
**Hochschule für Angewandte Wissenschaften Hamburg**
**Fakultät Life Sciences**
Department Medizintechnik

# Bachelorthesis

Studiengang

Medizintechnik

**Titel:**

## Automatische Diagnose von Thorax-Röntgenbildern mit Hilfe von Faltungsnetzwerken

Vorgelegt von:
Nassim Haidar
Matrikelnummer: [redacted]

Hamburg
am 10.01.2019

Gutachter:
Prof. Dr.-Ing. Andreas Meisel
Prof. Dr.-Ing. Thomas Schiemann

Nassim Haidar

# Automatische Diagnose von Thorax-Röntgenbildern mit Hilfe von Faltungsnetzwerken

Bachelorthesis eingereicht im Rahmen der Bachelorprüfung
im Studiengang Medizintechnik
an der Fakultät Life Sciences
der Hochschule für Angewandte Wissenschaften Hamburg

Erstprüfer: Prof. Dr.-Ing. Andreas Meisel

Zweitprüfer: Prof. Dr.-Ing. Thomas Schiemann

Abgegeben am 10. Januar 2019

**Nassim Haidar**

## Thema der Bachelorthesis

Automatische Diagnose von Thorax-Röntgenbildern mit Hilfe von Faltungsnetzwerken

## Stichworte

Thorax-Röntgenbild, Deep Learning, Faltungsnetzwerke, Screening, Konfusionsmatrix, Bildklassifizierung, ROC-Kurve, Fehldiagnosen, CNN

## Zusammenfassung

Studien in Deutschland haben gezeigt, dass über 10 % aller Diagnosen, die von Ärzten in Kliniken und Praxen gestellt werden falsch sind [Jörg Blech, 2011]. Fehldiagnosen können auf menschliche Fehlern beruhen und solange Menschen Entscheidung treffen müssen, wird sich die Zahl der Fehldiagnosen nicht verringern.

Ziel dieser Thesis ist es, ein Convolutional Neural Layer (*CNN*) aufzubauen, der die Fehldiagnosen verringern soll. Dazu wird folgende Forschungsfrage gestellt: Sind *CNN* aus Deep Learning, in der Lage, Fehldiagnosen die Radiologen anhand von Thorax-Röntgenbildern stellen, zu verringern? Dabei soll ebenfalls geprüft werden, inwiefern ein solches *CNN* sich als Screening-Programm eignet.

Um die Forschungsfrage zu beantworten, wird nach jedem Testverfahren eine *Konfusionsmatrix* dargestellt, die die Röntgenbilder in Klassen von richtig negativ bis falsch negativ unterteilt. Die *Konfusionsmatrix* hat gezeigt,

dass die aufgebauten *CNN* eine höhere Fehldiagnose und dementsprechend eine schlechtere *Accuracy* (wie oft das *CNN* die Bilder pro *Epoche* prozentual richtig prognostiziert hat) vorweisen als Radiologen. Dieses Resultat zeigt eine Diskrepanz zu anderen bekannten Studien.

Auf dieser Grundlage ist darauf hinzuweisen, dass eine homogene Verteilung der Daten essenziell ist und die *Accuracy* und Fehldiagnosen des *CNN*s somit beeinflusst werden kann. Daher soll in weiteren Forschungen darauf geachtet werden, dass die gesamte Datensammlung einheitlich ist und eine gleichmäßige Verteilung an positiven sowie negativen Befunden darstellt.

# Danksagung

Zuerst gebührt mein Dank Herrn Prof. Dr.-Ing. Andreas Meisel, der mir bei der Erstellung dieses Themas half und meine Bachelorthesis betreut und begutachtet hat. Außerdem möchte ich mich für das entgegengebrachte Vertrauen und für die konstruktive Kritik bei der Erstellung dieser Arbeit bedanken.

Außerdem möchte ich mich bei meinem Studienfreund Hasibullah Shafaq bedanken, der stets einen Hilfreichen Rat hatte und somit zur erfolgreichen Erstellung dieser Thesis beigetragen hat.

Ich möchte mich insbesondere bei meinen Eltern, meinen Geschwistern und meinen Freunden bedanken, die mich während dieser schwierigen Zeit stets unterstützt haben und stets für mich da waren.

Hamburg, 10. Januar 2019

**Nassim Haidar**

# Abkürzungsverzeichnis

| | |
|---|---|
| ANN | Artificial Neural Network |
| AI | Artificial Intelligence |
| ReLU | Rectifier Linear Unit |
| CNN | Convolutional Neural Layer |
| NLP | Natural Language Processing |
| CT | Computertomographie |
| MRT | Magnetresonaztomographie |
| ReLU | Rectified Linear Unit |
| AI | Artificial Intelligence |
| ROC | Receiver Operating Characteristic |
| ANN | Artificial Neural Network |
| AUC | Area Under the Curve |
| AUROC | Area Under the Receiver Operating Characteristic |
| NIH | National Institutes of Health |

# Inhaltsverzeichnis

# Abbildungsverzeichnis

# Tabellenverzeichnis

# 1. Einführung

## 1.1. Einleitung

Immer mehr Krankenhäuser werden privatisiert und werden zu einem Unternehmen umstrukturiert, das danach strebt, hohe Gewinne zu erzielen. Durch die Rationalisierung hat sich die Anzahl der behandelten Patienten pro Krankenhaus erhöht, hingegen sich die Anzahl der Ärzte kaum verändert hat [Ärzte Zeitung, 2014]. Somit werden Visiten und Anamnese sehr kurzgehalten. Die Folge ist erhöhte Fehldiagnosen.

In einer Studie wurde festgestellt, dass die Fehldiagnose in Krankenhäuser und Praxen über 10 % liegen, wobei die Dunkelziffer weit höher liegen könnte. Dabei belasten Fehldiagnosen in erster Linie nicht nur die Ärzte und Steuerzahler, die das Gesundheitssystem mitfinanzieren, sondern vor allem die Patienten. Die anhand einer falschen Diagnose mit hoher Wahrscheinlichkeit weiterführende Untersuchen ausgesetzt sind, die auf ihre Gesundheit schädlich sein können. Dennoch wäre es möglich die Entscheidungsfähigkeit durch Maschinen zu ersetzten.

Im Rahmen dieser Forschungsarbeit soll beantwortet werden, ob Fehldiagnosen, die durch menschliches Handeln bedingt sind, minimiert werden können, indem ein *CNN* aufgebaut wird, das Thorax-Röntgenbildern mit einer hohen *Accuracy* richtig vorhersagt. Zu diesem Zweck wurde mit Hilfe einer quantitativen Studie die *Konfusionsmatrix* und der *ROC-Kurve* der einzelnen *CNN* interpretiert. Zusätzlich wurden die Kontrollkriterien *Spezifität*, *Sensitivität* und der *positiver prädiktiver Wert* mit in die Bewertung einbezogen. *Sensitivität* ist ein Maß, um Menschen, die die Erkrankung haben, auch als krank bzw. als positiv zu erkennen. *Spezifität* steht dafür, gesunde Menschen nach einem Test tatsächlich als gesund bzw. als negativ zu Befunden. Der *positiver prädiktiver Wert* gilt als Parameter zur Einschätzung des Verfahrens. Dabei sagt dieser Parameter aus, wie viele

positive Befunde, die das Testverfahren macht, auch tatsächlich positiv sind. Weiterhin soll mithilfe der Kontrollkriterien untersucht werden, inwiefern sich *CNNs* als Screening Programme etablieren.

Der Beruf des Radiologen und allgemein des Arztes gilt bis heute als einer der sichersten Berufe, die nicht durch Maschinen ersetzt werden können. Der Arbeitsbereich eines Radiologen umfasst überwiegend das Auswerten von Röntgenbildern. Der erhebliche Fortschritt in der Medizin der vergangenen 30 Jahre ermöglichte es Radiologen, schneller Krankheitsbilder zu erkennen. Dennoch haben sich die Fehldiagnosen in dieser Zeit kaum verändert und bewegten sich in europäischen Universitätskliniken bei Werten von bis zu 10 % [Kirch, 2005].

Ein Radiologe beurteilt während seiner Ausbildung und später als Facharzt eine große Menge an Röntgenbildern. Dabei steigen die *Sensitivität* und *Spezifität* mit zunehmender Erfahrung des Radiologen an [Vincent C. A., et al., 1988]. Trotz langjähriger Erfahrungen können Befunde übersehen oder missdeutet werden. Weiterhin besteht die Problematik, dass ein Röntgenbild von verschiedenen Radiologen unterschiedlich bewertet werden kann. So kann eine physiologische Schwellung im Zahnfleisch unter Umständen im Röntgenbild als eine bakterielle Entzündung oder auch als eine schlechte Mundhygiene gesehen werden. So lange der Mensch Entscheidungen trifft, kann es menschlich bedingt zu Fehlentscheidungen kommen. Diese menschliche Fehlerquelle wird bei Hinzuziehen von Maschinen minimiert oder verhindert.

Maschinen sind nicht nur leistungsfähiger als Menschen, sie arbeiten über Stunden und sogar über Jahre mit derselben *Accuracy*. Dank der heutigen großen Datenansammlung an Röntgenbildern in Krankenhäusern ist es möglich, über Deep Learning ein *CNN* zu erstellen. Der die Erkrankungen im

Röntgenbild schnell erkennt, effizient ist und eine hohe *Accuracy* erreichen kann.

Maschinen sind statisch, da sie nur bei aufrufbaren Symptomen handeln können. Viele Erkrankungen haben hingegen einen symptomlosen Verlauf oder verlaufen atypisch. Hierbei reicht es nicht aus, sich die Laborwerte oder das Röntgenbild anzuschauen. Vielmehr müssen Ärzte dynamisch sein und intuitiv handeln, da sie den Patienten ansehen, anfassen, wahrnehmen, untersuchen und ihm zuhören müssen. Sie müssen empathisch wirken und auch in der Lage sein, ethisch schwierige Entscheidungen zu treffen. All diese Attribute können Maschinen nicht ersetzen.

In einem Gedankenexperiment von Ian Kerr wurde folgendes Szenario durchgespielt. Ein Supercomputer Dr. Watson, der von IBM entwickelt wurde, stellt dem Patienten die Diagnose Leukämie mit einer Wahrscheinlichkeit von 90 % dar. Die Diagnose des Arztes stimmt aufgrund seines Fachwissens und seiner Intuition mit einer Wahrscheinlichkeit von 50 % nicht mit der Diagnose des Computers überein. Dabei stellte Kerr die Frage, wem die Patienten eher ihr Leben anvertrauen würden, dem Menschen oder der Maschine? [Heckl, 2017]

## 1.2. Stand der Forschung

Eine Maschine, die Bilder klassifiziert, wird mit *CNN* realisiert. Ein *CNN* gehört zum Bereich des Deep Learnings, dieser wiederum ist eine Erfindung, die sich aus Artificial Intelligence (AI) ableitet. Die AI wurde anfangs des 19. Jahrhunderts erfunden [Manhart, 2018]. In den 50er-Jahren waren Forscher überzeugt, dass sie in der Lage seien, der Maschine eine menschenähnliche künstliche Intelligenz einzubauen. Zu dieser frühen Phase der AI wurden bereits die Grenzen des Machbaren erreicht. Das lag einerseits an der

Rechenleistung, die noch viel zu gering war, wodurch komplexere Berechnungen viel Zeit in Anspruch nahmen. Anderseits gab es zu wenige Trainingsdaten, mit denen die Maschine hätte trainiert werden können.

Es folgte eine Reihe von neuen Erfindungen wie die Backpropagation oder das *CNN*, dass das Deep Learning umfasst und so die Technologie wieder weiter vorantrieb [Hertwig, 2018]. So verfügen Computer in der heutigen Zet über hohe Rechenleistungen, wodurch komplexe Szenarien in kurzer Zeit simulierbar sind. Des Weiteren ist aufgrund des Internets und der globalen Datensammlung Trainingsmaterial entstanden, das nun für Forschungszwecke genutzt werden kann.

Deep Learning tritt immer häufiger in der menschlichen Gesellschaft auf. So werden Chatboots oder auch digitale Dialogsysteme verwendet, um eine Interaktion mit Kunden durchzuführen. Auch die Spielindustrie wurde auf Deep Learning aufmerksam. So wurde der weltbeste Go-Spieler mit der Software AlphaGo, die von Google DeepMind entwickelt wurde, mit 4:1 besiegt. Eine weitere Anwendung in Deep Learning ist die Verarbeitung von zeitlichen Sequenzen. So können gesprochene oder geschriebene Texte erkannt werden [Christoph Wick , 2017]. Auch das von Datenschützern umstrittene Massenüberwachungssystem der Gesichtserkennung in China basiert auf Deep Learning.

Auch in der Medizin gab es ähnliche Versuche. So wurde ein internationaler Wettbewerb ausgeschrieben, in dem Forscherteams aus aller Welt ein neuronales Netzwerk entwickeln sollen. Dieser soll aus *CNN* bestehen und zur Diagnose von Lymphknoten Auskunft geben. *CNN*s sind besonders gute Bildklassifikatoren, die unter bestimmten Umständen eine *Accuracy* von bis zu 99,5 % erreichen können [Chang, 2017]. Beim internationalen Wettbewerb musste das *CNN* gegen zwei Gruppen von elf Pathologen antreten. Die eine Gruppe hatte für 129 Präparate nur zwei Stunden Zeit, während der anderen Gruppe so viel Zeit gewährt wurde, wie sie es für nötig hielt. Die Aufgabe, die

den Pathologen und dem Deep-Learning-Netzwerk gestellt wurde, war es, ein Schnittpräparat in Hämatoxylin-Eosin(HE)-Färbung mit Krebszellen zu erkennen. Am Ende erzielten sowohl die erste Gruppe, die nur zwei Stunden Zeit hatte, als auch die zweite Gruppe, die ca. 30 Stunden brauchte, schlechtere *Accuracy* als das *CNN* [J. H. M. J. Vestjens, et al., 2012].

Es gibt mehrere Studien, die die *Accuracy*, *Sensitivität* und die *Spezifität* eines *CNN* bei der Befundung von Krankheiten exakt beschreiben ( [Zreik M, et al., 2018] oder [Lamash Y, et al., 2018]). Es ist jedoch kein Bezug auf unterschiedliche *CNN* zu finden, die ihr Verhalten anhand dem positiven prädiktiven Wert, der *Sensitivität* und der *Spezifität* sowie *CNN* als zukünftige Screening-Programme untersuchen.

## 1.3. Ziel der Arbeit

Im Rahmen dieser Forschungsarbeit soll die Frage beantwortet werden, ob ein *CNN* in der Lage ist, anhand von Röntgenbildern mit einer hohen *Accuracy* richtig zu Befunden, um Fehldiagnosen, die menschlich bedingt sind, zu minimieren. Diese Vorgehensweise kann auch in Screening-Programmen von Nutzen sein, somit wird auch der Bezug auf ein mögliches erweitertes Screening-Programm vorgenommen. Dies wird anhand eines Beispiels mit der Mammographie Screening verglichen und ausgewertet.

## 1.4. Methode

Anhand einer quantitativen Studie werden *CNN* mittels einer *Konfusionsmatrix* auf ihre *Accuracy* analysiert. Weiter wird eine *ROC-Kurve* gezeigt, um den *CNN* bewerten zu können. Schließlich wird ein Vergleich mit der gesetzlichen Krebsfrüherkennungsuntersuchung Mammographie gezogen. Die Entscheidung fiel aufgrund der hohen Validität durch messbare Werte auf eine quantitative Analyse.

In dieser Arbeit wird die *Accuracy* der drei *CNN* mittels des veröffentlichten Datenpakets von Thorax-Röntgenbildern in Kaggle miteinander verglichen. Hierbei wird in den ersten zwei *CNN* analysiert, wie hoch die *Accuracy* bei nur einer Erkrankung, *Kardiomegalie* ist. Im letzten *CNN* wird untersucht, wie hoch die *Accuracy* auf alle Erkrankung ist. Anschließend werden die Ergebnisse dieser *Konfusionsmatrix* zusammengefasst. Daraus lässt sich feststellen, ob ein *CNN* aufgrund der hohen *Accuracy* Fehldiagnosen minimieren kann. Zusätzlich wird untersucht, ob es sich als ein neues Screening-Programm eignet.

## 1.5. Struktur der Arbeit

Die vorliegende Arbeit ist in 7 Hauptkapitel gegliedert. Nach der Einleitung werden im zweiten Kapitel grundlegende Fachkenntnisse über neuronale Netze erklärt, insbesondere, wie ein neuronales Netz aufgebaut ist und welche Eigenschaften es benötigt. Hierfür wird nur in geringem Umfang Bezug auf mathematische Berechnungen genommen. Außerdem wird auf das *Overfitting*-Problem eingegangen. Schließlich wird das *CNN* als Bildklassifikator vorgestellt, der in dieser Arbeit eine bedeutende Rolle einnimmt.

Im dritten Kapitel wird die klinische Definition von Screening verdeutlicht. Dabei sollen Vor- und Nachteile bei einer Teilnahme von Screening gegenübergestellt werden. Darüber hinaus werden die Ziele, die sich aus einem Screening ergeben aufgeführt. Zusätzlich wird anhand eines Beispiels zur Mammographie dargestellt, wie Ergebnisse interpretiert werden können.

Das vierte Kapitel beschreibt den Versuchsaufbau bis hin zur Auswertung der Resultate. Zusätzlich wird auf die Datenvorbereitung eingegangen, da dies für den Verlauf des Versuches ein relevanter Bezugspunkt ist. Demnach ergeben sich in den Ursprungsdaten weitere Probleme, die von einem

Radiologen in Detail kritisiert werden. Anschließend werden die Erkrankungen in Stichpunkten beschrieben und um welche Art von Erkrankung es sich hierbei handelt. Für die Beantwortung der Forschungsfrage werden drei Versuche vorbereitet und anschließend ausgewertet.

Im fünften Kapitel werden die Ergebnisse aus dem vorherigen Kapitel zusammengefasst. Dabei wird die Forschungsfrage beantwortet.

Im vorletzten Kapitel werden die Ergebnisse kritisch betrachtet und interpretiert. Dabei werden noch einige Beispiele genannt und Verbesserungsmöglichkeiten vorgeschlagen.

Das letzte Kapitel soll einen Ausblick auf weitere Schritte geben, die aufgrund der zeitlichen Begrenzung in der Bachelorthesis nicht behandelt werden konnten.

# 2. Grundlagen

In diesem Kapitel werden die Grundlagen eines neuronalen Netzwerkes beschrieben. Darüber hinaus wird speziell auf das *CNN* eingegangen, da es in dieser Arbeit um ein Netzwerk geht, das Bilder klassifizieren soll.

## 2.1. Neuronales Netzwerk

Ein künstliches neuronales Netzwerk (eng. Artifical Neural Network (ANN)) gehört in die Kategorie des Deep Learnings und ist ein Teilgebiet des maschinellen Lernens. Ein simples neuronales Netzwerk, auch Single-Layer-Perceptron, besteht aus mindestens zwei Schichten, einem Eingangs-Layer

(eng. Input „*X*“) und einem Ausgangs-Layer. Das Konzept sowie die Struktur eines neuronalen Netzwerkes stammen aus der Biologie bzw. aus dem menschlichen Gehirn und werden mittels der Informatik modifiziert.

Komplexe Netzwerke verfügen zu den oben genannten Schichten noch über eine weitere Schicht, den Hidden Layer. Diese Schicht befindet sich zwischen dem Eingangs- und dem Ausgangs-Layer (Abb. 1). Jedes Neuron ist mit allen weiteren Neuronen in der nächsten Schicht verbunden. So gelangen die Eingangswerte über den Eingangs-Layer. Von dort aus werden die Werte modifiziert und zum nächsten Neuron im Hidden Layer überreicht. Zur Modifizierung gehört das multiplizieren der Werte der Neuronen mit ihrem Gewicht. Anschließend wird dieser neuer Wert der Aktivierungsfunktion übergeben. Im letzten Layer werden die Werte ausgegeben. Die Bezeichnung „Deep“ bezieht sich auf die Charaktereigenschaft eines neuronalen Netzes, die es ihm ermöglicht, über viele Schichten im Hidden Layer zu verfügen [Tariq Rashid, 2017]. Das Neuron ist die kleinste Einheit in einem neuronalen Netzwerk. Dabei ist zu beachten, dass das künstliche Neuron sich vom biologischen Neuron in der Funktionalität unterscheidet. Jedes Neuron hat neben seinem Eingangswert noch einen Wert für das Gewicht $w_1$.

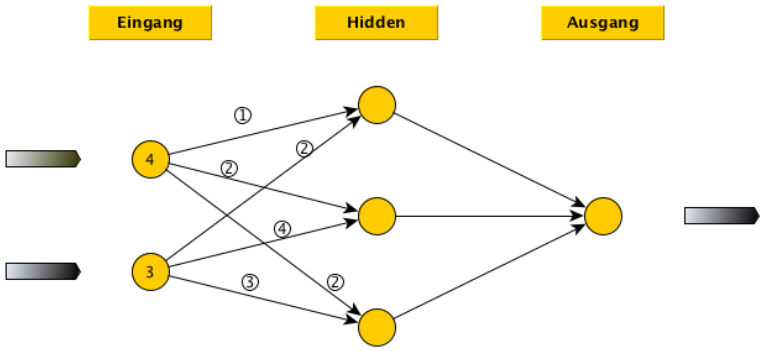



*Abbildung 1: Grundstruktur eines neuronalen Netzwerks mit einem Eingabe-, Ausgabe- sowie einem Hidden Layer*

Das Gewicht ist ein Wert, der die Neuronen verbindet. Er wird anfangs zufällig gewählt und nach jedem Durchlauf auch *Epoche* aufgrund der Fehlfunktion (s Kapitel 2.3) und dem *Backpropagation* (s Kapitel 2.4) neu berechnet. Eine *Epoche* bedeutet, wenn die gesamten Trainingsdaten denn

*CNN* einmalig durchlaufen hat. Die Variablen des Gewichts und der Eingangswerte werden miteinander multipliziert und mit dem Bias *b* addiert (Abb. 2). Der Summand ist ein zusätzlicher Eingang am Neuron mit einem konstanten Wert von 1. Dieser soll verhindern, dass das künstliche Neuron von Beginn einen Null-Wert, aufgrund der Multiplikation ausgibt (1) [Tariq Rashid, 2017].

