Praktikum Maschinelles Lernen
WS 17/18
Prof. Schilling

Florence Lopez
Jonas Einig
Julian Späth

Skin Cancer Detection

(28. November 2017)

Beschreibung

Hautkrebs ist eine der am häufigsten vorkommenden Krebsarten unserer Gesellschaft[1]. Behandlungen reichen von operativen Eingriffen bis hin zur Strahlen- oder Chemotherapie. Wie bei vielen anderen Krebsarten ist eine frühe Erkennung und Unterscheidung zwischen einem malignen oder benignem Hautkrebs von großer Wichtigkeit um die Chancen einer vollständigen Heilung zu erhöhen. Behandlung in einem spätem Stadium sind sehr viel schwieriger, da nach der Metastasierung der Krebs nicht mehr nur lokal begrenzt ist, sondern auch umliegendes Gewebe infiltrieren oder über die Blut- und Lymphbahnen metastasieren kann. Eine klare Trennung zwischen benigne und maligne ist allerdings selbst für Ärzte nicht immer einfach. Dabei liegt es auf der Hand, dass Betroffene selbst erst viel zu spät zum Arzt gehen, da sie den Krebs nicht rechtzeitig als solchen erkennen.

Im Zeitalter von "Big Dataërkennt auch der Medizinsektor langsam, welche Möglichkeiten eine genaue Analyse der vielen medizinischen Daten aus vergangenen Fällen ermöglichen. Durch unterschiedlichste Maschinelle Lernmethoden können Daten in verschiedene Gruppen eingeteilt und somit klassifiziert werden. Im Jahr 2017 wurde schließlich von Esteva et. al, ein Verfahren zur Erkennung von Hautkrebs vorgestellt [2]. Auf 129.450 klinischen Bildern trainierten sie ein *Deep convolutional neural networks* (CNNs) [3], das schließlich in etwa gleich gut Hautkrebs klassifizieren konnte wie ein menschlicher Arzt.

In dieser Arbeit werden wir auf Grundlage des Papers von Esteva et. al auch Verfahren entwickeln, das schließlich zwischen benignem und malignem Hautkrebs unterscheiden und einfach in einer Android App verpackt werden kann. Die App kann natürlich keine ärztliche Diagnose ersetzen, jedoch durch einen gut eingestellten Klassifikator eine Hilfe sein Hautkrebs früh zu erkennen und zum Arzt zu gehen.

Tools

Als Tools zur Umsetzung des Projekts, werden wir das Deep Learining Framework *Tensorflow* [4] in der Version 1.4 verwenden. Tensorflow bietet umfangreiche Möglichkeiten zur Implementierung und zum Training Neuronaler Netze. Zudem wird ein Deployment eines fertigen Netzes für eine Nutzung in einem Android System unterstützt.

Wir werden Python, die Python-API von Tensorflow 1.4 und möglicherweise die Python-Bibliotheken *Pillow* und *Numpy* für eine Vorverarbeitung der Daten nutzen.

Eine optionale Implementierung einer mobilen App, werden wir für Android in Java und den Bibliotheken aus Android Studio implementieren.

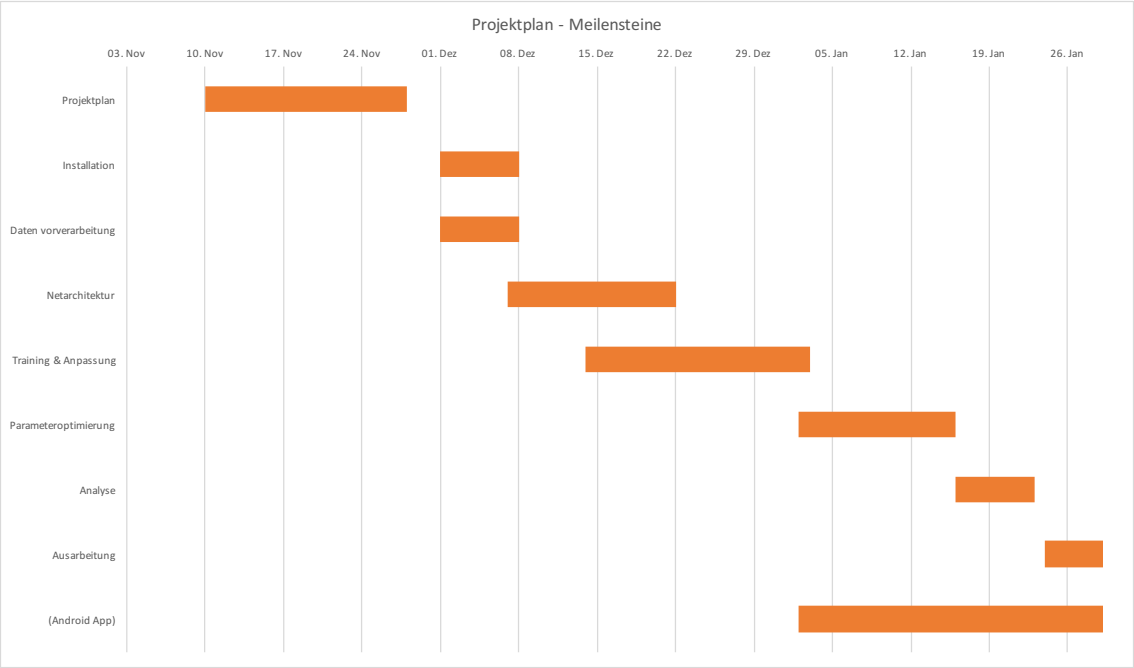

Abbildung 1: Meilensteine des Projekts als GANTT-Chart

Daten

Die Autoren des Papers nutzen in ihrer Arbeit einen Datensatz von insgesamt 129.450 klinischen Bildern, die 2.032 verschiedene Krankheiten abbilden. All diese Krankheiten wurden in drei Unterklassen unterteilt: maligne, benigne und nicht-neoplastische, also nicht-tumoröse Krankheiten bzw. Läsionen. Aus diesem gesamten Datensatz wurden 127.463 Bilder für das Training und die Validierung genutzt und die restlichen 1.942 Bilder wurden zum Testen gebraucht. Die Daten stammten aus 18 verschiedenen Quellen, unter anderem aus dem ISIC Dermoscopic Archive [7], welches ein freier Datensatz ist, der durch digitale Hautbilder dazu dienen soll die Sterblichkeitsrate bei Melanomen zu reduzieren. In unserer Ausarbeitung des Projekts werden wir unsere Daten ausschließlich aus diesem Datensatz beziehen, da sie frei erhältlich und für unsere Zwecke ausreichend sind. Somit verkleinert sich unsere Datenmenge auf 13.786 Bilder. Weiterhin werden wir zuerst lediglich eine Unterscheidung zwischen malignen und benignen Läsionen vornehmen und keine Einzelunterscheidungen zwischen den verschiedenen Krankheiten.
Die Bilder in dem ISIC Datensatz bestehen ausschließlich aus Aufnahmen von melanozytischen Hautläsionen, die anhand von Biopsien als maligne oder benigne Läsionen annotiert sind:

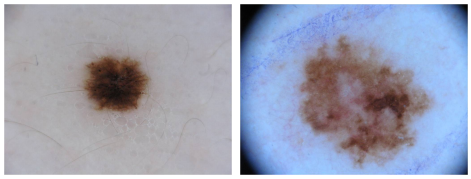

Abbildung 2: Beispiele einer benignen (links) und einer malignen (rechts) Läsion.

Esteva et. al (2017) [2] sortierten bei ihrer Datenverarbeitung alle unscharfen Bilder und solche, die

von zu weit weg aufgenommen aus. Zusätzlich beinhaltete ihr Datensatz auch Bilder der gleichen Läsion aus mehreren verschiedenen Blickwinkeln, was dem Training des Klassifizierers positiv beeinflusst hat. In unserem Fall tritt jede Läsion genau einmal auf, weswegen wir bei der Aufteilung des Datensatzes in Trainings- und Validierungsdaten, anders als die Autoren, auch keinen gesonderten Trenn-Algorithmus anwenden müssen, der dafür sorgt, dass zusammengehörige Bilder auch in das gleiche Subset kommen. Allerdings werden wir zur Verbesserung der Performanz unseres Klassifizierers alle Bilder sowohl in originaler Darstellung als auch in einer rotierten Darstellung präsentieren. Wir erhoffen uns dadurch einen stabileren Klassifizierer, der auch für die Anwendung in einer mobilen Android App geeignet wäre.
Zusammengefasst gestaltet sich unsere Datengewinnung und -verarbeitung etwas anders als die im originalen Paper von Esteva et. al (2017). Wir streben eine Vereinfachung des Klassifizierungsverfahrens an, indem wir nur zwischen zwei verschiedenen Klassen (maligne vs. benigne) unterscheiden und dafür insgesamt weniger Daten nutzen. Dies hat den Vorteil, dass wir anders als im Paper nicht auf die Mithilfe ausgebildeter Dermatologen angewiesen sind, die sich alle Bilder des Datensatzes einzeln ansehen müssten, um die Daten den 2.032 verschiedenen Krankheiten zuzordnen. Unser Ziel ist es einen generischen Klassifizierer zu finden, der dann für die jeweiligen Anwendungsfälle spezialisiert werden kann.

Methode

Analog zu [2] werden wir ein *GoogleNet* in der *Inception v3* [8] implementieren. Dazu werden wir ein fertiges Inception V3 Netz aus der *tensorflow.contrib.slim* Bibliothek verwenden, welches für die ImageNet Large Scale Visual Recognition Challenge [9] vor-trainiert wurde. Die vor-trainierten Gewichte werden von Slim zur Verfügung gestellt.
Um das Neuronale Netz, welches für die ImageNet Challenge trainiert wurde, an unser Problem anzupassen, werden wir die letzten Layer des Inception V3 Netzes anpassen. Dazu entfernen wir die letzen Layer und ersetzen diese durch eine vollständig verbundene Schicht, mit einem Outputlayer von zwei Neuronen (maligne, benigne).
Als Optimierer werden wir, analog zu Esteva et. al (2017) [2] anfangs RMSProp verwenden. Weitere Optimierer, wie beispielsweise Adam und SGD werden wir mit RMSProp vergleichen, um ein optimales Training zu gewährleisten.
Geplant ist, die Augmentierung der Daten in nativem Tensorflow Code zu implementieren und die Dataset-API von Tensorflow zu verwenden. Dies ermöglicht ein sehr effizientes Training. Jedoch werden wir als ersten Ansatz einen vereinfachte Methode zum Laden und Augmentieren der Daten umsetzen, um frühzeitig Netze trainieren zu können.
Als Fehlerfunktion während dem Training, werden wir anfangs einen einfachen 0,1-Loss nutzen. Es ist auch angedacht, andere Funktionen (L2-Loss) zu testen. Zur Vermeidung von Overfitting werden wir, wie Estava et. al [2], eine Kreuzvalidierung durchführen.

Optional: Android App

Damit der Klassifikator auch für jedermann nutzbar ist, soll falls wir noch genügend Zeit haben eine intuitiv bedienbare Android App entwickelt werden mit der ganz einfach über ein Foto gecheckt

werden kann ob man zum Art gehen sollte oder der potentielle Hautkrebs harmlos ist.

Kontakt

Name	Matrikelnummer	E-Mail Adresse
Florence Lopez	3878792	florence.lopez@student.uni-tuebingen.de
Jonas Einig	3766150	jonas.einig@student.uni-tuebingen.de
Julian Späth	3938726	julian.spaeth@student.uni-tuebingen.de

Literatur

[1] Rebecca L Siegel, Kimberly D Miller, and Ahmedin Jemal. Cancer statistics, 2016. CA: a cancer journal for clinicians, 66(1):7–30, 2016.

[2] Andre Esteva, Brett Kuprel, Roberto A Novoa, Justin Ko, Susan M Swetter, Helen M Blau, and Sebastian Thrun. Dermatologist-level classification of skin cancer with deep neural networks. Nature, 542(7639):115–118, 2017.

[3] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. Nature, 521(7553):436–444, 2015.

[4] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dan Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. Software available from tensorflow.org.

[5] Stéfan van der Walt, S Chris Colbert, and Gael Varoquaux. The numpy array: a structure for efficient numerical computation. Computing in Science & Engineering, 13(2):22–30, 2011.

[6] Android Studio. the official ide for android. Android Studio. URL: https://developer. android. com/studio/index. html(: 14.09. 2016), 2017.

[7] Memorial Sloan Kettering Cancer Center. Isic dermoscopic archive, 2017. Eingesehen am 26.11.2017.

[8] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jonathon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. CoRR, abs/1512.00567, 2015.

[9] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. International Journal of Computer Vision (IJCV), 115(3):211–252, 2015.