

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

Datenaugmentierung mit Online Bilevel Optimierung zur
Pathologie-Klassifikation in Röntgen-Thorax-Aufnahme

Autor: Fabian Gäullein <fabian.gaeullein@stud-mail.uni-wuerzburg.de>
MatNr. 2256243

Abgabe: 20.12.2021

1. Betreuer: Prof. Dr. Frank Puppe

2. Betreuer: M. Sc. Amar Hekalo

Abstract

Die Motivation dieser Bachelorarbeit war es, eine neue Methode für Datenaugmentierung auf Thorax-Röntgenbildern anzuwenden und zu evaluieren. Dazu wurden vier Experimente durchgeführt, um die Veränderung der Performance eines Classifiers zu sehen, der mit verschiedenen Arten von Augmentierungen trainiert wurde. Zuerst wurden das ResNET18 und das DenseNet121 ohne Augmentierungen auf dem CheXpert Datensatz trainiert. Danach wurden für das Training der beiden Classifier Augmentierungen, die inspiriert durch Related Work auf Röntgenbildern sind, auf dem Datensatz angewendet. Die dritte Methode war die Anwendung des Online Bilevel Algorithmus auf dem ResNET18 und dem DenseNet121 mit CheXpert Datensatz. Als letztes wurden die vortrainierten Augmentierungsnetze aus dem dritten Experiment verwendet, um das ResNET18 zu trainieren. Der Bilevel Augment Algorithmus ist zusammengesetzt aus einem inneren und einem äußeren Loop. Im inneren Loop wird ein Classifier mit den Augmentierungen aus dem Augmenternetz trainiert. Auf dem Classifier wird dann im äußeren Loop der Validationloss berechnet und mit diesem Validationloss wird dann das Augmenternetz trainiert. Die Änderungen durch die vier verschiedenen Augmentierungsmethoden wurden experimentell gezeigt. Der durchschnittliche AUC-Score des ResNET18s verbessert sich Schritt für Schritt mit den verschiedenen Augmentierungsmethoden. Dadurch wurde gezeigt, dass der Bilevel Augment Algorithmus nützlich für die Augmentierung von medizinischen Bildern sein kann.

Inhaltsverzeichnis

Abbildungsverzeichnis	5
Tabellenverzeichnis	6
1 Einleitung	8
2 Funktionsweisen	10
2.1 Neuronale Netze	10
2.1.1 Verlustfunktion	12
2.1.2 Lernrate	12
2.1.3 Backpropagation und Optimizer	13
2.2 Bildklassifikation	13
2.3 Overfitting und Datenaugmentierung	14
2.4 Online Bilevel Optimization	15
2.5 Transfer Learning	16
2.5.1 ResNET	16
2.5.2 DenseNet	17
3 Related Work	18
3.1 Online Bilevel Optimization	18
3.2 Generative Adversarial Networks	18
3.3 AutoAugment	19
3.4 Large-scale Robust Deep AUC Maximization: A New Surrogate Loss and Empirical Studies on Medical Image Classification	20
3.5 CheXpert Competition	21
4 Methoden	22
4.1 Datensatz	22
4.2 Experimente	23
5 Ergebnisse	25
5.1 Metriken	25
5.2 Klassifikation ohne Augmentierungen	26
5.2.1 ResNET18	26
5.2.2 DenseNet	29
5.3 Klassifikation mit Augmentierungen inspiriert durch Related Work . . .	31
5.3.1 ResNET18	31
5.3.2 DensNet	34
5.4 Klassifikation mit Online Bilevel Optimization	36
5.4.1 ResNET18	36
5.4.2 DenseNet	40
5.5 Klassifikation mit dem vortrainierten Augmentierungsnetz aus Experiment 3	44
6 Diskussion	48
6.1 Zusammenfassung	48
6.1.1 ResNET18	48
6.1.2 DensNet	49
6.2 Interpretation	50

6.3	Grenzen der Arbeit	51
6.4	Weiterführende Forschung	51
7	Fazit	53
	Literaturverzeichnis	54
	Anhang	57
8	USB-Stick mit Ausarbeitung und Programmcode	85
	Eidesstattliche Erklärung	85

Abbildungsverzeichnis

1	Ein Thoraxröntgenbild normal und horizontal geflippt[15]	9
2	Eine einfache Darstellung eines künstlichen Neurons(Perzeptron)[27] . .	11
3	Aktivierungsfunktionen[3]	11
4	Einfache Darstellung eines Feed-Forward-Netzes[11]	12
5	Schematische Darstellung eines Convolutional Layers[21]	14
6	Darstellung eines CNNs anhand Facebooks DeepFace Algorithmus[26] .	14
7	Verschiedene Arten von Bildrauschen[24]	15
8	Der Online Bilevel Optimization Algorithmus[25]	16
9	Datenblatt zu ResNET18	17
10	Ein 5-Layer Dense-Block aus einem DenseNet[18]	17
11	Genauigkeit des Online Bilevel Optimization Algorithmus auf verschie- denen Datensätzen[25]	18
12	Das DAGAN Netzwerk[1]	19
13	Die AutoAugment Methode	20
14	Der Loss auf dem Trainingsset (links) und der Validationloss auf dem Validierungsset (rechts) ohne Augmentierungen (ResNET18)	26
15	Trainings- und Validationloss ohne Augmentierungen (DenseNet) . . .	29
16	Der Loss auf dem Trainingsset (links) und der Validationloss auf dem Validierungsset (rechts) mit related Work Augmentierungen	31
17	Trainingsloss und Validationloss mit related work Augmentierungen mit dem DenseNet	34
18	Der Loss und Validationloss von affine_color	36
19	Der Loss und Validationloss von small_affine	37
20	Validation- und Trainingsloss auf dem DenseNet mit affine_color . . .	41
21	Der Trainings- und Validationloss auf dem DenseNet mit small_affine .	41
22	Der Trainloss und Validationloss mit dem vortrainierten small_affine .	45
23	Der Trainloss und Validationloss mit dem vortrainierten affine_color .	45
24	Die ROC für die einzelnen Pathologien ohne Augmentierungen (Res- NET18)	75
25	Die ROC für die einzelnen Pathologien ohne Augmentierungen (DenseNet)	76
26	Die ROC für die einzelnen Pathologien mit Augmentierungen inspiriert durch Related Work (ResNET18)	77
27	Die ROC für die einzelnen Pathologien mit Augmentierungen inspiriert durch Related Work (DenseNet)	78
28	Die ROC für die einzelnen Pathologien mit affine_color (ResNET) . . .	79
29	Die ROC für die einzelnen Pathologien mit small_affine (ResNET) . .	80
30	Die ROC für die einzelnen Pathologien mit affine_color (DenseNet) . .	81
31	Die ROC für die einzelnen Pathologien mit small_affine (DenseNet) . .	82
32	Die ROC für die einzelnen Pathologien mit vortrainiertem small_affine (ResNET)	83
33	Die ROC für die einzelnen Pathologien mit vortrainiertem affine_color (ResNET)	84

Tabellenverzeichnis

1	Überblick der verwendeten Hyperparameter der durchgeführten Experimente.	24
2	Aufbau einer Confusion Matrix	25
3	Precision, Recall und $F_1 - Score$ der einzelnen Pathologien ohne Augmentierungen und Threshold 0.5 (ResNET18)	27
4	Precision, Recall und $F_1 - Score$ der einzelnen Pathologien ohne Augmentierungen und Threshold 0.2 (ResNET18)	28
5	AUC-ROC-Score der einzelnen Pathologien ohne Augmentierungen (ResNET18)	28
6	Precision, Recall und $F_1 - Score$ der einzelnen Pathologien ohne Augmentierungen und Threshold 0.5 (DenseNet)	30
7	Precision, Recall und $F_1 - Score$ der einzelnen Pathologien ohne Augmentierungen und Threshold 0.2 (DenseNet)	30
8	AUC-ROC-Score der einzelnen Pathologien ohne Augmentierungen (DenseNet)	31
9	Precision, Recall und $F_1 - Score$ der einzelnen Pathologien mit related Work Augmentierungen und Threshold 0.5 (ResNET18)	32
10	Precision, Recall und $F_1 - Score$ der einzelnen Pathologien mit related Work Augmentierungen und Threshold 0.2 (ResNET18)	33
11	AUC-ROC-Score der einzelnen Pathologien mit Augmentierungen inspiriert durch Related Work (ResNET18)	33
12	Precision, Recall und $F_1 - Score$ mit Related Work Augmentierungen der einzelnen Pathologien mit Threshold 0.5 (DenseNet)	35
13	Precision, Recall und $F_1 - Score$ mit Related Work Augmentierungen der einzelnen Pathologien mit Threshold 0.2 (DenseNet)	35
14	AUC-ROC-Score der einzelnen Pathologien mit Related Work Augmentierungen (DenseNet)	36
15	Precision, Recall und $F_1 - Score$ der einzelnen Pathologien mit affine_color mit Threshold 0.5 (ResNET18)	38
16	Precision, Recall und $F_1 - Score$ der einzelnen Pathologien mit affine_color mit Threshold 0.2 (ResNET18)	38
17	Precision, Recall und $F_1 - Score$ der einzelnen Pathologien mit small_affine mit Threshold 0.5 (ResNET18)	39
18	Precision, Recall und $F_1 - Score$ der einzelnen Pathologien mit small_affine mit Threshold 0.2 (ResNET18)	39
19	AUC-ROC-Score der einzelnen Pathologien mit Augmentierungen von affine_color	40
20	AUC-ROC-Score der einzelnen Pathologien mit Augmentierungen von small_affine	40
21	Precision, Recall und $F_1 - Score$ der einzelnen Pathologien mit affine_color mit Threshold 0.2 (DenseNet)	42
22	Precision, Recall und $F_1 - Score$ der einzelnen Pathologien mit small_affiner mit Threshold 0.2 (DenseNet)	42
23	AUC-ROC-Score der einzelnen Pathologien mit Augmentierungen von affine_color (DenseNet)	43
24	AUC-ROC-Score der einzelnen Pathologien mit Augmentierungen von small_affine (DenseNet)	43

25	Precision, Recall und $F_1 - Score$ der einzelnen Pathologien mit vortrainiertem <code>small_affine</code> und Threshold 0.5	45
26	Precision, Recall und $F_1 - Score$ der einzelnen Pathologien mit vortrainiertem <code>small_affine</code> und Threshold 0.2	46
27	Precision, Recall und $F_1 - Score$ der einzelnen Pathologien mit <code>affine_color</code> und Threshold 0.5	46
28	Precision, Recall und $F_1 - Score$ der einzelnen Pathologien mit <code>affine_color</code> und Threshold 0.2	47
29	AUC-ROC-Score der einzelnen Pathologien mit dem vortrainierten Augmenter <code>small_affine</code>	47
30	AUC-ROC-Score der einzelnen Pathologien mit dem vortrainierten Augmenter <code>affine_color</code>	48
31	Die durchschnittlichen AUC-Scores der einzelnen Methoden	50
32	Confusion Matrizen der einzelnen Pathologien ohne Augmentierungen und Threshold 0.5 (ResNET18)	57
33	Confusion Matrizen der einzelnen Pathologien ohne Augmentierungen und Threshold 0.2 (ResNET18)	58
34	Confusion Matrizen der einzelnen Pathologien ohne Augmentierungen und Threshold 0.5 (DenseNet)	59
35	Confusion Matrizen der einzelnen Pathologien ohne Augmentierungen und Threshold 0.2 (DenseNet)	60
36	Confusion Matrizen der einzelnen Pathologien mit <code>related Work</code> Augmentierungen und Threshold 0.5 (ResNET18)	61
37	Confusion Matrizen der einzelnen Pathologien mit <code>related Work</code> Augmentierungen und Threshold 0.2 (ResNET18)	62
38	Confusion Matrizen der einzelnen Pathologien mit <code>related Work</code> Augmentierungen und Threshold 0.5(DenseNet)	63
39	Confusion Matrizen der einzelnen Pathologien mit <code>related Work</code> Augmentierungen und Threshold 0.2(DenseNet)	64
40	Confusion Matrizen der einzelnen Pathologien mit <code>affine_color</code> und Threshold 0.5 (ResNET)	65
41	Confusion Matrizen der einzelnen Pathologien mit <code>affine_color</code> und Threshold 0.2 (ResNET)	66
42	Confusion Matrizen der einzelnen Pathologien mit <code>small_affine</code> und Threshold 0.5 (ResNET)	67
43	Confusion Matrizen der einzelnen Pathologien mit <code>small_affine</code> und Threshold 0.2 (ResNET)	68
44	Confusion Matrizen der einzelnen Pathologien mit <code>affine_color</code> und Threshold 0.2 (DenseNet)	69
45	Confusion Matrizen der einzelnen Pathologien mit <code>small_affine</code> und Threshold 0.2 (DenseNet)	70
46	Confusion Matrizen der einzelnen Pathologien mit vortrainiertem <code>small_affine</code> und Threshold 0.5 (ResNET)	71
47	Confusion Matrizen der einzelnen Pathologien mit vortrainiertem <code>small_affine</code> und Threshold 0.2 (ResNET)	72
48	Confusion Matrizen der einzelnen Pathologien mit vortrainiertem <code>affine_color</code> und Threshold 0.5 (ResNET)	73
49	Confusion Matrizen der einzelnen Pathologien mit vortrainiertem <code>affine_color</code> und Threshold 0.2 (ResNET)	74

1 Einleitung

Die Verwendung von maschinellem Lernen in der Medizin ist keine neue Idee mehr: Anwendung von ML findet man bei der Diagnostik, Medikamentenentwicklung und der Epidemiologie[23]. Die Organisation DeepMind, die zum Beispiel für das Programm AlphaGo, das erste Programm, das einen professionellen Go Spieler besiegt hat, bekannt ist[13], widmet sich seit 2016 Problemstellungen im Gesundheitswesen. Ein Gebiet von DeepMind ist die Erkennung von Brustkrebsmetastasen mit Hilfe von neuronalen Netzen. Dabei wurde ein FROC-Score von 89 Prozent bei der Erkennung von Metastasen in den benachbarten Lymphknoten erreicht[22]. In der medizinischen Forschung kann maschinelles Lernen auch für die Entwicklung von Impfstoffen verwendet werden. Zikun Yang, Paul Bogdan und Shahin Nazarian verwenden einen Deep Learning Ansatz für die Entwicklung von Multi-epitop-Impfstoffen für den SARS-CoV-2[37].

Für die Qualität eines Neuronalen Netz ist der Datensatz, auf dem das Netz trainiert wird, ausschlaggebend. Was ist aber, wenn der Datensatz begrenzt ist bzw. man schwer an genügend Trainingsdaten kommt? Medizinische Daten sind sensibel und benötigen die Einwilligung der Patienten zur Verwendung. Des Weiteren ist die Erstellung von Röntgenbildern oder CT-Scans teuer und Krankenhäuser sind vorsichtig bei der Weitergabe dieser persönlichen Daten[28]. Es kommt noch dazu, dass das "labeln" der Daten eine schwierige Aufgabe sein kann[29]. Eine Möglichkeit, um den Datensatz künstlich zu erweitern, ist, die Trainingsdaten zu augmentieren. Um passende Augmentierungen für einen Datensatz zu finden, muss häufig Expertenwissen herangezogen werden. Zudem können falsche Augmentierungen beim Klassifizieren von Bilddaten Fehler erzeugen. Zum Beispiel führt bei Thoraxröntgenbildern ein horizontaler Flip dazu, dass das Herz auf der anderen Seite zu sehen ist. Dieser medizinische Zustand heißt Situs Inversus und könnte bei der Klassifizierung zu einem falschen Ergebnis führen[15].

Viele Arbeiten, die hohe Performance zeigen, verwenden Flipping dennoch. Zum Beispiel die Arbeit "Attention-based VGG-16 model for COVID-19 chest X-ray image classification" verwendet sowohl horizontales als auch vertikales Flipping[31]. Auch Anwendungen auf dem CIFAR-10 und ImageNet Datensätzen verwenden Flipping. Auf Datensätzen wie MNIST oder SVHN, also Texterkennung, ist Flipping nicht konsistent mit den Labels. Im Allgemeinen ist der Flip für Augmentierungen noch nicht ausreichend erforscht[29].

Des Weiteren kann es bei einem zu kleinen Datensatz zu Overfitting kommen. Die große Problematik bei Overfitting liegt darin, dass für die Bewertung eines künstlichen neuronalen Netzes unbekannte Daten, das heißt Daten, auf denen das Modell nicht trainiert wurde, verwendet werden[7]. Für das Verhindern von Overfitting gibt es verschiedene Verfahren. Zu den wichtigsten zwei gehören zum einen das Zurückhalten

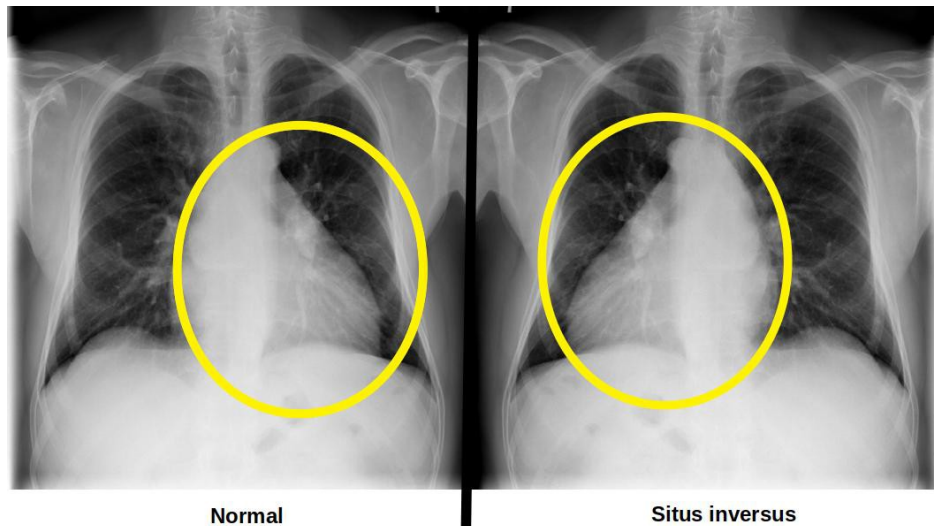


Abbildung 1: Ein Thoraxröntgenbild normal und horizontal geflippt[15]

eines Validationsets. Ein Validierungsset ist eine Teilmenge, die aus den Trainingsdaten genommen wird und nicht zum Trainieren des Netzes genutzt wird. Nachdem das Modell trainiert wurde, validiert man nun das Netz mit den zurückgehaltenen Daten und kann somit bestimmen, wie sich das Netz auf unbekanntem Daten verhält. Die zweite Möglichkeit, Overfitting zu verhindern, sind verschiedene Resampling-Techniken. Dazu gehört zum Beispiel die k-fache Kreuzvalidierung. Dabei wird das Modell k-mal auf verschiedenen Teilmengen der Trainingsdaten trainiert[7].

Einen Experten für die Augmentierungen heranzuziehen kann kostspielig werden und es kann sein, dass nicht immer ein Experte für ein bestimmtes Thema zur Verfügung steht. Mounsaveng et al. publizierten im Januar 2021 die Online Bilevel Optimization Methode, um für einen Datensatz über ein neuronales Netz passende Augmentierungen zu erhalten. Diese Methode ist ein Schritt in die Richtung vollautomatisierte Lernsystem zu konstruieren, die minimalen menschlichen Eingriff benötigen[25]. Im Folgenden wird diese Methode auf Thoraxröntgenbildern angewendet und evaluiert.

2 Funktionsweisen

2.1 Neuronale Netze

Ein neuronales Netzwerk ist ein Netz aus einzelnen Neuronen. Ein Neuron, auch Perzeptron genannt, ist der Grundbaustein von neuronalen Netzwerken (Abb 2). Dieses Neuron nimmt gewichtete Werte und eine Konstante, den Bias, als Eingabewerte auf und berechnet über die Übertragungsfunktion Σ und die Aktivierungsfunktion f (häufig auch φ) den Output o .

$$\Sigma = \sum_{i=0}^n w_i \cdot x_i + b \quad (1)$$

$$o = f(\Sigma) \quad (2)$$

Für die Aktivierungsfunktion gibt es verschiedene Funktionstypen, die abhängig von der gewählten Netztopologie sind. Allgemein sind diese Funktionen monoton steigend. Beispiele für solche Funktionen sind Sprungfunktionen (auch Schwellenwertfunktion), stückweise lineare Funktionen, Sigmoid Funktionen und Rectifer (ReLU = Rectified Linear Unit) (siehe Abb 3)[3]. Ein neuronales Netz besteht nun aus mehreren Schichten solcher Perzeptronen. Als Standard wird meist ein Feed-Forward-Netz (FFN) genommen. In einem solchen FFN beginnt man mit der Input-Schicht und rechnet durch sogenannte Hidden-Layers bis zur Output-Schicht. Die Anzahl dieser Hidden-Layers variiert mit der Aufgabe, die das neuronale Netz ausführen soll.