Das Ergebnis *z* wird in eine Aktivierungsfunktion eingesetzt (s. Kapitel 2.2). Das Produkt aus der Aktivierungsfunktion (Abb. 2) ist der Ausgang (eng. Output) *y* eines einzelnes Neurons.

$$z = \sum_{i=1}^{n} x_i w_i + b \tag{1}$$

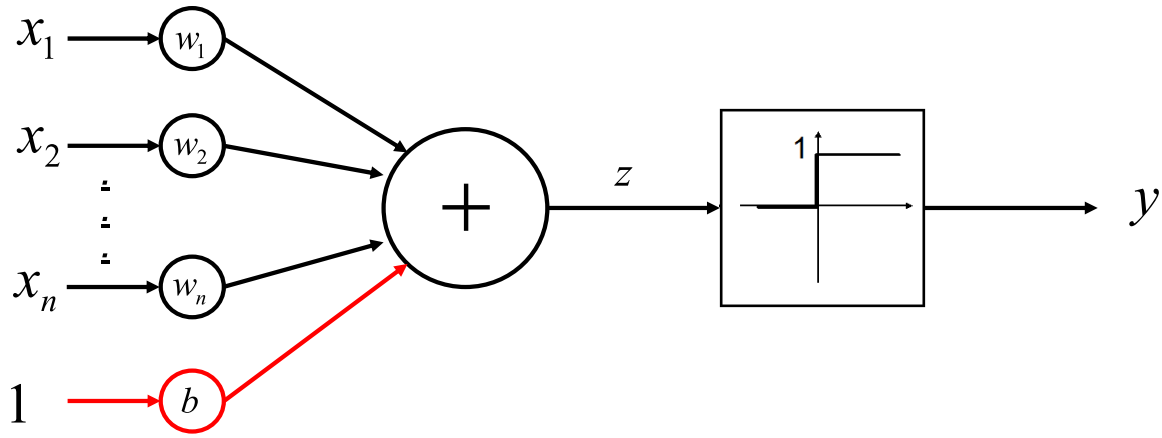



*Abbildung 2: Die kleinste Einheit, das Neuron. Rechts ist die Heaviside-Aktivierungsfunktion zu sehen, während links die Eingänge sind, die in das Neuron eingespeist werden [Meisel, Prof. Dr.-Ing. Andreas, 2018]*

## 2.2. Aktivierungsfunktion

Ein biologisches Neuron leitet den Reiz nur bei Überschreitung des Aktivierungswertes weiter. Ein ähnlicher Vorgang findet auch in den künstlichen Neuronen eines neuronalen Netzwerks statt. Mittels einer

Aktivierungsfunktion und bei einer nicht Überschreitung des Schwellenwertes wird das Neuron auf *Null* oder gesetzt. Wobei bei Überschreitung des Wertes auf *Eins* gesetzt wird. Der Ausgangswert *z* des Neurons wird daraufhin, der Aktivierungsfunktion übergeben (2). Es gibt mehrere Aktivierungsfunktionen, allerdings beschäftigt sich diese Arbeit ausschließlich mit der *Sigmoid*- (3) und der ReLU-Funktion (4).

$$y = \varphi(z) \tag{2}$$

$$\varphi(z)_{Sigmoid} = \frac{1}{1+e^{-z}} \tag{3}$$

$$\varphi(z)_{ReLU} = \begin{cases} 0 \; for \; x < 0 \\ x \; for \; x => 0 \end{cases} \tag{4}$$

Die *Sigmoid*-Funktion hat sich, aufgrund ihrer guten Differenzierbarkeit an jedem Punkt (Abb. 3) in der Praxis am besten bewährt. Außerdem hat die Funktion einen Sättigungsbereich von 0 bis 1. Dies erweist sich später als ein bedeutender Vorteil, da auch überdimensionierte Werte in dieser Funktion beschrieben werden können [Wender, 2018].

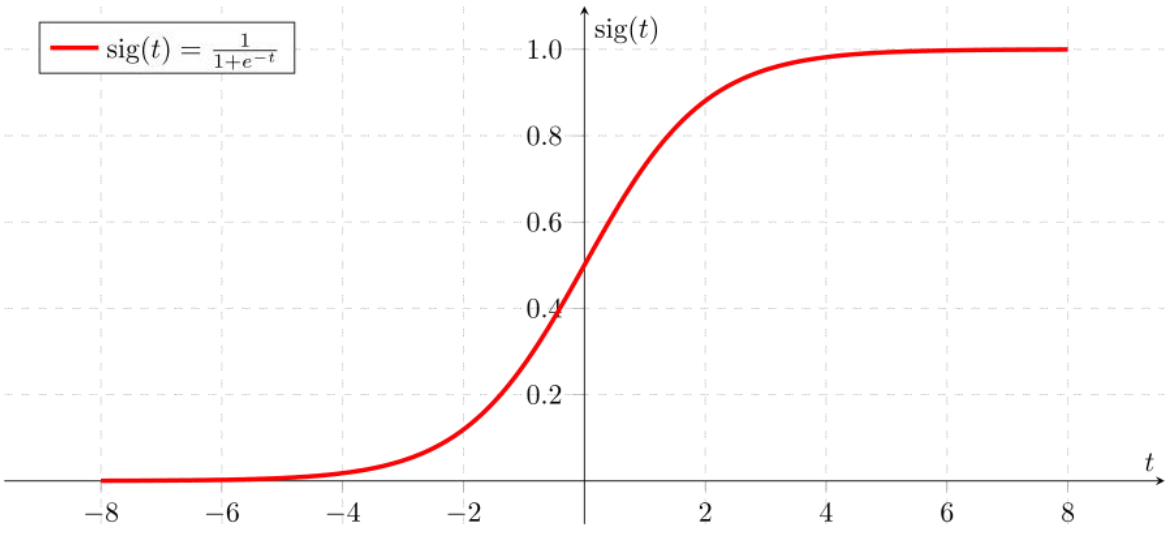



*Abbildung 3:Die logistische Kurve (Sigmoid-Funktion). Sie hat ihr Maximum bei 1 und das Minimum bei 0*

Der Wertebereich einer *Rectified Linear Unit* (ReLU) ist definiert von 0 bis x (Abb. 4). Somit werden alle negativen Ergebnisse mit 0 beziffert und bei positiver Zahl wird diese selbst ausgegeben. Die *ReLU* hat eine monotone Steigung und ist an jedem Punkt differenzierbar außer für x = 0 [Wender, 2018]. *ReLU*-Funktionen übersättigen viel langsamer als *Sigmoid*-Funktion. Dies ist aufgrund ihrer Nichtlinearität zurückzuführen. Darüber hinaus ist diese Aktivierungsfunktion schneller, genauer und effektiver als andere Aktivierungsfunktionen wie z. B. *Sigmoid*-Funktionen [Abien Fred M. Agarap, 2018].

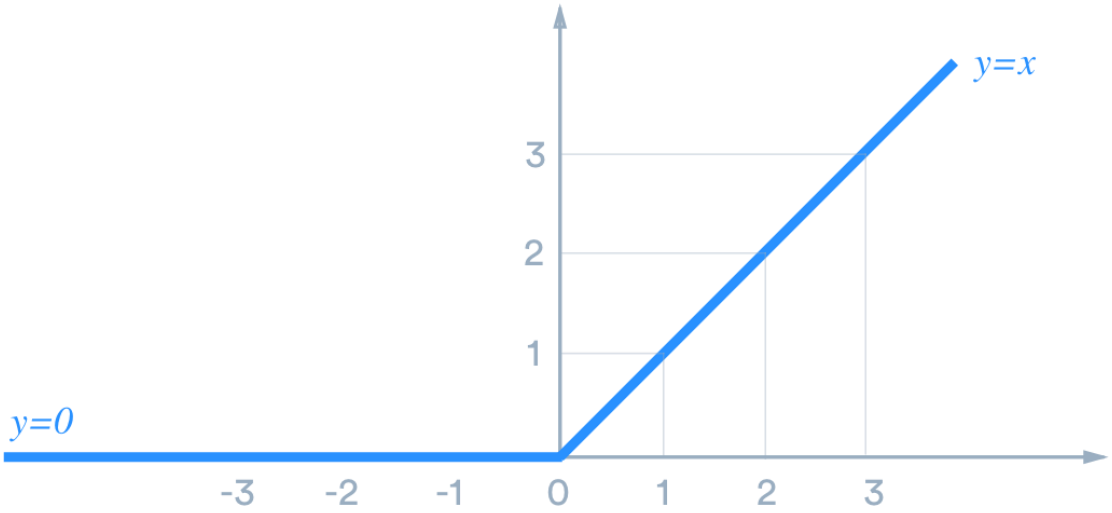



*Abbildung 4: ReLU-Funktion, die bei negativer Zahl 0 und bei positiver Zahl den positiven Wert ausgibt [TinyMind, 2018]*

## 2.3. Fehlfunktion

Die Kostenfunktion oder auch Fehlfunktion ergibt sich aus der Differenz des Sollwerts $t$ und des Ausgangswerts $y$ des Neurons im Quadrat (5). Anhand des Wertes aus der Fehlfunktion werden die Gewichte aufgrund ihren Prozentualen Anteile aktualisiert (s. Kapitel 2.4) [Tariq Rashad, 2017].

$$\delta_i = \frac{1}{n}\sum_i (t_i - y_i)^2 \tag{5}$$

Hierbei lassen sich zwei Schlüsselwörter herleiten. Die *Accuracy* und der *Loss*. *Accuracy* wird ermittelt indem die Vorhersage des *CNN*s über das Röntgenbild mit den richtigen Erkrankungen aus demselben Röntgenbild übereinstimmt. Dies geschieht indem der *CNN* die Bilder mit 0 und 1 prognostiziert. Wobei der Wert *0* für Falsch und der Wert *1* für Richtig steht. Die werden mit den wahren Werten verglichen. Der resultierende Wert wird prozentual nach jeder *Epoche* ausgegeben und entspricht die *Accuracy*.

*Loss* hingegen ist die Summation von Fehlern von über einer gesamten *Epoche*. Als Fehler wird die Falsch Aussage zwischen dem prognostizierten Bild und dem wahren Wert gesehen.

Das Ziel der Fehlfunktion ist es, nach mehreren *Epoche*n den Fehler minimal wie möglich zu halten. Ein kleiner Fehler ist ein Zeichen für minimale Differenzen zwischen dem Erwartungs- und dem Ausgangswert.

## 2.4. Gradientenverfahren

Die *Backpropagation* gehört zu den Gradientenverfahren und ist ein Optimierungsverfahren, das es dem Netzwerk ermöglicht, über mehrere Trainingsverläufe den kleinstmöglichen Fehler zu finden. Dies geschieht, indem die Gewichtswerte über ein Optimierungsverfahren nach jedem durchlauf neu berechnet und aktualisiert werden. Das Verfahren verläuft in zwei wesentlichen Schritten ab. Im Backward-Pass wird der Fehler durch $\delta_i$ bestimmt und rückwärts auf jedes einzelne Gewicht zurückgegeben. Im letzten Schritt folgt die neue Berechnung der Gewichte $w_{ij}$. Die Berechnung erfolgt, indem der Fehler mit dem Gewicht multipliziert wird und durch alle Gewichte im jeweiligen Layer dividiert wird. Schließlich wird der Fehler mit dem prozentuellen eignen Gewicht multipliziert [Francois Chollet, 2017].

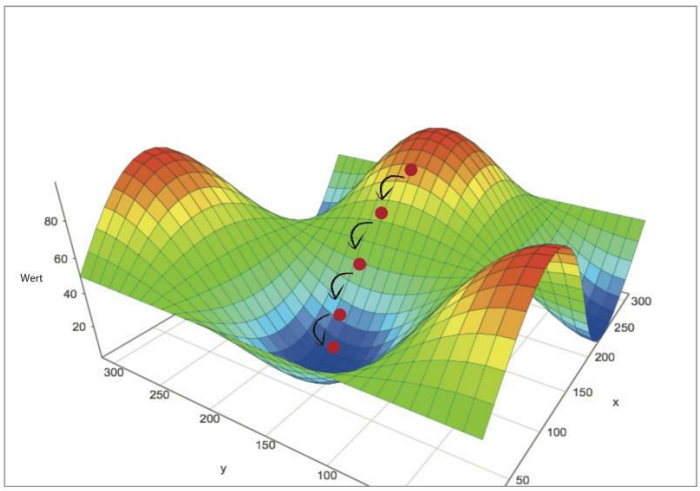



*Abbildung 5: Schritt für Schritt wird der Fehler minimiert, bis das globale Minimum gefunden ist [opening.Download, kein Datum]*

Beim Gradientenverfahren handelt es sich um eine Annäherung an den minimalen Fehler. In Abb. 5 wird eine Funktion grafisch dargestellt. Bei der ersten *Epoche* wird der Startwert zufällig auf diese Funktion gesetzt. Die Backpropagation hilft dem neuronalen Netzwerk, über mehrere *Epochen* den maximalen Gradientenabfall zu finden. Je größer die Fehlfunktion ist, desto größer ist der Gradientenabstieg. Das Ziel ist es, sich nach jeder *Epoche* dem globalen Minimum anzunähern. In komplexen Funktionen sind mehrere regionale Minima zu finden, die zu Problemen führen können [Francois Chollet, 2018].

## 2.5. Overfitting

Beim Antrainieren eines neuronalen Netzwerkes stehen einander immer zwei Größenordnungen gegenüber: O*ptimization* und G*eneralization*. Bei der *Optimization* versucht das Netzwerk, die bestmögliche *Accuracy* bei den Trainingsdaten zu erzielen, während beim letzteren das gelernte Netzwerk versucht, die gleichen Resultate auch bei unbekannten Daten zu erzielen [Francois Chollet, 2018]. Sind die Neuronen mit Informationen übersättigt, neigt das Netzwerk zum Optimieren. Dies führt zu den am häufigsten verursachten Problem aller neuronalen Netzwerke: der Übersättigung, auch

als *Overfitting* bezeichnet. Dabei neigt das Netzwerk dazu, sich sehr stark an die trainierten Daten anzupassen. Hingegen vorher nicht trainierte Daten nicht mehr richtig zugeordnet werden können. Ein neuronales Netz erzielt seine beste Performance, wenn es kurz vor der Schwelle zum *Overfitting* ist. Dabei kann *Overfitting* auf unterschiedliche Art und Weise entstehen. Ein Netzwerk mit erhöhten *Epochen* (Abb. 6), ein besonders tiefes Netzwerk mit vielen Layern oder auch eine erhöhte Lernrate können zu *Overfitting* führen. Bei erhöhten Trainingsepisoden fällt die Fehlfunktion der Trainingsdaten auf fast 0 %, gleichzeitig erhöht sich die Fehlfunktion nach einer Senkung bei den Testdaten. Dies ist ein typisches Verhalten eines Netzwerkes, das zu oft trainiert wird und nun unter *Overfitting* leidet. Deshalb ist es auch ratsam, die Daten immer in Trainings- und Testdaten zu unterteilen, da unter diesen Umständen leicht nachgeprüft werden kann, ob das Netzwerk unter *Overfitting* leidet [Francois Chollet, 2018]. In dieser Arbeit werden zwei aktive Varianten zur Verminderung von *Overfitting* eingesetzt. Hierbei handelt es sich um das Dropout-Verfahren sowie das Max- bzw. Average-Pooling.

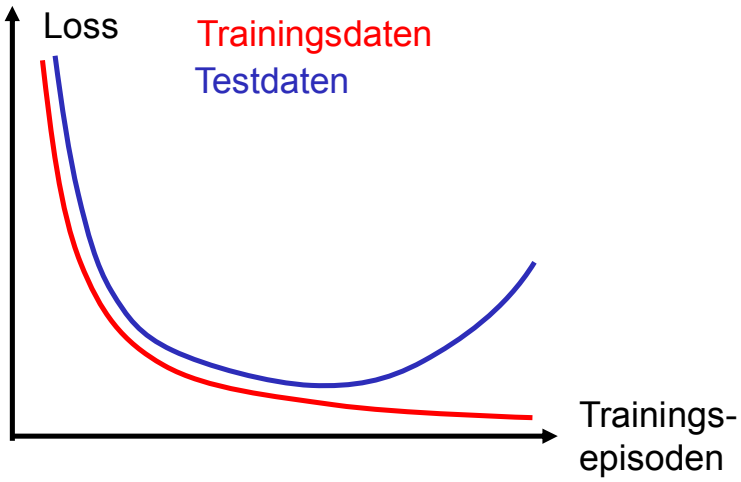



*Abbildung 6: Overfitting-Kurve in Abhängigkeit von Trainingsepisoden. Während die Fehlfunkton bei den Trainingsdaten auf bis zu 0 % fällt, steigt sie bei den Testdaten an [Meisel, Prof. Dr.-Ing. Andreas, 2018]*

## 2.6. Dropout

Bei dem oben genannten Verfahren werden nach jedem Trainingsschritt zufällig Neuronen ausgewählt und deren Ausgabewerte auf „Null“ gesetzt. Dies hat zur Folge, dass sie bei der nachfolgenden Rechnung keine wesentliche Rolle mehr spielen und die Gefahr von *Overfitting* nicht mehr besteht (Abb. 7) [Francois Chollet, 2018] .

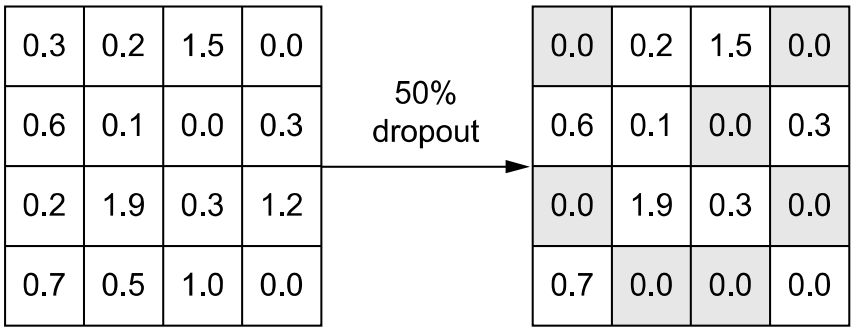



*Abbildung 7: Ausgabewert eines Layers ohne Dropout (links). Ausgabewert des Layers, bei dem 50 % der Neuronen auf „Null“ gesetzt wurden (rechts) [Chollet, 2018]*

## 2.7. Faltungsnetzwerk

*CNN* sind sogenannte Faltungsnetzwerke. Sie unterscheiden sich von Fully-Connected-Layers dahingehend, dass sie lokale Muster in Bildern speichern. Somit werden *CNN* insbesondere bei der Bildklassifizierung eingesetzt, da sie erheblich effizienter als ANN sind [Francois Chollet, 2018].

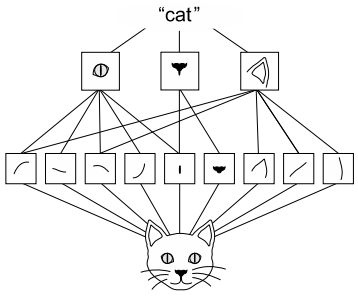


*Abbildung 8: Lokale Muster wie „Strähne“ oder „Haar“ werden erlernt, daraufhin mit „Augen“ oder „Ohren“ kombiniert und schließlich zu einem Objekt „Katze“ zusammengefügt [Francois Chollet, 2018]*

Die erlernbaren lokalen Muster werden auch Translationsinvarianten genannt. Der Vorteil bei Translationsinvarianten ist, dass die Muster unabhängig von ihrer Position im Bild immer erkannt werden. *CNN* erlernen die Muster auf eine systematische Vorgehensweise. So werden im ersten Abschnitt nur kleinere Muster wie Ränder oder Kanten erlernt. Die darauffolgenden Muster setzten sich immer aus den vorherigen zusammen. Auf diese Weise ist es dem *CNN* möglich, komplexere und abstraktere Konzepte zu erlernen und zu erkennen (Abb. 8) [Francois Chollet, 2018].

Der Zweck eines *CNN*s ist es, ein Bild, das eine große Anzahl an Parametern hat, anhand von Faltungen und Filtern Schritt für Schritt zu reduzieren. Am Ausgang wird aus einem 3D-Datenformat (Bild) ein 1D-Datenformat (Vektor), der am Neuron ausgegeben werden kann (Abb. 9). Die *Convolution* ist eine Schicht, basierend auf Neuronen. Dabei unterscheidet sie sich mit dem *Full connection* dadurch, dass jedes Neuronen nicht mit allen Neuronen auf der nächsten Schicht verbunden ist. Das Subsampling ist ein Poolingverfahren, das im Kapitel 2.8 erklärt wird [Francois Chollet, 2018].

