Julius-Maximilians-Universität Würzburg

Institut für Informatik
Lehrstuhl für Künstliche Intelligenz
und Angewandte Informatik

Bachelorarbeit

im Studiengang Informatik

zur Erlangung des akademischen Grades
Bachelor of Science

Vergleich von konditionalem und flachem Training von CNNs zur Klassifikation von Röntgen-Thorax-Bildern mit verschiedenen Labeling-Methoden

---

Autor: Lisa Rost <lisa.rost@stud-mail.uni-wuerzburg.de>
MatNr. 2285766

Abgabe: 25.05.2021

1. Betreuer: Prof. Dr. Frank Puppe
2. Betreuer: M. Sc. Amar Hekalo

Abstract

Röntgen-Thorax-Aufnahmen werden in der Medizin unter anderem für die Diagnose von verschiedenen Herz- und Lungenerkrankungen genutzt. Im Bereich der künstlichen Intelligenz spielen Convolutional Neural Networks für die Verarbeitung von Bilddaten eine große Rolle. Bei einer Verarbeitung von Röntgen-Thorax-Aufnahmen werden diese für das Erkennen von Pathologien genutzt. In dieser Arbeit wird hierarchische Multi-Label Klassifizierung umgesetzt, um Röntgen-Aufnahmen des CheXpert Datensatzes mit 14 Klassen einzuordnen. In dieser Arbeit wird das Konzept der Label Smoothing Regularization und des konditionalen Trainings untersucht. Ersteres wird genutzt, um unsichere Labels für das Training der Modelle verwenden zu können, indem diese durch Werte nahe bei 0 und 1 ersetzt werden. Dabei wird ein Vergleich zu strikten Methoden gezogen und eine neue Label Smoothing Methode U-Random getestet, welche unsichere Labels durch zufällige Werte zwischen 0 und 1 ersetzt. Außerdem wird konditionales Training umgesetzt, um die hierarchische Struktur der Pathologien zu nutzen. Damit wird eine Klasse unter der Bedingung ihrer Eltern-Klasse betrachtet und die Beziehungen zwischen Pathologien in das Training mit einbezogen. Basierend auf den vortrainierten Keras Netzen DenseNet121 und EfficientNetB4, werden Modelle auf einem eigens erstellten Datensplit des öffentlichen CheXpert Datensatzes trainiert und evaluiert. Zusätzlich wird für die Evaluation das Validierungsset des CheXpert Datensatzes herangezogen, um Vergleiche mit bestehenden Arbeiten zu ermöglichen. Die neue Label Smoothing Regularization Methode U-Random erzielt den größten durchschnittlichen AUC von 0,8695 auf dem Validierungsset, bei einem flachen Training von EfficientNetB4. Diese erhält unter anderem bessere Ergebnisse als die Label Smoothing Regularization Methoden U-Ones-LSR und U-Zeros-LSR, beispielsweise bei einem flachem Training über 5 Epochen mit EfficientNetB4, sowohl bei der Evaluation auf dem Testset, als auch auf dem Validierungsset. Das konditionale Training erlangt im Durchschnitt niedrigere Ergebnisse als vergleichbares flaches Training der beiden Netze. Flaches Training erreicht einen Durchschnitt über alle, im ersten Teil der Arbeit evaluierten Methoden auf dem Testset, von 0,8194, während konditionales Training im Vergleich einen AUC-Wert von 0,7954 erlangt. Die Verwendung von Label Smoothing Regularization bewirkt geringe Verbesserungen der Ergebnisse bei dem Großteil der evaluierten Modelle bei flachem Training. Ebenso zeigt sich eine geringe Steigerung der durchschnittlichen AUC-Werte für das EfficientNetB4, im Vergleich zu flachem Training mit DenseNet121. Schließlich wurden die erzielten Ergebnisse mit der bestehenden Arbeit von Pham et al. [27] verglichen. Die darin erlangten Höchstwerte werden in dieser Arbeit nicht erreicht, darin bestand jedoch nicht das Ziel der Arbeit. Das bestand darin, neue Erkenntnisse im Bereich des konditionalen Trainings und der Label Smoothing Regularization zu erlangen.

Inhaltsverzeichnis

Abbildungsverzeichnis

Tabellenverzeichnis

1 Einleitung

Laut dem Bundesamt für Strahlenschutz wurden allein im Jahr 2016 knapp 80 Millionen Röntgenaufnahmen (ohne Zahnmedizin) durchgeführt [4]. Röntgen-Thorax-Aufnahmen werden unter anderem für die Pathologie der Lunge genutzt [40]. Beispielsweise wird ein Pneumothorax, der eine lebensbedrohliche Situation darstellen kann, häufig durch Röntgen-Aufnahmen diagnostiziert [35]. Alleine Pneumonie ist laut der WHO für 15% der Tode von Kindern unter 5 Jahren verantwortlich [41]. Diese wird unter anderem auf Thorax-Röntgen-Bildern diagnostiziert. Das Erstellen von Röntgen-Bildern ist verhältnismäßig günstig und leichter zugänglich im Vergleich zu anderen Techniken, wie das CT [10]. Außerdem handelt es sich um eine global standardisierte Aufnahmetechnik [39]. Dennoch ist die Interpretation von Röntgenbildern keine einfache Aufgabenstellung und es passieren Fehler bei der Diagnosestellung [10]: Bei Röntgen-Aufnahmen handelt es sich um eine zweidimensionale Abbildung des dreidimensionalen Körpers, weshalb Gewebe aufeinander dargestellt wird. Daher ist es notwendig die Technik von Röntgen-Aufnahmen und deren Einschränkungen zu verstehen. Die Aufgabe besteht darin, Abnormalitäten zu identifizieren, weswegen die Anatomie und deren natürliche Variationen verstanden werden müssen. Das ist die Aufgabe von Radiologen, wobei ein CAD (Computer-aided diagnosis system) eine Unterstützung für den Radiologen sein kann [27, 10]: Es könnte genutzt werden, um die Diagnose des behandelnden Arztes zu überprüfen bzw. zu bestätigen und dadurch fehlerhafte Diagnosen zu reduzieren. Beispielsweise könnte eine vom Radiologen übersehene Pathologie entdeckt werden und womögliche Folgen für den Patienten vermeiden. Bei Notfallsituationen kann eine schnelle Reaktion entscheidend sein, weshalb ein CAD mit geringer benötigter Zeit zur Diagnosestellung von Vorteil sein kann. Auch bei weniger Zeitdruck kann durch den Einsatz von CADs die Wartezeit von Patienten reduziert werden und der Arbeitsablauf optimiert werden, indem es den behandelnden Arzt unterstützt.

Neben den Chancen, die die Benutzung eines CADs mit sich bringt, gibt es auch Einschränkungen und offene Fragen [39, 10]: Einerseits treffen Ärzte ihre Entscheidungen über Diagnosen und Behandlungen anhand mehrerer Faktoren, wie den Symptomen des Patienten, dessen Erscheinungsbild und den Erfahrungen des Arztes. Ein CAD, das eine Diagnose nur anhand der Röntgen-Aufnahme stellt, hat hierbei eine andere Entscheidungsgrundlage als der behandelnde Radiologe. Die Aufgabe von CADs besteht beispielsweise darin die Röntgen-Aufnahmen zu klassifizieren, also Pathologien auf diesen zu erkennen. Beim Thema dieser Arbeit handelt es sich um die Klassifizierung von Röntgen-Thorax-Aufnahmen, was durch Convolutional Neural Networks (CNNs) realisiert wird. In Datensätzen für das Training von CNNs können Pathologien nicht nur vorhanden (positiv) oder nicht vorhanden (negativ), sondern außerdem unsicher sein. Der CheXpert Datensatz, welcher in dieser Arbeit verwendet wird, besteht beispielsweise zu 37,2% aus Bildern mit unsicheren Labels. Daher ist es von Bedeutung,

die Frage zu klären, wie bei dem Training der Netze mit Unsicherheiten umgegangen werden soll. Aus diesem Grund besteht der erste Teil dieser Arbeit aus dem Vergleich verschiedener Ansätze im Bereich der Label Smoothing Regularization. Ein weiterer Teil der Arbeit ist die Evaluation von konditionalem Training, welches die Pathologien im Datensatz nicht unabhängig voneinander behandelt und in der Arbeit von Pham et al. [27] mit dem CheXpert Datensatz vorgestellt wurde. Dabei wird mit Hierarchien von Pathologien gearbeitet, weshalb man von einer hierarchischen Multi-Label Klassifizierung spricht. Das Ziel der Arbeit ist es, sowohl die Label Smoothing Methoden, als auch das konditionale Training in einer unabhängigen Studie zu testen und zu evaluieren. Anschließend sollen die erzielten Ergebnisse mit den Erkenntnissen von Pham et al. [27] verglichen werden.

Daher ist die Arbeit wie folgt aufgebaut: Im folgenden Kapitel werden die Grundlagen von Convolutional Neural Networks erklärt. Dabei wird auf die Architektur und die Funktionsweise genauer eingegangen. Im Kapitel 3 werden verwandte Arbeiten zu diesem Thema präsentiert. Daraufhin wird das für diese Arbeit durchgeführte Training genauer beschrieben. Darunter fallen die verwendeten Daten und Modelle, Label Smoothing Regularization, konditionales Training und die Evaluation der Netze. Nachfolgend werden die erzielten Ergebnisse diskutiert und mit dem Paper von Pham et al. [27] verglichen. Schließlich gibt das Kapitel 7 einen Ausblick und die Erkenntnisse der Arbeit werden zusammengefasst.

2 Grundlagen der verwendeten Methoden

Im Bereich der medizinischen Bildverarbeitung gibt es verschiedene Aufgabenstellungen, wie beispielsweise die Objekterkennung, Segmentierung oder Klassifizierung [42]. In dieser Arbeit geht es um die Multi-Label Klassifizierung von Röntgen-Bildern, das heißt den Bildern sollen verschiedene Klassen zugeteilt werden, in diesem Fall Pathologien und andere Beobachtungen. “Multi-Label” bedeutet, dass mehrere Klassen einer Röntgen-Aufnahme zugeteilt sein können, was der Tatsache entspricht, dass Menschen zu einem Zeitpunkt mehrere Pathologien haben können. Für die Verarbeitung von Bildern sind Convolutional Neural Networks (CNNs) essentiell, weshalb in Folgenden die wichtigsten Grundlagen von CNNs erklärt werden.

Bei einem CNN handelt es sich um eine spezielle Art eines Neuronalen Netzes. Diese sind auf die Verarbeitung von gitterartigen Daten spezialisiert [13]. Das sind beispielsweise Bilder, die zweidimensionalen Rastern aus Pixeln entsprechen. Inspiriert wurde die Idee des CNNs durch den visuellen Cortex, welcher den Prozess des Sehens durch eine Abfolge von Schichten verarbeitet [30]. Für herkömmliche Neuronale Netze wäre die Verarbeitung von Bilddaten, insbesondere größeren Bilder, zu komplex [24]. Bei einem Überwachten Lernen eines CNN werden zum Trainieren Bilder und deren erwartetes Ergebnis, also Klassen die darauf zu sehen sind, genutzt. Allgemein läuft der Trainingsprozess dabei folgendermaßen ab [42]: Zunächst wird durch die sogenannte “Forward propagation” die Ausgabe des CNNs für ein Bild erzeugt. Anschließend wird daraus und aus dem eigentlich richtigen Wert der Fehler durch eine Loss-Funktion berechnet. Das Netz lernt dann durch die sogenannte “Back propagation”, indem veränderbare Parameter, wie Gewichte im Netz, demnach geändert werden. D.h. die Differenz zwischen der Ausgabe des CNNs und dem eigentlich richtigen Wert wird minimiert. Dieser Prozess ist in Abbildung 1 dargestellt. Wird dies für ein Trainingsset mit ausreichend vielen Bildern durchgeführt, lernt das CNN, die Bilder zu klassifizieren. Anschließend kann ein Validationset genutzt werden, um bestimmte Hyperparameter zu verbessern und das beste Modell zu bestimmen. Mit einem Testset kann die Performance des fertigen Netzes evaluiert werden, indem es Bilder klassifiziert, die nicht für das Training genutzt wurden und somit vom Netz noch nicht “gesehen” wurden. In den folgenden Abschnitten werden die Architektur, das Training und die Evaluation von CNNs betrachtet.

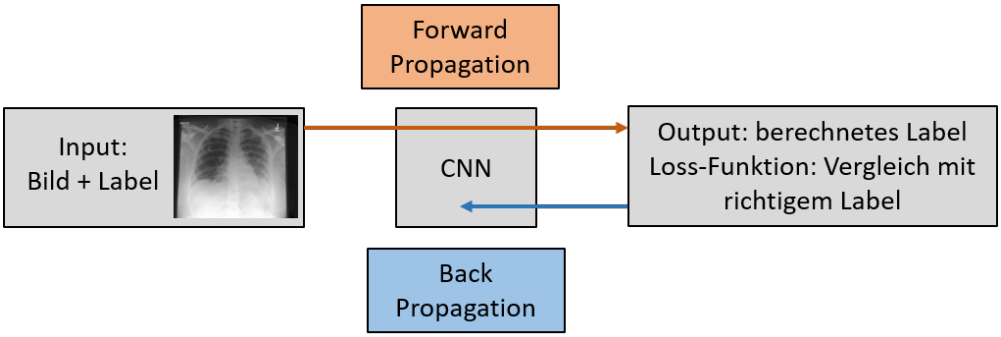


Abbildung 1: Der Trainingsprozess eines CNNs: Durch Forward Propagation wird aus dem Eingabebild und dessen Label der Output berechnet. Dieses berechnete Label wird durch eine Loss-Funktion mit dem richtigen Label verglichen, weshalb das CNN durch Back Propagation und einem Optimierungsalgorithmus daraus lernen kann.

2.1 Architektur eines CNNs

Die Architektur eines CNN besteht typischerweise aus drei Arten von Layern [24]:

- Convolutional Layer
- Pooling Layer
- Fully-Connected Layer

Das Convolutional Layer ist die fundamentale Komponente des CNNs und ist für die Feature Extraction verantwortlich [42]. Dabei wird das Eingabebild nach diesen "Features" (Merkmale) gefiltert, wobei es sich beispielsweise zunächst um Linien, Kanten, Punkte und Ecken handeln kann [32]. Diese werden in tieferen Teilen des Netzes komplexer. Das wird mit sogenannten "Kernels" bzw. "Filtern" realisiert. D.h. während des Trainingsprozesses werden die Werte dieser Kernels gelernt [18]. Dabei handelt es sich im Grunde um eine lineare Operation, die an jeder Stelle des Input Tensors berechnet wird [24]. Dadurch können Objekte durch das CNN auch an beliebiger Stelle im Bild erkannt werden. Diese Berechnung ist in Abbildung 2 dargestellt. Für einen Kernel und eine Position im Input Tensor werden die Werte elementweise multipliziert und anschließend die Summe der Produkte berechnet [42]. Das Ergebnis dieser Berechnung an jeder Stelle ergibt dann eine "Feature Map" [34]. In der Abbildung 2 oben, ergibt das folgende Berechnung: $(1\cdot1)+(2\cdot0)+(1\cdot1)+(2\cdot0)+(0\cdot1)+(0\cdot0)+(1\cdot1)+(0\cdot0)+(2\cdot1) = 5$

Bestimmte Hyperparameter beeinflussen dabei das Ergebnis [32]:

- Stride: Gibt die Distanz zwischen zwei Kernel Positionen im Input Tensor an [42]. In der Abbildung 2 ist der Stride beispielsweise 1, da der Kernel nur um einen Wert "weiter gerückt" wurde. Dabei überlappen sich diese Positionen des Kernels im Input Tensor.

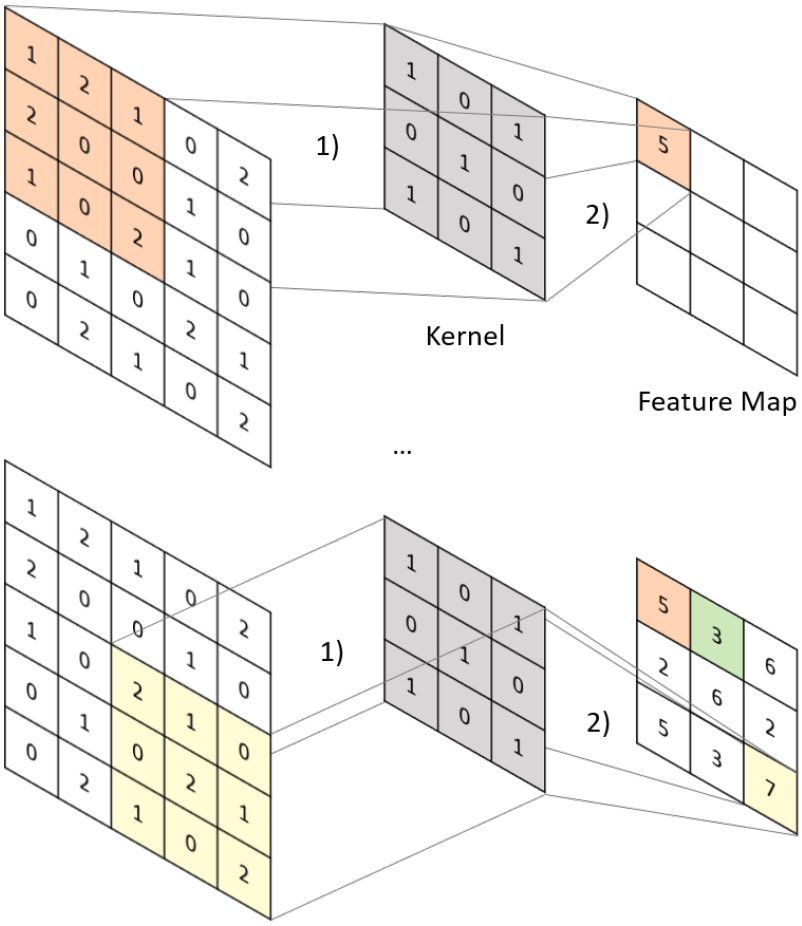


Abbildung 2: Die Berechnung der Convolution am Beispiel eines 5x5 Input Tensor und einem 3x3 Kernel: 1) elementweises Produkt des Kernels mit dem Input, 2) Summe der berechneten Produkte ergibt den Wert in der Feature Map. Das erste Bild zeigt dabei den Anfang der Berechnung, während im unteren Bild der letzte Schritt der Convolution dargestellt ist. Übernommen aus [42]

- Padding: In der Abbildung 2 ist die kleine Größe der berechneten Feature Map zu erkennen, verglichen mit dem Input Tensor vor der Convolution. Zero-Padding, eine Variante des Paddings, wirkt dem entgegen, indem Nullen an allen Seiten des Input Tensors angefügt werden [32]. Die Abbildung 3 zeigt das an einem Beispiel mit einem Kernel der Größe 3x3 und einem Stride von 1. Die berechnete Feature Map hat durch das Zero-Padding die gleiche Größe wie der ursprüngliche Input Tensor.

