Projektbericht: Skin Cancer Detection

Florence Lopez, Jonas Einig, Julian Späth

3. Mai 2018

In der Medizin unterscheidet man bei Hautverletzungen oder Hautveränderungen zwischen benignen und malignen Läsionen. Die malignen Läsionen gelten dabei als die bösartigen Hautveränderungen, die auf Hautkrebs hindeuten. Wir entwickelten daher auf der Grundlage einer vorangegangen Arbeit (Esteva et al. (2017)) einen Klassifikator, der aus einem Datensatz von Hautläsionen zwischen Benignen und Malignen unterscheidet. Durch diesen Klassifikator soll es zukünftig einfacher sein, bösartige Hautläsionen frühzeitig zu erkennen, um diese effizient behandeln zu können. Daher arbeiteten wir zusätzlich noch an einer mobilen Anwendung, welche es Nutzern ermöglicht, ihre Haut vorerst ohne eine ärztliche Analyse nach malignen und benignen Hautläsionen zu untersuchen. Diese Anwendung soll dabei keinen Mediziner ersetzen, sondern lediglich als vorsorgliche Unterstützung dienen.

GitHub Repositories

Klassifizierung: `https://github.com/spaethju/skin-cancer-detection`
Android App: `https://github.com/spaethju/skin-cancer-detection-app`

1 Einleitung

Hautkrebs gilt als eine der häufigsten Krebserkrankungen der Welt. Jährlich erkranken etwa 18.000 Menschen in Deutschland an dieser Krankheit, wobei Hautkrebs allgemein etwa für ein Prozent der Krebstodesfälle verantwortlich ist (Krebsgesellschaft, 2012). Findet eine Erkennung der malignen Hautläsionen frühzeitig statt, so ist es in den meisten Fällen möglich einen tödlichen Verlauf der Krankheit zu verhindern. Daher sind frühzeitige Erkennungssysteme sehr wichtig für die Bekämpfung von Hautkrebs.

Eine Ergänzung zum regelmäßigen Arztbesuch und dem damit verbundenem Hautscreening können daher neuronale Netze bieten, die aufgrund von medizinischen Datenbanken lernen zwischen malignen und benignen Hautläsion zu unterscheiden. Die genaue Implementierung und das Training eines solchen neuronalen Netzes werden wir in dieser Arbeit genauer erläutern.

2 Problemstellung und Zielsetzung

Im Rahmen des Praktikums “Maschinelles Lernen” beschäftigen wir uns mit der folgenden Problemstellung: Ist es möglich einen Klassifikator zu entwickeln, der Bilder von Hautläsionen in maligne und benigne Läsionen unterteilen kann. Maligne Hautläsionen sind die Läsionen, die für den Menschen gefährlich bis sogar tödlich verlaufen können, während benigne Hautläsionen gutartig und ungefährlich sind. Unser

Projekt basiert dabei auf der Arbeit von Esteva et al. (2017), wobei wir die originale Problemstellung jedoch etwas abgewandelt haben. Während Esteva et al. (2017) viele verschiedene Arten von Hautläsionen unterschieden haben, unterscheiden wir lediglich binär, zwischen zwei Klassen, nämlich den gutartigen und den bösartigen Läsionen. Dies vereinfacht die Problemstellung.

Unser Ziel war es, eine möglichst hohe Genauigkeit zu erreichen und vor allem die Anzahl der falsch negativen Vorhersagen möglichst gering zu halten. Im Zweifel soll der Klassifikator eher eine Läsion als maligne klassifizieren, auch wenn sie eigentlich benigne ist, anstatt eine maligne Läsion, die tödlich verlaufen könnte, zu verharmlosen und als benigne zu klassifizieren. Im Folgenden werden wir genauer auf die Methodik eingehen, die hinter unserem Klassifikator steckt und welche Ergebnisse dieser auf unbekannten Bildern von Hautläsionen liefert.

3 Methoden und Tools

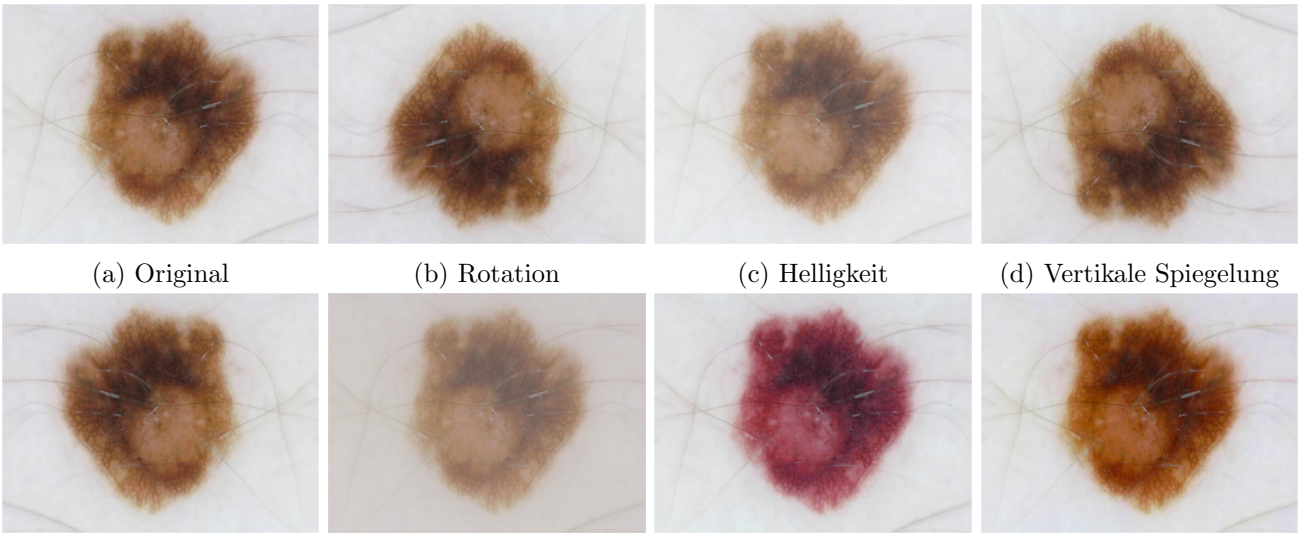
Zur Erstellung des Klassifikators nutzten wir, wie Esteva et al. (2017) auch, das öffentlich zur Verfügung stehende GoogleNet Inception v3, welches ein *convolutional neural network* ist und aus zahlreichen verschiedenen Schichten und Neuronen besteht (Szegedy et al., 2016). Dieses neuronale Netz wurde auf den Bildern des *ImageNet* vortrainiert (Russakovsky et al., 2015). Die von uns genutzten Gewichte stammen aus der Tensorflow (Abadi et al. (2015)) Bibliothek "slim". Ein solches vorheriges Training auf ImageNet etabliert einige grundlegende Strukturen im neuronalen Netz. Es werden Merkmalsdetektoren gelernt, die anschließend in dem Training der eigentlichen Aufgabe verfeinert werden. Das Training und die damit verbundene Vorverarbeitung implementierten wir mittels *Python*, in Kombination mit *Tensorflow* (Abadi et al. (2015)), *Scikit-Learn* (Pedregosa et al. (2011)) und *NumPy*. Für die Evaluierung wurde außerdem noch Matlab verwendet.