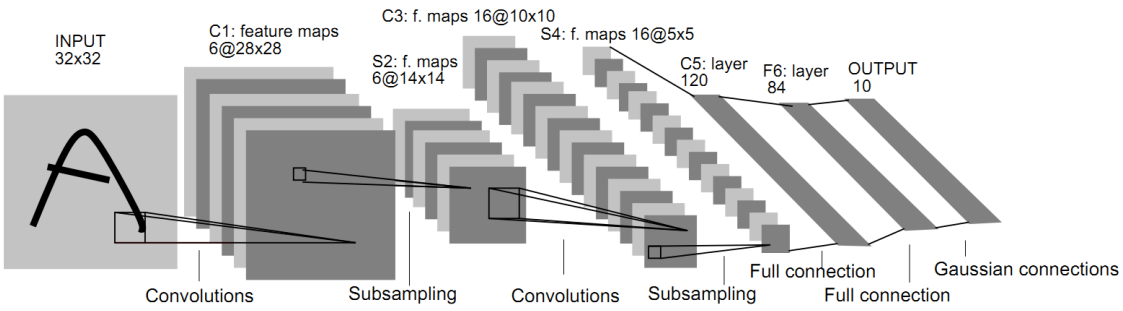



*Abbildung 9: Das Bild wird am Input eingespeist und erreicht den Ausgang über das Subsampling, die Convolution und die Full Connection. Am Ende werden aus einem 32x32-Bild nur zehn Klassen ausgegeben [Deshpande, 2017]*

## 2.8. Pooling

In dieser Thesis werden die Poolingverfahren Max-Pooling und Average-Pooling angewendet. Beim Falten wird üblicherweise einen 2x2-Filter verwendet, der Pixel für Pixel über das Eingangsbild, auch als Feature-Map bezeichnet, gleitet. Daraus resultiert das Ausgangsbild, auch Response-Map genannt.

Bei *Max-Pooling* wird nur der Maximalwert aus dem 2x2-Feld übernommen (Abb. 10). Der entscheidende Vorteil des *Poolings* besteht darin, die Parameter zu verkleinern (eng. Down-Sampling) und das *Overfitting* zu reduzieren. Dies wird dadurch gewährt, dass der 2x2-Filter nicht über jeden Pixel, sondern über jeden zweiten Pixel gleitet, während beim *Average-Pooling* jeweils der Mittelwert aus dem 2x2-Filter entnommen wird [Francois Chollet, 2018].

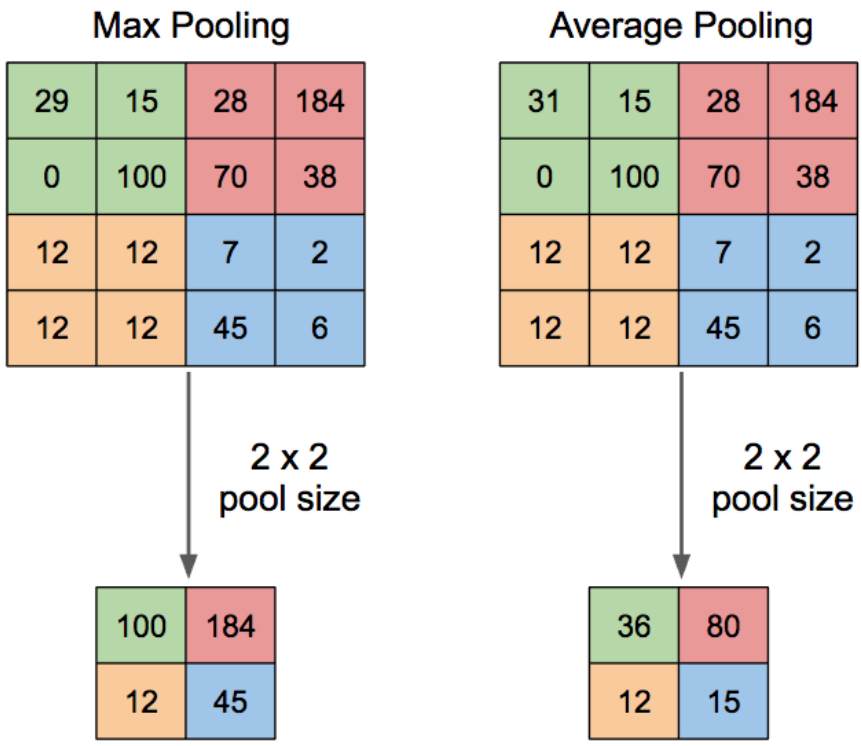



*Abbildung 10: Darstellung eines Max-Pooling-Layers mit der Filtergröße von 2 x 2 (links). Die Response-Map des Average-Poolings entnimmt jeweils den Mittelwert aus dem 2x2-Filter (rechts) [Deshpande, 2017]*

# 3. Screening

Das dritte Kapitel beschreibt detailliert das Screening als Verfahren sowie das Ziel, die Vorteile und die Auswirkung des Resultats für die Teilnehmer. Dabei ist die Zielgruppe Teilnehmer, die zum Zeitpunkt der Untersuchung gesund sind, keine diagnostizierte Erkrankung und keinen Krankheitsverdacht aufweisen können. Darüber hinaus wird verdeutlicht, für wen sich ein Screening lohnen wird. Dabei wird explizit auf die Kriterien eins Screenings eingegangen.

## 3.1. Definition-Screening

Laut Definition sind Screening-Verfahren Untersuchungen mit dem Ziel, asymptomatische Teilnehmer, in Bezug auf ihre Zielerkrankung in hoher und

geringer Wahrscheinlichkeit einzuteilen. Dabei laufen Screening-Verfahren in der Regel zweistufig hab. Das erste Verfahren ist sehr sensitiv und teilt Teilnehmer in Bezug auf ihrer Zielerkrankung in positiven- und in negativen Befunden ein. Aufgrund der hohen *Sensitivität* des Verfahrens, werden mehrere Teilnehmern einem positiven Befund zugeordnet. Für die endgültige Abklärung des ersten Befundes wird im zweiten Schritt ein weiteres Verfahren eingesetzt. Das zweite Verfahren hat eine höhere Aussagekraft, da es nicht nur den ersten Befund bestätigt oder auszuschließen, sondern weil der Arzt die Diagnose stellen kann. Durch das zweite Verfahren werden nun Erkrankte in richtig positiv Befunde und Gesunde in falsch positiv Befunde unterteilt. Bestätigt sich der Verdacht des ersten Befundes, wird anschließend die Therapie vorgeschlagen [Spix, Claudia; Blettner, Maria, 2012].

## 3.2. Ziel des Screenings

Jedes Screening-Programm versucht im Sinne der Früherkennung die Erkrankung in ihrem jüngsten Stadium zu erkennen, um die höchsten Erfolgsaussichten bei der darauffolgenden Behandlung zu erlangen. Das heißt, es verschlechtert sich die Prognose auf Heilung, wenn sich der Tumor in einem fortgeschrittenen Zustand befindet. In der Abb. 11 wird der zeitliche Verlauf der Krankheit dargestellt. Die Krankheit beginnt an einem bestimmten, aber meist nur schwer nachweislichen Zeitpunkt. Zu diesem Zeitpunkt sind weder Screening-Programme noch konventionelle Untersuchungen in der Lage, atypische Zellen zu erkennen. Somit wäre das Testergebnis bei einer klassischen Untersuchung sowie im Screening zu diesem Zeitpunkt negativ. Screening-Verfahren sind erst ab der präklinischen Phase imstande, z. B. atypische Zellen, die sich nun zu einem Tumor entwickelt haben, zu erfassen. Die präklinische Phase ist das Intervall, in der der Erkrankte auch ohne Früherkennung klinisch positiv auf die Zielerkrankung diagnostiziert werden kann. Dabei hängt diese Phase immer von der Krankheit und vom Individuum ab und endet bei den ersten erkennbaren Symptomen. Ein Screening-Programm könnte anhand eines

Beispiels wie der Mammographie nachfolgenden Gesichtspunkten ablaufen [Spix, Claudia; Blettner, Maria, 2012]:

- das Untersuchungsverfahren im ersten Schritt definieren ( z.B. Mammographie),
- das Alter der Teilnehmern eingrenzen (z.B. 50–69 Jahre),
- einen Zeitplan erstellen, nach dem die Personen untersucht werden sollen (z.B. alle zwei Jahre).

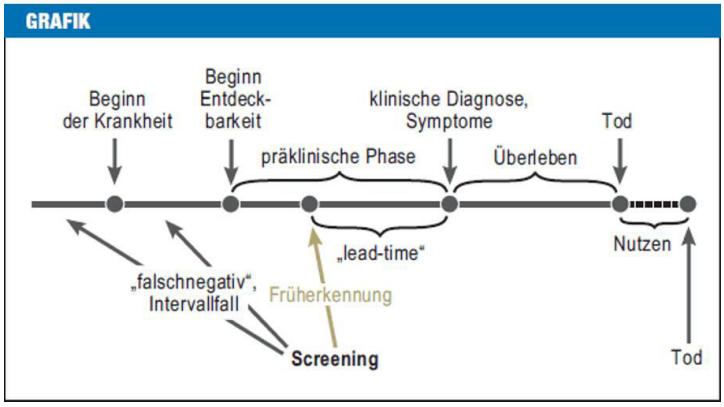



*Abbildung 11: Der theoretische Krankheitsverlauf von Beginn der Krankheit bis zum Tod des Patienten [Spix, Claudia; Blettner, Maria, 2012]*

Ziel eines Screening-Programms ist in erster Linie die Früherkennung der Zielerkrankung. Daraus resultierend sollte durch die vorgeschlagene Therapie eine längere Lebenszeit erzielt werden, die dem vorzeitigen Screening zu verdanken ist. Der Erfolg für Screening-Programmen wird durch die Reduktion der Mortalität definiert [Spix, Claudia; Blettner, Maria, 2012].

## 3.3. Eigenschaften des Screenings

In Deutschland umfasst die gesetzliche Vorsorge derzeit die Krebserkrankungen der Brust, des Darms, der Haut, des Gebärmutterhalses und der Prostata [Bundesministerium für Gesundheit, 2018]. In Studien werden folgende Kriterien für ein Screening beschrieben. So muss die

Erkrankung viele Menschen betreffen, damit sie ein erhebliches Problem darstellt. Ebenfalls sollte das neu eingeführte Screening zeit- und kostengünstig sein ( [Giersiepen K, Hense HW, Klug SJ, Antes G, Zeeb H, 2007] und [Eur J Cancer, 2000]). Ergänzend muss das gewählte Screening valide, risikoarm und von der Bevölkerung akzeptiert sein [Spix, Claudia; Blettner, Maria, 2012].

Die Validität wird durch die *Sensitivität* und die *Spezifität* beschrieben. Auch ein hoher *positiver prädiktiver Wert* ist wünschenswert. Ein erhöhter *Sensitivität-Wert* bedeutet weniger falsche negative und ein hoher *Spezifität-Wert* weniger falsche positive Befunde. Die genannten Kriterien sind notwendig für einen Erfolg, dennoch sind sie nicht hinreichend. Ein Screening-Programm, das alle genannten Bedingung erfüllt, muss nicht zwangsläufig erfolgreich sein. Dafür werden für jedes Screening-Programm individuelle wissenschaftliche Studien verlangt, die weitere Kriterien prüfen müssen [Spix, Claudia; Blettner, Maria, 2012].

## 3.4. Resultate des Screenings

Die Mehrheit der Teilnehmer, die an einem Screening-Programm teilnimmt, ist nicht von der Erkrankung betroffen. Dennoch tragen alle Teilnehmer die gleichen Risiken, die sich durch das Screening ergeben. Einen Nutzen aus dem Screening haben nur die Teilnehmer, die die Erkrankung besitzen und gleichzeitig einen positiven Befund erhalten. Wiederum nicht jeder Erkrankter der einen positiven Befund erhalten hat, profitiert von diesem Screening. Als Erkrankter profitiert man aus dem Screening nur, wenn sich die Lebenszeit aufgrund des früheren Befundes verlängert hat. Dabei muss sichergestellt werden, dass kein konventionelles Verfahren die Zielerkrankung ebenfalls erkannt hätte. Des Weiteren befinden sich unter den positiv getesteten auch Erkrankte, deren Gewinn an Lebenszeit unabhängig vom Screening genauso hoch wäre. Folglich hat auch diese

kleine Gruppe kein Nutzen an der Teilnahme an einem Screening [Spix, Claudia; Blettner, Maria, 2012]. Daraus wird ersichtlich, dass nur eine sehr kleine Gruppe an Teilnehmern, ein wirkliches Nutzen aus dem Screening-Programm hat.

Therapien entwickeln sich und zeigen mit der Zeit bessere Ergebnisse auch gegen fortgeschrittene Tumoren. Dadurch reduziert sich der Nutzen eines Screenings. Besonders kritisch sind Teilnehmern betroffen, die die Zielerkrankung haben, aber keine beeinträchtigenden Symptome aufweisen, sodass die Zielerkrankung nie ausbrechen wird. Ein klassisches Beispiel aus dem Alltag ist das Prostatakarzinom. Besonders Männer im höheren Alter haben ein Prostatakarzinom, das weder die Lebensqualität verschlechtert noch die Lebenszeit des Erkrankten verkürzt. Nur bei einer geringen Anzahl entstehen Komplikationen bzw. werden die Betroffenen in ihrer Lebensqualität eingeschränkt [Djulbegovic M, et al., 2010]. Hierbei ist die Rede von Überdiagnose. Folglich sterben diese Personen nicht aufgrund ihrer Zielerkrankung, sondern sterben eines natürlichen Tods mit der Zielerkrankung.

Laut dem Evaluationsbericht des Mammographie-Screening-Programms in Deutschland wird nur eine geringe Anzahl der Teilnehmer richtig positiv getestet. So sind von 1000 Teilnehmern durchschnittlich acht Teilnehmer wirklich positiv [Dr. Daniela Malek & Peter Rabe, 2008]. In dem Evaluationsbericht wird dagegen nicht erwähnt, wie viele von den acht Personen tatsächlich nicht an Brustkrebs starben.

Die offensichtlich größte Teilnehmergruppe sind die Gesunden, die auch richtig als negativ erkannt werden. Auch diese Gruppe profitiert vom Screening, da sie die Bestätigung vom Arzt erhalten, dass sie gesund sind. Bei einem falschen positiven Befund folgt die Abklärung, die je nach Zielerkrankung invasiv und auch risikobehaftet sein kann. Dabei ist die psychische Belastung der Teilnehmer nicht außer Acht zulassen, die bis zur Abklärung des falschen positiven Befundes anhält. Im Screening finden sich

immer mehr falsche positive als richtige positive Befunde [Spix, Claudia; Blettner, Maria, 2012].

Der Hauptkritikpunkt von Screening-Gegnern ist, dass bei einem frühzeitigen Erkennen einer Krankheit die Lebenszeit sich nicht verändert z.B. Prostatakarzinom. Hingegen die daraus resultierende Behandlung die Lebensqualität enorm verschlechtern kann.

# 4. Durchführung des Versuches

Das folgende Kapitel beschreibt die Vorbereitung des online bereitgestellten Datenpaktes. Das Datenpaket wurde von Forschern aus Bethesda (USA) vorbereitet und anschließend in der online Plattform Kaggle.com für Forschungszwecke im Bereich Deep Learning zu Verfügung gestellt. Die Trainingsdaten, die in Kaggle veröffentlicht wurden, werden von dem Radiologen Oakden-Rayner aus Australien stark kritisiert und für nicht zulässig geklärt [Luke Oakden-Rayner, 2017]. Zudem wird der Vorgang von der Vorbereitung bis zum Ergebnis der Versuche beschrieben. Die relevanten Aussagen und Thesen, die auch das Verfahren dieser Arbeit beeinträchtigen können, werden in diesem Kapitel 4.4 kurz erläutert und beschrieben. Zudem werden die Versuche 1-3 beschrieben und ausgewertet.

## 4.1. Vorbereitung

Im Rahmen dieser Arbeit wird ein *CNN* mit veränderten Hyperparametern aufgebaut. Dabei werden drei Versuche unternommen. Alle Versuche werden mit ähnlichen Verteilung an positiven und negativen Befunden trainiert. Das hierbei genutzte gesamt Datenpaket stammt aus der ACS-Datenbank des National Institute of Health Clinical Center in Bethesda und wurde auf Kaggle.com für Forschungszwecke veröffentlicht [Anon., 2017]. Das Datenpaket besteht aus 112.120 Thorax-Röntgenbildern mit 15 verschiedenen

Klassen (14 Erkrankungen und eine Klasse ‚gesund'). Für das Bereitstellen dieser großen Anzahl an Röntgenbildern haben die Forscher die Hilfe von Natural-Language-Processing (NLP) genutzt. Das NLP hat mit einer *Accuracy* von über 90% aus den Radiologieberichten die Erkrankungen zu dem jeweiligen Röntgenbild zugeordnet. Die Röntgenbilder wurden somit nicht von einem Radiologen erneut visuell begutachtet.

Zuerst wird geprüft, wie hoch die *Accuracy* eines *CNNs* mit nur einem Hidden Layer ist. Dabei soll das *CNN* nur zwischen einer Klasse unterscheiden: negativ und positiv (gesund/*Kardiomegalie*). Im zweiten Verfahren wird das *CNN* moduliert. Außerdem wird Data-Augmentation eingesetzt, um Bilder, die sich ähneln, minimal zu verändern. Überdies wird das *CNN* mit einer Faltungsbasis vom Type *VGG16* ergänzt. Im letzten Versuch wird die Einstellung aus dem vorherigen Versuch übernommen mit dem Unterschied, dass nicht nur eine Klasse getestet wird, sondern alle 14 und anstatt von *VGG16* wird *mobileNet* benutzt.

Die *CNN*, die zum Einsatz kommen, benötigen eine hohe Rechenleistung, sodass die Anwendung eines GPUs erforderlich ist. Ansonsten würde eine *Epoche* mehrere Stunden dauern. Hierfür wurde eine Online Cloud von Google „Collaboratory" benutzt, die mit einer starken NVIDIA-Grafikkarte betrieben wird. Das *CNN* wird mittels Python realisiert und mit Hilfe der Bibliothek Keras größtenteils umgesetzt. Über eine Dropbox werden die Bilder hochgeladen.

Nahezu alle Versuche bestehen aus drei Abschnitten. Im ersten Abschnitt finden sich alle genutzten Bibliotheken sowie die Datenpakete, die in die Cloud geladen werden müssen. Im mittleren Abschnitt wird er gesamte Datenpaket aufgezeigt, aussortiert und für das *CNN* vorbereitet. Im letzten Abschnitt fängt das Trainieren sowie das Evaluieren an.

## 4.2. Datenpakete

Die Daten bestehen aus 112.120 Röntgenbildern, dabei ist nicht jede Klasse gleichmäßig vertreten. Es gibt 15 unterschiedliche Klassen, wobei „No Finding“ ein gesundes Röntgen-Thoraxbild bezeichnet. In Abb. 12 sind tabellarisch die Häufigkeit der einzelnen Erkrankungen zusehen. Außerdem soll darauf hingewiesen werden, dass die Bilder nicht nur aus einer, sondern aus multiplen Erkrankungen bestehen und es somit in Summe mehr als 112.120 Erkrankung sind. In Abb. 13 sind die Klassen in Abhängigkeit ihrer Häufigkeit dargestellt. Der hellere Blauton gibt die Anzahl der Bilder, die aus mehreren Erkrankungen bestehen. Während der dunkele Blauton nur aus einer Erkrankung besteht. Am häufigsten sind die negativen Befunde, die mit *No Finding* beschriftet sind. Am geringsten ist die Klasse *Hernie* vorhanden. Da *Hernie* verglichen mit den anderen Klassen in geringer Anzahl vorhanden ist, wird sie aus dem Datenpaket entfernt und nicht mittrainiert.

```
Klasse
Atelectasis          11559
Cardiomegaly          2776
Consolidation         4667
Edema                 2303
Effusion             13317
Emphysema             2516
Fibrosis              1686
Hernia                 227
Infiltration         19894
Mass                  5782
No Finding           60361
Nodule                6331
Pleural_Thickening    3385
Pneumonia             1431
Pneumothorax          5302
Name: Anzahl, dtype: int64
```

*Abbildung 12: In der ersten Spalte sind die Klassen, in der zweiten ihre Häufigkeit dargestellt*

Um das *Overfitting* zu minimieren, wird der gesamte Datenpaket in drei Datenpakete (Trainings-, Validations-, Testdaten) unterteilt. Mit dem

Trainingsdaten wird das *CNN* trainiert. Die Validation wird nach jeder *Epoche* durchgeführt und dient zur Überprüfung von *Overfitting* und *Underfitting*. *Overfitting* tritt auf, wenn sich das *CNN* zu stark nur an die Trainingsdaten anpasst. Während *Underfitting* auftritt, wenn sich das *CNN* noch gar nicht an die Trainingsdaten angepasst hat. Ebenso zeigt die Validierung, ob das Netzwerk einen Lernfortschritt gemacht hat. Das Ziel nach jedem Trainingsvorgang ist es, dass das *CNN* nun in der Lage ist, Bilder richtig zu klassifizieren, die es vorher noch nie gesehen hat.

Für die Evaluation werden nur die Testdaten genutzt. Obwohl beim Training das Datenpaket Validierung primär nicht mittrainiert wird, können bei jeder *Epoche* einige Informationen an das *CNN* überreicht werden. Dadurch besteht die Gefahr, dass eine Anpassung vonseiten des *CNN*s auf die Validierungsdaten stattfinden kann. Somit wird das Testpaket nur zum Schluss für das Evaluieren eingesetzt.

## 4.3. Erkrankungen

In diesem Kapitel wird auf die pathologischen Erkrankungen eingegangen. Alle Bilder wurden aus einer Perspektive, frontal vom Thorax, aufgenommen. Außerdem bestehen die Bilder aus weiblichen sowie aus männlichen Brustkörben. In der Tab. 1 sind alle Erkrankungen und die dazugehörigen Erklärungen dargestellt.