- Größe und Anzahl der Kernels, die genutzt werden, um Feature Maps zu berechnen

Ein weiterer Teil der Architektur ist das Pooling Layer, welches zusätzlich zum Convolutional Layer zur Feature Extraction beiträgt [18]. Die Hauptaufgabe dieses Layers ist jedoch die Dimensionalität von Feature Maps zu reduzieren und damit auch die letztendliche Anzahl der Parameter zu verringern [42]. Das Pooling Layer selbst enthält

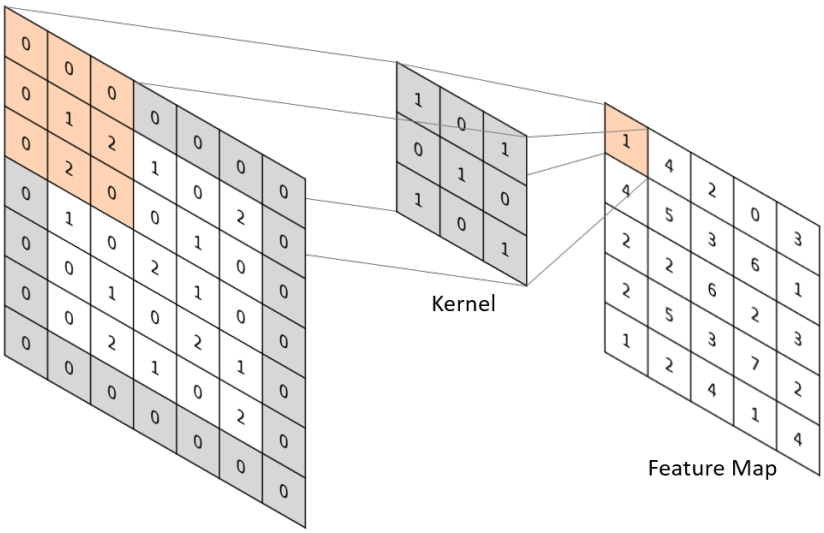


Abbildung 3: Durch das Zero-Padding wird bei der Convolution die ursprüngliche Größe des Input Tensors beibehalten. Übernommen aus [42]

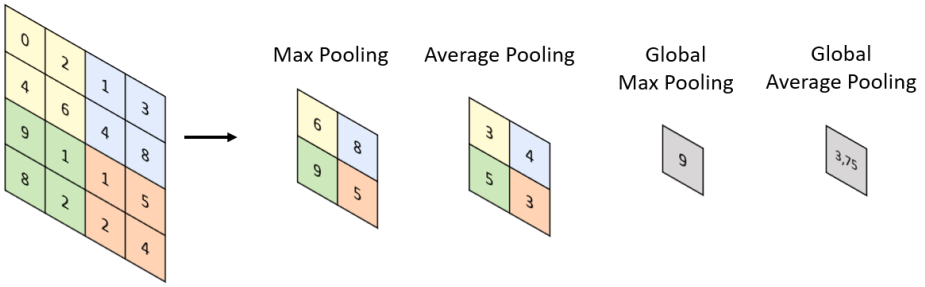


Abbildung 4: Beispiele von verschiedenen Pooling Methoden: Max Pooling und Average Pooling mit einem 2x2 Filter und einem Stride von 2, Global Max Pooling und Global Average Pooling mit einem 4x4 Filter

jedoch keine eigenen lernbaren Gewichte [32]. Jedoch gibt es, wie bei der Convolution, Hyperparameter wie Stride und die Größe des verwendeten Filters. Außerdem gibt es verschiedene Arten von Pooling Layern, wie beispielsweise “Max Pooling”, “Average Pooling”, “Global Max Pooling” und “Global Average Pooling”. Mit Hilfe eines Filters wird jeweils das Maximum bzw. der Durchschnitt der Werte darin berechnet, während alle restlichen Werte nicht verwendet werden [32]. Dadurch werden die Dimensionen des Input Tensors reduziert, da jeweils ein Bereich des Inputs als ein einzelner Wert abgebildet wird [18]. Bei Global Max Pooling und Global Average Pooling haben der Input Tensor und der Filter die gleiche Größe. Daher wird das Maximum bzw. der Durchschnitt über alle Werte des Input Tensors berechnet. Abbildung 4 zeigt jeweils ein Beispiel der Pooling-Berechnung. Global Average Pooling wird typischerweise als ein Layer vor dem Fully-Connected Layer verwendet [42]. Außerdem bietet es den Vorteil, dass Bilder mit variabler Größe durch das CNN verarbeitet werden können.

Das Fully-Connected Layer liefert den Output des Netzes, welches auch Dense Layer genannt wird [42]. Wie der Name sagt, ist in dieser Schicht jeder Input mit jedem

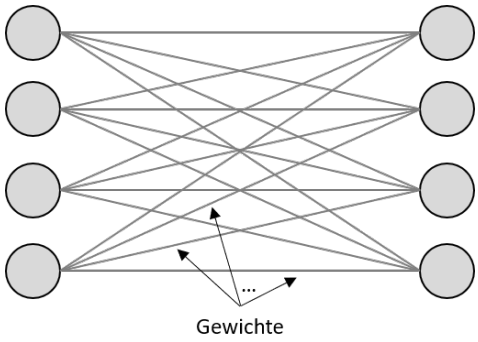


Abbildung 5: Aufbau einer Fully-Connected Schicht, bei der jeder Input mit jedem Output durch ein Gewicht verbunden ist.

Output durch ein Gewicht verbunden. Ein Beispiel hierfür ist in der Abbildung 5 dargestellt. Typischerweise entspricht die Anzahl der Klassen der Anzahl der Output-Knoten des Layers. Dadurch liefert das CNN für jedes Label eine Wahrscheinlichkeit dafür, dass dieses auf dem Bild vorhanden ist.

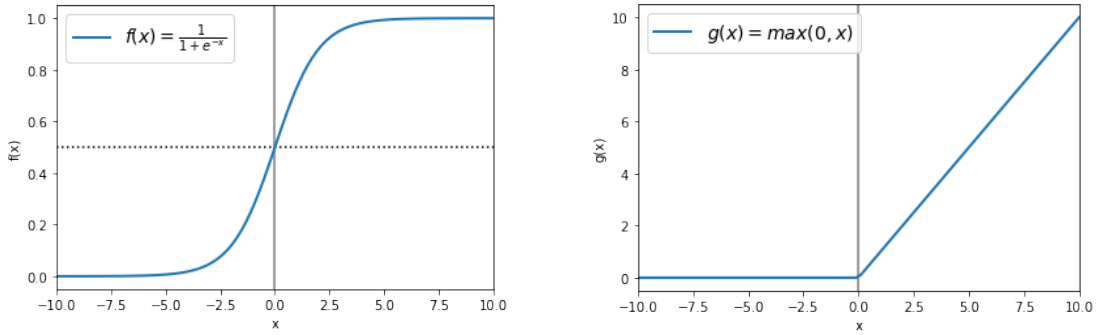
Nach der Convolution, einer linearen Operation, wird eine Aktivierungsfunktion berechnet, die nicht linear ist [42]. Das ist notwendig, um nicht-lineare und komplexere Aufgaben lösen zu können [33]. Beispiele für Aktivierungsfunktionen sind die Sigmoid-Funktion (vgl. Gl. 1) [20], "Rectified Linear Unit" (ReLU) (vgl. Gl. 1) [42] und Softmax (vgl. Gl. 1) [32]. Abbildung 6a zeigt die Sigmoid-Funktion, die bei einer Ausgabe von Wahrscheinlichkeiten geeignet ist, da sie Werte zwischen 0 und 1 liefert [34]. Sie ist außerdem für Multi-Label Klassifikationen nützlich, da im Vergleich zu der Softmax-Funktion das Ergebnis nicht von den anderen Klassen abhängig ist. Die Softmax-Funktion stellt sicher, dass die Summe über alle Klassen 1 ergibt. Daher wird sie unter anderem für die Multi-Class Klassifizierung genutzt, in der jeweils nur ein Label positiv sein kann. Eine weitere häufig genutzte Aktivierungsfunktion ist die ReLU, dargestellt in Abbildung 6b, welche das Maximum von 0 und x ausgibt [33].

$$\text{sigmoid}(x) = \frac{1}{1 + \exp(-x)} \tag{1}$$

$$\text{relu}(x) = \max(0, x) \tag{2}$$

$$\text{softmax}(x_j) = \frac{\exp(x_j)}{\sum_{k=1}^{K} \exp(x_k)} \text{ für } j = 1, ..., K \tag{3}$$

Die Abbildung 7, zeigt die typische Architektur eines CNNs. Dazu gehört eine Abfolge von Convolution und Pooling Layern im ersten Teil des Netzes. Darin wird nach der Convolution üblicherweise die ReLU Funktion als Aktivierungsfunktion verwendet [20]. Der zweite Teil des Netzes besteht aus einem oder mehreren Fully-Connected



(a) Graph der Sigmoid Funktion (b) Graph der ReLU Funktion

Abbildung 6: Darstellung von Aktivierungsfunktionen: Sigmoid und ReLu Funktion

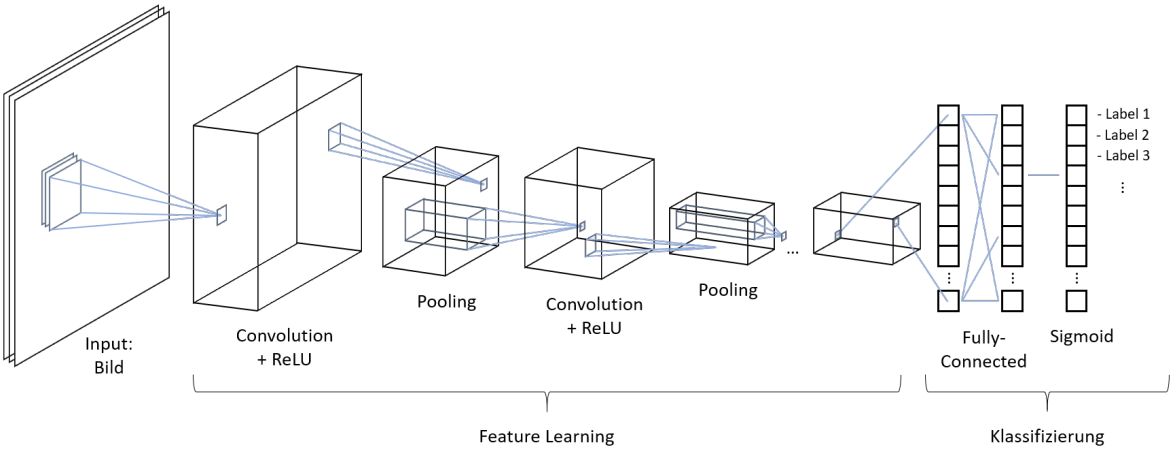


Abbildung 7: Typische Architektur eines CNNs mit Convolution-, Pooling- und Fully-Connected-Layern für die Multi-Label Klassifizierung. Übernommen aus [32]

Layern. Für die Multi-Label Klassifizierung wird hier die Sigmoid Funktion genutzt, welche schließlich die Wahrscheinlichkeit für jedes Label zurückgibt. Dabei sind die Wahrscheinlichkeiten der einzelnen Labels unabhängig voneinander, weshalb hierbei die Softmax Funktion nicht genutzt werden kann.

2.2 Optimierung

Für die Multi-Label Klassifizierung ist als Loss-Funktion die "Binary Cross-Entropy" geeignet (vgl. Gl. 4) [27]. Dabei sind θ die Parameter des Netzes und $D = \{(x^{(i)}, y^{(i)}); i = 1, ..., N\}$ das Trainingsset aus N Bildern. Darin wird jedes Bild $x^{(i)}$ mit dessen Label $y^{(i)}$ assoziiert. Zusammen mit dem vom Netz vorhergesagtem Label $y_k^{(i)}$ wird so der Loss berechnet. Das Ziel während des Trainings ist es, Parameter zu finden, die den Loss J minimieren. "Stochastic Gradient Descent" ist ein Optimierungsargorithmus, der für das Finden eines Minimums genutzt werden kann. Dessen Gradient ∇ wird auf der Basis eines Bildes $z = (x^{(i)}, y^{(i)})$ berechnet [3]. Die Gleichung 5 zeigt dabei die Updateregel, nach der die Änderung der Gewichte θ durchgeführt wird [3][32]. In

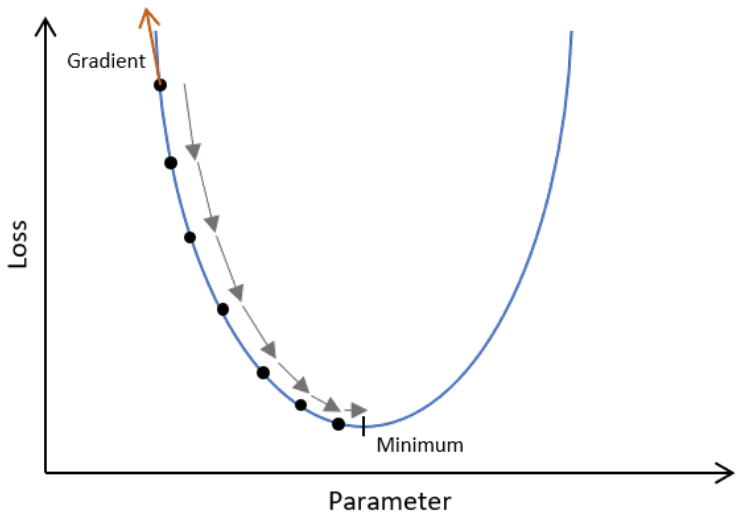


Abbildung 8: Durch Gradient Descent erfolgt eine schrittweise Annäherung an das Minimum der Loss Funktion

der Abbildung 8 ist dessen Ablauf an einer einfachen Funktion dargestellt. Dieser berechnet an einem Punkt der Loss Funktion den Gradienten ∇ mit Hilfe der ersten Ableitung [26]. Mit einer bestimmten Schrittweite (Lernrate η), die mit dem Gradienten multipliziert wird, "läuft" dieser in entgegengesetzter Richtung des Gradienten der Kurve entlang und nähert sich so schrittweise dem Minimum an [26].

$$J(\Theta) = \sum_{i=1}^{N} \sum_{k=1}^{K} y_k^{(i)} \cdot log(\hat{y}_k^{(i)}) + (1 - y_k^{(i)}) \cdot log(1 - \hat{y}_k^{(i)}) \tag{4}$$

$$\theta_{t+1} = \theta_t - h \tag{5}$$

$$h = \eta \cdot \nabla_\theta J(z_t, \theta_t)$$

Eine der Erweiterungen von "Stochastic Gradient Descent" ist "Adam" ("Adaptive Movement Estimation"), welche unter anderem in dieser Arbeit genutzt wird [19]. Dieser benutzt vorherige Gradienten, weshalb die Lernrate implizit je nach Steigung der Loss-Funktion angepasst wird (vgl. Gl. 6) [32]. Dabei wird der Durchschnitt aus Gradienten von vorherigen Iterationen h_t und dessen koordinatenweise berechnetes Quadrat h_t^2 verwendet [32]. Die bereits erwähnte "Back Propagation" wird dabei genutzt, um die Gradienten der Loss Funktion im Netz zu berechnen [26]. Diese ermöglicht es mit Hilfe de Kettenregel, die Information über die Kosten rückwärts durch das Netz zurück zu verfolgen [13].

$$\theta_{t+1} = \theta_t - \eta \cdot \text{Adam}(h_t, h_t^2) \tag{6}$$

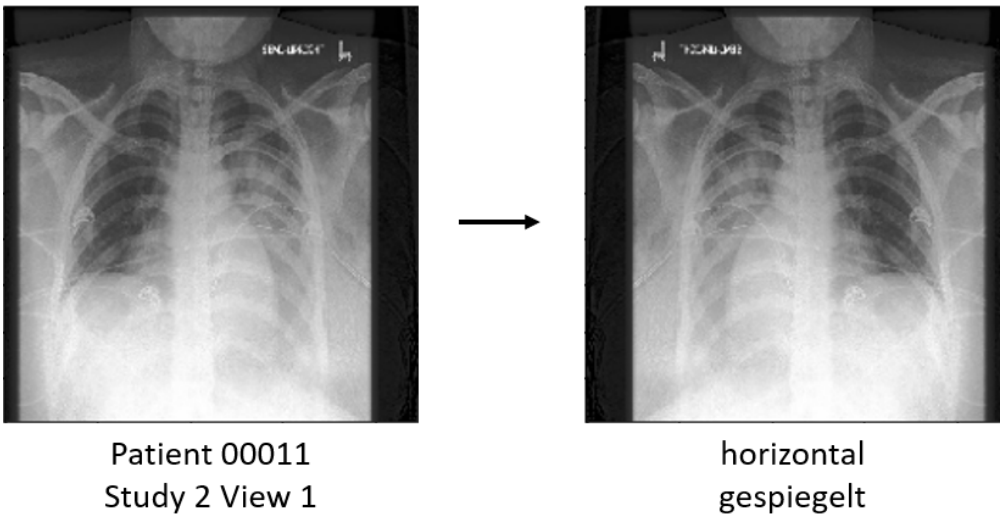


Abbildung 9: Datenaugmentierung am Beispiel einer Röntgen-Aufnahme: Horizontale Spiegelung

2.3 Overfitting

Erzielt das CNN gute Ergebnisse auf das Trainingsset, jedoch nicht auf das Testset, spricht man von “Overfitting” [26]. Das passiert, wenn das neuronale Netz die Trainingsdaten “auswendig lernt”, dadurch aber keine guten Entscheidungen über neue Testdaten treffen kann, die dem CNN noch unbekannt sind. Eine Möglichkeit Overfitting zu verhindern ist Datenaugmentierung, um damit die Anzahl an Trainingsdaten zu vergrößern [22]. Dabei werden die Bilder im Trainingsset beispielsweise gespiegelt, gedreht, verzerrt oder deren Farben verändert, was das Label des Bildes jedoch nicht verändern darf. Die Abbildung 10 zeigt die in dieser Arbeit verwendete Datenaugmentierung an einem Beispiel. Eine vertikale Spiegelung wird in diesem Fall nicht verwendet, da dies nicht der Arbeitsweise eines Radiologen entsprechen würde. Die Auswirkungen von Datenaugmentierungen sind in der Medizin jedoch relativ unerforscht. Zusätzlich kann sogenanntes Early Stopping verwendet werden, welches das Training des Netzes abbricht, falls sich dessen Ergebnisse auf dem Validierungsset nicht weiter verbessern und damit Overfitting verhindern [32].