Als Datensatz nutzten wir die ISIC-Datenbank (Memorial Sloan Kettering Cancer Center (2017)), welche aus insgesamt 13.768 Bildern von sowohl malignen als auch benignen Hautläsionen besteht. Von den Datensätzen, die von Esteva et al. (2017) genutzt wurden, ist dieser der Einzige, welcher frei verfügbar ist. Die Bilder dieses Datensatzes sind durch eine pathologische Untersuchung gelabelt worden. Somit sind die vorhandenen Daten relativ zuverlässig. Wir teilten den Datensatz in drei Teildatensätze auf: einen Trainingsdatensatz, einen Validierungsdatensatz sowie einen Testdatensatz. Dabei beträgt der Anteil der Trainingsdaten 60% und die Anteile der Validierungs- sowie der Testdaten jeweils 20% aller Bilder. Diese wurden zufällig den Datensätzen zugeordnet.

Es stellte sich heraus, dass die Verteilung der Bilder dieses Datensatzes auf die verschiedenen Klassen (maligne/benigne) sehr unausgeglichen ist. Der Anteil der malignen Läsionen ist viel geringer als der Anteil der benignen Läsionen. Dieses Problem gingen wir durch eine spezielle Art der Randomisierung an. Während des Trainings trennten wir die Menge der Trainingsbilder in die Klassen *maligne* und *benigne* auf. Anschließend haben wir kleine Gruppen von Bildern, welche in das Netz gegeben wurden, aus Bildern dieser beiden Klassen zufällig befüllt. Dabei war die Wahrscheinlichkeit, dass ein Bild aus der Klasse der malignen Bilder stammt $p = 0.5$. Somit wurde dieser unausgeglichene Datensatz ausbalanciert.

Ein weiteres Problem des ISIC-Datensatzes war die variable Größe der einzelnen Bilder. Für ein problemloses Trainieren des Netzes haben wir die Bilder daher vorverarbeitet. Dazu wurden die Bilder auf die kleinste verfügbare Größe skaliert. Für das Training des Netzes verwendeten wir, analog zu Esteva et al. (2017), das maximale zentrale Quadrat des Bildes und skalierten es auf 299px×299px herunter. Somit hatten wir eine homogene Menge an Bildern, die wir nun in einen Trainings-, Validierungs- und einen Testdatensatz unterteilten.

Während des Trainings wurden die Bilder zufällig augmentiert. Durch eine Augmentierung wird der



(a) Original (b) Rotation (c) Helligkeit (d) Vertikale Spiegelung

(e) Horizontale Spiegelung (f) Kontrast (g) Farbwert (Hue) (h) Sättigung

Abbildung 1: Augmentierungsmethoden angewendet auf das originale Bild (1a)

Datensatz künstlich vergrößert, um mehr diverse Trainingsbeispiele für das neuronale Netz zu erhalten. Dazu wendet man eine klassenerhaltende Transformation auf das Bild an. In unserem Ansatz setzen wir viele verschiedene Augmentierungen ein, die jeweils zufällig auf ein Bild angewendet wurden. Abbildung 1 zeigt diese Augmentierungen anhand einer Beispielläsion.

1. **Rotation** (Abbildung 1b): Bei dieser Transformation werden die Bilder je um 0°, 90°, 180° oder 270° gedreht. Diese Augmentierung soll das Netz invariant gegenüber Rotation machen. Dies ist wichtig, da die Orientierung der aufgenommenen Bilder willkürlich ist.

2. **Helligkeit** (Abbildung 1c): Die Helligkeit der Bilder wird hier um einen zufälligen Wert erhöht oder verringert. So soll dem Netz eine gewisse Invarianz gegenüber verschiedenen Lichtbedingungen antrainiert werden.

3. **Vertikales Spiegeln** (Abbildung 1d): Bei dieser Methode werden die Bilder vertikal gespiegelt. Hierbei wird die Charakteristik der Läsion nicht verändert, jedoch werden so weitere Trainingsbilder erzeugt.

4. **Horizontales Spiegeln** (Abbildung 1e): In diesem Fall werden die Bilder horizontal gespiegelt. Auch hier wird die Charakteristik der Läsion nicht verändert, lediglich weitere Trainingsbilder erzeugt.

5. **Kontrast** (Abbildung 1f): Analog zur Helligkeit wird hier der Kontrast um einen zufälligen Wert erhöht oder erniedrigt. Auch dies soll das Netz invariant gegenüber wechselnden Lichtverhältnissen machen und zudem die Trainings-Menge vergrößern.

6. **Farbwert (Hue)** (Abbildung 1g): Hier wird der Farbwert der Bilder zufällig verändert. Durch diese Methode soll die Unterscheidung der Klassen robuster gegenüber Farbänderungen in den Bildern gemacht werden.

7. **Sättigung** (Abbildung 1h): Die Sättigung der Bilder wird zufällig um einen gewissen Wert verändert. Auch dies soll gegen abweichende Lichtverhältnisse helfen und die Menge der Trainingsdaten vergrößern.

3.1 Training

Das Training des GoogLeNet Inception v3 (Szegedy et al. (2015)) implementierten wir in Tensorflow (Abadi et al. (2015)). Dem Netz wurden Gruppen von jeweils sechs Bildern gezeigt. Von den augmentierten und zugeschnittenen Bildern wurde vor dem Inferenz-Schritt der Mittelwert der Bilder des ImageNet Datensatzes (Russakovsky et al. (2015)) abgezogen. Dies ist ein gängiger Vorgang, um die Verteilung der RGB-Werte des ImageNet Datensatzes anzupassen. Nach einem Forward-Pass berechneten wir einen Loss. Auf diesem Loss trainierten wir das neuronale Netz mit dem *Adam-Optimizer*. Im folgenden Abschnitt werden die getesteten Loss-Funktionen näher erläutert.

3.1.1 L1-Loss

Dieser Loss berechnet die absolute Abweichung der Vorhersage von den wahren Labels:

$$L = \sum_{i=0}^{C} \mid lab_i - x_i \mid$$

Dabei ist C die Anzahl der Klassen, lab die wahren Labels und x die Vorhersage des neuronalen Netzes.

Das Ziel eines Trainings mit diesem Loss ist es, möglichst viele Nullen in dem Ergebnis-Vektor zu erhalten, da der L1-Loss spärliche (sparse) Lösungen begünstigt.

3.1.2 L2-Loss

Dieser Loss berechnet die quadratische Abweichung der Vorhersage von den wahren Labels:

$$L = \sum_{i=0}^{C} (lab_i - x_i)^2$$

Dabei ist C die Anzahl der Klassen, lab die wahren Labels und x die Vorhersage des neuronalen Netzes.

Das Ziel eines Trainings mit diesem Loss ist es, große Fehler stark zu bestrafen und die Abweichungen von den wahren Labels so klein wie möglich, jedoch nicht zwingend = 0, zu halten.

3.1.3 Softmax-Kreuzentropie-Loss

Dieser Loss nutzt das Maß der Kreuzentropie, um den Loss zu berechnen. Dabei wird zunächst die Softmax-Funktion auf die Vorhersage und die Labels angewendet und anschließend die Kreuzentropie der beiden Verteilungen berechnet.

Dieser Loss wird dazu benutzt, um Verteilungen einander anzupassen. In unserem Ansatz verwendeten wir jedoch einen Vektor der Länge zwei als Klassenlabel. Bei solch einer kleinen Anzahl an Werten erwarteten wir, dass diese Loss-Funktion vergleichsweise schlecht abschneidet.

Um die oben angesprochene Unausgeglichenheit der Klassen in unserem Datensatz weiter zu balancieren und um eine bessere Performance für maligne Läsionen zu erreichen, nutzten wir bei dem L1-, wie auch bei dem L2-Loss zusätzlich einen Gewichtsterm. Dieser gewichtete die Abweichungen abhängig von der Zielklasse. Dabei wurden die Werte mit lab = maligne mit 3 und die Werte mit lab = benigne mit 0.5 multipliziert.

3.2 Analysemethoden

Für die Bewertung und Optimierung unseres Klassifizierers wurden verschiedene Methoden angewandt. Die Genauigkeit konnte mit unserem Datensatz nur unter großer Skepsis betrachtet werden. Sie berechnet sich durch die Anzahl der richtig vorhergesagten Stichproben dividiert durch die Anzahl aller Stichproben:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

Da der in diesem Projekt verwendete Datensatz deutlich mehr negative als positive Beispiele enthält, können unter dieser Berechnung sehr hohe Genauigkeiten auftreten, obwohl eventuell kein positives Beispiel richtig vorhergesagt wurde. Eine weitaus bessere Analyse ist möglich, indem die richtig positiven und richtig negativen Beispiele getrennt betrachtet werden. Die Sensitivität, auch richtig-positiv-Rate oder auch Recall genannt, entspricht in unserem Fall dem Anteil an tatsächlich malignen Hautläsionen, bei denen diese auch als maligne erkannt wurden:

$$\text{Sensitivität} = \frac{TP}{TP + FN}$$

Die Spezifität, auch richtig-negativ-Rate genannt, entspricht dem Anteil an tatsächlich benignen Hautläsionen, die auch als benigne erkannt wurden:

$$\text{Spezifität} = \frac{TN}{TN + FP}$$

Die getrennte Betrachtung beider Werte erlaubt es uns direkt zu sehen, ob maligne oder benigne Beispiele besser erkannt werden und dementsprechend zu optimieren.

Zusätzlich zogen wir noch den *Matthews correlation coefficient* in Betracht. Dieser wird als Qualitätsmaßstab in binären maschinellen Lernmethoden verwendet, im Speziellen wenn der Datensatz sehr unausgeglichen ist.

$$\text{MCC} = \frac{TP * TN - FP * FN}{\sqrt{(TP + FP) * (TP + FN) * (TN + FP) * (TN + FN)}}$$

Im Gegensatz zu den anderen hier verwendeten Methoden, liefert der MCC Werte zwischen -1 und 1. Ein MCC von 0 ist somit nicht besser als eine Zufallsvorhersage. Ein MCC von 1 hingegen steht für eine komplette Übereinstimmung, während -1 für die komplette Misklassifizierung steht.

Außerdem verwendeten wir noch den F2-Score als Qualitätsmaßstab. Der F1-Score ist das harmonische Mittel zwischen dem Recall und der Precision, der F2-Score dagegen gewichtet den Recall stärker und setzt somit den Schwerpunkt mehr auf die falsch negativen Stichproben, also die malignen Hautläsionen, die als benigne erkannt wurden:

$$\text{Precision} = \frac{TP}{TP + FP}$$

$$\text{Recall/Sensitivität} = \frac{TP}{TP + FN}$$

$$\text{F1-Score} = 2 * \frac{\text{precision} * \text{recall}}{\text{precision} + \text{recall}}$$

$$\text{F2-Score} = 5 * \frac{\text{precision} * \text{recall}}{4 * \text{precision} + \text{recall}}$$

Schließlich entschieden wir uns noch dazu, die *Receiver-Operating-Characteristics-Kurve (ROC)* und die dazugehörige *area under the curve (AUC)* zu verwenden. Diese Methode stellt visuell die Abhängigkeit

der Sensitivität von der falsch-positiv-Rate dar. Eine ROC-Kurve nahe der Diagonalen deutet auf eine Zufallsvorhersage hin und hat eine AUC von 0.5. Eine gute ROC-Kurve liegt deshalb oberhalb der Diagonalen und steigt senkrecht an bevor die falsch-positiv-Rate erhöht wird. Eine AUC von 1 steht somit für eine perfekte Klassifizierung, eine AUC von 0 für komplette Misklassifizierung. In diesem Falle könnten die Labels einfach umgedreht werden. Ziel ist es also eine ROC-Kurve weit entfernt von der Diagonalen zu erhalten.

Wir entschieden uns bewusst für eine größere Anzahl von Qualitätsmaßstäben. Da ein Parameter für sich nie die Komplexität des Klassifizierers komplett beschreiben kann, sollen die verschiedenen Qualitätsmaßstäbe bei der Evaluierung und der Entscheidung für den für uns geeignetsten Klassifizierer als Gesamtheit betrachtet und verwendet werden. Dies ermöglicht es uns, die Klassifizierung gezielt in die von uns gewünschte Richtung zu lenken, um beispielsweise falsch negative Vorhersagen zu minimieren.

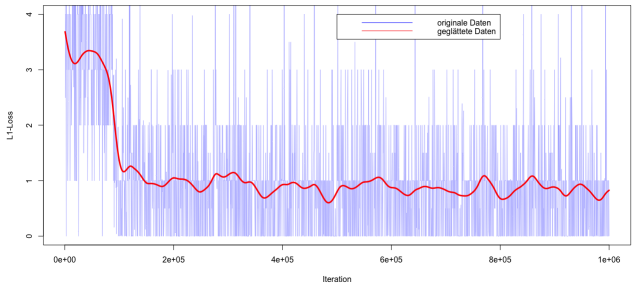
3.3 Framework

Um sowohl das Training als auch die Evaluation auf verschiedenen Systemen durchführen und Parameteränderungen leicht über die Konsole testen zu können, entwickelten wir ein Framework. Über dieses Framework können Trainingsdurchläufe einfach gestartet und später automatisiert evaluiert werden. Dabei lassen sich die Trainingsparameter (Batchsize, Loss-Funktion, Lernrate, Datensatz), sowie plattformspezifische Einstellungen, wie der Speicherort der Daten und den Bezeichner der gewünschten Grafikkarte, anpassen. Startet man das Python-Skript “start_training.py” mit den entsprechenden Parametern, wird automatisch ein neuronales Netz geladen und trainiert, sowie die Log-Dateien und die Gewichte entsprechend gespeichert. Für die Evaluation können analog die Systemparameter sowie der Datensatz, auf dem das Netzwerk evaluiert werden soll, eingestellt werden (Validierungsdatensatz oder Testdatensatz).

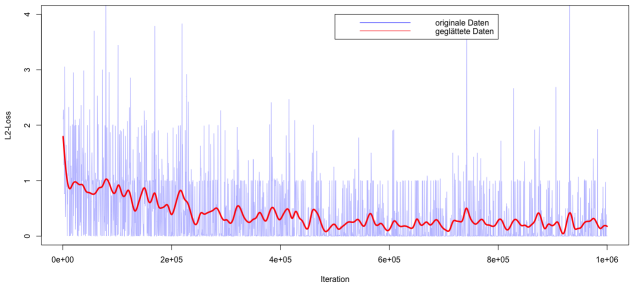
4 Ergebnisse

Um einen guten Klassfizierer zu bekommen, waren einige Trainingsdurchläufe notwendig. So kamen wir letzten Endes auf etwa 30 Trainingsdurchläufe, die jeweils einige Tage trainierten. Zur Parameteroptimierung für das Training wurden die trainierten Klassifizierer auf einen Validierungs-Datensatz angewandt, wobei die ersten Durchgänge dabei nur minimal bessere Ergebnisse als der Zufall lieferten. Durch verschiedene Anpassungen, die in Kapitel 3.1 näher erläutert wurden, konnten schließlich besser Ergebnisse erzielt werden.

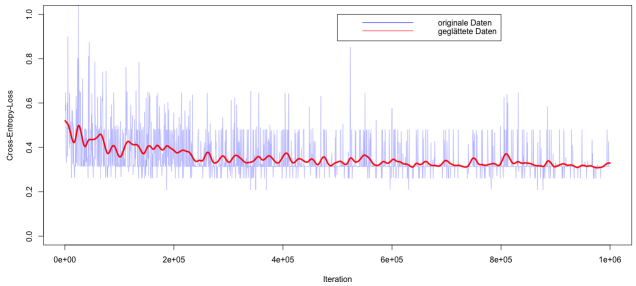
Einen großen Einfluss auf das Training hatte die Wahl der Loss-Funktion. Aus diesem Grund haben wir die Funktionen anfangs mit Blick auf ihr Konvergenzverhalten bei unserem Klassifizierungsansatz untersucht. In Abbildung 2 sind die in Kapitel 3.1 besprochenen Loss-Funktionen über den Verlauf je eines Trainings gezeigt. Dabei fällt auf, dass der L1-Loss (Abbildung 2a) anfangs stark abfällt, jedoch nach etwa 100000 Iterationen stagniert. Betrachtet man hingegen die Kurve des L2-Loss (Abbildung 2b), ist ein kontinuierlicher Abfall des Losses über eine deutlich längere Zeit (bis etwa Iteration 600000) erkennbar. Eine längere Trainingsphase verbessert die Performanz des Trainierten Netzes. Die Gewichte des Netzes können so besser trainiert werden. Im Vergleich dazu ändert sich der Softmax-Kreuzentropie-Loss (Abbildung 2c) kaum. Der Wert dieser Loss-Funktion ändert sich kaum. Dadurch ist für das Training unseres neuronalen Netzes ein nur wenig aussagekräftiger Gradient verfügbar. Somit scheint der L2-Loss die am besten geeignete Loss-Funktion für unsere Problemstellung zu sein. Dies zeigte sich auch bei der Evaluation der Netze.



(a) L1-Loss



(b) L2-Loss



(c) Softmax-Kreuzentropie-Loss

Abbildung 2: Konvergenzverhalten verschiedener Loss-Funktionen bei gleichen Trainingsparametern.

Unser erfolgreichstes Training (2018-03-10_19-16-32) wurde mit einem L2-Loss und einer Lernrate von $\eta = 0.000001$ trainiert. Die Ergebnisse einiger ausgewählter Trainingsdurchläufe werden in Abbildung 3 als ROC-Kurve dargestellt. Jede Kurve steht für einen ausgewählten Klassifizierer und die entsprechende Anwendung auf den Validierungsdatensatz. Wie man gut erkennen kann, verlaufen zwei der Kurven nahe der Diagonalen. Die beiden Klassifikatoren (2018-03-12_22-04-38 und 2018-03-12_22-17-46) liefern somit keine guten Ergebnisse und sind nur etwas besser als eine zufällige Entscheidung. Gute Klassifikatoren verlaufen in der ROC-Kurve links oben und sind in diesem Fall 2018-03-10_19-16-32, 2018-03-09_21-34-4 und 2018-04-13_20-38-20. Die anderen Klassifikatoren können auf den ersten Blick als in Ordnung eingestuft werden.

Um die ROC-Kurve nicht nur optisch, sondern auf Grund von Fakten zu evaluieren, zeigt Tabelle 1 die Klassifikatoren aus Abbildung 3 zusammen mit ihrer berechneten AUC. Es ist deutlich zu erkennen, dass die Kurven, die links oben verlaufen, gleichzeitig eine höhere AUC aufweisen. Somit liefert der Klassifikator 2018-03-12_22-04-38 mit einem AUC von 0.6 die schlechteste Performance und ist nur leicht besser als eine Zufallsvorhersage. Der Klassifikator 2018-03-10_19-16-32 hingegen hat mit einer AUC von 0.9 die in diesem Fall beste Performance.

Da wir bei der Hautkrebs-Erkennung noch ein paar Sonderanforderungen an den Klassifikator haben und einen sehr unausgeglichenen Datensatz verwendeten, entschieden wir uns die zwei Klassifikatoren mit der höchsten AUC (2018-03-10_19-16-32 und 2018-03-09_21-34-47) noch genauer zu evaluieren. Unser Klassifizierer soll am Ende eine hohe Vorhersagegenauigkeit haben, allerdings sollten nicht zu viele wirklich positive Ergebnisse (maligne) als negativ (benigne) klassifiziert werden. Unter Berücksichtigung wurden für diese zwei Klassifikatoren noch weitere Qualitätsmaßstäbe berechnet. Tabelle 2 zeigt die verschiedenen Qualitätsmaßstäbe, die schon in Kapitel 3.2 vorgestellt wurden. Wie man der Tabelle gut entnehmen kann, ist der Klassifikator 2018-03-10_19-16-32 in nahezu allen Belangen besser als 2018-03-09_21-34-47. Einzig und allein in der Spezifität weist er einen minimal schlechteren Wert auf.

Da unser Klassifikator maligne Hautläsionen auch wirklich als solche erkennen soll, präferieren wir in

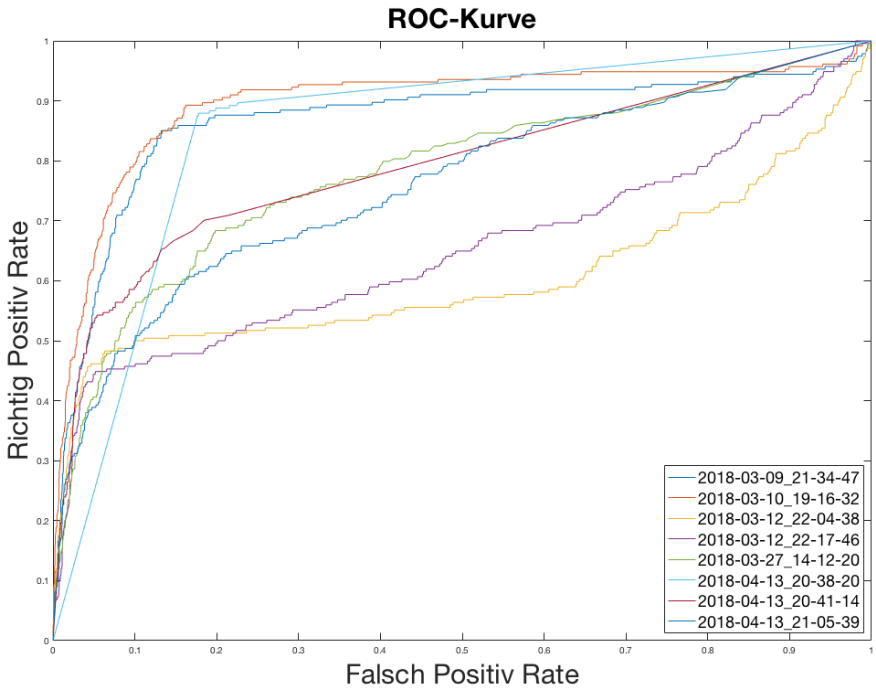


Abbildung 3: ROC-Kurven einiger trainierter Klassifizierer, angewandt auf den Validierungs-Datensatz.

Klassifikator	AUC
2018-03-09_21-34-47	0.87
2018-03-10_19-16-32	0.90
2018-03-12_22-04-38	0.60
2018-03-12_22-17-46	0.65
2018-03-27_14-12-20	0.78
2018-04-13_20-38-20	0.86
2018-04-13_20-41-14	0.79
2018-04-13_21-05-39	0.75

Tabelle 1: ROC-AUC Ergebnisse einiger trainierter Klassifikatoren mit variierenden Parametern

Klassifikator	TP	FN	TN	FP	MCC	F2	Acc	Sens	Spez
2018-03-09_21-34-47	119	115	2406	113	0.47	0.51	0.92	0.51	0.96
2018-03-10_19-16-32	142	91	2406	114	0.54	0.60	0.93	0.61	0.95

Tabelle 2: Scores der zwei besten Klassifizierer: TP = True Positives, FN = False Negatives, TN = True Negatives, FP = False Positives, MCC = Matthews Correlation Coefficient, F2 = F2-Score, Acc = Genauigkeit, Sens = Sensitivität und Spez = Spezifität

diesem Fall den, mit den wenigen falsch negativen und den mehr wirklich positiven Ergebnissen. Dabei wurde trotzdem nur ein negatives Ergebnis mehr fälschlicherweise als positiv (maligne) eingeordnet. Aufgrund dieser Evaluierung schlussfolgerten wir, dass der Klassifizierer mit der höheren AUC, Genauigkeit und Sensitivität sowie einem höherem MCC und F2-Score unser Problem der Hautkrebserkennung besser lösen kann. Eine etwas geringere Spezifität nehmen wir hierbei in Kauf, wobei diese so gering ist, dass es sich auch um eine Zufallserscheinung handeln könnte.

Obwohl der Klassifizierer insgesamt schon gute Vorhersagen macht, werden die Ergebnisse durch die hohe Spezifität von 95% etwas verfälscht. Mit einer Sensitivität von 0.6 würden wir nämlich nur 60% aller malignen Hautläsionen auch als wirklich maligne klassifizieren. Im Umkehrschluss heißt das, dass vier von zehn potentiellen malignen Hautläsionen unerkannt bleiben. Da diese Sensitivität noch zu nah an einer Zufallsklassifizierung liegt, schauten wir uns die Ergebnisse des neuronalen Netzes des Klassifikators 2018-03-10_19-16-32 noch einmal genauer an, da dieser in der vorherigen Evaluierung die besten Werte aufgewiesen hatte. Das Ergebnis des neuronalen Netzes ist ein Score, der zwischen null und eins liegt. Ein Score über 0.5 klassifiziert eine Hautläsion als maligne, ist er kleiner als 0.5 als benigne. Um den Klassifizierer nun weiter in die von uns gewünschte Richtung zu lenken, verschoben wir die Entscheidungsschranke (Threshold) und schauten, wie sich diese Verschiebung auf die Klassifikation des Validierungsdatensatzes auswirkte. Diese Auswirkung auf die verschiedenen Qualitätsmaßstäbe wird in Abbildung 4 dargestellt.

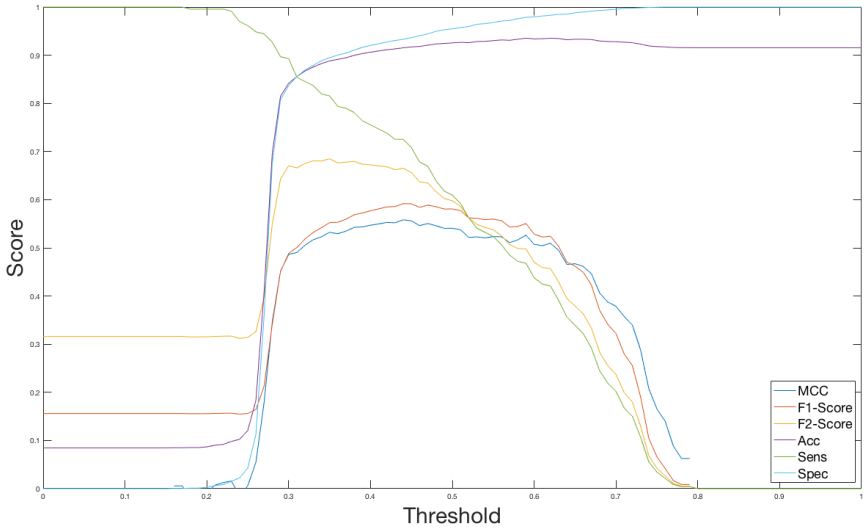


Abbildung 4: Score-Threshold Abhängigkeit

Wie man der Abbildung entnehmen kann, gibt es einen Bereich zwischen 0.3 und 0.6 in der der MCC und der F2-Score gute Werte aufweisen. Da der F2-Score die falsch negativen Stichproben noch stärker bestraft, konzentrierten wir uns hier mehr auf diesen Qualitätsmaßstab. Außerdem wollten wir eine möglichst hohe Sensitivität erreichen. Der F2-Score weist im Bereich 0.3 und 0.45 seine höchsten Werte auf, wobei der kleinste Threshold gleichzeitig die höchste Sensitivität liefert. Es wäre also naheliegend gewesen einfach diesen Wert zu nehmen. Da die Spezifität zwischen dem Threshold 0.25 und 0.3 rapide ansteigt, sollte allerdings ein gewisser Abstand zu diesem Bereich gewahrt werden.

Abbildung 5 zeigt die Verteilung der Scores des Validierungsdatensatzes, der durch 2018-03-10_19-16-32 klassifiziert wurde. Sie zeigt noch einmal deutlicher, dass die meisten benignen Stichproben einen Score zwischen 0.2 und 0.3 aufweisen. Somit entschieden wir uns den Threshold von den ursprünglichen 0.5 auf 0.35 herunterzusetzen. Damit halten wir genug Abstand von der extremen Änderung der Spezifität, klassifizieren gleichzeitig aber deutlich mehr maligne Stichproben richtig, ohne zu viele Fehler bei den negativen Stichproben zu machen.

Durch diese nun sensitivere Klassifizierung, erhielten wir schließlich die Werte, wie sie in Tabelle 3 gelistet sind. Die Sensitivität konnten wir durch die Verschiebung des Thresholds von 0.61 auf 0.81 erhöhen. Es werden also nur noch zwei von zehn maligne Hautläsionen fälschlicherweise benigne klassifiziert. Da-

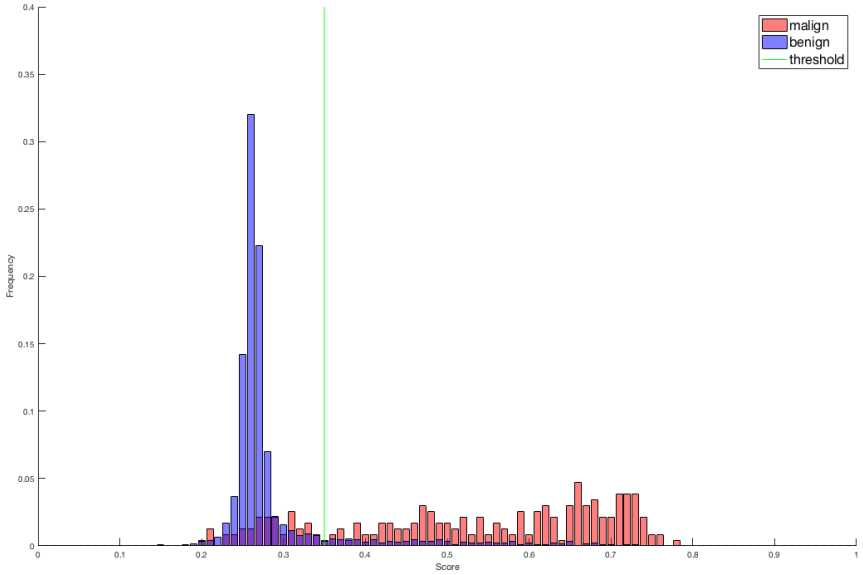


Abbildung 5: Verteilung der Vorhersagen auf dem Validierungsdatensatz. Maligne Stichproben werden in rot und benigne Stichproben in blau dargestellt. Die grüne Linie stellt den gewählten Threshold dar. Alle Vorhersagen links des Thresholds werden als benigne und alle Vorhersagen rechts des Thresholds werden als maligne klassifiziert.

bei akzeptierten wir eine Verringerung der Spezifität von 0.95 auf 0.89. Es wird also etwa jede zehnte benigne Hautläsion falsch als maligne klassifiziert. Die Genauigkeit verringerte sich dabei von 93% auf 89%, allerdings erhöhten wir den F2-Score von 0.6 auf 0.68, der bei einem unausgeglichenen Datensatz höher zu gewichten ist als die Gesamtgenauigkeit.

Threshold	TP	FN	TN	FP	MCC	F2	Acc	Sens	Spez
0.5	142	91	2406	114	0.54	0.60	0.93	0.61	0.95
0.35	190	43	2255	265	0.55	0.68	0.89	0.81	0.89

Tabelle 3: Scores des Klassifikators mit dem Threshold bei 0.35

Unser bester Klassifikator 2018-03-10_19-16-32 wurde mit einer Batchsize von sechs sowie einer Lernrate von $e^{-0.6}$ trainiert. Als Loss-Funktion verwendeten wir den L2-Loss. Als Entscheidungsschranke nahmen wir, wie oben beschrieben, 0.35. Dies bedeutet, dass alle Stichproben, die einen Score von über 0.35 erhalten, als maligne Hautläsionen eingestuft werden. Schließlich testeten wir unseren finalen Prädiktor noch auf unserem Testdatensatz. Dieser Datensatz wurde bis an diese Stelle isoliert aufbewahrt und keine der enthaltenen Daten waren dem Klassifikator bekannt. Dies war nötig, um zu testen, ob dieser auch generalisierbar ist. Das bedeutet, dass dieser nicht nur die Trainingsdaten auswendig gelernt hat, sondern auch auf nie gesehenen Daten gute Vorhersagen macht. Da wir anhand des Validierungsdatensatzes schon die Trainingsparameter angepasst haben, wurde dieser Testdatensatz jetzt am Schluss verwendet, um die endgültige Performance zu testen. Tabelle 4 zeigt die Qualitätsmaßstäbe, wie sie auf dem finalen Testdatensatz berechnet wurden.

Wie zu erwarten, sind die Ergebnisse ähnlich, aber etwas schlechter als auf dem Validierungsdatensatz, auf dem wir gezielt auf gute Werte hingearbeitet haben. Letzten Endes erkennt unser Prädiktor 90% der

TP	FN	TN	FP	MCC	F2	Acc	Sens	Spez
149	60	2285	114	0.48	0.60	0.88	0.71	0.9

Tabelle 4: Qualität des Klassifikators nach Anwendung auf ungesehenem Trainingsdatensatz, mit der Entscheidungsgrenze bei 0.35

benignen Hautläsionen richtig und 71% der malignen Hautläsionen. Der F2-Score ist mit 0.6 auch etwas gesunken, was an den häufigeren falsch negativen Vorhersagen liegt.

Zusammenfassend kann man sagen, dass diese Werte auf jeden Fall in einem guten Bereich liegen. Die drei von zehn falsch negativen Vorhersagen sind zwar nicht erwünscht, aber auch durch den unausgeglichenen Datensatz zu begründen. Mit mehr malignen Trainingsdaten wären womöglich noch bessere Vorhersagen möglich. Nichtsdestotrotz erhielten wir nach unzähligen Trainings einen Prädiktor, der mit Vorsicht zur häuslichen Vorsorge verwendet werden kann und akzeptable Vorhersagen für die Hautkrebserkennung liefert.

5 Aussicht und Diskussion

Wie unsere Ergebnisse zeigen konnten, ist es durchaus möglich einen Klassifikator zu erstellen, der Bilder von Hautläsionen in maligne und benigne unterteilt. Dieser Klassifikator kann nun in verschiedenen Bereichen erfolgreich eingesetzt werden. Einen der Anwendungsbereiche betrachteten wir in unserem Projekt noch etwas genauer: Wir wollten den erstellten Klassifikator in eine mobile Anwendung einbinden, die es Nutzern ermöglicht, vorerst ohne ärztlichen Rat, ihre Haut auf Unregelmäßigkeiten und Anomalien zu untersuchen und eine vage Einschätzung dieser vorzunehmen. Dazu implementierten wir mittels Android Studio eine mobile Applikation, bei der ein Nutzer mittels seiner Handykamera ein Foto einer Hautläsion aufnehmen kann. Dieses kann anschließend durch unseren Klassifikator ausgewertet werden.

Natürlich ist die Leistung, die eine solche mobile Anwendung bietet, ohne jegliche Gewähr, da eine Handykamera aufgrund ihrer geringeren Auflösung durchaus nicht einen ausgebildeten Hautarzt ersetzen kann. Trotzdem ist es durch diese Applikation möglich, eine erste Einschätzung des Hautzustandes vorzunehmen. Da wir weiterhin bei der Erstellung des Klassifikators sehr großen Wert darauf gelegt haben, vor allem die Rate der falsch negativen Vorhersagen zu minimieren, rät die mobile Anwendung dem Nutzer öfters dazu einen Hautarzt aufzusuchen, als es vielleicht wirklich nötig wäre.

Zusätzlich könnte eine solche mobile Applikation auch in Kombination mit einem speziellen Kamera-Aufsatz für präzise Aufnahmen der Haut genutzt werden. Damit wäre es möglich, die Genauigkeit der Klassifikation zu erhöhen und dadurch bessere Aussagen über den Gesundheitszustand des Nutzers zu treffen, da die Aufnahmen unseren Trainingsdaten noch mehr ähneln. Durch diese neue und einfache Art der Hautuntersuchung, wäre es vor allem auch Nutzern in ländlicheren Regionen, in denen ein Fachärzte-Mangel herrscht, möglich, eventuelle Unsicherheiten und Fragen bezüglich ihres Hautzustandes zu überprüfen. Obendrein birgt der Gebrauch von maschinellem Lernen in der Medizin in Kombination mit einer mobilen Anwendung einen weiteren großen Vorteil: Die Bevölkerung erhält dadurch die Gelegenheit sich von zu Hause aus mit wichtigen medizinischen Themen und Untersuchungen zu beschäftigen. Somit wird das Bewusstsein für schwere Krankheiten, wie zum Beispiel Hautkrebs, und die regelmäßige Auseinandersetzung mit ihnen gefördert. Infolgedessen wäre es möglich eine Prophylaxe vom eigenen Zuhause aus durchzuführen, was wiederum in einer frühzeitigen Erkennung von Anomalien und damit einhergehend, in einer schnellen und effizienten Behandlung dieser resultiert.

In unserem Projekt beschäftigten wir uns vorerst mit der allgemeinen Unterscheidung zwischen benignen und malignen Hautläsionen. Damit unterschied sich unsere Problemstellung von der des Originalpapers von Esteva et al. (2017) erheblich und wurde deutlich vereinfacht. Es wäre also in Zukunft möglich, die originale Fragestellung zu übernehmen und nicht nur zwischen benignen und malignen Hautläsionen, sondern auch zwischen den verschiedenen Unterklassen zu unterscheiden. Damit wäre es möglich, durch die mobile App eine genauere Einschätzung am Patienten vorzunehmen und die jeweils benötigte Medikation besser auf die klassifizierte Läsion einzustellen.

Leider ist unsere Implementierung der mobilen App bisher noch nicht vollständig beendet. Allerdings planen wir, an dieser weiterzuarbeiten und sie dann über das GitHub-Repository dieses Projekts nachzureichen.

Abschließend konnten wir durch unser Projekt zeigen, dass es durch maschinelles Lernen möglich ist, bösartige Hautveränderungen frühzeitig als diese zu erkennen. Diese Art der Klassifizierung bringt in der Medizin viele Vorteile mit sich, nicht zuletzt, da es dem Patienten dadurch möglich wird, eine erste eigene Einschätzung des Gesundheitszustandes vorzunehmen. Natürlich kann die Richtigkeit einer solchen Klassifizierung nicht in dem Maße garantiert werden, in dem sie durch einen fachlich ausgebildeten Hautarzt gegeben ist, dennoch kann sie einen ersten Anhaltspunkt liefern und dazu führen, dass sich Patienten intensiver mit der Krankheit und ihrer Gesundheit auseinandersetzen. Vor allem durch die Einbindung des Klassifikators in eine benutzerfreundliche mobile Applikation, kann der Vorteil des maschinellen Lernens in diesem Fall maximal ausgenutzt werden.

Wir konnten also zeigen, dass der Einsatz von maschinellem Lernen in der Medizin durchaus sinnvoll ist und einen Ansatz darstellt, den es sich lohnt weiterhin zu verfolgen. Es ist unserer Meinung nach, daher sehr wichtig die Forschung in diesem Bereich weiter auszubauen, um weitere Teil-Anwendungsgebiete in der Medizin ausfindig zu machen. Wir sind uns sicher, dass zukünftigen Arbeiten mit diesem Thema viele neue, interessante Erkenntnisse bieten können.

Literatur

Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G. S., Davis, A., Dean, J., Devin, M., Ghemawat, S., Goodfellow, I., Harp, A., Irving, G., Isard, M., Jia, Y., Jozefowicz, R., Kaiser, L., Kudlur, M., Levenberg, J., Mané, D., Monga, R., Moore, S., Murray, D., Olah, C., Schuster, M., Shlens, J., Steiner, B., Sutskever, I., Talwar, K., Tucker, P., Vanhoucke, V., Vasudevan, V., Viégas, F., Vinyals, O., Warden, P., Wattenberg, M., Wicke, M., Yu, Y., and Zheng, X. (2015). TensorFlow: Large-scale machine learning on heterogeneous systems. Software available from tensorflow.org.

Esteva, A., Kuprel, B., Novoa, R. A., Ko, J., Swetter, S. M., Blau, H. M., and Thrun, S. (2017). Dermatologist-level classification of skin cancer with deep neural networks. *Nature*, 542(7639):115.

Krebsgesellschaft, D. (2012). Patientenratgeber Hautkrebs.

Memorial Sloan Kettering Cancer Center (2017). Isic dermoscopic archive. Eingesehen am 26.11.2017.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A.,

Bernstein, M., et al. (2015). Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 115(3):211–252.

Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., and Wojna, Z. (2015). Rethinking the inception architecture for computer vision. *CoRR*, abs/1512.00567.

Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., and Wojna, Z. (2016). Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2818–2826.