*Tabelle 1: Auflistung aller Erkrankungen, die in dem Datenpaket vorkommen*

| **Erkrankung** | **Beschreibung** |
|---|---|
| Atelektase | in einem Teilabschnitt der Lunge herrscht Belüftungsdefizit, dadurch kein Gasaustausch zwischen O2 und CO2. |
| Kardiomegalie | symbolisiert ein Vergrößertes Herz |

| | |
|---|---|
| Ödem (Edema) | entsteht durch das Austreten von Blutflüssigkeit aus den Kapillargefäßen |
| Pleuraerguss (Effusion) | eine deutlich erhöhte Flüssigkeitsansammlung in der Pleurahöhle |
| Emphysema | wird bei einer irreversiblen Überblähung der kleinsten Strukturen der Alveolen in den Lungen diagnostiziert |
| Fibrose | bildet eine Bindegewebeschicht zwischen den Alveolen und den Blutgefäßen |
| Hernie | wenn Inhalte aus dem Darm über eine Verletzung der Muskelwand in die Lunge geraten |
| Infiltration | das Eindringen von fester oder flüssiger Substanz in die Lunge |
| Mass | Gewebssubstanzen, die einen größeren Durchmesser als drei cm haben |
| Knoten (Nodle) | runde Gewebssubstanzen, die kleiner als drei cm Durchmesser sind |
| Pneumonie | Lungenentzündung |
| Pneumothorax | eine Luftansammlung im Brustkorb |
| Pleuraverdickung | eine Verdickung der Pleurawand |
| Lungenkonsolidierung | ansammlung von Flüssigkeit anstatt von Luft in der Lunge |

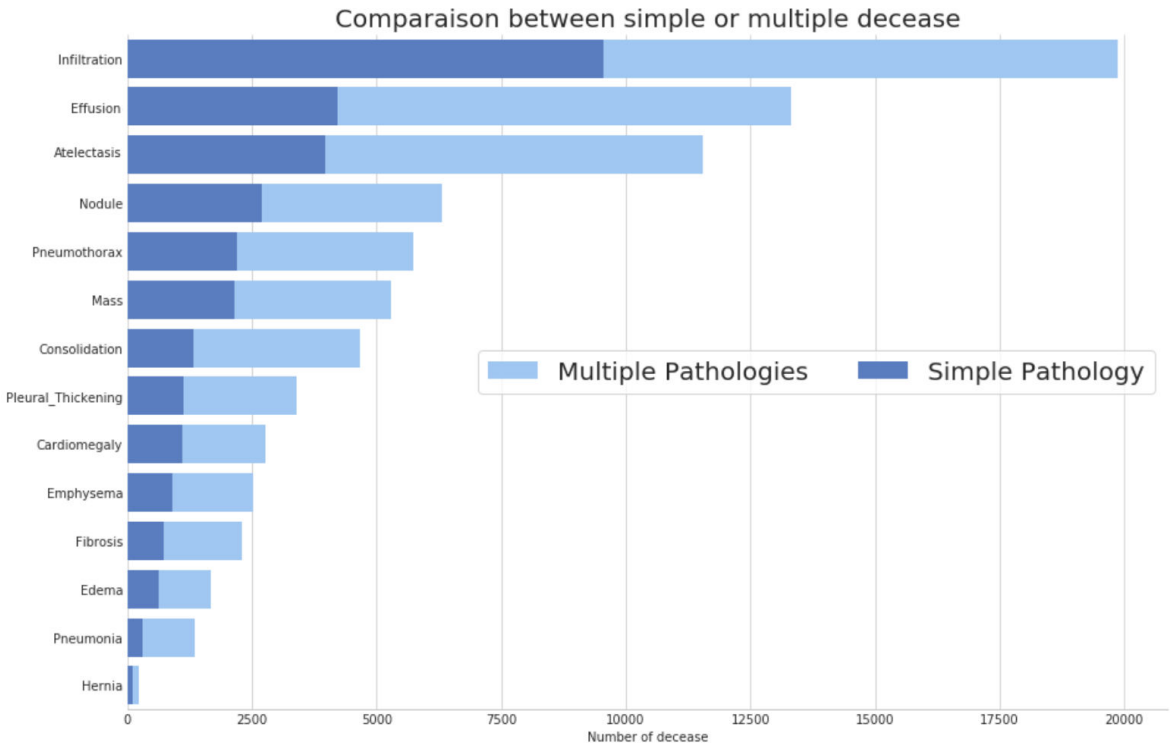



*Abbildung 13: Die Grafik zeigt die einzelnen Klassen und deren Häufigkeit. Dabei wird letztere unterteilt in einzelne und multiple Erkrankung [Shrikant, 2018]*

## 4.4. Problematik der Röntgenbilder

In diesem Kapitel werden die Thesen und die kritischen Äußerungen von dem Radiologen Rayner dargestellt. Sie sind sehr gravierend und Verzerren das Ergebnis dieser Arbeit, da sich die Versuche (s. Kapitel Versuch 1,2 und 3) hauptsächlich auf diese Daten basieren.

Die allgemeine Problematik bei Röntgenbildern besteht darin, dass ein Röntgenbild oftmals mehrere Erkrankungen wiedergibt. Einige sind subtil und atypisch, andere wiederum eindeutig. Der Befund eines Radiologen aus einem Röntgenbild ist keine objektive und sachliche Bildbeschreibung aller vertretbaren Erkrankungen, sondern sein Ziel ist es, dem überweisenden Arzte nützliche und relevante Informationen bereitzustellen, die zum Erfolg der Therapie führen [Luke Oakden-Rayner, 2017].

Somit werden irrelevante Informationen, die Indizien für andere Erkrankung sind, aber für den Verlauf der Therapie nicht relevant sind, nicht genannt bzw. entfernt. Somit können zwei Berichte von unterschiedlichen Radiologen zu demselben Röntgenbild ein unterschiedliches Ergebnis darstellen. Des Weiteren werden in klassischen Radiologieberichten die Erkrankungen beschrieben, aber nie namentlich genannt, da der überweisende Arzt ansonsten zu einer bestimmten Behandlungstherapie gezwungen wäre. Es sind noch viele weitere Faktoren vorhanden, die den Radiologiebericht verzerren. Deshalb ist es nicht sinnvoll, über NLP die Erkrankungen aus den Radiologieberichten zu entnehmen [Luke Oakden-Rayner, 2017].

Der Grund für diese Überlegung ist, dass das National Institutes of Health - Team (NIH), dass die Datenpakete veröffentlicht hat, die Erkrankung aus den Radiologieberichten über NLP gefiltert hat. Eine weitere visuelle Überprüfung, ob das ausgewählte Röntgenbild tatsächlich der Erkrankung entspricht, wurde versäumt und vernachlässigt. Nach einer kurzen Überprüfung durch den Radiologen zeigten sich bereits mehrere Differenzen [Luke Oakden-Rayner, 2017]. Somit kann angenommen werden, dass mehrere Röntgenbilder mit der falschen Erkrankung befundet wurden.

Dies wird anhand eines Beispiels mit der Erkrankung Fibrose dargestellt (Abb. 14). Dabei untersuchte der Radiologe 18 Bilder, die mit der Erkrankung Fibrose vermerkt wurden, und markierte sie mit zwei unterschiedlichen Farben, Rot und Orange. Nach seiner Meinung gehören die Bilder, die rot markiert sind, nicht zur Erkrankung Fibrose. Bei der Markierung in Orange sei es nicht eindeutig zu erkennen. Außerdem sind auf den Bildern sowohl Pleuraergüsse als auch Konsolidierungen zu erkennen. Ob daraus Fibrose entsteht, ist nicht sicher auf den Bildern zu erkennen. Das größte Problem, das der Radiologe hierbei sieht, besteht darin, dass Fibrose, Pneumonie und Emphysema klinisch und nicht bildgebend befundet werden. Es ist generell nach medizinischen Aspekten fragwürdig, den Radiologiebericht in seiner Gesamtheit zu ignorieren und nur das Endresultat bzw. die Enderkrankung

für die Verarbeitung zur Bildklassifizierung zu entnehmen [Luke Oakden-Rayner, 2017].

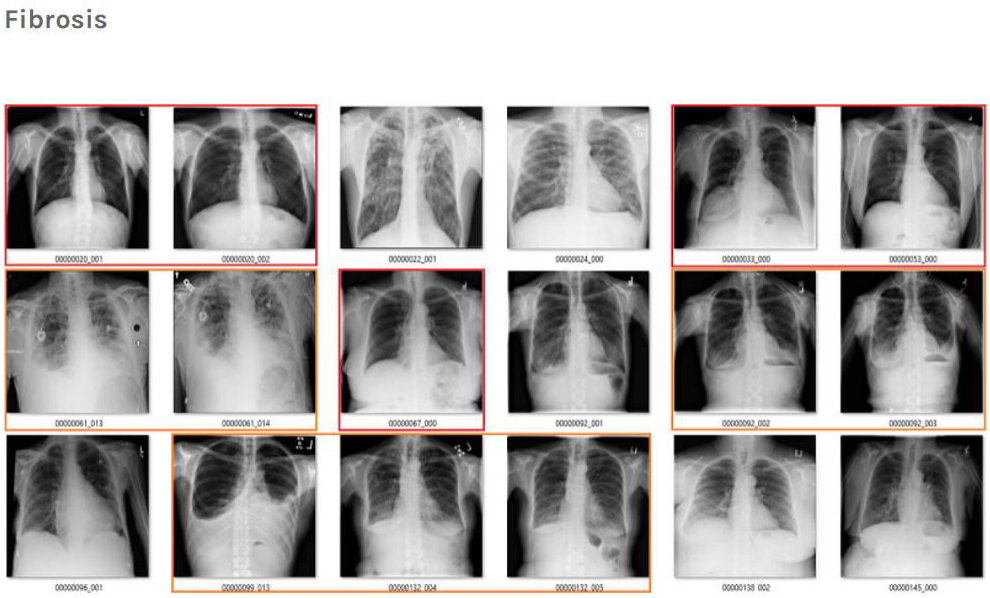



*Abbildung 14: 18 Röntgenbilder, die der Radiologe Oakden-Rayner auswertete. Fragwürdige Befunde wurden orange markiert, während rote Markierungen falsche Befunde anzeigen [Luke Oakden-Rayner, 2017]*

Weiterhin sind dem Radiologen in einigen Bilder Objekte aufgefallen, die für die Befundung der Bilder nur stören und ggf. das *CNN* mit falschen Informationen füttern. Auch hierfür wird ein Beispiel angezeigt. In einigen Bildern des Pneumothorax hat der Radiologe festgestellt, dass sie die Resultate des *CNN* erheblich verzerren können (Abb. 15). Die in Grün markierten Bilder deuten auf die Erkrankung Pneumothroax. Nur beim orangenfarbenen Bild ist die Erkrankung laut dem Radiologen nicht eindeutig zu erkennen. Das erschreckende an allen Bildern in Abb. 15 ist, das zusätzlich eine Thoraxdrainage in allen Bildern abgebildet ist. Das *CNN* lernt die Muster aus den Trainingsdaten. Beinhalten nun alle oder die meisten Pneumothoraxbilder eine Drainage, wird das *CNN* Pneumothoraxbilder nur in Kombination mit einer Drainage erkennen. Dies würde definitiv eine Verzerrung der Befunde verursachen [Luke Oakden-Rayner, 2017].

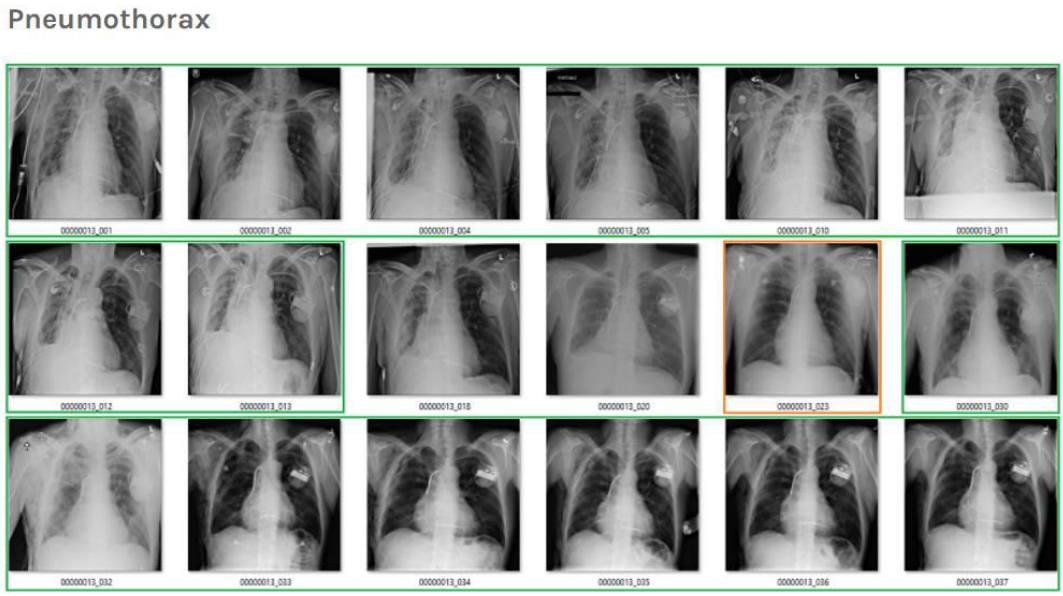



*Abbildung 15: Röntgenbilder mit der Erkrankung Pneumothorax. Alle zeigen Thoraxdrainagen an [Luke Oakden-Rayner, 2017]*

## 4.5. Verarbeitung der Datenpakete

Wie aus Abb. 12 zu sehen ist, handelt es sich bei der Anzahl der Bilder um eine große ungleichmäßige Verteilung. Um dieser entgegenzuwirken und die Resultate besser vergleichen zu können, müssen ähnliche Maßstäbe in allen 3 Versuche aufgebaut werden. Zunächst wird für den Versuch 1 und 2 nur eine Klasse *Kardiomegalie* zum Auswerten genutzt. Beim dritten und letzten Versuch werden alle Klassen verwendet mit Ausnahme der *Hernie*, da die Klasse im Verhältnis zu den anderen Klassen um ein Vielfaches geringer ist.

## 4.6. Versuch 1

Der gesamte Datenpaket der im Versuch 1 verwendet wird besteht aus 11104 Röntgenbilder. Davon gehören 2776 zu der Erkrankung *Kardiomegalie* (Abb. 12). Das gesamte Datenpaket besteht aus ca. einem Viertel aus Bildern mit der Erkrankung *Kardiomegalie*. Drei Viertel bestehen aus Bildern, die nicht die Erkrankung *Kardiomegalie* beinhalten. Obwohl es in der Praxis üblich ist, eine außerordentlich hohe Anzahl an negativen Befunden zu haben, wird in

dieser Arbeit ein Szenario aufgegriffen, bei dem die Anzahl der negativen Befunde um das Dreifache erhöht ist.

In Abb. 16 ist nun zu sehen, dass Verhältnis zwischen positiven und negativen Befunden im gesamten Datenpaket vor der Vorbereitung der Daten. Während Abb. 17 das Verhältnis nach der Vorbereitung darstellt. Mit dieser Anzahl der Bildern wird nun im gesamten Versuch weitergearbeitet.

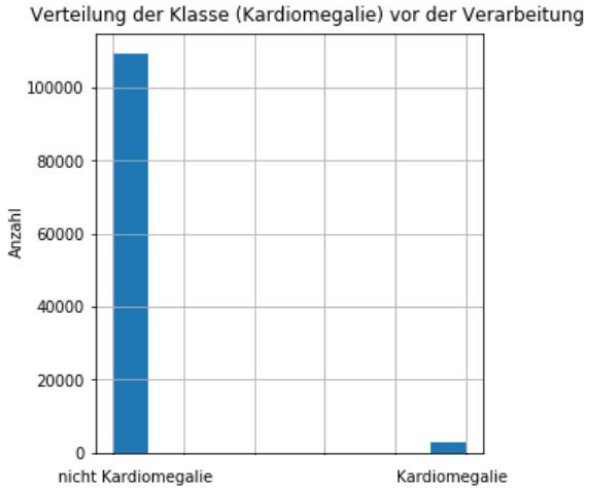



*Abbildung 16: Verhältnis von negativen und positiven Befunde anhand der Erkrankung Kardiomegalie*

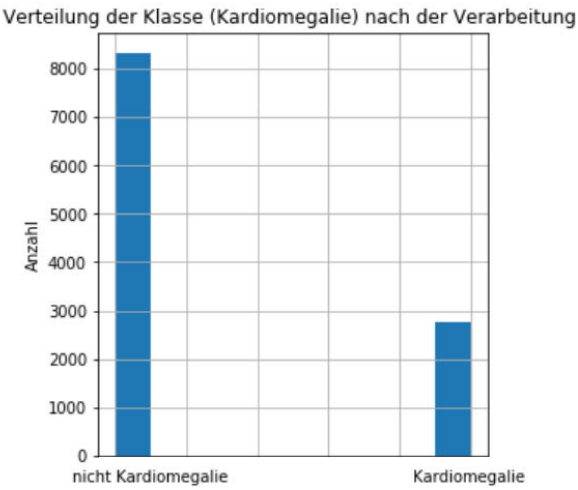



*Abbildung 17: Das Verhältnis nach von positiven und negativen Befunde nach der Verarbeitung*

Alle relevanten Informationen über die Bilder sind in Tab. 2 zusammengefasst. Bei der Auswahl der Bilder musste ich prüfen, ob in der Spalte *Kardiomegalie True* oder *False* Bilder enthalten sind. Dieser Schritt zeigte mir, ob der zusammengestellte Datenpaket explizit die Bilder enthält, die ich für den Versuch 1 nutzen möchte. Aus der Spalte *Finding Labels* können die ursprünglichen Erkrankungen entnommen werden.

*Tabelle 2: Relevante Informationen der Thorax-Röntgenbilder*

| | Image Index | Finding Labels | Patient Age | path | Cardiomegaly |
|---|---|---|---|---|---|
| **0** | 00005204_001.jpg | No Finding | 6 | Downloads\Kaggle\Images1024_JPG\00005204_001.jpg | False |
| **1** | 00027236_003.jpg | No Finding | 67 | Downloads\Kaggle\Images1024_JPG\00027236_003.jpg | False |
| **2** | 00017448_000.jpg | Cardiomegaly\|Mass\|Pleural_Thickening | 57 | Downloads\Kaggle\Images1024_JPG\00017448_000.jpg | True |
| **3** | 00015044_004.jpg | Atelectasis | 42 | Downloads\Kaggle\Images1024_JPG\00015044_004.jpg | False |
| **4** | 00019847_001.jpg | No Finding | 31 | Downloads\Kaggle\Images1024_JPG\00019847_001.jpg | False |
| **5** | 00019551_004.jpg | Cardiomegaly\|Infiltration | 30 | Downloads\Kaggle\Images1024_JPG\00019551_004.jpg | True |
| **6** | 00013911_004.jpg | Effusion\|Nodule | 63 | Downloads\Kaggle\Images1024_JPG\00013911_004.jpg | False |
| **7** | 00027389_009.jpg | Cardiomegaly\|Effusion | 65 | Downloads\Kaggle\Images1024_JPG\00027389_009.jpg | True |
| **8** | 00013052_007.jpg | Atelectasis\|Consolidation\|Pneumothorax | 41 | Downloads\Kaggle\Images1024_JPG\00013052_007.jpg | False |
| **9** | 00018802_000.jpg | Cardiomegaly | 25 | Downloads\Kaggle\Images1024_JPG\00018802_000.jpg | True |
| **10** | 00004809_001.jpg | Cardiomegaly | 37 | Downloads\Kaggle\Images1024_JPG\00004809_001.jpg | True |

Die Bibliothek *sklearn.model_selection* ermöglicht eine zufällige Verteilung der Daten in den Trainings- und in den Testdaten. Die Trainingsdaten bestehen aus 80 % des gesamten Datenpaketes. Dementsprechend hat das letztere nur 20 %, die später für das Evaluieren benötigt werden. Außerdem ermöglicht die Funktion *random_state,* dass immer die gleiche Verteilung von positiven und negativen Befunden bei jedem wiederholten Vorgang gegeben ist. Anschließend bestehen die Trainingsdaten aus 8.883 Röntenbildern und die Tesdaten aus 2.221 Bildern.