2.4 Transfer Learning

Bei sogenanntem Transfer Learning werden vortrainierte Neuronale Netze verwendet [32]. Vor allem bei größeren Bildern verursacht das Training viel Rechenaufwand und große Mengen an Trainingsdaten werden benötigt. Daher ist das Ziel von Transfer Learning Informationen von einem Themengebiet auf ein anderes zu übertragen [38]. Beispielsweise können die Gewichte der Netze verwendet werden, die mit ImageNet [11] trainiert wurden. Um Transfer Learning zu nutzen, kann das verwendete Netz mit diesen Gewichten initialisiert werden. In dieser Arbeit sind das die CNNs DenseNet121 [14]

und EfficientNetB4 [36]. Bei einem Dense Convolutional Network (DenseNet) handelt es sich um CNNs, dessen Schichten innerhalb eines Dense Blocks mit allen darauffolgenden Schichten verbunden sind. EfficientNets wurden 2019 vorgestellt, welche durch eine bestimmte Skalierung der Netzwerkdimensionen, eine höhere Genauigkeit und Effizienz erreichen.

2.5 Evaluationsmetriken

Um ein CNN zu bewerten und verschiedene Methoden untereinander vergleichen zu können, sind Metriken notwendig. Beispiele hierfür sind AUC, Precision und Recall. In Abbildung 10 ist die Konfusionsmatrix dargestellt, in der die Vorhersage eines Labels eingeordnet werden kann. Ein Label kann dem wahren Label entsprechend als positiv (TP) bzw. negativ (TN), oder fälschlicherweise als positiv (FP) bzw. negativ (FN) vorhergesagt werden. Der sogenannte Recall (vgl. Gl. 7) spiegelt den Anteil der richtig positiv vorhergesagten Labels von allen tatsächlich positiven Labels wieder [9]. Je näher dieser an 1 liegt, desto genauer klassifiziert das CNN. Ist Recall = 1, werden alle positiven Labels vom Netz als positiv erkannt, dieser beschreibt also die Anzahl an vermeintlichen Treffern. Eine weitere Möglichkeit die Genauigkeit von CNNs zu betrachten sind sogenannte ROC ("Receiver Operating Characteristics") Kurven. Sie bilden das Verhältnis zwischen False Positive Rate fpr (vgl. Gl. 8) und True Positive Rate (Recall) ab [15]. Ein Punkt auf der Kurve entspricht einem Threshold, ab welchem eine vorhergesagte Wahrscheinlichkeit als positiv vorhergesagt gilt [32]. Der AUC-Wert ("Area under the ROC Curve") berechnet sich dann als die Fläche unterhalb der ROC Kurve. In Abbildung 11 sind verschiedene ROC Kurven abgebildet. Darunter ist das Optimum in grün dargestellt, mit einer True Positive Rate von 1, einer False Positive Rate von 0 und einem AUC-Wert bei 1. Ist der AUC = 0,5 (blaue Kurve), handelt es sich um einen Zufallsprozess [32]. In der Regel liegen ROC Kurven zwischen diesen zwei Kurven (beispielsweise rote Kurve). Diese AUC-Werte sind leicht miteinander vergleichbar und sind außerdem unabhängig von dem gewählten Threshold [15].

$$\text{recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{7}$$

$$\text{fpr} = \frac{\text{FP}}{\text{FP} + \text{TN}} \tag{8}$$

Konfusions matrix		Vorhergesagtes Label 1	Vorhergesagtes Label 0
Tatsächliches Label	1	True Positive (TP)	False Negative (FN)
	0	False Positive (FP)	True Negative (TN)

Abbildung 10: Konfusionsmatrix: Stellt die vier Möglichkeiten “True Positive”, “False Positive”, “False Negative” und “True Negative” einer Vorhersage dar

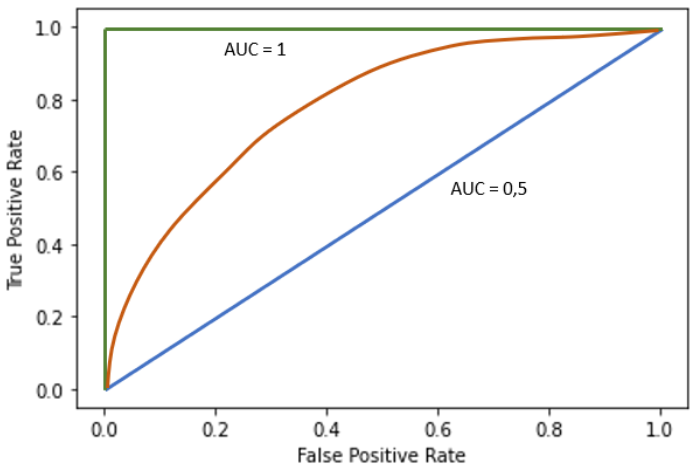


Abbildung 11: verschiedene ROC Kurven: AUC-Wert von 1 entspricht dem Optimum (grün); AUC von 0,5 entspricht einem Zufallsprozess (blau)

3 Verwandte Arbeiten

Diese Arbeit greift das Thema des Papers “Interpreting chest X-rays via CNNs that exploit hierarchical disease dependencies and uncertainty labels” von Hieu H. Pham et al. [27] auf. Dieser nutzt DenseNet121, um die Röntgen-Bilder aus dem CheXpert Datensatz [16] mit 14 Labels zu klassifizieren. Diese Labels sind “No Finding”, “Enlarged Cardiomediastinum”, “Cardiomegaly”, “Lung Opacity”, “Lung Lesion”, “Edema”, “Consolidation”, “Pneumonia”, “Atelectasis”, “Pneumothorax”, “Pleural Effusion”, “Pleural Other”, “Fracture” und “Support Devices” [16]. Dabei ergibt sich “No Finding” aus der Abwesenheit der 12 Pathologien der 14 Klassen (ohne “Support Devices”). Medizinische Datensätze für das Training von CNNs, wie der verwendete CheXpert Datensatz, werden aus bestehenden Arztberichten erstellt. Diese enthalten Klassen, die nicht nur negativ oder positiv gelabelt sein können, sondern auch einem unsicheren Wert zugeordnet sein können, wofür es mehrere Ursachen gibt. Einerseits kann ein solches Label durch die Unsicherheit des Radiologen bei der Diagnosestellung entstehen. Ausdrucksweisen wie “diffuse reticular pattern may represent mild interstitial pulmonary edema” [16] eines Arztes führen in der Erstellung des Datensatzes zu unsicheren Labels. Andererseits kann der Bericht unklar geschrieben sein, indem mehrdeutige Ausdrucksweisen, wie “heart size is stable” [16] genutzt werden. Daher besteht ein Teil der Arbeit von Pham et al. [27] aus dem Testen von Label Smoothing Regularization, indem die Methoden U-Ignore, U-Ones, U-Ones-LSR, U-Zeros und U-Zeros-LSR evaluiert wurden. Je nach verwendeter Methode, werden unsichere Labels vor dem Training auf bestimmte Werte gesetzt. Zusätzlich wird die Idee des konditionalen Trainings auf den CheXpert Datensatz getestet. Hierbei werden die 14 Klassen nicht unabhängig voneinander behandelt, sondern es wird eine Hierarchie der Labels benutzt. Das soll dazu führen, Fachwissen zu nutzen, das bei flachem Training von CNNs nicht genutzt wird. Abbildung 12 zeigt die Hierarchie des CheXpert Datensatzes mit 14 Klassen. Darin sind allgemeinere Pathologien (z.B. “Lung Opacity”) näher an der Wurzel, während die spezielleren Pathologien (z.B. “Pneumonia”) die Blätter bilden. Das heißt, die Pathologien sind, im Gegensatz zu flachem Training, bedingt durch ihre Eltern-Labels. Beispielsweise würde die Präsenz einer “Cardiomegaly” ein “Enlarged Cardiomediastinum” voraussetzen. Diese Voraussetzung ist sinnvoll, da es sich bei einer “Cardiomegaly” um ein vergrößertes Herz handelt, während das sogenannte “Mediastinum” einen Bereich meint, der sowohl das Herz als auch andere Organe, Gefäße etc. einschließt [23]. Um Sicherheit über das exakte Vorgehen in [27] zu erlangen, bestand zudem Kontakt via E-Mail zu Hieu H. Pham. Für den Vergleich der LSR Methoden wurde jede Methode über 5 Epochen trainiert. Das konditionale Training wird zunächst durch die Verwendung eines Teils des Trainingssets realisiert, mit Bildern die hierarchische Voraussetzungen erfüllen (Ist ein Label positiv, muss auch sein Eltern-Label positiv sein). Dieses wir ebenfalls 5 Epochen durchgeführt und anschließend mit dem

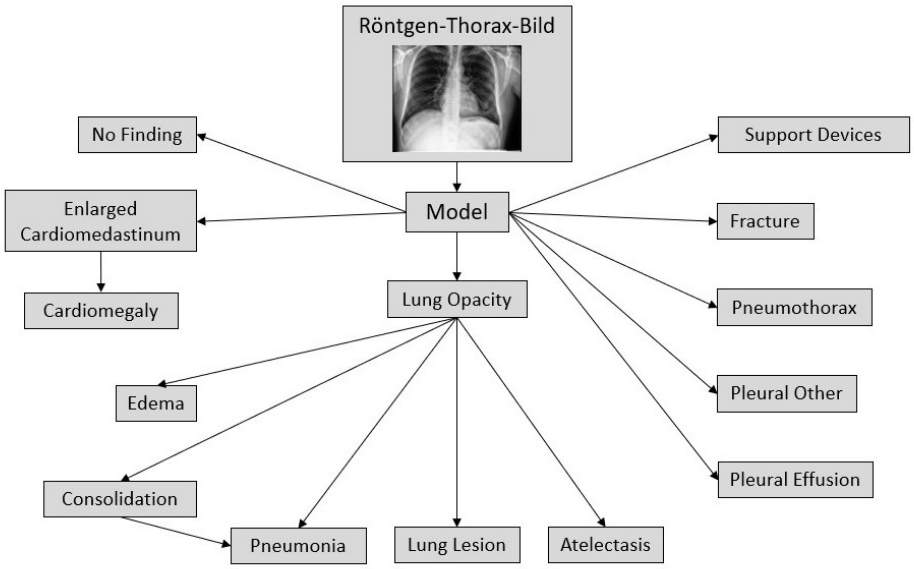


Abbildung 12: Klassen des CheXpert Datensatzes als Hierarchie. Vorgestellt in [16]

ganzen Trainingsset über weitere 5 Epochen weiter verbessert. In der Arbeit von Pham et al. [27] werden die ersten 5 Epochen als “konditionales Training” und die folgenden 5 Epochen als “Transfer Learning” bezeichnet. Diese Bezeichnungen werden in dieser Arbeit übernommen. Pham kommt zu dem Schluss, dass konditionales Training in Verbindung mit U-Ones-LSR mit einem AUC von 0,894 die besten Ergebnisse erzielt. Außerdem erhält er mit einem Ensemble aus 6 Modellen (DenseNet121, DenseNet169, DenseNet201, Inception-ResNet-v2, Xception und NASNetLarge) einen durchschnittlichen AUC-Wert von 0,94. Zusätzlich ist zu beachten, dass Template Matching genutzt wurde um 224x224 Pixel große Bilder zu erhalten und diese anschließend normalisiert wurden. Für das Ensemble Modell wurde außerdem Test-Time Augmentation für die Evaluation auf das Validationset genutzt.

Der Ansatz des konditionalen Trainings stammt aus der Arbeit von Haomin Chen et al. [7], welcher das konditionale Training mit dem manuell gelabelten PLCO Datensatz mit 19 Labels [12] behandelt. Darin wird ebenso ein auf ImageNet vortrainiertes DenseNet121 verwendet, welches in zwei Schritten trainiert wird. Chen realisiert den hierarchischen Ansatz durch die Formulierung verschiedener Loss-Funktionen. Durch eine Binary Cross Entropy Loss-Funktion, welche den Loss nur für Labels mit positivem Eltern-Label berechnet, wird zunächst konditionales Training durchgeführt. Anschließend wird durch einen Loss, der für alle Labels berechnet wird, Fine Tuning durchgeführt. Dieses in zwei Schritten trainierte Modell wird schließlich mit Modellen verglichen, die lediglich konditional oder flach trainiert wurden. Durch durchgeführtes Fine Tuning nach konditionalem Training erzielt [7] einen durchschnittlichen AUC von 0,887 der 14 Labels in den Blättern der Hierarchie und einen durchschnittlichen

AUC von 0.866 über alle 19 Labels. Außerdem erreicht dessen rein konditionales Training bessere Ergebnisse als das rein flache Training, während der Ansatz des Trainings in zwei Schritten, mit konditionalem Training und Fine Tuning, die besten Ergebnisse erzielt. Das Trainieren von Neuronalen Netzen auf hierarchische Daten ist auch in anderen Themenbereichen vertreten. Beispielsweise im Bereich der Text Klassifizierung [2][31][21], um Textdokumente in einer Hierarchie aus Genres einzuordnen. Auch im Bereich der Bioinformatik wird hierarchische Multi-Label Klassifizierung genutzt, zum Beispiel um die Funktionen von Proteinen vorherzusagen [5][6]. Auch Label Smoothing Regularization wurde bereits erfolgreich in anderen Themengebieten verwendet, wie Chorowski und Jaitly in der Spracherkennung [8]. Pereyra et al. [25] evaluiert den Einfluss von Label Smoothing bei verschiedenen Aufgabenstellungen, wie der maschinellen Übersetzung und der Bild Klassifizierung. Beispielsweise auf dem MNIST Datensatz zur Klassifizierung von handschriftlich geschriebenen Zahlen und im Bereich der maschinellen Übersetzung. Durch die Verfügbarkeit verschiedener Datensätze wie CheXpert [16], ChestX-ray8 [37] oder MIMIC-CXR [17], gibt es eine Vielzahl an Arbeiten mit Röntgen-Thorax-Aufnahmen. Um einzelne Pathologien zu erkennen, wie Pneumothorax [35], oder im Bereich der Multi-Label Klassifizierung wie Rajpurkal et al. [28][29] und Irvin et al. [16]. Letzterer stellt in seiner Arbeit den CheXpert Datensatz vor, welcher mit Bildern des Stanford Krankenhauses zwischen Oktober 2002 und Juli 2017 erstellt wurde. Dessen Labels stammen aus den Arztberichten der Bilder, welche automatisch durch einen Labeler extrahiert wurden. Dieser hat außerdem mit DenseNet121 die Label Smoothing Methoden U-Ignore, U-Ones und U-Zeros evaluiert, welche auch in dieser Arbeit behandelt werden. Für diese erzielt er mit einem Ensemble aus 30 Modellen durchschnittliche AUC-Werte von 0,8892 für U-Ignore, 0,8886 für U-Zeros und 0,8928 für U-Ones. Neben [27] und [16] haben auch Allaouzi und Ahmed [1] den CheXpert Datensatz verwendet, um verschiedene Transformation Methoden (“Binary Relevance”, “Label Powerset” und “Classifier Chain”) für die Multi-Label Klassifizierung zu vergleichen. “Binary Relevance” behandelt Labels unabhängig voneinander, indem die Multi-Label Klassifizierung in mehrere binäre Klassifikationsprobleme aufgeteilt wird. Bei “Label Powerset” und “Classifier Chain” werden die Beziehungen zwischen den Labels beachtet. Ersteres transformiert die Multi-Label Klassifizierung in ein Multi-Class Problem, dessen Klassen allen möglichen Kombinationen der ursprünglichen Labels entsprechen. Letzteres funktioniert ähnlich wie “Binary Relevance”, jedoch sind die einzelnen binären Klassifikatoren verbunden und tauschen Informationen der Labels miteinander. Insgesamt wird DenseNet121 als Feature Extractor genutzt, dessen letzte fully-connected Schicht durch die jeweiligen Klassifikatoren ersetzt wurde. Bei [1] erreicht “Binary Relevance” mit 0,812 den größten durchschnittlichen AUC-Wert, dabei wurden unsichere Labels des Datensatzes ignoriert (U-Ignore).

Eine Übersicht der Ergebnisse mit dem CheXpert Datensatz sind in der Tabelle 1 dargestellt. Dabei ist wichtig zu beachten, dass die Ergebnisse von Irvin et al. [16] von

Tabelle 1: Übersicht der Ergebnisse aus den Arbeiten von Irvin et al. [16], Allaouzi et al. [1] und Pham et al. [27].