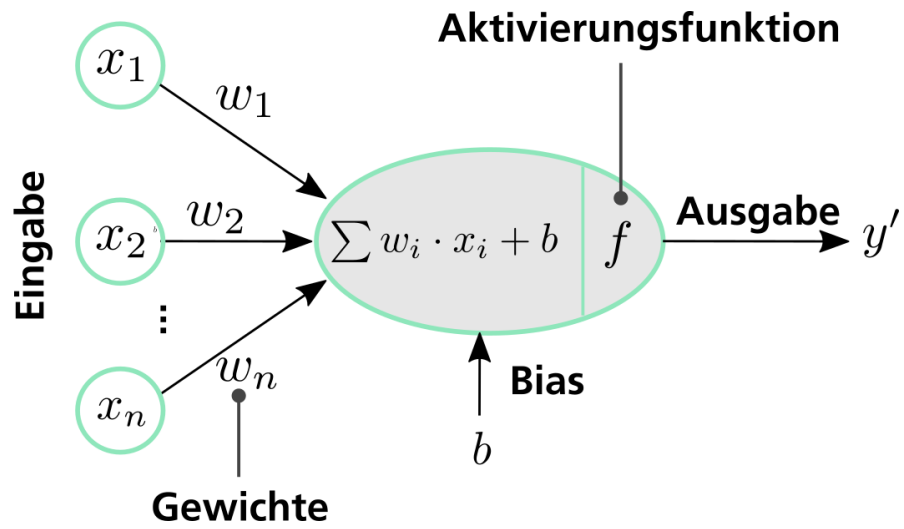
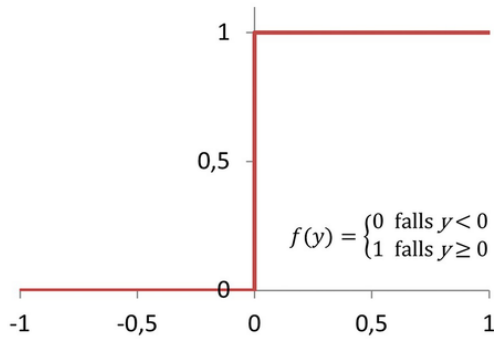
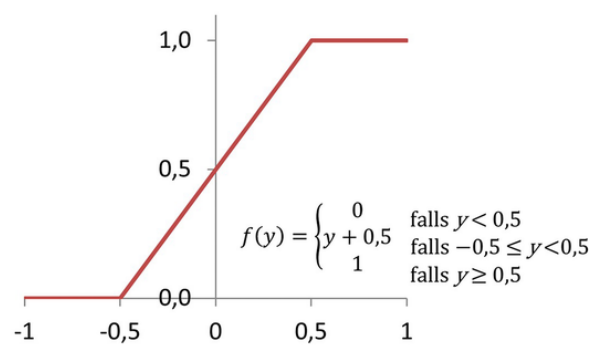


Abbildung 2: Eine einfache Darstellung eines künstlichen Neurons(Perzeptron)[27]

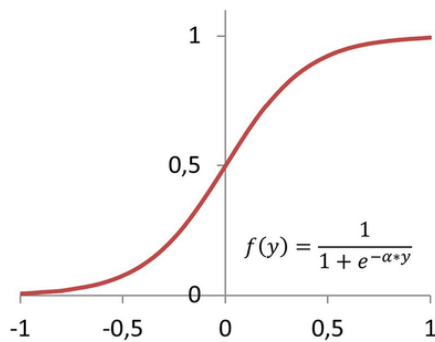
Schwellenwertfunktion



Stückweise lineare Funktion



Sigmoid



ReLU

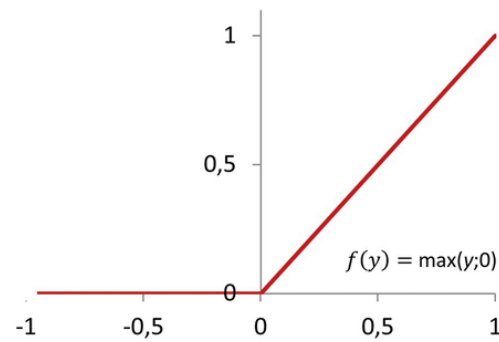


Abbildung 3: Aktivierungsfunktionen[3]

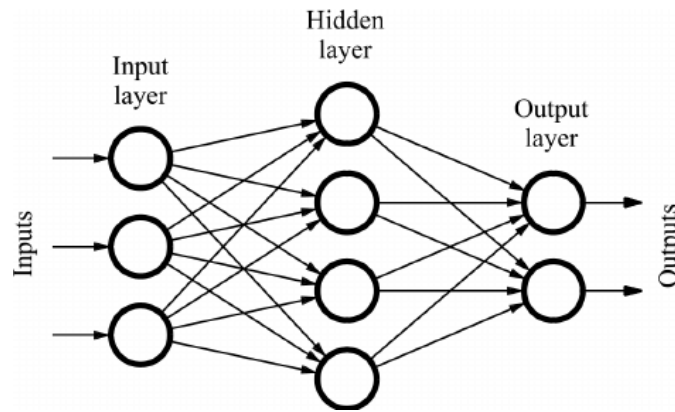


Abbildung 4: Einfache Darstellung eines Feed-Forward-Netztes[11]

2.1.1 Verlustfunktion

Um die Güte eines neuronalen Netzes zu bestimmen, definiert man eine Verlustfunktion, diese kann auch mehrdimensional sein. Diese Verlustfunktion gilt es dann zu minimieren. Ist die Verlustfunktion gleich null, so wurde jede Instanz korrekt klassifiziert. Bei Aufgaben zur Klassifikation von Daten wird häufig eine logarithmische Verlustfunktion verwendet. Die Minimierung der Verlustfunktion erfolgt durch Gradient Descent. Durch diesen Algorithmus wird das globale Minimum der Verlustfunktion gefunden. Wie schnell und ob man das globale Minimum findet, hängt von der Lernrate ab[3]. Beim Trainieren wird für die Klassifizierungsnetze in dieser Arbeit der Loss sowohl für das Trainingsset als auch für das Validierungsset berechnet. Der Loss wird mit Gleichung 3, dem Binary-Cross-Entropy-Loss, berechnet. Dabei ist y das Label und $p(y)$ die vorhergesagte Wahrscheinlichkeit, dass der berechnete Punkt zu y gehört für alle Punkte N . Je höher die prognostizierte Wahrscheinlichkeit einer Klasse bzw. eines Labels ist, desto geringer ist der Loss.

$$H_p(a) = -\frac{1}{N} \sum_{i=1}^N y_i \cdot \log p(y_i) + (1 - y_i) \cdot \log (1 - p(y_i)) \quad (3)$$

2.1.2 Lernrate

Ist die Lernrate zu groß, kann das globale Minimum übersprungen werden, ist die Lernrate zu klein, bleibt man gegebenenfalls in einem lokalen Minimum stecken. Daher beginnt man idealerweise mit einer großen Lernrate und verringert diese im Laufe des Prozesses. Da man bei neuronalen Netzen mit großen Datenmengen arbeitet, setzt man auf die Methode des stochastic gradient descent, dabei wird anstatt des kompletten Trainingssets ein kleiner, zufällig gewählter Teil (sog Mini-Batch) oder ein einzelnes zufälliges Beispiel für die Berechnung des Gradient der Verlustfunktion verwendet[3].

2.1.3 Backpropagation und Optimizer

In einem neuronalen Netze werden die Parameter normalerweise im ersten Schritt zufällig gesetzt. Somit ist die Wahrscheinlichkeit gering, am Output die richtigen Ergebnisse zu erhalten. Man berechnet nun den Loss mit diesen zufällig gesetzten Parametern. Dazu nutzt man die vorher festgelegte Verlustfunktion. Um die Parameter der einzelnen Schichten anzupassen, geht man rückwärts durch das Netz und gewichtet die einzelnen Neuronen nach ihrem Fehleranteil. Dies wiederholt man bis an alle Neuronen bis zur Inputschicht gewichtet hat und passt dann die Parameter entgegen dem Verlustes an. Das ist der Algorithmus der Backpropagation[2]. Als Optimizer wird in dieser Arbeit Adam verwendet. Adam kann als die Kombination von stochastic gradient descent (SGD) und RMSprop gesehen werden. Der Optimizer nutzt die quadrierten Gradienten zur Skalierung der Lernrate wie RMSprop und nutzt den moving average des Gradienten wie SGD mit momentum[8]. Der Optimizer berechnet den Gradienten.

Somit lernt ein neuronales Netz über die Losses und den Gradienten mit dem Backpropagation-Algorithmus.

2.2 Bildklassifikation

Für Bildklassifikation haben sich Convolutional Neural Networks, kurz CNNs, als hilfreiches Werkzeug herausgestellt. Man findet CNNs zum Beispiel in Facebooks Gesichtserkennung (Abb 6)[26] oder auch in selbstfahrenden Autos[14]. Bildklassifikation ist der Prozess, einen Input, wie z. B. ein Bild, in eine Klasse einzuordnen bzw. eine Wahrscheinlichkeit anzugeben zum Beispiel, dass zu 90% auf dem Bild eine Katze zu sehen ist.

CNNs sind aufgebaut wie Feed-Forward-Netze, das heißt, es gibt eine Input-Schicht, danach folgen Hidden-Layer und am Schluss wieder die Output-Schicht (Abb 4). Die Hidden-Layers bestehen wiederum aus verschiedenen Schichten, dazu gehören Convolutional-Layers (CL), ReLU-Layers, Pooling-Layers und Fully-Connected-Layers (FC). Der wichtigste Teil davon sind die CL, diese wenden Faltungen, daher auch der Name (engl. convolution), am Input an. Ein schematisches Beispiel zu einem solchen CL sieht man in Abbildung 5 Diese CL extrahieren bestimmte Features des Inputs, das Netz lernt dieses Features durch diese Hidden-Layers. Je tiefer ein Netz ist, desto komplexer werden die Eigenschaften, die das Netz erlernen kann. Am Ende steht die Output-Schicht, die aus einem FC besteht. Diese Schicht mapped die Daten dann auf die Anzahl der Klassen, die zu erlernen sind und gibt die Klassenwahrscheinlichkeit an. Zusammengefasst extrahiert ein CNN über die Convolutional-Layers Eigenschaften aus den Inputbildern und erlernt diese. Mit den erlernten Eigenschaften kann das Netz dann die Bilder klassifizieren[5].

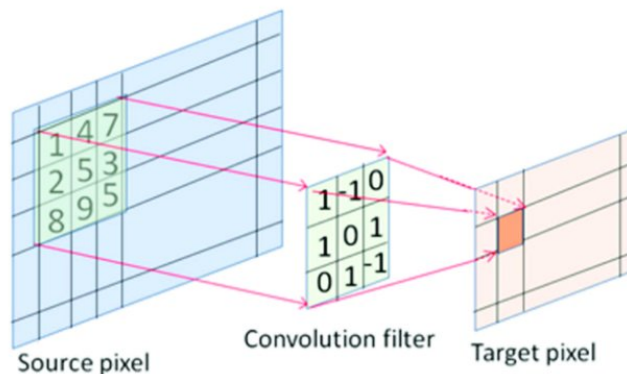


Abbildung 5: Schematische Darstellung eines Convolutional Layers[21]

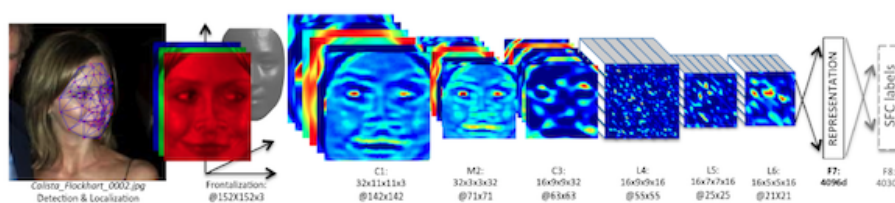


Abbildung 6: Darstellung eines CNNs anhand Facebooks DeepFace Algorithmus[26]

2.3 Overfitting und Datenaugmentierung

Damit Convolutional Neural Networks effizient lernen, ist ein gut ausbalancierter Datensatz notwendig. Da es in manchen Fachgebieten häufig schwierig ist, an Daten zu kommen, kann es beim Trainieren von Netzen zu Problemen kommen. Ein großes Problem dabei ist Overfitting. Overfitting bedeutet, dass bestimmte Eigenschaften des Datensatzes wie zufälliges Rauschen oder kleine Details erlernt werden. Beispiele für Bildrauschen sieht man in Abbildung 7. Dadurch wird der Datensatz zu gut erlernt und das Netz verallgemeinert nicht richtig. Wendet man ein Netz, auf dem Overfitting aufgetreten ist, nun auf neue, für das Netz unbekannte Daten an, so können auch vollkommen unerwartete Ergebnisse bei eigentlich trainierten Eigenschaften auftreten. Bei kleinen Datensätzen kommt es schneller zum Overfitting, da dort das Netz häufiger dieselben Daten zum Lernen bekommt[7].

Um Overfitting zu verhindern, kann man den Datensatz unter anderem künstlich erweitern. Dazu gibt es das Verfahren der Datenaugmentierung. Augmentierungen kann man mit Hilfe von Deep Learning Verfahren erzeugen oder auch durch häufig erprobte und angewendete Bildtransformationen. Unter Augmentierungen auf Bildern versteht man die Veränderung von Daten mit geometrischen Transformationen oder Bildbearbeitung. Für die geometrischen Transformationen haben sich Augmentierungen wie Skalierung, Translationen und Rotation als hilfreich erwiesen[17]. Für Bildbearbeitung eignen sich Veränderungen am Kontrast, Sättigung und Helligkeit der Bilder[25][29].

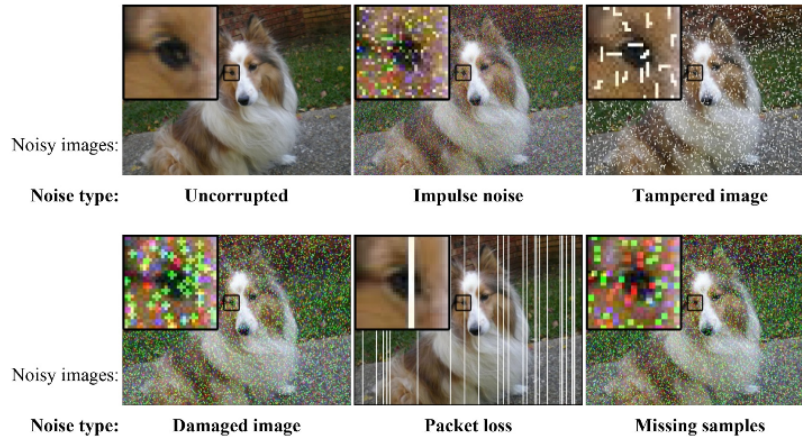


Abbildung 7: Verschiedene Arten von Bildrauschen[24]

Durch diese Augmentierungen an den Bildern erhält man zusätzliche Daten, die dazu noch mit passenden Labels versehen sind, die dann direkt zum Training eines Netzes verwendet werden können. Durch Datenaugmentierung kann jedoch nicht jedes Problem eines Datensatzes gelöst werden. Als Beispiel: Ein Datensatz von Hundebildern, der nur aus Golden Retrievers besteht, wird mit diesen Augmentierungen die Erkennung von Bulldoggen nicht fördern. Enthält der Datensatz aber eine kleine Menge an Bildern von Bulldoggen, kann durch Augmentierung die Erkennung dieser Hunderasse gesteigert werden[29].

2.4 Online Bilevel Optimization

Der Online Bilevel Optimization Algorithmus von Mounsaveng et al. besteht aus einem inneren und einem äußerem Loop, wie in Abbildung 8 zu sehen. Das Augmentierungsnetz berechnet auf Basis eines Noise-Vektors die Augmentierungen, die daraufhin auf ein Bild angewendet werden. Die augmentierten Bilder werden in das Klassifizierungsnetz eingespielt und der Loss auf den Bildern berechnet und der Classifier trainiert. Derselbe Classifier wird für die Validierung im äußerem Loop verwendet. Hierbei wird auf Bildern, die fremd zum Trainingsdatensatz sind, der Validationloss berechnet, dieser Loss wird dann für die Anpassung des Augmentierungsnetzes verwendet[25].

Die Methode kann mal als folgende Formeln darstellen:

$$\theta^* = \arg_{\theta} \min \mathcal{L}(\mathcal{X}_{val}, \omega^*) \quad (4)$$

$$s.t. \omega^* = \arg_{\omega} \min \mathcal{L}(\mathcal{A}_{\theta}(\mathcal{X}_{tr}), \omega) \quad (5)$$

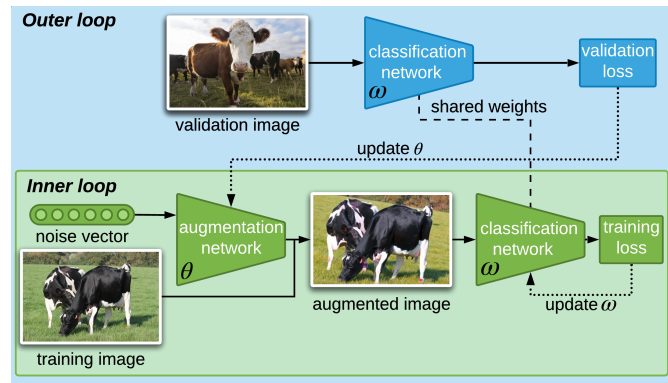


Abbildung 8: Der Online Bilevel Optimization Algorithmus[25]

Das ist ein Bilevel Optimization Problem. Bei dem der Loss \mathcal{L} auf dem Validationset \mathcal{X}_{val} im äußeren Loop minimiert werden soll. Der Classifier ω^* wird standardmäßig als überwachtes Lernen auf den Trainingsdaten \mathcal{X}_{tr} durchgeführt.

2.5 Transfer Learning

Unter Transfer Learning versteht man die Methode, ein Modell, das für eine andere Aufgabe entwickelt wurde, als Startpunkt für eine zweite Aufgabe wiederzuverwenden. Für die Bildklassifizierung ist diese Methode üblich. Dazu verwendet man ein auf einem großen Datensatz wie ImageNET mit 1000 Klassen vortrainiertes CNN. Damit der eigene Datensatz besser zu den Klassen von ImageNET passt, ist es üblich, seine Bilddaten mit dem Mittelwert (mean) und der Standardabweichung (std) der Bilder von ImageNET anzupassen. Somit spart man sich die zeitintensive Aufgabe, ein großes Netz zu erstellen und zu trainieren. Diese vortrainierten Modelle wie z.B das ResNET kann man aus der PyTorch Library einfach als vortrainiertes Netz downloaden und in seine Experimente einbinden[6].

2.5.1 ResNET

Die ResNET Architektur ist eine CNN-Architektur und beruht auf Deep Residual Learning. In der Architektur des ResNETs sind Abkürzungen zwischen einigen Convolutional-Layers gegeben, welche die Berechnungen nehmen, wenn die Dimension des Inputs gleich dem des Outputs ist. Das ResNET gibt es in verschiedenen Varianten: ResNET18 (Abb. 9), ResNET34, ResNET50, ResNET101 und ResNET152. Die Zahl am Ende des Namens gibt die jeweilige Tiefe des Netzes an. Im Paper wurde gezeigt, dass die ResNET Architektur auf dem ImageNET Datensatz sowohl einen geringeren Top1 Error als auch einen geringeren Top5 Error als State-of-the-Art Methoden wie das VGG-Netz hat[16]. Der Top1 Error gibt an, wie oft das Netz das richtige Label mit der höchsten Wahrscheinlichkeit berechnet hat. Der Top5 Error gibt an, wie häufig das richtige Label in den fünf häufigsten vorhergesagten Klassen vorkommt[12].

ResNet18 - Structural Details														
#	Input Image			output			Layer	Stride	Pad	Kernel		in	out	Param
1	227	227	3	112	112	64	conv1	2	1	7	7	3	64	9472
	112	112	64	56	56	64	maxpool	2	0.5	3	3	64	64	0
2	56	56	64	56	56	64	conv2-1	1	1	3	3	64	64	36928
3	56	56	64	56	56	64	conv2-2	1	1	3	3	64	64	36928
4	56	56	64	56	56	64	conv2-3	1	1	3	3	64	64	36928
5	56	56	64	56	56	64	conv2-4	1	1	3	3	64	64	36928
6	56	56	64	28	28	128	conv3-1	2	0.5	3	3	64	128	73856
7	28	28	128	28	28	128	conv3-2	1	1	3	3	128	128	147584
8	28	28	128	28	28	128	conv3-3	1	1	3	3	128	128	147584
9	28	28	128	28	28	128	conv3-4	1	1	3	3	128	128	147584
10	28	28	128	14	14	256	conv4-1	2	0.5	3	3	128	256	295168
11	14	14	256	14	14	256	conv4-2	1	1	3	3	256	256	590080
12	14	14	256	14	14	256	conv4-3	1	1	3	3	256	256	590080
13	14	14	256	14	14	256	conv4-4	1	1	3	3	256	256	590080
14	14	14	256	7	7	512	conv5-1	2	0.5	3	3	256	512	1180160
15	7	7	512	7	7	512	conv5-2	1	1	3	3	512	512	2359808
16	7	7	512	7	7	512	conv5-3	1	1	3	3	512	512	2359808
17	7	7	512	7	7	512	conv5-4	1	1	3	3	512	512	2359808
	7	7	512	1	1	512	avg pool	7	0	7	7	512	512	0
18	1	1	512	1	1	1000	fc					512	1000	513000
Total													11,511,784	

Abbildung 9: Datenblatt zu ResNET18

2.5.2 DenseNet

Die Architektur des DenseNets ist wieder die eines CNNs. Im Vergleich zu einem herkömmlichen Netz mit L Layern und L Verbindungen besitzt das DenseNet $\frac{L(L+1)}{2}$ Verbindungen bei L Layern. Das kommt daher, dass jeder Output zu jedem nachkommenden Input eine Verbindung besitzt. In Abbildung 10 sieht man die Darstellung eines Dense-Blocks. Implementationen des DenseNets gibt es wie auch beim ResNET in verschiedenen Tiefen: DenseNet121, DenseNet161, DenseNet169 und DenseNet201. Im Vergleich zum ResNET performt das DenseNet auf demselben Niveau aber mit deutlich weniger Parametern, die trainiert werden müssen. So erhält man bei einem DenseNet201 mit rund 20 Millionen Parametern ähnliche Ergebnisse wie bei einem ResNET101 mit mehr als 40 Millionen Parametern. Ein DenseNet, das den Berechnungsaufwand eines ResNET50 hat, performt auf dem Niveau eines ResNET101, welches doppelt so viel Berechnungsaufwand benötigt[18].

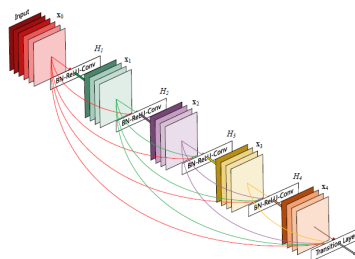


Abbildung 10: Ein 5-Layer Dense-Block aus einem DenseNet[18]

3 Related Work

3.1 Online Bilevel Optimization

Mounsaveng et al. publizierten im Januar 2021 ihre Methode, über Online Bilevel Optimization optimale Augmentierungen für Datensätze zu generieren. Für ihre Experimente wurden der CIFAR10, der CIFAR100, ImageNet, Tiny ImageNet und der BATCH Datensatz verwendet. Ihr Modell besteht aus einem Augmenter und einem Classifier. Für den Classifier wird das ResNET18 und das ResNET50 verwendet. Der Augmenter für geometrische und der für Farbtransformationen ist jeweils ein Multi-Layer-Perzeptron. Das Ziel der Arbeit ist es, dass die Methode automatisch die Augmentierungen erlernt. Die Experimente wurden in zwei Gruppen eingeteilt, in geometrische und farbliche Transformationen. Im Vergleich zu State-of-the-Art Methoden wie GAN-basierte Augmentierungen, hat diese Methode eine höhere Accuracy. Mit einem kleineren Netz hat die Methode von Mounsaveng et al. in etwa die Genauigkeit wie die von AutoAugment. Im Vergleich zu Suchproblem-basierten Methoden performt die Methode auf größeren Netzen in etwa auf demselben Niveau. Dabei braucht die Bilevel Augment Methode weniger Vorwissen der Daten. In Anbetracht stellt dies einen interessanten Kompromiss zwischen Trainingsgeschwindigkeit und Genauigkeit dar, insbesondere für Datensätze, bei denen potenziell nützliche Augmentierungen nicht trivial zu definieren sind[25]. In Abbildung 11 sieht man die des Modells angewendet auf verschiedene Datensätze.

	CIFAR10 ResNet18	CIFAR100 ResNet18	Tiny ImageNet ResNet18	ImageNet ResNet50	BACH ResNet18
Baseline	88.55	68.99	59.69	69.39	49.5
Predefined	94.69	73.61	61.10	76.02	50.6
Ours (affine)	95.16	74.31	62.92	76.10	55.7
Ours (full)	95.42	76.10	63.61	76.20	56.5

Abbildung 11: Genauigkeit des Online Bilevel Optimization Algorithmus auf verschiedenen Datensätzen[25]

3.2 Generative Adversarial Networks

Generative Adversarial Networks erzeugen in erster Linie keine Augmentierungen sondern modellieren Trainingsdaten. Das heißt GANs werden genutzt, um neue Daten aus dem Trainingsset zu generieren. Um die Daten zu erzeugen, werden bei GANs zwei neuronale Netze gegeneinander trainiert. Das Generator Netz wird hierbei verwendet mit dem Ziel, realistische Daten zu erzeugen, das Diskriminator Netz erhält zum einen die vom Generator erzeugten Bilder und zum anderen echte, authentische Bilder und

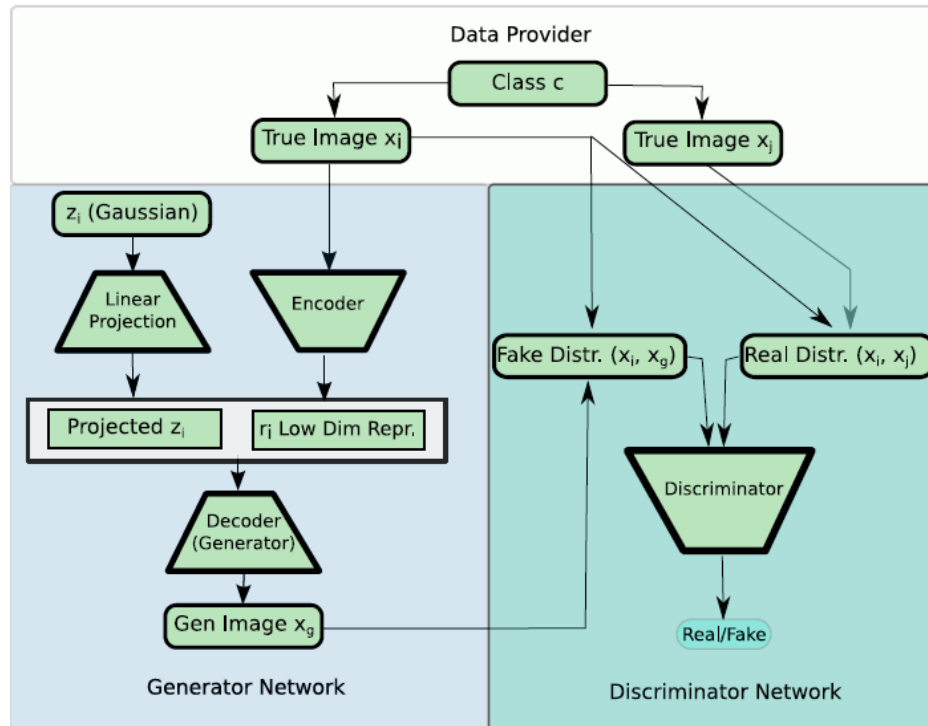


Abbildung 12: Das DAGAN Netzwerk[1]

versucht diese zu unterscheiden. Beide Netze werden gleichzeitig trainiert. Der Generator selbst hat keinen Zugang zu echten Bildern und lernt nur über die Interaktion mit dem Diskriminator[9].

Es gibt verschiedene Varianten von GANs die für Augmentierungen verwendet werden. Eine Variante davon ist CatGAN. CatGAN steht für categorical generative adversarial networks. Man kann CatGANs als ein Framework für unüberwachtes und semi-überwachtes Lernen verwenden[32] [25]. Ein weiteres Netzwerk ist DAGAN, kurz für Data Augmentation Generative Adversarial Network, es wird direkt für die Generierung von Augmentierungen genutzt. Der Aufbau dieses Netzwerk bleibt gleich zu den Standard GANs bis auf den Unterschied, dass aus einer Klasse von echten Bildern ein encodetes Bild mit dem zufälligen Vektor konkateniert wird. Daraus erzeugt der Generator wieder ein Bild. Der Diskriminator unterscheidet echte und generierte Bilder[1]. In Abbildung 12 sieht man das DAGAN Netzwerk.

3.3 AutoAugment

AutoAugment ist eine einfache Methode, die automatisch passende Augmentierungen finden soll. Das Problem, die besten Augmentierungen zu finden, wird als diskretes Suchproblem formuliert. Die Methode von AutoAugment besteht aus zwei Komponenten. Einem Suchalgorithmus und einem Suchraum. Der Suchalgorithmus ist als rekurrentes neuronales Netzwerk implementiert, der Augmentierungsregeln sampelt. Die gesampelten Regeln enthalten Informationen über Operationen, die an den Bildern

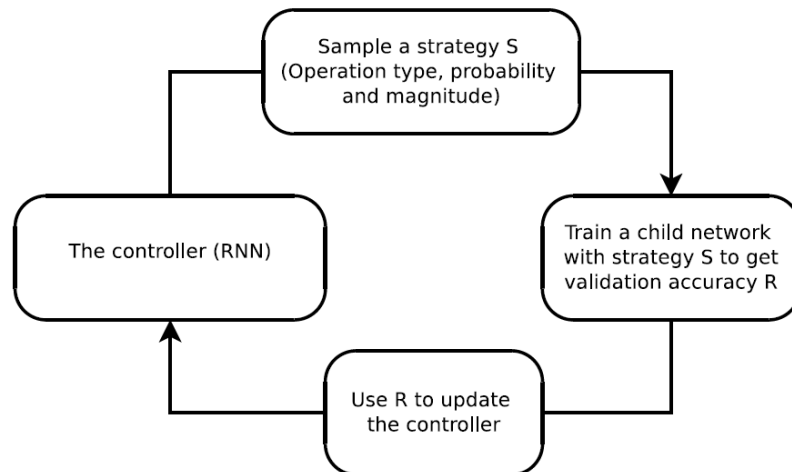


Abbildung 13: Die AutoAugment Methode

durchgeführt werden sollen, über die Wahrscheinlichkeit der Anwendung der Operationen für einen Batch und die Größenordnung der Operationen. Diese Augmentierungsregeln werden verwendet, um ein neuronales Netz zu trainieren. Die Validierungsgenauigkeit des trainierten Netzes wird zurück an den Suchalgorithmus gegeben, um diesen zu updaten[10][25]. In Abbildung 13 sieht man einen Überblick der AutoAugment Methode.

3.4 Large-scale Robust Deep AUC Maximization: A New Surrogate Loss and Empirical Studies on Medical Image Classification

Die Deep AUC Maximization Methode von Yuan et al. ist die Methode, die bisher auf dem CheXpert Datensatz zu dem besten Classifier geführt hat. Diese Methode hat einen AUC-Score von 0.930 erreicht[20]. Yuan et al. optimieren einen neuartigen Loss für Deep Learning anstatt, wie in früheren Arbeiten, den Standard Cross-Entropy-Loss zu optimieren. Die gewählte Methode hat das Ziel den AUC-Score zu maximieren. Für die Evaluierung medizinischer Klassifikationsproblemen ist der AUC-Score der Standardwert für Vergleiche. Des Weiteren sind Datensätze für medizinische Bildklassifikation meist unbalanciert. Das heißt, es gibt keine gute Verteilung der einzelnen Klassen für die Klassifizierung. Der AUC-Score ist besser geeignet unbalancierte Datensätze zu optimieren. Für die Anwendung zur AUC-Score Maximierung wird die Lösung vorgeschlagen, einen Margin-Based Surrogate Loss in MinMax-Form zu verwenden. Dadurch ist der Loss robuster gegen Noise und einfachen Daten. Aus der Implementierung die-

ses Papers wurden die Augmentierungen für das Experiment mit den Related Work Augmentierungen genommen¹[38].

3.5 CheXpert Competition

Die Stanford ML Group hat mir ihrem CheXpert Datensatz auch eine Competition gestartet. Jeder kann den öffentlichen Datensatz herunterladen und auf seinem Classifier, seiner Methode usw. trainieren. Um die einzelnen Modelle unabhängig zu evaluieren, haben die Publisher ein Testset von Bildern von 500 Patienten, das nicht an die Öffentlichkeit freigegeben wird. Für die Evaluierung muss das trainierte Modell eingereicht werden sowie fünf Pathologien ausgewählt werden, auf denen das Modell getestet wird. Zum Stand des 16. Dezembers 2021 sind auf der Seite des CheXpert Datensatz 186 evaluierte Modelle bzw. Arbeiten nach absteigender Performance gelistet. Der Top AUC-Score wurde von Yuan et al.[38] aufgestellt und liegt bei 0.930² [38][20].

¹https://github.com/optimization-ai/iccv2021_deepauc

²<https://stanfordmlgroup.github.io/competitions/chexpert/>

4 Methoden

Um die Auswirkungen von Overfitting und Augmentierung auf einem Datensatz darzustellen, werden im Folgenden der Datensatz und vier Methoden beschrieben. Die Implementation der Experimente wurde in Python mit PyTorch (Version 1.8.1) umgesetzt, da der Code³ aus dem Paper von Mounsaveng et al. ebenfalls mit PyTorch geschrieben ist. Des Weiteren werden im Paper verschiedene Varianten des ResNETs verwendet. In dieser Arbeit wird das vortrainierte ResNET18 aus der Library torchvision.models (Version 0.9.1) benutzt, da es mit rund 11 Millionen trainierbaren Parametern recht schnell trainiert werden kann. Das letzte Layer des Netzes, das Fully Connected Layer (FC), wird im Vergleich zu dem ResNET18 aus der Library, wie in Abbildung 9 zu sehen, mit einem FC mit 13 Knoten geändert.

4.1 Datensatz

Als Datensatz für die folgenden Experimente wird der CheXpert Datensatz verwendet. Dieser Datensatz besteht aus 224.316 Thorax-Röntgenbildern von insgesamt 65.240 Patienten. Diese Bilder und die dazugehörigen radiologischen Berichte wurden im Zeitraum von 2002 bis 2017 gesammelt und stammen aus dem Stanford Hospital. Die Bilder wurden sowohl von stationären als auch von ambulanten Patienten aufgenommen⁴. Diese Röntgen-Bilder sind gelabelt mit 14 der häufigsten Thorax-Röntgenbefunde. Zur Klassifizierung dieser 14 Befunde wird der Erste weggelassen, da es sich dabei um das Label "Kein Befund" handelt. Die Label haben für eine Pathologie den Wert 0, wenn für diese kein Befund vorliegt, einen Wert von 1, wenn ein Befund vorliegt und einen Wert von -1, wenn der Befund ungewiss ist. In dieser Arbeit soll der Effekt der uncertain Labels nicht untersucht werden und deswegen werden alle Werte von -1 auf 1 gesetzt. Dieser Ansatz wird im Paper zum CheXpert Datensatz als Binary Mapping bezeichnet. Des Weiteren werden in dieser Arbeit nicht alle Bilddateien verwendet, da einige der Röntgenbilder Lateral-Aufnahmen sind. Dadurch, dass nur Frontal-Aufnahmen verwendet werden, verringert sich die Anzahl der Bilder auf 191.229. Diese 191.229 Bilder werden in das Train-, Validation- und das Test-Set eingeteilt. Die Sets werden in 80/10/10 aufgeteilt. Dies entspricht 152.983 Bilder für das Trainings-Set, 19.122 Bilder für das Validation-Set und 19.124 Bilder für das Test-Set. Im CheXpert Datensatz gibt es 202 Frontal-Röntgen Bilder, welche mit Hilfe von 3 Radiologen evaluiert wurden und keine uncertain Labels beinhalten, diese 202 Bilder sind in der Gesamtmenge enthalten, wurden aber dem Test-Set zugeteilt[20].

³https://github.com/ElementAI/bilevel_augment

⁴<https://stanfordmlgroup.github.io/competitions/chexpert/>

4.2 Experimente

Um die Auswirkungen von Datenaugmentierungen zu erkennen, werden im Folgenden 4 Experimente durchgeführt.

- Klassifikation ohne Augmentierungen (1)
- Klassifikation mit Augmentierungen inspiriert durch Related Work (2)
- Klassifikation mit Online Bilevel Optimazation (3)
- Klassifikation mit dem vortrainierten Augmentierungsnetz aus Experiment 3 (4)

(1) Als Erstes wird ein Classifier ohne Augmentierungen trainiert. Der Classifier wird einmal ein ResNET18 von torchvision.models und einmal ein DenseNet ebenfalls von torchvision.models sein. Das ResNET wird über 100 Epochen trainiert und mit einer Batchsize von 32 und das DenseNet über 25 Epochen mit einer Batchsize von 64, für beide Netze wird eine Lernrate von 0.001 verwendet. Bei beiden Netzen wird alle 500 Schritte der Trainingsloss und der Validationloss in einer Datei gespeichert. Berechnet wird der Loss über den Binary-Cross-Entropy-Loss (BCE).

(2) Das zweite Experiment hat denselben Aufbau wie das erste, es wird jeweils ein Classifier aus dem ResNET18 sowie dem DenseNet berechnet. Für das ResNET über 100 Epochen mit einer Batchsize von 32 und für das DenseNet über 25 Epochen mit Batchsize 64. Wieder ist die Lernrate für beide Netze 0.001. Jedoch mit dem Unterschied, dass die Inputbilder, bevor sie das Netz durchlaufen, zufällig augmentiert werden. Die Augmentierungen sind inspiriert durch die Arbeit von Yuan et al. und enthalten Rotationen im Intervall von $[-15^\circ, 15^\circ]$, Translation um 0.05 sowohl vertikal und horizontal und die Skalierungen liegen im Intervall $[0.95, 1.05]$ [38].

(3) Als Drittes wird die Augmentierungsmethode von Mounsaveng et al. verwendet. Dazu werden insgesamt vier Experimente durchgeführt, jeweils zwei mit dem ResNET und zwei mit dem DenseNet. Die zwei Experimente auf den jeweiligen Netzen sind zum Einen das Training des *small_affine* und die des *affine_color* Augmentierungsnetzes aus dem Paper⁵[25]. Das *affine_color* Netz erzeugt geometrische Augmentierungen wie solche im vorherigen Experiment. *affine_color* erzeugt Augmentierungen auf der Basis von Bildverarbeitung wie Anpassung von Kontrast oder Helligkeit. Hierbei werden beide Classifier, also für ResNET und DenseNet, über 25 Epochen trainiert und einer Batchsize von 64. Die Lernrate ist dem Paper nachempfunden und beträgt hierbei 0.1, da im Paper mit dieser Lernrate begonnen wird. Bei diesem Experiment wird das Augmentierungsnetz mit dem geringsten Validationloss gespeichert, da dieses für Experiment vier benötigt wird.

⁵https://github.com/ElementAI/bilevel_augment

(4) Für das letzte Experiment werden die gespeicherten Augmentierungsnetze aus Experiment drei für das Training eines Classifiers verwendet. Da im Online Bilevel Algorithmus der Classifier gleichzeitig mit dem Augmenter trainiert wird, ist es interessant zu sehen, wie sich der Classifier verhält, wenn für die Augmentierungen ein vortrainiertes Augmentierungsnetz verwendet wird. Die Trainingsdauer beträgt 25 Epochen mit einer Batchsize von 64 und einer Lernrate von 0.1. Es wird jeweils ein Classifier für *small_affine* und *affine_color* trainiert.

Alle vier Ansätze für Augmentierungen werden Validiert indem der Classifier mit dem geringsten Validationloss genutzt wird um den AUC-Score und Precision, Recall und der $F_1 - Score$ auf dem Testset berechnet werden.

Netz	Ansatz	Epochen	Batchsize	Lernrate	Bildgröße
ResNET18	1	100	32	0.001	224
ResNET18	2	100	32	0.001	224
ResNET18	3 small_affine	25	64	0.1	224
ResNET18	3 affine_color	25	64	0.1	224
ResNET18	4 small_affine	25	64	0.1	224
ResNET18	4 affine_color	25	64	0.1	224
DenseNet	1	25	64	0.001	112
DenseNet	2	25	64	0.001	112
DenseNet	3 small_afine	25	64	0.1	112
DenseNet	3 affine_color	25	64	0.1	112

Tabelle 1: Überblick der verwendeten Hyperparameter der durchgeführten Experimente.

5 Ergebnisse

5.1 Metriken

Für die Evaluierung werden für die trainierten Netze der AUC-ROC-Score und die Precision, der Recall und der $F_1 - Score$ der einzelnen Labels bzw. Pathologien berechnet. Eine Confusion Matrix ist eine 2x2 Matrix in der die True-Positives, False-Negatives, False-Positives und die True-Negatives der berechneten Werte des Netzes für eine einzelne Pathologie eingetragen werden. Durch die Matrix lassen sich die Metriken Recall, Precision und F_1 recht schnell berechnen, da man die benötigten Werte auf einen Blick erhält. Die Confusion Matrix ist so aufgebaut wie in Tabelle 2[35]. Im Anhang findet man die Confusion Matrizen zu den einzelnen Experimenten.

$$\text{Precision} = \frac{TP}{TP + FP} \quad (6)$$

$$\text{Recall} = \frac{TP}{TP + FN} \quad (7)$$

$$F_1 - Score = \frac{TP}{TP + \frac{1}{2} * (FP + FN)} \quad (8)$$

Ist der Nenner bei Precision, Recall oder dem $F_1 - Score$ null, so wird der Wert der Metrik gleich null gesetzt. Diese drei Metriken werden für die 13 Label des Datensatzes berechnet und der Durchschnitt auf dem Trainingsset ausgegeben. Precision gibt dabei an, welcher Satz an True-Positives unter den gesamten predicted positiv Ergebnissen sind. Die Precision ist somit ein gutes Maß, wenn die Kosten eines falsch positiv bestimmten Ergebnisses hoch sind, zum Beispiel bei der Klassifizierung von Testergebnissen. Bei Virenskans ist eine Precision nahe eins wünschenswert. Der Recall gibt dazu noch an, welcher Satz von True-Positives unter den Actual-Positives liegt. Somit ist der Recall der Wert, der optimiert sein sollte, wenn es hohe Kosten bei falsch negativen Werten gibt. Als Beispiel kann man sich die Krebsdiagnose vorstellen. Eine als falsch negative eingestufte Krebsdiagnose kann für den Patienten ernste gesundheitliche Folgen haben. Der $F_1 - Score$ gibt das Gleichgewicht zwischen Precision und Recall an, im Unterschied zur Accuracy werden hierbei die Ergebnisse nicht von der Anzahl der echt negativen Ergebnisse beeinflusst, da in den meisten Fällen die falsch negativen und falsch positiven Werte zu unerwünschten Situationen führen können[30].

Im Bezug auf den CheXpert Datensatz und den Gedanken an Krankheitserkennung ist

Gesamtzahl	Predicted Negative	Predicted Positivs
Actual Negative	True-Negatives	False Positive
Actual Positive	False Negative	True Positive

Tabelle 2: Aufbau einer Confusion Matrix

ein hoher Wert für Precision, Recall und F1 wünschenswert, da bei falschem Diagnostizieren für einen Patienten großer Schaden entstehen kann.

AUC-ROC steht für die Fläche unter der Receiver Operating Characteristic Kurve. Man erhält die ROC-Kurve indem man die True-Positive-Rate (TPR = Recall) auf der y-Achse und die False-Positive-Rate auf der x-Achse aufträgt. Die TPR und FPR liegen im Intervall $[0,1]$. Die ROC-Kurven der einzelnen Experimente findet man im Anhang.

$$\text{FalsePositiveRate} = FPR = \frac{FP}{FP + TN} \quad (9)$$

Man berechnet die TPR und die FPR mit verschiedenen Thresholds somit verändert sich immer der Anteil der positiv bzw. negativ klassifizierten Werte. Der AUC-ROC-Score liegt im Intervall $[0,1]$. Hat der Classifier einen AUC-ROC-Score von 1 so ist er ein perfekter Classifier und bestimmt alle positiven Werte als solche, ist der Wert 0 so klassifiziert er alle positiven Werte als Negativ und anders herum. Wenn der AUC = 0.5 ist, heißt das, dass der Classifier nicht zwischen Positiv und Negativ entscheiden kann oder eine konstante Klasse für alle Datenpunkte liefert. Der Classifier ist also so gut wie als würde er Raten[4][36].

5.2 Klassifikation ohne Augmentierungen

5.2.1 ResNET18

In Abbildung 14 sieht man die dazugehörigen Losses zum ResNET18. Auf der y-Achse ist der Loss aufgetragen und auf der x-Achse die Berechnungszeit in Sekunden, somit auch der Trainingsfortschritt nach Epochen (50000s \approx 15 Epochen). Man erkennt, dass der Loss auf dem Trainingsset schnell gegen null geht. Der Validierungsverlust beginnt schon frühzeitig anzusteigen. Daraus lässt sich schließen, dass früh Overfitting eintritt.

In Tabelle 3 sieht man die für die einzelnen Pathologien berechneten Metriken mit einem Threshold von 0.5. Für neun von 13 Pathologien sind Precision, Recall und

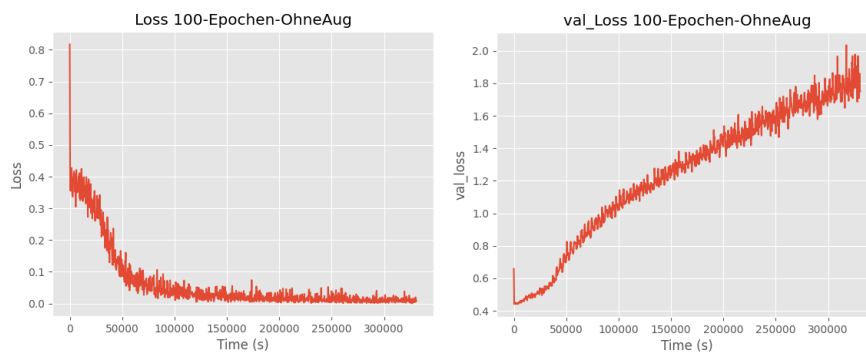


Abbildung 14: Der Loss auf dem Trainingsset (links) und der Validationloss auf dem Validierungsset (rechts) ohne Augmentierungen (ResNET18)

Pathologie	Precision	Recall	$F_1 - Score$
Enlarged Cardi mediastinum	0	0	0
Cardiomegaly	0	0	0
Lung Opacity	0.485	1	0.654
Lung Lesion	0	0	0
Edema	0	0	0
Consolidation	0	0	0
Pneumonia	0	0	0
Atelectasis	0	0	0
Pneumothorax	0	0	0
Pleural Effusion	0.367	0.031	0.057
Pleural Other	0	0	0
Fracture	0	0	0
Support Devices	0.509	1	0.675

Tabelle 3: Precision, Recall und $F_1 - Score$ der einzelnen Pathologien ohne Augmentierungen und Threshold 0.5 (ResNET18)

$F_1 - Score$ gleich Null, da in diesen Fällen das trainierte Netz auf dem Testset keine echten Positiven Werte errechnet hat. In den anderen drei Fällen (Lung Opacity, Pleural Effusion und Support Devices) wurden einige Pathologien richtig klassifiziert. Somit lassen sich auch die drei Metriken berechnen.

Die Precision weist in diesem Fall keinen guten Wert auf, der höchste Wert liegt bei 0.509 für Support Devices. Der Recall ist für Lung Opacity und Support Devices gleich eins. Somit wurden hierbei keine Werte als falsch negativ klassifiziert. Für Pleural Effusion liegt der Wert vom Recall bei 0.031, somit wurde für diese Pathologie die meisten Erkrankungen als falsch negativ eingestuft.

Setzt man den Threshold auf 0.2, erhält man die Werte aus Tabelle 4. So lassen sich für die drei Metriken mehr Werte berechnen. Vier Pathologien haben einen mit dem Threshold 0.2 einen Recall von eins. Setzt man also einen niedrigeren Threshold so erhält man mehr Werte. Auffällig ist, dass der Recall von Pleural Effusion bei dem Threshold von 0.2 gleich eins ist und bei einem Threshold von 0.5 gleich 0.031. Ist der Threshold 0.2 berechnet das Netz keine falsch negativen Werte mehr. Im Gegensatz bei einem Threshold von 0.5 fast ausschließlich falsch negative Werte berechnet wurden.

In Tabelle 5 sieht man die ROC-AUC-Scores für die einzelnen Labels. Der Durchschnitt liegt bei 0.498. Somit kann man darauf schließen, dass das trainierte ResNET18 nicht besser als "Raten" ist.

Pathologie	Precision	Recall	$F_1 - Score$
Enlarged Cardimediastinum	0	0	0
Cardiomegaly	0	0	0
Lung Opacity	0.485	1	0.654
Lung Lesion	0	0	0
Edema	0.334	0.586	0.425
Consolidation	0.218	0.494	0.302
Pneumonia	0	0	0
Atelectasis	0.344	1	0.512
Pneumothorax	0	0	0
Pleural Effusion	0.354	1	0.523
Pleural Other	0	0	0
Fracture	0	0	0
Support Devices	0.509	1	0.675

Tabelle 4: Precision, Recall und $F_1 - Score$ der einzelnen Pathologien ohne Augmentierungen und Threshold 0.2 (ResNET18)

Pathologie	AUC-ROC-Score
Enlarged Cardimediastinum	0.492
Cardiomegaly	0.527
Lung Opacity	0.431
Lung Lesion	0.480
Edema	0.572
Consolidation	0.554
Pneumonia	0.536
Atelectasis	0.489
Pneumothorax	0.503
Pleural Effusion	0.447
Pleural Other	0.489
Fracture	0.456
Support Devices	0.493
Average Score	0.498

Tabelle 5: AUC-ROC-Score der einzelnen Pathologien ohne Augmentierungen (ResNET18)

5.2.2 DenseNet

Der Validationloss, wie in Abbildung 15 bleibt nach ca. zehn Epochen konstant. Das impliziert, dass der Classifier nach diesen zehn Epochen nicht mehr optimiert wird, bzw. das Netz sozusagen ausgelernt hat.

Auch in der Precision, dem Recall und den F_1 – Score sieht man den Trend, dass das Netz nicht gut klassifiziert. Für alle Pathologien außer Support Devices wurden bei einem Threshold von 0.5 keine echt positiven Werte errechnet. Dafür ist der Recall bei Support Devices fast eins. Die Precision ist in etwa 0.5 (Tab 6).

Setzt man beim DenseNet den Threshold auf 0.2 so ändert sich an den Metriken fast nichts. Nur bei Support Devices beträgt der Recall bei einem Threshold von 0.2 eins, anstatt 0.994. Die anderen Werte bleiben gleich 0 (Tab. 7).

Der Durchschnitt des AUC-Scores mit dem DenseNet ohne Augmentierungen ist nach 25 Epochen ein wenig geringer als der des ResNETs. In den einzelnen Pathologien weichen die Werte im Vergleich zwischen ResNET18 und dem DenseNet auch stark ab. So ist beim DenseNet Atelectasis die am besten klassifizierte Klasse mit einem Wert von 0.540 und beim ResNET Edema mit 0.572. Dennoch sind beide Netze ohne Augmentierung schlechter als "Raten". Das die Werte des DenseNets schlechter sind als die des ResNETs könnte zum einen daran liegen, dass DenseNet nicht lang genug trainiert wurde. Außerdem wurde bei dem DenseNet eine geringer Bildauflösung verwendet was die Performance der Klassifizierung beeinflusst.

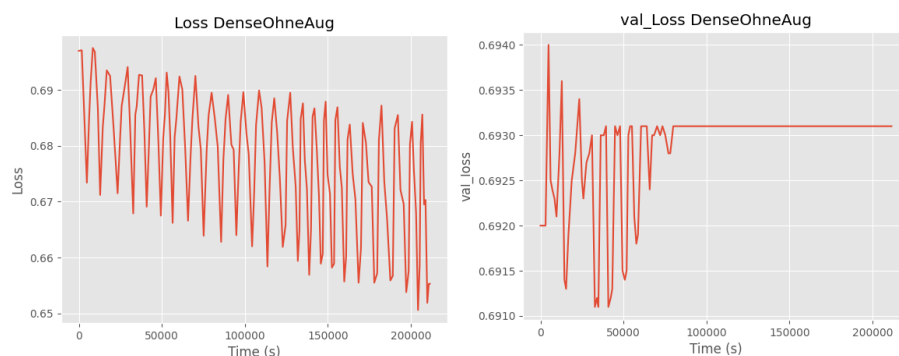


Abbildung 15: Trainings- und Validationloss ohne Augmentierungen (DenseNet)

Pathologie	Precision	Recall	$F_1 - Score$
Enlarged Cardimediastinum	0	0	0
Cardiomegaly	0	0	0
Lung Opacity	0	0	0
Lung Lesion	0	0	0
Edema	0	0	0
Consolidation	0	0	0
Pneumonia	0	0	0
Atelectasis	0	0	0
Pneumothorax	0	0	0
Pleural Effusion	0	0	0
Pleural Other	0	0	0
Fracture	0	0	0
Support Devices	0.509	0.994	0.673

Tabelle 6: Precision, Recall und $F_1 - Score$ der einzelnen Pathologien ohne Augmentierungen und Threshold 0.5 (DenseNet)

Pathologie	Precision	Recall	$F_1 - Score$
Enlarged Cardimediastinum	0	0	0
Cardiomegaly	0	0	0
Lung Opacity	0	0	0
Lung Lesion	0	0	0
Edema	0	0	0
Consolidation	0	0	0
Pneumonia	0	0	0
Atelectasis	0	0	0
Pneumothorax	0	0	0
Pleural Effusion	0	0	0
Pleural Other	0	0	0
Fracture	0	0	0
Support Devices	0.509	1	0.675

Tabelle 7: Precision, Recall und $F_1 - Score$ der einzelnen Pathologien ohne Augmentierungen und Threshold 0.2 (DenseNet)

Pathologie	AUC-ROC-Score
Enlarged Cardi mediastinum	0.505
Cardiomegaly	0.508
Lung Opacity	0.510
Lung Lesion	0.476
Edema	0.477
Consolidation	0.510
Pneumonia	0.464
Atelectasis	0.540
Pneumothorax	0.453
Pleural Effusion	0.511
Pleural Other	0.482
Fracture	0.477
Support Devices	0.514
Average Score	0.494

Tabelle 8: AUC-ROC-Score der einzelnen Pathologien ohne Augmentierungen (DenseNet)

5.3 Klassifikation mit Augmentierungen inspiriert durch Related Work

5.3.1 ResNET18

Die Losses auf dem Trainingsset und dem Validationset mit den Augmentierungen inspiriert durch Related Work sind in Abbildung 16 zu sehen. Im Loss auf dem Trainingsset sieht man im Vergleich zum Training ohne Augmentierungen, dass in diesem Fall der Loss nicht so schnell gegen null geht. Auch auf dem Validierungsset erkennt man, dass Overfitting erkennbar später, nämlich nach ca. 15 Epochen, einsetzt und nicht wie davor quasi direkt zu Beginn. Für die Evaluierungen wurde das Netz mit dem minimalen Validationloss gespeichert, damit man nicht das Netz am Ende des Trainings nutzt, welches deutliche Anzeichen von Overfitting nach 100 Epochen zeigt.

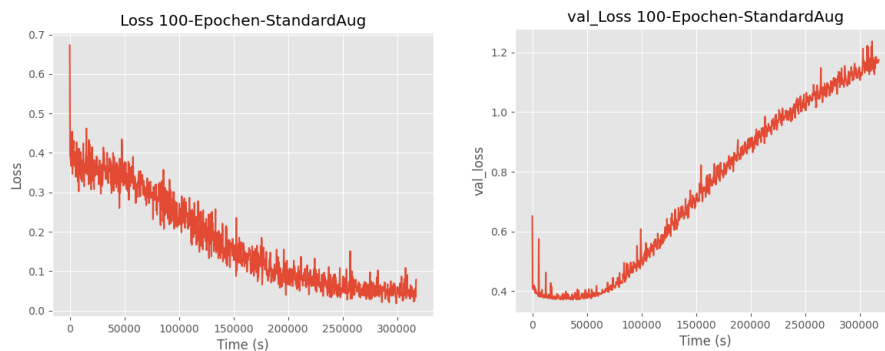


Abbildung 16: Der Loss auf dem Trainingsset (links) und der Validationloss auf dem Validierungsset (rechts) mit related Work Augmentierungen

Pathologie	Precision	Recall	$F_1 - Score$
Enlarged Cardiomediastinum	0	0	0
Cardiomegaly	0	0	0
Lung Opacity	0.592	0.433	0.500
Lung Lesion	0	0	0
Edema	0.643	0.001	0.003
Consolidation	0	0	0
Pneumonia	0	0	0
Atelectasis	0	0	0
Pneumothorax	0	0	0
Pleural Effusion	0.490	0.635	0.553
Pleural Other	0	0	0
Fracture	0	0	0
Support Devices	0.538	0.943	0.685

Tabelle 9: Precision, Recall und $F_1 - Score$ der einzelnen Pathologien mit related Work Augmentierungen und Threshold 0.5 (ResNET18)

In Tabelle 9 sind Precision, Recall und der $F_1 - Score$ für das mit Augmentierungen trainierte ResNET zu sehen bei einem Threshold von 0.5. In diesem Fall gibt es nun Werte für vier der Pathologien. In den anderen acht Fällen wurden für die True-Positives wieder keine Werte berechnet, somit sind also auch die Werte null. Der Durchschnitt der Precision über allen Labels ist gestiegen im Vergleich zu den Werten ohne Augmentierungen. Der durchschnittliche Recall im Gegensatz dazu ist gesunken. Der niedrigste Wert für den Recall liegt bei dem Label Edema und ist 0.001, das bedeutet, dass fast alle Werte als falsch negativ eingestuft wurden, was bei der Erkennung von Krankheiten ein großes Problem darstellt. Der schlechteste Recall des ohne Augmentierungen trainierten Classifiers hat sich mit den hier verwendeten Augmentierungen deutlich verbessert, von 0.031 auf 0.635.

Setzt man den Threshold auf 0.2 so erhält man die Werte in Tabelle 10. Für drei Klassen ist der Wert null. Der Recall ist für fünf der 13 Pathologien hoch, bei drei sogar fast bzw. gleich eins. Vier Pathologien haben einen Recall kleiner als 0.1. Somit berechnet der Classifier mit einem Threshold von 0.2 auf fünf Pathologien einen hohen Anteil von echt positiven Werten. Bei den restlichen Klassen überwiegen die falsch negativen Werte.

Der AUC-ROC-Score für das Netz mit Augmentierungen hat sich im Vergleich zu dem vorherigen Versuch ohne Augmentierungen verbessert. Der Durchschnitt ist um 0.128 Punkte gestiegen auf 0.626. Somit performed das mit Augmentierungen trainierte Netz besser als das vorherige.

Pathologie	Precision	Recall	$F_1 - Score$
Enlarged Cardimediastinum	0.133	0.002	0.004
Cardiomegaly	0.429	0.001	0.002
Lung Opacity	0.485	1	0.654
Lung Lesion	0	0	0
Edema	0.346	0.855	0.493
Consolidation	0.237	0.710	0.355
Pneumonia	0.526	0.005	0.010
Atelectasis	0.355	0.983	0.522
Pneumothorax	0.436	0.012	0.023
Pleural Effusion	0.357	0.998	0.526
Pleural Other	0	0	0
Fracture	0	0	0
Support Devices	0.509	1	0.675

Tabelle 10: Precision, Recall und $F_1 - Score$ der einzelnen Pathologien mit related Work Augmentierungen und Threshold 0.2 (ResNET18)

Pathologie	AUC-ROC-Score
Enlarged Cardimediastinum	0.545
Cardiomegaly	0.606
Lung Opacity	0.623
Lung Lesion	0.597
Edema	0.665
Consolidation	0.622
Pneumonia	0.643
Atelectasis	0.615
Pneumothorax	0.645
Pleural Effusion	0.686
Pleural Other	0.605
Fracture	0.600
Support Devices	0.690
Average Score	0.626

Tabelle 11: AUC-ROC-Score der einzelnen Pathologien mit Augmentierungen inspiriert durch Related Work (ResNET18)

5.3.2 DensNet

In der Abbildung 17 sieht man die Losses auf dem Trainingsset und dem Validationset vom DenseNet. Dadurch, dass das DenseNet tiefer ist als das ResNET sind nach 25 Epochen noch keine Anzeichen von Overfitting zu erkennen. Im Vergleich tritt beim ResNET18 schon nach ca. 15 Epochen Overfitting ein.

Für den Threshold von 0.5 konnte das DenseNet nur bei drei Pathologien True-Positives berechnen, das ResNET mit den geringsten Validationloss hatte zusätzlich noch bei der Pathologie Edema echt positive Werte errechnet. Deshalb sieht man in Tabelle 12 nur bei Lung Opacity, Pleural Effusion und Support Devices Werte für Precision, Recall und den F_1 - Score. Die Werte die das DenseNet errechnen hat können, fallen schlechter aus als die des ResNETs. Das liegt daran, dass das DenseNet nach 25 Epochen noch nicht optimal trainiert wurde bzw. der Verlust auf dem Validierungsset noch nicht minimal war.

Für einen Threshold von 0.2 ändern sich nur die Werte der drei Pathologien für die auch mit einem Threshold von 0.5 Werte berechnet werden konnten. Bei diesen Klassen steigt jeweils der Recall und die Precision sinkt (Tab 13).

Der durchschnittliche AUC-Score (Tab. 14) ist im Vergleich zu dem Netz ohne Augmentierungen höher. Dennoch ist der Durchschnitt geringer als der des ResNETs mit Related Work Augmentierungen. Bei einigen Pathologien liegen die Werte unter 0.5, was bedeutet, dass auf diesen Klassen das DenseNet schlechter als "Raten" ist. Der Durchschnitt liegt knapp über 0.5 bei 0.533, somit ist der Classifier besser als "Raten", aber ist dennoch schlechter als das ResNET18. Das liegt wiederum an dem Punkt, dass das DenseNet nach den 25 Epochen noch nicht das Minimum im Validationloss erreicht hat.

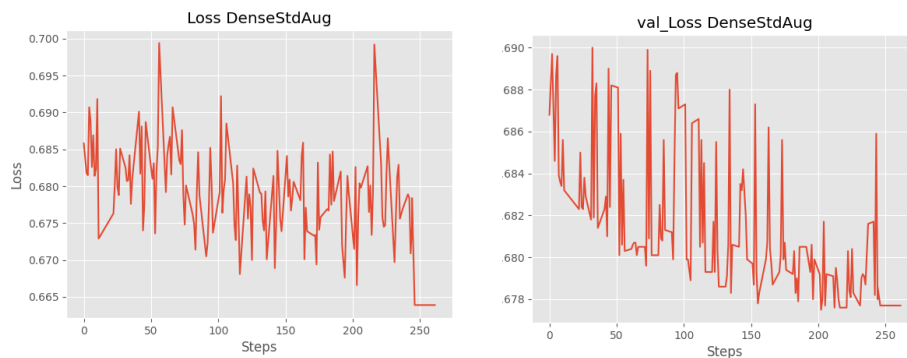


Abbildung 17: Trainingsloss und Validationloss mit related work Augmentierungen mit dem DenseNet

Pathologie	Precision	Recall	$F_1 - Score$
Enlarged Cardimediastinum	0	0	0
Cardiomegaly	0	0	0
Lung Opacity	0.596	0.412	0.487
Lung Lesion	0	0	0
Edema	0	0	0
Consolidation	0	0	0
Pneumonia	0	0	0
Atelectasis	0	0	0
Pneumothorax	0	0	0
Pleural Effusion	0.492	0.348	0.407
Pleural Other	0	0	0
Fracture	0	0	0
Support Devices	0.699	0.443	0.542

Tabelle 12: Precision, Recall und $F_1 - Score$ mit Related Work Augmentierungen der einzelnen Pathologien mit Threshold 0.5 (DenseNet)

Pathologie	Precision	Recall	$F_1 - Score$
Enlarged Cardimediastinum	0	0	0
Cardiomegaly	0	0	0
Lung Opacity	0.551	0.726	0.626
Lung Lesion	0	0	0
Edema	0	0	0
Consolidation	0	0	0
Pneumonia	0	0	0
Atelectasis	0	0	0
Pneumothorax	0	0	0
Pleural Effusion	0.455	0.601	0.518
Pleural Other	0	0	0
Fracture	0	0	0
Support Devices	0.644	0.619	0.631

Tabelle 13: Precision, Recall und $F_1 - Score$ mit Related Work Augmentierungen der einzelnen Pathologien mit Threshold 0.2 (DenseNet)

Pathologie	AUC-ROC-Score
Enlarged Cardi mediastinum	0.526
Cardiomegaly	0.507
Lung Opacity	0.622
Lung Lesion	0.492
Edema	0.447
Consolidation	0.504
Pneumonia	0.446
Atelectasis	0.540
Pneumothorax	0.534
Pleural Effusion	0.649
Pleural Other	0.471
Fracture	0.505
Support Devices	0.681
Average Score	0.533

Tabelle 14: AUC-ROC-Score der einzelnen Pathologien mit Related Work Augmentierungen (DenseNet)

5.4 Klassifikation mit Online Bilevel Optimization

5.4.1 ResNET18

In Abbildung 18 sieht man den Loss und Validationloss des ersten Augmentierungsnetzes `affine_color`. Nach etwa zehn Epochen stellt sich ein Mittel in dem Validationloss ein. Allgemein fällt der Loss auf den Trainings- und Validierungsdaten langsamer als bei den zwei vorherigen Experimenten.

Das zweite Augmentierungsnetz `small_affine` besitzt die Kurven, die in Abbildung 19 zu sehen sind. Man erkennt, dass nach ca. 15 Epochen ein Minimum im Validationloss erreicht wurde und danach der Validierungsloss wieder zu steigen anfängt. Daraus lässt sich schließen, dass nach diesen in etwa 15 Epochen das Netz wieder zu overfitten beginnt. Der Loss auf dem Trainingsset ist in etwa zur selben Zeit auf dem niedrigsten Wert sowie das Minimum im Validationloss erreicht wird.

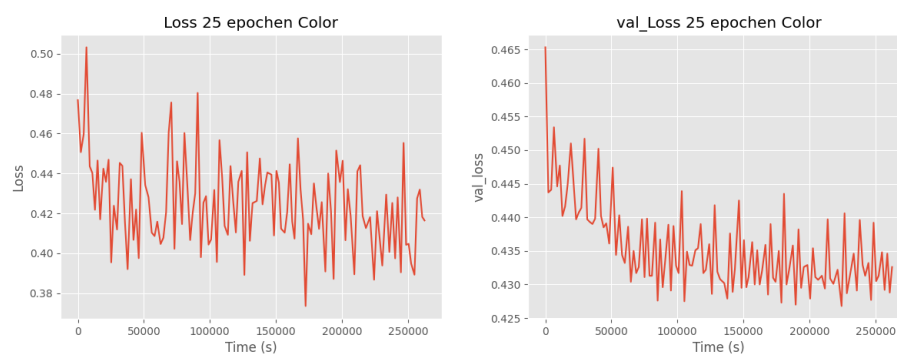


Abbildung 18: Der Loss und Validationloss von `affine_color`

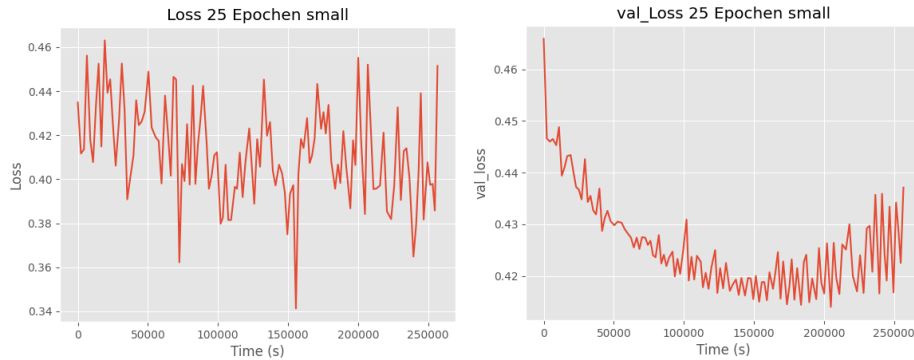


Abbildung 19: Der Loss und Validationloss von `small_affine`

Bei den Metriken Precision, Recall und dem F_1 – Score schneidet das `affine_color` Netz sehr schlecht ab. Von 13 Pathologien wurden bei zwölf davon keine echt positiven Werte errechnet bei einem Threshold von 0.5. Nur bei Support Devices konnten True-Positives errechnet werden, dennoch sind die Werte nicht optimal. Der Recall liegt bei 0.371 und die Precision bei 0.568 (Tab. 15.)

Liegt der Threshold bei 0.2 so erreicht erhält man für vier Pathologien einen Recall von eins. Insgesamt können bei sieben von 13 Klassen die drei Metriken berechnet werden. Mit diesem Threshold werden für fünf der Pathologien hauptsächlich echt positive Werte berechnet. Für Consolidation liegt der Recall bei 0.049, somit werden hierbei überwiegend falsch negative Werte berechnet.

Das `small_affine` Netz schneidet in der Summe der Treffer bis dato am besten ab. Bei 6 von 13 Pathologien errechnete der Classifier mit den Augmentierungen aus `small_affine` die meisten True-Positives. Die Werte der Metriken sind dennoch nicht optimal, der Recall liegt bei Edema und Atelectasis unter 0.1 und den höchsten Wert für den Recall findet man bei Support Devices mit 0.506 (Tab 17).

Auch wenn man den Threshold für `small_affine` auf 0.2 setzt werden für mehr Klassen Werte berechnet (Tab 18). In diesem Fall für acht der 13 Pathologien. Im Allgemeinen steigt der Recall für diese acht Klassen. Die Precision fällt.

Der durchschnittliche AUC-Score des Classifiers, der mit `affine_color` trainiert wurde, ist höher als der Wert des Classifiers, der ohne Augmentierungen trainiert wurde. Im Vergleich zu den Augmentierungen aus dem Paper von Yuan et al.[38] schneidet `affine_color` schlechter ab, der durchschnittliche AUC-Score ist mit 0.611 geringer als der mit den Related Work Augmentierungen.

Der mit dem `small_affine` Augmentierungsnetz trainierte Classifier schneidet unter den bisherigen Methoden knapp als Bestes ab. Der durchschnittliche AUC-Score ist mit 0.003 Punkten mehr als die Augmentierungen von Yuan et al. erreicht haben und somit marginal besser. Bei zwei Pathologien wurde ein Wert von über 0.7 Punkten erreicht. Cardiomegaly hat einen AUC-Score von 0.726, Pleural Effusion sogar von 0.755. Der Classifier trainiert mit dem `small_affine` Netz schneidet besser als der mit `affi-`

ne_color ab. Das fällt vorallem beim Precision, Recall und $F_1 - Score$ auf. Bei gleichem Threshold, von 0.5, erhält man bei dem Classifier mit *small_affine* Werte für sechs von 13 Pathologien. Beim Classifier mit *affine_color* nur bei einer.

Pathologie	Precision	Recall	$F_1 - Score$
Enlarged Cardimediastinum	0	0	0
Cardiomegaly	0	0	0
Lung Opacity	0	0	0
Lung Lesion	0	0	0
Edema	0	0	0
Consolidation	0	0	0
Pneumonia	0	0	0
Atelectasis	0	0	0
Pneumothorax	0	0	0
Pleural Effusion	0	0	0
Pleural Other	0	0	0
Fracture	0	0	0
Support Devices	0.568	0.371	0.449

Tabelle 15: Precision, Recall und $F_1 - Score$ der einzelnen Pathologien mit *affine_color* mit Threshold 0.5 (ResNET18)

Pathologie	Precision	Recall	$F_1 - Score$
Enlarged Cardimediastinum	0	0	0
Cardiomegaly	0.204	0.292	0.240
Lung Opacity	0.485	1	0.654
Lung Lesion	0	0	0
Edema	0.308	0.977	0.468
Consolidation	0.296	0.049	0.084
Pneumonia	0	0	0
Atelectasis	0.344	1	0.512
Pneumothorax	0	0	0
Pleural Effusion	0.354	1	0.523
Pleural Other	0	0	0
Fracture	0	0	0
Support Devices	0.509	1	0.675

Tabelle 16: Precision, Recall und $F_1 - Score$ der einzelnen Pathologien mit *affine_color* mit Threshold 0.2 (ResNET18)

Pathologie	Precision	Recall	$F_1 - Score$
Enlarged Cardi mediastinum	0	0	0
Cardiomegaly	0.485	0.129	0.204
Lung Opacity	0.651	0.363	0.466
Lung Lesion	0	0	0
Edema	0.585	0.086	0.151
Consolidation	0	0	0
Pneumonia	0	0	0
Atelectasis	0.518	0.044	0.081
Pneumothorax	0.712	0.350	0.469
Pleural Effusion	0	0	0
Pleural Other	0	0	0
Fracture	0	0	0
Support Devices	0.595	0.506	0.547

Tabelle 17: Precision, Recall und $F_1 - Score$ der einzelnen Pathologien mit small_affine mit Threshold 0.5 (ResNET18)

Pathologie	Precision	Recall	$F_1 - Score$
Enlarged Cardi mediastinum	0	0	0
Cardiomegaly	0.264	0.604	0.367
Lung Opacity	0.486	0.999	0.654
Lung Lesion	0	0	0
Edema	0.421	0.662	0.515
Consolidation	0.279	0.447	0.344
Pneumonia	0	0	0
Atelectasis	0.368	0.888	0.520
Pneumothorax	0.159	0.043	0.068
Pleural Effusion	0.532	0.734	0.617
Pleural Other	0	0	0
Fracture	0	0	0
Support Devices	0.509	1	0.675

Tabelle 18: Precision, Recall und $F_1 - Score$ der einzelnen Pathologien mit small_affine mit Threshold 0.2 (ResNET18)

Pathologie	AUC-ROC-Score
Enlarged Cardimediastinum	0.560
Cardiomegaly	0.635
Lung Opacity	0.648
Lung Lesion	0.547
Edema	0.679
Consolidation	0.643
Pneumonia	0.579
Atelectasis	0.585
Pneumothorax	0.575
Pleural Effusion	0.733
Pleural Other	0.580
Fracture	0.618
Support Devices	0.557
Average Score	0.611

Tabelle 19: AUC-ROC-Score der einzelnen Pathologien mit Augmentierungen von affine_color

Pathologie	AUC-ROC-Score
Enlarged Cardimediastinum	0.518
Cardiomegaly	0.726
Lung Opacity	0.654
Lung Lesion	0.605
Edema	0.693
Consolidation	0.628
Pneumonia	0.597
Atelectasis	0.606
Pneumothorax	0.584
Pleural Effusion	0.755
Pleural Other	0.583
Fracture	0.631
Support Devices	0.603
Average Score	0.629

Tabelle 20: AUC-ROC-Score der einzelnen Pathologien mit Augmentierungen von small_affine

5.4.2 DenseNet

Der Loss auf dem DenseNet mit affine_color ist in Abbildung 20 zu sehen. Man sieht, dass der Validationloss stark um ca. 0.6934 schwankt. Der Loss auf dem Trainingsset

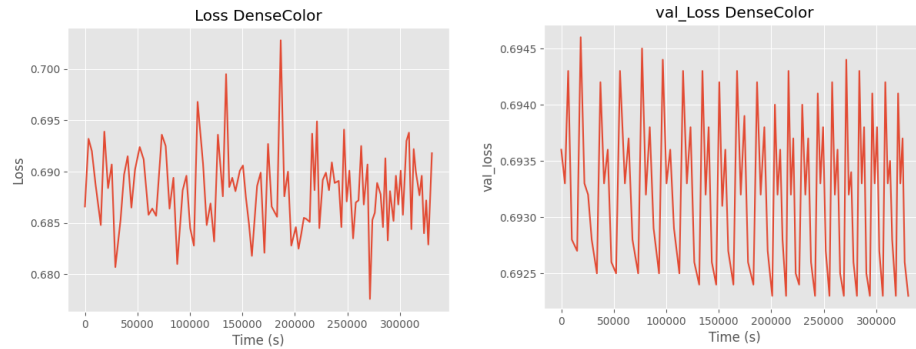


Abbildung 20: Validation- und Trainingsloss auf dem DenseNet mit affine_color

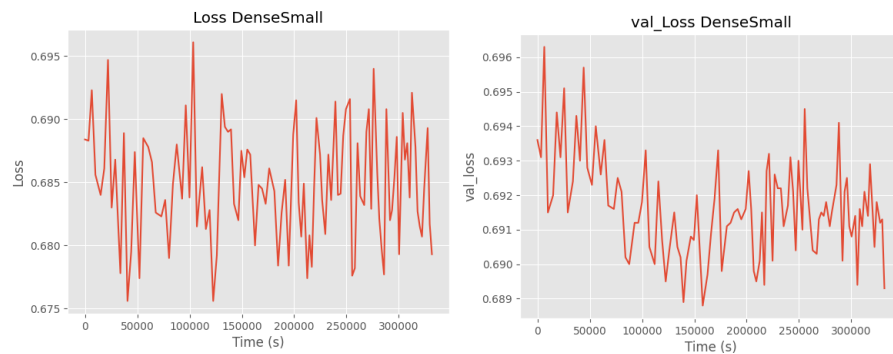


Abbildung 21: Der Trainings- und Validationloss auf dem DenseNet mit small_affine

schwankt auch um den Mittelwert, der in etwa 0.688 beträgt. Somit lässt sich sagen, dass das Netz quasi nicht gelernt hat.

Für den Classifier der mit small_affine trainiert wurde, sieht man das bis in etwa der Hälfte der Berechnungszeit der Validationloss sinkt (Abb. 21). Danach beginnt er wieder leicht zu steigen was darauf schließen lässt, dass auf dem Netz Overfitting auftritt. Der Loss auf dem Trainingsset schwankt auch in etwa um seinen Mittelwert. Dennoch ist ein leichter Abwärtstrend erkennbar.

Das DenseNet mit affine_color schneidet auch in den Metriken schlecht ab (Tab 21). Bei einem Threshold von 0.5 wurden in keiner Klasse True-Positives berechnet. Um trotzdem eine Tabelle erstellen zu können wird hier ein Threshold von 0.2 verwendet. Dadurch erhält man in den Klassen Lung Opacity und Support Devices Ergebnisse für Precision, Recall und $F_1 - Score$. Der Recall dieser beiden Klassen liegt bei eins. Somit wurden bei diesem Threshold keine False-Negatives berechnet.

Auch das mit small_affine trainiert DenseNet liefert schlechte Ergebnisse (Tab 22). Auch hier musste der Threshold auf 0.2 gesetzt werden, damit man in zwei Klassen True-Positives erhält. Damit lassen sich für die Pathologien Pleural Effusion und Support Devices die drei Metriken berechnen. Für Pleural Effusion fallen die Werte sehr niedrig aus. Somit wurden in dieser Klasse trotz des niedrigen Thresholds kein gutes

Ergebnis erzielt. Bei Support Devices ist der Recall gleich eins.

Der durchschnittliche AUC-Score für das DenseNet, das mit `affine_color` trainiert wurde, liegt bei 0.477. Dieser Wert ist niedriger als der AUC-Score für das DenseNet, das ohne Augmentierungen trainiert wurde. Der Classifier in diesem Experiment ist somit der, der am schlechtesten abschneidet unter den DenseNets.

Auch bei `small_affine` ist der Classifier schlechter als das DenseNet, das ohne Augmentierungen trainiert wurde. Der durchschnittliche AUC-Score liegt bei 0.485.

Pathologie	Precision	Recall	$F_1 - Score$
Enlarged Cardimediastinum	0	0	0
Cardiomegaly	0	0	0
Lung Opacity	0.485	1	0.654
Lung Lesion	0	0	0
Edema	0	0	0
Consolidation	0	0	0
Pneumonia	0	0	0
Atelectasis	0	0	0
Pneumothorax	0	0	0
Pleural Effusion	0	0	0
Pleural Other	0	0	0
Fracture	0	0	0
Support Devices	0.509	1	0.675

Tabelle 21: Precision, Recall und $F_1 - Score$ der einzelnen Pathologien mit `affine_color` mit Threshold 0.2 (DenseNet)

Pathologie	Precision	Recall	$F_1 - Score$
Enlarged Cardimediastinum	0	0	0
Cardiomegaly	0	0	0
Lung Opacity	0	0	0
Lung Lesion	0	0	0
Edema	0	0	0
Consolidation	0	0	0
Pneumonia	0	0	0
Atelectasis	0	0	0
Pneumothorax	0	0	0
Pleural Effusion	0.180	0.015	0.028
Pleural Other	0	0	0
Fracture	0	0	0
Support Devices	0.509	1	0.674

Tabelle 22: Precision, Recall und $F_1 - Score$ der einzelnen Pathologien mit `small_affiner` mit Threshold 0.2 (DenseNet)

Pathologie	AUC-ROC-Score
Enlarged Cardi mediastinum	0.493
Cardiomegaly	0.431
Lung Opacity	0.440
Lung Lesion	0.520
Edema	0.432
Consolidation	0.446
Pneumonia	0.488
Atelectasis	0.458
Pneumothorax	0.541
Pleural Effusion	0.419
Pleural Other	0.504
Fracture	0.559
Support Devices	0.469
Average Score	0.477

Tabelle 23: AUC-ROC-Score der einzelnen Pathologien mit Augmentierungen von affine_color (DenseNet)

Pathologie	AUC-ROC-Score
Enlarged Cardi mediastinum	0.504
Cardiomegaly	0.454
Lung Opacity	0.453
Lung Lesion	0.519
Edema	0.418
Consolidation	0.454
Pneumonia	0.476
Atelectasis	0.483
Pneumothorax	0.524
Pleural Effusion	0.452
Pleural Other	0.542
Fracture	0.541
Support Devices	0.490
Average Score	0.485

Tabelle 24: AUC-ROC-Score der einzelnen Pathologien mit Augmentierungen von small_affine (DenseNet)

5.5 Klassifikation mit dem vortrainierten Augmentierungsnetz aus Experiment 3

In den Abbildungen 22 und 23 sieht man einen ähnlich Verlauf der Losses wie im Experiment mit dem Online Bilevel Algorithmus. Bei `small_affine` sieht man wie zum Ende des Trainings der Validationloss zu steigen beginnt, also Overfitting eintritt. Auch der Validationloss von `affine_color` steigt langsam an.

Die drei Metriken Precision, Recall und $F_1 - Score$ schneiden bei einem Threshold von 0.5 mit dem vortrainierten Augmentierungsnetzen schlechter ab als bei der normalen Variante des Online Bilevel Optimization Algorithmus. Bei `small_affine` wurden nur bei zwei Pathologien True-Positives berechnet (Tab 26). Dafür ist der Recall dieser beiden Pathologien recht hoch, bei Support Devices ist er sogar eins, das heißt es wurde kein Wert falsch negativ zugeordnet. Auch bei `affine_color` ist der Recall von Support Devices eins (Tab 27). Jedoch ist diese Pathologie auch die einzige bei der True-Positives für den Threshold 0.5 berechnet wurden.

Setzt man den Threshold auf 0.2 so erhält man bei `small_affine` für fünf Pathologien Werte in den drei Metriken (Tab 26). Bei vier davon ist der Recall gleich eins. Bei Edema ist der Recall 0.0002, der kleinste Wert für einen Recall in allen Experimenten. Die Precision von der Klasse Edema ist eins. Für die fünf Klassen ist der Recall höher und die Precision niedriger bei dem Threshold von 0.2. Auch bei `affine_color` sieht man, dass durch den geringer Threshold mehr Werte berechnet werden konnten (Tab 28) und das für dieselben Pathologien wie für `small_afine`. Der Recall bei der Klasse Edema liegt bei 0.804. Im Gegensatz zu `small_affine` mit einem Wert von 0.0004. Atelectase schneidet hierbei schlechter ab als mit den geometrischen Transformationen. Bei den anderen Klassen ist der Recall wieder gleich eins.

Der durchschnittliche AUC-Score beider Classifier, die mit den vortrainierten Augmentierungsnetzen trainiert wurden, ist höher als der der Classifier, die mit dem Online Bilevel Algorithmus trainiert wurden. Bei der Pathologie Pleural Effusion wird bei beiden Classifiern sogar ein Wert von über 0.8 erreicht. Im Allgemeinen wurden die Pathologien besser erkannt, die auch im Vergleich mit dem Bilevel Augment zu den Standard-Augmentierungen verbessert wurden. Auch `affine_color` hat geholfen, dass der Classifier bessere Ergebnisse berechnet. Man erkennt dennoch leichte Unterschiede zu `small_affine` in den einzelnen Pathologien.

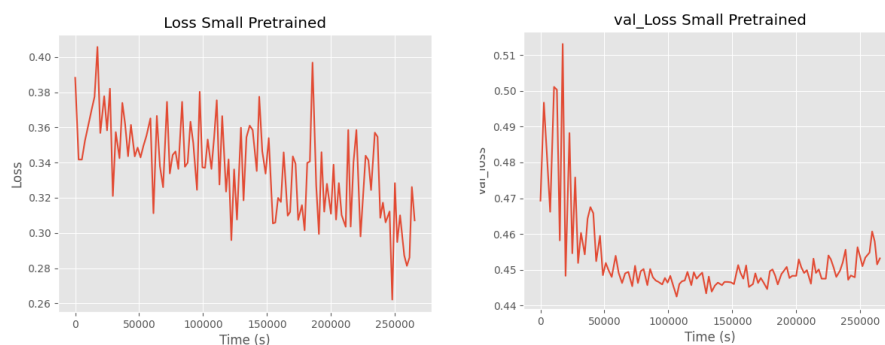


Abbildung 22: Der Trainloss und Validationloss mit dem vortrainierten small_affine

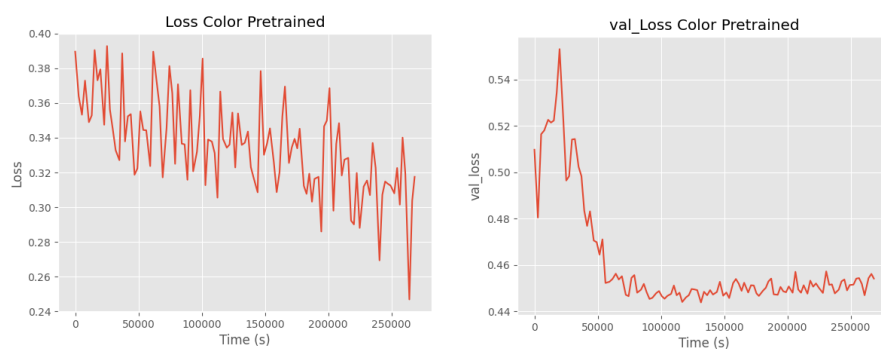


Abbildung 23: Der Trainloss und Validationloss mit dem vortrainierten affine_color

Pathologie	Precision	Recall	$F_1 - Score$
Enlarged Cardimediastinum	0	0	0
Cardiomegaly	0	0	0
Lung Opacity	0.573	0.836	0.680
Lung Lesion	0	0	0
Edema	0	0	0
Consolidation	0	0	0
Pneumonia	0	0	0
Atelectasis	0	0	0
Pneumothorax	0	0	0
Pleural Effusion	0	0	0
Pleural Other	0	0	0
Fracture	0	0	0
Support Devices	0.509	1	0.675

Tabelle 25: Precision, Recall und $F_1 - Score$ der einzelnen Pathologien mit vortrainiertem small_affine und Threshold 0.5

Pathologie	Precision	Recall	$F_1 - Score$
Enlarged Cardimediastinum	0	0	0
Cardiomegaly	0	0	0
Lung Opacity	0.485	1	0.654
Lung Lesion	0	0	0
Edema	1	0.0002	0.0004
Consolidation	0	0	0
Pneumonia	0	0	0
Atelectasis	0.344	1	0.512
Pneumothorax	0	0	0
Pleural Effusion	0.354	1	0.523
Pleural Other	0	0	0
Fracture	0	0	0
Support Devices	0.509	1	0.675

Tabelle 26: Precision, Recall und $F_1 - Score$ der einzelnen Pathologien mit vortrainiertem small_affine und Threshold 0.2

Pathologie	Precision	Recall	$F_1 - Score$
Enlarged Cardimediastinum	0	0	0
Cardiomegaly	0	0	0
Lung Opacity	0	0	0
Lung Lesion	0	0	0
Edema	0	0	0
Consolidation	0	0	0
Pneumonia	0	0	0
Atelectasis	0	0	0
Pneumothorax	0	0	0
Pleural Effusion	0	0	0
Pleural Other	0	0	0
Fracture	0	0	0
Support Devices	0.509	1	0.675

Tabelle 27: Precision, Recall und $F_1 - Score$ der einzelnen Pathologien mit affine_color und Threshold 0.5

Pathologie	Precision	Recall	$F_1 - Score$
Enlarged Cardimediastinum	0	0	0
Cardiomegaly	0	0	0
Lung Opacity	0.485	1	0.654
Lung Lesion	0	0	0
Edema	0.405	0.804	0.539
Consolidation	0	0	0
Pneumonia	0	0	0
Atelectasis	0.504	0.032	0.061
Pneumothorax	0	0	0
Pleural Effusion	0.354	1	0.523
Pleural Other	0	0	0
Fracture	0	0	0
Support Devices	0.509	1	0.675

Tabelle 28: Precision, Recall und $F_1 - Score$ der einzelnen Pathologien mit affine_color und Threshold 0.2

Pathologie	AUC-ROC-Score
Enlarged Cardimediastinum	0.520
Cardiomegaly	0.724
Lung Opacity	0.681
Lung Lesion	0.569
Edema	0.716
Consolidation	0.659
Pneumonia	0.621
Atelectasis	0.639
Pneumothorax	0.626
Pleural Effusion	0.807
Pleural Other	0.509
Fracture	0.535
Support Devices	0.767
Average Score	0.644

Tabelle 29: AUC-ROC-Score der einzelnen Pathologien mit dem vortrainierten Augmenter small_affine

Pathologie	AUC-ROC-Score
Enlarged Cardi mediastinum	0.528
Cardiomegaly	0.715
Lung Opacity	0.681
Lung Lesion	0.552
Edema	0.719
Consolidation	0.664
Pneumonia	0.628
Atelectasis	0.639
Pneumothorax	0.587
Pleural Effusion	0.806
Pleural Other	0.515
Fracture	0.549
Support Devices	0.763
Average Score	0.642

Tabelle 30: AUC-ROC-Score der einzelnen Pathologien mit dem vortrainierten Augmenter affine_color

6 Diskussion

Das Ziel der Arbeit war die Auswirkung des Online Bilevel Optimization Algorithmus von Mounsaveng et al. auf Thorax-Röntgen Bildern zu zeigen. Um einen Vergleich zu ziehen, wurden das ResNET18 und das DenseNet121, zuerst ohne Augmentierung der Daten, danach mit Augmentierungen, inspiriert von der Arbeit von Yuan et al., trainiert. Danach wurde die Bilevel Augment Methode getestet. Die im Bilevel Augment Algorithmus trainierten Augmentierungsnetze für das ResNET18 wurden zusätzlich noch verwendet, um einen eigenen Classifier zu trainieren.

6.1 Zusammenfassung

6.1.1 ResNET18

Das ResNET18 zeigt bei dem ersten Experiment ohne Augmentierungen schnell Anzeichen von Overfitting. Auch bei der Evaluierung des trainierten Classifiers sieht man, dass ohne Augmentierungen das ResNET18 keine guten Werte liefert. Der AUC-Score liegt unter 0.5. Somit ist der trainierte Classifier nicht besser als "Raten". Nutzt man beim Trainieren des Klassifizierungsnetzes Augmentierungen, wie sie auch in der Arbeit von Yuan et al. verwendet wurden, verbessert sich der durchschnittliche Wert des AUC-Scores auf 0.626. Insbesondere Klassen wie Pleural Effusion und Support Devices haben sich stark verbessert im Vergleich zur Klassifizierung ohne Augmentierungen. Bei dem Online Bilevel Optimization Algorithmus gibt es zwei Augmentierungsnetze, small_affine und affine_color. small_affine generiert geometrische Transformationen für Bilder, dieselbe Art von Transformationen, wie sie auch in der zweiten Methode

genutzt wurden. *affine_color* generiert digitale Bildtransformationen z. B. Änderungen am Kontrast, der Helligkeit oder der Sättigung der Bilder. Im Vergleich zum Experiment mit Related Work Augmentierungen schneidet *affine_color* schlechter ab, dennoch besser als der Classifier der ohne Augmentierungen trainiert wurde. Pathologien wie Lung Lesion oder Pneumothorax schneiden schlechter ab als zuvor. *affine_color* erreicht einen durchschnittlichen AUC-Score von 0.611. Der durchschnittliche AUC-Score von *small_affine* ist um 0.003 Punkte besser als der der Related Work Augmentierungen. Der AUC-Score ist 0.629. Die meisten Pathologien, die auch im Vergleich mit dem Classifier ohne Augmentierungen und dem mit Related Work Augmentierungen verbessert wurden, schneiden mit dem *small_affine* Augmenter Netz ebenfalls besser ab als mit Related Work Augmentierungen. Dies trifft nicht auf alle Klassen zu, da sich Support Devices im Vergleich verschlechtert hat. Nutzt man die vortrainierten Augmentierungsnetze aus dem Bilevel Augment Algorithmus so verbessert sich der durchschnittliche AUC-Score auf 0.644 für *small_affine* und für *affine_color* auf 0.642. Mit beiden Augmentierungsnetzen haben sich die Pathologien erneut verbessert, die sich in den Schritten zuvor schon verbessert haben, wie z.B. Pleural Effusion oder Cardiomegaly.

6.1.2 DensNet

Aus zeitlichen Gründen konnten die Experimente mit dem DenseNet nicht in der Ausführlichkeit wie mit dem ResNET durchgeführt werden. Da der Online Bilevel Optimization Algorithmus sehr zeitintensiv ist, wurde der Ansatz vier mit den vortrainierten Augmenternetzen nicht mit dem DenseNet durchgeführt. Das DenseNet, das als Classifier ohne Augmentierungen trainiert wurde, zeigte nach 25 Epochen in etwa dasselbe Niveau wie das ResNET18 mit dem minimalen Validationloss trotz der geringeren Auflösung. Der AUC-Score des DenseNets ist mit 0.494 um 0.004 Punkt kleiner als der des ResNETs. Somit auch nicht besser als "Raten". Verwendet man die Augmentierungen inspiriert durch Yuan et al. mit dem DenseNet, sieht man auch hier wieder eine Verbesserung. Diese Verbesserung ist nicht so hoch wie die des ResNETs. Der DenseNet-Classifier mit Related Work Augmentierungen erreicht nur einen durchschnittlichen AUC-Score von 0.553. Dennoch sind in Klassen wie Support Devices und Pleural Effusion deutliche Steigerungen im AUC-Score zu sehen. Mit diesem durchschnittlichen AUC-Score ist der Classifier dennoch nur marginal besser als "Raten". Das DenseNet schneidet mit dem Bilevel Augment Algorithmus schlechter ab als die beiden Ansätze davor. Somit ist für das DenseNet der Ansatz der Online Bilevel Optimization in dieser Form unbrauchbar. Der Durchschnitt des AUC-Scores liegt für *small_affine* bei 0.485 und bei *affine_color* bei 0.477.

Versuch	Netz	$\emptyset AUC - Score$
1	ResNET18	0.498
1	DenseNet	0.494
2	ResNET18	0.626
2	DenseNet	0.533
3	ResNET + <i>affine_color</i>	0.611
3	ResNET + <i>small_affine</i>	0.629
3	DenseNet + <i>affine_color</i>	0.477
3	ResNET + <i>small_affine</i>	0.485
4	ResNET + vortrainiertes <i>affine_color</i>	0.642
4	ResNET + vortrainiertes <i>small_affine</i>	0.644

Tabelle 31: Die durchschnittlichen AUC-Scores der einzelnen Methoden

6.2 Interpretation

Am effektivsten haben sich die Augmentierungen gezeigt, die aus geometrischen Transformationen der Bilder bestehen. Das sieht man daran, dass sich der AUC-Score des Classifiers von *small_affine* in derselben Größenordnung wie der der Augmentierungen von Yuan et al. befindet. Auch der Classifier, der mit den vortrainierten Augmentierungsnetzen trainiert wurde, besitzt in *small_affine* einen höheren Wert als der mit *affine_color* trainierte Classifier. Die Verbesserungen im Gegensatz zu dem Netz ohne Augmentierungen sind vor allem bei Pathologien zu sehen, die mit Ausdehnungen, Vergrößerungen, kollabierten Organen usw. im Brustkorb einhergehen. Wie zum Beispiel Kardiomegalie, bei der es sich um eine Vergrößerung des Herzmuskels handelt[34] oder bei Atelektase, einem kollabierten Lungenflügel[33]. Durch die geometrischen Augmentierungen wird es für das Klasifizierungsnetz einfacher solche Muster zu erkennen und zu erlernen.

Dadurch, dass *affine_color* keine geometrischen Transformationen durchführt, sondern digitale Bildtransformationen wie Kontrast und Helligkeit, ist es für den Classifier schwieriger, Muster zu erkennen. Krankheiten wie eine Pneumonie sind in Röntgenbildern als sogenannte Verschattungen, also Aufhellungen sichtbar[19]. Ändert man nun die Helligkeit o. Ä. auf den Röntgenbildern, so können Stellen als Verschattungen erscheinen, die eigentlich keine sind, sodass die Augmentierungen den Classifier aktiv behindern und das Lernen der Eigenschaften erschweren.

Dass der Bilevel Augment Algorithmus, für das DenseNet, die schlechtesten Ergebnisse aufzeigt, kann daran liegen, dass die beiden Augmentierungsnetze angepasst für die ResNET Architektur sind. Mounsaveng et al. nutzen in ihren Experimenten ausschließlich ResNET Architekturen.

Schaut man sich nur den Durchschnitt der AUC-Werte an, so lohnt sich der zeitliche Mehraufwand für den Online Bilevel Optimization Algorithmus nicht. Begrenzt man die Evaluierung aber auf die Pathologien, die sich im Verlauf der vier Versuche am deutlichsten verbessert haben, so sieht man, dass der Bilevel Augment Algorithmus die

Erkennung dieser verbessert hat. Somit ist der Online Bilevel Optimization Algorithmus nicht für alle Pathologien, die im CheXpert Datensatz gegeben sind, anwendbar. Pathologie, die sich verbessert haben wie Pleural Effusion oder Cardiomegaly, hingegen zeigen starke Verbesserungen mit der Methode von Mousaveng et al. Für diese Befunde hat der Algorithmus Anwendung. Die vortrainierten Augmentierungsnetze haben beide besser abgeschnitten als die Bilevel Augment Methode. Somit lohnt es sich, die trainierten Augmenternetze zu Speicher und wieder zu verwenden, da dies zeitlich effizienter ist als jedes Mal den Online Bilevel Optimization Algorithmus laufen zu lassen.

6.3 Grenzen der Arbeit

Das Ziel der Arbeit war es, die verschiedenen Augmentierungsansätze zu vergleichen und zu prüfen, ob der Online Bilevel Optimization Algorithmus sinnvoll im Vergleich zu herkömmlichen Augmentierungsmethoden ist. Die Klassifizierungsnetze zu Optimieren hätte nicht in den zeitlichen Rahmen der Arbeit gepasst. Aus diesem Grund sind die Netze nicht gut trainiert und für eine Anwendung ungeeignet. Eine Beschränkung dabei war die Berechnungszeit der Methode. Dadurch, dass zwei neuronale Netze gleichzeitig trainiert wurden, dauerten die Berechnungen deutlich länger. Im Vergleich brauchten die Methode ohne Augmentierungen und die mit den Related Work Augmentierungen etwa 3.5 Tage für 100 Epochen. Der Online Bilevel Optimization Algorithmus brauchte für 25 Epochen in etwa drei Tage mit dem ResNET18 als Classifier. Für die Klassifizierungsnetze wurden nur kleine, schnell zu trainierende Netze gewählt. Somit ist die Leistung der Classifier von vornherein limitiert gewesen.

6.4 Weiterführende Forschung

Als zukünftige Arbeit kann diese Forschung auf andere medizinische Datensätze erweitert werden. Ein Beispiel wäre zum einen wie in der Arbeit von Yuan et al., die ihren Ansatz auch auf weiteren medizinischen Datensätzen getestet haben[38], eine Anwendung auf dem SIIM-ISIC Melanoma Classification⁶ Datensatz von Kaggle der für die Erkennung von Melanomen genutzt werden kann. Des Weiteren kann diese Studie auch auf anderen Netzen angewendet werden, in dieser Arbeit wurden nur das ResNET18 und das DenseNet verwendet, Letzteres aus Zeitgründen nicht in der Ausführlichkeit wie das ResNET18. Weiter können noch die Hyperparameter wie Lernrate, Batch-Size und die Auflösung noch Angepasst werden. Außerdem könnte man noch die mit dem ResNET trainierten Augmenter wie in Versuch vier verwenden können, um ein anderes Klassifizierungsnetz wie das DenseNet zu trainieren und im Vergleich zur hier gezeigten Methode evaluieren. Auch kann man den Algorithmus auf Daten anwenden, die

⁶<https://www.kaggle.com/c/siim-isic-melanoma-classification/overview>

hauptsächlich die Pathologien enthalten, die in dieser Arbeit die höchsten AUC-Scores erreicht haben, wie Pleural Effusion und Cardiomegaly.

7 Fazit

In dieser Arbeit wurden vier verschiedene Ansätze zu Bildklassifizierung auf Thorax-Röntgenbildern mit verschiedenen Augmentierungsmethoden gezeigt. Die vier Methoden bestanden daraus einen Classifier ohne Augmentierungen, mit Standard-Augmentierungen, mit der Online Bilevel Optimization Methode und mit den trainierten Augmentierungsnetzen aus den Bilevel Algorithmus zu trainieren. Die ersten beiden Versuchen waren dazu da, um einen Vergleich für den Online Bilevel Optimization Algorithmus ziehen zu können. Die Methode von Mounsaveng et al. hilft dabei für bestimmte Pathologien optimale Augmentierungen zu finden. Die Methode mit dem ResNET18 erreicht das Niveau des Classifiers mit Standard-Augmentierungen. Auch kann man die vortrainierten Augmenternetze weiter verwenden, um für den Datensatz optimale Augmentierungen zu generieren. Dadurch, dass die trainierten Augmenter weiter genutzt werden können und einen besseren AUC-Score erzielen als der Online Bilevel Optimization Algorithmus ist es trotzdem nützlich, die Methode von Mounsaveng et al. zu verwenden, obwohl diese mit einem zeitlichen Mehraufwand einhergeht. Hat man die Augmenter Netze, die optimale Augmentierungen für den Datensatz erzeugen, muss man die Methode von Mounsaveng et al. nicht nochmals ausführen und spart sich somit Zeit und erhält dadurch Augmentierungen die zu einem besseren Ergebnis als mit Standard-Augmentierungen führen.

Literaturverzeichnis

- [1] Antreas Antoniou, Amos Storkey, and Harrison Edwards. Data augmentation generative adversarial networks, 2018.
- [2] Benjamin Aunkofer. Fehler-rückführung mit der backpropagation, 2019. [Online; Stand 17. Dezember 2021].
- [3] Holger Aust. *Das Gehirn kopieren? – Künstliche neuronale Netze*, pages 174 – 181. Springer Berlin Heidelberg, Berlin, Heidelberg, 2021.
- [4] Aniruddha Bhandari. Auc-roc curve in machine learning clearly explained, 16. Juni 2016.
- [5] Anne Bonner. The complete beginner’s guide to deep learning: Convolutional neural networks and image classification, 2019. [Online; Stand 14. Dezember 2021].
- [6] Jason Brownlee. A gentle introduction to transfer learning for deep learning, 2017. [Online; Stand 14. Dezember 2021].
- [7] Jason Brownlee. Overfitting and underfitting with machine learning algorithms, 21.03.2016.
- [8] Vitaly Bushaev. Adam — latest trends in deep learning optimization, 2018. [Online; Stand 17. Dezember 2021].
- [9] Antonia Creswell, Tom White, Vincent Dumoulin, Kai Arulkumaran, Biswa Sengupta, and Anil A. Bharath. Generative adversarial networks: An overview. *IEEE Signal Processing Magazine*, 35(1):53–65, 2018.
- [10] Ekin D. Cubuk, Barret Zoph, Dandelion Mane, Vijay Vasudevan, and Quoc V. Le. Autoaugment: Learning augmentation strategies from data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.
- [11] DeepAI. What is a feed forward neural network? [Online; Stand 14. Dezember 2021].
- [12] deepchecks. Top-1 error rate. [Online; Stand 17. Dezember 2021].
- [13] DeepMind. Where we began.
- [14] Hironobu Fujiiyoshi, Tsubasa Hirakawa, and Takayoshi Yamashita. Deep learning-based image recognition for autonomous driving. *IATSS Research*, 43(4):244–252, 2019.
- [15] Cody Glickman. Data augmentation in medical images, 12.10.2020.
- [16] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition, 2015.
- [17] Benlin Hu, Cheng Lei, Dong Wang, Shu Zhang, and Zhenyu Chen. A preliminary study on data augmentation of deep learning for image classification, 2019.

- [18] Gao Huang, Zhuang Liu, Laurens van der Maaten, and Kilian Q. Weinberger. Densely connected convolutional networks, 2018.
- [19] Lungenärztl. im Netz. Lungenentzündung - untersuchungsmethoden. [Online; Stand 10. Dezember 2021].
- [20] Jeremy Irvin, Pranav Rajpurkar, Michael Ko, Yifan Yu, Silvana Ciurea-Ilcus, Chris Chute, Henrik Marklund, Behzad Haghgoo, Robyn L. Ball, Katie S. Shpan-skaya, Jayne Seekins, David A. Mong, Safwan S. Halabi, Jesse K. Sandberg, Ricky Jones, David B. Larson, Curtis P. Langlotz, Bhavik N. Patel, Matthew P. Lungren, and Andrew Y. Ng. Chexpert: A large chest radiograph dataset with uncertainty labels and expert comparison. *CoRR*, abs/1901.07031, 2019.
- [21] Nobuaki Kimura, Ikuo Yoshinaga, Kenji Sekijima, Issaku Azechi, and Daichi Ba-ba. Convolutional neural network coupled with a transfer-learning approach for time-series flood predictions. *Water*, 12:96, 12 2019.
- [22] Till Köhler. Deep learning in der krebsdiagnostik- chancen überstrahlen die risiken. 2018.
- [23] Nature Materials. Ascent of machine learning in medicine. 18. April 2019.
- [24] Mohammad Momeny, Ali Mohammad Latif, Mehdi Agha Sarram, Razieh Sheikhpour, and Yu Dong Zhang. A noise robust convolutional neural network for image classification. *Results in Engineering*, 10:100225, 2021.
- [25] Saypraseuth Mounsaveng, Issam Laradji, Ismail Ben Ayed, David Vazquez, and Marco Pedersoli. Learning data augmentation with online bilevel optimization for image classification. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 1691–1700, January 2021.
- [26] Computer Vision Online. Deepface: Facebook’s face verification algorithm. [Online; Stand 14. Dezember 2021].
- [27] Nina Schaaf. Neuronale netze: Ein blick in die black box, 14. Januar 2020.
- [28] Joshua Sewell. Four challenges in developing ai algorithms for medical imaging, 10.05.2019.
- [29] Connor Shorten and Taghi M. Khoshgoftaar. A survey on image data augmentation for deep learning. *Journal of Big Data*, 6, 2019.
- [30] Koo Ping Shung. Accuracy, precision, recall or f1?, 15. März 2018.
- [31] Chiranjibi Sitaula and Mohammad Belayet Hossain. Attention-based vgg-16 model for covid-19 chest x-ray image classification. *Applied Intelligence*, 2021.
- [32] Jost Tobias Springenberg. Unsupervised and semi-supervised learning with categorical generative adversarial networks, 2016.
- [33] Wikipedia. Atelektase — wikipedia, die freie enzyklopädie, 2021. [Online; Stand 10. Dezember 2021].
- [34] Wikipedia. Kardiomegalie — wikipedia, die freie enzyklopädie, 2021. [Online; Stand 10. Dezember 2021].

-
- [35] Wikipedia contributors. Confusion matrix — Wikipedia, the free encyclopedia, 2021. [Online; accessed 1-December-2021].
- [36] Wikipedia contributors. Receiver operating characteristic — Wikipedia, the free encyclopedia, 2021. [Online; accessed 1-December-2021].
- [37] Zikun Yang, Paul Bogdan, and Shahin Nazarian. An in silico deep learning approach to multi-epitope vaccine design: a sars-cov-2 case study. *Scientific Reports*, 11, 2021.
- [38] Zhuoning Yuan, Yan Yan, Milan Sonka, and Tianbao Yang. Large-scale robust deep auc maximization: A new surrogate loss and empirical studies on medical image classification. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021.

Anhang

Pathologie	Confusion Matrix	
Enlarged Cardiomediastinum	17024	0
	2100	0
Cardiomegaly	16462	0
	2662	0
Lung Opacity	0	9841
	0	9283
Lung Lesion	18263	0
	861	0
Edema	13505	0
	5619	0
Consolidation	15481	0
	3643	0
Pneumonia	17082	0
	2042	0
Atelectasis	12543	0
	6581	0
Pneumothorax	17685	0
	1439	0
Pleural Effusion	11985	364
	6564	211
Pleural Other	18810	0
	314	0
Fracture	18244	0
	880	0
Support Devices	0	9388
	0	9736

Tabelle 32: Confusion Matrizen der einzelnen Pathologien ohne Augmentierungen und Threshold 0.5 (ResNET18)

Pathologie	Confusion Matrix				
Enlarged Cardimediastinum	<table border="1"> <tr> <td>17024</td> <td>0</td> </tr> <tr> <td>2100</td> <td>0</td> </tr> </table>	17024	0	2100	0
17024	0				
2100	0				
Cardiomegaly	<table border="1"> <tr> <td>16462</td> <td>0</td> </tr> <tr> <td>2662</td> <td>0</td> </tr> </table>	16462	0	2662	0
16462	0				
2662	0				
Lung Opacity	<table border="1"> <tr> <td>0</td> <td>9841</td> </tr> <tr> <td>0</td> <td>9283</td> </tr> </table>	0	9841	0	9283
0	9841				
0	9283				
Lung Lesion	<table border="1"> <tr> <td>18263</td> <td>0</td> </tr> <tr> <td>861</td> <td>0</td> </tr> </table>	18263	0	861	0
18263	0				
861	0				
Edema	<table border="1"> <tr> <td>6938</td> <td>6567</td> </tr> <tr> <td>2326</td> <td>3293</td> </tr> </table>	6938	6567	2326	3293
6938	6567				
2326	3293				
Consolidation	<table border="1"> <tr> <td>9014</td> <td>6467</td> </tr> <tr> <td>1842</td> <td>1801</td> </tr> </table>	9014	6467	1842	1801
9014	6467				
1842	1801				
Pneumonia	<table border="1"> <tr> <td>17082</td> <td>0</td> </tr> <tr> <td>2042</td> <td>0</td> </tr> </table>	17082	0	2042	0
17082	0				
2042	0				
Atelectasis	<table border="1"> <tr> <td>0</td> <td>12543</td> </tr> <tr> <td>0</td> <td>6581</td> </tr> </table>	0	12543	0	6581
0	12543				
0	6581				
Pneumothorax	<table border="1"> <tr> <td>17685</td> <td>0</td> </tr> <tr> <td>1439</td> <td>0</td> </tr> </table>	17685	0	1439	0
17685	0				
1439	0				
Pleural Effusion	<table border="1"> <tr> <td>0</td> <td>12349</td> </tr> <tr> <td>0</td> <td>6775</td> </tr> </table>	0	12349	0	6775
0	12349				
0	6775				
Pleural Other	<table border="1"> <tr> <td>18810</td> <td>0</td> </tr> <tr> <td>314</td> <td>0</td> </tr> </table>	18810	0	314	0
18810	0				
314	0				
Fracture	<table border="1"> <tr> <td>18244</td> <td>0</td> </tr> <tr> <td>880</td> <td>0</td> </tr> </table>	18244	0	880	0
18244	0				
880	0				
Support Devices	<table border="1"> <tr> <td>0</td> <td>9388</td> </tr> <tr> <td>0</td> <td>9736</td> </tr> </table>	0	9388	0	9736
0	9388				
0	9736				

Tabelle 33: Confusion Matrizen der einzelnen Pathologien ohne Augmentierungen und Threshold 0.2 (ResNET18)

Pathologie	Confusion Matrix	
Enlarged Cardiomeastinum	17117	0
	2107	0
Cardiomegaly	16537	0
	2687	0
Lung Opacity	9882	0
	9342	0
Lung Lesion	18360	0
	864	0
Edema	13579	0
	5645	0
Consolidation	15558	0
	3666	0
Pneumonia	17171	0
	2053	0
Atelectasis	12613	0
	6611	0
Pneumothorax	17773	0
	1451	0
Pleural Effusion	12415	0
	6809	0
Pleural Other	18910	0
	314	0
Fracture	18341	0
	883	0
Support Devices	42	9385
	58	9739

Tabelle 34: Confusion Matrizen der einzelnen Pathologien ohne Augmentierungen und Threshold 0.5 (DenseNet)

Pathologie	Confusion Matrix	
Enlarged Cardiomeastinum	17024	0
	2100	0
Cardiomegaly	16462	0
	2662	0
Lung Opacity	9841	0
	9283	0
Lung Lesion	18263	0
	861	0
Edema	13505	0
	5619	0
Consolidation	15481	0
	3643	0
Pneumonia	17082	0
	2042	0
Atelectasis	12543	0
	6581	0
Pneumothorax	17685	0
	1439	0
Pleural Effusion	12349	0
	6775	0
Pleural Other	18810	0
	314	0
Fracture	18244	0
	880	0
Support Devices	0	9388
	0	9736

Tabelle 35: Confusion Matrizen der einzelnen Pathologien ohne Augmentierungen und Threshold 0.2 (DenseNet)

Pathologie	Confusion Matrix	
Enlarged Cardimediastinum	17024	0
	2100	0
Cardiomegaly	16462	0
	2662	0
Lung Opacity	7069	2772
	5264	4019
Lung Lesion	18263	0
	861	0
Edema	13500	5
	5610	9
Consolidation	15481	0
	3643	0
Pneumonia	17082	0
	2042	0
Atelectasis	12542	1
	6581	0
Pneumothorax	17685	0
	1439	0
Pleural Effusion	7875	4474
	2472	4303
Pleural Other	18810	0
	314	0
Fracture	18244	0
	880	0
Support Devices	1507	7881
	554	9182

Tabelle 36: Confusion Matrizen der einzelnen Pathologien mit related Work Augmentierungen und Threshold 0.5 (ResNET18)

Pathologie	Confusion Matrix				
Enlarged Cardimediastinum	<table border="1"> <tr><td>16998</td><td>26</td></tr> <tr><td>2096</td><td>4</td></tr> </table>	16998	26	2096	4
16998	26				
2096	4				
Cardiomegaly	<table border="1"> <tr><td>16458</td><td>4</td></tr> <tr><td>2659</td><td>3</td></tr> </table>	16458	4	2659	3
16458	4				
2659	3				
Lung Opacity	<table border="1"> <tr><td>0</td><td>9841</td></tr> <tr><td>1</td><td>9282</td></tr> </table>	0	9841	1	9282
0	9841				
1	9282				
Lung Lesion	<table border="1"> <tr><td>18263</td><td>0</td></tr> <tr><td>861</td><td>0</td></tr> </table>	18263	0	861	0
18263	0				
861	0				
Edema	<table border="1"> <tr><td>4429</td><td>9076</td></tr> <tr><td>817</td><td>4802</td></tr> </table>	4429	9076	817	4802
4429	9076				
817	4802				
Consolidation	<table border="1"> <tr><td>7137</td><td>8344</td></tr> <tr><td>1055</td><td>2588</td></tr> </table>	7137	8344	1055	2588
7137	8344				
1055	2588				
Pneumonia	<table border="1"> <tr><td>17073</td><td>9</td></tr> <tr><td>2032</td><td>10</td></tr> </table>	17073	9	2032	10
17073	9				
2032	10				
Atelectasis	<table border="1"> <tr><td>815</td><td>11728</td></tr> <tr><td>112</td><td>6469</td></tr> </table>	815	11728	112	6469
815	11728				
112	6469				
Pneumothorax	<table border="1"> <tr><td>17663</td><td>22</td></tr> <tr><td>1422</td><td>17</td></tr> </table>	17663	22	1422	17
17663	22				
1422	17				
Pleural Effusion	<table border="1"> <tr><td>165</td><td>12184</td></tr> <tr><td>14</td><td>6761</td></tr> </table>	165	12184	14	6761
165	12184				
14	6761				
Pleural Other	<table border="1"> <tr><td>18810</td><td>0</td></tr> <tr><td>314</td><td>0</td></tr> </table>	18810	0	314	0
18810	0				
314	0				
Fracture	<table border="1"> <tr><td>18244</td><td>0</td></tr> <tr><td>880</td><td>0</td></tr> </table>	18244	0	880	0
18244	0				
880	0				
Support Devices	<table border="1"> <tr><td>1</td><td>9387</td></tr> <tr><td>0</td><td>9736</td></tr> </table>	1	9387	0	9736
1	9387				
0	9736				

Tabelle 37: Confusion Matrizen der einzelnen Pathologien mit related Work Augmentierungen und Threshold 0.2 (ResNET18)

Pathologie	Confusion Matrix	
Enlarged Cardimediastinum	17024	0
	2100	0
Cardiomegaly	16462	0
	2662	0
Lung Opacity	7254	2587
	5459	3824
Lung Lesion	18263	0
	861	0
Edema	13505	0
	5619	0
Consolidation	15481	0
	3643	0
Pneumonia	17082	0
	2042	0
Atelectasis	12543	0
	6581	0
Pneumothorax	17685	0
	1439	0
Pleural Effusion	9919	2430
	4420	2355
Pleural Other	18810	0
	314	0
Fracture	18244	0
	880	0
Support Devices	7535	1853
	5424	4312

Tabelle 38: Confusion Matrizen der einzelnen Pathologien mit related Work Augmentierungen und Threshold 0.5(DenseNet)

Pathologie	Confusion Matrix	
Enlarged Cardiomediatinum	17024	0
	2100	0
Cardiomegaly	16462	0
	2662	0
Lung Opacity	4353	5488
	2548	6735
Lung Lesion	18263	0
	861	0
Edema	13505	0
	5619	0
Consolidation	15481	0
	3643	0
Pneumonia	17082	0
	2042	0
Atelectasis	12543	0
	6581	0
Pneumothorax	17685	0
	1439	0
Pleural Effusion	7478	4871
	2701	4074
Pleural Other	18810	0
	314	0
Fracture	18244	0
	880	0
Support Devices	6064	3324
	3710	6026

Tabelle 39: Confusion Matrizen der einzelnen Pathologien mit related Work Augmentierungen und Threshold 0.2(DenseNet)

Pathologie	Confusion Matrix	
Enlarged Cardiomeastinum	17024	0
	2100	0
Cardiomegaly	16462	0
	2662	0
Lung Opacity	9841	0
	9283	0
Lung Lesion	18263	0
	861	0
Edema	13505	0
	5619	0
Consolidation	15481	0
	3643	0
Pneumonia	17082	0
	2042	0
Atelectasis	12543	0
	6581	0
Pneumothorax	17685	0
	1439	0
Pleural Effusion	12349	0
	6775	0
Pleural Other	18810	0
	314	0
Fracture	18244	0
	880	0
Support Devices	6641	2747
	6119	3617

Tabelle 40: Confusion Matrizen der einzelnen Pathologien mit affine_color und Threshold 0.5 (ResNET)

Pathologie	Confusion Matrix	
Enlarged Cardimediastinum	17024	0
	2100	0
Cardiomegaly	16462	0
	2662	0
Lung Opacity	9548	293
	8752	531
Lung Lesion	18263	0
	861	0
Edema	13505	0
	5619	0
Consolidation	15481	0
	3643	0
Pneumonia	17082	0
	2042	0
Atelectasis	12543	0
	6581	0
Pneumothorax	17685	0
	1439	0
Pleural Effusion	12347	2
	6773	2
Pleural Other	18810	0
	314	0
Fracture	18244	0
	880	0
Support Devices	0	9388
	0	9736

Tabelle 41: Confusion Matrizen der einzelnen Pathologien mit affine_color und Threshold 0.2 (ResNET)

Pathologie	Confusion Matrix	
Enlarged Cardimediastinum	17024	0
	2100	0
Cardiomegaly	16098	364
	2319	343
Lung Opacity	8035	1806
	5914	3369
Lung Lesion	18263	0
	861	0
Edema	13160	345
	5133	486
Consolidation	15481	0
	3643	0
Pneumonia	17082	0
	2042	0
Atelectasis	12274	269
	6292	289
Pneumothorax	17685	0
	1439	0
Pleural Effusion	11387	962
	4402	2373
Pleural Other	18810	0
	314	0
Fracture	18244	0
	880	0
Support Devices	6037	3351
	4811	4925

Tabelle 42: Confusion Matrizen der einzelnen Pathologien mit small_affine und Threshold 0.5 (ResNET)

Pathologie	Confusion Matrix	
Enlarged Cardimediastinum	17024	0
	2100	0
Cardiomegaly	15786	676
	2144	518
Lung Opacity	7289	2552
	4829	4454
Lung Lesion	18263	0
	861	0
Edema	12851	654
	4772	847
Consolidation	15480	1
	3642	1
Pneumonia	17082	0
	2042	0
Atelectasis	11814	729
	5835	746
Pneumothorax	17685	0
	1439	0
Pleural Effusion	11114	1235
	3985	2790
Pleural Other	18810	0
	314	0
Fracture	18244	0
	880	0
Support Devices	4741	4647
	3480	6256

Tabelle 43: Confusion Matrizen der einzelnen Pathologien mit small_affine und Threshold 0.2 (ResNET)

Pathologie	Confusion Matrix				
Enlarged Cardimediastinum	<table border="1"> <tr><td>17024</td><td>0</td></tr> <tr><td>2100</td><td>0</td></tr> </table>	17024	0	2100	0
17024	0				
2100	0				
Cardiomegaly	<table border="1"> <tr><td>16462</td><td>0</td></tr> <tr><td>2662</td><td>0</td></tr> </table>	16462	0	2662	0
16462	0				
2662	0				
Lung Opacity	<table border="1"> <tr><td>0</td><td>9841</td></tr> <tr><td>0</td><td>9283</td></tr> </table>	0	9841	0	9283
0	9841				
0	9283				
Lung Lesion	<table border="1"> <tr><td>18263</td><td>0</td></tr> <tr><td>861</td><td>0</td></tr> </table>	18263	0	861	0
18263	0				
861	0				
Edema	<table border="1"> <tr><td>13505</td><td>0</td></tr> <tr><td>5619</td><td>0</td></tr> </table>	13505	0	5619	0
13505	0				
5619	0				
Consolidation	<table border="1"> <tr><td>15481</td><td>0</td></tr> <tr><td>3643</td><td>0</td></tr> </table>	15481	0	3643	0
15481	0				
3643	0				
Pneumonia	<table border="1"> <tr><td>17082</td><td>0</td></tr> <tr><td>2042</td><td>0</td></tr> </table>	17082	0	2042	0
17082	0				
2042	0				
Atelectasis	<table border="1"> <tr><td>12543</td><td>0</td></tr> <tr><td>6581</td><td>0</td></tr> </table>	12543	0	6581	0
12543	0				
6581	0				
Pneumothorax	<table border="1"> <tr><td>17685</td><td>0</td></tr> <tr><td>1439</td><td>0</td></tr> </table>	17685	0	1439	0
17685	0				
1439	0				
Pleural Effusion	<table border="1"> <tr><td>12349</td><td>0</td></tr> <tr><td>6775</td><td>0</td></tr> </table>	12349	0	6775	0
12349	0				
6775	0				
Pleural Other	<table border="1"> <tr><td>18810</td><td>0</td></tr> <tr><td>314</td><td>0</td></tr> </table>	18810	0	314	0
18810	0				
314	0				
Fracture	<table border="1"> <tr><td>18244</td><td>0</td></tr> <tr><td>880</td><td>0</td></tr> </table>	18244	0	880	0
18244	0				
880	0				
Support Devices	<table border="1"> <tr><td>0</td><td>9388</td></tr> <tr><td>0</td><td>9736</td></tr> </table>	0	9388	0	9736
0	9388				
0	9736				

Tabelle 44: Confusion Matrizen der einzelnen Pathologien mit affine_color und Threshold 0.2 (DenseNet)

Pathologie	Confusion Matrix	
Enlarged Cardiomedastinum	17024	0
	2100	0
Cardiomegaly	16462	0
	2662	0
Lung Opacity	9841	0
	9283	0
Lung Lesion	18263	0
	861	0
Edema	13505	0
	5619	0
Consolidation	15481	0
	3643	0
Pneumonia	17082	0
	2042	0
Atelectasis	12543	0
	6581	0
Pneumothorax	17685	0
	1439	0
Pleural Effusion	11871	478
	6670	105
Pleural Other	18810	0
	314	0
Fracture	18244	0
	880	0
Support Devices	0	9388
	0	9736

Tabelle 45: Confusion Matrizen der einzelnen Pathologien mit small_affine und Threshold 0.2 (DenseNet)

Pathologie	Confusion Matrix	
Enlarged Cardiomeastinum	17024	0
	2100	0
Cardiomegaly	16462	0
	2662	0
Lung Opacity	4049	5792
	1524	7759
Lung Lesion	18263	0
	861	0
Edema	13505	0
	5619	0
Consolidation	15481	0
	3643	0
Pneumonia	17082	0
	2042	0
Atelectasis	12543	0
	6581	0
Pneumothorax	17685	0
	1439	0
Pleural Effusion	12349	0
	6775	0
Pleural Other	18810	0
	314	0
Fracture	18244	0
	880	0
Support Devices	0	9388
	0	9736

Tabelle 46: Confusion Matrizen der einzelnen Pathologien mit vortrainiertem small_affine und Threshold 0.5 (ResNET)

Pathologie	Confusion Matrix				
Enlarged Cardiomeastinum	<table border="1"> <tr><td>17024</td><td>0</td></tr> <tr><td>2100</td><td>0</td></tr> </table>	17024	0	2100	0
17024	0				
2100	0				
Cardiomegaly	<table border="1"> <tr><td>16462</td><td>0</td></tr> <tr><td>2662</td><td>0</td></tr> </table>	16462	0	2662	0
16462	0				
2662	0				
Lung Opacity	<table border="1"> <tr><td>0</td><td>9841</td></tr> <tr><td>0</td><td>9283</td></tr> </table>	0	9841	0	9283
0	9841				
0	9283				
Lung Lesion	<table border="1"> <tr><td>18263</td><td>0</td></tr> <tr><td>861</td><td>0</td></tr> </table>	18263	0	861	0
18263	0				
861	0				
Edema	<table border="1"> <tr><td>13505</td><td>0</td></tr> <tr><td>5619</td><td>0</td></tr> </table>	13505	0	5619	0
13505	0				
5619	0				
Consolidation	<table border="1"> <tr><td>15481</td><td>0</td></tr> <tr><td>3643</td><td>0</td></tr> </table>	15481	0	3643	0
15481	0				
3643	0				
Pneumonia	<table border="1"> <tr><td>17082</td><td>0</td></tr> <tr><td>2042</td><td>0</td></tr> </table>	17082	0	2042	0
17082	0				
2042	0				
Atelectasis	<table border="1"> <tr><td>0</td><td>12543</td></tr> <tr><td>0</td><td>6581</td></tr> </table>	0	12543	0	6581
0	12543				
0	6581				
Pneumothorax	<table border="1"> <tr><td>17685</td><td>0</td></tr> <tr><td>1439</td><td>0</td></tr> </table>	17685	0	1439	0
17685	0				
1439	0				
Pleural Effusion	<table border="1"> <tr><td>0</td><td>12349</td></tr> <tr><td>0</td><td>6775</td></tr> </table>	0	12349	0	6775
0	12349				
0	6775				
Pleural Other	<table border="1"> <tr><td>18810</td><td>0</td></tr> <tr><td>314</td><td>0</td></tr> </table>	18810	0	314	0
18810	0				
314	0				
Fracture	<table border="1"> <tr><td>18244</td><td>0</td></tr> <tr><td>880</td><td>0</td></tr> </table>	18244	0	880	0
18244	0				
880	0				
Support Devices	<table border="1"> <tr><td>0</td><td>9388</td></tr> <tr><td>0</td><td>9736</td></tr> </table>	0	9388	0	9736
0	9388				
0	9736				

Tabelle 47: Confusion Matrizen der einzelnen Pathologien mit vortrainiertem small_affine und Threshold 0.2 (ResNET)

Pathologie	Confusion Matrix	
Enlarged Cardiomeastinum	17024	0
	2100	0
Cardiomegaly	16462	0
	2662	0
Lung Opacity	9841	0
	9283	0
Lung Lesion	18263	0
	861	0
Edema	13505	0
	5619	0
Consolidation	15481	0
	3643	0
Pneumonia	17082	0
	2042	0
Atelectasis	12543	0
	6581	0
Pneumothorax	17685	0
	1439	0
Pleural Effusion	12349	0
	6775	0
Pleural Other	18810	0
	314	0
Fracture	18244	0
	880	0
Support Devices	0	9388
	0	9736

Tabelle 48: Confusion Matrizen der einzelnen Pathologien mit vortrainiertem affine_color und Threshold 0.5 (ResNET)

Pathologie	Confusion Matrix				
Enlarged Cardiomeastinum	<table border="1"> <tr><td>17024</td><td>0</td></tr> <tr><td>2100</td><td>0</td></tr> </table>	17024	0	2100	0
17024	0				
2100	0				
Cardiomegaly	<table border="1"> <tr><td>16462</td><td>0</td></tr> <tr><td>2662</td><td>0</td></tr> </table>	16462	0	2662	0
16462	0				
2662	0				
Lung Opacity	<table border="1"> <tr><td>0</td><td>9841</td></tr> <tr><td>0</td><td>9283</td></tr> </table>	0	9841	0	9283
0	9841				
0	9283				
Lung Lesion	<table border="1"> <tr><td>18263</td><td>0</td></tr> <tr><td>861</td><td>0</td></tr> </table>	18263	0	861	0
18263	0				
861	0				
Edema	<table border="1"> <tr><td>6866</td><td>6639</td></tr> <tr><td>1100</td><td>4519</td></tr> </table>	6866	6639	1100	4519
6866	6639				
1100	4519				
Consolidation	<table border="1"> <tr><td>15481</td><td>0</td></tr> <tr><td>3643</td><td>0</td></tr> </table>	15481	0	3643	0
15481	0				
3643	0				
Pneumonia	<table border="1"> <tr><td>17082</td><td>0</td></tr> <tr><td>2042</td><td>0</td></tr> </table>	17082	0	2042	0
17082	0				
2042	0				
Atelectasis	<table border="1"> <tr><td>12334</td><td>209</td></tr> <tr><td>6369</td><td>212</td></tr> </table>	12334	209	6369	212
12334	209				
6369	212				
Pneumothorax	<table border="1"> <tr><td>17685</td><td>0</td></tr> <tr><td>1439</td><td>0</td></tr> </table>	17685	0	1439	0
17685	0				
1439	0				
Pleural Effusion	<table border="1"> <tr><td>0</td><td>12349</td></tr> <tr><td>0</td><td>6775</td></tr> </table>	0	12349	0	6775
0	12349				
0	6775				
Pleural Other	<table border="1"> <tr><td>18810</td><td>0</td></tr> <tr><td>314</td><td>0</td></tr> </table>	18810	0	314	0
18810	0				
314	0				
Fracture	<table border="1"> <tr><td>18244</td><td>0</td></tr> <tr><td>880</td><td>0</td></tr> </table>	18244	0	880	0
18244	0				
880	0				
Support Devices	<table border="1"> <tr><td>0</td><td>9388</td></tr> <tr><td>0</td><td>9736</td></tr> </table>	0	9388	0	9736
0	9388				
0	9736				

Tabelle 49: Confusion Matrizen der einzelnen Pathologien mit vortrainiertem affine_color und Threshold 0.2 (ResNET)

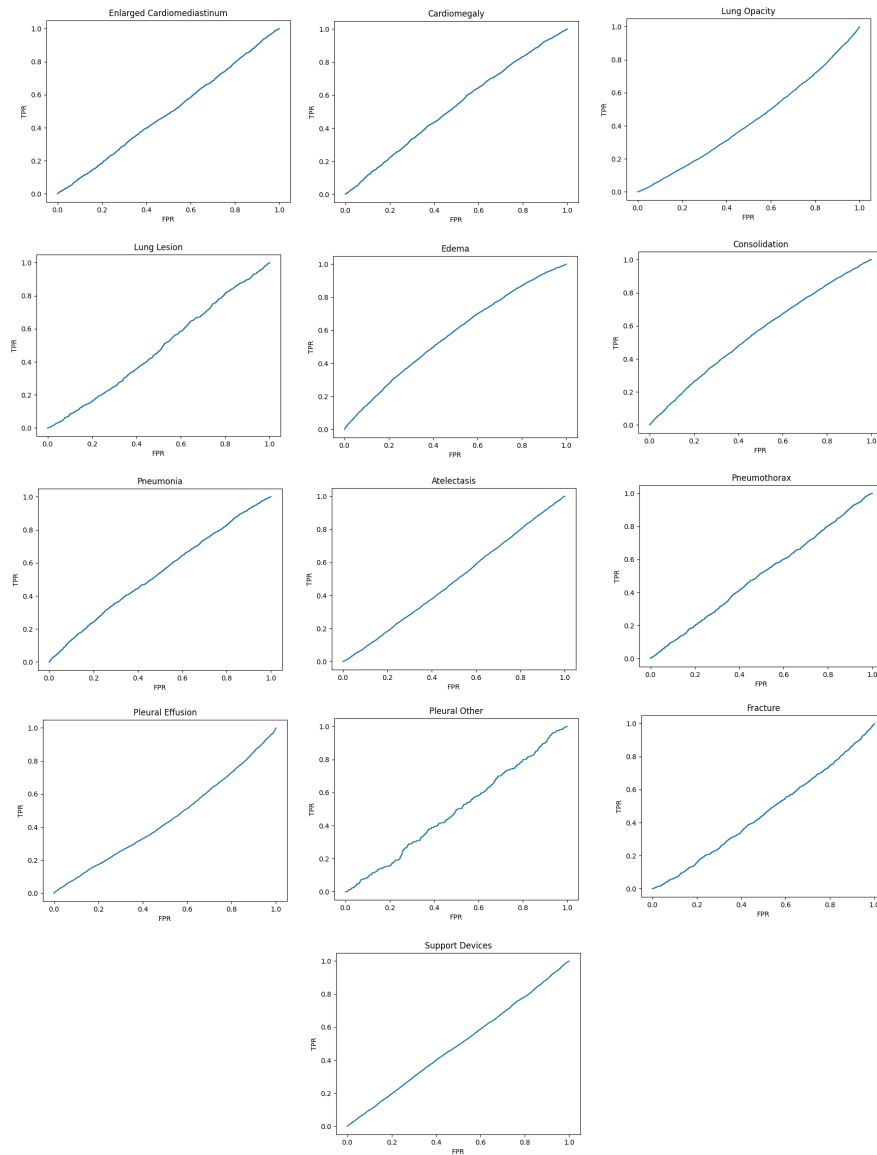


Abbildung 24: Die ROC für die einzelnen Pathologien ohne Augmentierungen (ResNET18)

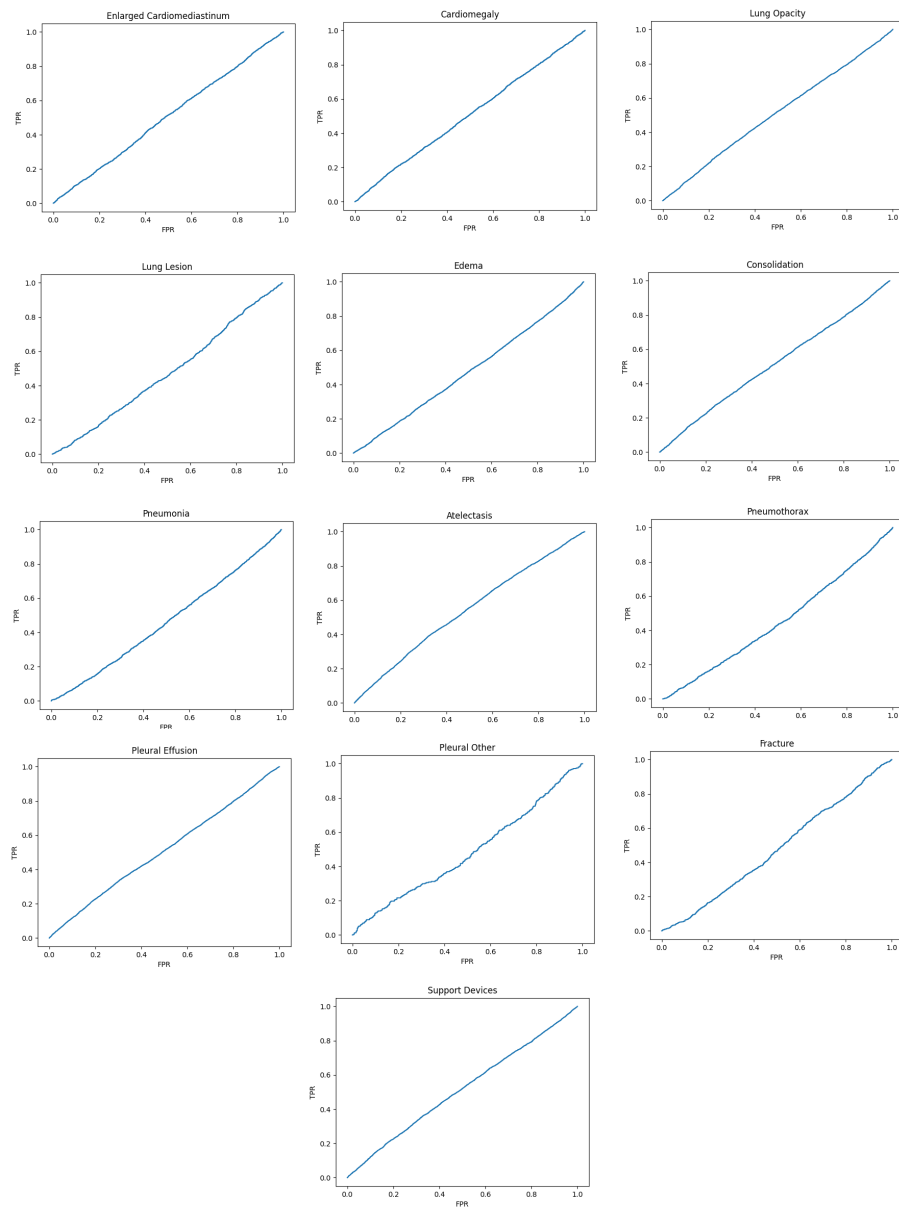


Abbildung 25: Die ROC für die einzelnen Pathologien ohne Augmentierungen (Dense-Net)

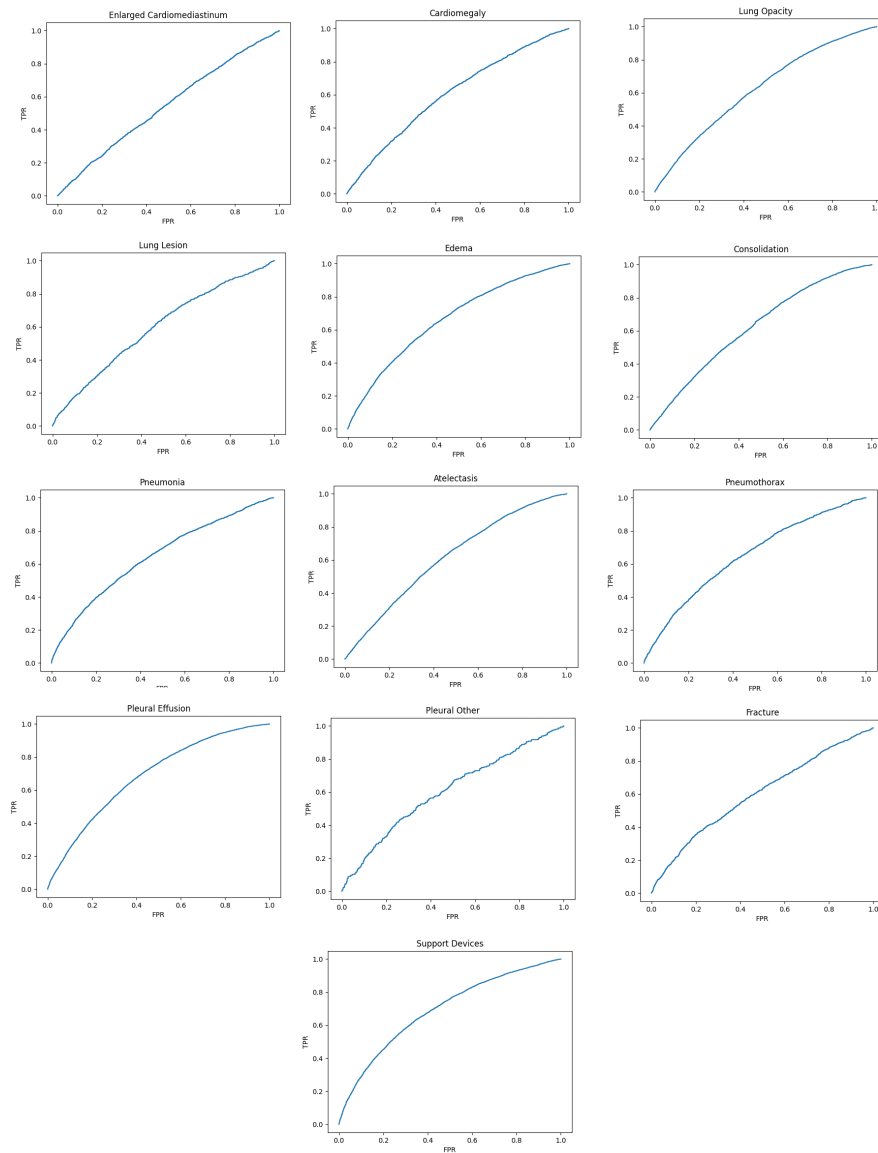


Abbildung 26: Die ROC für die einzelnen Pathologien mit Augmentierungen inspiriert durch Related Work (ResNET18)

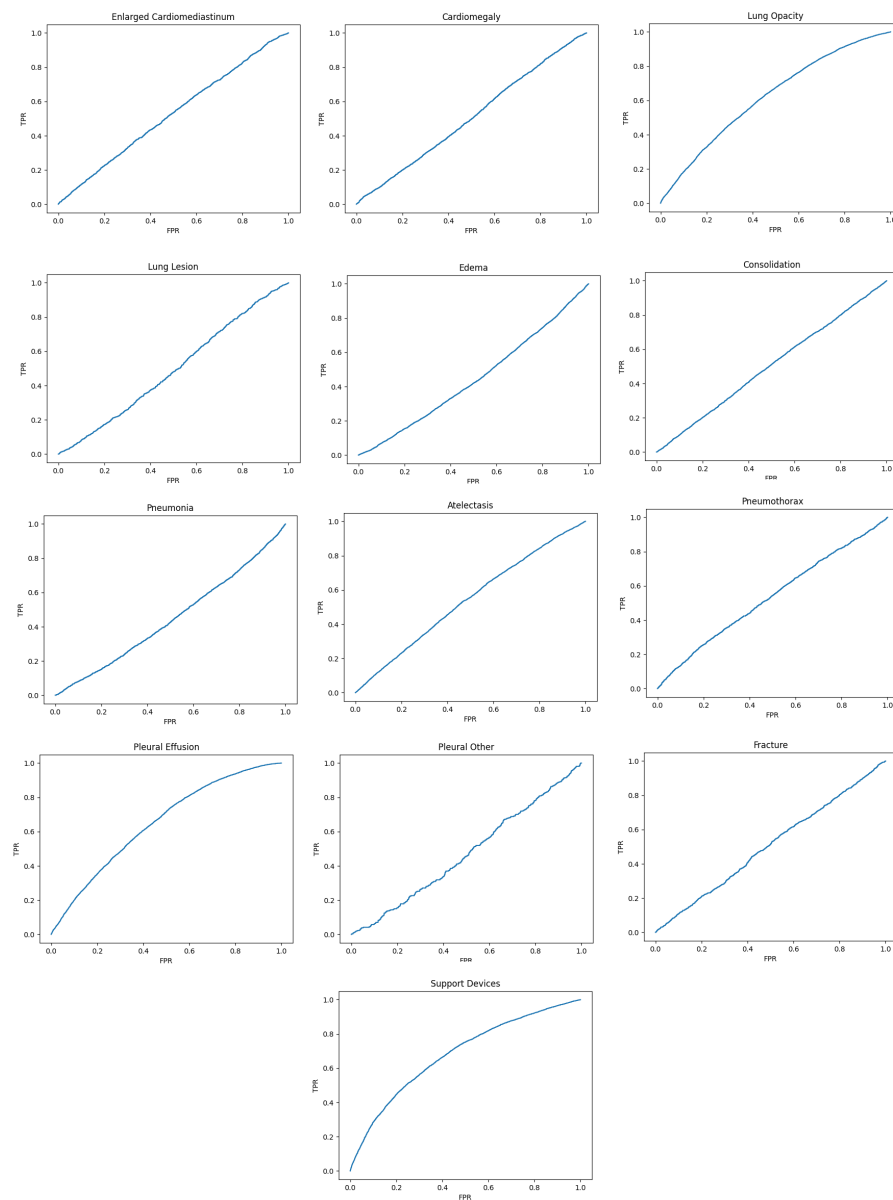


Abbildung 27: Die ROC für die einzelnen Pathologien mit Augmentierungen inspiriert durch Related Work (DenseNet)

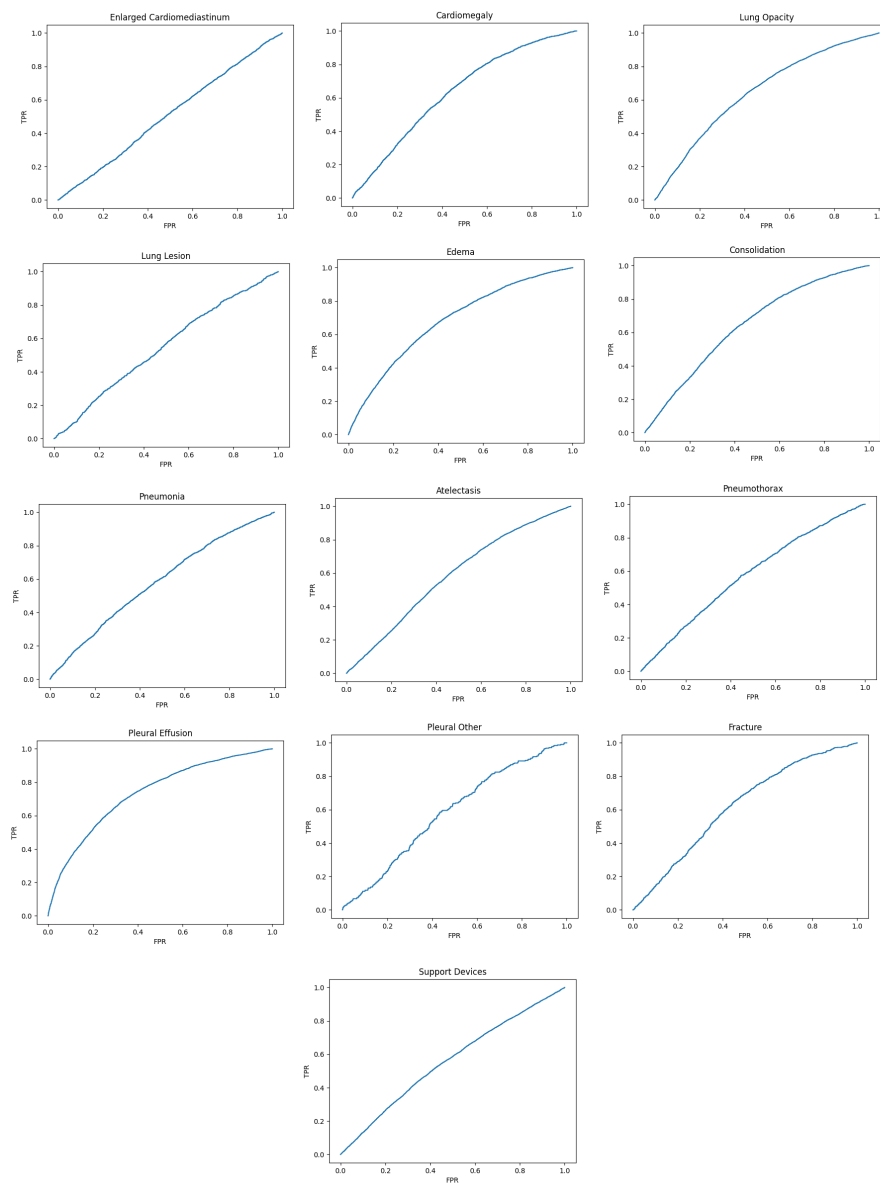


Abbildung 28: Die ROC für die einzelnen Pathologien mit affine_color (ResNET)

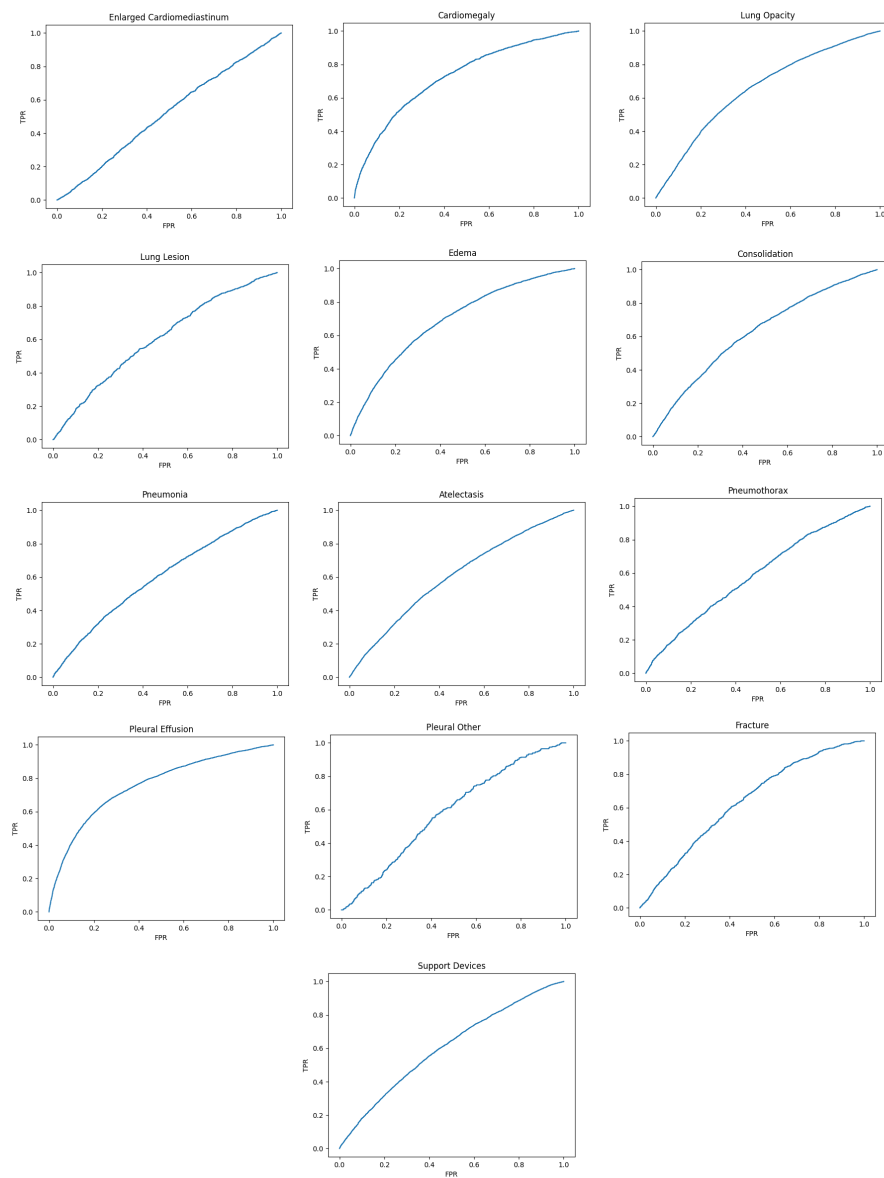


Abbildung 29: Die ROC für die einzelnen Pathologien mit small_affine (ResNET)

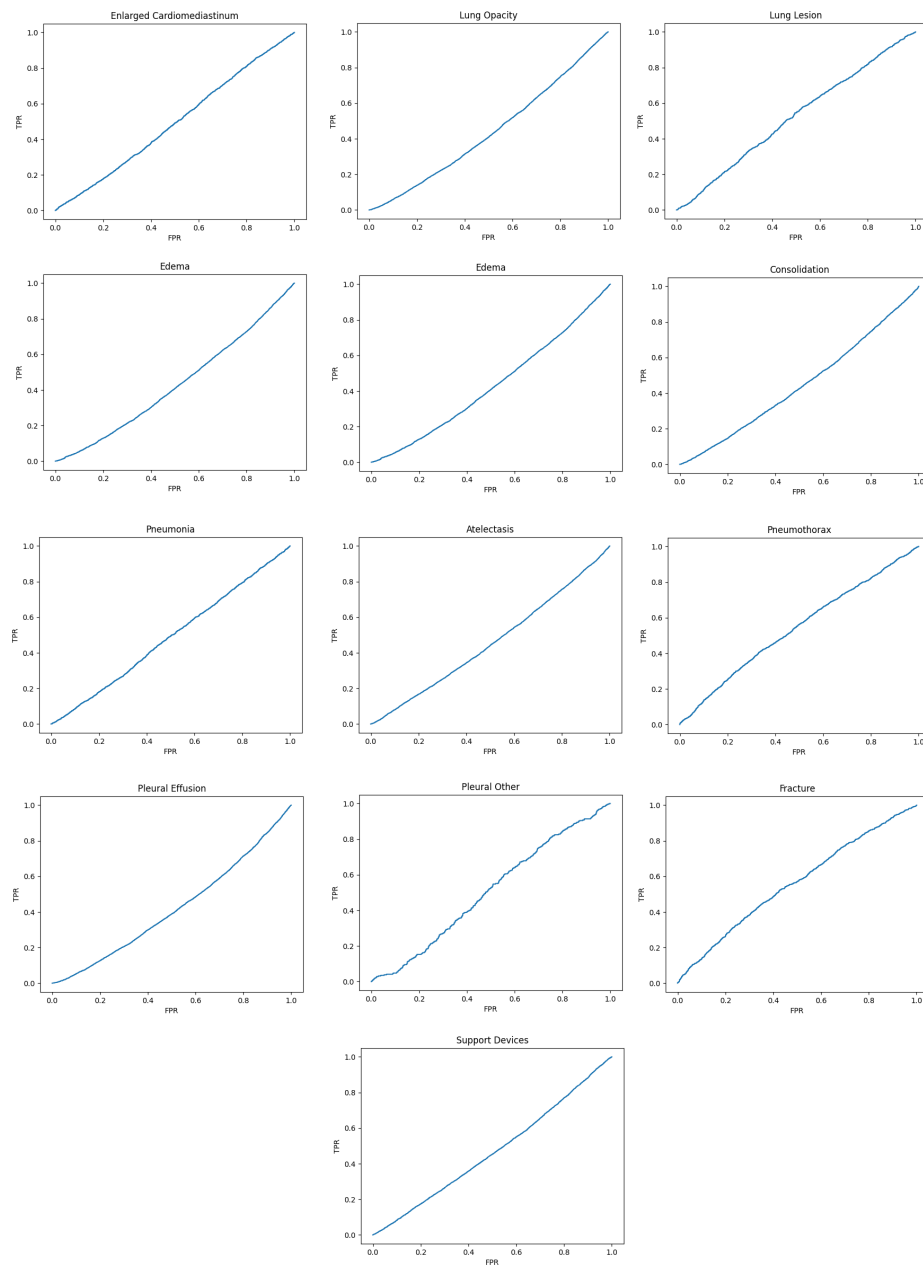


Abbildung 30: Die ROC für die einzelnen Pathologien mit affine_color (DenseNet)

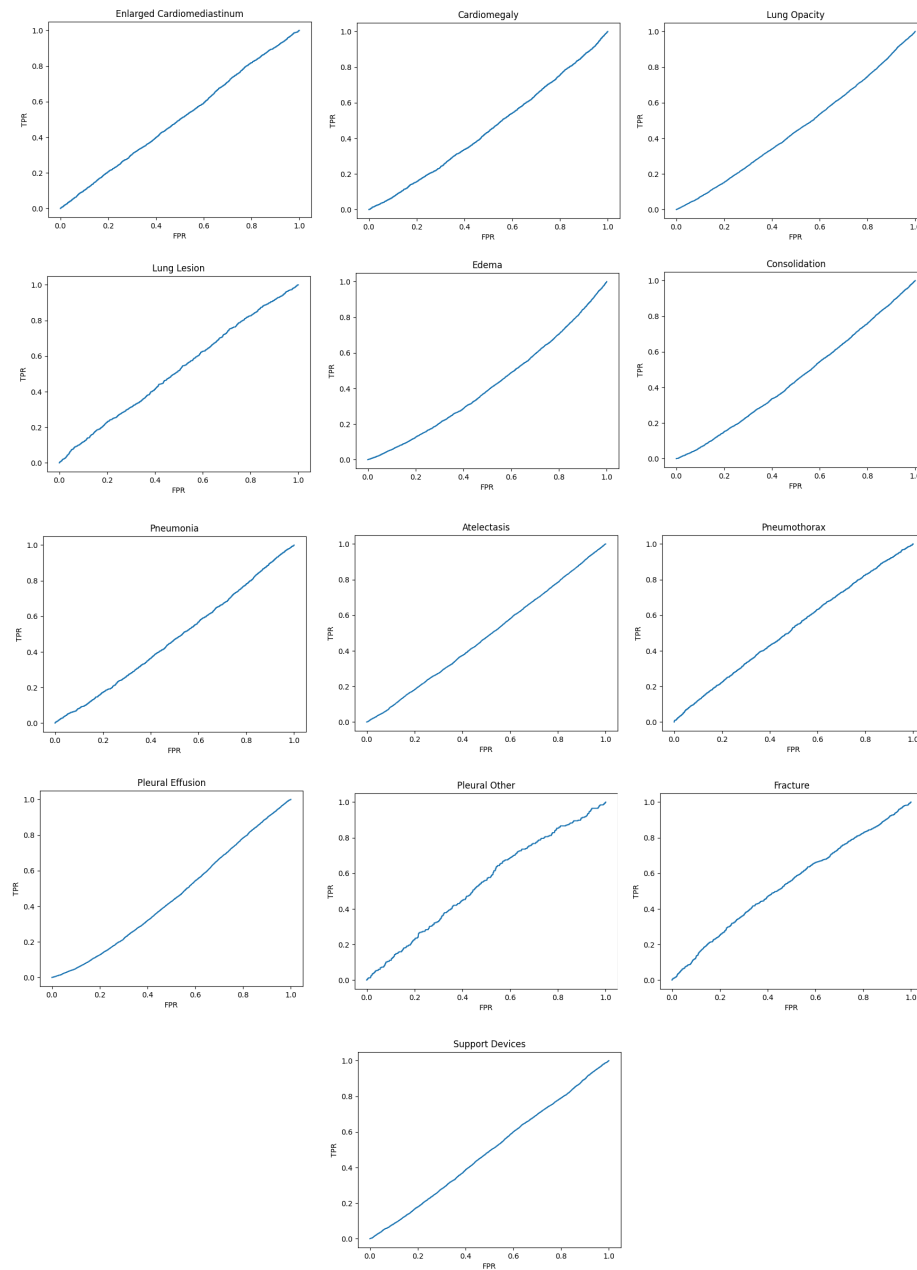


Abbildung 31: Die ROC für die einzelnen Pathologien mit small_affine (DenseNet)

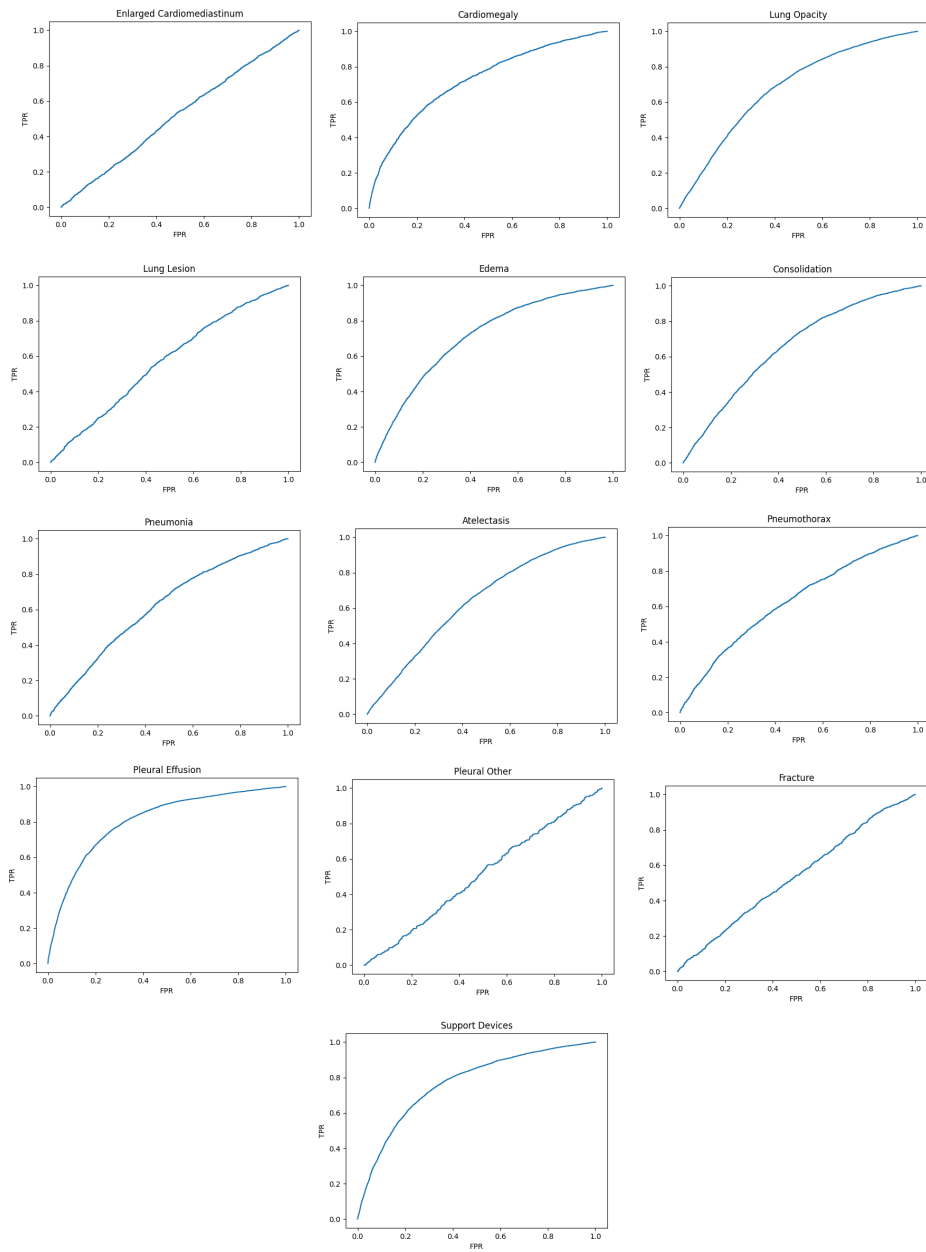


Abbildung 32: Die ROC für die einzelnen Pathologien mit vortrainiertem small_affine (ResNET)

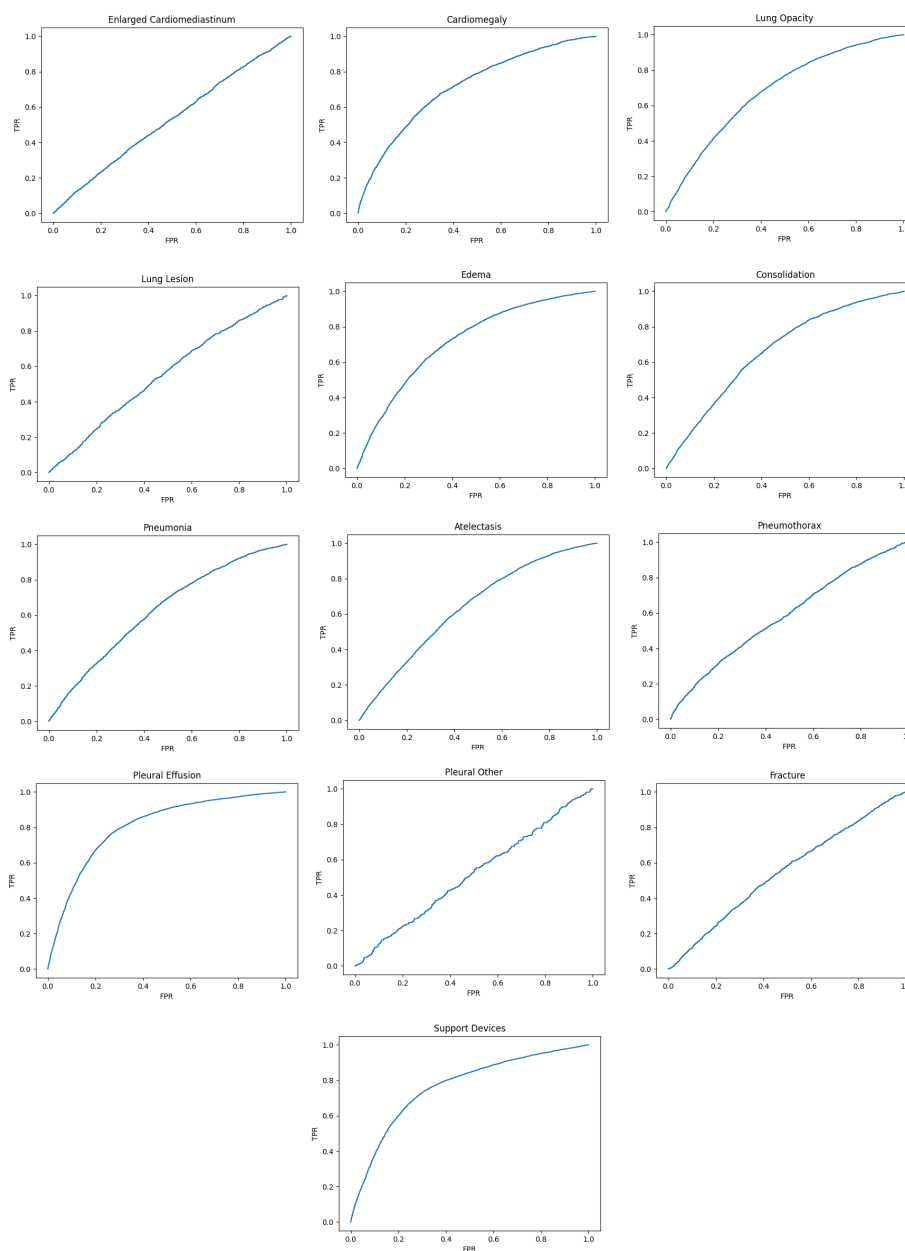


Abbildung 33: Die ROC für die einzelnen Pathologien mit vortrainiertem affine_color (ResNET)

8 USB-Stick mit Ausarbeitung und Programmcode

Der Arbeit ist ein USB-Stick mit digitaler Fassung der Arbeit beigelegt. Zudem befindet sich auf dem Stick der Programmcode. Der Datensatz ist zugänglich auf Scratch vom Lehrstuhl VI

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 :