Im Versuch 1 werden die Bilder nicht über Data-Augmentation verarbeitet, somit sieht die Ausgabe der Bilder wie in Abb. 18 aus:

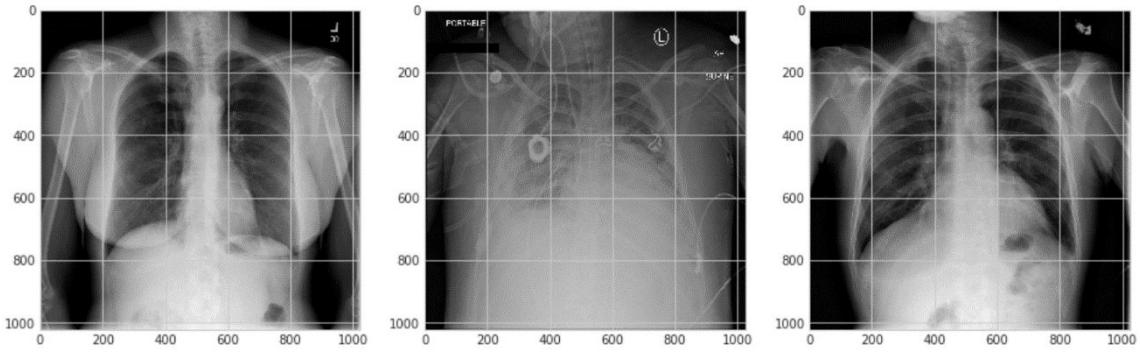



*Abbildung 18: Bilder aus dem Datensatz, der für das CNN verwendet wird*

Nachdem die Aufteilung der Daten erfolgt ist, müssen die Bilder in passende Datenformate konvertiert werden. Dieser Schritt ist deshalb relevant, da das *CNN* nur bestimmte Datenformate am Eingang akzeptiert. Datenformat sind mehrdimensionale Numpy-Arrays, die anhand ihrer Größen in ein einheitliches Schema umgewandelt werden. Das *CNN* soll am Eingang nur 4D- Datenformate mit einem Datenformat (*,128,128,1) aufnehmen. Die ersten beiden Zahlen in einem Datenformat stehen für die Höhe und Breite des Bildes. Die letzte Zahl signalisiert, die Farbkanäle des Bildes. RGB-Bilder bestehen aus 3 Farbkanäle während ein schwarz/weiß Bild nur ein Farbkanal besitzt. Der Stern ist der Platzhalter für die Anzahl der Bilder, die je nach Datenmenge variieren kann. In diesem Fall lautet der Datenformat der in das CNN integriert wird: (8883,128,128,1). Davor muss jedes Element im Array in den Wertebereich 0 bis 1 transformiert werden. Dies wird mithilfe der Normalisierung (6) durchgeführt.

$$x' = \frac{x - x_{min}}{x_{max} - x_{min}} \tag{6}$$

Im letzten Schritt wird das Grundgerüst des *CNNs* aufgebaut (Tab. 3). Im Eingangs-Layer wird ein 4D-Datenformat benötigt und über 32 Neuronen aufgeteilt. Mittels des Dropouts und MaxPooling2D wird der 4D-Tensor von Layer zu Layer verringert, bis er anschließend mit der Flatten() zum 1D-Datenformat umgewandelt wird. Dieser Schritt tritt immer nur im unteren Abschnitt auf, da ab diesem Punkt die Layer eines ANNs anfangen und deshalb die Parameter von einem 4D-Datenforamt in die eines 1D-Datenformat umgewandelt werden müssen. Im letzten Layer findet sich der Ausgangs-Layer (dense_16), der nur ein Neuron besitzt. Besitzt dieses Neuron nach der Aktivierungsfunktion *Sigmoid* einen Wert von 0, so wird das Bild vom *CNN* als negativ markiert, während es sich bei einer 1 um einen positiven Befund der *Kardiomegalie* handelt.

*Tabelle 3: „Summary" des aufgebauten CNNs. Das CNN besteht aus einem Eingangs-Layer (conv2d_15), einem Ausgangs-Layer (dense_16) und die übrigen Layer gehören zum Hidden Layer*

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d_15 (Conv2D)           (None, 128, 128, 32)      1184
_________________________________________________________________
max_pooling2d_15 (MaxPooling (None, 64, 64, 32)        0
_________________________________________________________________
dropout_10 (Dropout)         (None, 64, 64, 32)        0
_________________________________________________________________
conv2d_16 (Conv2D)           (None, 64, 64, 64)        8256
_________________________________________________________________
max_pooling2d_16 (MaxPooling (None, 32, 32, 64)        0
_________________________________________________________________
flatten_8 (Flatten)          (None, 65536)             0
_________________________________________________________________
dense_15 (Dense)             (None, 128)               8388736
_________________________________________________________________
dropout_11 (Dropout)         (None, 128)               0
_________________________________________________________________
dense_16 (Dense)             (None, 1)                 129
=================================================================
Total params: 8,398,305
Trainable params: 8,398,305
Non-trainable params: 0
```

Tabelle 3 ist folgendermaßen interpretierbar:

**Layer (type):**

- Hier werden alle vertretbaren Layer im *CNN* angezeigt. Zusätzlich sind auch Werkzeuge zusehen, die die Parameter der Bilder reduzieren, um *Overfitting* zu verhindern.

**Output Shape:**

- In der mittleren Spalte ist das Datenformat zu sehen. *None* steht für Anzahl der Bilder, die dem *CNN* für das Antrainieren übergeben werden Die Neuronen Anzahl ist aus der letzten Zahl im Datenformat abzulesen.

**Param:**

- Die letzte Spalte zeigt die Parameter, die aus dem Datenformat aus der mittleren Spalte entstehen. Sie zeigen an wie viele Parameter im jeweiligen Layer trainiert werden. In summe werden in dem Versuch 1

In diesem Versuch wird das *CNN* in sechs *Epochen* mit den Trainingsdaten trainiert und wird mit den Validierungsdaten validiert. Diese werden kurz vor dem Trainingsvorgang aus den Trainingsdaten mit einem weiteren Verhältnis von 80:20 erzeugt. Für den Eingangs-Layer sowie den Hidden Layer wird die *ReLU*-Funktion als Aktivierungsfunktion gewählt. Diese hat den Vorteil gegenüber den *Sigmoid* Funktion, dass die Konvergenz des stochastischen Gradientenabfalls wesentlich höher ist. Der entscheidende Kritikpunkt an der *Sigmoid* Funktion ist, dass sie schnell zu *Overfitting* führt. Eine *Overfitting* hat zur Folge, dass das *CNN* sich sehr stark an die Trainingsdaten anpasst und somit nur noch Bilder aus dem Trainingsset klassifizieren kann. Bilder aus den Validierungsdaten werden sehr schlecht klassifiziert [Chaitanya Asawa, kein Datum].

Im Versuch 1 wird eine Unterscheidung zwischen zwei Klassen vorgenommen. Bei zwei Klassen ist die *binary_crossentropy* als Loss-Funktion am besten geeignet.

```
Train on 7106 samples, validate on 1777 samples
Epoch 1/6
7106/7106 [==============================] - 384s 54ms/step - loss: 0.3499 - acc: 0.8446 - val_loss: 0.5501 - val_acc: 0.7321
Epoch 2/6
7106/7106 [==============================] - 283s 40ms/step - loss: 0.3109 - acc: 0.8615 - val_loss: 0.5795 - val_acc: 0.7226
Epoch 3/6
7106/7106 [==============================] - 279s 39ms/step - loss: 0.2884 - acc: 0.8724 - val_loss: 0.5966 - val_acc: 0.7366
Epoch 4/6
7106/7106 [==============================] - 285s 40ms/step - loss: 0.2624 - acc: 0.8854 - val_loss: 0.6152 - val_acc: 0.7479
Epoch 5/6
7106/7106 [==============================] - 285s 40ms/step - loss: 0.2284 - acc: 0.9016 - val_loss: 0.6340 - val_acc: 0.7338
Epoch 6/6
7106/7106 [==============================] - 271s 38ms/step - loss: 0.1997 - acc: 0.9174 - val_loss: 0.6742 - val_acc: 0.7271
```

*Abbildung 19: Ergebnisse aus dem Trainingsverlauf im ersten Versuch*

*Loss* und *Accuracy* acc stehen für den Fehler und *Accuracy* in den Trainingsdaten, während *val* die Werte für die Validierung bezeichnet (Abb. 19). In der ersten Zeile ist die Anzahl der Trainingsbilder sowie der Validierungsbilder zu lesen. Die *Epoche* in diesem Versuch ist besonders kurz gehalten. Das liegt daran, dass das *CNN* schnell zu *Overfitting* führt. In der sechsten *Epoche* sind erste Anzeichen eines *Overfitting*s ersichtlich. Der *Loss* sinkt rapide ab, während die *Accuracy* aufgrund der Anpassung an die Trainingsdaten kontinuierlich steigt. Diese Eigenschaft spiegelt sich ebenfalls in den Validierungsdaten wider mit dem Unterschied, dass der

*Validierungs-Loss* zunehmend steigt und die *Validierungs-Accuracy* minimal sinkt. Dies führt zu dem Ergebnis, dass das *CNN* nur noch Bilder aus den Trainingsdaten richtig vorhersagen kann. Unbekannte Bilder, die in der Validierung vorkommen, werden schlechter bzw. falsch vorhergesagt. Es ist notwendig, diesen Vorfall zu verhindern.

Die *Accuracy* in der ersten *Epoche* ist bereits mit ca. 84 % relativ hoch. Dies wird dadurch begünstigt, dass die Bilder mit den negativen Befunden in den Trainingsdaten aus ca. Drei Viertel des gesamten Datenpaktes bestehen. Hier ist zu beachten, wenn ein *CNN* grundsätzlich jedes Bild als negativen Befund vorhersagt, ist bereits hohe *Accuracy* von 75 % erzielt worden.

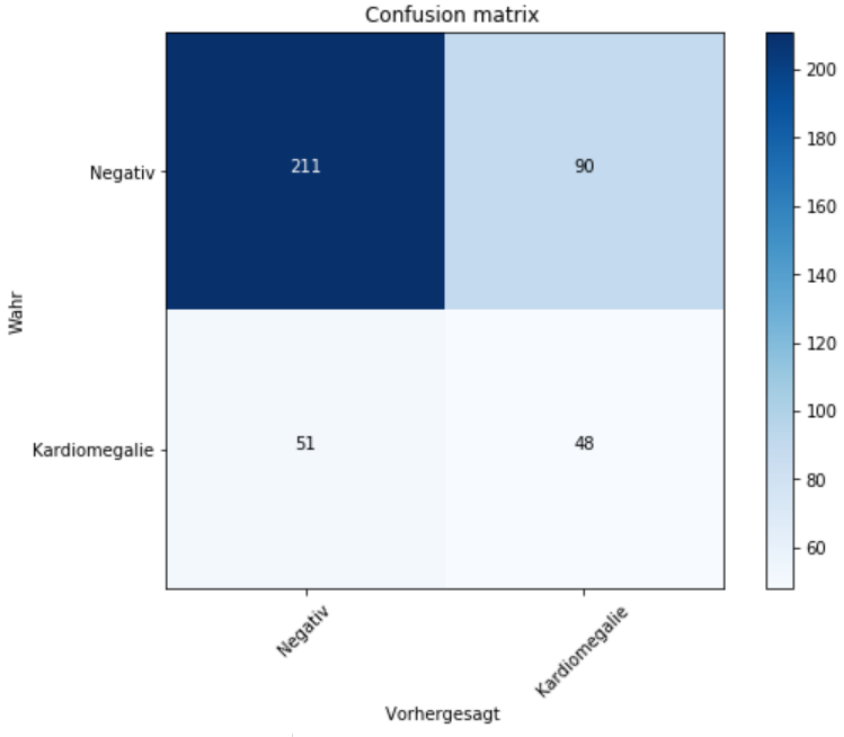



*Abbildung 20: Konfusionsmatrix im ersten Versuch. Hierbei werden die Stärken und Schwächen des CNN schnell deutlich. Richtige negative Diagnosen erkennt das CNN gut, während bei richtigen positiven noch Schwierigkeiten auftreten*

In der Regel wird ein *CNN* mithilfe von Testdaten evaluiert, um festzustellen, wie hoch die *Accuracy* des *CNN*s ist. Doch bei medizinischen Befunden wird nicht nur zwischen positiv und negativ unterschieden. Vielmehr gibt es richtig positiv bzw. -negativ und falsch positiv bzw. -negativ. Zur

Beobachtung dieses Sachverhaltes wird eine *Konfusionsmatrix* vewendet, die die Bilder in 4 unterschiedliche Klassen klassifiziert (Abb. 20).

Die horizontale Spalte in Abb. 20 zeigt die Klasse die das Testbild besitzt. Hingegen die vertikale Zeile die Vorhersage des *CNN*s darstellt. Für das Evaluieren werden 400 Testbilder entnommen. Diese teilen sich in 301 negative und 99 positive Befunde auf. Aus der *Konfusionsmatrix* ist abzulesen, wie das trainierte *CNN* diese 400 Bilder vorhersagt. Dabei werden 211 von 301 als richtig-negativ und nur 48 von 99 als richtig-positiv prognostiziert. Die Fehldiagnose lässt sich aus den falsch-negativen und falsch-positiven Vorhersagen bestimmen. Sie liegt bei 35 % und verfehlt somit das Ziel von unter 10 %. Das *CNN* erkennt dagegen viele Bilder als richtig negative, während es nur die Hälfte aller positiven Befunde als richtig vorhersagt.

Die Gründe für die hohe *Accuracy* bei den negativen Bildern liegen daran, dass einerseits Drei Viertel der Befunde negativ sind. Somit besteht eine höhere Wahrscheinlichkeit, dass bei einer negativen Vorhersage das Bild auch tatsächlich negativ ist. Anderseits ist das *CNN* durch die heterogene Verteilung an positiven und negativen Bildern in der Lage, negative Befunde viel besser vorherzusagen als positive.

Nun wird geprüft, inwiefern das *CNN* als Screening geeignet ist. Screening-Verfahren laufen immer zweistufig ab. In der ersten Stufe wird der Teilnehmer als positiv oder negativ befundet und in der zweiten Stufe findet die Abklärung des Befundes mit Hilfe von weiteren medizinischen Untersuchungen statt. Daraus lässt sich ableiten, dass ein Screening-Programm in erster Linie eine hohe *Spezifität* besitzen sollte. Dadurch werden den Teilnehmern weitere Untersuchungen, die eventuell schädigend sind, erspart. Darüber hinaus sollte das Screening-Programm einen hohen positiven prädiktiven Wert aufweisen. Dadurch lassen sich Fehldiagnosen bzw. falsche positive Befunde minimieren. Anschließend sei es auch das Ziel,

eine hohe *Sensitivität* zu erzielen. Da aber bei einem Screening nur ein kleiner Prozentualer Anteil erkrankt ist, ist es von Bedeutung das gesunde nach dem Versuch auch wieder als Gesund erkannt werden.

In Tab. 4 sind die Formeln für die Berechnung der *Spezifität*, *Sensitivität* sowie der *positiver prädiktiver Wert* wiedergegeben. Für den ersten Versuch erhält das *CNN* eine *Sensitivität* von 48 %. Verglichen mit der Mammographie, die bei 78 % liegt, ist dieses *CNN* noch ausbaufähig [Hans-Werner Hense, 2014]. Der *positiver prädiktiver Wert* liegt bei 35 %, während im Mammographie-Screening der Wert bei 15,4 % liegt. Das *CNN* würde bereits einen Mehrwert erbringen. Zwei Drittel der positiven Diagnosen würden sich bei der Abklärung nicht bestätigen lassen, während bei der Mammographie bereits mehr als vier Fünftel bei einer Abklärung negativ wären. Die *Spezifität* erreichte nur 70 % und auch in der Mammographie liegt sie wesentlich höher, bei 95 %. Somit ist dieses *CNN* ungeeignet für Screening-Programme.