		Atelectasis	Cardiom.	Consolid.	Edema	Pl. Effusion	Durchschnitt
Irvin et al. [16][a]	U-Ignore	0,818	0,828	0,938	0,934	0,928	0,8892
	U-Zeros	0,811	0,840	0,932	0,929	0,931	0,8886
	U-Ones	0,858	0,832	0,899	0,941	0,934	0,8928
Allaouzi et al. [1][a]	Binary Relevance	0,720	0,880	0,770	0,870	0,900	0,828
	Label Powerset	0,720	0,870	0,770	0,870	0,900	0,826
	Classifier Chain	0,700	0,870	0,740	0,860	0,900	0,814
Pham et al. [27][b]	Ensemble	0,909	0,910	0,957	0,958	0,964	0,940

[a] DenseNet121
[b] DenseNet121, DenseNet169, DenseNet201, Inception-ResNet-v2, Xception und NASNetLarge

einem Ensemble aus 30 Modellen stammen und auf dem Validationset von CheXpert evaluiert wurden. Dagegen nutzt Pham et al. [27] ein Ensamble mit 6 Modellen und das Validationset von CheXpert. Allaouzi und Ahmed [1] teilen den CheXpert Datensatz auf und nutzt 20% der Bilder für das Testen der Transformation Methoden. Außerdem wurden die Röntgen-Aufnahmen normalisiert und horizontale Spiegelungen als Datenaugmentierung genutzt.

4 Methodik

4.1 Verwendeter Datensatz

Für das Training der Netze wurde der CheXpert Datensatz verwendet, welcher aus 224.316 Röntgen-Thorax-Aufnahmen von insgesamt 65.240 Patienten besteht [16]. Im Datensatz sind sowohl frontal, als auch lateral aufgenommene Bilder enthalten. Bei den 14 Klassen von CheXpert handelt es sich um: “No Finding”, “Enlarged Cardiomediastinum”, “Cardiomegaly”, “Lung Opacity”, “Lung Lesion”, “Edema”, “Consolidation”, “Pneumonia”, “Atelectasis”, “Pneumothorax”, “Pleural Effusion”, “Pleural Other”, “Fracture” und “Support Devices”. Dabei ist jede Aufnahme mit 1 (positiv), 0 (negativ) oder -1 (unsicher) jeder der Klassen zugeteilt. Die Beispiele zweier Bilder und deren Labels sind in Abbildung 13 dargestellt. Die Tabelle 2 zeigt die Verteilung des Trainingssets, welches aus 223.414 Röntgen-Aufnahmen besteht, wobei 85.056 dieser Bilder mindestens ein unsicheres Label enthalten. Für das Trainieren der Netze wurden ausschließlich die frontal aufgenommenen Röntgen-Bilder verwendet, weshalb 32.387 laterale Aufnahmen aus dem Trainingsset wegfallen. Der Datensatz enthält außerdem ein Validierungsset mit 200 Patienten und ein Testset mit 500 Studien. Aus dem Validierungsset werden 32 laterale Bilder entfernt, weshalb sich insgesamt ein Trainingsset aus 191.027 und ein Validierungsset aus 202 Röntgen-Aufnahmen ergibt. Das Testset von CheXpert ist jedoch öffentlich nicht direkt zugänglich, weshalb eine Aufteilung des Trainingssets durchgeführt wurde, vergleichbar zu der Arbeit von Allaouzi et al. [1]. Zusätzlich enthält das Validierungsset lediglich Bilder von 200 Patienten. Darin ist beispielsweise kein Röntgenbild mit dem Label “Fracture” enthalten und die Klasse “Lung Lesion” einmal vertreten (siehe Tabelle 3). Die Aufteilung der frontal aufgenommenen Bilder in Trainings-, Validierungs- und Testset wurde daher folgendermaßen durchgeführt: Da die Anzahl der Aufnahmen pro Patient variiert und um Korrelationen zwischen Trainings- und Validierungs- bzw. Testset zu verhindern, wurden die Patienten im Verhältnis 80:10:10 aufgeteilt. Infolgedessen wird Patientenüberlappung zwischen den Sets verhindert. Der Anteil für Validierungs- und Testset enthält jeweils nur Patienten, deren Bilder keine unsicheren Label haben. Für das Validierungsset wurden zu den 200 Patienten weitere 6324 Patienten hinzugefügt. Die Bilder der Aufteilung der Patienten ergeben dann das Trainings-, Validierungs- und Testset für die Experimente dieser Arbeit. Die Verteilungen von Validierungs-, Testset und dem ursprünglichen Validierungsset mit 200 Patienten sind in der Tabelle 3 aufgelistet.

Tabelle 2: Daten des CheXpert Datensatzes. Das Trainingsset enthält 223414 Röntgen-Thorax-Bilder. Für jedes Label ist jeweils angegeben, wie oft es positiv, negativ und unsicher im Trainingsset vorkommt.

	positiv (1)	negativ (0)	unsicher (-1)
No Finding	22381	201033	0
Enlarged Cardiom.	10798	200213	12403
Cardiomegaly	27000	188327	8087
Lung Opacity	105581	112235	5598
Lung Lesion	9186	212740	1488
Edema	52246	158184	12984
Consolidation	14783	180889	27742
Pneumonia	6039	198605	18770
Atelectasis	33376	156299	33739
Pneumothorax	19448	200821	3145
Pleural Effusion	86187	125599	11628
Pleural Other	3523	217238	2653
Fracture	9040	213732	642
Support Devices	116001	106334	1079

Tabelle 3: Daten über das Validation- und Testset: Anzahl der Bilder, Patienten und die Häufigkeiten der verschiedenen Labels. Diese enthalten keine unsicheren Labels.

	Validationset[a]	Validationset[b]	Testset
Bilder	9627	202	9832
Patienten	6524	200	6524
No Finding	3137	56	3098
Enlarged Cardiom.	570	105	511
Cardiomegaly	947	66	942
Lung Opacity	2698	117	2827
Lung Lesion	464	1	445
Edema	1770	42	1934
Consolidation	414	32	430
Pneumonia	259	8	278
Atelectasis	1603	75	1638
Pneumothorax	917	7	916
Pleural Effusion	2411	64	2621
Pleural Other	116	1	115
Fracture	530	0	552
Support Devices	4175	99	4322

[a] Validationset der Aufteilung aus Kapitel 4.1
[b] Ursprüngliches Validationset des CheXpert Datensatzes mit 200 Patienten

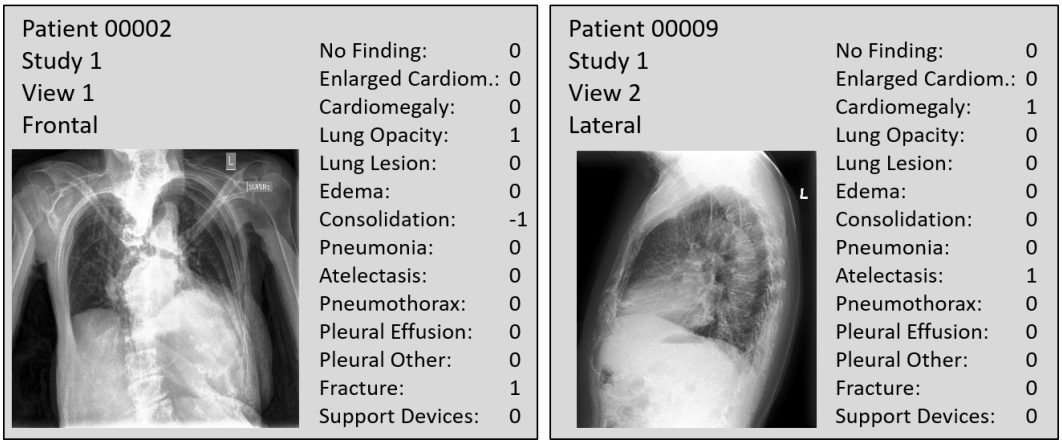


Abbildung 13: Röntgen-Aufnahmen zweier Patienten und deren Labels aus dem CheXpert Datensatz

4.2 Aufbau der Netze und allgemeine Trainingsprozedur

Für das Training der Modelle wurden TensorFlow 2.3.0 und vortrainierte CNNs aus der Keras Bibliothek verwendet. Sowohl die Label Smoothing Methoden als auch das konditionale Training wurden mit "EfficientNetB4" [36] und "DenseNet121" [14] umgesetzt, welche mit vortrainiertem ImageNet initialisiert sind. Die verwendete Architektur der Modelle ist in der Abbildung 14 dargestellt: Angefügt an das vortrainierte EfficientNet bzw. DenseNet ist ein Layer, welches Global Average Pooling realisiert. Die letzte Schicht besteht aus einem 13-dimensionalen Dense-Layer, dessen Ausgabe die Wahrscheinlichkeiten der 13 Klassen ausgibt. Angelehnt an [29], wurden als Datenaugmentierung die Eingabebilder horizontal gespiegelt und zusätzlich im "Preprocessing"-Schritt durch den Mittelwert und die Standardabweichung von ImageNet normalisiert. Die Netze wurden mit einer Batch size von 16 und dem Adam Optimizer mit den Parametern $\beta_1 = 0,9$ und $\beta_2 = 0,999$ trainiert. Als Aktivierungsfunktion des Dense-Layers wurde die Sigmoidfunktion und als Verlustfunktion die "Binary Cross-Entropy"-Funktion genutzt (siehe Kapitel 2). Trainiert wurde mit einer Lernrate von $1e^{-4}$, welche nach jeder Epoche um den Faktor 10 reduziert wurde.

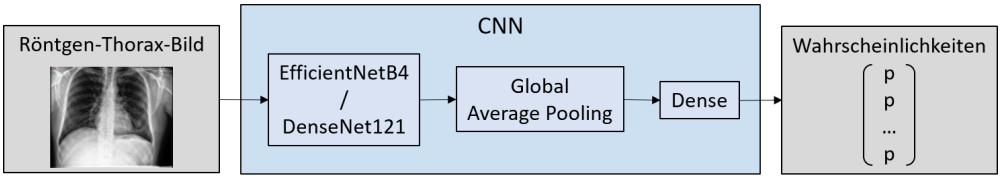


Abbildung 14: Aufbau der verwendeten Modelle für die Label Smoothing Methoden und das konditionale Training: Diese bestehen aus einem vortrainierten EfficientNetB4 bzw. DenseNet121, Global Average Pooling und einem Dense Layer. Das Netz berechnet aus dem Eingabebild die Wahrscheinlichkeiten für jedes Label.

4.3 Label Smoothing Regularization

Ein Teil dieser Arbeit besteht darin, die Auswirkungen von Label Smoothing Regularization zu untersuchen. Daher wurden die Label Smoothing Methoden U-Ones-LSR und U-Zeros-LSR getestet, welche in dem Paper von Pham et al. [27] verwendet wurden. Das sind Ansätze mit sogenannter Label Smoothing Regularization (LSR), bei der die unsicheren Labels nicht auf einen harten Wert von 1 oder 0 gesetzt werden, sondern auf eine zufällige Zahl nahe bei 1 bzw. 0. Um den Effekt von LSR zu bestimmen, wurden darüber hinaus U-Ignore, U-Ones und U-Zeros genutzt, welche unsichere Labels ignorieren oder durch feste Werte ersetzt. Zusätzlich soll eine neue LSR-Methode U-Random evaluiert werden, welche in [27] nicht untersucht wird. Die verschiedenen Methoden unterscheiden sich jeweils lediglich in den Werten der unsicheren Labels, weshalb für das Training diese unsicheren Labels im erstellten Trainingssets je nach Methode ersetzt wurden:

- U-Ignore: Es werden lediglich Bilder ohne unsichere Labels verwendet, daher wurde jedes Bild mit mindestens einem unsicheren Label aus dem Trainingsset entfernt.
- U-Ones: Unsichere Labels werden als positiv angenommen, weshalb alle unsicheren Labels im Trainingsset auf 1 gesetzt wurden.
- U-Ones-LSR: Unsichere Labels werden auf zufällige Werte aus dem Bereich [0.55, 0.85] (vgl. [27]) gesetzt.
- U-Ones: Unsichere Labels werden als negativ angenommen, weshalb alle unsicheren Labels im Trainingsset auf 0 gesetzt wurden.
- U-Ones-LSR: Unsichere Labels werden auf zufällige Werte aus dem Bereich [0, 0.3] (vgl. [27]) gesetzt.
- U-Random: Unsichere Labels werden durch zufällige Werte zwischen 0 und 1 ersetzt.

Das daraus resultierende Trainingsset, welches aus 171.770 Röntgen-Aufnahmen von 51.686 Patienten besteht, ist in Tabelle 4 dargestellt. Eine Ausnahme ist die Methode U-Ignore mit 99.029 Bildern von 36.556 Patienten, bei der zusätzlich alle Bilder mit mindestens einem unsicheren Label aus dem Trainingsset entfernt wurden.

Jede Methode wurde jeweils sowohl mit EfficientNetB4 als auch mit DenseNet121 trainiert. Das Training der verschiedenen Label Smoothing Methoden unterscheidet sich dabei lediglich durch die Daten im Trainingsset, genauer um die Werte der unsicheren Labels. Anhand des Validierungssets wird jeweils das Model mit dem niedrigsten "Loss" gespeichert. Das Training wird durch Early Stopping beendet, wenn sich dieser über 15 Epochen nicht verbessert.

Tabelle 4: Daten über das erstellte Trainingsset: Anzahl der Bilder, Patienten und die Häufigkeiten der verschiedenen Labels.

	U-Ignore	U-Ones	U-Ones-LSR[b]	U-Zeros	U-Zeros-LSR[b]	U-Random[b]
Bilder	99029	171770	171770	171770	171770	171770
Patienten	36556	51686	51686	51686	51686	51686
No Finding[a]	14882	14979	14979	23839	15054	15003
Enlarged Cardiom.	5070	18497	8211+10286	8211	8211+10095	8267+10179
Cardiomegaly	13619	28269	21562+6707	21562	21562+6582	21592+6644
Lung Opacity	42976	93234	88803+4431	88803	88803+4368	88829+4375
Lung Lesion	3489	7240	6132+1108	6132	6132+1085	6140+1092
Edema	30443	57831	46013+11818	46013	46013+11597	46058+11714
Consolidation	6934	36552	12171+24381	12171	12171+23987	12286+24115
Pneumonia	2445	20127	4146+15981	4146	4146+15724	4237+15808
Atelectasis	18151	56417	26554+29863	26554	26554+29360	26713+29548
Pneumothorax	11876	18575	15867+2708	15867	15867+2668	15877+2681
Pleural Effusion	45157	81509	71931+9578	71931	71931+9439	71988+9469
Pleural Other	1197	4087	2275+1812	2275	2275+1784	2288+1787
Fracture	3907	6853	6354+499	6354	6354+492	6356+492
Support Devices	60002	99687	98772+915	98772	98772+898	98775+911

[a] Die Klasse “No Finding” ist positiv, wenn alle Pathologien (ohne “Support Devices”) negativ sind.
[b] Vor dem “+” steht die Anzahl an Bildern der Pathologie mit dem sicheren Wert 1. Nach dem “+” steht die Anzahl an Bildern der Pathologie mit unsicheren Werten zwischen 0 und 1.

4.4 Konditionales Training

Das Training der Modelle teilt sich in zwei Schritte auf. Für den ersten Schritt des Trainings werden 5 Epochen mit konditionalen Daten trainiert. Die anfängliche Lernrate von $1e^{-4}$ wird nach jeder Epoche um den Faktor 10 reduziert. Nach diesem konditionalen Training werden alle Schichten des Netzes, bis auf die letzte Klassifizierungsschicht “eingefroren”. Das heißt, nur das an DenseNet121 bzw. EfficientNetB4 angefügte Output Layer lernt im nächsten Schritt. In diesem werden weitere 5 Epochen mit einer Lernrate von $1e^{-4}$ trainiert, die wie beim konditionalen Training nach jeder Epoche reduziert wird. Wichtig sind die verwendeten Daten in den beiden Schritten. In diesem sogenannten Transfer Learning (übernommene Terminologie von Pham et al. [27]), werden für das Training alle Daten genutzt (Trainingset aus Kapitel 4.3). Für das konditionale Training wurden nicht konditionale Daten aus dem Trainingsset aussortiert, das heißt es werden Röntgen-Aufnahmen ignoriert, die negative Eltern-Labels enthalten [27]. Dafür werden die Klassen in den Blättern der Hierarchie mit Eltern-Labels betrachtet (“Cardiomegaly”, “Lung Lesion”, “Edema”, “Consolidation”, “Pneumonia”, “Atelectasis”). Ist eines dieser Labels positiv, muss auch das Eltern-Label dieser Klasse positiv sein (“Enlarged Cardiomediastinum”, “Lung Opacity”, “Consolidation”), sonst wird die Röntgen-Aufnahme nicht für das Konditionale Training verwendet. Ist beispielsweise in einem Bild “Atelectasis” positiv und dessen Eltern-Label “Lung Opacity” negativ, wird es aussortiert. Ein Beispiel hierfür ist das Röntgen-Bild in Abbildung 15b. Im Vergleich dazu, wird das Bild in der Abbildung 15a für das konditionale Training

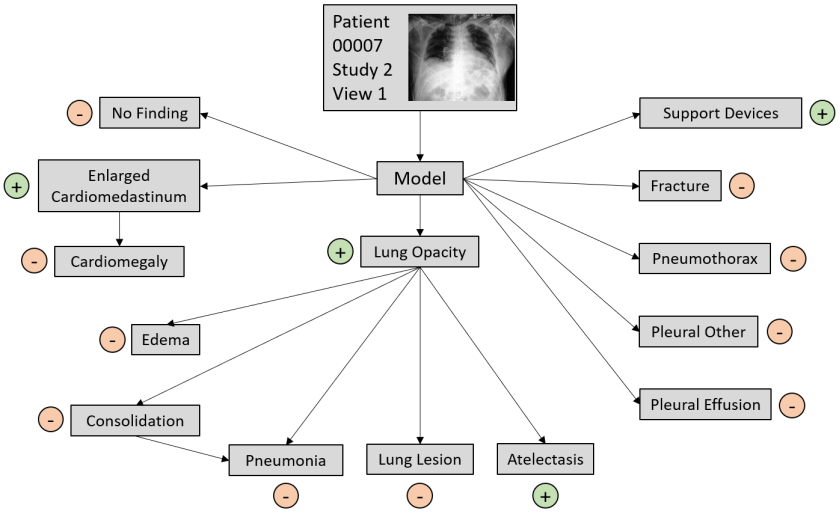
Tabelle 5: Daten über das Trainingsset für das konditionale Training: Anzahl der Bilder, Patienten und die Häufigkeiten der verschiedenen Labels.

