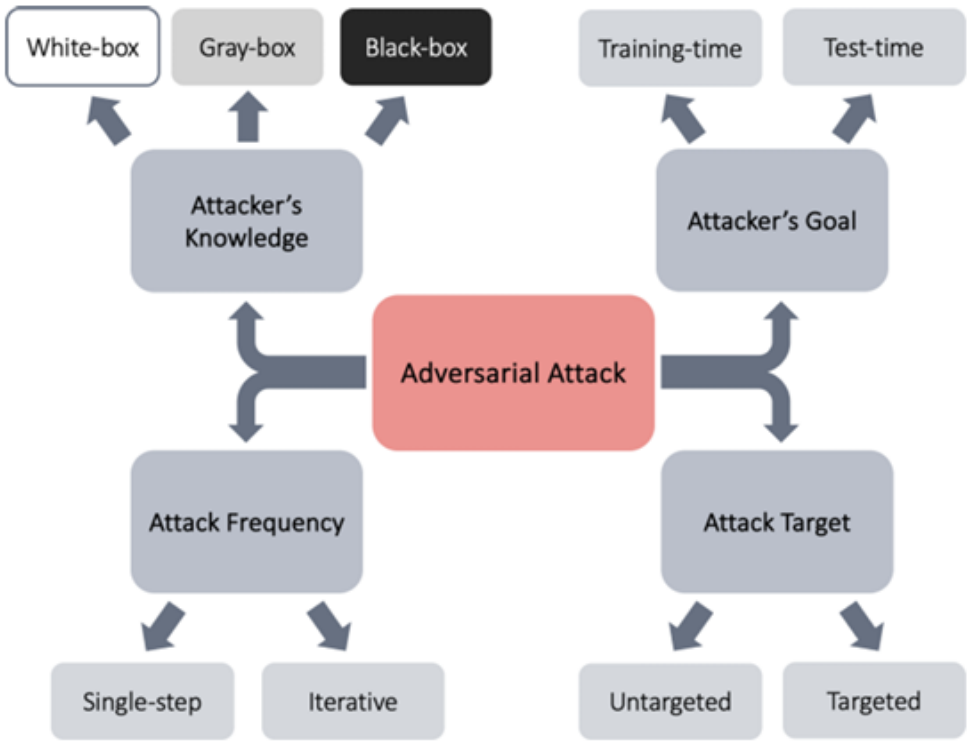

Abbildung 3.1: Taxonomie adversarieller Angriffe in KI-Systemen [25]

- **White-Box-Attacks:** White-Box-Angriffe sind Angriffe auf KI-Modelle, bei denen der Angreifer das System sehr genau kennt: den Aufbau (Architektur), die verwendeten Trainingsdaten, die Funktionsweise und auch die Schutzmaßnahmen. Mit diesem Wissen kann er berechnen, wie sich das Ergebnis ändert, wenn man die Eingabe ganz leicht verändert. So fügt er ein kleines Störsignal hinzu, das für Menschen kaum sichtbar ist, das Modell aber in die Irre führt. Dadurch entstehen falsche Klassifizierungen oder andere unerwünschte Ergebnisse, zum Beispiel wird ein Bild falsch erkannt. [27]

  Zu den am häufigsten verwendeten Angriffen zählen gradientenbasierte Verfahren, bei denen durch gezielte, minimale Perturbationen Fehlklassifikationen hervorgerufen werden:

  – Die Fast Gradient Sign Method (FGSM) ist eine bekannte Methode, um gezielte Störungen in Eingabedaten zu machen. So entstehen sogenannte adversariale Beispiele. Sie soll tiefe neuronale Netze, vor allem CNNs, bewusst in die Irre führen.

    Dabei wird die Eingabe nur ganz wenig verändert, aber in die Richtung, die den Fehler des Modells größer macht. Dafür werden zuerst den Gradienten der Verlustfunktion zur Eingabe berechnet . Danach wird die Eingabe minimal angepasst, damit der Gesamtverlust steigt und das Modell falsch entscheidet.

    Wie stark diese kleine Änderung ist, bestimmt der Epsilon-Wert ϵ. Er legt fest, wie groß das Rauschen sein darf. Es soll das Bild für Menschen kaum sichtbar verändern, das Modell jedoch stark beeinflussen. [28]
    Die neue (angreifende) Eingabe wird mathematisch wie folgt berechnet:

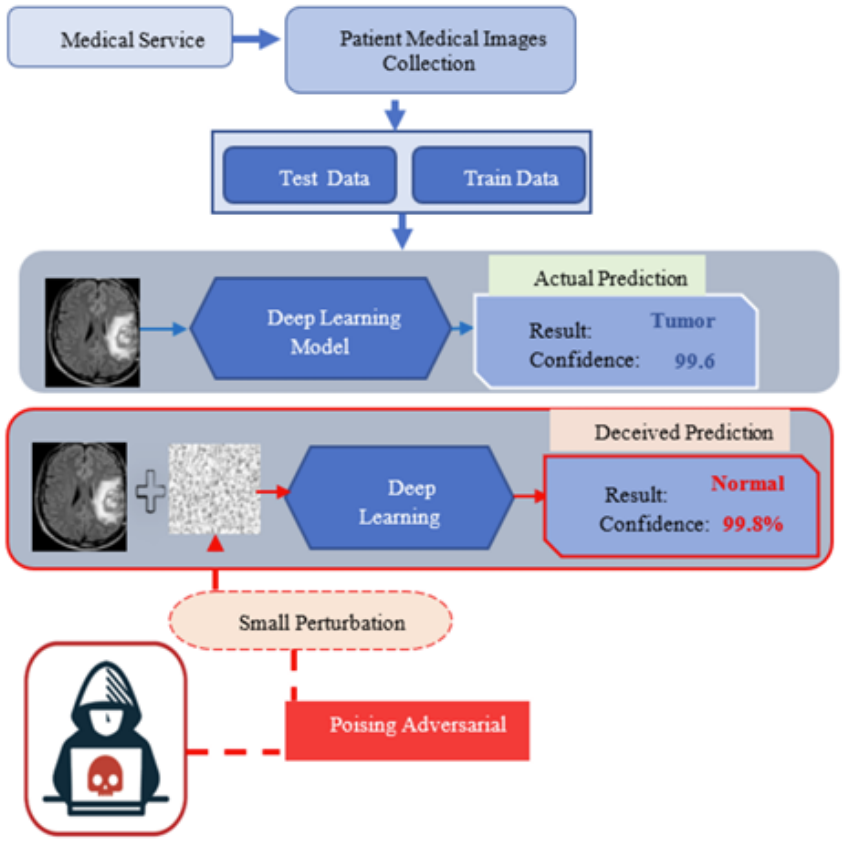

Abbildung 3.2: Beispiel für einen gegnerischen Angriff [26]

$$x' = x + \epsilon \cdot \text{sign}\left(\nabla_x J(\theta, x, y)\right)$$

Dabei gilt:

- x: Die originale Eingabe (z. B. ein MRT-Bild), die in das Modell geht.
- x': Die veränderte Eingabe. Sie sieht fast gleich aus wie x, bringt das Modell aber durcheinander.
- ϵ: Die Stärke der Störung. Je größer ϵ , desto stärker wird x verändert.
- $\nabla_x J(\theta, x, y)$: Gradient der Verlustfunktion J in Bezug auf die Eingabe x.
- θ: Die Parameter des Modells (das, was das Modell gelernt hat).
- y: Das wahre Label der Eingabe (die richtige Antwort).

FGSM fügt der Eingabe gezielt eine kleine Störung hinzu. Dadurch wird die Eingabe in einen Bereich verschoben, in dem das Modell wahrscheinlicher falsch entscheidet. Trotzdem bleibt die Eingabe dem Original sehr ähnlich.

– Basic Iterative Method (BIM): Die Basic Iterative Method (BIM) ist eine Weiterentwicklung von FGSM. Ziel ist es, stärkere und genauere Angriffe zu erzeugen. Statt die Eingabe einmal mit dem Gradienten zu verändern, macht BIM viele kleine Schritte hintereinander. Nach jedem Schritt wird die veränderte Eingabe zurückprojiziert, damit die Gesamtstörung nicht zu groß wird. Sie bleibt in einem festen

Bereich um die Original Eingabe (oft wird das eine ϵ-Kugel genannt, d.h. die Veränderung darf höchstens ϵ betragen).

Weil BIM in mehreren Schritten arbeitet, kann es den Verlust des Modells nach und nach erhöhen. Dabei bleibt die Störung immer innerhalb eines erlaubten Bereichs. Darum ist BIM im Vergleich zu FGSM oft wirksamer und genauer beim Erzeugen adversarieller Beispiele.

Der Prozess wird mit der originalen Eingabe x begonnen. Optional wird mit einer leicht veränderten Version $x0$ gestartet. Die Aktualisierung pro Schritt wird wie folgt berechnet:

$$x^{(i+1)} = \text{Clip}_{x,\epsilon}\left\{x^{(i)} + \alpha \cdot \text{sign}\left(\nabla_x J(\theta, x^{(i)}, y)\right)\right\}$$

Dabei gilt:

* $x^{(i+1)}$: Dies ist die gestörte Eingabe beim nächsten Schritt der Iteration.
* $x^{(i)}$: Zwischenstand nach i Iterationen
* α: Schrittweite jeder Iteration
* ϵ: Maximal erlaubte Gesamtstörung
* $\nabla_x J(\theta, x^{(i)}, y)$: Gradient der Verlustfunktion J in Bezug auf die Eingabe $x^{(i)}$
* $\text{Clip}_{x,\epsilon}$: Funktion, die sicherstellt, dass $x^{(i+1)}$ innerhalb eines ϵ-Umfelds von x bleibt

BIM ändert die Eingabe in vielen kleinen, genau gesteuerten Schritten. Nach jedem Schritt wird geprüft, dass die Änderung im erlaubten Bereich bleibt. So entstehen täuschende Beispiele (adversariale Beispiele), die das Modell falsch erkennt, obwohl sie für Menschen fast gleich wie das Original aussehen.

– Projected Gradient Descent (PGD): Der PGD-Angriff ist eine starke, wiederholte Methode, um adversariale Beispiele zu bauen, die Deep-Learning-Modelle täuschen. Er baut auf FGSM auf. Statt nur einem großen Schritt macht PGD viele kleine Schritte mit dem Gradienten und passt die Eingabe jedes Mal ein bisschen an. Dadurch wird der Angriff genauer und robuster. Bei PGD wird zuerst zufällig ein Startpunkt in der ϵ-Umgebung des Originalbildes gewählt. Die Störung wird dann schrittweise berechnet ähnlich wie bei BIM und nach jedem Schritt in den erlaubten Bereich zurückprojiziert.

Bei PGD werden oft mehrere zufällige Neustarts verwendet (es wird also mehrfach von verschiedenen Punkten begonnen). Bei BIM wird normalerweise am gleichen Punkt begonnen. Darum wird PGD meist als stärker und robuster als BIM angesehen. [29]

Die Erzeugung der adversarialen Beispiele erfolgt nach einer iterativen Aktualisierungsregel:

$$x_{t+1} = \text{Proj}_{S_\epsilon(x)}\left(x_t + \alpha \cdot \text{sign}\left(\nabla_x J(\theta, x_t, y)\right)\right)$$

* x_{t+1}: Die aktualisierte Eingabe nach der t-ten Iteration.
* x_t: Zwischenstand nach $i - ten$ Iterationen.

  - α: Die Schrittweite ist eine kleine positive Zahl. Sie bestimmt, wie groß jede Änderung (Störung) in jedem Schritt des Gradienten-Aufstiegs ist.
  - sign(·): Die Vorzeichenfunktion zeigt die Richtung des Gradienten an. Sie macht aus dem Gradienten einen einfachen Vektor nur mit +1 oder -1. Dieser Vektor sagt, in welche Richtung die Störung gehen soll.
  - $\nabla_x J(\theta, x_t, y)$: Der Gradient der Verlustfunktion J bezüglich der Eingabe x_t. Dieser Gradient zeigt an, wie die Eingabe angepasst werden muss, um die Verlustfunktion zu erhöhen, was das Modell dazu bringt, die Eingabe mit höherer Wahrscheinlichkeit falsch zu klassifizieren.
  - $\text{Proj}_S(x, \epsilon)$: Der Projektionsoperator passt die veränderte Eingabe x_{t+1} so an, dass sie nah beim Original x bleibt. Er sorgt dafür, dass x_{t+1} innerhalb der erlaubten ϵ-Umgebung liegt. Das heißt: x_{t+1} wird zurück in den erlaubten Bereich gebracht, damit die maximale Störgröße ϵ nicht überschritten wird.

  Im Gegensatz zu FGSM, das nur einmal eine Störung hinzufügt, verändert PGD das Eingabebild schrittweise und kontrolliert mehrfach. Dadurch entstehen stärkere gegnerische Beispiele. PGD ist deshalb ein wirkungsvollerer und schwierigerer Angriff, da es die Verlustfunktion genauer untersucht und gleichzeitig unauffällig bleibt.

- **Black-Box-Angriffe:** Hier hat der Angreifer keinen direkten Zugriff auf die Modell-Parameter. Stattdessen stellt er dem System Eingaben und sieht sich die Ausgaben an. Aus diesen Antworten versucht er zu verstehen, wie sich das Modell verhält
  Es gibt zwei Arten dieses Angriffs:

  – **Score-based attacks:** Bei scorebasierten Angriffen kann der Angreifer das Modell abfragen. Er bekommt nicht nur das vorhergesagte Label, sondern auch die Vertrauenswerte für alle Klassen. Diese Werte geben ihm laufend Hinweise. So kann er ungefähr herausfinden, in welche Richtung das Modell sich ändert, und kleine, absichtliche Störungen am Eingang machen. [30]

    Zuerst wird das Modell mit einem Bild abgefragt und die Konfidenz-Scores werden gespeichert. Dann wird grob geschätzt, wie sich der Fehler ändert, da exakte Gradienten nicht verfügbar sind. Anschließend wird das Bild schrittweise in die Richtung verändert, die die Wahrscheinlichkeit einer Fehlklassifikation maximiert. Am Ende entsteht ein adversariales Beispiel mit möglichst kleiner Veränderung.

    Der Hauptvorteil dieses Angriffs ist, dass er weniger Abfragen benötigt als entscheidungsbasierte Methoden.

  – **Decision-baesd attacks:** Bei entscheidungsbasierten Angriffen kann der Angreifer das Modell nur abfragen und bekommt nur das Endergebnis, zum Beispiel „Tumor“ oder „Kein Tumor“.

    Dabei wird die Eingabe durch das Hinzufügen von kleinem, zufälligem Rauschen verändert. Das Modell wird abgefragt, um zu prüfen, ob die Vorhersage vom wahren Label abweicht. Bei Erfolg wird die Störung verfeinert, sodass die Verzerrung

verringert und die Fehlklassifikation beibehalten wird. Dieser Ablauf wird wiederholt, bis der Angriff erfolgreich ist oder die maximale Anzahl an Iterationen erreicht wird.

Der Hauptvorteil ist, dass nur mit dem Label-Feedback gearbeitet wird. Die großen Herausforderungen sind meist, dass sehr viele Abfragen nötig sind und die Änderungen am Bild können größer sein als bei scorebasierten Angriffen.

- **Gray-Box-Angriff:** Gray-Box-Angriffe liegen zwischen White- und Black-Box. Der Angreifer hat Teilwissen über das Zielmodell oder seine Umgebung, aber nicht alles. Er kennt zum Beispiel die Architektur, nicht aber die trainierten Gewichte; er weiß etwas über die grobe Verteilung der Trainingsdaten; oder er kennt den Typ der Verteidigung, jedoch nicht deren genaue Parameter.

Angriffe werden danach sortiert, wann sie im KI-Prozess passieren und was sie bewirken sollen in zwei Kategorien:

- Angriffe in der Trainingsphase (Training-time attacks): Bei Angriffen in der Trainingsphase wird das KI-Modell schon beim Lernen geschwächt. Beim Datenvergiften werden absichtlich falsche oder veränderte Beispiele in den Trainingssatz gegeben. Dadurch werden falsche Regeln oder eine geheime Hintertür (Backdoor) gelernt. Später kann durch ein kleines Zeichen eine falsche Entscheidung ausgelöst werden.

  Abbildung 3.3 zeigt einen Backdoor-Angriff. Bei diesem Angriff werden manchen Trainingsbildern kleine, kaum sichtbare Muster hinzugefügt. Beim Lernen wird gemerkt, dass dieses Muster zu einer bestimmten (aber falschen) Klasse gehört. Später reicht das kleine Muster in einem neuen Bild aus. Dann wird eine falsche Klasse gewählt, egal was sonst im Bild ist. So wird die KI absichtlich in die Irre geführt.

  Beim Label Poisoning sind die Beschriftungen in einem Datensatz falsch, unregelmäßig oder nicht klar. In der medizinischen Bildgebung, besonders bei MRT von Hirntumoren, kann das passieren durch Fehler bei der Beschriftung, Bias der Personen und Unterschiede zwischen Radiolog*innen. Auch Technik-Probleme wie Scanner-Fehler oder schlechte Bildqualität spielen eine Rolle. Im Gegensatz zu natürlichen Bildern (z. B. ImageNet) sind die Folgen hier ernst. Ein bösartiger Tumor kann als gutartig markiert werden, oder die Grenzen des Tumors werden nicht genau gezeichnet. Diese Fehler stören das Lernen von tiefen neuronalen Netzen und führen zu schlechteren Ergebnissen. [32]

- Testzeit-Angriffe: Bei Testzeit-Angriffen wird ein fertiges KI-Modell während der Nutzung angegriffen. Die Eingabe wird verändert, nicht das Modell. Aus einem echten Röntgenbild wird mit sehr kleinem, fast unsichtbarem Rauschen ein verändertes Bild gemacht. Für Menschen wird kein Unterschied gesehen. Durch das kleine Signal wird eine falsche Entscheidung ausgelöst.

Man kann diese Angriffe auch danach einteilen, wie die Störung gemacht wird: in einem Schritt oder in mehreren, wiederholten Schritten in zwei Ktegorien:

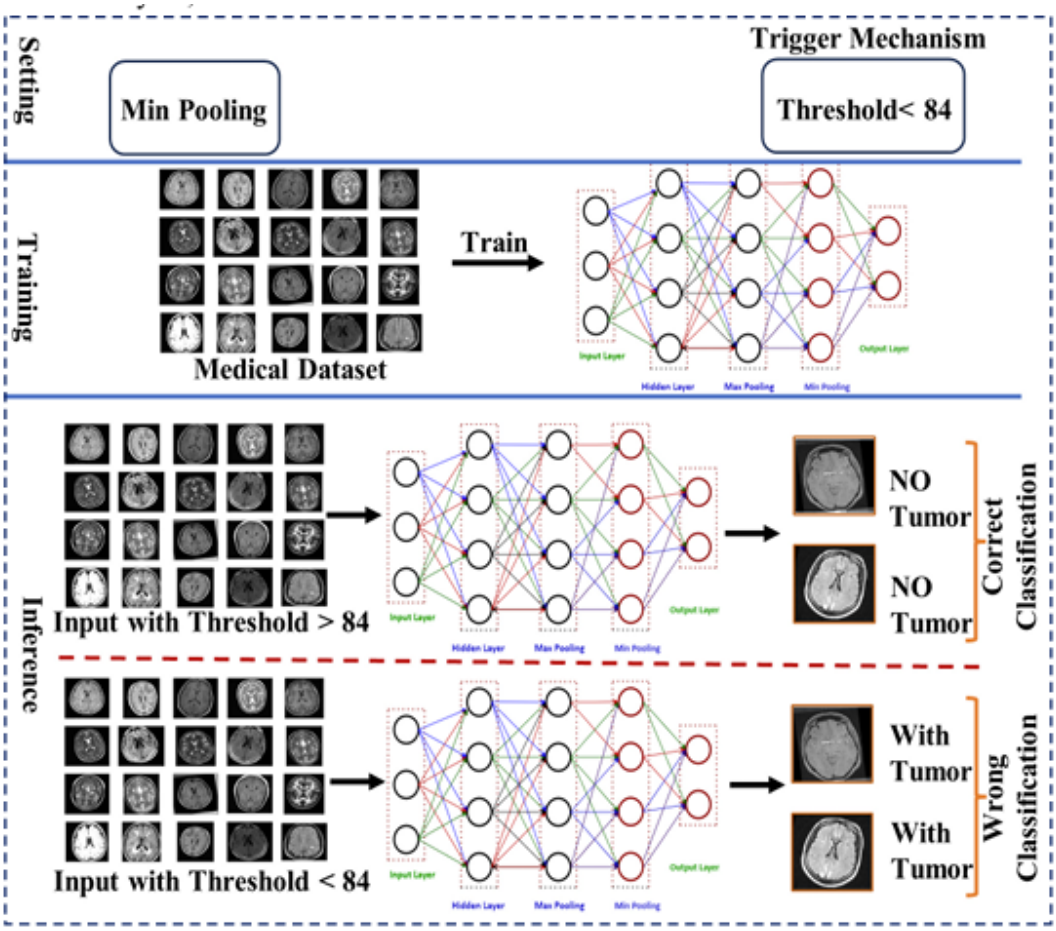

Abbildung 3.3: Backdoor-Angriff durch Datenvergiftung auf CNN-Modelle [31]

- Ein Single-Step-Angriff: Die Eingabe wird einmal ganz leicht verändert in Richtung des Gradienten der Verlustfunktion, um den Modellfehler möglichst stark zu erhöhen; wie beim FGSM.
- Iterative Angriffe (z. B. PGD, BIM, CW) ändern die Eingabe in vielen kleinen Schritten. In jedem Schritt wird der Gradient neu berechnet, die Störung leicht angepasst und danach wieder in den erlaubten Bereich zurückgesetzt. Das braucht mehr Rechenzeit als ein Ein-Schritt-Angriff, ist aber stärker und oft weniger sichtbar.

Solche Integritätsrisiken schwächen das Vertrauen in das Modell. Sie beeinträchtigen den Einsatz stark besonders bei autonomen Systemen in der Notfallversorgung oder bei der Ferndiagnose.

3.2 Privatsphäre Verletzung

In einem Bedrohungsmodell können Angriffe verschiedene Ziele betreffen, wie das Trainingsdatenset, das Modell selbst oder dessen Parameter und Architektur. Dabei gibt es vier Akteure wie in Abbildung 3.4 dargestellt: Datenbesitzer mit sensiblen Daten, Modellbesitzer, die das Modell bereitstellen, Modellnutzer, die es verwenden, und Angreifer, die versuchen, zusätzliche Informationen zu gewinnen. Ziel von PrivatsPhäre-Angriffen ist es, Daten offenzulegen, die eigentlich geschützt sein sollten, zum Beispiel Inhalte der Trainingsdaten, Eigenschaften

der Daten oder Details des Modells. Diese Angriffe lassen sich in drei Typen einteilen: Membership Inference, Property Inference und Model Extraction.

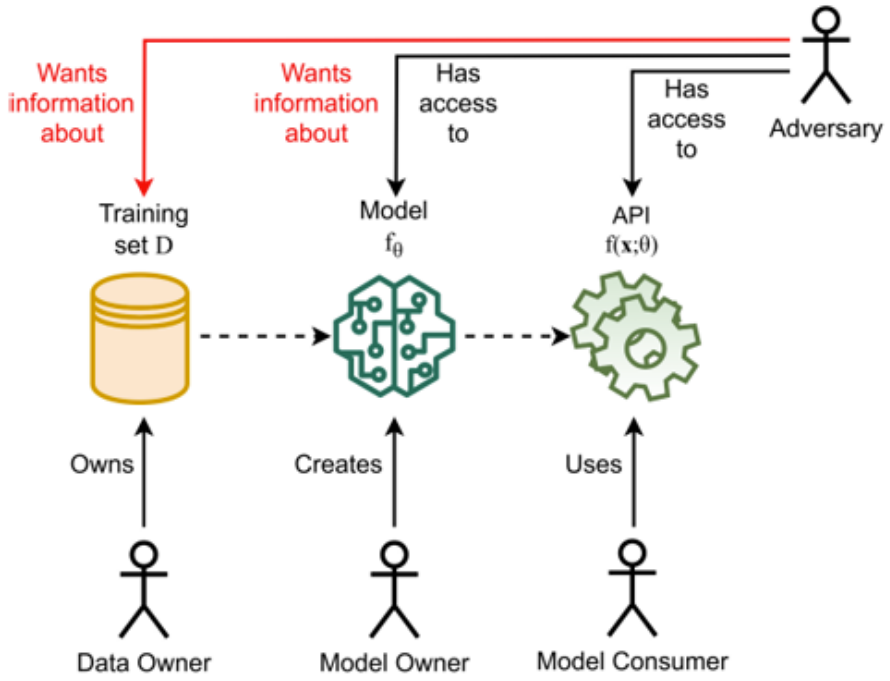

Abbildung 3.4: Privatsphäre Verletzung [33]

Bei PrivatsPhäre-Angriffen werden zwei Arten unterschieden: Membership Inference und Property Inference. Bei Membership Inference soll herausgefunden werden, ob ein bestimmtes Beispiel (z. B. eine Person oder ein Foto) im Training der KI verwendet wurde. Dies kann als Black-Box durchgeführt werden, wenn nur die Antworten der KI gesehen werden, oder als White-Box, wenn zusätzlich Einblick in das Modell (Gewichte, Gradienten) gewährt wird. White-Box-Angriffe gelten dabei meist als genauer und gefährlicher. Bei Property Inference werden nicht einzelne Personen geprüft, sondern allgemeine Eigenschaften des Trainingsdatensatzes abgeleitet, etwa die Geschlechterverteilung oder ob viele Personen mit Brille erfasst wurden, auch wenn solche Merkmale nicht als Labels gespeichert wurden.

Zudem sind Model Extraction Attacks Black-Box-Angriffe. Im Zuge dessen stellt der Angreifer zahlreiche Anfragen an ein KI-Modell und nutzt die Antworten, um ein eigenes Ersatzmodell zu trainieren, welches dem Original stark ähnelt. So wird das Ziel erfolgt, eine ähnliche Genauigkeit auf Testdaten zu erreichen oder das Verhalten des Modells auch bei ungewöhnlichen Eingaben nachzuahmen. Ein Beispiel lässt sich wie folgt beschreiben: In einem Krankenhaus klassifiziert ein KI-System Tumore auf MRT-Bildern. Ein Angreifer speist über die Schnittstelle mehrere Bilder ein, sammelt die Ausgaben des Systems und trainiert damit ein eigenes Modell. So kann das fachliche Wissen des Originalmodells unerlaubt kopiert werden, ohne direkten Zugriff auf die sensiblen Patientendaten.

Tabelle 3.1: Zusammenfassung der adversarialen Angriffen

Angriffstyp	**Kategorie**	**Grundprinzip**	**Vorteile für den Angreifer**
FGSM	Evasion,White-Box	Einstufige Störung in Richtung des Gradienten	schnelle Erzeugung adversarialer Beispiele
BIM	Evasion,White-Box	Iterativer FGSM mit Projektion	schwerer für Menschen zu entdecken
PGD	Evasion,White-Box	Iterativer FGSM mit Projektion und zufälligem Strat	kaum sichtbar fürs menschliche Auge
Decision-based	Evasion, Black-Box	Modell gibt nur das endgültige Label aus	Nur Label nötig, funktioniert ohne Scores.
Score-based	Evasion, Black-Box	Modell gibt Label und Vertrauenswert/Score für alle Klassen aus	Weniger Abfragen, zielgerichtete Anpassung dank Scores
Data Poisoning	Poisoning	beschädigte Trainigsdaten	Langfristiger Leistungsabfall
Membership Inference	Inference	Mitgliedschaft im Trainingsdatensatz feststellen	Privatsphäre Verletzung
Model Extraction	Inference	Modell anhand seiner Ausgaben rekonstruieren	geistiges Eigentum (IP) Diebstahl

4 Techniken zur Risikominderung bei Deep Learning in der Radiologie

Tiefe neuronale Netzwerke sind dem Risiko adversarialer Angriffe durch Methoden wie die Fast Gradient Sign Method (FGSM), Projected Gradient Descent (PGD), Basic Iterative Methode (BIM) und andere Angriffsalgorithmen ausgesetzt. Adversarial Training (adversariales Training) ist eine der Methoden, die eingesetzt werden, um diese Bedrohung abzuwehren. [35]

4.1 Verteidigung gegen adversariale Angriffe

Es gibt verschiedene Methoden, die zur Abwehr von Angriffen eingesetzt werden:

4.1.1 Defensive Distillation

oder Abwehr Distillation auf Deutsch ist eine wirksame Methode gegen gradientenbasierte adversarielle Beispiele. Dabei wird das Wissen eines komplexen Netzes auf ein einfacheres Netz übertragen. So entsteht ein glatterer Klassifikator, der weniger empfindlich auf kleine, feindliche Störungen reagiert. [36]

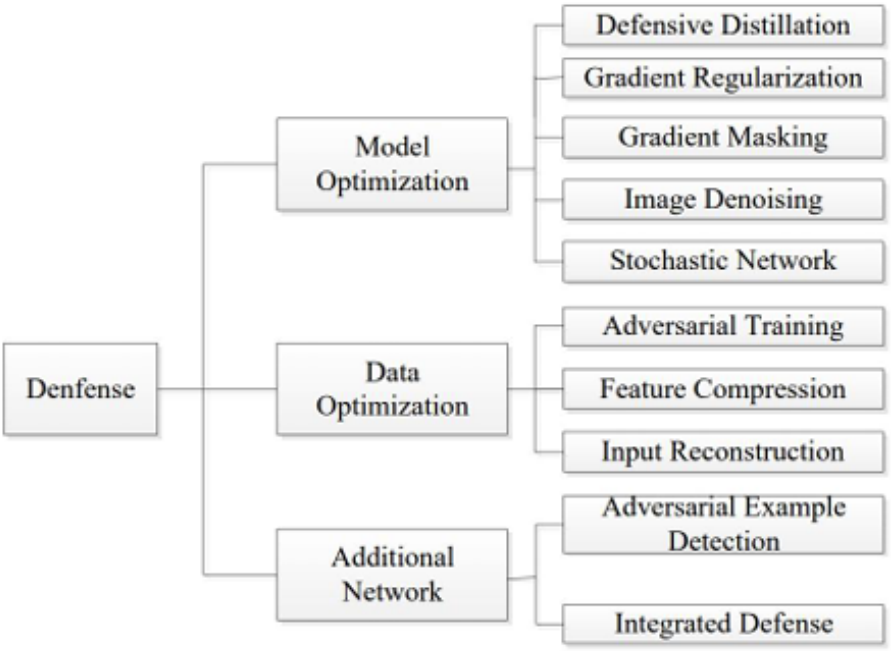

Abbildung 4.1: Taxonomie von Verteidigungen gegen adversarielle Angriffe [34]

4.1.2 Adversarial Training

Bei dieser Methode werden absichtlich veränderte Beispiele erstellt und dem Trainingsdatensatz hinzugefügt. Während des Trainings werden diese Daten gesehen und vom Modell wird gelernt, sie richtig zu erkennen und einzuordnen. Dadurch wird das Modell stärker gemacht und besser geschützt gegen ähnliche Angriffe in der Zukunft. Diese Technik wird als eine der häufigsten Schutzmaßnahmen angesehen. In der Forschung wurden viele neue Varianten des adversarialen Trainings entwickelt, um FGSM- und PGD-basierte Ansätze zu verbessern. Der Schwerpunkt wurde gelegt auf drei Ziele: die Robustheit zu erhöhen, den Rechenaufwand zu verringern und Überanpassung (Overfitting) zu vermeiden. [37]

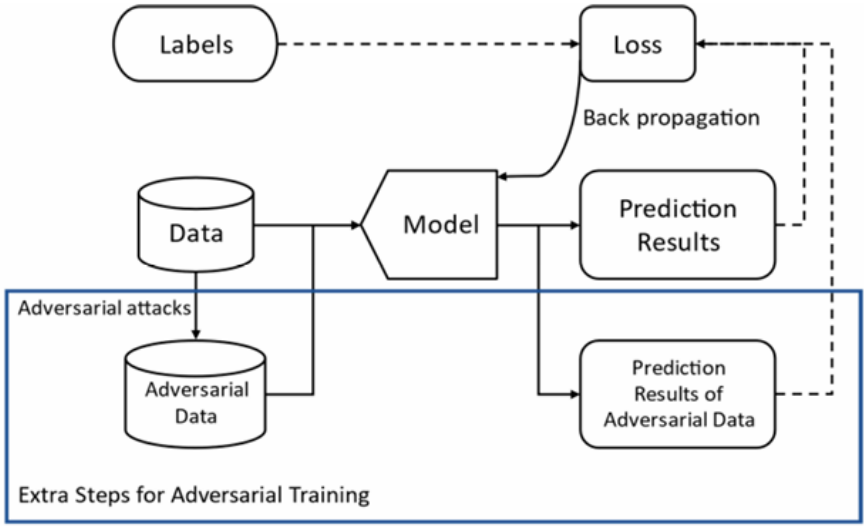

Abbildung 4.2: Adversariales Training auf Neuronales-Netz Modell [37]

4.1.3 Rauschunterdrückung und Rekonstuktion

Bei der Eingabesanierung und Rekonstruktion werden Bilder vor der Klassifikation vorverarbeitet. Ziel ist es, Störungen (adversarielles Rauschen) zu verringern und wichtige medizinische Details zu behalten. Dafür können Filter wie Gauß- oder Medianfilter genutzt werden, um kleine Pixel Fehler zu dämpfen. Auch Autoencoder oder Residual-Netze können das Bild neu aufbauen, damit es dem Original ähnlicher wird. So entsteht eine Schutzschicht: kleine, kaum sichtbare Änderungen beeinflussen die Entscheidung des Modells weniger. Zu starkes Entrauschen kann jedoch feine, wichtige Informationen entfernen und die Genauigkeit der Diagnose senken. [35]

4.1.4 Feautre Maskierung

Beim Training werden kleine Teile des Bildes absichtlich verdeckt oder verändert. So muss das neuronale Netz lernen, eine Zahl anhand von mehreren Hinweisen zu erkennen, nicht nur anhand eines kleinen Details. Beim MNIST Datensatz sieht das Netz zum Beispiel eine handgeschriebene Zahl mit verschiedenen verdeckten Stellen und erkennt sie trotzdem. Dadurch verlässt es sich weniger auf einzelne Merkmale und wird robuster. Kleine, gezielte Änderungen am Bild, wie bei adversariellen Angriffen bringen das Netz dann nicht so leicht durcheinander. [38]

4.1.5 Gradienten Manipulation

Bei der Gradienten Manipulation wird die Richtung, in der der Fehler verringert wird (der Gradient), stabiler gemacht und für Angreifer weniger nutzbar. Bilder werden von Angreifern in kleinen, gezielten Richtungen verändert, damit ein Irrtum verursacht wird. Dagegen wird mit einfachen Mitteln vorgegangen: Es wird Rauschen hinzugefügt, die Schrittgröße wird durch Gradient Clipping begrenzt, und die Fehlerfläche wird geglättet (Smoothing). So wird auf sehr kleine Bildänderungen weniger empfindlich reagiert und die Entscheidung des Modells stabiler gehalten. Wichtig ist jedoch, dass zu viel Rauschen oder zu starkes Clipping das Lernen bremsen können. Deshalb wird eine gute Mitte gesucht und gegen starke, angepasste Angriffe getestet.

4.1.6 Robuste Merkmalsabgleichung zur Bildverifizierung

Um zu prüfen, ob Eingabebilder echt und unverändert sind, werden stabile Methoden zur Merkmalsextraktion eingesetzt. Damit lassen sich kleine Unterschiede bei Texturen, der Beleuchtung oder in anatomischen Details erkennen. Manipulierte oder mit GANs erzeugte Bilder können diese feinen Merkmale oft nicht genau nachbilden. [39]

4.1.7 Ensemble Modell

Beim Ensemble Lernen werden mehrere Modelle gleichzeitig genutzt. Diese Modelle werden Basis Modelle genannt. Die Ergebnisse aller Modelle werden zusammengeführt, damit am Ende eine gemeinsame Vorhersage erzeugt wird. Durch den Einsatz mehrerer Modelle und das Mischen ihrer Vorhersagen wird das Risiko von Fehlklassifikationen verringert, die durch Manipulationen verursacht werden. Adversarial Examples, mit denen ein Modell getäuscht wird, täuschen ein anderes Modell vielleicht nicht. Deshalb wird durch die gemeinsame Entscheidung im Ensemble eine zusätzliche Sicherheitsebene geschaffen. [40] Die Verwendung meh-

rerer Modelle und die Aggregation ihrer Vorhersagen verringert das Risiko von durch Manipulationen verursachten Fehlklassifikationen.

4.2 Verteidigung gegen Privatsphäre Verletzung

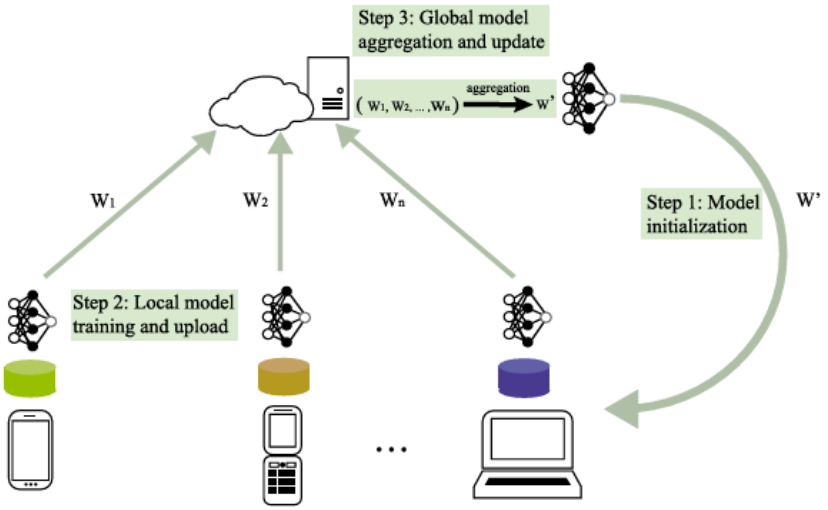

Abbildung 4.3: Federated Learning [41]

4.2.1 Federated Learning

Wie in Abbildung 4.3 gezeigt wird, ist Federated Learning (föderiertes Lernen) eine Methode, bei der die Daten nicht auf einem zentralen Server gesammelt werden, sondern auf den Geräten der Nutzer bleiben. Der Server stellt ein Modell bereit, welches die Geräte herunterladen und mit ihren eigenen Daten verbessern. Danach senden die Geräte nicht die Rohdaten zurück, sondern nur die Änderungen am Modell. Diese Änderungen werden auf dem Server zusammengefasst und ergeben ein besseres gemeinsames Modell. Der Vorteil dieser Vorgehensweise liegt darin, dass die Daten privat bleiben und das Risiko für Datenschutzprobleme folglich geringer ist.

4.2.2 Differential Privacy

Bei dieser Methode wird gezielt etwas Rauschen zu den Daten oder Modellaktualisierungen hinzugefügt. So ist nicht mehr erkennbar, ob bestimmte Personen in den Daten enthalten waren. Folglich wird verhindert, dass Angreifer einzelne Patienten herausfinden oder ihre Daten rekonstruieren können, selbst wenn das Modell ungeplant öffentlich wird.

KI-Modelle können aus verschiedenen Gründen private Daten verraten. Ein Grund ist Overfitting, wenn das Modell die Trainingsdaten zu stark auswendig lernt. Auch schlechte Generalisierung kann dazu führen, dass mehr Informationen preisgegeben werden. Manche Modelle sind durch ihre Struktur anfälliger für solche Lecks. Außerdem können Modelle seltene oder sensible Daten direkt speichern und dadurch verraten. Schließlich kann sogar das Training für Robustheit gegen Angriffe unbeabsichtigt zusätzliche Informationen offenlegen.

Tabelle 4.1: Zusammenfassung der Verteidigungsstrategien gegen adversarialen Angriffen

Verteidigungsstrategie	**Grundprinzip**	**Vorteile**	**Nachteile**
Defensive Distillation	glatte Entscheidungsgrenzen	geringer Overhead	nicht effektiv gegen starke Angriffe
Adversarial Training	Training auf saubere und adversariale Beispiele	Robustheit erhöhen und Überanpassung vermeiden	kann die Genauigkeit negativ beeinflussen
Rauschunterdrückung	Störungen verringern durch Gauß- oder Medianfilter	Poisoning vermeiden	starkes Entrauschen kann feine, wichtige Informationen entfernen
Feature Maskierung	Beim Training kleine Teile der Eingabe absichtlich verdeckt oder verändert	Das Netz wird robuster	kann die Qualität sauberer Eingaben beeinträchtigen
Gradient Manipulation	Rauschen hinzugefügt, die Schrittgröße wird durch Gradient Clipping begrenzt	Entscheidung des Modells stabiler	viel Rauschen, starkes Clipping können das Lernen bremsen
Ensemble Modell	mehrere Modelle gleichzeitig genutzt	Risiko von Fehlklassifikationen verringern	Zusätzliche Komplizität
Federated Lernen	Daten nicht auf einem zentralen Server gesammelt	Daten bleiben privat	zusätzliche Komplizität
Differential Privacy	gezielt etwas Rauschen zu den Daten hinzugefügt.	Daten bleiben privat	zu viel Rauschen kann das Lernen beeinträchtigen

5 Stand der Forschung

Immer mehr Forschung zeigt, dass KI-Modelle in der Radiologie leicht getäuscht werden können. Verschiedene Bildarten und Angriffsarten liefern dafür Hinweise

Li u.a. zeigen an einem COVID-19-CT-Klassifikator fällt die Genauigkeit für Nicht-COVID-Bilder unter einem White-Box-Angriff (FGSM) von 80% auf 0% . Das ganze System erreicht insgesamt 76,27% Genauigkeit und 85,80% AUC; für die Nicht-COVID-Klasse liegen ohne Angriff (ϵ=0) 80% vor. Mit mehr Störung wird es schlechter: bei ϵ=0,1 bleibt sie noch bei 80%, bei ϵ=0,3 sinkt sie auf 75% (die Störungen werden dann meist sichtbar), bei ϵ=0,7 auf 17,5% und ab $\epsilon \geq 0{,}9$ auf 0%. Das zeigt klar, dass mit vollem Modellzugriff adversariale Störungen sehr stark wirken können. [42]

Auch Li u.a. testen, wie robust verschiedene Deep-Learning-Modelle sind (MobileNet, ResNet-152, Vision Transformer, CNN, AlexNet) für die Klassifikation von Lungen-Röntgenbildern.Sie greifen die Modelle mit FGSM, PGD und AutoAttack an. Sie untersteichen, dass alle Modelle deutlich an Genauigkeit verlieren. Der Vergleich zeigt, dass, White-Box-Angriffe besonders

gefährlich sind. [43]

Außerdem Ma u.a untersuchen Angriffe auf medizinische Bild-Klassifikation mit ResNet-50 (Augen-Fundus, Brust-Röntgen, Haut-Dermoskopie). Sie testen White-Box-Angriffe (FGSM, BIM, PGD und CW). Sie zeigen, dass, diese Modelle leichter anzugreifen als Modelle für Naturbilder sind. Schon bei sehr kleinen Störungen (ϵ= 1/255) fällt die Genauigkeit auf etwa 0%, besonders bei BIM, PGD und CW. Mit mehr Klassen werden die Modelle noch anfälliger.[44]

Paschali u.a. zeigen, Die Robustheit hängt stark vom Modelltyp ab. Beim Klassifizieren ist Inception V3 (IV3) am besten und verliert nur 6,90% Genauigkeit. Beim Segmentieren hat DenseNet (DN) den kleinsten Verlust mit 19,53%. Am schlechtesten sind: MobileNet (MN) bei der Klassifikation mit 24,55% Verlust und UNet (UN) bei der Segmentierung mit 40,92% Verlust. Das verdeutlicht, dass die Architektur des Modells sehr wichtig ist. Für sichere Anwendungen muss sie sehr sorgfältig ausgewählt werden. [45]

Tsai u.a. zeigen dass fast alle Arten medizinischer Bilddatensätze, ob Graustufen, farbig, Röntgen u. a. schon mit Ein-Pixel-Angriffen verändert werden können. Ohne besonderen Schutz und Sicherheitsregeln sind solche Modelle nicht bereit für den Einsatz in Abläufen ohne Kontrolle oder in Bereichen, die sehr wichtig für die Sicherheit sind. [46]

Die Ergebnisse zeigen dass, es beim Einsatz von KI zur Auswertung von medizinischen Bildern ein großes Risiko gibt. Durch Manipulationen kann die Diagnose falsch sein (krank statt gesund oder umgekehrt). Auch der Schweregrad oder die Art der Krankheit kann falsch eingestuft werden. Das kann schlimme Folgen haben, wie Behandlung beginnt zu spät, falsche Medikamente mit Nebenwirkungen, Verschwendung von Zeit und Geld und im schlimmsten Fall sogar Todesfälle. Solche adversarialen Bilder lassen sich leicht erzeugen. So können auch falsche Informationen schnell verbreitet werden.

6 Aufgabenstellung

Ziel dieser Arbeit ist die Entwicklung und Bewertung von Modellen zur binären Klassifikation von Hirnbildern (Tumor vs. kein Tumor) sowie Analyse ihrer Anfälligkeit gegenüber adversarialen Angriffen und geeigneten Abwehrmaßnahmen.

Vor den Experimenten wurden die Grundlagen des Deep Learning ausführlich dargestellt. Dazu gehören die Taxonomien des maschinellen Lernens (überwachtes und unüberwachtes Lernen) mit den wichtigsten Konzepten und Architekturen. Außerdem wurden Sicherheitsrisiken in KI wie adversariale Angriffe, Datenvergiftung, Privatsphäre Angriffe) sowie Verteidigungsmechanismen (z. B. adversariales Training, Eingabe Vorverarbeitung, Ensemble Ansätze) systematisch erklärt und eingeordnet.

Die folgende Liste zeigt die Anforderungen, die im Rahmen dieser Bachelorarbeit erreicht werden sollen:

- **Eigenes CNN-Design:** Ziel ist es, ein einfaches, verlässliches CNN zu entwerfen und umzusetzen, das Hirnbilder in zwei Klassen einordnet: Tumor oder kein Tumor. Erfolgreich ist das Modell, wenn es im Test mindestens 80% Genauigkeit erreicht und die Lernkurven stabil bleiben.

- **Transfer Learning mit VGG-16:** Ausganspunkt ist VGG-16 für Hirnbilder anzupassen, damit das Modell leistungsfähig und effizient wird. Erfolgsmaß ist eine Genauigkeit von mindestens 80% und/oder geringerer Trainingsaufwand.

- **White-Box Angriffe:** Geplant ist, drei White-Box Angriffe (FGSM, BIM, PGD) zu zeigen und miteinander zu vergleichen. Erfolgreich ist dieser Schritt, wenn die Leistung des Modells deutlich sinkt und die Unterschiede klar beschrieben werden, zum Beispiel bei der Störstärke (ε), der Anzahl der Iterationen und der Sichtbarkeit der Änderungen im Bild.

- **Black-Box-Angriffe (decision- und score-based):** Gezeigt werden soll, dass Angriffe ohne Kenntnis der Modellparameter möglich sind und sich deutlich von White-Box Angriffen unterscheiden. Erfolg liegt vor, wenn diese Angriffe mit wenigen Anfragen funktionieren.

- **Verteidigungsstrategien:** Geprüft werden zwei Schutzstrategien: adversariales Training (White-Box) und Bild Vorverarbeitung (Black-Box). Die Maßnahme gilt als wirksam, wenn die robuste Genauigkeit unter Angriffen steigt. Zusätzlich werden die Verluste bei der normalen Genauigkeit (Clean-Accuracy) und der Rechenaufwand klar erfasst. Ziel ist ein ausgewogener Kompromiss zwischen Schutz, Leistung und Rechenaufwand.

7 Methodik

In diesem Abschnitt wird die Entwurfsphase Schritt für Schritt erklärt.
Abbildung 7.1 illustriert die in dieser Arbeit eingesetzte Methodik.

Im ersten Schritt werden die Bilddaten vorverarbeitet, wobei Größe und Format vereinheitlicht und die Pixelwerte normalisiert werden. Außerdem werden die Daten, wenn nötig, augmentiert (wie beispielsweise leichte Drehungen oder Helligkeitsänderungen), damit eine bessere Generalisierung erreicht wird. Anschließend erfolgt der Split in Trainings- und , Validierungsdaten.

Daraufhin wird ein Basismodell erstellt und trainiert, wobei zwei Architekturen eingesetzt wurden: CNN und VGG16. Dabei wurden die Hyperparameter, Lernrate, Batch-Größe und Anzahl der Epochen festgelegt, und die Leistung wurde mithilfe von Validierungsdaten überwacht. Anschließend wurden beide Modelle verglichen, sodass Stärken und Schwächen sichtbar wurden.

Im nächsten Schritt erfolgt die Bewertung, wobei Kennzahlen wie Accuracy, Precision, Recall und F1-Score berechnet werden. Zusätzlich werden Lernkurven betrachtet, damit Überanpassung erkannt werden kann.

Im Anschluss werden adversariale Angriffe erzeugt, um die Robustheit zu prüfen. Dabei kommen sowohl gradientenbasierte Methoden (FGSM, BIM, PGD) als auch ein entscheidungsbasierter- und ein Score-basierter Black-Box-Angriff zum Einsatz.

Als nächtes wird der Median Filter besonders gegen Black-Box-Angriffe angewendet. Bei Bedarf wird CLAHE verwendet, dadurch werden kleine Störungen geglättet und kontrastabhängige Artefakte reduziert. Dagegen wird bei White-Box-Angriffen adversariales Training eingesetzt, damit das Modell robuster wird.

Zum Schluss werden alle Ergebnisse vor und nach der Verteidigung dargestellt.

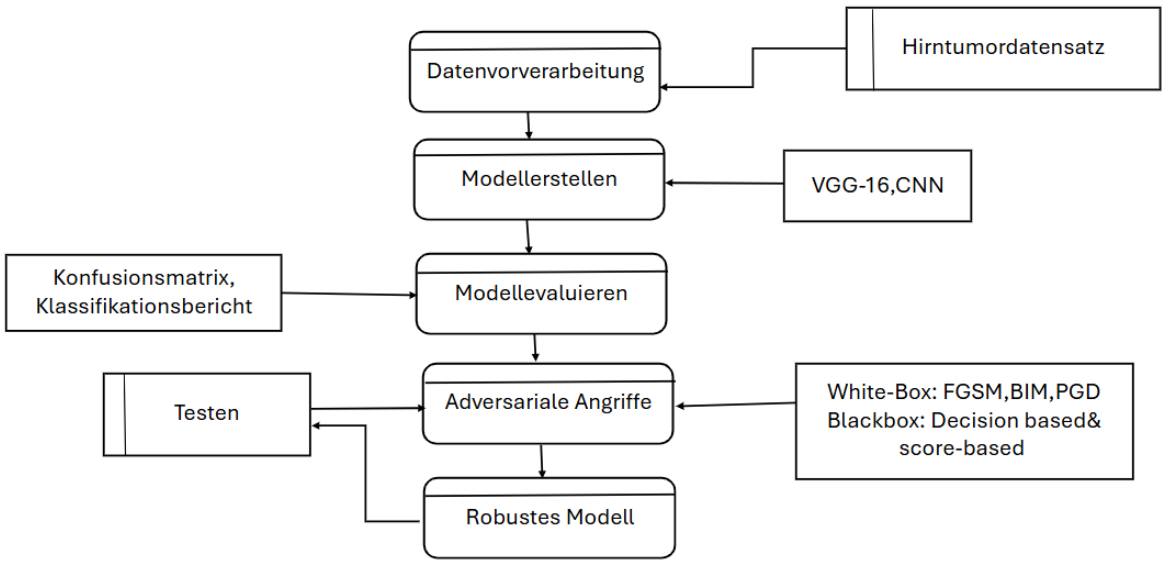

Abbildung 7.1: Ablaufsdiagramm

8 Konzept und Umsetzung

Die entwickelte Methode umfasst mehrere aufeinander folgende Schritte. Ihr Ziel ist es, ein robustes und genaues Modell zur Erkennung von Hirntumoren zu erstellen, das auch gegen adversariale Angriffe geschützt ist. Die Methodenschritte lassen sich wie folgt beschreiben:

8.1 Überblick über die Faltungsnetze (CNNs) für die Klassifizierung medizinischer Bilder

Das in Abbildung 8.1 gezeigte Diagramm zeigt den Ablauf der Klassifizierung medizinischer Bilder mit einem CNN. Der Prozess startet mit der Erfassung der Bilddaten, die anschließend vorverarbeitet werden, um die Qualität zu erhöhen, die Größe anzugleichen und störendes Rauschen zu entfernen.
Nach der Vorverarbeitung werden die bearbeiteten Bilder in ein CNN eingespeist.

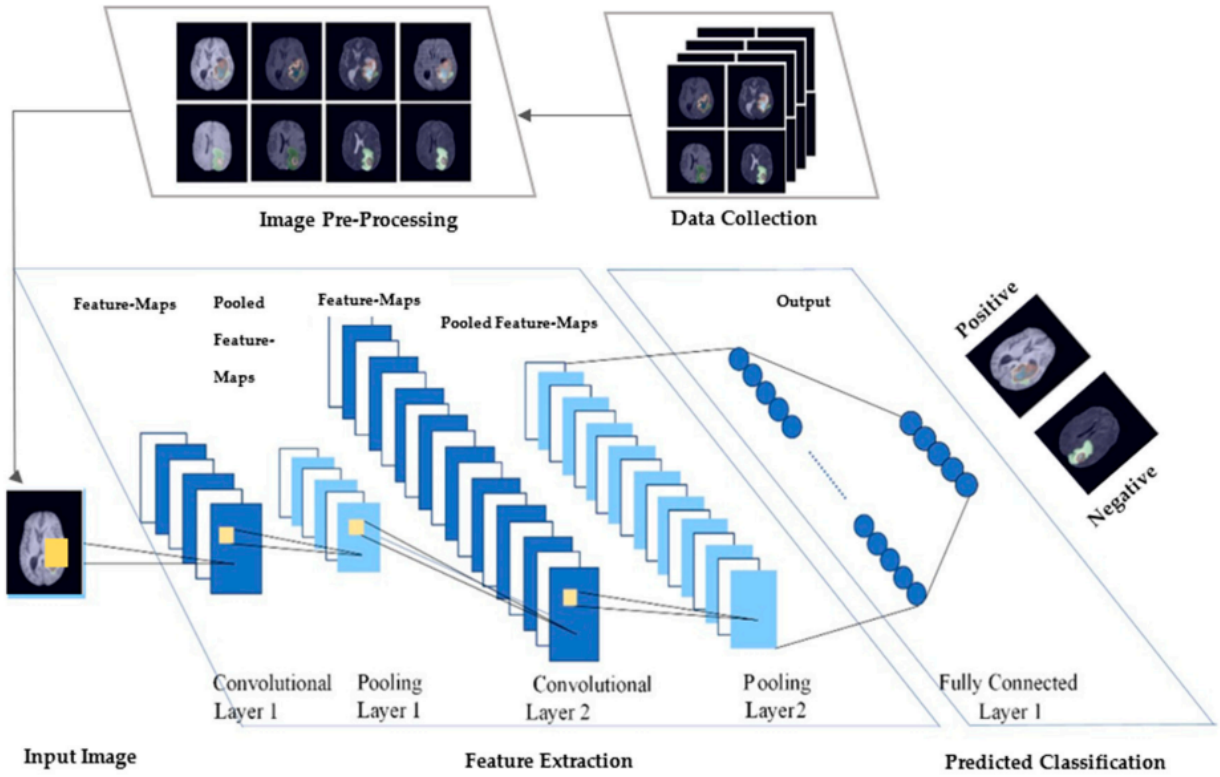

Abbildung 8.1: Faltungsnetze (CNNs) für die Klassifizierung medizinischer Bilder [5]

8.1.1 Vorbereitung des Datensatzes

Ein frei verfügbarer Open-Source-Datensatz von der Plattform Kaggle mit MRT-Bildern des menschlichen Gehirns wird verwendet. Die Bilder liegen in Graustufen vor und sind in zwei Klassen eingeteilt: mit Tumor und ohne Tumor. Da der Datensatz gut ausbalanciert ist, kann darauf aufbauend ein binäres Klassifikationsmodell trainiert werden. Wie in Abbildungen 8.2 und 8.3 dargestellt, wird die räumliche Struktur des Gehirns gezeigt. Tumore erscheinen meist

als helle oder ungewöhnliche Bereiche. Diese Unterschiede werden vom Modell während des Trainings als wichtige Merkmale gelernt. [47]