In Abb. 21 ist die Receiver-Operating-Characteristic (ROC) des *CNNs* abgebildet. Die *ROC-Kurve* ist eine Bewertungsmatrix, die alle möglichen Klassifizierungsschwellen zwischen richtigen positiven und falschen positiven berücksichtigt. Vor allem zeigt die Grafik die Leistung des Klassifikators unter verschiedenen Klassifizierungsschwellen an. Die daraus resultierende Area-Under-the-Receiver-Operating-Characteristic-Curve (AUROC oder AUC) steht für die Fläche unterhalb der Funktion. Die Fläche gilt als Maß, wie hoch die *Accuracy* des Klassifikators arbeitet und lässt sich somit sehr optimal mit anderen AUC vergleichen.

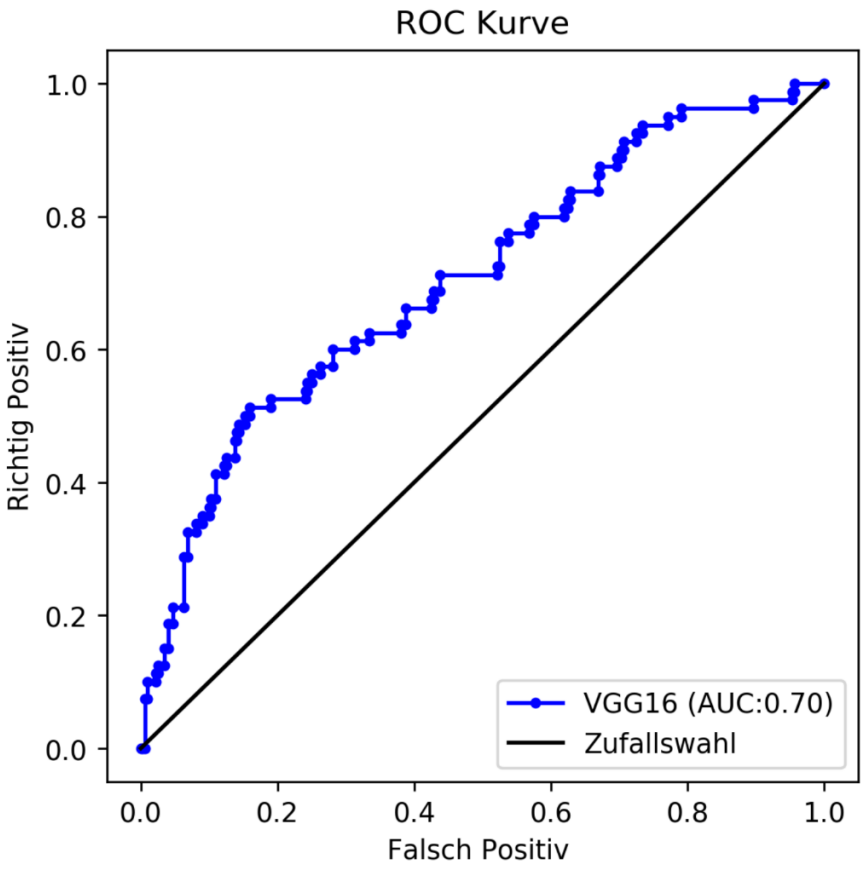



*Abbildung 21: Die ROC-Kurve zeigt das Verhältnis von richtigen positiven und falschen positiven Diagnosen vom CNN an*

*ROC-Kurve*n, die sich unterhalb der Diagonale befinden oder eine Fläche von 0,5 aufweisen, können als Zufallsereignisse interpretiert werden, während ein sehr guter Klassifikator eine Fläche von 1 aufweist.

Das University of Nebraska Medical Center nutzt ein Benotungsverfahren, das *ROC-Kurve*n von A bis F benotet (Tab. 5). Während A für exzellent steht, gelten *CNN* mit der Benotung F als fehgeschlagen [Thomas G. Tape, MD, kein Datum]. Mit einem AUC von 0.7 erhält dieses *CNN* die Note C für befriedigend.

*Tabelle 4: Formeln, die hinter den Bezeichnungen stehen* [Wilfried Bautsch, 2010]

| **Kriterium** | **Formeln** |
|---|---|
| Fehldiagnosen | $= \frac{Falsch_{pos} + Falsch_{neg}}{Total_{neg+pos}}$ |

| | |
|---|---|
| Positiver prädiktiver Wert | $= \frac{Richtig_{pos}}{Richtig_{pos} + Falsch_{pos}}$ |
| Sensitivität | $= \frac{Richtig_{pos}}{Richtig_{pos} + Falsch_{neg}}$ |
| Spezifität | $= \frac{Richtig_{neg}}{Total_{neg}}$ |

Das erste *CNN* hat einen positiven prädiktiven Wert von 35 %. Die *Sensitivität* liegt bei 48 % und die *Spezifität* liegt bei 70 %. Mit diesem *CNN* ist es möglich, mit einer Wahrscheinlichkeit von 35 % Erkrankte auch als positiv zu erkennen. Da die Fehldiagnose-Rate nicht unter 10% liegt, kann der erste Versuch die Forschungsfrage nicht erfüllen.

*Tabelle 5: Benotung für CNN im medizinischen Bereich*

| | |
|---|---|
| 1–0,9 | exzellent (A) |
| 0,9–0,8 | gut (B) |
| 0,8–0,7 | befriedigend (C) |
| 0,7–0,6 | mangelhaft (D) |
| 0.6–0,5 | durchgefallen (F) |

Alle wichtigen Eigenschaften im Versuch 1 sind aus der Tabelle 6 zu entnehmen. Die auch gleichzeitig als eine kurze Zusammenfassung zu sehen ist.

*Tabelle 6:Eigenschaften des CNNs im Versuch 1*

| Eigenschaften | Werte |
|---|---|
| Fehldiagnose | 35% |

| | |
|---|---|
| Trainings Accuracy nach letzter Epoche | 91,4% |
| Kontrollkriterien | Positiver prädiktiver Wert: 35%<br>*Sensitivität*: 48%<br>*Spezifität*: 70% |
| ROC-Kurve | Note C mit 0,70 AUC |
| *Konfusionsmatrix* | richtig-negativ :211<br>richtig-positiv : 48<br>falsch-negativ : 51<br>falsch-positiv : 90 |
| Epochen | 6 |
| Klassen | *Kardiomegalie*= positiv<br>nicht *Kardiomegalie*=negativ |
| Verteilung | Trainingsdaten: 7106<br>Validierungsdaten: 1776<br>Testdaten: 2221 |
| Loss-Funktion | Binary Crossentropy |
| Optimizer | SGD |
| Aktivierungsfunktion | Am Ausgangs-Layer Sigmoid<br>ansonsten ReLU |
| Overfitting Werkzeuge | Dropout und Max-Pooling |
| Trainierte Parameter | 8,4 Millionen |
| Keine Data-Augmentation | --- |

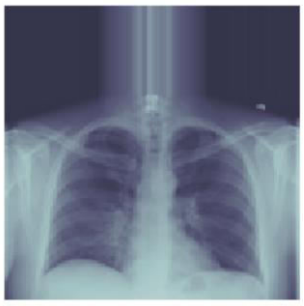
## 4.7. Versuch 2

Die Datenauswertung im Versuch 2 ist der im Versuch 1 sehr ähnlich. So wird das gleiche Verhältnis zwischen dem Trainings-, Validierungs- und dem

Testpaket wie im Versuch 1 erhoben. Dagegen kommt in diesem Versuch die Data-Augmentation zum Einsatz. Zunächst ist zu sagen, dass die Data-Augmentation nur auf die Trainingsdaten ausgeübt wird. Die Validierungs- und Testdaten werden hingegen verschont. Das *CNN* soll beim Evaluieren nicht modifizierte Bilder erkennen sollen, da auch in der Realität die Bilder nicht modifiziert sind. Dies wird verursacht, indem aus einem Bild mehrere Variationen geschaffen werden, z. B. durch Rotieren, Scheren oder Zoomen. Somit unterscheidet sich das Bild immer vom Original ab. Infolgedessen werden im Versuch 2 keine neuen Bilder erzeugt, sondern es wird lediglich jedes Bild modifiziert. Jedes Bild im Trainingspaket wird über die Data-Augmentation modifiziert, aber es werden keine neuen Bilder erzeugt, sondern nur durch die modifizierten Data-Augmentation Bilder ersetzt. Dadurch werden einerseits keine Duplikate dem *CNN* zum Trainieren übergeben, anderseits wird das *Overfitting* gehemmt.

Für die Data-Augmentation werden sieben Varianten benutzt, die in Tab. 7 dargestellt sind.

*Tabelle 7: Sieben Varianten, wie die Bilder für die Data-Augmentation verzerrt werden*

| | |
|---|---|
| Height-Shift  | Height-Shift wird das Bild horizontal verzerrt |
| Brightness-Range | gibt den Wert für die Helligkeit des Bildes an |

| | |
|---|---|
|  | |
| Rotation<br>  | Der Wert gibt den Grad der Rotation wieder |
| Width-Shift<br>  | ähnlich zum Height-Shift; nur das Bild wird vertikal verschoben |
| Zoom-in<br>  | wird in das Zentrum des Bildes hineingezoomt. |
| Zoom-out | |

| | |
|---|---|
|  | wird aus dem Zentrum hinausgezoomt. |
| Shear-Range  | wird durch den in Grad angegeben Wert geschoren. |

Bei einer starken Bildverarbeitung können Informationen, die sich am Rand befinden und wichtig für das Erkennen der Erkrankung ist, verloren gehen. Deshalb werden in diesem Versuch und auch im nächsten Versuch nur minimale Veränderungen vorgenommen, die auf dem ersten Blick nicht erkennbar sind. Dementsprechend wurden zur Veranschaulichung nur in der Tab. 7 die Bildverarbeitungsvarianten verstärkt vorgenommen. Im Versuch 2 werden die Bilder über die Data-Augmentation nach Zufallsprinzip verarbeitet. Währenddessen zeigt die Abb. 22 die Bilder, die im Versuch 2 während des Versuchsvorgang mittels Data-Augmentation modifiziert werden.

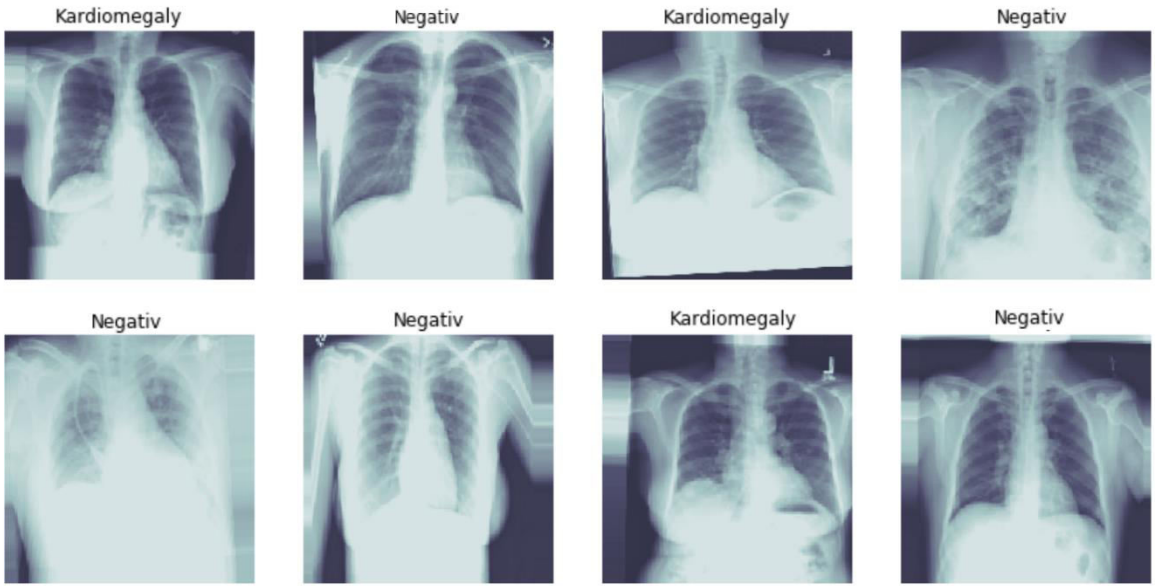



*Abbildung 22: Röntgenbilder, die nach der Data-Augmentation verarbeitet werden. Die Bilder werden zufällig verändert*

Nachdem die Data-Augmentation erfolgreich durchgeführt wurde, werden die Bilder für das *CNN* vorbereitet. Die Bildgröße wird von 128 x 128 auf 512 x 512 vergrößert. Folglich wird die *Batch_size* unterschiedlich eingestellt. Für das Antrainieren des CNNs wird eine *Batch_size* von acht Bilder verwendet, während für das Validieren und Evaluieren eine Größe von 400 gewählt wird. *Batch_size*s geben die Bilder an, die während eines Durchlaufs antrainiert werden sollen. Also werden beim Trainingspaket pro Durchgang acht Bilder durch das *CNN* trainiert. Dieser Vorgang wiederholt sich solange, bis alle Bilder im Trainingspaket das *CNN* einmal überquert haben. Der Vorteil einen geringen *Batch_size-Größe* ist, dass der Rechner weniger Speicher für das Trainieren benötigt und somit der Prozess des Antrainierens schneller abläuft. Hingegen leidet die *Accuracy* bei einem geringen *Batch_size*. Aus diesem Grund wird die *Batch_size* beim Validieren und beim Evaluieren angehoben.

Im Unterschied zum ersten Versuch besteht dieses *CNN* aus zwei Komponenten: aus einer Faltungsbasis mit der Bezeichnung *VGG16* (Tab. 9) und einem Klassifikator (Tab. 8). *VGG16* ist ein Netzwerk, das für die Bildklassifizierung eingesetzt wird. Dieses Netzwerk besteht aus einer Reihenfolge mit mehreren Layern, Dropout und Global Average (Abb. 23).

Das *VGG16* unterschiedet sich von dem *CNN* aus dem ersten Versuch dahingehend, dass dieses bereits im Voraus mit einer hohen Anzahl an Bildern von Katzen und Hunden trainiert wurde. Ein *CNN* erzeugt in den Anfangs-Layern nur lokale und allgemeine Feature-Maps wie Ränder, Farben oder Texturen. Somit ist es möglich eine Faltungsbasis zu nehmen, die mit anderen Daten trainiert wurden.

*Tabelle 8: Das CNN besteht aus einem VGG16-Model und dem letzten Ausgangs-Layer*

| Layer (type) | Output Shape | Param # |
|---|---|---|
| vgg16 (Model) | (None, 4, 4, 512) | 14714688 |
| Model (Model) | (None, 1) | 138690 |

Total params: 14,853,378
Trainable params: 137,154
Non-trainable params: 14,716,224

*Tabelle 9: Aufbau der Faltungsbasis VGG-16 im Versuch 2*

| Layer (type) | Output Shape | Param # | Connected to |
|---|---|---|---|
| feature_input (InputLayer) | (None, 4, 4, 512) | 0 | |
| batch_normalization_2 (BatchNor | (None, 4, 4, 512) | 2048 | feature_input[0][0] |
| conv2d_6 (Conv2D) | (None, 4, 4, 128) | 65664 | batch_normalization_2[0][0] |
| conv2d_7 (Conv2D) | (None, 4, 4, 32) | 4128 | conv2d_6[0][0] |
| conv2d_8 (Conv2D) | (None, 4, 4, 16) | 528 | conv2d_7[0][0] |
| average_pooling2d_2 (AveragePoo | (None, 4, 4, 16) | 0 | conv2d_8[0][0] |
| conv2d_9 (Conv2D) | (None, 4, 4, 1) | 17 | average_pooling2d_2[0][0] |
| conv2d_10 (Conv2D) | (None, 4, 4, 512) | 512 | conv2d_9[0][0] |
| multiply_2 (Multiply) | (None, 4, 4, 512) | 0 | conv2d_10[0][0]<br>batch_normalization_2[0][0] |
| global_average_pooling2d_3 (Glo | (None, 512) | 0 | multiply_2[0][0] |
| global_average_pooling2d_4 (Glo | (None, 512) | 0 | conv2d_10[0][0] |
| RescaleGAP (Lambda) | (None, 512) | 0 | global_average_pooling2d_3[0][0]<br>global_average_pooling2d_4[0][0] |
| dropout_3 (Dropout) | (None, 512) | 0 | RescaleGAP[0][0] |
| dense_3 (Dense) | (None, 128) | 65664 | dropout_3[0][0] |
| dropout_4 (Dropout) | (None, 128) | 0 | dense_3[0][0] |
| dense_4 (Dense) | (None, 1) | 129 | dropout_4[0][0] |

Total params: 138,690
Trainable params: 137,154
Non-trainable params: 1,536

Dagegen erzeugen die Layer im letzten Abschnitt abstrakte Konzepte. Dies passiert im Klassifikator. Aufgrund dieser Erkenntnis wird der Versuch 2 so umgestellt, dass die Anfangs-Layer, die die allgemeinen Informationen besitzen, mit dem *VGG16* ausgestattet wird.

Der Ausgabewert des *VGG16* wird dem Eingabewert des Klassifikators übergeben. Dieser besteht nur aus einem Layer bzw. es wird nur ein Layer hinzugefügt, der die Werte aus dem *VGG16* annehmen soll, verarbeitet und ausgibt (Tab. 8).

In diesem Fall läuft das *CNN* 15 *Epoche*n, da bei mehreren Versuchen sich bei dieser Anzahl von *Epoche*n die besten Wahrscheinlichkeiten des *CNNs* ergeben haben. Durch den Einsatz von Data-Augmentation wird ein anderer *fit_generator* zum Trainieren des *CNN* benutzt. Dadurch kommen weitere neue Einstellung hinzu. *steps_per_epoch* steht für eine Iteration über den Trainingsverlauf und wird mit der Anzahl der gesamten Bilder durch die *Batch_size* gebildet. Bei einer Gesamtanzahl von 7.772 Bilder und einer

gewählten *Batch_size* von 8 wäre ein *steps_per_epoch*-Wert über 800 optimal. Dennoch ist ein solch hoher *steps_per_epoch* nur mit viel Rechenleistung möglich. Da diese aus technischen Gründen zum Ende der Bachelorthesis nicht mehr zu Verfügung stand, wird nur ein *steps_per_epoch* von 100 gewählt. Es werden ähnliche Hyperparameter verwendet wie im vorherigen Versuch.

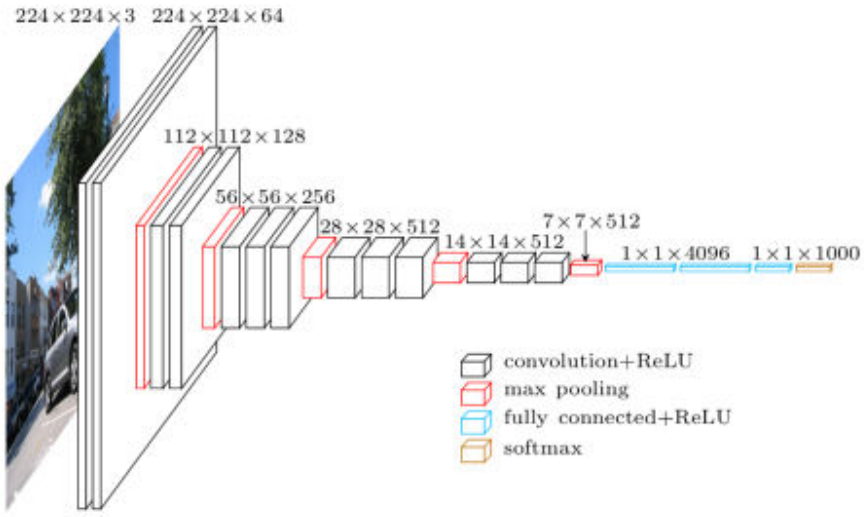



Abbildung 23: Grundstruktur eines *VGG16* bestehend aus mehreren Layern *[Davi Frossard, 2016]*

Der Trainingsverlauf (Abb. 24) zeigt eine leichte Senkung der *Loss*-Funktion. Auch die *Accuracy* ist leicht gestiegen. Die Validierungsdaten zeigen ein ähnliches Verhalten. Hier ist nun erkenntlich, dass das *CNN* ungesättigt ist auch *Underfitting*. Dies lässt sich aus der *Validerung-Accuracy* und der *Accuracy* erkennen. Die *Accuracy* in den Validierungsdaten liegt viel höher als in den Trainingsdaten. Dadurch kann die Annahme gemacht werden, dass das *CNN* noch weitere *Epoche*n trainiert werden muss.