	U-Ignore	U-Ones	U-Ones-LSR[b]	U-Zeros	U-Zeros-LSR[b]	U-Random[b]
Bilder	62680	98065	118525	116466	118500	118367
Patienten	28564	38724	44368	43926	44365	44337
No Finding[a]	14882	14979	14979	23839	23839	18148
Enlar. Cardiom.	3926	13141	6687+8241	6598	6684+8091	6730+8141
Cardiomegaly	807	1637	1393+4963	1161	1388+4874	1391+4910
Lung Opacity	36021	67390	76587+3815	76095	76578+3759	76524+3767
Lung Lesion	1644	3420	3472+899	3308	3470+881	3475+880
Edema	11191	20875	20004+9197	19210	19994+9021	19998+9082
Consolidation	2266	19530	4872+19103	4189	4864+18784	4936+18872
Pneumonia	184	2128	668+12906	293	661+12689	672+12753
Atelectasis	6127	25996	10503+24133	10042	10496+23724	10603+23818
Pneumothorax	9510	13744	12922+2172	12832	12921+2141	12925+2149
Pleural Effusion	27867	46163	48103+6902	47258	48096+6811	48092+6811
Pleural Other	864	2679	1769+1415	1727	1769+1391	1773+1397
Fracture	2934	4652	4902+402	4835	4902+396	4898+398
Support Devices	37698	57729	67250+578	66199	67234+563	67179+577

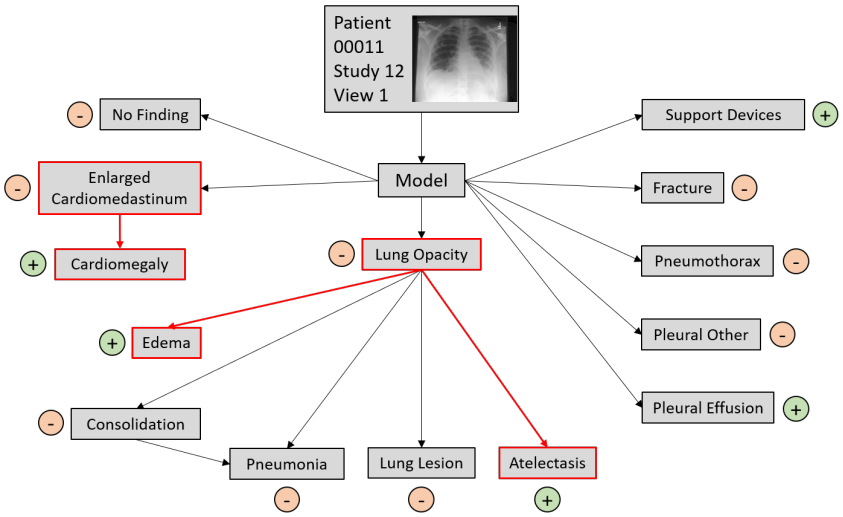
[a] Die Klasse “No Finding” ist positiv, wenn alle Pathologien (ohne “Support Devices”) negativ sind.
[b] Vor dem “+” steht die Anzahl an Bildern der Pathologie mit dem sicheren Wert 1. Nach dem “+” steht die Anzahl an Bildern der Pathologie mit unsicheren Werten zwischen 0 und 1.

benutzt, da die Eltern-Labels von positiven Labels selbst positiv sind. Das daraus resultierende Trainingsset ist in Tabelle 5 beschrieben. Durch die Bedingung der positiven Eltern-Labels verringert sich die Anzahl der Bilder im Trainingsset. Dabei ist bei U-Ones-LSR, U-Zeros-LSR und U-Random zu beachten, dass für das Aussortieren aus dem Trainingsset ein negatives Eltern-Label mit dem Wert 0, bei positivem Kind-Label mit dem Wert 1, Voraussetzung war. Das heißt, ein Röntgen-Bild mit beispielsweise “Edema”= 0,75 und dessen Eltern-Label “Lung Opactiy”= 0 wird in das konditionale Trainingsset aufgenommen. Ebenso wird ein Bild mit den Werten “Enlarged Cardiomediastinum” = 0,2 und “Cardiomegaly” = 1 nicht aus dem Trainingsset aussortiert.

Um den Vergleich von konditionalem Training mit flachem Training (Kapitel 4.3) zu ermöglichen, wurde das flache Training mit 10 Epochen (ohne Early Stopping) wiederholt. Hierfür wurde, ebenso wie beim konditionalen Training, 5 Epochen das gesamte Netz trainiert. Jedoch wurden dabei im Gegensatz zum konditionalen Training alle Daten im Trainingsset genutzt. Anschließend wurden die Layer eingefroren, damit nur das letzte Dense-Layer bei dem darauf folgenden Transfer Learning für weitere 5 Epochen trainiert wird. In [27] wurden für den Vergleich mit konditionalem Training Modelle über 5 Epochen flach trainiert, ohne anschließendes Transfer Learning durchzuführen. Daher wurde flaches Training ebenso über 5 Epochen durchgeführt.



(a) Labels einer Röntgen-Aufnahme des Patienten 00007. Das Bild wird für das konditionale Training verwendet, da jedes positive Kind-Label ein positives Eltern-Label hat.



(b) Labels einer Röntgen-Aufnahme des Patienten 00011. Das Bild wird nicht für das konditionale Training verwendet, da die Eltern-Labels “Lung Opacity” und “Enlarged Cardiomediastinum” negativ sind.

Abbildung 15: Beispiele jeweils einer Röntgen-Aufnahme, welche die Bedingungen für konditionales Training erfüllt bzw. nicht erfüllt.

4.5 Inferenz und Evaluation

Für die Evaluation der verschiedenen Modelle wurde das in Kapitel 4.1 beschriebene Testset verwendet. Dessen Röntgen-Aufnahmen wurden von dem jeweiligen Netz vorhergesagt. Aus den berechneten Wahrscheinlichkeiten und den eigentlichen Labeln wurden zunächst die ROC Kurven und AUC-Werte der Labels berechnet. Zusätzlich wurde die Evaluation auf dem Validierungsset von CheXpert (Kapitel 4.1) wiederholt, um den Vergleich mit den Ergebnissen in [27] zu ermöglichen.

Eine Besonderheit ist dabei, wie in [27] beschrieben, beim konditionalen Training zu beachten: Der Output des jeweiligen Netzes entspricht durch das konditionale Trai-

ning den bedingten Wahrscheinlichkeiten der einzelnen Labels. Das heißt, das Netz liefert beispielsweise die Wahrscheinlichkeit für eine “Cardiomegaly”, unter der Voraussetzung eines “Enlarged Cardiomediastinums”. Daher müssen die unbedingten Wahrscheinlichkeiten der einzelnen Labels beim Testen des Netzes berechnet werden. Diese können nach Bayes berechnet werden, indem die Wahrscheinlichkeiten entlang des Pfades in der Hierarchie bis zur Wurzel multipliziert werden. Nimmt man eine Hierarchie mit der Klasse A, dessen Kind-Label B und wiederum dessen Kind-Label C an, zeigt die Gleichung 11 diese Berechnung für B aus der Wahrscheinlichkeit des Eltern-Labels A. Während Gleichung 12 dies für C aus dessen Eltern-Labels A und B zeigt. Für tiefere Hierarchien erfolgt die Berechnung analog. Diese ist am Beispiel von “Consolidation”, “Pneumonia” und “Edema” ist in der Abbildung 16 dargestellt.

$$P(B) = P(A) \cdot P(B|A) \tag{9}$$

$$P(C) = P(A) \cdot P(B|A) \cdot P(C|B, A) \tag{10}$$

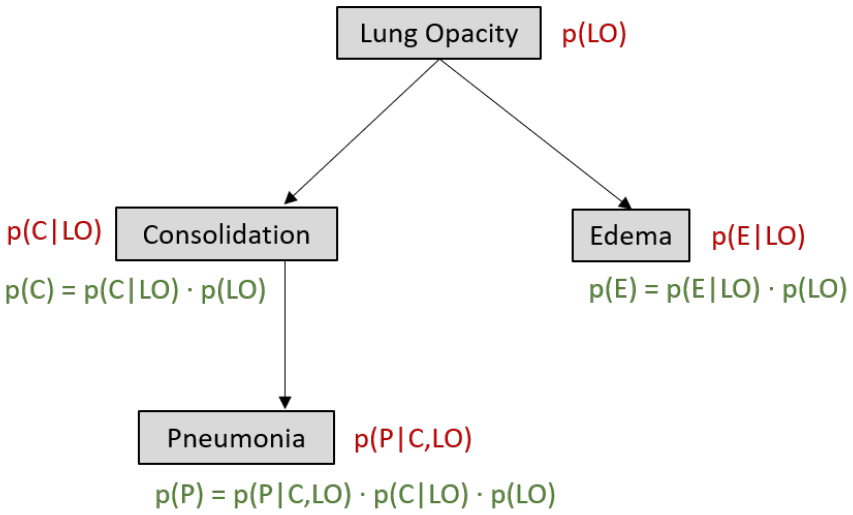


Abbildung 16: Berechnung der unbedingten Wahrscheinlichkeiten (grün) aus den vom Netz gelieferten Wahrscheinlichkeiten (rot) für “Consolidation”, “Pneumonia” und “Edema”

5 Ergebnisse

Im Folgenden werden die Ergebnisse aus der Evaluation der Label Smoothing Methoden und des konditionalen Trainings dargestellt.

5.1 Label Smoothing Regularization

In Tabelle 6 sind die Ergebnisse der Architekturen DenseNet121 und EfficientNetB4 mit den Label Smoothing Methoden aufgelistet. Dargestellt ist jeweils der AUC-Wert für jedes Label und der Durchschnitt aller 13 Labels. Gekennzeichnet ist jeweils die Methode mit dem besten Wert pro Architektur und Label (AUC fett gedruckt). Für die Vorhersagen wurde das Testset bestehend aus 9.832 Röntgen-Aufnahmen genutzt, wie in Kapitel 4.5 beschrieben. Wie erwartet schneidet U-Ignore bei beiden Netzen am schlechtesten ab, da alle fünf anderen Methoden einen größeren AUC-Wert erzielen. Bei dem Training mit DenseNet121 ist im Durchschnitt die Methode U-Zeros-LSR mit 0,82109 am besten. Dagegen ist U-Zeros im Schnitt mit 0,822782 bei der Verwendung von EfficientNetB4 am besten. Vergleicht man die zusätzliche Label Smoothing Regularization (U-Ones-LSR und U-Zeros-LSR) mit den Methoden U-Ones und U-Zeros, ergibt sich für das DenseNet121 durchschnittlich eine geringe Verbesserung. Der AUC-Wert von U-Zeros verbessert sich mit LSR von 0,819551 auf 0,82109 und bei U-Ones von 0,814675 auf 0,81829. Betrachtet man EfficientNetB4, verbessert sich U-Ones mit LSR von 0,817805 auf 0,821193. Bei U-Zeros ergibt sich im Durchschnitt keine Verbesserung, da U-Zeros (0,822782) minimal besser als U-Zeros-LSR (0,822099) ist. Dabei ist zu beachten, dass es sich hier um Durchschnittswerte handelt. Beispielsweise verbessert sich das Label “Enlarged Cardiomediastinum” durch U-Ones-LSR um ca. 0,01392 im Vergleich zu U-Ones (DenseNet121). Ebenso verschlechtert sich “Fracture” durch LSR um ca. 0,00981 im Vergleich zu U-Ones (DenseNet121). Die neue Label Smoothing Methode U-Random erhält mit 0,818714 (DenseNet121) und 0,82194 (EfficientNetB4) Ergebnisse in der Größenordnung der anderen Methoden. Für einzelne Labels, wie “Fracture”, “Pneumothorax” und “Lung Opacity” hat U-Random den höchsten AUC-Wert. Außerdem ist interessant, dass im Durchschnitt bei jeder Label Smoothing Methode EfficientNetB4 einen größeren AUC-Wert erreicht als DenseNet121.

Zusätzlich wurden die trainierten Netze auf dem Validierungsset von CheXpert evaluiert, dessen Ergebnisse in der Tabelle 6 aufgelistet wurden. Darin erhält U-Ignore, bei einer Verwendung von DenseNet121, mit 0,860928 im Durchschnitt ein besseres Ergebnis als U-Zeros. Bei EfficientNetB4 dagegen erzielt U-Ignore schlechtere Ergebnisse als die restlichen Methoden. Im Gegensatz zu der Evaluation auf dem Testset, ist die Methode U-Ones-LSR mit 0,868564 bei der Verwendung von DenseNet121 und U-Random bei der Verwendung von EfficientNetB4 am besten. Die zusätzliche Label Smoothing Regularization bewirkt eine kleine durchschnittliche Verbesserung im Vergleich zu U-Ones

und U-Zeros. U-Zeros-LSR erzielt 0.866051, während U-Zeros einen AUC von 0.855539 mit DenseNet121 erhält. Mit EfficientNetB4 verbessert sich dieser von 0.860712 bei U-Zeros auf 0.865783. U-Ones-LSR erzielt mit 0.868564 (DenseNet121) und 0.862843 (EfficientNetB4) eine kleine Verbesserung verglichen mit 0,864556 (DenseNet121) und 0.862111 (EfficientNetB4) bei U-Ones. Die neue Methode U-Random erhält mit 0.862181 bei DenseNet121 und 0.869547 bei EfficientNetB4, wie bei der Evaluation auf dem Testset, Ergebnisse in der Größenordnung der anderen Methoden. Im Gegensatz zu der Tabelle 6, erzielt EfficientNetB4 durchschnittlich keine höheren AUC-Werte als DenseNet121 in allen Methoden. U-Ignore, U-Zeros-LSR, U-Ones und U-Ones-LSR sind im Durchschnitt unter der Verwendung von DenseNet121 besser.

Insgesamt zeigt sich also eine minimale Verbesserung durch die Verwendung von Label Smoothing Regularization, unter der Ausnahme von U-Zeros-LSR mit EfficientNetB4 bei der Evaluation auf dem Testset. Während bei der Evaluation auf dem Testset EfficientNetB4, bei der Verwendung aller Methoden, durchschnittlich bessere Ergebnisse als DenseNet121 erzielt, ist dieser Trend bei der Evaluation auf dem kleineren Validierungsset nicht erkennbar. U-Random erhält, sowohl bei der Evaluation auf dem Validierungsset als auch auf dem Testset, vergleichbare Ergebnisse zu den anderen LSR-Methoden. Bei beiden Evaluationen erhält U-Random den besten AUC-Wert für einzelnen Pathologien.

Tabelle 6: AUC Werte der Label Smoothing Methoden mit DenseNet121 für alle 13 Klassen, evaluiert auf das Testset mit 6524 Patienten

	Enl. Cardiom.	Cardiomegaly	Lung Opacity	Lung Lesion	Edema	Consolidation	Pneumonia
U-Ignore [DenseNet]	0.715233	0.884197	0.778133	0.755845	0.882706	0.80831	0.773106
U-Ones [DenseNet]	0.695742	0.889659	0.778704	0.767911	**0.886007**	0.803388	0.780075
U-Ones-LSR [DenseNet]	0.709669	0.892762	0.777747	0.769354	0.885684	0.810492	0.780817
U-Zeros [DenseNet]	0.720325	0.889166	0.777209	**0.775268**	0.881331	0.813875	0.789542
U-Zeros-LSR [DenseNet]	**0.721086**	**0.89306**	0.778918	0.773724	0.883528	**0.816486**	**0.792853**
U-Random [DenseNet]	0.71227	0.890742	**0.77903**	0.772555	0.884318	0.816388	0.785884
U-Ignore [EfficientNet]	**0.719559**	0.882166	0.776924	0.75157	0.878359	0.808075	0.783889
U-Ones [EfficientNet]	0.713633	0.885402	0.780537	0.77249	0.884747	0.803427	0.786459
U-Ones-LSR [EfficientNet]	0.71562	**0.887526**	**0.781936**	0.769543	**0.885426**	0.812966	0.785552
U-Zeros [EfficientNet]	0.718352	0.886634	0.77975	**0.776223**	0.880707	0.817979	**0.797713**
U-Zeros-LSR [EfficientNet]	0.718972	0.885724	0.778482	0.76901	0.882549	**0.821966**	0.793085
U-Random [EfficientNet]	0.719234	0.886212	0.780286	0.774953	0.884516	0.816315	0.788263

	Atelectasis	Pneumothorax	Pleural Effusion	Pleural Other	Fracture	Support Devices	Durchschnitt
U-Ignore [DenseNet]	0.747925	0.845944	0.916938	0.784271	0.763123	0.86841	0.809549
U-Ones [DenseNet]	0.745235	0.859202	0.919398	0.806173	0.776008	0.88327	0.814675
U-Ones-LSR [DenseNet]	0.749671	0.865109	0.91965	**0.811986**	0.777981	**0.88685**	0.81829
U-Zeros [DenseNet]	0.751382	0.861858	0.918985	0.805196	**0.78593**	0.884097	0.819551
U-Zeros-LSR [DenseNet]	**0.757189**	0.865606	**0.921198**	0.808891	0.776112	0.885519	**0.82109**
U-Random [DenseNet]	0.74981	**0.867012**	0.920285	0.801371	0.779937	0.883675	0.818714
U-Ignore [EfficientNet]	0.754402	0.85592	0.913468	0.801174	0.7729	0.871535	0.809549
U-Ones [EfficientNet]	0.744292	0.864895	0.919442	0.80834	0.785287	0.88252	0.814675
U-Ones-LSR [EfficientNet]	0.754381	0.864711	0.918904	**0.819178**	0.794686	0.885083	0.81829
U-Zeros [EfficientNet]	**0.758499**	**0.872092**	0.920298	0.810123	0.791525	**0.886266**	0.819551
U-Zeros-LSR [EfficientNet]	0.758349	0.868657	**0.92149**	0.815493	0.788625	0.884887	**0.82109**
U-Random [EfficientNet]	0.754563	0.868278	0.919686	0.807675	**0.799938**	0.885294	0.818714

Tabelle 7: AUC Werte der Label Smoothing Methoden mit DenseNet121 für 5 Klassen, evaluiert auf das Validierungsset von CheXpert mit 200 Patienten

