

# Pflanzenklassifizierung mit tiefen neuronalen Netzen

## Bachelorthesis & Projektarbeit im Studiengang Medieninformatik

Patrick Hidiringer  
Hochschule der Medien

ph047@hdm-stuttgart.de

**Betreuer:**

Prof. Dr. Johannes Maucher  
Andreas Stiegler

### Zusammenfassung

In dieser Arbeit soll überprüft werden, wie effektiv aktuelle tiefe neuronale Netze bei der Pflanzenartenkennung sind. Dafür wird die Netzwerkarchitektur "Inception ResNet" [Szegedy et al., 2016] verwendet, nachdem diese sehr erfolgreich in der *Imagenet Classification Challenge (ILSVRC)* angewendet wurde. Dabei soll nun zusätzlich der Trainings- und Evaluierungsdatensatz automatisch zusammengestellt und mithilfe eines weiteren neuronalen Netzes gefiltert werden.

## Einführung

Die immer weiter fortschreitende Entwicklung tiefer neuronaler Netze im Bereich der Bilderkennung erlaubt es, immer komplexere Objekte oder Sachverhalte zu klassifizieren. Inzwischen nähert sich deren Erkennungsrate schon stark an die des Menschen an [Karpathy, 2014]. Pflanzen als Klassifizierungsobjekt stellen jedoch noch immer, egal ob für den Computer oder den Mensch, eine Herausforderung dar, da sich diese zum einen oft kaum voneinander unterscheiden lassen, und zum anderen da sich gleiche Pflanzenspezies oft trotzdem sehr in der Form unterscheiden können. Wenn man also an die Grenzen von state-of-the-art neuronalen Netzen gehen möchte, ist Pflanzenklassifizierung eine gute Möglichkeit, da für ein akzeptables Klassifizierungsergebnis viele Optimierungen nötig sind.

Nun ist jedoch beim maschinellen Lernen oft ein großes Problem, dass man sehr viele Trainingsdaten benötigt. Meistens werden diese vor der Trainingsphase mühsam von Hand zusammengestellt und mit Labels versehen. Auch wenn Tools wie Amazons *Mechanical Turk* dies ein wenig erleichtern, so ist es doch wieder mit weiteren Kosten verbunden. Eine Idee wäre nun, gerade bei der Klassifizierung von Bildern, sich das Internet zu Nutze zu machen, und über Suchmaschinen oder andere Bild-Aggregatoren automatisch einen Datensatz zusammenzustellen.

## Ziele

1. Nutzung von state-of-the-art Deep-Learning Methoden zur Klassifizierung von Pflanzen
2. Entwicklung eines Tools zur automatischen Zusammenstellung von Datensätzen
3. Aufbauend auf Tensorflow, Implementierung einer generischen Trainingsumgebung
4. Entwicklung einer Web-Schnittstelle, um Bilder komfortabel mit dem eigenen neuronalen Netz zu klassifizieren
5. Experimente mit verschiedenen Netzwerk-Konfigurationen

## Datensatz

Eine größere Herausforderung bei der Klassifikation von Pflanzen ist die enorme Anzahl an Klassen. Bei den bisherigen *Imagenet Classification Challenges* wurden bei ähnlichen Aufgaben circa 2000 Klassen verwendet, ein großer Unterschied zu den Millionen von Pflanzen die es gibt.

Es müssen also ein paar Einschränkungen gemacht werden, am besten ohne dass man größere Einbußen bei der Nützlichkeit einer solchen Anwendung hat.

Eine solche Einschränkung ist beispielsweise, dass man sich nur auf die Pflanzen von einem Land konzentriert. Dies ist jedoch noch immer zu umfangreich, weshalb ein weiterer Schritt nötig ist. Pflanzen, wie alle biologischen Organismen, werden anhand ihrer Charakteristiken in einer taxonomischen Hierarchie (siehe als Beispiel Abbildung 1) eingeordnet. Nun kann man dies nutzen, um mehrere Pflanzen vom gleichen Typ zusammenzufassen. Damit konnte die Anzahl der Klassen auf bis zu 700 reduziert werden.

Bei dem Tool zur Erstellung eines Datensatzes ist die Eingabe nun schlicht eine Liste an Klassennamen, und die Ausgabe schließlich eine nach Namen sortierte Ordnerstruktur, welche jeweils die entsprechenden Bilder von dieser Klasse enthalten. Intern wird der Klassennamen als Suchbegriff für verschiedene Bilder-Aggregatoren wie Google, Bing, oder Flickr genutzt und die gefundenen Bilder heruntergeladen.

Kingdom:	Plantae
Division:	Pinophyta
Class:	Pinopsida
Order:	Pinales
Family:	Pinaceae
Genus:	<i>Abies</i>
Species:	<i>A. alba</i>

Abbildung 1: Taxonomie von *Abies Alba*, Wikipedia

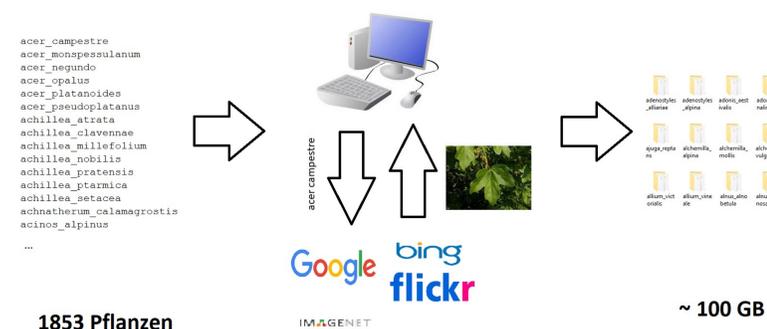


Abbildung 2: Pipeline der Datensatz-Erstellung

Anschließend wird noch für jede Klasse eine *self-organizing map* trainiert und die Gewichte abgespeichert, welche für spätere Filter notwendig sind.

## Architektur

Als Architektur wurde das von Google entwickelte "Inception ResNet" [Szegedy et al., 2016] verwendet, da es bei ähnlichen Aufgaben sehr gute Ergebnisse erzielte. Dieses verbindet herkömmliche *Convolutional Neural Networks*, welche aufgrund der Einbeziehung von räumlichen Abhängigkeiten gut für die Bildklassifizierung geeignet sind, mit *Residual Networks* [He et al., 2015], bei denen im Wesentlichen Inputs aus vorhergehenden Layern wiederverwendet werden und damit das sogenannte "vanishing gradient problem" bei sehr tiefen neuronalen Netzen bekämpft wird.

## Training

Als Framework wurde *Tensorflow* von Google verwendet, da dieses open-source Projekt bereits von einer großen Community genutzt wird und viele Features wie beispielsweise Multi-GPU-Training oder diverse Testing- und Visualisierungstools anbietet. Gleichzeitig ist es relativ "low-level", und ermöglicht dadurch viele Freiheiten bei der Optimierung.

Als Trainingsrechner stand der DeepLearning-Rechner der HdM zur Verfügung, welcher mit vier GeForce GTX 1080 mit jeweils 8GB RAM ausgestattet ist. Bei ungefähr 1 Millionen Bildern dauert es circa fünf Stunden bis eine Trainingsepoche abgeschlossen ist. Zwei bis drei Wochen später ist dann der gesamte Trainingsvorgang mit ungefähr 70 Epochen fertig. Dadurch erhält man unter anderem trainierte Gewichte für das Netzwerk, welche man anschließend für Evaluationszwecke nutzen kann.

Ziel ist es, möglichst oft mit verschiedenen Parametern zu trainieren, um Auswirkungen einzelner Komponenten zu erforschen.

## Evaluation

Bei der Evaluation wird ein spezieller Datensatz, welcher nicht beim Training verwendet wurde, mit den trainierten Gewichten klassifiziert. Dabei wird unter anderem eine sogenannte *Confusion Matrix* erzeugt, aus der man Accuracy, Precision, und Recall berechnen kann.

Beschreibung	Top1-Accuracy	Top5-Accuracy	Precision	Recall
Inception ResNet v1 mit Data Augmentation, Seed 1	0.545	0.741	0.632	0.501
Inception ResNet v1 mit Data Augmentation, Seed 2	0.542	0.741	0.629	0.492
Inception ResNet v1 ohne Data Augmentation, Seed 1	0.760	0.908	0.785	0.715
Inception ResNet v2 ohne Data Augmentation, Seed 1	?	?	?	?

Tabelle 1: Bisherige Evaluationsergebnisse

## Literatur

[He et al., 2015] He, K., Zhang, X., Ren, S., and Sun, J. (2015). Deep residual learning for image recognition. *CoRR*, abs/1512.03385.

[Karpathy, 2014] Karpathy, A. (2014). What i learned from competing against a convnet on imagenet. <http://karpathy.github.io/2014/09/02/what-i-learned-from-competing-against-a-convnet-on-imagenet/>, aufgerufen am 19.06.2017.

[Szegedy et al., 2016] Szegedy, C., Ioffe, S., and Vanhoucke, V. (2016). Inception-v4, inception-resnet and the impact of residual connections on learning. *CoRR*, abs/1602.07261.

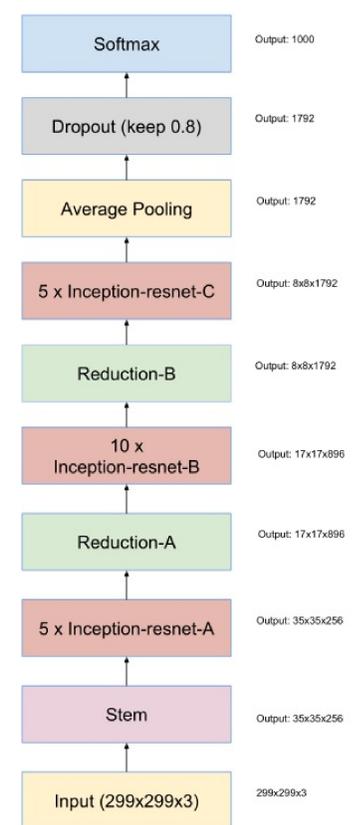


Abbildung 3: Schema des Inception ResNet