```
Epoch 1/15
100/100 [==============================] - 216s 2s/step - loss: 0.5683 - binary_accuracy: 0.7300 - val_loss: 0.5195 - val_binary_accuracy: 0.7650
Epoch 2/15
100/100 [==============================] - 191s 2s/step - loss: 0.5428 - binary_accuracy: 0.7438 - val_loss: 0.5094 - val_binary_accuracy: 0.7625
Epoch 3/15
100/100 [==============================] - 194s 2s/step - loss: 0.5170 - binary_accuracy: 0.7630 - val_loss: 0.4975 - val_binary_accuracy: 0.7700
Epoch 4/15
100/100 [==============================] - 190s 2s/step - loss: 0.5324 - binary_accuracy: 0.7437 - val_loss: 0.5009 - val_binary_accuracy: 0.7450
Epoch 5/15
100/100 [==============================] - 193s 2s/step - loss: 0.5255 - binary_accuracy: 0.7463 - val_loss: 0.4883 - val_binary_accuracy: 0.7700
Epoch 6/15
100/100 [==============================] - 190s 2s/step - loss: 0.5426 - binary_accuracy: 0.7419 - val_loss: 0.4978 - val_binary_accuracy: 0.7525
Epoch 7/15
100/100 [==============================] - 190s 2s/step - loss: 0.5015 - binary_accuracy: 0.7704 - val_loss: 0.4909 - val_binary_accuracy: 0.7775
Epoch 8/15
100/100 [==============================] - 193s 2s/step - loss: 0.5237 - binary_accuracy: 0.7467 - val_loss: 0.4776 - val_binary_accuracy: 0.7750
Epoch 9/15
100/100 [==============================] - 192s 2s/step - loss: 0.5207 - binary_accuracy: 0.7629 - val_loss: 0.4604 - val_binary_accuracy: 0.7800
Epoch 10/15
100/100 [==============================] - 192s 2s/step - loss: 0.4923 - binary_accuracy: 0.7613 - val_loss: 0.4549 - val_binary_accuracy: 0.7875
Epoch 11/15
100/100 [==============================] - 193s 2s/step - loss: 0.4886 - binary_accuracy: 0.7654 - val_loss: 0.4550 - val_binary_accuracy: 0.7900
Epoch 12/15
100/100 [==============================] - 192s 2s/step - loss: 0.5015 - binary_accuracy: 0.7575 - val_loss: 0.4361 - val_binary_accuracy: 0.7825
Epoch 13/15
100/100 [==============================] - 192s 2s/step - loss: 0.4738 - binary_accuracy: 0.7753 - val_loss: 0.4585 - val_binary_accuracy: 0.7875
Epoch 14/15
100/100 [==============================] - 193s 2s/step - loss: 0.4829 - binary_accuracy: 0.7667 - val_loss: 0.4298 - val_binary_accuracy: 0.7975
Epoch 15/15
100/100 [==============================] - 190s 2s/step - loss: 0.4866 - binary_accuracy: 0.7687 - val_loss: 0.4408 - val_binary_accuracy: 0.8100
```

*Abbildung 24: Trainingsverlauf des CNNs. Daraus wird ersichtlich, dass das CNN unter Underfitting leidet*

Zur Übersicht wird erneut eine *Konfusionsmatrix* abgebildet (Abb. 25). Wie auch im Versuch 1 wird die *Konfusionsmatrix* mit den Testdaten durchgeführt. Von den 302 negativen Befunden wurden 284 als richtig negativ prognostiziert, im Gegensatz dazu nur 29 Bilder von 97 als richtig positiv prognostiziert. Die falschen positiven sind mit 19 Bilder prognostiziert. Dagegen ist die Kategorie falsch negativ mit 68 Bildern recht hoch und hinterlässt den Eindruck, dass das *CNN* aufgrund der einseitigen Trainingsdaten, Bilder eher negativ vorhersagt. Der positiver prädiktiver Wert für diesen Versuch liegt bei 60 %, was eine Steigerung gegenüber im Versuch 1 ist. Die *Sensitivität* liegt hier nur bei 30 %. Dies wiederum ist gegenüber dem Wert im Versuch 1 schlechter. Die *Spezifität* liegt in diesem Fall bei 93 % und nähert sich dem Wert der Mammographie.

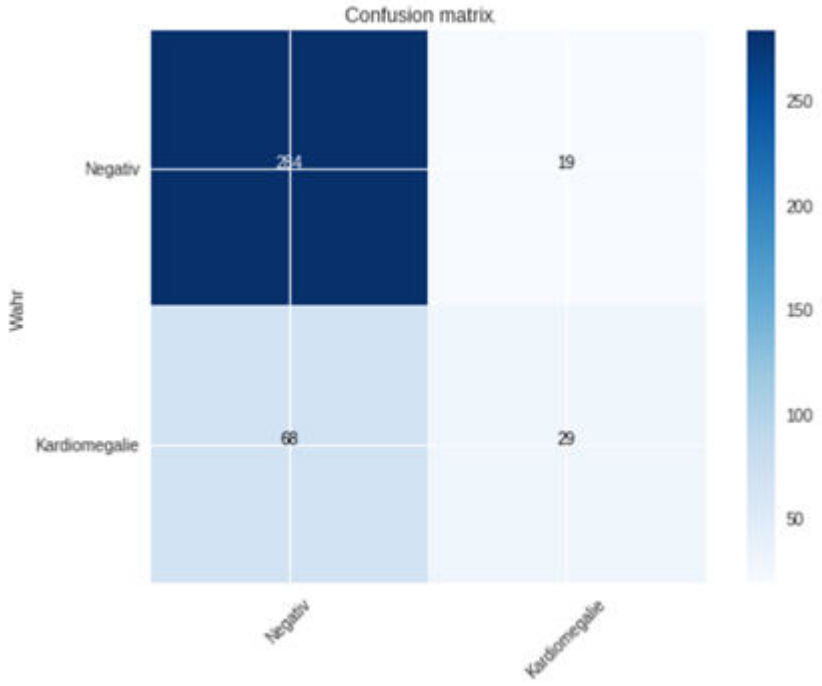



*Abbildung 25: Aus der Konfusionsmatrix wird ersichtlich, dass auch hier die richtig negative Bilder deutlich besser erkannt werden als die richtig positiven*

Das *CNN* zeigt bei einer positiven Vorhersage mit einer Wahrscheinlichkeit von ca. zwei Dritteln, dass es sich um einen richtigen positiven Befund handelt. Dieser Wert ist wesentlich höher als im ersten Versuch und somit ist die Wahrscheinlichkeit für einen falschen positiven Befund minimiert. Dieser Wert ist auch wesentlich höher als beim Mammographie-Screening (aktuell 15,4 %).

Für eine bessere Verständlichkeit zwischen den Kontrollkriterien wird nun anhand eines Rechenbeispiels die Unterschiede zwischen beiden Screening-Ergebnissen verdeutlicht. Wenn 100 Teilnehmer an einem Mammographie-Screening teilnehmen, ergeben sich anhand der *Sensitivität* und des positiven prädiktiven Werts folgende Resultate: 78 Teilnehmer erhalten einen positiven Befund. Bei der Abklärung würden sich nur bei 12 Teilnehmern die Befunde positiv bestätigen lassen und 66 hätten einen falschen positiven Befund. Beim *CNN* im Versuch 2 würden sich folgende Werte zusammentragen: Bei der gleichen Anzahl von Teilnehmern würden nur 30 einen positiven Befund erhalten. Nach der Abklärung würden

hingegen 18 Teilnehmer einen richtigen positiven und nur zwölf einen falschen positiven Befund erhalten. Dabei wird deutlich, dass die Anzahl der positiven Befunde rückgängig ist. Dagegen sind 60 % aller positiven Befunde richtig positiv. Somit erkennt das *CNN* mehr positive Befunde richtig und weniger falsch positive Befunde. Allerdings ist die *Sensitivität* nicht besonders hoch ausgefallen. Da die Teilnehmer im Screening über ein Intervall mehrmals untersucht werden, besteht noch die Möglichkeit, beim nächsten Termin vom Screening abgefangen zu werden. Die Fehldiagnosen liegen hier bei 22 %. Auch diese haben sich im Vergleich zum ersten Versuch erheblich verbessert.

Der AUC-Wert liegt bei 0,82 (Abb. 26). Er ist aufgrund der hohen *Spezifität* höher ausgefallen. Folglich ist die Benotungsskala für das zweite *CNN* die Note B für „gut“.

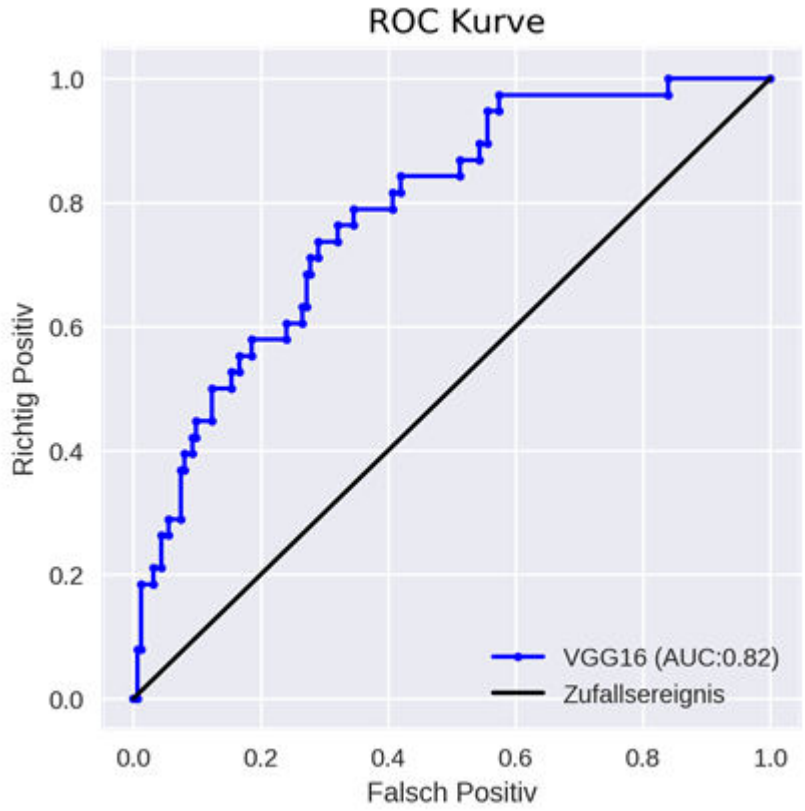



*Abbildung 26: ROC-Kurve des Versuchs 2*

Das zweite *CNN* zeigt Verbesserungen im Bereich der *Spezifität*, des positiven prädiktiven Werts und bei den Fehldiagnosen. Die Fehldiagnosen sind allerdings weiterhin hoch. Doch für ein Screening-Programm kann das zweite *CNN* gute Ergebnisse liefern. Somit werden gesunde Teilnehmer mit einer Wahrscheinlichkeit von 93 % als gesund befundet. Alle positiven Prognosen bestätigen sich mit ca. 66 % auch bei der Abklärung. Mit diesen Werten könnte ein neues Screening-Programm durch weitere klinische Studien evaluiert werden. Dennoch sind die Zahlen mit Vorsicht zu betrachten. Denn das *CNN* hat nur wenige positive Prognosen gemacht. Dadurch ist der positiver prädiktiver Wert höher ausgefallen als im vorherigen Versuch. Aufgrund der hohen richtigen negativen Prognosen erzielt das *CNN* gute Ergebnisse in der *Spezifität* und beim positiven prädiktiven Wert. Es müssten weitere Tests mit einer gleichmäßigen Verteilung der positiven und negativen Klassen durchgeführt werden.

Alle wichtigen Eigenschaften im Versuch 2 sind aus der Tab. 10 zu entnehmen. Die auch gleichzeitig als eine kurze Zusammenfassung zu sehen ist.

*Tabelle 10: Eigenschaften des CNNs im Versuch 2*

| Eigenschaften | Werte |
|---|---|
| Fehldiagnose | 32% |
| Trainings Accuracy nach letzter Epoche | 77% |
| Kontrollkriterien | Positiver prädiktiver Wert: 66%<br>*Sensitivität*: 30%<br>*Spezifität*: 93% |
| ROC-Kurve | Note B mit 0,82 AUC |
| Konfusionsmatrix | richtig-negativ :284<br>richtig-positiv : 29<br>falsch-negativ : 68 |

| | |
|---|---|
| | falsch-positiv : 19 |
| Epochen | 15 |
| Klassen | *Kardiomegalie*= positiv<br>nicht *Kardiomegalie*=negativ |
| Verteilung | Trainingsdaten: 7772<br>Validierungsdaten: 1666<br>Testdaten: 1666 |
| Loss-Funktion | Binary Crossentropy |
| Optimizer | Adam |
| Aktivierungsfunktion | Am Ausgangs-Layer Sigmoid<br>ansonsten ReLU |
| Overfitting Werkzeuge | Dropout und Average-Pooling |
| Trainierte Parameter | 137.000 |
| Data-Augmentation | Ja |
| Faltungsbasis | VGG16 |

## 4.8. Versuch 3

Im letzten Versuch wird nun das vollständige Datenpaket der NIH getestet. In Abb. 27 ist der unverarbeitete Datenpaket grafisch dargestellt. In dieser Grafik sind die 15 häufigsten Erkrankungen zu sehen. Die Klassen haben eine heterogene Verteilung und können sich während des Trainingsverlaufs negativ auf das *CNN* auswirken. Somit werden die Klassen angepasst. Aufgrund der Komplexität in den gesamten Datenpaket war eine bessere Anpassung der Häufigkeit der Bilder in den einzelnen Klassen nicht besser gelungen als in Abb. 28 zu sehen ist. Über eine Funktion werden alle Klassen, die weniger als 1.000 Bilder besitzen, für den Versuch 3 nicht mit ausgewertet. Dies würde die Klasse *Hernie* betreffen.

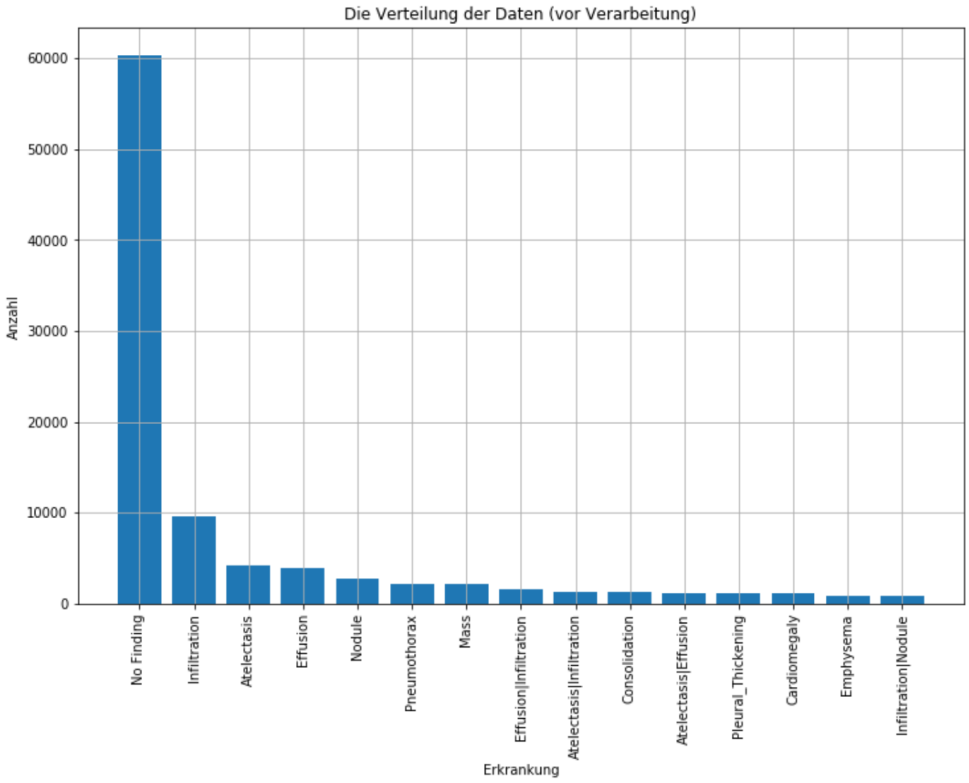



*Abbildung 27: Datensatz der 15 häufigsten Erkrankungen*

Außerdem wird die Klasse *No Finding* ebenfalls nicht ausgewertet, da bei diesem Versuch der Fokus auf die einzelnen Klassen gelegt wird. Der Datenpaket wird auf 40.000 herabgesetzt und die Verteilung.

Die Daten werden erneut in Trainings-, Validierung und Testdaten aufgeteilt. Die Trainingsdaten bestehen aus 30.000 Bildern, während die Validierungsdaten und die Testdaten aus jeweils 10.000 Bildern bestehen. Auch hier wird die *Stratify*-Funktion benutzt, um eine gleichmäßige Verteilung der Klassen zu erhalten. Anschließend wird die Data-Augmentation eingesetzt. Dabei werden die gleichen Variationen für eine Bildverarbeitung wie im Versuch 2 angewendet.

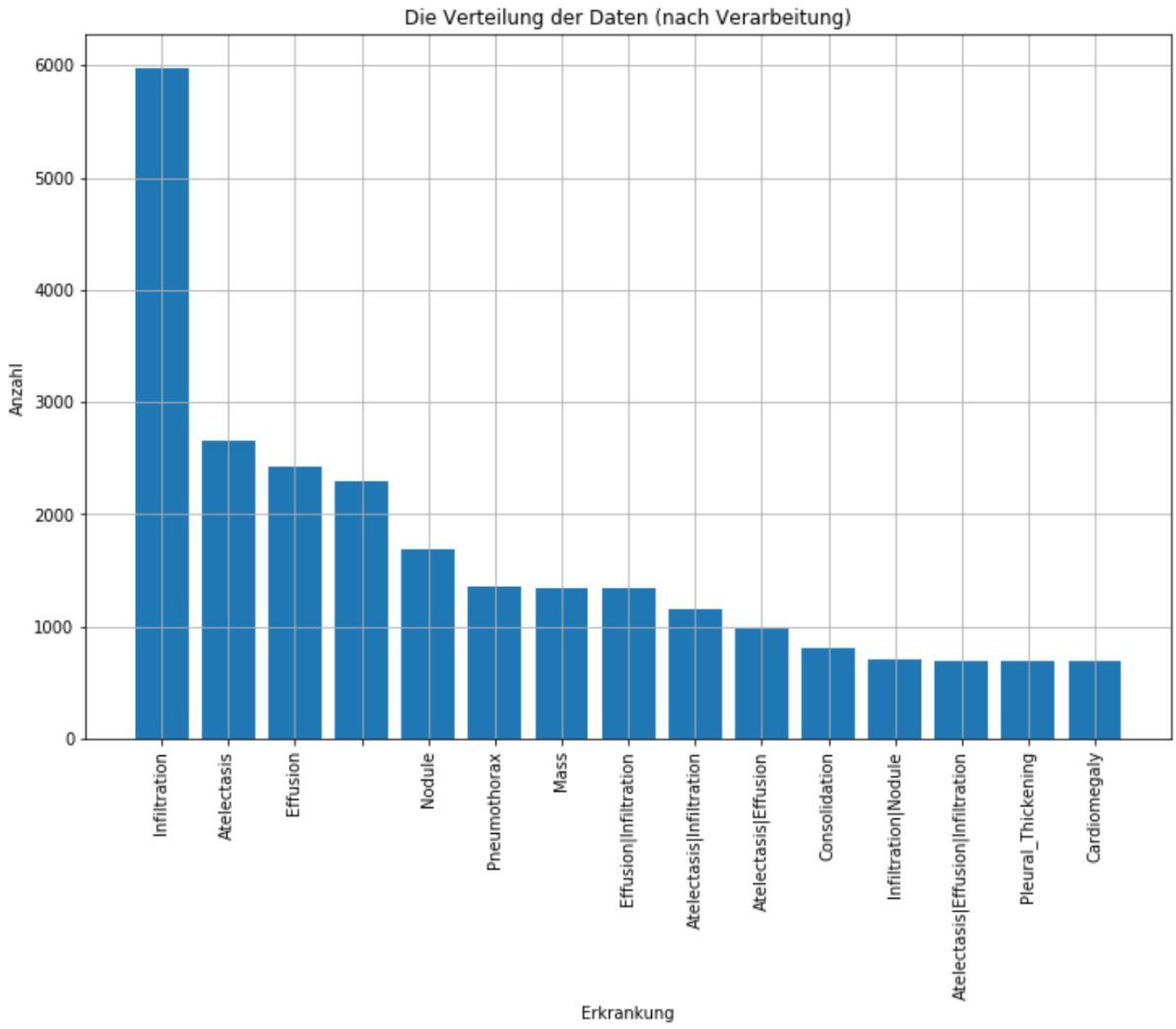



*Abbildung 28: Die häufigsten 15 Klassen nach der Verarbeitung*

Im Versuch 1 und 2 wurde mit nur einer binären Klasse gearbeitet und dabei das *CNN* moduliert. Im letzten Versuch soll nun das modulierte *CNN* aus dem zweiten Versuch übernommen werden und als Ausgang alle Klassen vorhersagen. Somit ist die Herausforderung des *CNN* nicht nur, bei einem Bild zwischen positiv und negativ zu entscheiden, sondern das Bild anhand der Merkmale, in die richtige Klasse einzuordnen.

Im Versuch 3 werden ähnliche Einstellungen wie im vorherigen verwendet mit dem Unterschied, dass nun nicht das *VGG16* als Faltungsbasis dient, sondern der *MobileNet*. *MobileNet* weist nur minimale Differenzen zu

*VGG16* auf. Dagegen ist die Entscheidung für die neue Faltungsbasis gefallen, da *MobileNet* schneller läuft als *VGG16*.

Nach dem Training mit dem MobileNet werden noch ein Hidden Layer und zwei Dropouts verwendet (Tab. 11). Dadurch soll das *Overfitting* gehemmt werden. Es werden 13 Klassen am Ausgang ausgegeben und mit binary_crossentropy als Loss-Funktion berechnet.

*Tabelle 11: Struktur des CNNs im dritten Versuch*

| Layer (type) | Output Shape | Param # |
|---|---|---|
| mobilenet_1.00_512 (Model) | (None, 16, 16, 1024) | 3228864 |
| global_average_pooling2d_1 ( | (None, 1024) | 0 |
| dropout_1 (Dropout) | (None, 1024) | 0 |
| dense_1 (Dense) | (None, 512) | 524800 |
| dropout_2 (Dropout) | (None, 512) | 0 |
| dense_2 (Dense) | (None, 13) | 6669 |

Total params: 3,760,333
Trainable params: 3,738,445
Non-trainable params: 21,888

Der Trainingsverlauf zeigt eine Senkung des *Trainings-Loss*, aber einen konstante *Accuracy* (Abb. 29). Auch die *Validierung-Loss* in der Validierung ist um 50% gesunken. Ein *Overfitting* ist aus den Daten noch nicht zu erkennen. Dennoch ist zu bemerken, dass die *Validierungs-Accuracy* ab der zweiten *Epoche* konstant bleibt. Ein *CNN* wird trainiert und anschließend evaluiert, um eine hohe *Accuracy* bei unbekannten Daten zu erreichen. Die Validation symbolisiert die unbekannten Daten in jeder *Epoche*. Wenn diese *Accuracy* nach der 2 *Epoche* nicht mehr steigt, kann die Annahme gemacht werden, dass das *CNN* keinen Lernfortschritt gemacht hat.