	Atelectasis	Cardiomegaly	Consolidation	Edema	Pleural Effusion	Durchschnitt
U-Ignore [DenseNet]	0.75895	0.812166	**0.898713**	0.908631	**0.926178**	0.860928
U-Ones [DenseNet]	**0.800735**	0.812611	0.855699	**0.930952**	0.922781	0.864556
U-Ones-LSR [DenseNet]	0.799475	**0.83389**	0.860478	0.925744	0.923234	**0.868564**
U-Zeros [DenseNet]	0.741102	0.804144	0.894301	0.923065	0.915082	0.855539
U-Zeros-LSR [DenseNet]	0.773018	0.815508	0.896507	0.921875	0.923347	0.866051
U-Random [DenseNet]	0.786667	0.816845	0.85864	0.925744	0.923007	0.862181
U-Ignore [EfficientNet]	0.769239	0.786319	0.886765	0.909077	0.92663	0.855606
U-Ones [EfficientNet]	**0.815643**	0.801805	0.829228	**0.936458**	0.927423	0.862111
U-Ones-LSR [EfficientNet]	0.798005	0.803699	0.85	0.932143	**0.930367**	0.862843
U-Zeros [EfficientNet]	0.737428	0.810829	**0.90625**	0.925595	0.92346	0.860712
U-Zeros-LSR [EfficientNet]	0.775433	0.804479	0.893934	0.924702	**0.930367**	0.865783
U-Random [EfficientNet]	0.799685	**0.814505**	0.875551	**0.936458**	0.921535	**0.869547**

5.2 Konditionales Training

Die Tabelle 8 enthält die durchschnittlichen AUC-Werte von DenseNet121 und EfficientNetB4 mit der Evaluation auf dem Testset. Die erste Spalte enthält die Ergebnisse nach einem flachen Training über 5 Epochen. Die zweite Spalte enthält die Ergebnisse nach 5 Epochen flachem Training und anschließenden 5 Epochen Transfer Learning. Im Vergleich dazu sind in der dritten Spalte die Ergebnisse des konditionalen Trainings mit anschließendem Transfer Learning dargestellt. Der größte Wert dieser drei Modelle ist jeweils hervorgehoben. Die Werte der einzelnen Labels sind im Anhang (Kapitel 7) abgebildet. Die Tabellen 10 und 11 zeigen die Werte des flachen Trainings über 5 Epochen. Das flache Training mit anschließendem Transfer Learning ist in den Tabellen 12 und 13 abgebildet. In den Tabellen 14 und 15 sind die Ergebnisse des konditionalen Trainings enthalten. Vergleicht man das flache und das konditionale Training, ist der AUC-Wert von DenseNet121 und EfficientNetB4 aller Label Smoothing Methoden des flachen Trainings größer. In der Tabelle 8, also den Ergebnissen auf dem Testset, hat das flache Training mit anschließendem Transfer Learning die größten Werte. Allerdings unterscheiden sich die Ergebnisse des flachen Trainings mit und ohne Transfer Learning nur um durchschnittlich 0,000509. Das zusätzliche Transfer Learning bewirkt daher keine wesentlichen Verbesserungen. Dennoch erzielt das konditionale Training niedrigere Werte als das flache Training ohne Transfer Learning. Dieser Trend zeigt sich, sowohl bei der Verwendung von DenseNet121, als auch bei der Verwendung von EfficientNetB4 für alle LSR-Methoden.

Tabelle 8: Vergleich von flachem und konditionalem Training durch die Evaluation von DenseNet121 und EfficientNetB4 auf dem Testset: Durchschnitt der AUC-Werte der 13 Labels

	Flach	Flach mit TL	Konditional
U-Ignore [DenseNet]	0.811452	**0.811996**	0.780216
U-Ones [DenseNet]	0.818086	**0.818216**	0.792321
U-Ones-LSR [DenseNet]	0.819004	**0.819382**	0.800743
U-Zeros [DenseNet]	0.820507	**0.82128**	0.795336
U-Zeros-LSR [DenseNet]	0.822213	**0.823106**	0.802866
U-Random [DenseNet]	0.822353	**0.822446**	0.800804
U-Ignore [EfficientNet]	0.810595	**0.81154**	0.787536
U-Ones [EfficientNet]	0.814844	**0.815125**	0.793706
U-Ones-LSR [EfficientNet]	0.81886	**0.819186**	0.799409
U-Zeros [EfficientNet]	0.819856	**0.82039**	0.798058
U-Zeros-LSR [EfficientNet]	0.818601	**0.819325**	0.800351
U-Random [EfficientNet]	0.819274	**0.819765**	0.801729

6 Diskussion

U-Ignore. Die Ergebnisse der Experimente bestätigen die Funde von Pham et al. [27], dass unsichere Labels nicht ignoriert werden sollten. Die verschiedenen Label Smoothing Methoden erzielen bessere Ergebnisse als U-Ignore, da bei dieser Methode zu viele Röntgen-Bilder aus dem Trainingsset wegfallen. Ist bei einem Bild eine Pathologie unsicher, wird das Bild nicht für das Training verwendet und die Informationen der restlichen sicheren Labels werden nicht genutzt. Dabei handelt es sich um 72.741 Bilder, die nicht für das Training benutzt werden (99.029 Bilder bei U-Ignore im Vergleich zu 171.770 Bilder bei den anderen Methoden siehe Tabelle 4). Das sind mehr als 42% des Trainingssets nach der Aufteilung, wie im Kapitel 4.3 beschrieben. Dies zeigt sich bei der Evaluation des flachen Trainings auf dem Testset in den Tabellen 6, 10 und 12, d.h. sowohl bei einem Training mit Early Stopping, als auch bei flachem Training über 5 Epochen mit und ohne Transfer Learning. Nur bei der Pathologie "Enlarged Cardiomediastinum" erzielt U-Ignore gute Ergebnisse im Vergleich zu den anderen Methoden: In der Tabelle 10 ist lediglich die LSR-Methode U-Random besser als U-Ignore, während in der Tabelle 12 lediglich U-Zeros einen höheren AUC-Wert erhält, jeweils bei einer Verwendung von DenseNet121. Dieser Trend ist auch in der Tabelle 6 erkennbar, in welcher U-Ignore den besten Wert bei einer Verwendung von EfficientNetB4 erzielt. Dennoch ist U-Ignore bei Pham et al. [27] im Durchschnitt besser als U-Ones und U-Zeros (siehe Tabelle 11), bei der Evaluation auf dem Validierungsset mit 200 Patienten. Ein ähnlicher Trend zeigt sich bei der Evaluation des flachen Trainings auf dem Validierungsset, jeweils bei der Verwendung von DenseNet121: In der Tabelle 7 ist dieser bei U-Zeros und in den Tabellen 11 und 13 bei U-Ones zu sehen. In diesen Fällen scheint das Netz zu sehr in eine Richtung (positiv bei U-Ones bzw. negativ bei U-Zeros) beeinflusst zu werden, die nicht der Wahrheit entspricht. Außerdem scheint ein wichtiger Punkt das gewählte Datenset für das Testen der Modelle zu sein, da dieser einen signifikanten Einfluss auf die Ergebnisse hat.

Label Smoothing Regularization bei U-Ones und U-Zeros. Die Verwendung von Label Smoothing Regularization bewirkt Verbesserungen auf die Ergebnisse der Modelle. In der Tabelle 11 sieht man diese Verbesserung durch LSR bei den Ergebnissen von Pham et al. [27] im Durchschnitt um 0,0144 bei U-Ones und um 0,0132 bei U-Zeros. Laut [27], soll das zusätzliche LSR das Netz daran hindern, sichere Entscheidungen über Klassen zu treffen, die möglicherweise falsch gelabelt wurden. Im Vergleich dazu ist die Differenz zwischen den Methoden mit und ohne LSR geringer: U-Ones-LSR ist um 0,00236 und U-Zeros-LSR um 0,00056 größer verglichen mit dem Training ohne LSR (DenseNet121). Dabei ist bei dem Vergleich mit [27] zu beachten, dass durch die erneute Aufteilung des CheXpert Datensatzes im Verhältnis 80:10:10 sich der Anteil an unsicheren Labels unterscheidet. Da für das Validierungsset und das Testset lediglich Patienten ohne unsichere Labels verwendet wurden, enthält das erstellte Trainingsset vergleichsweise

mehr unsichere Labels. Das ist in der Tabelle 4 zu sehen, nach der 72.741 Bilder unsichere Labels enthalten, was einem Anteil von ca. 42,3% des Trainingssets entspricht. Dagegen enthalten 85.056 von den 223.414 Aufnahmen des ursprünglichen Trainingssets von CheXpert unsichere Labels, was einem Anteil von ca. 37,2% entspricht. Die Differenz im Anteil der unsicheren Labels könnte eine Ursache für den Unterschied zu den Ergebnissen von [27] sein. Hier bietet es sich an, den Einfluss des Anteils von unsicheren Labels im Trainingsset in zukünftigen Arbeiten zu untersuchen. In dem Vergleich zu [27] zeigen sich außerdem Unterschiede bei den Ergebnissen einzelner Pathologien (siehe Tabelle 11): Während der AUC von "Cardiomegaly" im Durchschnitt bei 0,890292 (DenseNet121) bzw. 0,884468 (EfficientNetB4) liegt, ist dieser bei Pham 0,8074. Bei "Edema" hingegen erzielt Pham mit 0,9202 einen durchschnittlich höheren AUC verglichen mit 0,883186 bei DenseNet121 und 0,882732 bei EfficientNetB4. In den Experimenten dieser Arbeit zeigt sich insgesamt eine kleine Verbesserung durch LSR bei der Evaluation auf dem Validierungsset, ebenso bei Pham et al. [27]. Diese ist im Verhältnis jedoch geringer. Bei der Evaluation auf dem Testset erhält U-Zeros bei den folgenden Netzen jedoch höhere AUC-Werte im Durchschnitt: Sowohl bei flachem Training mit, als auch ohne Transfer Learning (Tabellen 10 und 12) mit DenseNet121 und bei der Verwendung von Early Stopping (Tabellen 6) und EfficientNetB4.

U-Random. Die neue Methode U-Random wurde weder in der Arbeit von Pham et al. [27] noch von Irvin et al. [16] untersucht, weshalb kein Vergleich zu deren Ergebnissen möglich ist. Dennoch erzielt U-Random durchschnittlich gute Ergebnisse verglichen mit den anderen Methoden: In der Evaluation auf dem Testset erhält U-Random, sowohl bei den Modellen mit Early Stopping (siehe Tabelle 6), als auch bei flachem Training über 5 Epochen (siehe Tabelle 10) und Transfer Learning (siehe Tabelle 12) bessere AUC-Werte als U-Ignore, U-Ones und U-Ones-LSR. Dies zeigt sich ebenso bei der Evaluation auf dem Validierungsset (Tabellen 7, 11 und 13), mit der Ausnahme von U-Ones und U-Ones-LSR bei der Verwendung von DenseNet121 (siehe Tabelle 7). Bei einzelnen Modellen mit EfficientNetB4 erreicht U-Random den höchsten durchschnittlichen AUC-Wert: Dem Training über 5 Epochen, evaluiert auf dem Testset und dem Validierungsset (Tabellen 10 und 11) und dem Training mit Early Stopping, evaliert auf dem Validierungsset (Tabelle 7). Der Vorteil dieser Label Smoothing Methode könnte es sein, dass Modelle bei unsicheren Labels nicht in eine Richtung (diese als positiv oder negativ zu betrachten) beeinflusst wird, aber gleichzeitig alle Daten für das Training genutzt werden können (verglichen mit U-Ignore). Für das Modell scheint es besser zu sein, diese unsicheren Labels zu betrachten, als sie zu ignorieren, auch wenn diese durch zufällig gewählte Werte unscharf sind. Ein zusätzlicher Vorteil ist, dass kein Hyperparametertuning nötig ist, um die Bereiche für unsichere Labels zu wählen. Für U-Ones-LSR und U-Zeros-LSR müssen zunächst Wertebereiche gewählt werden, die für das Ersetzen der unsicheren Labels genutzt werden. Dies ist bei U-Random nicht nötig, da unsichere Labels durch zufällige Werte zwischen 0 und 1 ersetzt werden. U-Random

zeigt eine Stabilität, da diese Methode gute Ergebnisse bei flachem und konditionalem Training, bei DenseNet121 und EfficientNetB4 sowie bei der Evaluation auf dem Test- und dem Validierungsset hervorbringt. Daher sollte U-Random und dessen Performance bei anderen Datensätzen in zukünftigen Arbeiten weiter untersucht werden.

Insgesamt zeigt sich eine Tendenz in den durchschnittlichen Ergebnissen der Methoden des flachen Trainings. Darin erhalten U-Ones und U-Ignore in jeder Evaluation, mit Ausnahme der Tabelle 7, die niedrigsten AUC-Werte, während in dem Großteil der Ergebnisse U-Ones-LSR höhere Werte erzielt. U-Zeros, U-Zeros-LSR und U-Random erlangen tendenziell die besten Ergebnisse.

Konditionales Training. Die Evaluation des konditionalen Trainings auf dem Validierungsset von CheXpert ist in der Tabelle 9 dargestellt. Zusätzlich sind darin jeweils die Werte aus dem Paper von Pham et al. [27] enthalten. Darin ist der beste AUC-Wert der drei Trainingsmethoden hervorgehoben. Betrachtet man das konditionale Training, zeigen die Ergebnisse eine konsistente Verschlechterung im Vergleich zu flachem Training. Die Tabelle 8 stellt dies, sowohl für DenseNet121, als auch für EfficientNetB4 bei allen Label Smoothing Methoden dar. In dem Paper von Pham et al. [27] werden Ergebnisse vorgestellt, in welchen das konditionale Training im Durchschnitt bei allen Labeling Methoden (ohne U-Random) einen höheren AUC-Wert erzielt als das flache Training. Zum Vergleich wurde dabei flaches Training über 5 Epochen ohne Transfer Learning herangezogen, während das konditionale Training insgesamt 10 Epochen trainiert wurde. Dagegen zeigt das konditionale Training in den Tabellen 8 und 9, sowohl bei der Evaluation auf dem Testset als auch auf dem Validierungsset, schlechtere Ergebnisse als das flache Training. Dabei werden, als vergleichbares flaches Training, zusätzlich die Werte von Modellen mit Transfer Learning angegeben. Dadurch unterscheiden sich das flache und das konditionale Training lediglich in den verwendeten konditionalen Daten in den ersten 5 Epochen und um die Berechnung der Wahrscheinlichkeiten aus den bedingten Wahrscheinlichkeiten (siehe Kapitel 4.4). Die Anzahl der trainierten Epochen, das "Einfrieren" der Layer und das Transfer Learning ist bei beiden Methoden gleich. Jedoch zeigt sich bei beiden Vergleichen, mit flachem Training mit und ohne Transfer Learning, eine Verschlechterung durch konditionales Training. Dies spricht dafür, dass der Einfluss der wegfallenden nicht-konditionalen Daten größer sein muss, als der Einfluss des konditionalen Trainings. Dieser wird in dem Paper von Pham et al. [27] wie folgt beschrieben: Laut ihm haben Klassen, die in der Hierarchie niedriger sind, weniger positive Vorkommen, weshalb flache Modelle mehr dazu neigen, diese als negativ zu labeln. Durch das konditionale Training soll das Netz die Beziehungen zwischen Eltern- und Kind-Labels lernen und dazu führen, dass niedrigere Klassen besser eingeordnet werden können. Das anschließende Transfer Learning soll dafür sorgen, dass auch die Eltern-Labels richtig klassifiziert werden können. Durch die Begrenzung auf konditionale Daten, fallen jedoch bei U-Ignore 36.349, bei U-Ones 73.705, bei U-Ones-LSR 53.245, bei U-Zeros 55.304, bei U-Zeros-LSR 53.270 und bei

U-Random 53.403 Röntgen-Aufnahmen weg. Diese haben keinen Einfluss auf die Convolution Layer der Modelle, sondern lediglich bei Transfer Learning auf die letzte fully-connected Schicht. Dabei ist zu beachten, dass die Bedingung für das Aussortieren für das konditionale Training ein positives Kind-Label mit dem Wert 1 und ein negatives Eltern-Label mit dem Wert 0 war. Das ist insbesondere für die Label Smoothing Regularization bei den Methoden U-Ones-LSR, U-Zeros-LSR und U-Random wichtig, da unsichere Labels Werte zwischen 0 und 1 haben. Beispielsweise wird eine Röntgen-Aufnahme mit "Atelectasis"=0.8 und dessen Eltern-Label "Lung Opacity"=0 für das konditionale Training von U-Ones-LSR genutzt. Dieses Bild wird bei U-Ones durch das negative Eltern-Label aussortiert. Denkbar wäre, bei der Bedingung für das Aussortieren mit Abstufungen zu arbeiten. Indem beispielsweise ein Label mit einem Wert >=0.75 als positiv angesehen wird. Dies hätte zur Folge, dass die Röntgen-Aufnahme aussortiert wird, was der Idee des konditionalen Trainings entsprechen würde. Dies sollte in zukünftigen Arbeiten untersucht werden, allerdings würde sich die Anzahl der Bilder im Trainingsset dadurch wesentlich verringern und eine Verschlechterung der Ergebnisse wäre wahrscheinlich. Insgesamt zeigen die Experimente eine durchschnittliche Verschlechterung der Ergebnisse durch konditionales Training, diese ist jedoch je nach betrachteter Pathologie unterschiedlich. Verglichen mit flachem Training mit Transfer Learning (Tabelle 12), verbessert das konditionale Training das Eltern-Label "Enlarged Cardiomediastinum" durchschnittlich um 0,001638 mit EfficientNetB4 bzw. verschlechtert sich um 0,000055 mit DenseNet121. Außerdem verschlechtert sich "Lung Opacitiy" um 0,005812 mit DenseNet121 und um 0,006368 mit EfficientNetB4. Diese Differenz ist bei Kind-Labels, wie beispielsweise "Cardiomegaly" oder "Edema", größer. Der durchschnittliche AUC des konditionalen Trainings von "Cardiomegaly" ist um 0,065167 (DenseNet121) bzw. 0,066422 (EfficientNetB4) geringer, während sich die Werte für "Edema" um 0,016232 (DenseNet121) bzw. 0,018278 (EfficientNetB4) verschlechtern. Dies kann jedoch durch die Änderung der Verteilung im Trainingsset begründet sein, die durch die Begrenzung auf konditionale Daten entsteht (siehe Tabelle 4 und 5). Beispielsweise fallen bei U-Zeros für "Enlarged Cardiomediastinum" 1.613 (entspricht ca. 19,6%) positive Röntgen-Aufnahmen und für "Lung Opacity" 12.708 positive Bilder (entspricht ca. 14,3%) weg. Bei dem Kind-Label "Cardiomegaly" sind das 20.401 (entspricht ca. 94,6%) Bilder bzw. 26.803 (entspricht ca. 58,3%) Stück bei "Edema". In der Arbeit von Allaouzi und Ahmed [1] wird der Datensatz ebenfalls aufgeteilt und die gleichen Methoden der Datenaugmentierung verwendet. Darin erzielt der Ansatz "Binary Relevance", welcher Labels unabhängig voneinander behandelt, bessere Ergebnisse als solche, die Beziehungen zwischen den Labels beachten. Dieses Bild zeigt sich auch in dieser Arbeit, weshalb die Aufteilung des Datensatzes als Grund für die Unterschiede zu Pham et al. [27] nahe liegt.