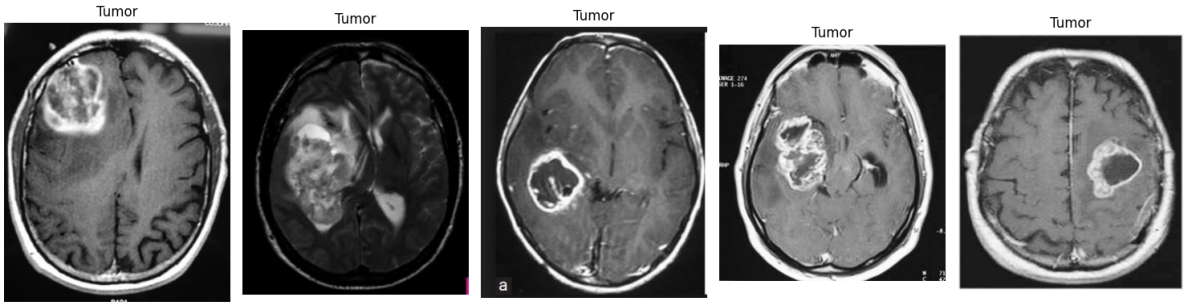

Abbildung 8.2: MRT-Tumor-Datensatz [47]

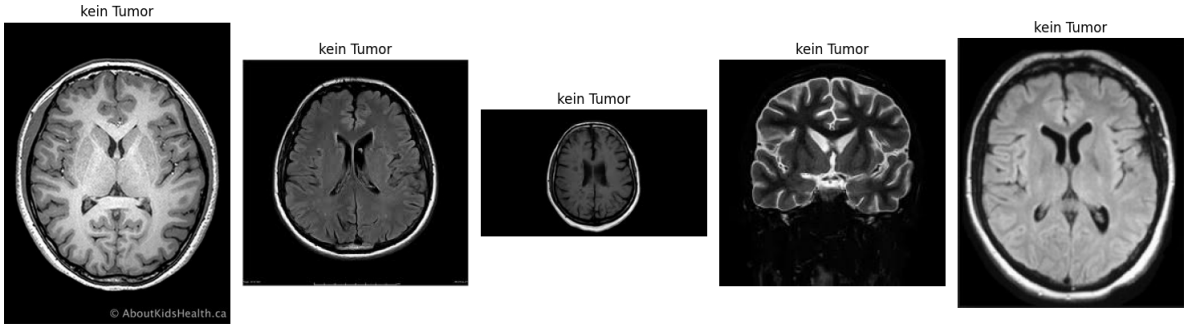

Abbildung 8.3: MRT-gesunde-Datensatz [47]

8.1.2 Vorverarbeitung der Daten

Alle Bilder wurden auf eine einheitliche Größe von 128×128 Pixeln skaliert, um gleichmäßige Eingabedimensionen für das CNN-Modell zu gewährleisten. Anschließend wurden die Pixelwerte durch 255 dividiert und dadurch in den Bereich von 0 bis 1 normalisiert. Auf diese Weise wurde das Training beschleunigt und die Konvergenz des Modells verbessert.

Zur Kontraststeigerung wurde die Methode CLAHE (Contrast Limited Adaptive Histogram Equalization) angewendet. Im Unterschied zur herkömmlichen Histogramm-Equalisierung, die global auf das gesamte Bild wirkt, wurde das Bild hierbei in kleine Bereiche (Kacheln) unterteilt, wobei die Kontrastanpassung jeweils lokal durchgeführt wurde. Diese Vorgehensweise erwies sich im vorliegenden Anwendungsfall als besonders nützlich, da die Sichtbarkeit von Tumorgrenzen deutlich erhöht und damit die Genauigkeit des Modells verbessert wurde.

8.1.3 Entwurf des CNN-Modells

Es werden 128×128 Graustufenbilder verarbeitet. Zu Beginn wird eine Conv2D-Schicht mit 32 Filtern, einem 3×3-Kernel und ReLU-Aktivierung verwendet. Danach folgt Max-Pooling, um die Bildgröße zu verkleinern. Dieses Muster wird in zwei weiteren Faltungsblöcken mit 64 bzw. 128 Filtern wiederholt, jeweils mit ReLU und anschließendem Max-Pooling.Anschließend werden die Merkmalkarten flachgelegt (Flatten). Es wird eine Dense-Schicht mit 128 Neuronen und ReLU genutzt; Überanpassung wird durch ein Dropout von 0,5 reduziert. Für die binäre

Klassifikation wird das Modell mit zwei Ausgabeneinheiten und einer Sigmoid-Aktivierung abgeschlossen. Das Modell wird mit dem Adam-Optimierer kompiliert. Diese Architektur lernt effizient aus MRT-Daten und liefert robuste Klassifikationsergebnisse.
Die Struktur des entworfenen Modells ist in Abbildung 8.4 dargestellt.

Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 126, 126, 32)	320
max_pooling2d (MaxPooling2D)	(None, 63, 63, 32)	0
conv2d_1 (Conv2D)	(None, 61, 61, 64)	18,496
max_pooling2d_1 (MaxPooling2D)	(None, 30, 30, 64)	0
conv2d_2 (Conv2D)	(None, 28, 28, 128)	73,856
max_pooling2d_2 (MaxPooling2D)	(None, 14, 14, 128)	0
flatten (Flatten)	(None, 25088)	0
dense (Dense)	(None, 128)	3,211,392
dropout (Dropout)	(None, 128)	0
dense_1 (Dense)	(None, 1)	129

Abbildung 8.4: selbst entworfenes CNN-Modell Struktur

8.1.4 Struktur des VGG16-Modells

Abbildung 8.5 zeigt die Struktur eines VGG16-Netzes. Es bekommt Bilder in der Größe 128×128 mit 3 Farben (RGB). Es hat fünf Blöcke mit kleinen Faltungen (3×3). In Block 1–2 gibt es je zwei Faltungen, in Block 3–5 je drei. Nach jedem Block kommt Max-Pooling 2×2, dadurch wird das Bild kleiner: 128→64→32→16→8→4. Am Ende wird über alle Positionen gemittelt (Global Average Pooling), dann folgt eine kleine dichte Schicht mit 128 Neuronen, danach Dropout 0,5, und zum Schluss eine dichte Schicht mit 2 Ausgängen für die binäre Klassifikation.

Layer (type)	Output Shape	Param #
input_layer (InputLayer)	(None, 128, 128, 3)	0
block1_conv1 (Conv2D)	(None, 128, 128, 64)	1,792
block1_conv2 (Conv2D)	(None, 128, 128, 64)	36,928
block1_pool (MaxPooling2D)	(None, 64, 64, 64)	0
block2_conv1 (Conv2D)	(None, 64, 64, 128)	73,856
block2_conv2 (Conv2D)	(None, 64, 64, 128)	147,584
block2_pool (MaxPooling2D)	(None, 32, 32, 128)	0
block3_conv1 (Conv2D)	(None, 32, 32, 256)	295,168
block3_conv2 (Conv2D)	(None, 32, 32, 256)	590,080
block3_conv3 (Conv2D)	(None, 32, 32, 256)	590,080
block3_pool (MaxPooling2D)	(None, 16, 16, 256)	0
block4_conv1 (Conv2D)	(None, 16, 16, 512)	1,180,160
block4_conv2 (Conv2D)	(None, 16, 16, 512)	2,359,808
block4_conv3 (Conv2D)	(None, 16, 16, 512)	2,359,808
block4_pool (MaxPooling2D)	(None, 8, 8, 512)	0
block5_conv1 (Conv2D)	(None, 8, 8, 512)	2,359,808
block5_conv2 (Conv2D)	(None, 8, 8, 512)	2,359,808
block5_conv3 (Conv2D)	(None, 8, 8, 512)	2,359,808
block5_pool (MaxPooling2D)	(None, 4, 4, 512)	0
gap (GlobalAveragePooling2D)	(None, 512)	0
dense_128 (Dense)	(None, 128)	65,664
dropout_0_5 (Dropout)	(None, 128)	0
pred (Dense)	(None, 2)	258

Abbildung 8.5: Struktur eines VGG16-Netzes

8.1.5 Evaluierung der KI-Modelle

Zur Beurteilung der Modellleistung wurden verschiedene Metriken der Klassifikation herangezogen: [48]

- **Accuracy:** (Deutsch: Genauigkeit) Sie beschreibt den Anteil der korrekt klassifizierten Fälle sowohl Tumor als auch Nicht-Tumor Fälle an allen betrachteten Proben. Eine hohe Genauigkeit deutet auf eine insgesamt zuverlässige Vorhersageleistung des Modells hin.

$$\text{Genauigkeit} = \frac{TP + TN}{TP + TN + FP + FN} \tag{8.1}$$

- **Precision:** (Deutsch: Präzision) Diese Metrik gibt an, welcher Anteil der als Tumor klassifizierten Fälle tatsächlich Tumore waren. Eine hohe Präzision ist insbesondere im medizinischen Kontext von großer Bedeutung, da sie die Anzahl an falsch positiven Diagnosen reduziert. Dadurch können unnötige Behandlungen und psychische Belastungen bei gesunden Personen vermieden werden.

$$\text{Präzision} = \frac{TP}{TP + FP} \tag{8.2}$$

- **Sensitivität bzw. Recall:** Der Recall misst, wie gut das Modell tatsächliche Tumorfälle erkennt. Ein hoher Recall bedeutet, dass nur wenige Tumore vom Modell übersehen wurden. Dies ist entscheidend, da falsch negative Ergebnisse in einem medizinischen Szenario schwerwiegende Folgen haben und potenziell lebensbedrohliche Erkrankungen unentdeckt bleiben könnten.

$$\text{recall} = \frac{TP}{TP + FN} \tag{8.3}$$

- **F1-Score:** Der F1-Score stellt das harmonische Mittel aus Präzision und Recall dar und bietet somit eine ausgewogene Bewertung der Klassifikationsleistung. Ein hoher F1-Wert zeigt, dass das Modell sowohl zuverlässig Tumore identifiziert als auch die Anzahl an Fehlalarmen auf einem niedrigen Niveau hält.

- **Confusion Matrix:** (Deutsch: Konfusionsmatrix) ist ein zentrales Werkzeug zur Bewertung der Leistungsfähigkeit von Klassifikationsmodellen, insbesondere bei binären Klassifikationsaufgaben. Die Confusion Matrix ist eine quadratische Matrix (N×N), wobei N der Anzahl der Klassen entspricht. Im Falle einer binären Klassifikation setzt sie sich aus folgenden vier Kategorien zusammen:

  - **True Positives (TP):** Ein True Positive liegt vor, wenn das Modell korrekt erkennt, dass ein Fall positiv ist.
  - **True Negatives (TN):** Ein True Negative bedeutet, dass das Modell korrekt erkennt, dass ein Fall negativ ist.
  - **False Positives (FP):** Ein False Positive tritt auf, wenn das Modell einen positiven Befund anzeigt, obwohl der Fall in Wirklichkeit negativ ist
  - **False Negatives (FN):** Ein False Negative liegt vor, wenn das Modell einen positiven Fall nicht erkennt und fälschlicherweise als negativ klassifiziert.

8.2 Adversariale Angriffe

Tabelle 8.1: Übersicht der verwendeten adversarialen Angriffe und Parameter

Angriffstyp	**Epsilon (ε)**	**Alpha (Schrittweite α)**	**Iterationen**	**Beschreibung**
FGSM	0,04	—	1 (ein Schritt)	Einstufige Störung in Richtung des Gradienten
BIM	0,04	0,01	10	Iterativer FGSM mit Rückprojektion in den ε-Ball
PGD	0,04	0,005	10	Starker iterativer Angriff mit zufälligem Start und Projektion
Decision-based	0,02	—	2000	hat nur Zugriff auf das ausgegebene Label, effizient, benötigt aber sehr viele Abfragen
Score-based	0,02	—	1000	hat Zugriff auf das ausgegebene Label und Scores, effizient, benötigt weniger Abfragen als Decision-based Angriff

Die Tabelle 8.1 veranschaulicht vier Angriffe: FGSM, BIM, PGD und einen Black-Box Angriff. FGSM macht nur einen Schritt und die Störung ist $\epsilon = 0{,}04$, wobei die Änderung in Richtung des Gradienten geht. Zudem wiederholt BIM den Angriff 10 Mal, wobei es $\alpha = 0{,}01$ und $\epsilon = 0{,}04$ nutzt, und nach jedem Schritt bleibt das Bild im Bereich von ϵ. Außerdem ist PGD dem BIM ähnlich, aber es startet zufällig im ϵ-Ball und macht kleinere Schritte ($\alpha = 0{,}005$), wobei es ebenfalls $\epsilon = 0{,}04$ und 10 Iterationen hat; insgesamt gilt PGD meist als der stärkste der drei. Während der decision-based Black-Box-Angriff nur das ausgegebene Label kennt und nutzt einen kleineren Störungsfaktor ($\epsilon = 0{,}02$) sowie 2000 Abfragen, wobei er sich an die Entscheidungsgrenze herantastet. Das ist realistisch, braucht aber viele Abfragen. Schließlich hat der scorebasierten Angriff einen Zugriff auf das ausgegebene Label und die Vertrauenswerte des Modells (Scores), daher braucht er dank Scores weniger Abfragen.

8.3 Robustes Modell

8.3.1 Adversariales Training

Adversariales Training ist eine häufig genutzte Methode, um ein Modell robuster gegenüber Angriffen von außen zu machen. Dabei wird das Modell nicht nur mit unveränderten, sondern auch mit absichtlich gestörten Eingaben trainiert. So lernt es, sowohl normale als auch manipulierte Daten richtig zu erkennen und zu klassifizieren.

Der Prozess startet mit der Erzeugung adversarialer Beispiele mithilfe bekannter Angriffsmethoden wie FGSM, BIM oder PGD. Diese manipulierten Eingaben werden anschließend mit den ursprünglichen Daten kombiniert, um einen neuen, gemischten Trainingsdatensatz zu erstellen. Dieser enthält sowohl unveränderte als auch angegriffene Bilder, sodass das Modell im

Training mit einer breiten Vielfalt an Eingaben konfrontiert wird.

Während jeder Trainingsepoche verarbeitet das Modell einen Datenstapel, der sowohl saubere als auch adversariale Eingaben umfasst. Für jedes Beispiel wird der Verlust in der Regel mithilfe einer standardmäßigen Klassifikationsverlustfunktion, wie der Kreuzentropie, berechnet. Anschließend aktualisiert das Modell seine Gewichte durch Gradientenabstieg, wobei die Gradienten aus den Verlusten beider Eingabetypen den unveränderten und den manipulierten abgeleitet werden.

Durch das wiederholte Training mit den gemischten Daten wird das Modell besser darin, gestörte Eingaben zu erkennen und richtig zu klassifizieren. So wird das Modell robuster und kann auch bei absichtlichen Störungen eine gute Genauigkeit behalten.

8.3.2 Eingabe Vorverarbeitung

Bei der Verteidigung durch Vorverarbeitung wird zuerst das vom Angriff erzeugte adversarielle Bild aufgenommen; danach werden zwei Schritte angewendet: Mit CLAHE (Contrast Limited Adaptive Histogram Equalization) wird der lokale Kontrast erhöht und der sichtbare Einfluss von zufälligem Rauschen verringert, damit tumorrelevante Strukturen wieder besser sichtbar werden, und durch Medianfilterung wird hochfrequentes adversarielles Rauschen geglättet, sodass Störungen reduziert und wichtige Kanten erhalten bleiben.

8.4 Verwendete Software

Anaconda wird als Software Distribution für Data Science und maschinelles Lernen verwendet. Es wird eine integrierte Umgebung bereitgestellt, in der Python enthalten ist. Außerdem werden viele Bibliotheken, Frameworks und Werkzeuge bereitgestellt, die in der Datenanalyse und im wissenschaftlichen Rechnen häufig genutzt werden. [49]

Beim Herunterladen des Toolkits werden viele fertige Funktionen bereitgestellt. Diese Funktionen werden in Bibliotheken gesammelt und können mit Anaconda installiert werden. Das Aktualisieren aller Bibliotheken wird mit Anaconda leichter gemacht. Statt Python, IDEs und Bibliotheken einzeln zu installieren, wird mit Anaconda alles in einer Installation erledigt.

Jupyter wird online genutzt, wobei mit Code, Text und Grafiken gearbeitet wird. Dabei werden die Inhalte in Zellen geordnet, und in diesen Zellen kann Code oder Text eingetragen werden. Anschließend werden die Zellen ausgeführt, wodurch die Ausgaben sofort sichtbar gemacht werden.
Durch Jupyter-Notebooks wird eine klare Dokumentation von Analysen bereitgestellt, damit Ergebnisse leicht reproduziert werden können. Außerdem werden sie wegen der interaktiven Struktur gern für Datenexploration, Entwicklung und Tests von Algorithmen verwendet und für Lernzwecke genutzt. Deshalb finden Jupyter-Notebooks weite Anwendung in der Datenwissenschaft, der Forschung, der Lehre und in zahlreichen technischen Fachgebieten.

Python wurde von Guido van Rossum entwickelt und 1991 veröffentlicht. Die Sprache wird als interpretierte High-Level-Sprache eingesetzt, wobei Einfachheit und Vielseitigkeit hervorgehoben werden. Die gute Lesbarkeit von Code und Ausdrücken wird besonders geschätzt, dadurch wird Python von Anfängern und erfahrenen Entwicklern gleichermaßen genutzt. Außerdem wird eine umfangreiche Standardbibliothek bereitgestellt, mit der ein großer Bereich

von Aufgaben bewältigt werden kann, von Webentwicklung und Automatisierung bis hin zu wissenschaftlicher Datenanalyse, maschinellem Lernen und künstlicher Intelligenz. Die Syntax betont die Lesbarkeit und wird durch einen einfachen, intuitiven Umgang mit Programmierkonstrukten geprägt.

9 Messung und Ergebnisse

In diesem Abschnitt werden die Ergebnisse mit dem selbst entworfenen CNN-Modell und dem vortrainierten VGG 16-Modell dargestellt. Zuerst werden die Modelle auf sauberen MRT-Daten trainiert und validiert. Danach wird geprüft, wie anfällig die Modelle für adversariale Angriffe ist (FGSM, BIM, PGD, score-based und decision-based Black-Box). Zum Schluss wird untersucht, ob durch adversariales Training und Median-Filter die Robustheit der Modelle verbessert werden kann.

Die Leistung wird mit mehreren Metriken bewertet, wie Genauigkeit, Konfusionsmatrix, Präzision, Sensitivität und F1-Score. In der Diskussion werden die wichtigsten Ergebnisse und Beobachtungen beschrieben und ihre Bedeutung für die Robustheit gegen adversariale Angriffe bei der Klassifikation medizinischer Bilder erklärt.

9.1 CNN Modelltraining und Bewertung anhand sauberer Daten

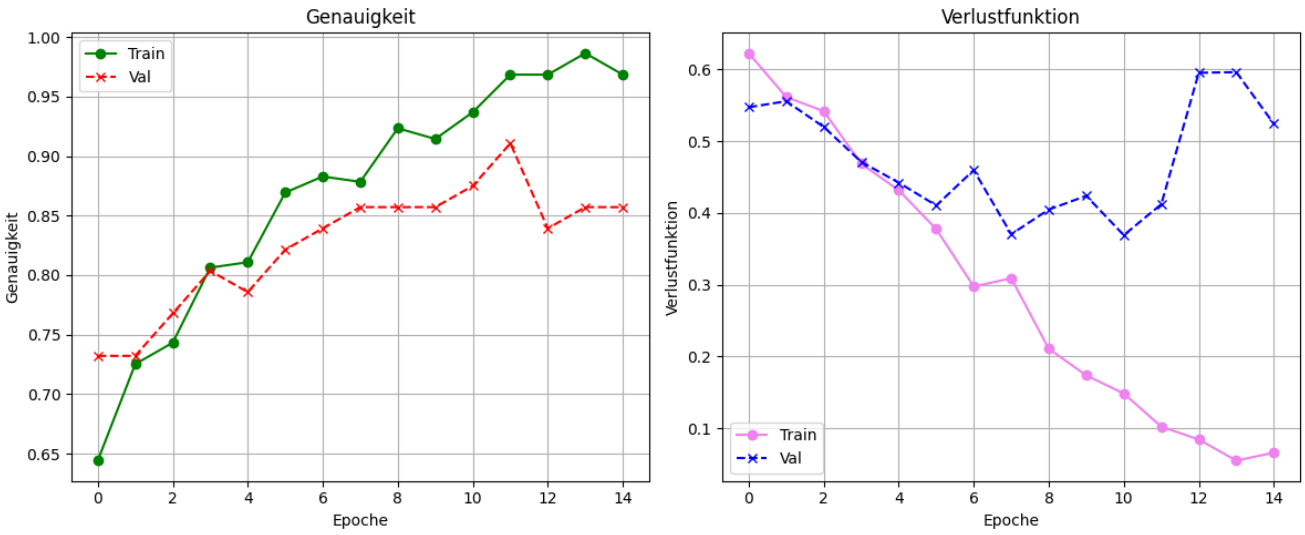

Abbildung 9.1: CNN-Bewertung

In Abbildung 9.1 ist zu sehen, dass die Trainingsgenauigkeit des selbst entworfenen CNN-Modells über 15 Epochen stetig steigt. Die Genauigkeit beginnt in Epoche 1 bei 62,62% und steigt nach und nach auf 96,67% in der letzten Epoche. Das Modell scheint demnach schnell aus den Trainingsdaten zu lernen. Die Validierungsgenauigkeit ist ebenfalls gut. Sie startet bei 73,21%, steigt bis etwa Epoche 12 auf den maximalen Wert von 91,07% und sinkt danach leicht bis sie den Wert von 85,71% in der letzten Epoche erreicht.

Der Trainingsverlust sinkt während des Trainings nach und nach. Von etwa 62,74% in der ersten Epoche auf rund 6,33% in der letzten. Das zeigt, dass das Modell die Trainingsdaten immer besser gelernt hat und sich im Verlauf des Trainings weiter verbessert. Der Validierungsverlust sinkt. Am Anfang liegt er bei 54,71%. In Epoche 11 erreicht er den Minimum 36,89%. Ab Epoche 12 steigt er wieder auf 41,2% bis er in der letzten Epoche den Wert von 52,41 erreicht. Das ist ein typisches Anzeichen für Overfitting. Diese Ergebnisse deuten darauf hin, dass das Modell robust und gut trainiert ist mit nur minimalen Anzeichen von Overfitting.

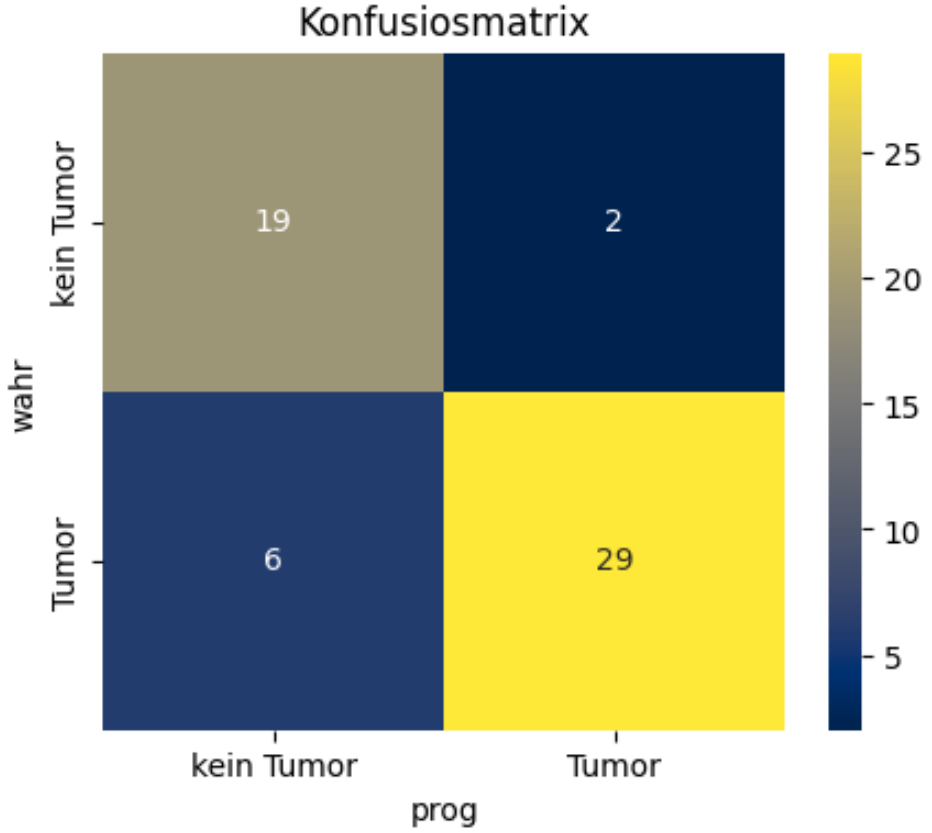

Abbildung 9.2: CNN-Konfusionsmatrix

In Abbildung 9.2 ist ersichtlich, dass das CNN-Modell MRT-Bilder mit und ohne Tumor gut unterscheiden kann. Von 21 Bildern ohne Tumor wurden 19 richtig als (kein Tumor) erkannt. Von 35 Bilder mit Tumor wurden 29 richtig als (Tumor) erkannt. Während 2 Bilder ohne Tumor wurden fälschlich als Tumor gemeldet (False Positives). Sechs Bilder mit Tumor wurden fälschlich als (kein Tumor) eingestuft (False Negatives).

Das Modell hat sechs falsch negative Ergebnisse. Das bedeutet, es hat in dieser Auswertung fast 17% der echten Tumorfälle nicht erkannt. In der Medizin ist das ein großes Problem, denn wenn Tumoren übersehen werden, beginnt die Behandlung zu spät oder findet gar nicht statt.

Klassifikationsreport:

	precision	recall	f1-score	support
kein Tumor	0.76	0.90	0.83	21
Tumor	0.94	0.83	0.88	35
accuracy			0.86	56
macro avg	0.85	0.87	0.85	56
weighted avg	0.87	0.86	0.86	56

Abbildung 9.3: CNN-Klassifikationsreport

Abbildung 9.3 demonstriert den Klassifikationsbericht. Das CNN-Modell arbeitet gut für Tumor und kein Tumor. Die Gesamtgenauigkeit ist $\frac{19+29}{56} \approx 86\%$. Für kein Tumor Klasse: Präzision

$\frac{19}{19+6} \approx 0{,}76$, Recall $\frac{19}{19+2} \approx 0{,}90$, F1-Score 0,83. Für Tumor Klasse: Präzision $\frac{29}{29+2} \approx 0{,}94$, Recall $\frac{29}{29+6} \approx 0{,}83$, F1-Score 0,88.

Die Mittelwerte liegen zwischen 0,85 und 0,87 und das Modell ist damit ausgewogen und zuverlässig,was wichtig in der Medizin ist.

9.2 VGG-16 Modelltraining und Bewertung anhand sauberer Daten

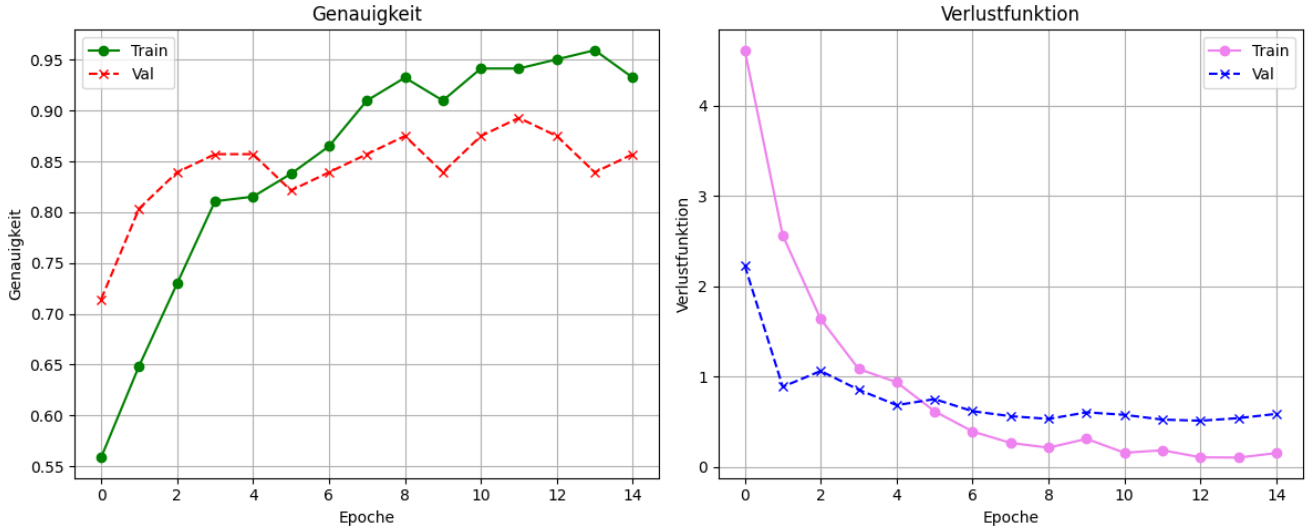

Abbildung 9.4: VGG16-Bewertung

Laut Abbildung 9.4 wird die Leistung des VGG-Modells in der Trainingsphase dargestellt. Zu Beginn wird in der ersten Epoche eine eher niedrige Genauigkeit von etwa 53,83% erreicht. Anschließend wird über insgesamt 15 Epochen eine stetige Steigerung der Trainingsgenauigkeit beobachtet. Außerdem wird der Fortschritt ohne größere Einbrüche verzeichnet, und die Werte werden von Epoche zu Epoche verbessert.
In Epoche 15 wurde eine Genauigkeit von 92,49% erreicht. Der stetige Anstieg zeigt, dass die vortrainierten Convolutional-Layer erfolgreich feinjustiert und gut an die Klassifikation von Gehirn-MRTs angepasst wurden.

Die Validierungsgenauigkeit lag in Epoche 1 bei 71,43% und steigt in Epoche 12 auf 89,29% (der höchste Wert) und schwankt ab Epoche 13 bis sie den Wert von 85,71% erreicht. Die geringe Streuung zeigt, dass das Modell gut auf neue Daten überträgt und kein starkes Overfitting vorliegt.
Der Trainingsverlust sank der in 15 Epochen deutlich, von etwa 4,99 in der ersten Epoche auf 0,16 in der letzten Epoche.
Der Validierungsverlust sank über 15 Epochen von $\approx 2{,}23$ auf 0,58.
Insgesamt deutet der Verlauf auf eine wirksame Optimierung hin, ohne starkes Overfitting.

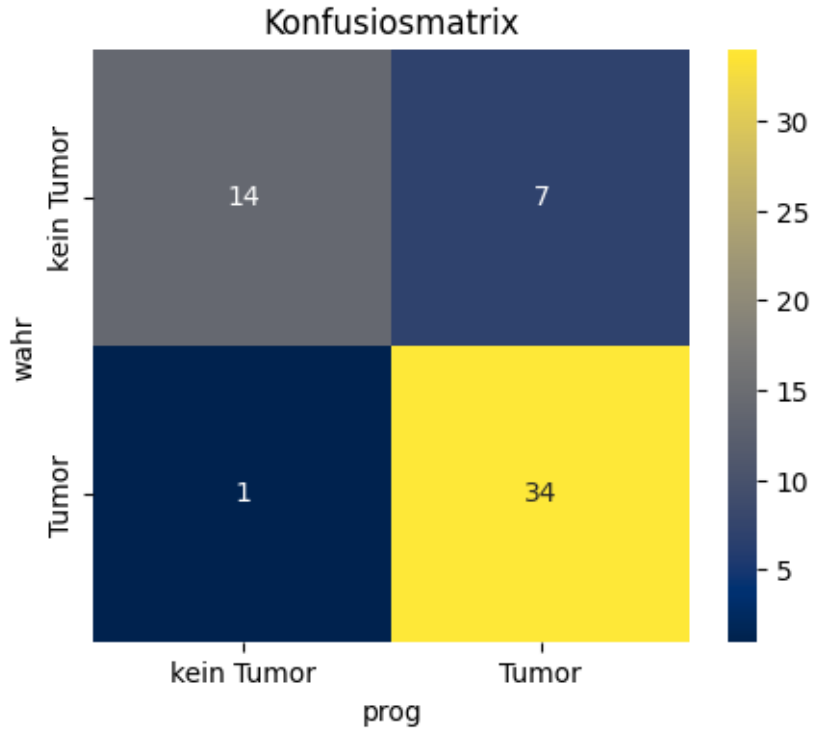

Abbildung 9.5: VGG16-Konfusionsmatrix

Basierend auf Abbildung 9.5 erkennt das Modell 34 echte Tumorfälle korrekt (True Positives) und klassifiziert 14 gesunde Fälle richtig als (kein Tumor) (True Negatives). Allerdings werden 7 gesunde Bilder fälschlich als Tumor eingestuft (False Positives), das weist auf eine mäßige Überdiagnoserate hin. Das Modell ist eher vorsichtig und neigt dazu, zur Sicherheit (Tumor) zu sagen. Das erhöht die Sensitivität, kann aber zu unnötigen weiteren Untersuchungen führen. Auf der anderen Seite verpasst das Modell einen tatsächlichen Tumorfall (False Negatives). Das ist in der klinischen Praxis kritischer, da ein übersehener Tumor die Diagnose und Behandlung verzögern kann.

Klassifikationsreport (Validation):

	precision	recall	f1-score	support
kein Tumor	0.93	0.67	0.78	21
Tumor	0.83	0.97	0.89	35
accuracy			0.86	56
macro avg	0.88	0.82	0.84	56
weighted avg	0.87	0.86	0.85	56

Abbildung 9.6: VGG16-Klassifikationsreport

Wie in Abbildung 9.6 zu erkennen, dass durch den Klassifikationsbericht des VGG16-Modells wichtige Hinweise zur Leistung auf dem Brain-MRI Datensatz gegeben werden. Die Gesamtgenauigkeit wird mit 86% angegeben. Damit werden 86% aller Validierungsbeispiele richtig klassifiziert.

Für kein Tumor Klasse liegen Präzision= 0,8, Recall = 0,76 und F1-Score = 0,78. d.h. einige Gesunde werden unnötig weiter untersucht (Fehlalarm). Auf der anderen Seite für Tumor Bilder liegen Präzision $= 14/(14+1) \approx 0{,}93$, Recall $= 14/(14+7) \approx 0{,}67$ und F1-Score = 0,78. Die Leistung für (Tumor) ist insgesamt gut und ausgewogen, denn Modell findet die meisten Tumoren und macht vergleichsweise wenige Fehlalarme.

9.3 Testen der beiden Modellen

In dieser Phase wurden CNN und VGG16 mit einer zufällig ausgewählten Gruppe von MRT-Bildern mit Hirntumoren getestet. So wurde die Genauigkeit geprüft. Danach wurde das Modell mit adversarialen Beispielen weiterverwendet.

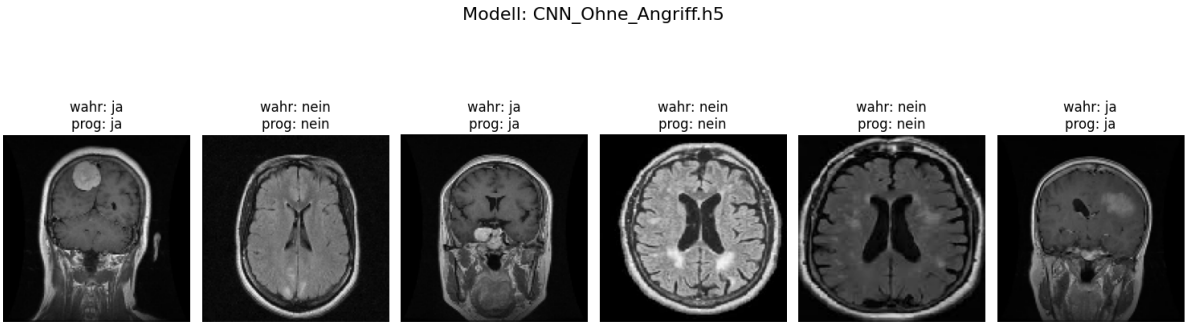

Abbildung 9.7: CNN-Modell Test

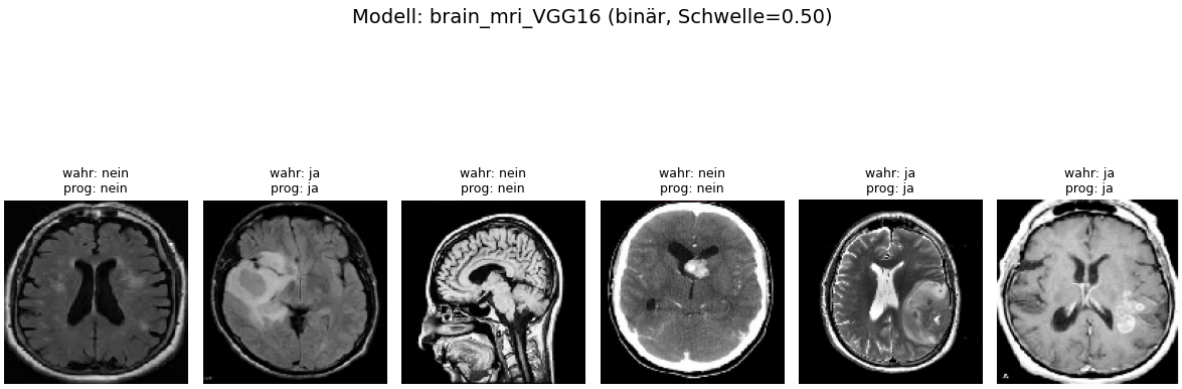

Abbildung 9.8: VGG16-Modell Test

Tabelle 9.1: Leistungsvergleich, CNN vs. VGG-16 (saubere Daten, Klasse: Tumor)

	Genauigkeit	**Präzision**	**recall**	**F1-score**
CNN	0,86	0,94	0,83	0,88
VGG-16	0,86	0,83	0,97	0,89

Sowhl Abbildungen 9.7 und 9.8 als auch Tabellen 9.1 und 9.2 illustrieren, dass sowohl CNN-Modell als auch VGG-16 Modell eine hohe Genauigkeit erricht haben Zusammenfassend erreichten der beiden Modellen eine hohe Genauigkeit. CNN erkannte 76% der Bilder ohne Tumor richtig und 94% der Tumor Bilder. VGG-16 dagegen erkannte 93% der Bilder ohne Tumor und 83% der Bilder mit Tumor. Diese ausgewogene Leistung ist in der Medizin wichtig, weil

Tabelle 9.2: Leistungsvergleich, CNN vs. VGG-16 (saubere Daten, Klasse: Kein Tumor)

	Genauigkeit	**Präzision**	**recall**	**F1-score**
CNN	0,86	0,76	0,90	0,83
VGG-16	0,86	0,93	0,67	0,87

sowohl Fehlalarme als auch übersehene Fälle vermieden werden müssen. Um die Arbeit übersichtlich zu halten, wird ein einzelnes Modell verwendet, auf das Angriffe angewandt werden; robuste Modelle werden anschließend untersucht.

9.4 Adversariale Angriffe auf das CNN-Modell

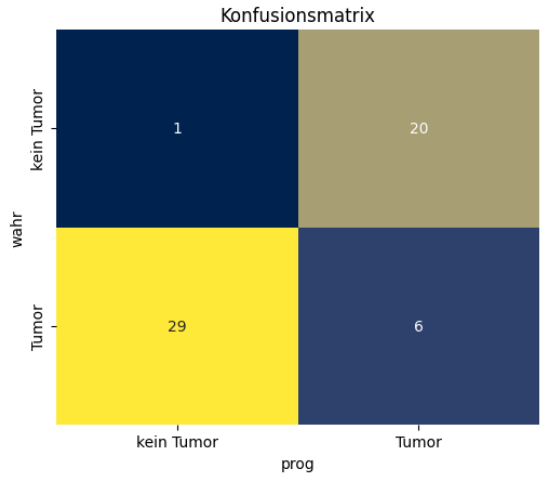

Abbildung 9.9: Konfusionsmatrix für angegeriffenes CNN-Modell

```
=== CNN-Modell - ADVERSARIAL ===
Accuracy: 0.125
Klassifikationsbericht:
              precision    recall  f1-score   support

        nein       0.03      0.05      0.04        21
          ja       0.23      0.17      0.20        35

    accuracy                           0.12        56
   macro avg       0.13      0.11      0.12        56
weighted avg       0.16      0.12      0.14        56
```

Abbildung 9.10: Klassifikationsbericht für angegeriffenes CNN-Modell

Wie in Abbildungen 9.9 und 9.10 unterstreicht, dass die Gesamtgenauigkeit des angegriffenem CNN-Modells mit etwa 12,5% erreicht ((6 + 1)/56). Die Trefferquote für Tumor (Recall) wird mit rund 17% erzielt (6/(6 + 29)); dadurch wird gezeigt, dass viele Tumoren übersehen werden. Die Präzision für Tumor wird mit ungefähr 23% erreicht (6/(6 + 20)), also werden unter den als Tumor gemeldeten Bildern viele Fehlalarme erzeugt. Insgesamt wird durch den Angriff eine deutliche Verschlechterung verursacht. Es werden viele echte Tumoren verpasst (viele falsche Negative) und zugleich häufig fälschlich Tumor gemeldet (falsche Positive). Damit ist ersichtlich, dass das Modell unter Angriff nicht robust ist.

9.4.1 Fast Gradient Sign Methode (FGSM)

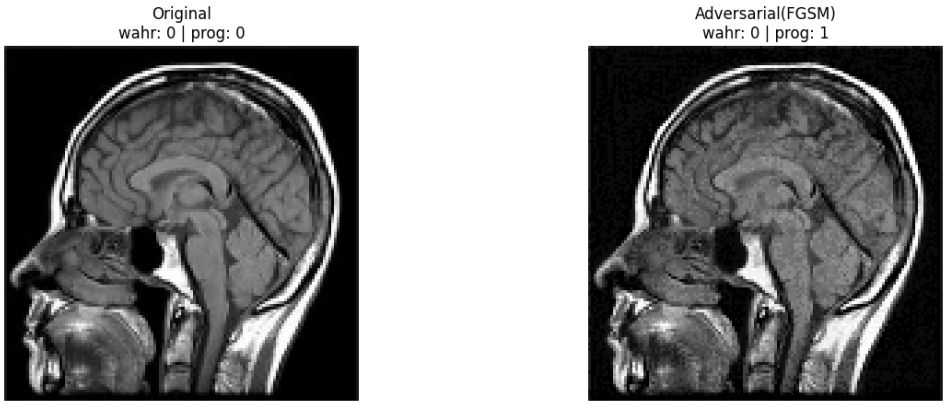

Abbildung 9.11: FGSM auf gesundes MRT-Bild

In Abbildung 9.11 werden die Folgen eines FGSM-Angriffs auf ein MRT-Bild eines gesunden Gehirns präsentiert. Zunächst wurde das Bild vom CNN-Modell korrekt als (0 = kein Tumor) erkannt. Nach dem Anwenden einer kleinen Störung durch FGSM wurde das gleiche Bild jedoch fälschlich als (1 = Tumor) eingestuft. Das Modell scheint also durch solche Angriffe getäuscht werden zu können, sodass falsche Entscheidungen getroffen werden und insgesamt schlechtere Ergebnisse entstehen.

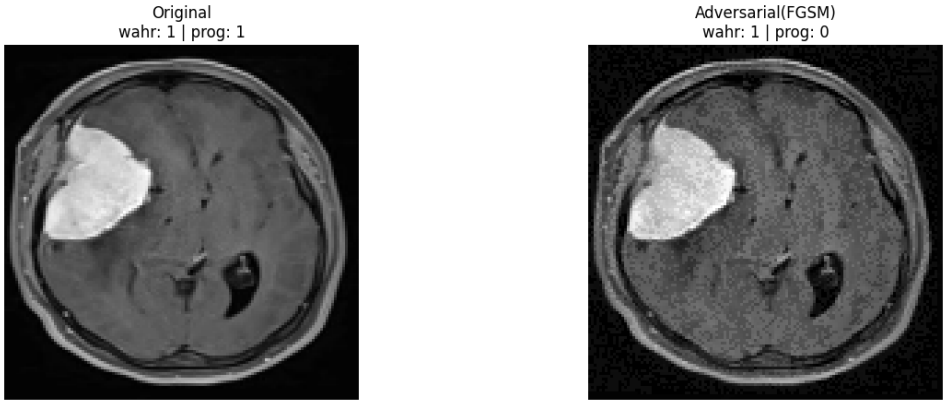

Abbildung 9.12: FGSM auf MRT-Bild mit Hirntumor

In Abbildung 9.12 werden die Effekte des FGSM-Angriffs auf ein Bild mit Hirntumor gezeigt, das ursprünglich als klassifiziert worden war. Vor der Störung wurde das Bild vom CNN-Modell korrekt erkannt. Nach der Anwendung der FGSM-Methode wurde es jedoch fälschlicherweise als klassifiziert. Das Modell ist demnach überaus anfällig gegenüber kleinen Störungen. Selbst geringe Änderungen im Eingabebild können die Zuverlässigkeit der Vorhersage deutlich beeinträchtigen, was besonders bei medizinischen Diagnosen als problematisch angesehen wird.

9.4.2 Basic Iterative Methode (BIM)

In Abbildung 9.13 wird hervorgehoben, dass das Modell durch den BIM-Angriff getäuscht wurde. Ein Bild ohne Tumor wurde zunächst korrekt als gesund erkannt. Nach kleinen Verän-

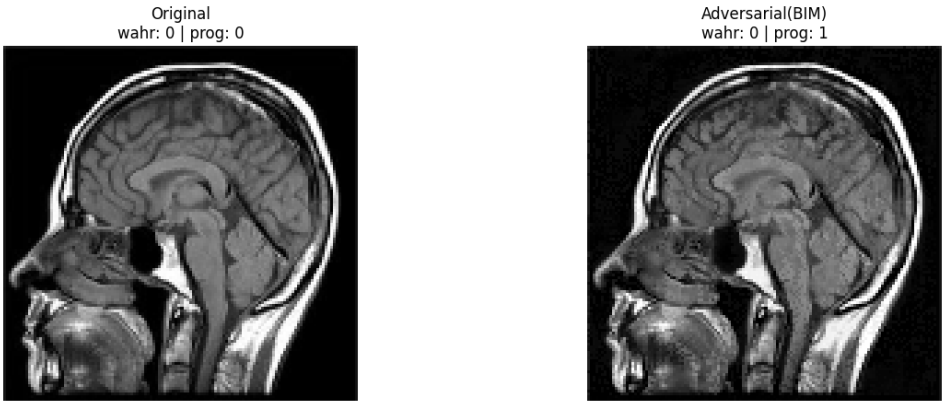

Abbildung 9.13: BIM auf gesundes MRT-Bild

derungen am Bild wurde es jedoch fälschlich als tumorös eingestuft. Obwohl die Änderungen sehr klein waren, wurde die Entscheidung des Modells stark beeinflusst. Damit wird deutlich, dass die Zuverlässigkeit des Modells durch den BIM-Angriff stark beeinträchtigt werden kann, sogar bei eigentlich klaren Bildern. In der medizinischen Praxis könnten dadurch unnötige Behandlungen ausgelöst und Patientinnen und Patienten verunsichert werden. Deshalb wird betont, dass KI-Modelle in der Medizin gezielt robuster und sicherer gemacht werden sollten.

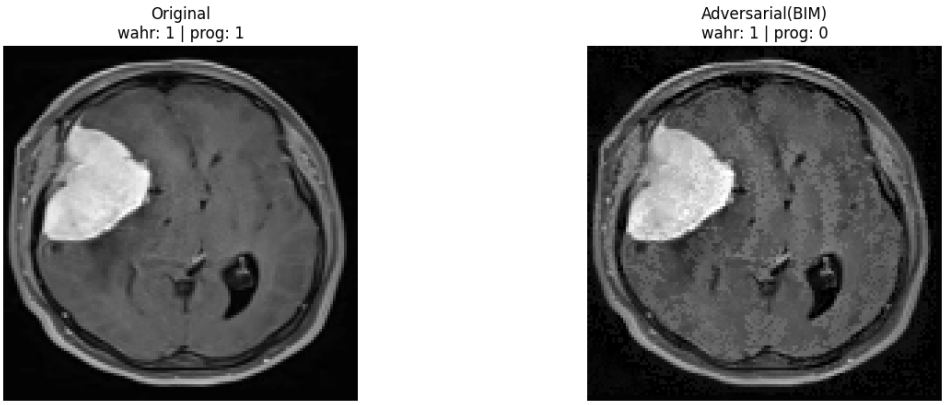

Abbildung 9.14: BIM auf MRT-Bild mit Hirntumor

In Abbildung 9.14 wird ein weiteres Beispiel für den BIM-Angriff gezeigt. Ein MRT-Bild mit Tumor wurde zuerst vom CNN-Modell richtig als (Tumor) erkannt. Nach den gezielten Veränderungen wurde es jedoch fälschlich als (kein Tumor) eingestuft. Die Sicherheit der Entscheidung wurde verringert, obwohl am Bild mit dem bloßen Auge keine Änderung gesehen wurde. Damit wird gezeigt, dass das Modell gegen solches künstlich erzeugtes Rauschen anfällig ist. Besonders in der Medizin, wo falsch-negative Diagnosen schwere Folgen haben können, wird die Notwendigkeit robuster und sicherer Modelle betont.

9.4.3 Projected Gradient Descent (PGD)

In Abbildung 9.15 wird klar, dass ein Bild, das zuerst korrekt als gesund erkannt wurde, mit PGD-Angriff gezielt verändert wurde. Nach dieser Manipulation wurde es vom Modell fälschlicherweise als tumorös eingestuft und ein Fehlalarm (False Positive) ausgelöst. Verursacht

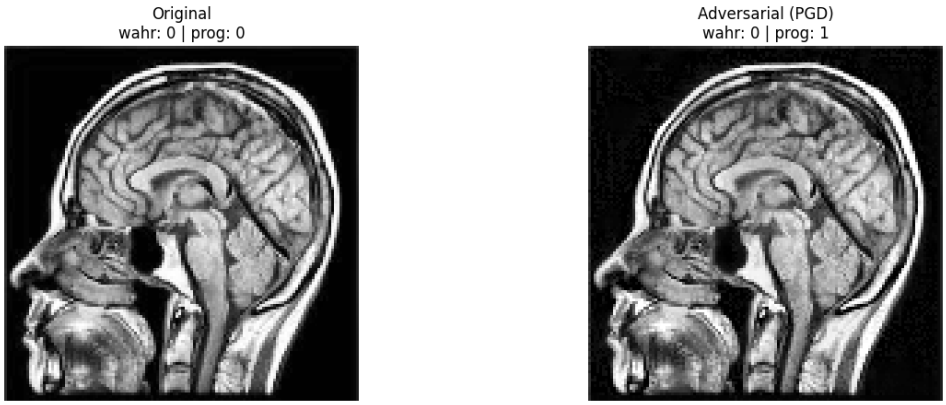

Abbildung 9.15: PGD auf gesundes MRT-Bild

wurde dies durch viele kleine, kaum sichtbare Änderungen. Damit wird deutlich, wie empfindlich das Modell auf solche Störungen reagiert. In der medizinischen Diagnostik können dadurch unnötige Behandlungen ausgelöst werden.

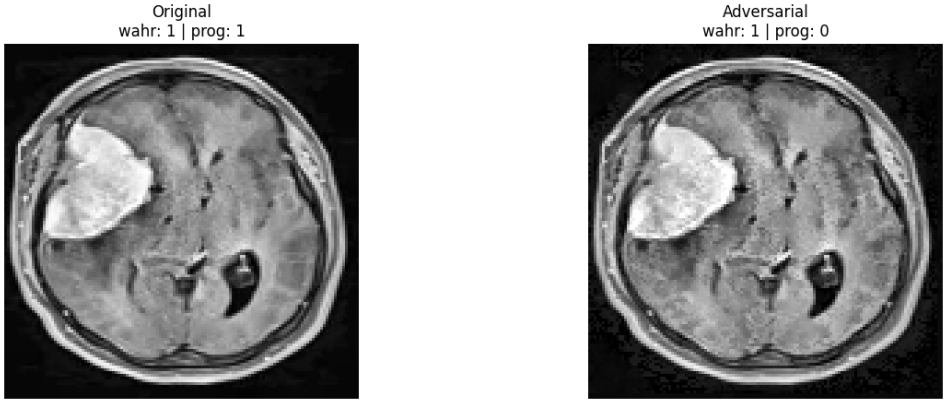

Abbildung 9.16: PGD auf MRT-Bild mit Hirntumor

In Abbildung 9.16 wird ein Beispiel gezeigt. Ein Bild mit Tumor wurde zuerst vom CNN-Modell richtig erkannt. Nach dem PGD-Angriff wurde es fälschlich als gesund eingestuft. Damit wird gezeigt, dass PGD sehr stark ist. Viele kleine, gezielt gesetzte Änderungen werden gemacht; sie sind mit dem Auge kaum zu sehen. Durch diese Änderungen wird das Modell leicht getäuscht. Im Vergleich zu FGSM wird PGD in mehreren Schritten angewendet und gilt als stärker. So wird deutlich, dass das Modell anfällig ist. In der medizinischen Diagnostik können dadurch falsche Einschätzungen entstehen, und es können falsche Entscheidungen mit ernsten Folgen getroffen werden.

9.4.4 Entscheidungsbasierter Black-Box Angriff

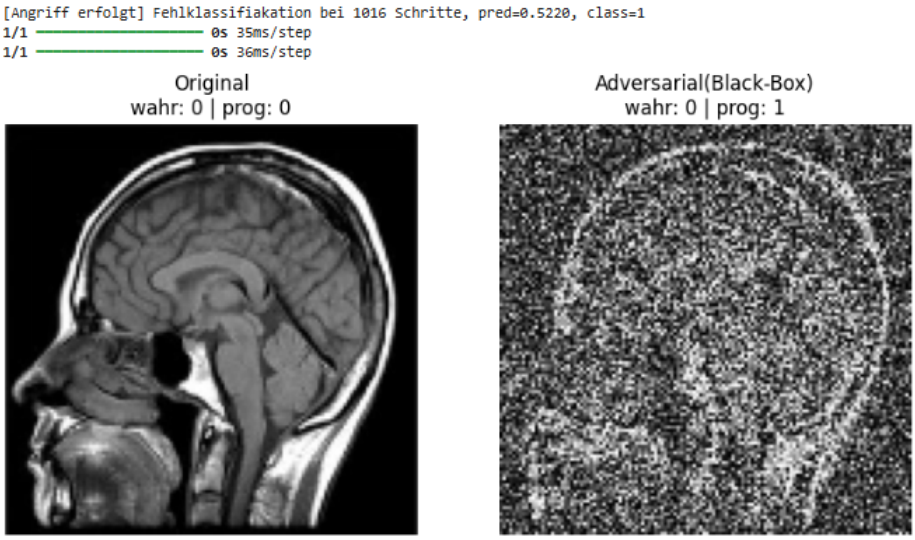

Abbildung 9.17: Entscheidungsbasierter Black-Box Angriff auf gesundes MRT-Bild

In Abbildung 9.17 wird sichtbar, dass das ursprüngliche Eingabebild zur Klasse (kein Tumor) gehört und wird vom Modell zunächst korrekt als (kein Tumor) eingestuft. Nach dem Anwenden eines entscheidungsbasierten Black-Box Angriffs werden kleine zufällige Störungen schrittweise hinzugefügt, wodurch der Klassifikator allmählich in die Irre geführt wird, bis (Tumor) vorhergesagt wird. Bis zur Fehlklassifikation waren insgesamt 1016 Abfragen an das Modell erforderlich.
Dieses Ergebnis wird als falsch positiv bezeichnet. Im medizinischen Kontext können durch einen solchen Fehler unnötiger Stress und zusätzliche Behandlungskosten für gesunde Patientinnen und Patienten verursacht werden.

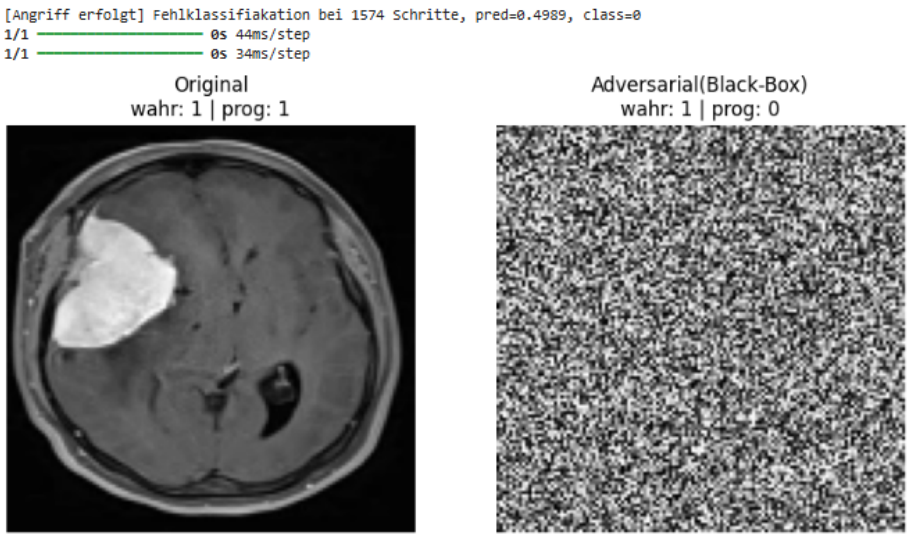

Abbildung 9.18: Entscheidungsbasierter Black-Box Angriff auf MRT-Bild mit Hirntumor

Abbildung 9.18 verdeutlicht, dass ein Bild der Klasse (Tumor) vom Modell zunächst korrekt als (Tumor) vorhergesagt wird. Nach dem Angriff wird das adversarial veränderte Bild falsch

als (kein Tumor) eingestuft. Erst nach 1574 Abfragen kam es zur Fehlklassifikation. Dieser Fall wird als falsch negativ bezeichnet und gilt in medizinischen Anwendungen als besonders gefährlich. Im Unterschied zum vorherigen Fall können dadurch Diagnosen übersehen, Behandlungen verzögert und schwere Folgen für die Gesundheit verursacht werden.

9.4.5 Score-basierter Black-Box Angriff

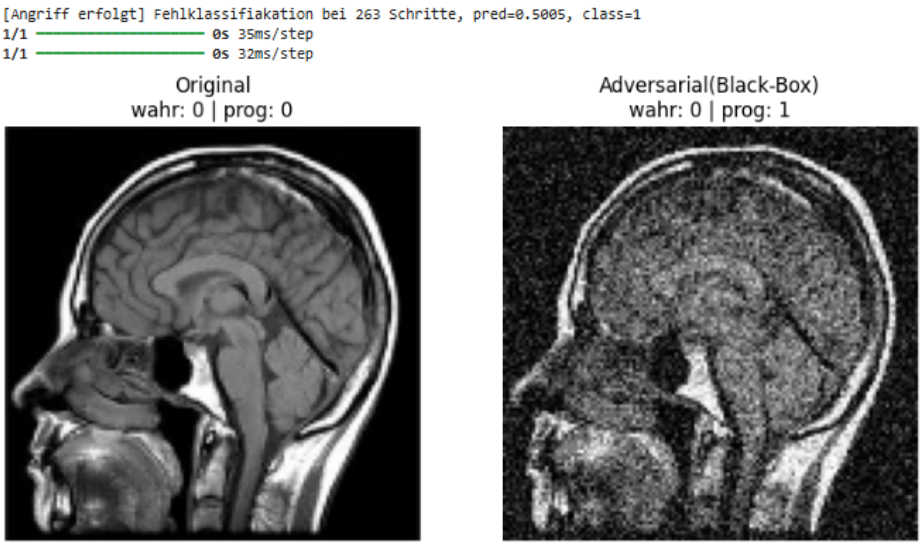

Abbildung 9.19: Score-basierter Black-Box Angriff auf gesundes MRT-Bild

In Abbildung 9.19 wird die Wirkung eines score-basierten Black-Box-Angriffs auf ein gesundes MRT-Bild gezeigt. Das Modell klassifiziert das Bild zunächst korrekt; nach dem Angriff kommt es jedoch zu einer Fehlklassifikation. Im Verlauf von 263 Abfragen wurde die Fehlentscheidung erzielt. Der Vergleich mit dem entscheidungsbasierten Angriff verdeutlicht, dass Letzterer deutlich mehr Abfragen benötigt.

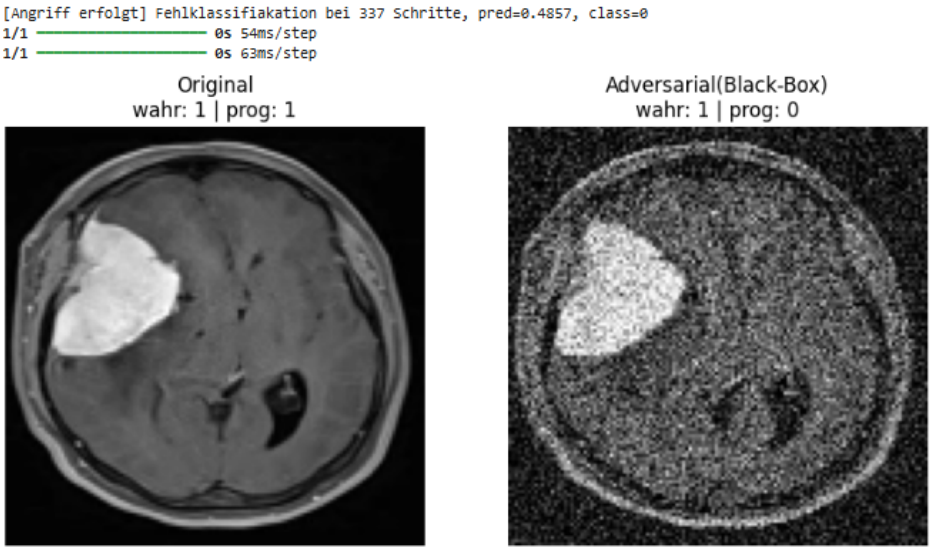

Abbildung 9.20: Score-basierter Black-Box Angriff auf MRT-Bild mit Hirntumor

Abbildung 9.20 präsentiert ein weiteres Beispiel mit einem MRT-Bild mit Hirntumor. Auch hier ist die Vorklassifikation korrekt; erst nach 337 Modellabfragen führte der score-basierte Angriff zur ersten Fehlklassifikation.

Tabelle 9.3: Entscheidungsbasiert vs. Score-basiert für Klasse: Tumor

Angriffstyp	**Epsilon (ε)**	**Abfragezahl**
Entscheidungsbasiert	0,1	1574
Score-basiert	0,02	337

Tabelle 9.4: Entscheidungsbasiert vs. Score-basiert für Klasse: Kein Tumor

Angriffstyp	**Epsilon (ε)**	**Abfragezahl**
Entscheidungsbasiert	0,02	1016
Score-basiert	0,02	263

Tabellen 9.3 und 9.4 zeigen score-basierte Black-Box Angriffe eine wesentlich geringere Abfragezahl bis zur erzwungenen Fehlklassifikation als entscheidungsbasierte, was wichtig für die Robustheitsbewertung medizinischer KI-Systeme ist.

9.5 Aufbau robuster Modelle

9.5.1 Aufbau eines Robusten Modells gegen White-Box Angriffe

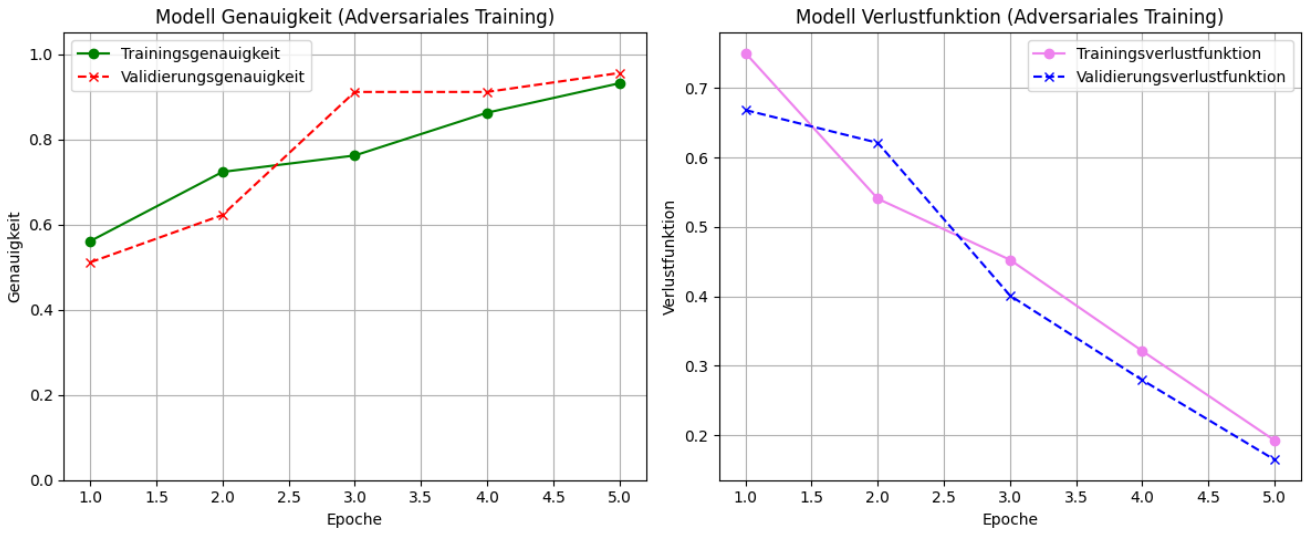

Abbildung 9.21: Bewertung des rubusten Modells durch Adversariales Training

In Abbildung 9.21 wird gezeigt, dass die Genauigkeit des adversarial trainierten Modells über fünf Epochen stetig gestiegen ist. Am Anfang wurde eine Trainingsgenauigkeit von 56,09% erreicht; bis zur letzten Epoche stieg sie schnell auf 93,14%. Die Validierungsgenauigkeit lag zu Beginn bei 51,11% und erreichte in der letzten Epoche 95,56%. Damit wird deutlich, dass das Modell auch mit einer Mischung aus sauberen und adversarialen Beispielen gut umgehen konnte. Die sehr hohe Validierungsgenauigkeit zeigt, dass das adversariale Training die Robustheit deutlich verbessert hat und zuvor schwierige Fälle korrekt erkannt wurden.

Neben der Genauigkeit haben sich auch die Verluste während des Trainings deutlich verringert. Der Trainingsverlust wurde von 75,06% in der ersten Epoche auf 19,23% in der fünften Epoche gesenkt; damit wird ein effizientes Lernen und eine gute Konvergenz des Modells bestätigt. Auch der Validierungsverlust wurde stark reduziert, nämlich von 66,85% auf 16,45%. Dadurch wird gezeigt, dass die Vorhersagen sicherer und stabiler gemacht wurden, nicht nur genauer.

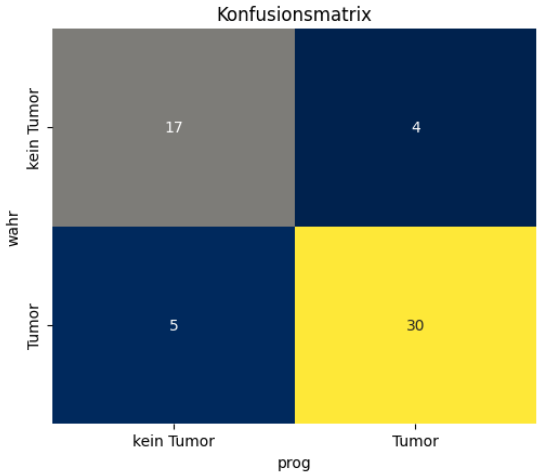

Abbildung 9.22: Konfusionsmatrix für robustes CNN-Modell

```
=== Robustes Modell – ADVERSARIAL ===
Accuracy: 0.839
Klassifikationsbericht:
              precision    recall  f1-score   support

        nein       0.77      0.81      0.79        21
          ja       0.88      0.86      0.87        35

    accuracy                           0.84        56
   macro avg       0.83      0.83      0.83        56
weighted avg       0.84      0.84      0.84        56
```

Abbildung 9.23: Klassifikationsbericht für robustes CNN-Modell

Wie in Abbildungen 9.22 und 9.23 dargestellt wird, die Gesamtgenauigkeit des robusten CNN-Modells erreicht mit etwa 84% ((17+30)/56). Die Trefferquote für Tumor(Recall) wird mit rund 86% erzielt (30/(30 + 5)); dadurch wird gezeigt, dass das adversariale Training wirksam eingesetzt wurde, um die Auswirkungen von Angriffen zu verringern und die Zuverlässigkeit des Modells zu steigern, besonders wichtig bei sensiblen Aufgaben wie der Klassifikation von Gehirntumoren.

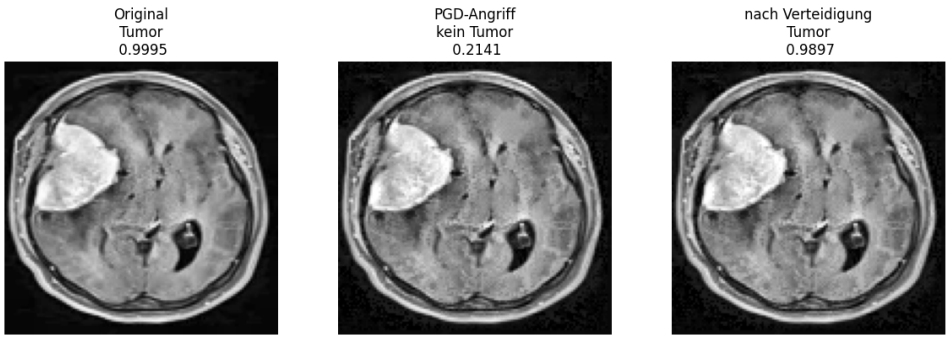

Abbildung 9.24: Robustes Modell auf MRT-Bild mit Hirntumor

Abbildung 9.24 zeigt, dass das ursprünglich trainierte Modell (ohne adversariales Training) auf einem sauberen Bild den Tumor mit 99,95% erkannte. Es funktioniert also gut, wenn die Bilder nicht verändert sind. Wird dieselbe Aufnahme jedoch mit kleinen Angriffen, wie beispielsweise PGD Angriff, manipuliert, fällt die Vorhersage auf etwa 21,41% und das Bild wird falsch als (kein Tumor) eingestuft. Das zeigt eine starke Anfälligkeit gegenüber solchen Störungen.
Das robust trainierte Modell (mit adversarialem Training) zeigt unter denselben Angriffen eine deutlich höhere Robustheit. Für das manipulierte Bild gibt es 98,97% aus und erkennt den Tumor korrekt. Damit wird gezeigt, dass adversariales Training die Widerstandsfähigkeit und die

Zuverlässigkeit des Modells klar verbessert, besonders wichtig für die medizinische Diagnose von Gehirntumoren.

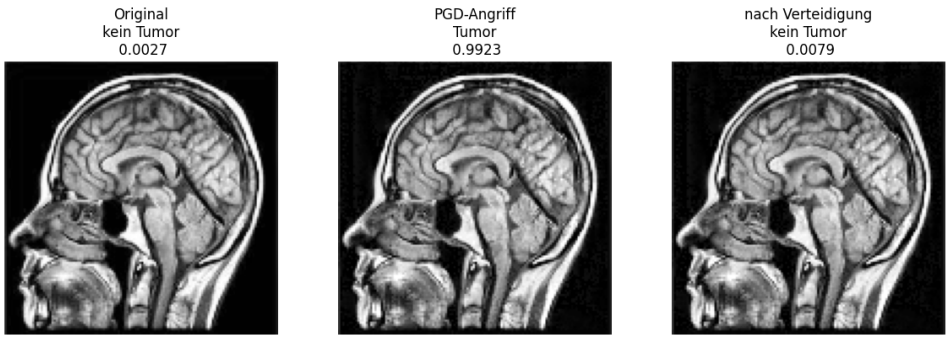

Abbildung 9.25: Robustes Modell auf gesundes MRT-Bild

Abbildung 9.25 belegt die Wirkung von adversarialem Training auf gesundes MRT-Bild, vor dem Angriff wurde beim ursprünglichen Modell ein sauberes Bild ohne Tumor mit einer sehr kleinen Wahrscheinlichkeit bewertet 0,27% und richtig als (kein Tumor) eingestuft. Nach einem adversarialen Angriff auf dasselbe Bild stieg die Vorhersage jedoch auf 99,23%. Dadurch wurde das Bild falsch als (Tumor) klassifiziert. Das macht deutlich, wie stark solche Störungen das Modell beeinflussen und wie anfällig es für Fehlalarme ist.
Das robust trainierte Modell mit adversarialem Training erkannte das Bild auch unter Angriff korrekt als (kein Tumor) und gab 0,79% aus. Dieses Ergebnis zeigt, dass adversariales Training die Robustheit erhöht und Fehlentscheidungen vermeidet, ein wichtiger Punkt in der medizinischen Bildgebung, weil falsche Diagnosen ernste Folgen für Patientinnen und Patienten haben können.

9.5.2 Aufbau eines Robusten Modells gegen Black-Box Angriffe

Abbildung 9.26: Robustes Modell auf gesundes MRT-Bild

Wie in Abbildung 9.26 gezeigt wird, wurde vor dem Angriff korrekt (kein Tumor) vorhergesagt. Nach dem Angriff wurden durch die adversarialen Störungen Fehlklassifikationen als (Tumor) verursacht, also ein Fehlalarm (falsches Positiv). Nach der Anwendung der Vorverarbeitung mit CLAHE + Medianfilter wurde die meiste Störung entfernt, und in vielen Fällen kehrte die Vorhersage des Modells zu (kein Tumor) zurück. Zusammenfassend lässt sich sagen, dass durch die Verteidigung unnötige Fehlalarme verringert werden und gesunde Patientinnen und Patienten nicht fälschlich für weitere Untersuchungen markiert werden.

Abbildung 9.27: Robustes Modell auf MRT-Bild mit Hirntumor

Wie in Abbildung 9.27 zu sehen, wurde vor dem Angriff der Tumor korrekt vorhergesagt. Nach dem Angriff wurde das Modell durch adversariales Rauschen dazu gebracht, (kein Tumor) zu prognostizieren; dadurch entstand ein Falschnegativ. Nach der Anwendung der Vorverarbeitungsstrategie wurden wichtige Tumormerkmale wiederhergestellt, sodass in mehreren Fällen erneut (Tumor) vorhergesagt wurde. Zusammenfassend ist dieser Schritt entscheidend, um übersehene Diagnosen zu verhindern, die gefährlicher als falsch positive Befunde sind.

10 Fazit und Ausblick

Diese Arbeit zeigt, dass KI-Modelle zur Tumorerkennung auf sauberen Daten gut funktionieren, aber gegenüber adversarialen Störungen deutlich anfällig sind. Unser CNN erreicht ohne Angriffe 86% Genauigkeit. Die Klasse Tumor wird sehr zuverlässig erkannt (F1 = 0,88), während kein Tumor schwächer ausfällt (F1 = 0,83). Diese Unterschiede passen zur Klassenverteilung und sind klinisch wichtig, weil Fehlalarme und übersehene Befunde unterschiedliche Folgen haben.

Unter adversarialen Angriffen (FGSM, BIM, PGD, entscheidungsbasierter und score-basierter Angriff)) fällt die Leistung des Basismodells stark ab. Die Genauigkeit sinkt auf 12,5% . Alle Angriffe erreichen ihr Ziel und verursachen Fehlklassifikationen. Das zeigt deutlich, dass Nur auf Genauigkeit auf sauberen Daten zu achten, für Sicherheit nicht ausreicht.

Adversariales Training macht das Modell robuster. Unter Angriffen erreicht das robuste Modell 84% Genauigkeit. Die Tumorerkennung bleibt hoch (Recall = 0,86, F1 = 0,87). Gleichzeitig sinkt der Recall für kein Tumor auf 0,81. Das bedeutet mehr Fehlalarme. In der Klinik ist dieser Tausch oft okay, weil übersehene Tumoren (False Negatives) schlimmer sind als Fehlalarme (False Positives). Bei Black-Box-Angriffen helfen Medianfilter und CLAHE als Vorverarbeitung. Sie machen die Störungen kleiner und die Vorhersagen stabiler. Aber, Je nach Einstellungen können sich Bilddetails verändern.

In Zukunft können stärkere und anpassbare Schutzmethoden getestet werden. Dabei werden besispielsweise Feature-Denoising Netzwerke, zertifizierte Verteidigungen und Ensemble Adversarial Training eingesetzt. Die Robustheit sollte mit verschiedenen Datensätzen und Bildarten geprüft werden, damit können die Ergebnisse besser verallgemeinert werden.

Als nächster Schritt kann erklärbare KI (XAI) eingesetzt werden. So können Modellentscheidungen besser verstanden und Vertrauen in der Klinik aufgebaut werden.

Wichtig ist auch, dass Angriffe und Verteidigungen in Echtzeit Systemen angewendet und ihre Wirkungen im praktischen Einsatz geprüft werden. Dieses Thema bleibt für weitere Forschung offen.

11 Zusammenfassung

In dieser Arbeit wird die binäre Klassifikation von Gehirn-MRT-Bildern untersucht: Tumor oder kein Tumor. Dafür wurden ein eigenes CNN und VGG-16 benutzt. Erste Tests mit sauberen Daten zeigten gute und ausgewogene Ergebnisse. Bei adversarialen Angriffen wurde die Leistung jedoch stark schlechter.

Die Angriffe waren White-Box (FGSM, BIM, PGD) und Black-Box (entscheidungsbasierter und score-basierter Angriff). Zur Verteidigung wurde adversariales Training gegen White-Box Angriffe eingesetzt. Gegen Black-Box Angriffe wurden CLAHE und der Medianfilter als Vorverarbeitung verwendet.

So wurde ein robusteres Modell gebaut, das saubere und manipulierte Bilder nutzt. Dadurch stieg die Robustheit und es gab weniger Fehlklassifikationen. Das zeigt, dass Adversariales Training in der medizinischen Bildgebung wirksam ist.

Eine große Herausforderung war es, das gegnerische Rauschen vor der Klassifikation zu entfernen. Die Filter-Methode sollte manipulierte Bilder bereinigen und die richtigen Labels zurückholen, die Methode brachte nicht die erhoffte Verbesserung bei der Verteidigung gegen White-Box Angriffe. Wichtige Tumor Merkmale wurden oft unscharf. Bei Black-Box-Angriffen hat die Verteidigung trotzdem funktioniert, bei White-Box-Angriffen jedoch nicht.

Literaturverzeichnis

[1] G. P. Emmanuel Chris Anita Johnson. „Deep Learning vs. Traditional Machine Learning: Key Differences.“ (2024), Adresse: https://www.researchgate.net/publication/389991583_Deep_Learning_vs_Traditional_Machine_Learning_Key_Differences.

[2] C. A. Shrey Srivastava* Amit Vishvas Divekar. „Comparative analysis of deep learning image detection algorithms.“ (2024), Adresse: https://journalofbigdata.springeropen.com/articles/10.1186/s40537-021-00434-w.

[3] N. Y. Rammah Yousef Gaurav Gupta1. „A holistic overview of deep learning approach in medical imaging.“ (2022), Adresse: https://www.researchgate.net/publication/358008673_A_holistic_overview_of_deep_learning_approach_in_medical_imaging.

[4] CEOsBay. „An overview of the supervised machine learning methods.“ (2017), Adresse: https://www.researchgate.net/profile/Vladimir-Nasteski/publication/328146111_An_overview_of_the_supervised_machine_learning_methods/links/5c1025194585157ac1bba147/An-overview-of-the-supervised-machine-learning-methods.pdf.

[5] G. G. Ahmad Waleed Salehi Shakir Khan. „A Study of CNN and Transfer Learning in Medical Imaging: Advantages, Challenges, Future Scope.“ (2023), Adresse: https://www.mdpi.com/2071-1050/15/7/5930.

[6] H. K. JHossein Gholamalinezhad. „Pooling Methods in Deep Neural Networks, a Review.“ (), Adresse: https://arxiv.org/pdf/2009.07485.

[7] D. D. D. P. S. Nurshazlyn Mohd Aszemi. „Hyperparameter Optimization in Convolutional Neural Network using Genetic Algorithms.“ (2019), Adresse: https://www.researchgate.net/publication/334151021_Hyperparameter_Optimization_in_Convolutional_Neural_Network_using_Genetic_Algorithms.

[8] „Convolutional Kernel Networks for Graph-structured Data.“ (2020), Adresse: https://www.researchgate.net/publication/339873203_Convolutional_Kernel_Networks_for_Graph-Structured_Data.

[9] P.-C. H. FAHAD ALRASHEEDI XIN ZHONG. „Padding Module: Learning the Padding in Deep Neural Networks.“ (2023), Adresse: https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=10021573.

[10] O. M. Luiz Zaniolo1. „Ontheuseofvariablestride in convolutional neural networks.“ (2020), Adresse: https://link.springer.com/article/10.1007/s11042-019-08385-4.

[11] D.-K. Imrus Salehin. „AReviewonDropoutRegularization Approaches for Deep Neural Networks within the Scholarly Domain.“ (2023), Adresse: https://www.mdpi.com/2079-9292/12/14/3106.

[12] F. Sarikaya. „Learning Rate Optimization in Neural Networks: Challenges and Solutions in Training Dynamics.“ (2024), Adresse: https://www.researchgate.net/publication/385698289_Learning_Rate_Optimization_in_Neural_Networks_Challenges_and_Solutions_in_Training_Dynamics.

[13] O. Ramesh Paudyal 1 Akash D. Shah 2. „Artificial Intelligence in CT and MR Imaging for Oncological Applications.“ (2023), Adresse: https://www.mdpi.com/2072-6694/15/9/2573.

[14] R. A. AdhamAleid * KhalidAlhussaini. „Artificial Intelligence Approach for Early Detection of Brain Tumors Using MRIImages.“ (2023), Adresse: https://www.mdpi.com/2076-3417/13/6/3808.

[15] R. Pedro R. A. S. Bassi1. „A deep convolutional neural network for COVID-19 detection using chest X-rays.“ (2021), Adresse: https://link.springer.com/article/10.1007/s42600-021-00132-9.

[16] S. A. A. Al-Omaisi Asia 1 Cheng-Zhang Zhu. „Detection of Diabetic Retinopathy in Retinal Fundus Images Using CNNClassification Models.“ (2022), Adresse: https://www.mdpi.com/2079-9292/11/17/2740.

[17] J. Z. Masaki Ikuta. „A Deep Recurrent Neural Network with Gated Momentum Unit for CT Image Reconstruction.“ (2015), Adresse: https://www.researchgate.net/publication/353557120_A_Deep_Recurrent_Neural_Network_with_Gated_Momentum_Unit_for_CT_Image_Reconstruction.

[18] R. G. Sanjeev Kumar Saini. „Artificial intelligence methods for analysis of electrocardiogram signals for cardiac abnormalities: state-of-the-art and future challenges.“ (2021), Adresse: https://link.springer.com/article/10.1007/s10462-021-09999-7.

[19] L. I. Saad Bin Ahmed* RobertoSolis-Oba. „Explainable-AI in Automated Medical Report Generation Using Chest X-ray Images.“ (2021), Adresse: https://www.mdpi.com/2076-3417/12/22/11750.

[20] W. A. S. Charithea Stylianides Andria Nicolaou. „AI Advances in ICU with an Emphasis on Sepsis Prediction: An Overview.“ (2025), Adresse: https://www.researchgate.net/publication/387853503_AI_Advances_in_ICU_with_an_Emphasis_on_Sepsis_Prediction_An_Overview.

[21] R. L. Georg Langs1 Ulrike Attenberger2. „Maschinelles Lernen in der Radiologie.“ (2020), Adresse: https://link.springer.com/article/10.1007/s00117-019-00624-x.

[22] F. P. Donovan Slabbert, „Classical-quantum approach to image classification: Autoencoders and quantum SVMs,“ 2023. Adresse: https://www.researchgate.net/publication/392439298_Classical-quantum_approach_to_image_classification_Autoencoders_and_quantum_SVMs.

[23] G. B. Alankrita Aggarwal a Mamta Mittal b. „Generative adversarial network: An overview of theory and applications.“ (), Adresse: https://www.sciencedirect.com/science/article/pii/S2667096820300045.

[24] P. S. S. Vidyadhar Upadhya. „An Overview of Restricted Boltzmann Machines.“ (2019), Adresse: https://link.springer.com/content/pdf/10.1007/s41745-019-0102-z.pdf.

[25] M. N. P. Nicholas Dietrich Bo Gong. „Adversarial artificial intelligence in radiology: Attacks, defenses, and future considerations.“ (2025), Adresse: https://www.researchgate.net/publication/391941548_Adversarial_artificial_intelligence_in_radiology_Attacks_defenses_and_future_considerations.

[26] N. E. K. Atrab A. Abd El-Aziz1 Reda A. El-Khoribi. „RDMAA: Robust Defense Model against Adversarial Attacks in Deep Learning for Cancer Diagnosis.“ (2024), Adresse: https://www.researchgate.net/publication/378863297_RDMAA_Robust_Defense_Model_against_Adversarial_Attacks_in_Deep_Learning_for_Cancer_Diagnosis.

[27] S. Z. Shilin Qiu Qihe Liu. „Review of Artificial Intelligence Adversarial Attack and Defense Technologies.“ (2019), Adresse: https://www.mdpi.com/2076-3417/9/5/909.

[28] S. F. MAFIZUR RAHMAN1 PROSENJIT ROY2. „Evaluating Pretrained Deep Learning Models for Image Classification Against Individual and Ensemble Adversarial Attacks.“ (2025), Adresse: https://www.researchgate.net/publication/389196953_Evaluating_Pretrained_Deep_Learning_Models_for_Image_Classification_Against_Individual_and_Ensemble_Adversarial_Attacks.

[29] A. Y. Elif DEGIRMENCI Ilker OZCELIK. „Effects of Untargeted Adversarial Attacks on Deep Learning Methods.“ (2022), Adresse: https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9931786.

[30] L. Z. Meng Shen Changyue Li. „Decision-Based Query Efficient Adversarial Attack via Adaptive Boundary Learning.“ (2024), Adresse: https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=10163476.

[31] S. S. Kusum Lata Prashant Singh. „Exploring Model Poisoning Attack to Convolutional Neural Network Based Brain Tumor Detection Systems.“ (2024), Adresse: https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=10528710.

[32] G. M. Michał Marcinkiewicz. „Quantitative Impact of Label Noise on the Quality of Segmentation of Brain Tumors on MRI scans.“ (2019), Adresse: https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8859971.

[33] S. G. MARIA RIGAKI. „A Survey of Privacy Attacks in Machine Learning.“ (2023), Adresse: https://www.researchgate.net/publication/373958404_A_Survey_of_Privacy_Attacks_in_Machine_Learning.

[34] C. C. U. Gladys W. Muoka1 DingYi. „AComprehensive Review and Analysis of Deep Learning-Based Medical Image Adversarial Attack and Defense.“ (2023), Adresse: https://www.researchgate.net/publication/374707820_A_Comprehensive_Review_and_Analysis_of_Deep_Learning-Based_Medical_Image_Adversarial_Attack_and_Defense.

[35] A. Z. Sheikh Burhan ul haque. „Robust Medical Diagnosis: A Novel Two-Phase Deep Learning Framework for Adversarial Proof Disease Detection in Radiology Images.“ (2024), Adresse: https://link.springer.com/article/10.1007/s10278-023-00916-8.

[36] S. H. Zhiping Lu Hongchao Hu. „Ensemble Learning Methods of Adversarial Attacks and Defenses in Computer Vision: Recent Progress.“ (2022), Adresse: https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=10013347.

[37] Q. Weimin Zhao SanaaAlwidian. „Adversarial Training Methods for Deep Learning: A Systematic Review.“ (2022), Adresse: https://www.mdpi.com/1999-4893/15/8/283.

[38] S. P. Ganesh Ingle. „Enhancing Adversarial Defense in Neural Networks by Combining Feature Masking and Gradient Manipulation on the MNIST Dataset.“ (2024), Adresse: https://www.researchgate.net/publication/377814416_Enhancing_Adversarial_Defense_in_Neural_Networks_by_Combining_Feature_Masking_and_Gradient_Manipulation_on_the_MNIST_Dataset.

[39] X. J. J. M. A. Fan. „Robust Feature Matching for Remote Sensing Image Registration via Linear Adaptive Filtering.“ (2021), Adresse: https://ieeexplore.ieee.org/abstract/document/9130049.

[40] M. A. W. Talib Iqball. „Weighted ensemble model for image classification.“ (2023), Adresse: https://link.springer.com/article/10.1007/s41870-022-01149-8.

[41] H. B. Chen Zhanga Yu Xieb. „A Survey of Federated Learning.“ (2021), Adresse: https://www.sciencedirect.com/science/article/pii/S0950705121000381.

[42] S. L. Yang Li. „The Threat of Adversarial Attack on a COVID-19 CT Image-Based Deep Learning System.“ (2023), Adresse: https://www.researchgate.net/publication/389068929_Threats_to_medical_diagnosis_systems_analyzing_targeted_adversarial_attacks_in_deep_learning-based_COVID-19_diagnosis.

[43] Y. L. Xuanyi Li Yajie Pang. „The Robustness of Deep Learning Models to Adversarial Attacks in Lung X-ray Classi cation.“ (2024), Adresse: https://www.researchsquare.com/article/rs-4923634/v1.

[44] L. G. Xingjun Ma Yuhao Niu. „Understanding adversarial attacks on deep learning based medical image analysis systems.“ (2019), Adresse: https://www.researchgate.net/publication/334669159_Understanding_Adversarial_Attacks_on_Deep_Learning_Based_Medical_Image_Analysis_Systems.

[45] F. N. Magdalini Paschali1 Sailesh Conjeti2. „Generalizability vs. Robustness: Adversarial Examples for Medical Imaging.“ (2018), Adresse: https://www.researchgate.net/publication/324167403_Generalizability_vs_Robustness_Adversarial_Examples_for_Medical_Imaging.

[46] M.-E. L. Min-Jen Tsai Ping-Yi Lin. „Adversarial Attacks on Medical Image Classification.“ (2023), Adresse: https://www.mdpi.com/2072-6694/15/17/4228.

[47] (), Adresse: https://www.kaggle.com/datasets/sartajbhuvaji/brain-tumor-classification-mri.

[48] B. R. T. Sathyanarayanan Swaminathan. „Confusion Matrix-Based Performance Evaluation Metrics.“ (2024), Adresse: https://www.researchgate.net/publication/386347454_Confusion_Matrix-Based_Performance_Evaluation_Metrics.

[49] T. R.-M. Damien Rolon-M´erette Matt Ross. „Introduction to Anaconda and Python: Installation and setup.“ (2020), Adresse: https://www.researchgate.net/publication/347785403_Introduction_to_Anaconda_and_Python_Installation_and_setup.

Anhang

Anhangsverzeichnis

A Anhang A

A.1 Versionen der verwendeten Software

- python: 3.9.13
- cv2: 4.11.0
- tensorflow: 2.19.0
- scikit-learn (sklearn): 1.6.1
- jupyterlab: 4.4.3
- seaborn: 0.13.2

Erklärung zur selbständigen Bearbeitung einer Abschlussarbeit

Gemäß der Allgemeinen Prüfungs- und Studienordnung ist zusammen mit der Abschlussarbeit eine schriftliche Erklärung abzugeben, in der der Studierende bestätigt, dass die Abschlussarbeit „— bei einer Gruppenarbeit die entsprechend gekennzeichneten Teile der Arbeit [(§ 18 Abs. 1 APSO-TI-BM bzw. § 21 Abs. 1 APSO-INGI)] — ohne fremde Hilfe selbständig verfasst und nur die angegebenen Quellen und Hilfsmittel benutzt wurden. Wörtlich oder dem Sinn nach aus anderen Werken entnommene Stellen sind unter Angabe der Quellen kenntlich zu machen."

Quelle: § 16 Abs. 5 APSO-TI-BM bzw. § 15 Abs. 6 APSO-INGI

Dieses Blatt, mit der folgenden Erklärung, ist nach Fertigstellung der Abschlussarbeit durch den Studierenden auszufüllen und jeweils mit Originalunterschrift als letztes Blatt in das Prüfungsexemplar der Abschlussarbeit einzubinden.
Eine unrichtig abgegebene Erklärung kann -auch nachträglich- zur Ungültigkeit des Studienabschlusses führen.

Erklärung zur selbständigen Bearbeitung der Arbeit

Hiermit versichere ich,

Name: Al Methiab

Vorname: May

dass ich die vorliegende Bachelorarbeit – bzw. bei einer Gruppenarbeit die entsprechend gekennzeichneten Teile der Arbeit – mit dem Thema:

Adversariale Angriffe auf KI-Modelle zur Tumorerkennung: Sicherheitsrisiken in der medizinischen Bildklassifikation

ohne fremde Hilfe selbständig verfasst und nur die angegebenen Quellen und Hilfsmittel benutzt habe. Wörtlich oder dem Sinn nach aus anderen Werken entnommene Stellen sind unter Angabe der Quellen kenntlich gemacht.

- die folgende Aussage ist bei Gruppenarbeiten auszufüllen und entfällt bei Einzelarbeiten -

Die Kennzeichnung der von mir erstellten und verantworteten Teile der Bachelorarbeit ist erfolgt durch:

Hamburg	23.10.2025	
Ort	Datum	Unterschrift im Original

Erklärung zur selbständigen Bearbeitung

Hiermit versichere ich, May Al Methiab, dass ich die vorliegende Bachelorarbeit mit dem Thema:

Adversariale Angriffe auf KI-Modelle zur Tumorerkennung: Sicherheitsrisiken in der medizinischen Bildklassifikation

ohne fremde Hilfe selbständig verfasst und nur die angegebenen Quellen und Hilfsmittel benutzt habe. Wörtlich oder dem Sinn nach aus anderen Werken entnommene Stellen sind unter Angabe der Quellen kenntlich gemacht.

Ort	Datum	Unterschrift im Original