```
Epoch 1/7
100/100 [ 83s 830ms/step - loss: 0.4401 - binary_accuracy: 0.8537 - val_loss: 0.8628 - val_binary_accuracy: 0.8167
Epoch 2/7
100/100 [ 73s 734ms/step - loss: 0.3660 - binary_accuracy: 0.8690 - val_loss: 0.3940 - val_binary_accuracy: 0.8746
Epoch 3/7
100/100 [ 74s 737ms/step - loss: 0.3542 - binary_accuracy: 0.8713 - val_loss: 0.4023 - val_binary_accuracy: 0.8746
Epoch 4/7
100/100 [ 74s 739ms/step - loss: 0.3518 - binary_accuracy: 0.8703 - val_loss: 0.3668 - val_binary_accuracy: 0.8744
Epoch 5/7
100/100 [ 74s 742ms/step - loss: 0.3649 - binary_accuracy: 0.8678 - val_loss: 0.3609 - val_binary_accuracy: 0.8746
Epoch 6/7
100/100 [ 74s 743ms/step - loss: 0.3493 - binary_accuracy: 0.8716 - val_loss: 0.3471 - val_binary_accuracy: 0.8744
Epoch 7/7
100/100 [ 74s 744ms/step - loss: 0.3321 - binary_accuracy: 0.8750 - val_loss: 0.3435 - val_binary_accuracy: 0.8752
```

*Abbildung 29: Trainingsverlauf aus dem Versuch 3*

Die *ROC-Kurve* bestätigt die Vermutung aus dem Trainingsverlauf. Fast alle 13 Klassen haben einen AUC-Wert um die 0,5. Dies bedeutet, dass die Bilder nach einem Zufallsprinzip prognostiziert werden (Abb. 30). Nur die *Kardiomegalie* (AUC: 0,65) und die *Atelektase* (AUC: 0,6) fallen beim Ergebnis besonders auf. Das *CNN* war im Versuch 3 nicht in der Lage, alle Klassen mit einer guten *Accuracy* vorherzusagen. Somit wird im letzten Versuch auf eine Auswertung der *Konfusionsmatrix* verzichtet. Da das *CNN* nach einem Zufallsprinzip die Bilder prognostiziert hat. Der letzte Versuch kann aufgrund der sehr schlechten Werte, die Forschungsfrage nicht bestätigen.

Alle wichtigen Eigenschaften im Versuch 3 sind aus der Tabelle 12 zu entnehmen. Die auch gleichzeitig als eine kurze Zusammenfassung zu sehen ist.

*Tabelle 12:Eigenschaften des CNNs im Versuch 3*

| Eigenschaften | Werte |
|---|---|
| Fehldiagnose | --- |
| Trainings Accuracy nach letzter Epoche | 88% |

| | |
|---|---|
| Kontrollkriterien | --- |
| ROC-Kurve | Durchschnittlich 0,5 AUC |
| Konfusionsmatrix | --- |
| Epochen | 7 |
| Klassen | Alle Klassen außer Hernie |
| Verteilung | Trainingsdaten: 30.00<br>Validierungsdaten: 10.000<br>Testdaten: 10.000 |
| Loss-Funktion | Binary Crossentropy |
| Optimizer | Adam |
| Aktivierungsfunktion | Am Ausgangs-Layer Sigmoid<br>ansonsten ReLU |
| Overfitting Werkzeuge | Dropout und Average-Pooling |
| Trainierte Parameter | 22.000 |
| Data-Augmentation | Ja |
| Faltungsbasis | MobileNet |

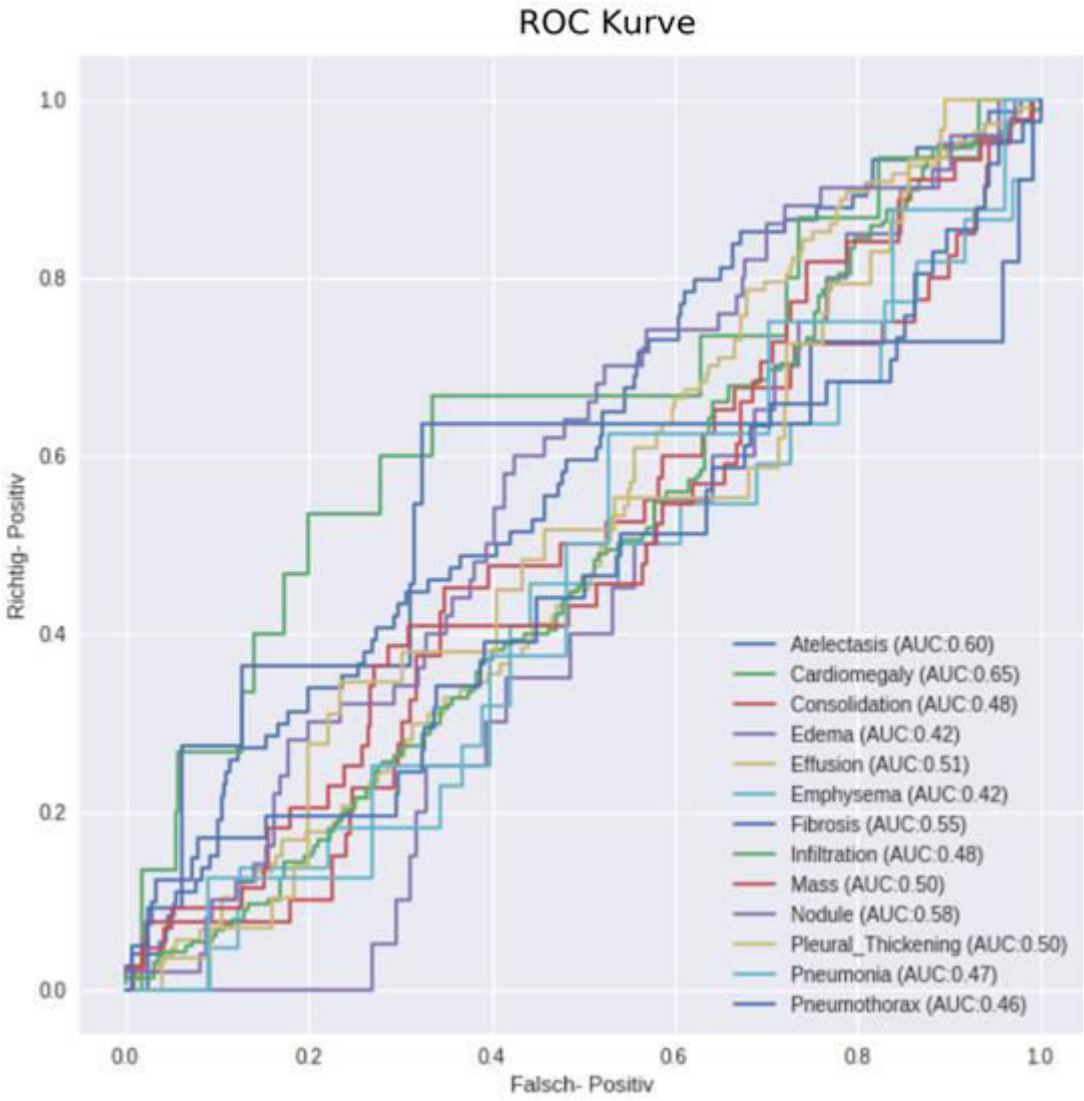



*Abbildung 30: Zeigt das Verhältnis von richtigen positiven zu falschen negativen in der ROC-Kurve*

# 5. Zusammenfassung und Fazit

Diese Bachelorthesis sollte die Frage beantworten, ob Fehldiagnosen, die durch menschliches Handeln bedingt sind, minimiert werden können, indem ein *CNN* aufgebaut wird, das Thorax-Röntgenbildern mit einer hohen Wahrscheinlichkeit richtig vorhersagt. Zu diesem Zweck wurde mit Hilfe einer quantitativen Studie die *Konfusionsmatrix* und der *ROC-Kurve* der einzelnen *CNN* interpretiert. Zusätzlich wurden die Kontrollkriterien *Spezifität*, *Sensitivität* und der *positiver prädiktiver Wert* mit in die Bewertung einbezogen. Weiterhin soll mithilfe der Kontrollkriterien

untersucht werden, inwiefern sich *CNN*s als Screening Programme etablieren.

Beim Versuch 2 wurde die Verteilung der Daten aus dem ersten Versuch übernommen. Somit musste das *CNN* nur die Vorhersage einer Erkrankung machen. Dagegen wurde die Struktur des *CNN* verändert, indem eine Faltungsbasis *VGG16* am Eingang hinzugefügt wurde. Zudem wurden die Trainingsdaten mittels Data-Augmentation verarbeitet. Das Resultat des zweiten *CNN* zeigt, dass die Fehldiagnosen bei 22 % liegen (Tab. 13). Trotz einer *Accuracy* von ca. 77 %, wobei die *Accuracy* im Versuch 1 weit höher bei 92% liegt. Die Fehldiagnosen im Versuch 2 hat sich im Verhältnis zu Versuch 1 deutlich verbessert. Dort liegt die Fehldiagnose bei 35 %. Dennoch liegt dieser Wert weit über den Fehldiagnosen der Ärzte von ca. 10 %. Somit können die aufgebauten *CNN* in dieser Arbeit, die erste Forschungsfrage nicht erfüllen.

Dagegen erfüllt das *CNN* die Kriterien für ein Screening-Programm. Der *positiver prädiktiver Wert* des *CNN* steigerte sich von 35 % vom Versuch 1 auf 60 %. Auch die *Spezifität* ist von 70 % auf 93 % gestiegen. Die *Sensitivität* ist jedoch von 48 % auf 30 % gesunken. Auch der AUC-Wert steigerte sich von 70 % auf 82 %. Wird nun der Vergleich zum Mammographie-Screening gezogen, weist das zweite *CNN* bessere Werte auf.

*Tabelle 13: Ergebnisse der Versuche. Der dritte Versuch wurde herausgenommen, da das CNN nur Zufallswerte ausgab.*

| | Versuch 1 (%) | Versuch 2 (%) | Mammographie (%) |
|---|---|---|---|
| Fehldiagnosen | 35 | 22 | / |
| Spezifität | 70 | 93 | 95 |
| Sensitivität | 48 | 30 | 78 |

| | | | |
|---|---|---|---|
| Positiver prädiktiver Wert | 35 | 60 | 15,4 |
| AUC | 70 | 82 | / |

Im Vergleich zum Mammographie-Screening ist der *positiver prädiktiver Wert* wesentlich höher. Die *Spezifität* liegt auf demselben Niveau. Die *Sensitivität* im *CNN* fiel jedoch geringer aus.

Im letzten Versuch wurde das gesamte Datenpaket zum Evaluieren genutzt. Dabei wurde die Struktur des *CNNs* bis auf eine neue Faltungsbasis nicht geändert. Die Ergebnisse, die das *CNN* ausgab, sind schlecht ausgefallen, sodass die Vorhersage einer Zufallsvorhersage entspricht. Das *CNN* war nicht in der Lage, die Bilder anhand der unterschiedlichen Klassen richtig auszusortieren. Dies zeigt sich auch daran, dass das AUC von fast allen Klassen im Bereich von 0,5 lag. Somit wurde die *Konfusionsmatrix* nicht mit den anderen *CNN* verglichen.

Diese quantitative Forschung hat gezeigt, dass die aufgebauten *CNN* nicht imstande sind, Thorax-Röntgenbilder mit einer geringen Fehldiagnose als 10 % zu Befunden. Dies wird anhand der *Konfusionsmatrix* sehr deutlich. Nur das zweite *CNN* kann als Screening-Programme eingesetzt werden. Da es einen höheren positiven prädiktiver Wert bei gleicher *Spezifität* aufweist.

# 6. Diskussion

Für diese Studie wurden eine *Konfusionsmatrix* und eine *ROC-Kurve* für die Auswertung der *CNN* genutzt. Trotz hoher *Accuracy* muss bei einer medizinischen Bildgebung die Ausgabe in richtig positiv bzw. -negativ und falsch positiv bzw. -negativ unterteilt werden. Zudem zeigt die Auswertung der *ROC-Kurve* das Verhältnis zwischen *richtig positiv* und *falsch positiv*. Dabei

wird die Auswertung der *ROC-Kurve* auch als Maß für einen guten bzw. schlechten *CNN* gesehen. Auf dieser Basis können die Ergebnisse aus den *CNNs* mit anderen *CNN* vergleichen werden.

Die Auswertung der *Konfusionsmatrix* und der *ROC-Kurve* zeigen, dass der *CNN* die Erwartung, die Fehldiagnosen zu minimieren, nicht erfüllt. Dieses Resultat steht nicht im Einklang der Erwartung, denn mehrere Studien beweisen, dass heutige *CNN* eine hohe *Accuracy* beim Klassifizieren von Bildern erreichen können. Dabei befinden sich auch die Fehldiagnosen unter 10 % [J. H. M. J. Vestjens, et al., 2012]. Stattdessen zeigt das zweite *CNN* Ergebnisse, die für einen Einsatz als Screening-Programm ausreichen. Dennoch sind diese Ergebnisse mit Vorsicht zu betrachten. Da ein zulässiges Screening-Programm viel mehr Kriterien erfüllen muss, außer den Kontrollkriterien, die in dieser Arbeit besprochen worden sind.

Eine Erklärung für das gute Abschneiden im Screening kann die Verzerrung der Werte sein, die in der *Konfusionsmatrix* dargestellt werden. Das Problem ist die ungleichmäßige Verteilung der Daten von negativen und positiven Befunden. Durch die hohe Anzahl von negativen Befunden wird die *Konfusionsmatrix* verzerrt. Das *CNN* macht insgesamt nur 48 positive Prognosen und 385 negative Prognosen. Aus den 385 negativen sind 284 richtig. Die hohe Anzahl der richtigen negativen verzerrt das Bild der *Konfusionsmatrix* und vermittelt den Eindruck, dass das *CNN* gute Prognosen macht. Wären die positiven und negativen Befunden gleichmäßig verteilt, wäre das Ergebnis besser ausgefallen. Dies wird in der Studie „The Impact of Imbalanced Training Data for Convolutional Neural Networks“ beschrieben. Die Ergebnisse eines *CNN*s werden bei einer ungleichmäßigen Verteilung der Klassen vom CNN negativ beeinflusst. Bei einer gleichmäßigen Verteilung können dagegen die besten Ergebnisse vom *CNN* erzielt werden [Hensman & Masko, 2015].

Eine weitere Erklärung wird von dem erfahrenen Radiologen Rayner gegeben. Er beschreibt, dass beim Zuordnen der jeweiligen Erkrankungen zu den Röntgenbildern fehlerhaft gearbeitet wurde. So wurden die Röntgenbilder mittels NLP den Erkrankungen zugeordnet, ohne diese von einem Radiologen visuell überprüfen zu lassen. Den nach einer kurzen Überprüfung fielen ihm bereits mehrere Röntgenbilder auf, deren Befunde fragwürdig waren. Außerdem beinhalteten einige Röntgenbilder wie im Beispiel des Pneumothorax zusätzlich Drainagen. Diese zusätzliche Information kann das Muster, dass das *CNN* in den letzten Ebenen lernt, verzerren. So besteht die Gefahr, dass weitere Bilder ohne Drainagen nicht mehr der Erkrankung zugeordnet werden können. Dadurch wird die Drainage ein elementares Muster für die Erkennung dieser Erkrankung [Luke Oakden-Rayner, 2017].

Eine weitere Erklärung ist der Aufbau der *CNN*. Die verwendeten *CNN* waren nicht optimal aufgebaut. Es gibt viele Studien, in den *CNN* wesentlich bessere Ergebnisse als Menschen erzielten. So wurde in einer Studie bewiesen, dass das *CNN* eine höhere *Accuracy* vorweist, als zwei Gruppen von Pathologen, die Krebszellen aus einem Präparat erkennen sollen. [J. H. M. J. Vestjens, et al., 2012]. Dabei erzielte das *CNN* in mehreren Disziplinen, in denen unter anderem Krebszellen zu erkennen waren, eine wesentlich höhere *Accuracy* als beide Gruppen. Zudem haben die Pathologen einige Krebszellen nicht gesehen bzw. nicht erkennen können, was dem *CNN* dagegen gelang.

Des Weiteren muss berücksichtigt werden, dass sich diese Forschung ausschließlich auf ein Datenpaket mit Röntgen-Thoraxbildern mit einer begrenzten Anzahl an Erkrankungen konzentriert. Es könnten auch Röntgenbildern mit Frakturen oder Bilder mit Hauterkrankungen getestet werden. Außerdem schränkt sich diese Studie nur auf drei *CNN* ein. Dabei können weitere bekannte *CNN* mit einer höheren *Accuracy* benutzt werden, die ein anderes Ergebnis erzielen würden.

Die Empfehlung für die weitere Forschung ist daher, die genannten Ansätze, die zur negativen Beeinflussung des *CNNs* geführt haben, zu minimieren und zu prüfen, ob der genutzte Datenpaket die Verzerrung aufweist. Dabei können neue *CNN* eingesetzt werden, die wiederum eine höhere *Accuracy* erzielen. Eine weitere Empfehlung besteht darin, ein eigenes Datenpaket an Röntgenbildern in Krankenhäusern zusammenzustellen. Dadurch können bekannte Fehler im Voraus gefiltert werden, was zu einem besseren Resultat beim *CNN* führen kann.

# 7. Ausblick

In diesem Kapitel werden weitere Optimierungs- und Ergänzungsmöglichkeiten vorgegeben, die anhand der kurzen Dauer der vorliegenden Arbeit von 10 Wochen nicht mehr durchgeführt werden konnten.

1. Datenmaterial selbst herstellen.

   Zunächst wurde angestrebt, für diese Arbeit ein Datenpaket an Röntgenbildern aus einem Krankenhaus mit Hilfe eines Radiologen herzustellen. Dies erschwerte sich aus bürokratischen Gründen und sollte in den Folgestudien nachgeholt werden. Dadurch können zusätzliche Informationen, die in Röntgenbildern vorhanden sind, wie zum Beispiel Drainagen oder Herzschrittmacher, aussortiert werden. Außerdem ist es auf diese Weise möglich, einen ausgeglichene Datenpaket herzustellen.

2. Ein vierter Versuch

   In dieser Studie war noch ein weiterer Versuch geplant, indem ein CNN aufgebaut werden sollte, das eine Ähnlichkeit mit dem zweiten Versuch hat nur mit dem Unterschied, dass einige Layer der Faltungsbasis eingefroren werden sollten. In weiteren Studien sollte die Faltungsbasis eines *CNN* nicht mittrainiert werden. Als Resultat

wird eine Verminderung des *Overfitting*s erwartet. Zudem sollte das Resultat am Klassifikator positiv beeinflusst werden.

3. Darstellung der falschen positiven Prognose

   Die Prognosen des CNN sollten bildlich dargestellt werden. Somit sollte aus den Bildern ersichtlich werden, ob sie vom CNN als positiv oder negativ prognostiziert wurden und anhand welcher Merkmale dieser ein Bild einer Klasse zuordnet.

4. *CNN*-Filter visualisieren

   Als Ergänzung zu Punkt 3 war es in dieser Studie auch geplant, die Röntgenbilder, die über die einzelnen Filter des CNNs übergeben werden, zu visualisieren. In weiteren Studien soll ersichtlich werden, welche Muster die einzelnen Filter aus den Röntgenbildern erlernen. Außerdem zeigt die Heatmap, welche Muster verantwortlich sind, damit der Klassifikator seine Entscheidung trifft.

# 8. Literaturverzeichnis

Abien Fred M. Agarap, 2018. *Deep Learning using Rectified Linear Units (ReLU).* [Online]
Available at: https://arxiv.org/pdf/1803.08375.pdf
[Zugriff am 12 16 2018].

Angermayer, J., 2002. *Angermayer.* [Online]
Available at: https://www.angermayer.de/padochp/menue5.html
[Zugriff am 31 08 2018].

Anon., 2017. *Kaggle.* [Online]
Available at: https://www.kaggle.com/nih-chest-xrays/data

Anon., 2018. *TinyMind.* [Online]
Available at: https://www.tinymind.com/learn/terms/relu

Ärzte Zeitung, 2014. Weniger Ärzte behandeln mehr Patienten. *Ärzte Zeitung*.

Bundesministerium für Gesundheit, 2018. *Bundesministerium für Gesundheit.* [Online]
Available at:
https://www.bundesgesundheitsministerium.de/krebsfrueherkennung.html
[Zugriff am 07 12 2018].

Chaitanya Asawa, kein Datum *github*. [Online]
Available at: http://cs231n.github.io/neural-networks-1/#actfun
[Zugriff am 01 12 2018].

Chang, R., 2017. *Kaggle.* [Online]
Available at: https://www.kaggle.com/juiyangchang/cnn-with-pytorch-0-995-accuracy

Chollet, F., 2018. Deep Learning with Python. In: New York: Manning, p. 123.

Christoph Wick , 2017. *Gesellschaft für Informatik,* Würzburg: Springer-Verlag.

Davi Frossard, 2016. *cs.toronto.* [Online]
Available at: https://www.cs.toronto.edu/~frossard/post/vgg16/
[Zugriff am 2018].

Deshpande, M., 2017. *Zenva.* [Online]
Available at: https://pythonmachinelearning.pro/introduction-to-convolutional-neural-networks-for-vision-tasks/

Deshpande, M., 2017. *Zenva.* [Online]
Available at: https://pythonmachinelearning.pro/introduction-to-convolutional-neural-networks-for-vision-tasks/

Djulbegovic M, et al., 2010. *Screening for prostate cancer: systematic review and meta-analysis of randomised controlled trials.*. s.l.:PubMed.

Dr. Daniela Malek & Peter Rabe, 2008. *Evaluationsbericht 2005–2007. Ergebnisse des Mammographie-Screening-Programms in Deutschland 2009.,* s.l.: Kooperationsgemeinschaft Mammographie.

Ehteshami Bejnordi B, et al., 2017. Deep learning in chest radiography: Detection of findings and presence of change. 12 12.

Esteva A, et al., 2017. Dermatologist-level classification of skin cancer with deep neural networks.. 2 02.

Eur J Cancer, 2000. *Recommendations on cancer screening in the European union. Advisory Committee on Cancer Prevention.* s.l.:s.n.

Francois Chollet, 2017. Deep Learning with Python. In: *Chaining derivatives:the Backpropagation algorithm*. New York: Manning, p. 52.

Francois Chollet, 2018. Deep Learning with Python. In: *The engine of neural networks: gradient-based optimization*. New York: s.n., pp. 46-49.

Francois Chollet, 2018. Deep Learning with Python. In: *Overfitting and underfitting*. New York: Manning, pp. 104-105.

Francois Chollet, 2018. Deep Learning with Python. In: *Adding dropout*. New York: Manning, pp. 109-110.

Francois Chollet, 2018. Deep Learning with Python. In: *Introduction to convnets*. New York: Manning, pp. 120-124.

Francois Chollet, 2018. Deep Learning with Python. In: *The max-pooling operation*. New York: Manning, pp. 127-128.

Giersiepen K, Hense HW, Klug SJ, Antes G, Zeeb H, 2007. *Entwicklung, Durchführung und Evaluation von Programmen zur Krebsfrüherkennung. Ein Positionspapier.* s.l.:s.n.

Hans-Werner Hense, 2014. *Aerzteblatt.de.* [Online]
Available at: https://www.aerzteblatt.de/nachrichten/59496/Die-Sensitivitaet-des-Mammographie-Screeningprogramms-ist-gut
[Zugriff am 09 12 2018].

Heckl, M., 2017. *scilogs.spektrum.* [Online]
Available at: https://scilogs.spektrum.de/marlenes-medizinkiste/gestatten-mein-arzt-der-roboter/

Hensman, P. & Masko, D., 2015. *The Impact of Imbalanced Training Data for Convolutional Neural Networks.* Schweden: DEGREE PROJECT, IN COMPUTER SCIENCE , FIRST LEVEL.

Hertwig, F., 2018. *Mailborn Wolf.* [Online]
Available at: https://www.maibornwolff.de/blog/geschichte-von-neural-networks-und-deep-learning
[Zugriff am 24 10 2018].

J. H. M. J. Vestjens, et al., 2012. *Relevant impact of central pathology review on nodal classification in individual breast cancer patients,* s.l.: PubMed.

Jörg Blech, 2011. Wenn Ärzte irren. *Sprachlos in der Sprechstunde*, Februrar.

Karsten J Jørgensen, Per-Henrik Zahl & Peter C Gøtzsche, 2009. *Overdiagnosis in organised mammography screening in Denmark. A comparative study.* [Online]
Available at: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC2807851/
[Zugriff am 07 12 2018].

Kenneth C. Laudon, Jane P. Laudon & Detlef Schoder, 2010. In: *Wirtschaftsinformatik Eine Einführung*. s.l.:Pearson Studim.

Kevin Mader, 2017. *Kaggle*. [Online]
Available at: https://www.kaggle.com/kmader/cardiomegaly-pretrained-vgg16
[Zugriff am 2018].

Kevin Mader, 2017. *Kaggle*. [Online]
Available at: https://www.kaggle.com/kmader/train-simple-xray-cnn
[Zugriff am 2018].

Kirch, W., 2005. In: *Fehldiagnosen und Patientensicherheit*. Heidelberg: Springer Verlag, p. 3.

Kriesel, D., 2005. *DKreisel*. [Online]
Available at: http://www.dkriesel.com/science/neural_networks

Lamash Y, Warfield SK & Kurugol S, 2018. Semi-Automated Extraction of Crohns Disease MR Imaging Markers using a 3D Residual CNN with Distance Prior.. *Deep Learn Med Image Anal Multimodal Learn Clin Decis Support (2018)*, 09, pp. 218-226.

Luke Oakden-Rayner, 2017. *Exploring the ChestXray14 dataset: problems,* s.l.: s.n.

Manhart, K., 2018. *Computerwoche*. [Online]
Available at: https://www.computerwoche.de/a/eine-kleine-geschichte-der-kuenstlichen-intelligenz,3330537,2
[Zugriff am 24 10 2018].

Meisel, Prof. Dr.-Ing. Andreas, 2018. *Klassifikation mit Neuronalen Netzen*. Hamburg: s.n.

Meisel, Prof. Dr.-Ing. Andreas, 2018. *Vermeidung von Overfitting*. Hamburg: s.n.

Meisel, P. D.-I. A., 2018. *Robot Vision*. Hamburg: s.n.

opening.Download, kein Datum *opening.Download*. [Online]
Available at: http://opening.download/open-downloading.html
[Zugriff am 05 12 2018].

Plaum, A., 2018. *oreillyblog*. [Online]
Available at: https://blog.oreilly.de/category/merkwurdige-begriffe/
[Zugriff am 04 09 2018].

Shrikant, 2018. *Kaggle*. [Online]
Available at: https://www.kaggle.com/shrikantds/cnn-in-nih-dataset
[Zugriff am 20 12 2018].

Spix, Claudia; Blettner, Maria, 2012. *aerzteblatt.de*. [Online]
Available at: https://www.aerzteblatt.de/archiv/126279/Screening
[Zugriff am 07 12 2018].

Tariq Rashad, 2017. Neuronale Netze selbst programmieren. In: *Ein verständlicher Einstieg mit Python*. Paderborn: O'REILLY, pp. 81-83.

Tariq Rashid, 2017. Neuronale Netze selbst programmieren. In: *Ein verständlicher EInstieg mit Python*. Paderborn: O'REILLY, pp. 39-42.

Tariq Rashid, 2017. Neuronale Netze selbst programmieren. In: *Ein verständlicher EInstieg mit Python*. Paderporn: O'REILLY, p. 42.

Thomas G. Tape, MD, kein Datum *University of Nebraska Medical Center*. [Online]
Available at: http://gim.unmc.edu/dxtests/ROC3.htm
[Zugriff am 05 12 2018].

TinyMind, 2018. *TinyMind*. [Online]
Available at: https://www.tinymind.com/learn/terms/relu
[Zugriff am 12 01 2018].

Vincent C. A., Driscoll P. A., Audley R. J. & Grant D., 1988. Accuracy of detection of radiographic abnormalities by junior doctors. p. 101.

Welch HG & Frankel BA, 2011. *Likelihood that a woman with screen-detected breast cancer has had her "life saved" by that screening*.. s.l.:s.n.

Wender, G. D. R. /. K. F., 2018. *neuronales Netz*. [Online]
Available at: http://www.neuronalesnetz.de/hebb.html
[Zugriff am 08 29 2018].

Wender, G. D. R. /. K. F., 2018. *Neuronales Netz*. [Online]
Available at: http://www.neuronalesnetz.de/aktivitaet.html
[Zugriff am 30 08 2018].

Wilfried Bautsch, 2010. aerzteblatt.de. *Anforderung und Bewertung der Ergebnisse von Laboruntersuchung,* Issue 2009.0403, pp. 1-2.

Zreik M, et al., 2018. A Recurrent CNN for Automatic Detection and Classification of Coronary Artery Plaque and Stenosis in Coronary CT Angiography.. *IEEE Trans Med Imaging*..