Tabelle 9: Vergleich von flachem und konditionalem Training mit den Ergebnissen von Pham et al. [27] durch Evaluation auf dem Validierungsset von CheXpert: Durchschnitt der AUC-Werte der 5 Labels

	Flach	Flach mit TL	Konditional
U-Ignore [Pham]	0.8634	-	**0.8718**
U-Ignore [DenseNet]	0.848804	**0.849051**	0.816046
U-Ignore [EfficientNet]	**0.847108**	0.847103	0.813527
U-Ones [Pham]	0.86	-	**0.8718**
U-Ones [DenseNet]	0.847798	**0.848049**	0.823671
U-Ones [EfficientNet]	**0.847903**	0.847881	0.820969
U-Ones-LSR [Pham]	0.8744	-	**0.894**
U-Ones-LSR [DenseNet]	0.850163	**0.850353**	0.83196
U-Ones-LSR [EfficientNet]	0.850512	**0.850579**	0.831758
U-Zeros [Pham]	0.8582	-	**0.8766**
U-Zeros [DenseNet]	0.853349	**0.853681**	0.826321
U-Zeros [EfficientNet]	0.851785	**0.852101**	0.822787
U-Zeros-LSR [Pham]	0.8714	-	**0.8844**
U-Zeros-LSR [DenseNet]	0.853912	**0.854284**	0.832782
U-Zeros-LSR [EfficientNet]	0.852461	**0.852822**	0.830444
U-Random [Pham]	-	-	-
U-Random [DenseNet]	0.852873	**0.853073**	0.831512
U-Random [EfficientNet]	0.852531	**0.852578**	0.829951

EfficientNetB4 und DenseNet121. Die Ergebnisse von EfficientNetB4 und DenseNet121 unterscheiden sich in flachem und konditionalem Training: Die Evaluation auf dem Testset zeigt durchschnittlich eine Verbesserung durch die Verwendung von EfficientNetB4. Bei flachem Training über 5 Epochen ergibt sich eine Differenz von ca. 0,00193 (siehe Tabelle 10), mit anschließendem Transfer Learning von ca. 0,00185 (siehe Tabelle 12) und bei einer Verwendung von Early Stopping von ca. 0,00284. Dagegen ist bei der Evaluation auf dem Validierungsset keine klare Tendenz erkennbar, ebenso bei konditional trainierten Netzen, die auf dem Testset evaluiert wurden. Im Gegensatz dazu erhält DenseNet121 bei der Evaluation von konditionalem Training auf dem Validierungsset bessere Ergebnisse. Diese sind im Durchschnitt um 0,00214 größer als bei EfficientNetB4 (siehe Tabelle 15). Laut den Ergebnissen der Experimente scheint EfficientNetB4 besser für flaches Training geeignet zu sein. Jedoch ist auch hier der Einfluss des verwendeten Sets für das Testen zu beachten, welcher weiter untersucht werden sollte.

7 Ausblick und Fazit

In dieser Arbeit wurden verschiedene Ansätze betrachtet, um mit unsicheren Labels in Datensätzen umzugehen und konditionales Training untersucht, um Röntgen-Thorax-Aufnahmen zu klassifizieren. Diese wurden in einer unabhängigen Studie mit dem CheXpert Datensatz evaluiert. Zunächst wurden die Auswirkungen untersucht unsichere Labels nicht für das Training zu nutzen, während außerdem der Einfluss von Label Smoothing Regularization betrachtet wurde. Schließlich wurde eine neue Label Smoothing Methode U-Random beschrieben und mit den bisherigen Methoden verglichen. In einem weiteren Experiment wurde ein Vergleich von konditionalem Training mit vergleichbarem flachem Training erbracht. Schließlich wurden die Ergebnisse dieser Arbeit mit den Erkenntnissen aus dem Paper von Pham et al. [27] verglichen. Dessen Schlussfolgerung, dass unsichere Labels nicht ignoriert werden sollten, hat sich in den Ergebnissen dieser Arbeit bestätigt. Die zusätzliche Label Smoothing Regularization hat in den Experimenten Verbesserungen im Vergleich zu strikten Labels erbracht. Die neue Label Smoothing Methode U-Random erzielt Ergebnisse in der Größenordnung der anderen LSR-Methoden. Bei einigen Modellen erlangt diese Methode durchschnittlich den höchsten AUC, unabhängig von de verwendeten Daten in der Evaluation (Test- oder Validierungsset) und dem angewandtem Training (flach oder konditional). Diese erhält beispielsweise den höchsten AUC von 0,852531 in der Evaluation von flachem Training über 5 Epochen von EfficientNetB4 mit dem Validierungsset, ebenso wie den größten AUC-Wert von 0,801729 in der Evaluation von konditionalem Training von DenseNet121 mit dem Testset. Abschließend zeigen die Ergebnisse dieser Arbeit eine bessere Performance von flachem Training im Vergleich zu konditionalem Training. Letzteres erlangt im Durchschnitt über alle, im ersten Teil der Arbeit evaluierten Methoden, einen AUC von 0,7954, während vergleichbares flaches Training einen AUC von 0,8194 bei der Evaluation auf dem Testset erzielt. Außerdem ist ein Einfluss des genutzten Netzes im Experiment auf die Ergebnisse zu erkennen. Durch die Verwendung von EfficientNetB4 erreichen Modelle, die flach trainiert werden, durchschnittlich höhere Ergebnissen als DenseNet121. In zukünftigen Studien sollten die Auswirkungen von Aufteilungen bei Datensätzen bestimmt werden. Insbesondere dessen Einfluss auf konditionales Training, ebenso wie die Wahl des Thresholds für die Bestimmung der konditionalen Daten. Zusätzlich sollte die Label Smoothing Methode U-Random genauer untersucht werden und deren Performance bei anderen Datensätzen evaluiert werden.

Literaturverzeichnis

[1] I. Allaouzi and M. Ben Ahmed. A Novel Approach for Multi-Label Chest X-Ray Classification of Common Thorax Diseases. *IEEE Access*, 7:64279–64288, 2019. https://doi.org/10.1109/ACCESS.2019.2916849.

[2] R. Aly, S. Remus, and C. Biemann. Hierarchical Multi-label Classification of Text with Capsule Networks. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop*, pages 323–330, Florence, Italy, July 2019. Association for Computational Linguistics. https://www.aclweb.org/anthology/P19-2045.

[3] L. Bottou. *Neural Networks: Tricks of the Trade: Second Edition*, chapter Stochastic Gradient Descent Tricks, pages 421–436. Springer Berlin Heidelberg, Berlin, Heidelberg, 2012. ISBN: 978-3-642-35289-8, https://doi.org/10.1007/978-3-642-35289-8_25.

[4] Bundesamt für Strahlenschutz. Röntgendiagnostik: Häufigkeit und Strahlenexposition. https://www.bfs.de/DE/themen/ion/anwendung-medizin/diagnostik/roentgen/haeufigkeit-exposition.html.

[5] R. Cerri, R. Barros, A. Carvalho, and Y. Jin. Reduction Strategies for Hierarchical Multi-Label Classification in Protein Function Prediction. *BMC Bioinformatics*, 17, 09 2016. https://doi.org/10.1186/s12859-016-1232-1.

[6] R. Cerri, R. C. Barros, and A. C. de Carvalho. Hierarchical multi-label classification using local neural networks. *Journal of Computer and System Sciences*, 80(1):39–56, 2014. https://doi.org/10.1016/j.jcss.2013.03.007.

[7] H. Chen, S. Miao, D. Xu, G. D. Hager, and A. P. Harrison. Deep Hierarchical Multi-label Classification of Chest X-ray Images. In *Proceedings of The 2nd International Conference on Medical Imaging with Deep Learning*, volume 102 of *Proceedings of Machine Learning Research*, pages 109–120. PMLR, 2019. http://proceedings.mlr.press/v102/chen19a.html.

[8] J. Chorowski and N. Jaitly. Towards better decoding and language model integration in sequence to sequence models, 2016. arXiv:1612.02695.

[9] J. Davis and M. Goadrich. The Relationship between Precision-Recall and ROC Curves. In *Proceedings of the 23rd International Conference on Machine Learning*, ICML '06, page 233–240. Association for Computing Machinery, 2006. ISBN: 1595933832, https://doi.org/10.1145/1143844.1143874.

[10] L. Delrue, R. Gosselin, B. Ilsen, A. Landeghem, J. De Mey, and p. duyck. *Comparative Interpretation of CT and Standard Radiography of the Chest*, chapter Difficulties in the Interpretation of Chest Radiography, pages 27–49. 09 2011. ISBN: 978-3-540-79941-2.

[11] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255, 2009. https://doi.org/10.1109/CVPR.2009.5206848.

[12] J. K. Gohagan, P. C. Prorok, R. B. Hayes, and B.-S. Kramer. The Prostate, Lung, Colorectal and Ovarian (PLCO) Cancer Screening Trial of the National Cancer Institute: History, organization, and status. *Controlled Clinical Trials*, 21(6, Supplement 1):251S–272S, 2000. https://doi.org/10.1016/S0197-2456(00)00097-0.

[13] I. Goodfellow, Y. Bengio, and A. Courville. *Deep Learning*. MIT Press, 2016. http://www.deeplearningbook.org.

[14] G. Huang, Z. Liu, and K. Q. Weinberger. Densely Connected Convolutional Networks. *CoRR*, abs/1608.06993, 2016.

[15] J. Huang and C. Ling. Using AUC and accuracy in evaluating learning algorithms. *IEEE Transactions on Knowledge and Data Engineering*, 17(3):299–310, 2005. https://doi.org/10.1109/TKDE.2005.50.

[16] J. Irvin, P. Rajpurkar, M. Ko, Y. Yu, S. Ciurea-Ilcus, C. Chute, H. Marklund, B. Haghgoo, R. Ball, K. Shpanskaya, J. Seekins, D. A. Mong, S. S. Halabi, J. K. Sandberg, R. Jones, D. B. Larson, C. P. Langlotz, B. N. Patel, M. P. Lungren, and A. Y. Ng. CheXpert: A large chest radiograph dataset with uncertainty labels and expert comparison, 2019. https://doi.org/10.1609/aaai.v33i01.3301590.

[17] A. E. W. Johnson, T. J. Pollard, S. J. Berkowitz, N. R. Greenbaum, M. P. Lungren, C. Deng, R. G. Mark, and S. Horng. MIMIC-CXR: A large publicly available database of labeled chest radiographs. *CoRR*, abs/1901.07042, 2019.

[18] P. Kim. *MATLAB Deep Learning: With Machine Learning, Neural Networks and Artificial Intelligence*, chapter Convolutional Neural Network, pages 121–147. Apress, Berkeley, CA, 2017. ISBN: 978-1-4842-2845-6, https://doi.org/10.1007/978-1-4842-2845-6_6,.

[19] D. P. Kingma and J. Ba. Adam: A Method for Stochastic Optimization, 2017. arXiv:1412.6980.

[20] J. Loy. *Neural Network Projects with Python*. Packt Publishing, 2019. ISBN: 9781789138900.

[21] A. Mayne and R. Perry. Hierarchically classifying documents with multiple labels. In *2009 IEEE Symposium on Computational Intelligence and Data Mining*, pages 133–139, 2009. https://doi.org/10.1109/CIDM.2009.4938640.

[22] A. Mikołajczyk and M. Grochowski. Data augmentation for improving deep learning in image classification problem. In *2018 International Interdisciplinary PhD Workshop (IIPhDW)*, pages 117–122, 2018. https://doi.org/10.1109/IIPHDW.2018.8388338.

[23] Mitteldeutsche Gesellschaft für Pneumologie und Thoraxchirurgie. Erkrankungen des Mediastinums. https://www.mdgp.de/wichtige-lungenerkrankungen/erkrankungen-des-mediastinums/ Besucht: 2021-05-08.

[24] K. O'Shea and R. Nash. An Introduction to Convolutional Neural Networks. *CoRR*, abs/1511.08458, 2015.

[25] G. Pereyra, G. Tucker, J. Chorowski, Łukasz Kaiser, and G. Hinton. Regularizing Neural Networks by Penalizing Confident Output Distributions, 2017. arXiv:1701.06548.

[26] P. Perrotta. *Programming Machine Learning.* Pragmatic Bookshelf, 2020. ISBN: 9781680506600.

[27] H. H. Pham, T. T. Le, D. Q. Tran, D. T. Ngo, and H. Q. Nguyen. Interpreting chest X-rays via CNNs that exploit hierarchical disease dependencies and uncertainty labels. 6 2019.

[28] P. Rajpurkar, J. Irvin, R. L. Ball, K. Zhu, B. Yang, H. Mehta, T. Duan, D. Ding, A. Bagul, C. P. Langlotz, B. N. Patel, K. W. Yeom, K. Shpanskaya, F. G. Blankenberg, J. Seekins, T. J. Amrhein, D. A. Mong, S. S. Halabi, E. J. Zucker, A. Y. Ng, and M. P. Lungren. Deep learning for chest radiograph diagnosis: A retrospective comparison of the CheXNeXt algorithm to practicing radiologists. *PLOS Medicine*, 15(11):1–17, 11 2018. https://doi.org/10.1371/journal.pmed.1002686.

[29] P. Rajpurkar, J. Irvin, K. Zhu, B. Yang, H. Mehta, T. Duan, D. Ding, A. Bagul, C. Langlotz, K. Shpanskaya, M. P. Lungren, and A. Y. Ng. CheXNet: Radiologist-Level Pneumonia Detection on Chest X-Rays with Deep Learning, 2017. arXiv:1711.05225.

[30] B. Ramsundar, P. Eastman, P. Walters, V. Pande, and H. Schock. *Deep Learning für die Biowissenschaften : Einsatz von Deep Learning in Genomik, Biophysik, Mikroskopie und medizinischer Analyse.* O'Reilly, Heidelberg, 2020. ISBN: 978-3-96009-130-1.

[31] J. Rousu, C. Saunders, S. Szedmak, and J. Shawe-Taylor. Kernel-Based Learning of Hierarchical Multilabel Classification Models. *Journal of Machine Learning Research*, 7:1601–1626, 12 2006.

[32] R. Schwaiger and J. Steinwendner. *Neuronale Netze programmieren mit Python.* Rheinwerk Verlag, Bonn, 2019. ISBN: 978-3-8362-6142-5.

[33] S. Sharma, S. Sharma, and A. Athaiya. ACTIVATION FUNCTIONS IN NEURAL NETWORKS. *International Journal of Engineering Applied Sciences and Technology*, 4:310–316, 2020.

[34] V. Silaparasetty. *Deep Learning Projects Using Tensorflow 2, Neural Network Development with Python and Keras.* Apres, Berkeley, CA, 2020. https://doi.org/10.1007/978-1-4842-5802-6.

[35] A. Sze-To and Z. Wang. tCheXNet: Detecting Pneumothorax on Chest X-Ray Images Using Deep Transfer Learning. In F. Karray, A. Campilho, and A. Yu, editors, *Image Analysis and Recognition*, pages 325–332, Cham, 2019. Springer International Publishing. ISBN: 978-3-030-27272-2.

[36] M. Tan and Q. V. Le. EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks, Sept. 2020.

[37] X. Wang, Y. Peng, L. Lu, Z. Lu, M. Bagheri, and R. M. Summers. ChestX-Ray8: Hospital-Scale Chest X-Ray Database and Benchmarks on Weakly-Supervised

Classification and Localization of Common Thorax Diseases. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3462–3471, 2017. https://doi.org/10.1109/CVPR.2017.369.

[38] K. Weiss, T. M. Khoshgoftaar, and D. Wang. A survey of transfer learning. *Journal of Big Data*, 3(9), 2016. https://doi.org/10.1186/s40537-016-0043-6.

[39] M. Wenzel, F. Chun, O. Hinz, and B. Abdel-Karim. Möglichkeiten einer automatisierten Auswertung der Thorax-Röntgenaufnahme durch künstliche Intelligenz für Klinik und Praxis. *Pneumologe*, 17:59–64, 2020.

[40] World Health Organization. Diagnostic imaging. https://www.who.int/diagnostic_imaging/imaging_modalities/dim_plain-radiography/en/# Besucht: 2021-05-12.

[41] World Health Organization. Pneumonia. https://www.who.int/health-topics/pneumonia#tab=tab_1 Besucht: 2021-05-12.

[42] R. Yamashita, M. Nishio, R. Do, and K. Togashi. Convolutional neural networks: an overview and application in radiology. *Insights into Imaging*, 9, 06 2018. https://doi.org/10.1007/s13244-018-0639-9.

Anhang

Tabelle 10: AUC-Werte der einzelnen Labels der Evaluation des flachen Trainings ohne Transfer Learning auf dem Testset

	Enlarged Cardiomediastinum		Cardiomegaly	
Methode	DenseNet	EfficientNet	DenseNet	EfficientNet
U-Ignore	0.717492	0.71041	0.888374	0.881273
U-Ones	0.70261	0.713319	0.88842	0.885261
U-Ones-LSR	0.71609	0.707684	0.892265	0.883814
U-Zeros	0.716697	0.714311	0.892518	0.885893
U-Zeros-LSR	0.71325	0.724688	0.888956	0.883875
U-Random	0.71816	0.721369	0.891216	0.886696
	Lung Opacity		Lung Lesion	
Methode	DenseNet	EfficientNet	DenseNet	EfficientNet
U-Ignore	0.778892	0.779236	0.747417	0.749285
U-Ones	0.776476	0.782408	0.770355	0.770496
U-Ones-LSR	0.776852	0.781916	0.767815	0.764417
U-Zeros	0.780038	0.780914	0.772032	0.765298
U-Zeros-LSR	0.778265	0.78028	0.770087	0.763054
U-Random	0.779089	0.781031	0.771212	0.769308
	Edema		Consolidation	
Methode	DenseNet	EfficientNet	DenseNet	EfficientNet
U-Ignore	0.880247	0.879624	0.80913	0.804655
U-Ones	0.884921	0.884789	0.803335	0.803448
U-Ones-LSR	0.885514	0.885546	0.804631	0.811081
U-Zeros	0.881736	0.879791	0.821152	0.818611
U-Zeros-LSR	0.882816	0.882536	0.824041	0.816669
U-Random	0.883884	0.884103	0.813373	0.815373
	Pneumonia		Atelectasis	
Methode	DenseNet	EfficientNet	DenseNet	EfficientNet
U-Ignore	0.780989	0.776878	0.748567	0.753839
U-Ones	0.777597	0.790042	0.744659	0.747774
U-Ones-LSR	0.786817	0.783623	0.748411	0.75261
U-Zeros	0.792615	0.793015	0.75138	0.754769
U-Zeros-LSR	0.783152	0.793921	0.753638	0.758997
U-Random	0.783072	0.790873	0.755335	0.75633
	Pneumothorax		Pleural Effusion	
Methode	DenseNet	EfficientNet	DenseNet	EfficientNet
U-Ignore	0.854605	0.852346	0.917703	0.916148
U-Ones	0.865056	0.867663	0.917656	0.918244
U-Ones-LSR	0.863284	0.869428	0.919993	0.91951
U-Zeros	0.862184	0.86731	0.919956	0.919861
U-Zeros-LSR	0.863127	0.867691	0.920108	0.920229
U-Random	0.863831	0.868393	0.920557	0.920152

	Pleural Other		Fracture	
Methode	DenseNet	EfficientNet	DenseNet	EfficientNet
U-Ignore	0.772217	0.797838	0.768876	0.776234
U-Ones	0.801663	0.802538	0.778312	0.785311
U-Ones-LSR	0.80729	0.809149	0.791399	0.791565
U-Zeros	0.802286	0.809948	0.781529	0.791325
U-Zeros-LSR	0.796414	0.824435	0.784051	0.785625
U-Random	0.803714	0.824291	0.782526	0.787158
	Support Devices		Durchschnitt	
Methode	DenseNet	EfficientNet	DenseNet	EfficientNet
U-Ignore	0.873229	0.871113	0.810595	0.811452
U-Ones	0.881914	0.883821	0.814844	0.818086
U-Ones-LSR	0.884817	0.886707	0.81886	0.819004
U-Zeros	0.884002	0.885551	0.819856	0.820507
U-Zeros-LSR	0.883906	0.886773	0.818601	0.822213
U-Random	0.884589	0.885507	0.819274	0.822353

Tabelle 11: AUC Werte der 5 Labels der Evaluation des flachen Trainings ohne Transfer Learning auf dem Validierungsset von CheXpert mit den Ergebnissen von Pham et al. [27].

	Atelectasis			Cardiomegaly		
Methode	Pham	DenseNet	EfficientNet	Pham	DenseNet	EfficientNet
U-Ignore	0.768	0.748567	0.753839	0.795	0.888374	0.881273
U-Ones	0.8	0.744659	0.747774	0.78	0.88842	0.885261
U-Ones-LSR	0.818	0.748411	0.75261	0.834	0.892265	0.883814
U-Zeros	0.745	0.75138	0.754769	0.813	0.892518	0.885893
U-Zeros-LSR	0.781	0.753638	0.758997	0.815	0.888956	0.883875
U-Random	-	0.755335	0.75633	-	0.891216	0.886696
	Consolidation			Edema		
Methode	Pham	DenseNet	EfficientNet	Pham	DenseNet	EfficientNet
U-Ignore	0.915	0.80913	0.804655	0.914	0.880247	0.879624
U-Ones	0.882	0.803335	0.803448	0.918	0.884921	0.884789
U-Ones-LSR	0.874	0.804631	0.811081	0.925	0.885514	0.885546
U-Zeros	0.882	0.821152	0.818611	0.921	0.881736	0.879791
U-Zeros-LSR	0.92	0.824041	0.816669	0.923	0.882816	0.882536
U-Random	-	0.813373	0.815373	-	0.883884	0.884103
	Pleural Effusion			Durchschnitt		
Methode	Pham	DenseNet	EfficientNet	Pham	DenseNet	EfficientNet
U-Ignore	0.925	0.917703	0.916148	0.8634	0.848804	0.847108
U-Ones	0.92	0.917656	0.918244	0.86	0.847798	0.847903
U-Ones-LSR	0.921	0.919993	0.91951	0.8744	0.850163	0.850512
U-Zeros	0.93	0.919956	0.919861	0.8582	0.853349	0.851785
U-Zeros-LSR	0.918	0.920108	0.920229	0.8714	0.853912	0.852461
U-Random	-	0.920557	0.920152	-	0.852873	0.852531

Tabelle 12: AUC-Werte der einzelnen Labels der Evaluation des flachen Trainings mit Transfer Learning auf dem Testset

	Enlarged Cardiomediastinum		Cardiomegaly	
Methode	DenseNet	EfficientNet	DenseNet	EfficientNet
U-Ignore	0.718286	0.710706	0.888814	0.880971
U-Ones	0.70237	0.713535	0.888734	0.885722
U-Ones-LSR	0.716441	0.70492	0.89254	0.884131
U-Zeros	0.718386	0.718394	0.89284	0.886324
U-Zeros-LSR	0.713866	0.726389	0.889392	0.883981
U-Random	0.717013	0.720975	0.891367	0.886712
	Lung Opacity		Lung Lesion	
Methode	DenseNet	EfficientNet	DenseNet	EfficientNet
U-Ignore	0.779143	0.780416	0.749896	0.75056
U-Ones	0.776084	0.782438	0.770455	0.772209
U-Ones-LSR	0.777451	0.781341	0.768083	0.765703
U-Zeros	0.780058	0.780446	0.772573	0.765268
U-Zeros-LSR	0.778208	0.780405	0.771957	0.763386
U-Random	0.779429	0.781336	0.772019	0.769614
	Edema		Consolidation	
Methode	DenseNet	EfficientNet	DenseNet	EfficientNet
U-Ignore	0.880697	0.880396	0.809138	0.803508
U-Ones	0.88486	0.884959	0.804389	0.802645
U-Ones-LSR	0.886057	0.885748	0.805128	0.810979
U-Zeros	0.88255	0.88073	0.821222	0.819242
U-Zeros-LSR	0.883631	0.883065	0.823983	0.817398
U-Random	0.884174	0.884428	0.813291	0.81485
	Pneumonia		Atelectasis	
Methode	DenseNet	EfficientNet	DenseNet	EfficientNet
U-Ignore	0.783363	0.77891	0.748818	0.754633
U-Ones	0.779143	0.790699	0.744313	0.747986
U-Ones-LSR	0.785977	0.786864	0.74802	0.752605
U-Zeros	0.793997	0.793998	0.751795	0.754406
U-Zeros-LSR	0.784574	0.796574	0.754221	0.759342
U-Random	0.784572	0.790374	0.756089	0.75682
	Pneumothorax		Pleural Effusion	
Methode	DenseNet	EfficientNet	DenseNet	EfficientNet
U-Ignore	0.854574	0.853374	0.917791	0.916005
U-Ones	0.864706	0.867404	0.917951	0.918094
U-Ones-LSR	0.863028	0.8693	0.920022	0.919431
U-Zeros	0.862852	0.868029	0.919997	0.919806
U-Zeros-LSR	0.863176	0.86802	0.920195	0.920325
U-Random	0.864121	0.868823	0.920445	0.920082

	Pleural Other		Fracture	
Methode	DenseNet	EfficientNet	DenseNet	EfficientNet
U-Ignore	0.775391	0.797239	0.770718	0.777711
U-Ones	0.800496	0.800594	0.78065	0.786359
U-Ones-LSR	0.809559	0.810802	0.792086	0.793305
U-Zeros	0.801518	0.81177	0.783051	0.792559
U-Zeros-LSR	0.798038	0.826389	0.785819	0.788085
U-Random	0.805743	0.822561	0.783768	0.78937
	Support Devices		Durchschnitt	
Methode	DenseNet	EfficientNet	DenseNet	EfficientNet
U-Ignore	0.873388	0.871516	0.81154	0.811996
U-Ones	0.882472	0.884165	0.815125	0.818216
U-Ones-LSR	0.885025	0.886841	0.819186	0.819382
U-Zeros	0.884227	0.885672	0.82039	0.82128
U-Zeros-LSR	0.884163	0.887013	0.819325	0.823106
U-Random	0.884918	0.885851	0.819765	0.822446

Tabelle 13: AUC Werte der 5 Labels der Evaluation des flachen Trainings mit Transfer Learning auf dem Validierungsset von CheXpert

	Atelectasis			Cardiomegaly		
Methode	Pham	DenseNet	EfficientNet	Pham	DenseNet	EfficientNet
U-Ignore	-	0.748818	0.754633	-	0.888814	0.880971
U-Ones	-	0.744313	0.747986	-	0.888734	0.885722
U-Ones-LSR	-	0.74802	0.752605	-	0.89254	0.884131
U-Zeros	-	0.751795	0.754406	-	0.89284	0.886324
U-Zeros-LSR	-	0.754221	0.759342	-	0.889392	0.883981
U-Random	-	0.756089	0.75682	-	0.891367	0.886712
	Consolidation			Edema		
Methode	Pham	DenseNet	EfficientNet	Pham	DenseNet	EfficientNet
U-Ignore	-	0.809138	0.803508	-	0.880697	0.880396
U-Ones	-	0.804389	0.802645	-	0.88486	0.884959
U-Ones-LSR	-	0.805128	0.810979	-	0.886057	0.885748
U-Zeros	-	0.821222	0.819242	-	0.88255	0.88073
U-Zeros-LSR	-	0.823983	0.817398	-	0.883631	0.883065
U-Random	-	0.813291	0.81485	-	0.884174	0.884428
	Pleural Effusion			Durchschnitt		
Methode	Pham	DenseNet	EfficientNet	Pham	DenseNet	EfficientNet
U-Ignore	-	0.917791	0.916005	-	0.849051	0.847103
U-Ones	-	0.917951	0.918094	-	0.848049	0.847881
U-Ones-LSR	-	0.920022	0.919431	-	0.850353	0.850579
U-Zeros	-	0.919997	0.919806	-	0.853681	0.852101
U-Zeros-LSR	-	0.920195	0.920325	-	0.854284	0.852822
U-Random	-	0.920445	0.920082	-	0.853073	0.852578

Tabelle 14: AUC-Werte der einzelnen Labels der Evaluation des konditionalen Trainings auf dem Testset

	Enlarged Cardiomediastinum		Cardiomegaly	
Methode	DenseNet	EfficientNet	DenseNet	EfficientNet
U-Ignore	0.716967	0.716653	0.796538	0.7889
U-Ones	0.700467	0.716154	0.821723	0.812835
U-Ones-LSR	0.716134	0.71388	0.844764	0.842849
U-Zeros	0.719209	0.718284	0.809884	0.801257
U-Zeros-LSR	0.71614	0.72441	0.83601	0.824438
U-Random	0.717116	0.715361	0.843766	0.839034
	Lung Opacity		Lung Lesion	
Methode	DenseNet	EfficientNet	DenseNet	EfficientNet
U-Ignore	0.771997	0.77323	0.68622	0.675652
U-Ones	0.770042	0.771699	0.724503	0.70952
U-Ones-LSR	0.772843	0.776558	0.724325	0.720443
U-Zeros	0.772154	0.773403	0.718269	0.712508
U-Zeros-LSR	0.772981	0.777819	0.714511	0.714263
U-Random	0.775483	0.775462	0.723716	0.7178
	Edema		Consolidation	
Methode	DenseNet	EfficientNet	DenseNet	EfficientNet
U-Ignore	0.858123	0.854856	0.796055	0.794793
U-Ones	0.867965	0.867472	0.793008	0.791916
U-Ones-LSR	0.871438	0.870035	0.798362	0.802741
U-Zeros	0.86702	0.862673	0.80406	0.798307
U-Zeros-LSR	0.869188	0.866601	0.805391	0.807308
U-Random	0.870845	0.868021	0.800509	0.799555
	Pneumonia		Atelectasis	
Methode	DenseNet	EfficientNet	DenseNet	EfficientNet
U-Ignore	0.708825	0.69406	0.716916	0.718024
U-Ones	0.710065	0.70737	0.719576	0.718827
U-Ones-LSR	0.716848	0.725872	0.726644	0.726277
U-Zeros	0.731259	0.723001	0.734228	0.73445
U-Zeros-LSR	0.724807	0.725637	0.734127	0.73517
U-Random	0.725837	0.722469	0.72565	0.725885
	Pneumothorax		Pleural Effusion	
Methode	DenseNet	EfficientNet	DenseNet	EfficientNet
U-Ignore	0.845737	0.838252	0.912599	0.911063
U-Ones	0.850931	0.849061	0.916084	0.913795
U-Ones-LSR	0.856348	0.863523	0.918593	0.916889
U-Zeros	0.852844	0.855108	0.916413	0.91725
U-Zeros-LSR	0.850008	0.859884	0.919193	0.918702
U-Random	0.852202	0.85929	0.916792	0.917258

	Pleural Other		Fracture	
Methode	DenseNet	EfficientNet	DenseNet	EfficientNet
U-Ignore	0.793363	0.758727	0.769668	0.758152
U-Ones	0.799911	0.787778	0.768977	0.780168
U-Ones-LSR	0.791228	0.789951	0.774008	0.779186
U-Zeros	0.789013	0.790451	0.781964	0.773757
U-Zeros-LSR	0.802457	0.813005	0.777456	0.787174
U-Random	0.810334	0.80466	0.779589	0.785295
	Support Devices		Durchschnitt	
Methode	DenseNet	EfficientNet	DenseNet	EfficientNet
U-Ignore	0.864954	0.860447	0.787536	0.780216
U-Ones	0.874934	0.873574	0.793706	0.792321
U-Ones-LSR	0.880776	0.881454	0.799409	0.800743
U-Zeros	0.878433	0.878915	0.798058	0.795336
U-Zeros-LSR	0.882296	0.882843	0.800351	0.802866
U-Random	0.880643	0.880357	0.801729	0.800804

Tabelle 15: AUC Werte der 5 Labels der Evaluation des konditionalen Trainings auf dem Validierungsset von CheXpert mit den Ergebnissen von Pham et al. [27].

	Atelectasis			Cardiomegaly		
Methode	Pham	DenseNet	EfficientNet	Pham	DenseNet	EfficientNet
U-Ignore	0.78	0.716916	0.718024	0.815	0.796538	0.7889
U-Ones	0.813	0.719576	0.718827	0.816	0.821723	0.812835
U-Ones-LSR	0.825	0.726644	0.726277	0.855	0.844764	0.842849
U-Zeros	0.782	0.734228	0.73445	0.835	0.809884	0.801257
U-Zeros-LSR	0.806	0.734127	0.73517	0.833	0.83601	0.824438
U-Random	-	0.72565	0.725885	-	0.843766	0.839034
	Consolidation			Edema		
Methode	Pham	DenseNet	EfficientNet	Pham	DenseNet	EfficientNet
U-Ignore	0.922	0.796055	0.794793	0.914	0.858123	0.854856
U-Ones	0.895	0.793008	0.791916	0.923	0.867965	0.867472
U-Ones-LSR	0.937	0.798362	0.802741	0.93	0.871438	0.870035
U-Zeros	0.922	0.80406	0.798307	0.923	0.86702	0.862673
U-Zeros-LSR	0.929	0.805391	0.807308	0.933	0.869188	0.866601
U-Random	-	0.800509	0.799555	-	0.870845	0.868021
	Pleural Effusion			Durchschnitt		
Methode	Pham	DenseNet	EfficientNet	Pham	DenseNet	EfficientNet
U-Ignore	0.928	0.912599	0.911063	0.8718	0.816046	0.813527
U-Ones	0.912	0.916084	0.913795	0.8718	0.823671	0.820969
U-Ones-LSR	0.923	0.918593	0.916889	0.894	0.83196	0.831758
U-Zeros	0.921	0.916413	0.91725	0.8766	0.826321	0.822787
U-Zeros-LSR	0.921	0.919193	0.918702	0.8844	0.832782	0.830444
U-Random	-	0.916792	0.917258	-	0.831512	0.829951

8 Digitale Ausarbeitung und Programmcode

Der Arbeit ist ein USB-Stick mit digitaler Fassung der Arbeit beigelegt. Zudem befinden sich darauf der Programmcode für das Training der Neuronalen Netze und das erstellte Datenset. Weiterer Programmcode steht unter `https://gitlab2.informatik.uni-wuerzburg.de/s364514/bachelorarbeit.git` zur Verfügung. Die Bilder des CheXpert Datensatzes sind öffentlich verfügbar: `https://stanfordmlgroup.github.io/competitions/chexpert/`

Eidesstattliche Erklärung

Ich versichere, die von mir vorgelegte Arbeit selbstständig verfasst zu haben. Alle Stellen, die wörtlich oder sinngemäß aus veröffentlichten oder nicht veröffentlichten Arbeiten anderer entnommen sind, habe ich als entnommen kenntlich gemacht. Sämtliche Quellen und Hilfsmittel, die ich für die Arbeit benutzt habe, sind angegeben. Die Arbeit hat mit gleichem Inhalt bzw. in wesentlichen Teilen noch keiner anderen Prüfungsbehörde vorgelegen.

$Unterschrift$: $Ort, Datum$: