

KARLSRUHER INSTITUT FÜR TECHNOLOGIE

PFLICHTENHEFT

Neuronale Netze zur Bilderklassifizierung auf Heterogenen Plattformen

vorgelegt von:

Viet Doan Xuan Pham
Friedemann David Claus
Aleksandr Eismont
Jakub Marceli Trzcinski
Dmitrii Seletkov

Betreuer:

M.Sc. Dennis Weller
M.Sc. Sarath Mohanachandran Nair

6. Juli 2019

Inhaltsverzeichnis

1	Motivation	4
2	Zielbestimmung	5
2.1	Musskriterien	5
2.1.1	Backend	5
2.1.2	Frontend	5
2.2	Wunschkriterien	5
2.2.1	Backend	5
2.2.2	Frontend	6
2.3	Abgrenzungskriterien	6
3	Produkteinsatz	7
3.1	Anwendungsbereiche	7
3.2	Zielgruppen	7
3.3	Betriebsbedienungen	7
4	Produktumgebung	8
4.1	Software	8
4.2	Hardware	8
4.3	Orgware	8
5	Funktionale Anforderungen	9
5.1	Eingabe der Bilder	9
5.2	Einstellungen	9
5.3	Berechnungsausführung	9
5.4	Ergebnisanzeige	10
5.5	Transfer Learning	10
6	Produktdaten	11
6.1	Benutzerdaten	11
6.2	Programmdaten	11
7	Nicht-funktionale Anforderungen	12
8	Globale Testfälle	13
8.1	Basis-Testfälle	13
8.2	Folgende Funktionssequenzen sind zu überprüfen:	13
8.3	Folgende Datenkonsistenzen sind einzuhalten:	13
9	Systemmodelle	15
9.1	Szenarien	15
9.1.1	Hauptscenario: Klassifizierung eines Bildes	15

9.1.2	Trainieren eines ausgewählten Neuronalen Netzes	15
9.1.3	Wahl und Anwendung eines Neuronalen Netzes	15
9.2	Anwendungsfälle	16
9.2.1	Bilderklassifizierung und Transfer Learning	16
9.3	Benutzerschnittstelle	17
9.4	Dynamische Modelle	23
9.4.1	Aktivitätsdiagramm 1 - Nutzerinteraktion Klassifizierung	23
9.4.2	Aktivitätsdiagramm 2 - Nutzerinteraktion Transfer Learning	24
9.4.3	Sequenzdiagramm	25

1 Motivation

Dieses Projekt entsteht im Rahmen des Moduls „Praxis der Softwareentwicklung – PSE“ am Lehrstuhl Programmierparadigmen - IPD Snelting. Das Modul ist Teil des Bachelorstudiengangs Informatik am Karlsruher Institut für Technologie.

Die Bilderklassifizierung mittels Neuronaler Netze findet zunehmend in immer mehr Bereichen des Lebens Anwendung. In den letzten Jahren kam es, infolge der gestiegenen Rechenleistung als auch aufgrund des Vorhandenseins von großen Mengen an Trainingsdaten zu enormen Fortschritten bei der Bilderklassifizierung. Smartphones lassen sich mit der Gesichtserkennung entsperren, Röntgenbilder und CT-Bilder können auf Anomalien untersucht werden und Teslas Autopilot ebnet uns den Weg in das vollautonome Fahren, um nur ein paar von vielen Beispielen zu nennen.

Doch auch umstrittene Projekte sind durch diese Fortschritte möglich. Das Skynet Projekt in China zum Beispiel wurde in den weltweiten Medien stark kritisiert, da es der chinesischen Regierung eine Massenüberwachung der Bürger in Echtzeit ermöglicht. Hierzu wurden bereits über 170 Millionen Kameras landesweit in Betrieb genommen.

Ziel unseres Projektes ist es, essenzielle praktische Erfahrungen im Bereich der Softwareentwicklung sowie der Teamarbeit in der Softwareentwicklung zu sammeln. Außerdem werden wesentliche Kenntnisse über die Funktionsweise von künstlichen Neuronalen Netzen zur Bilderklassifizierung inklusive der Implementierung dieser auf verschiedenen Plattformen erlangt. Da auch in Zukunft Neuronale Netze eine immer größere Rolle spielen werden, haben wir uns für dieses Projekt entschieden.

2 Zielbestimmung

2.1 Musskriterien

2.1.1 Backend

- MK10 Bildklassifizierung mittels des vortrainierten Deep Neural Networks AlexNet
- MK20 Schnittstellenklassen zwischen Host PC (Desktop PC mit Xubuntu 16.04) zu Beschleuniger Intel Movidius Neural Compute Stick und CPU
- MK30 Klasse für die Schätzung der Performance (Bandweite, FLOPS), Leistungsverbrauch für jeden Beschleuniger und das jeweils implementierte Neuronale Netz
- MK40 Zuweisung der rechenintensiven Tasks zu Beschleunigern entsprechend des gewählten Modus und basierend auf den Schätzungen der Performance und des Leistungsverbrauchs
- MK50 Unterstützung von mehreren Intel Movidius Neural Compute Sticks
- MK60 Implementierung des Lernalgorithmus Transfer Learning

2.1.2 Frontend

- MK70 GUI auf einem System mit Xubuntu 16.04
- MK80 Wahl des Modus: Hohe Performance, Geringer Leistungsverbrauch, Hohe Energieeffizienz
- MK90 Wahl der zu klassifizierenden Bilder
- MK100 Anzeigen der Ergebnisse

2.2 Wunschkriterien

2.2.1 Backend

- WK10 GPU als weiteren Beschleuniger

WK20 Implementierung des Lernalgorithmusses Supervised Learning

WK30 Implementierung von Data Augmentation

WK40 Optimaler Modus (GPU + Intel Movidius Neural Compute Stick)

WK50 Weiteres Deep Neural Network (Unter Nutzung von Bibliotheken)

WK60 Visualisierung der Objektposition

WK70 Unterstützung der FPGA-Plattform

2.2.2 Frontend

WK80 Visualisierung der Topologien der Neuronalen Netze

2.3 Abgrenzungskriterien

AK10 Keine Echtzeitanforderungen

AK20 Produkt ist nicht für mobile Anwendung gedacht

3 Produkteinsatz

Das Produkt dient zur Bildklassifizierung auf heterogenen Plattformen mittels Neuronaler Netze (u.a. AlexNet). Es soll zwischen verschiedenen Betriebsmodi ausgewählt werden können. Zudem gibt es die Möglichkeit das vorhandene Neuronale Netz mittels Transfer Learning weiter zu trainieren.

3.1 Anwendungsbereiche

Personen verwenden diese Anwendung, um Bilder mittels eines CNNs zu klassifizieren oder CNNs weiter zu trainieren. Die Anwendung soll dem Nutzer ermöglichen, die Performance der Klassifizierung auf verschiedenen heterogenen Plattformen, unter Kenntnis der jeweiligen Leistungsaufnahme, einzusehen.

3.2 Zielgruppen

Personengruppen, die Bilder mittels des Neuronalen Netzes AlexNet klassifizieren wollen und dabei zwischen verschiedenen Plattformen auswählen und vergleichen möchten. Es werden keinerlei Vorkenntnisse vorausgesetzt.

3.3 Betriebsbedienungen

- Anwendung agiert ausschließlich in einer Büroumgebung
- Betriebszeit ist nicht vorgegeben
- Anwendung soll wartungsfrei sein

4 Produktumgebung

- Das Produkt läuft auf dem Host PC am CDNC Institut

4.1 Software

- Betriebssystem : Xubuntu 16.04
- Bilderdatenbank

4.2 Hardware

- PC mit einer dedizierten Grafikkarte
- Vier Intel Movidius Neural Compute Sticks

4.3 Orgware

- Netzwerkverbindung zum Herunterladen der Bilderdatenbank

5 Funktionale Anforderungen

5.1 Eingabe der Bilder

FA10 Eingabe der Bilder optional über Drag and Drop oder per Dateiauswahl

FA20 Falls die Eingabebilder nicht die passende Auflösung /NF160/ besitzen, werden sie entsprechend skaliert oder durch Cropping umgewandelt.

FA30 Anzeigen der Eingabebilder

5.2 Einstellungen

FA40 Wahl des Berechnungsmodus

- Hohe Performance: der schnellste und performanteste Modus, in dem alle verfügbaren heterogenen Plattformen benutzt werden
- Geringer Leistungsverbrauch: der Modus, in dem genau ein Intel Movidius Neural Compute Stick benutzt wird
- Hohe Energieeffizienz : der Modus, in dem alle verfügbaren Intel Movidius Neural Compute Sticks benutzt werden
- (optional) Optimal: der optimale Modus, in dem GPU und Intel Movidius Neural Compute Stick benutzt werden

FA50 Wahl des Neuronalen Netzes

5.3 Berechnungsausführung

FA60 Initiieren der Bilderklassifizierung

FA65 Batch-Processing: jede Recheneinheit arbeitet an jeweils einem Bild. Sobald eine Recheneinheit ein Bild fertig klassifiziert hat, bekommt sie das nächste Bild aus der plattformübergreifenden Warteschlange.

5.4 Ergebnisanzeige

FA70 Anzeigen der Wahrscheinlichkeiten der fünf wahrscheinlichsten Objektklassen (bzw. alle, wenn das Neuronale Netz weniger als fünf Objektklassen erkennen kann) zu jedem Bild

5.5 Transfer Learning

FA80 Eingabemöglichkeit eines Pfades einer neuen Bilderdatenbank inkl. der Labeldatei oder optional per Drag and Drop

FA90 Auswahl des Netzes, von dem aus weitergelernt werden soll

FA100 Initiieren des Transfer Learnings

FA110 Benennung des entstandenen Neuronalen Netzes

FA120 Entstandene Netze in /FA50/ auswählbar und benutzbar

6 Produktdaten

6.1 Benutzerdaten

PD10 Eingeladene Bilder

PD20 Dateipfade der eingelesenen Bilder

PD30 Bilder-Metadaten bei der /FA100/

6.2 Programmdateien

PD40 Ergebnisse der Bilderklassifizierungen mit den Wahrscheinlichkeiten

PD50 Ausgewählter Modus der Bilderklassifizierung /FA40/

PD60 Ausgewähltes Neuronales Netz

PD70 Auswählbare Neuronale Netze und deren Namen

PD80 Bilderdatensatz für das Trainieren des neuronalen Netzes, der in Unterdatensätzen Training, Validation und Test durch Cross Validation aufgeteilt wird

PD90 Ergebnisse des Trainierens: vom Programm anwendbare Matrizen mit Gewichten jeden Layers für Bildklassifizierung

7 Nicht-funktionale Anforderungen

- NF10 Implementierung in C++
- NF20 Entwurfsmuster: Model View Controller
- NF30 Entwurf in Unified Modeling Language (UML)
- NF40 ≈ 10.000 Code-Zeilen
- NF50 Objekt-orientierter Ansatz (40-80 Klassen ohne Schnittstellenklassen)
- NF60 Qualitätssicherung (BullseyeCoverage, CppUnit)
- NF70 Produktumgebung: Host PC am CDNC Institut
- NF80 Betriebssystem: Xubuntu 16.04
- NF90 Die Software kann mehrere Bilder gleichzeitig bearbeiten, wenn mehr als eine Berechnungseinheit verwendet wird. Sonst nur sequentiell mehrere Bilder.
- NF100 Der Nutzer muss in der Lage sein, ein Bild einzugeben, ohne bereits existierendes Neuronales Netz Neuronales Netz trainieren zu müssen.
- NF110 Ein Hilfe-Button zeigt eine kurze Beschreibung der Software.
- NF120 Daten sollen über eine graphische Oberfläche ein- und ausgegeben werden.
- NF130 Zuverlässigkeit auf Host PC
- NF140 Eingabe von einem Bild bis 1024 Bilder möglich (begrenzt)
- NF150 Erstellen von bis zu 64 Neuronalen Netzen durch Transfer Learning möglich
- NF160 Eingabebilder werden von der Anwendung auf die Bildgröße 256x256 Pixel gebracht
- NF170 Eingabebilder haben das Dateiformat JPG und PNG
- NF180 Eingabebilder haben das Dateigrößenlimit 10 MB

8 Globale Testfälle

8.1 Basis-Testfälle

T10 Überprüfung auf kompatible und vollständige Hardware des Host PCs

8.2 Folgende Funktionssequenzen sind zu überprüfen:

T20 Auswahl zwischen Bildklassifizierung oder Transfer Learning eines neutralen Netzes, Einfügen von einem Bild bzw. mehreren Bildern durch Dateipfadeingabe oder per Drag and Drop, Auswahl des Operationsmodus, Auswahl des Neuronalen Netzes, Initiieren der Klassifizierung / des Transfer Learnings sowie Visualisierung der Ergebnisse bzw. Darstellung der Verlustfunktion

T30 Speicherung eines trainierten Neuronalen Netzes

T40 Löschen von gespeicherten trainierten Neuronalen Netzen

8.3 Folgende Datenkonsistenzen sind einzuhalten:

T50 Die Klassifizierung kann nur dann ausgeführt werden wenn folgende Voraussetzungen erfüllt sind:

- Der Bildklassifizierungsmodus ist ausgewählt
- Es wurde mindestens ein zu klassifizierendes Bild hinzugefügt und die Anzahl der Eingabebilder erfüllt /NF140/
- Alle Eingabebilder haben das korrekte Dateiformat /NF170/ und sind nicht defekt
- Alle Eingabebilder überschreiten das Dateigrößenlimit /NF180/ nicht
- Alle Eingabebilder haben die korrekte Bildgröße /NF160/
- Es wurde ein Operationsmodus ausgewählt
- Es wurde ein Neuronales Netz ausgewählt

T60 Das Transfer Learning kann nur dann ausgeführt werden, wenn folgende Voraussetzungen erfüllt sind:

- Benutzer befindet sich im Reiter "Transfer Learning"
- Ein Trainingssatz wurde hinzugefügt
- Der Trainingssatz ist vollständig und korrekt (Dateiformate /NF170/, Dateigrößen /NF180/, Bildgröße /NF160/, Labels vorhanden)
- Die Anzahl von erstellbaren Neuronalen Netzen /NF150/ ist nicht überschritten

T70 Ein trainiertes Neuronales Netz kann nur dann gespeichert werden, wenn genügend Speicherplatz vorhanden ist

9 Systemmodelle

9.1 Szenarien

9.1.1 Hauptszenario: Klassifizierung eines Bildes

Der Nutzer wählt den Standardreiter „Klassifizierung“. Er fügt per Drag and Drop sein zu klassifizierendes Bild ein. Danach wählt er aus einer Liste der verfügbaren Netze das AlexNet aus. In dem mittleren Bereich oben werden alle möglichen Operations-Modi aufgelistet. Der Nutzer wählt „Hohe Energieeffizienz“ aus. Nach dem Knopfdruck „Start“ fängt der Prozess der Bildklassifizierung an. Der Nutzer erhält das Ergebnis, das jeweils aus den errechneten Wahrscheinlichkeiten der fünf zutreffendsten Objektklassen besteht rechts angezeigt.

9.1.2 Trainieren eines ausgewählten Neuronalen Netzes

Der Nutzer wählt den Reiter „Transfer Learning“. Er drückt auf den Knopf „Durchsuchen“ und wählt den Verzeichnispfad des Trainingssatzes aus. Danach wählt er aus einer Liste der verfügbaren Netze im oberen mittleren Bereich das unveränderte AlexNet aus. Darunter werden alle möglichen Operationsmodi aufgelistet. Der Nutzer wählt „Hohe-Performance“ aus. Nach dem Knopfdruck „Start“ fängt der Trainingsprozess an. Wenn der Nutzer den Prozess beenden möchte, drückt er auf den Knopf „Beenden“ und der Trainingsprozess wird anschließend beendet.

9.1.3 Wahl und Anwendung eines Neuronalen Netzes

Der Nutzer wählt den Standardreiter „Klassifizierung“. Danach wählt er ein bereits trainiertes Netz (8.1.2); die verfügbaren Neuronale Netze werden entsprechend markiert. Daraufhin fügt er seine zu klassifizierenden Bilder durch Dateipfadeingabe oder ein. Dann kann der Nutzer das Bild mittels des gewählten Netzes klassifizieren (8.1.1).

9.2 Anwendungsfälle

9.2.1 Bilderklassifizierung und Transfer Learning

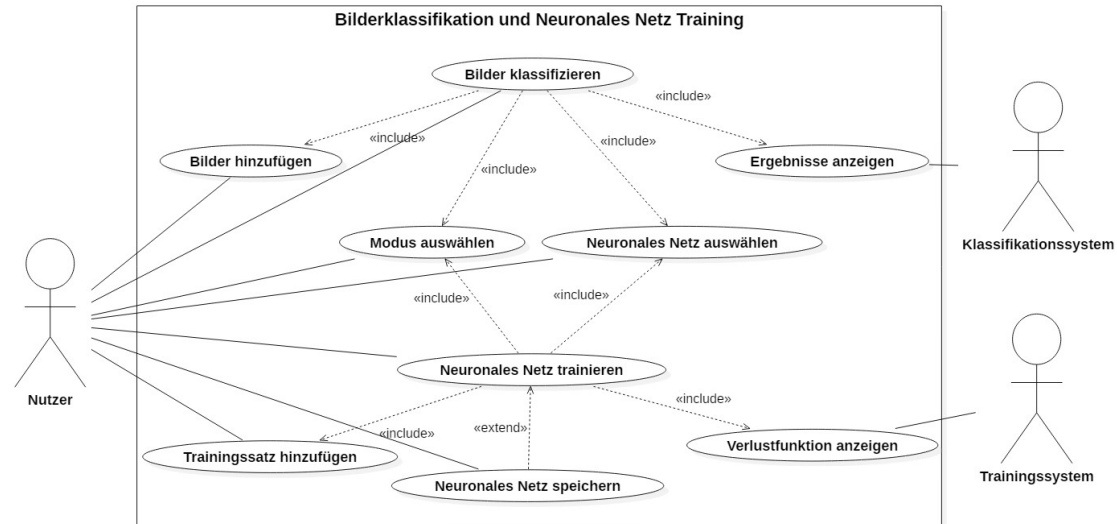


Abbildung 1: Das Anwendungsfalldiagramm.

Akteure: Nutzer, Klassifizierungssystem, Trainingssystem

Anwendungsfälle: Bilder klassifizieren, Transfer Learning, Neuronales Netz speichern, Bilder hinzufügen, Modus auswählen, Neuronales Netz auswählen, Ergebnisse anzeigen, Verlustfunktion anzeigen.

Beschreibung: In den beiden Anwendungsfällen Bilder klassifizieren und Neuronales Netz trainieren muss der Nutzer zunächst Bilder hinzufügen sowie den Modus und das Neuronale Netz auswählen. Je nach Anwendungsfall werden dann die Ergebnisse oder die Verlustfunktion angezeigt. Wurde ein Neuronales Netz trainiert, kann dieses im Anschluss abgespeichert werden.

9.3 Benutzerschnittstelle

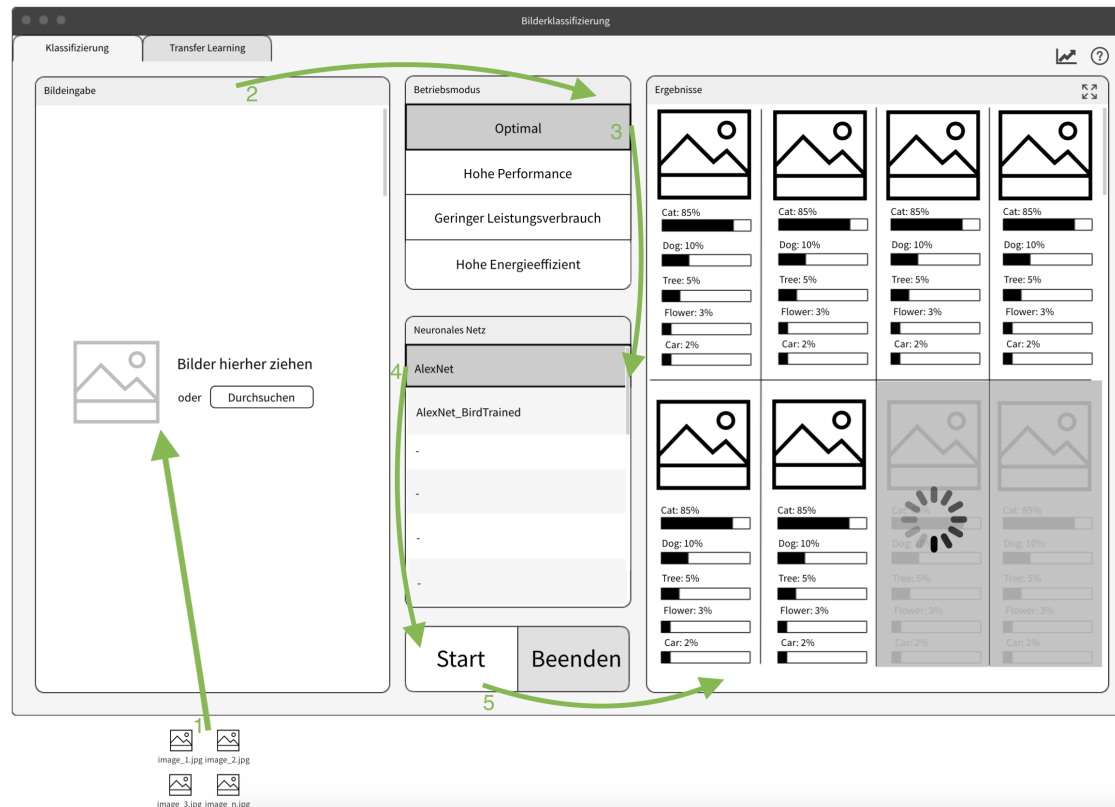


Abbildung 2: Klassifizierungsfenster



Abbildung 3: Transfer Learning Fenster

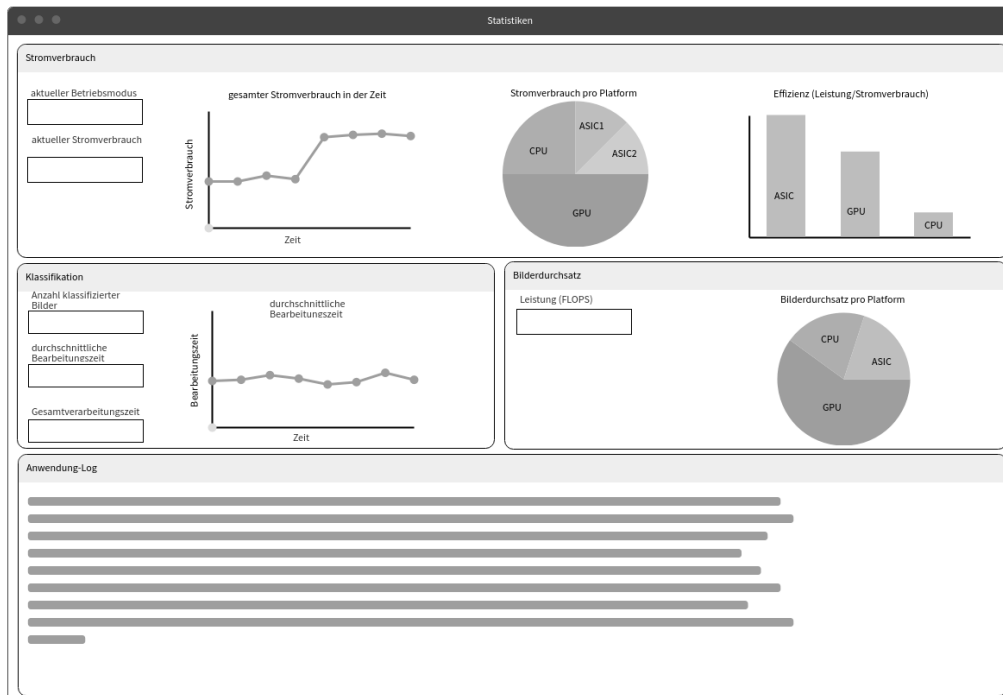


Abbildung 4: Statistiken Fenster

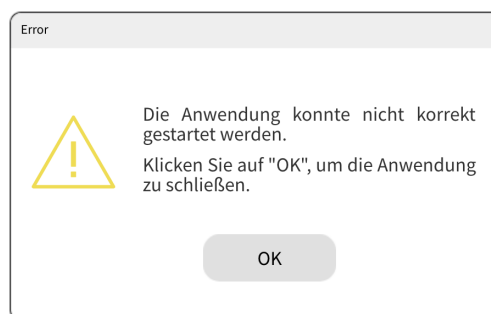


Abbildung 5: Die Fehlermeldung, wenn die Anwendung nicht korrekt gestartet werden konnte.

Das Produkt dient zur Bilderklassifizierung auf heterogenen Plattformen mittels neuronaler Netze u.a. AlexNet. Es soll zwischen verschiedenen Betriebsmodi ausgewählt werden können. Zudem gibt es die Möglichkeit das vorhandene neuronale Netz mittels Transfer Learning zu trainieren.

AlexNet ist der Name eines konvolutionellen neuronalen Netzwerks, das von Alex Krizhevsky entworfen und mit Ilya Sutskever und Krizhevskys PhD-Berater Geoffrey Hinton veröffentlicht wurde. AlexNet nahm am 30. September 2012 an der ImageNet Large Scale Visual Recognition Challenge teil.

Unter **Transfer Learning** versteht man das Übertragen der Ergebnisse eines fertig trainierten neuronalen Netzes auf eine neue Aufgabe. Die fertig trainierten Layer werden dabei entweder konstant gehalten und nur am Output-Layer nachtrainiert, oder es werden einige oder alle Layer auf Basis des aktuellen Trainings-Stands weiter trainiert.

Optimal Modus: der optimale Modus, in dem GPU und Intel Movidius Neural Compute Stick benutzt werden.

Hohe Performance: der schnellste und performanteste Modus, in dem alle verfügbaren heterogenen Plattformen benutzt werden.

Geringer Leistungsverbrauch: der Modus, in dem genau ein Intel Movidius Neural Compute Stick benutzt wird.

Hohe Energieeffizienz: der Modus, in dem alle verfügbaren Intel Movidius Neural Compute Sticks benutzt werden.

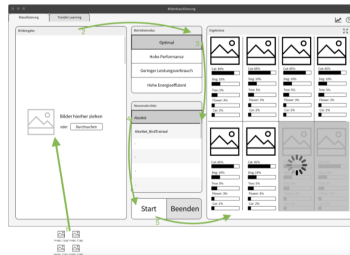


Abbildung 1: Schritt-für-Schritt-Anleitung zur Bildklassifizierung

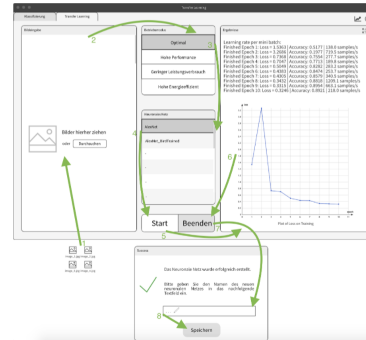


Abbildung 2: Schritt-für-Schritt-Anleitung zum Transfer Learning

OK

Abbildung 6: Das Hilfefenster zeigt eine kurze Beschreibung der Software.

Erläuterung zu den Abbildungen der Benutzeroberfläche:

Abbildung 2:

Im Standardreiter „Klassifizierung“ kann man im linken Bereich per Drag and Drop die zu klassifizierenden Bilder einfügen. Alternativ öffnet sich über den Button „Durchsuchen“ ein Fenster, mit dessen Hilfe man das Dateisystem durchsuchen und Bildpfade manuell angeben kann. Die geladenen Bilder mit der Auflösung /NF160/ werden im linken Bereich angezeigt. Werden mehr Bilder geladen, als in den Bereich passen, kann man über eine Leiste am rechten Rand des Bereichs nach oben und unten scrollen. Es können maximal /NF140/ Bilder in diesen Bereich geladen werden.

Im mittleren Bereich befinden sich oben vier Buttons, die jeweils einem Berechnungsmodus entsprechen und von denen immer genau einer gleichzeitig aktivierbar ist. Standardmäßig ist der „Hohe Performance Modus“ ausgewählt.

Darunter ist ein wenn nötig über die Leiste an dessen rechten Rands scrollbarer Bereich mit der Überschrift „Neuronales Netz“. In diesem lässt sich genau ein bereits trainiertes Netz auswählen. Hier werden auch die durch Transfer Learning trainierten und abgespeicherten Netze angezeigt. Im mittleren Bereich unten befindet sich ein Start-Button und ein Beenden-Button, welche die Klassifizierung starten bzw. beenden.

Im rechten Bereich werden nochmals die Bilder und darunter deren jeweiliges Ergebnis angezeigt. Das jeweilige Ergebnis besteht aus den errechneten Wahrscheinlichkeiten der fünf wahrscheinlichsten Objektklassen. Diese werden sowohl als Prozentzahl, als auch als entsprechend ausgefüllter Balken zu jeder Objektklasse und deren Namen dargestellt.

Analog zum linken Bereich erscheint wenn nötig eine Leiste zum Scrollen am rechten Rand des Bereichs und es werden entsprechend ebenfalls maximal /NF140/ Bilder und deren Ergebnisse in diesem Bereich angezeigt.

Dieser Bereich füllt sich sofort nach dem Betätigen des Start-Buttons, er ist aber zu Beginn noch vollständig ausgegraut. Das Bild bzw. je nach Anzahl der verwendeten Recheneinheiten die Bilder, die gerade (gleichzeitig) klassifiziert werden, werden durch einen Ladekreis kenntlich gemacht. Ist die Klassifizierung eines Bildes abgeschlossen, löst sich der Grauschleier über dessen Bereich und die Ergebnisse des Bildes werden angezeigt.

Unabhängig vom gewählten Reiter gibt es rechts oben einen Button für Statistiken (Graphsymbol) und einen Hilfebutton (Fragezeichensymbol). Beim daraufklicken öffnet sich jeweils ein entsprechendes Fenster. Diese sind in Abbildung 6 und 7 dargestellt und werden später noch genauer beschrieben.

Abbildung 3:

Im Reiter „Transfer Learning“ ist links ein zum Standardreiter analoger Bildeingabebereich eingerichtet. Beim Klick auf Durchsuchen wird man allerdings aufgefordert, einen Pfad zu einem Ordner (Verzeichnis) anzugeben, statt direkt Bilder auszuwählen. Analog kann man per Drag and Drop nur einen Ordner einfügen. Dies hat den Hintergrund, dass nicht nur Bilder, sondern auch deren Tags (Zuordnungen zu den Objektklassen) eingegeben werden müssen, welche sich im selben Ordner wie die Bilder befinden müssen.

In mittleren Bereich befindet sich wie beim Standardreiter oben die selbe Auswahl des Berechnungsmoduses. Eine Änderung des Moduses in einem der beiden Reiter wirkt sich auch auf den Modus des anderen Reiters aus.

Darunter wählt man das Neuronale Netz aus, von dem aus man Transfer Learning betreiben will. Dies kann auch ein durch Transfer Learning selbst neu erstelltes Netz sein. Der Bereich ist optisch und was die Bedienung betrifft analog zum Bereich Neuronales Netz aus dem Standardreiter.

Ganz unten im mittleren Bereich befindet sich ein Start-Button und ein Beenden-Button, welche das Transfer Learning starten beziehungsweise beenden.

Im rechten Bereich wird der Berechnungsfortschritt angezeigt. Dafür informiert oben ein Log nach jeder abgeschlossenen Epoche über deren Verlust, Genauigkeit und Geschwindigkeit (in Bildern pro Sekunde). Darunter wird die Verlustfunktion als Graph angezeigt.

Wenn das Transfer Learning über den Beenden-Button erfolgreich beendet wurde, erscheint ein Fenster „Erfolg“. In diesem wird mitgeteilt, dass das Neuronale Netz erfolgreich erstellt wurde und der Nutzer dieses nun benennen und speichern oder verwerfen kann. Entscheidet sich der Nutzer das Netz zu behalten, muss er dem Netz über das Texteingabefeld einen Namen geben und auf „Speichern“ drücken, sonst auf „Verwerfen“.

9.4 Dynamische Modelle

9.4.1 Aktivitätsdiagramm 1 - Nutzerinteraktion Klassifizierung

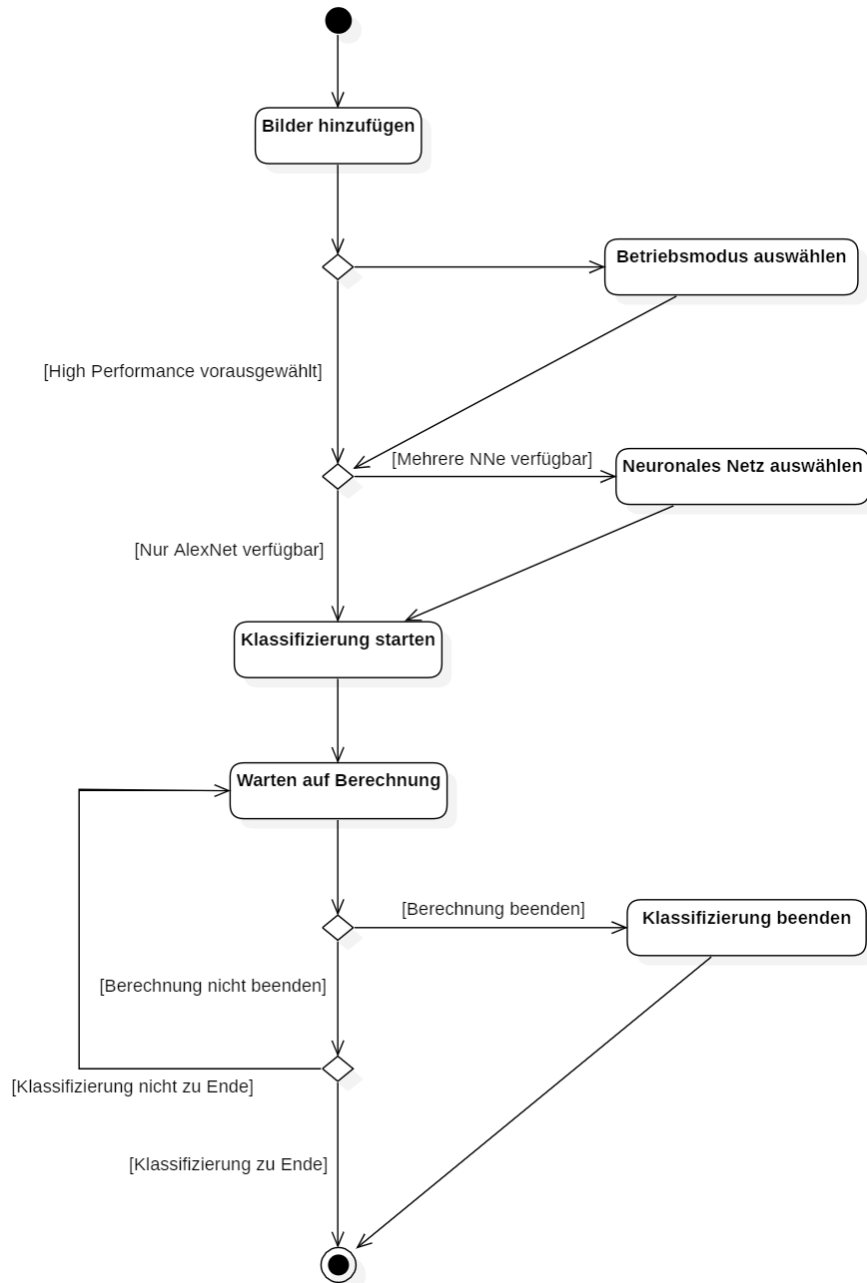


Abbildung 7: Das Aktivitätsdiagramm zeigt die Nutzerinteraktion beim Klassifizierungsprozess

9.4.2 Aktivitätsdiagramm 2 - Nutzerinteraktion Transfer Learning

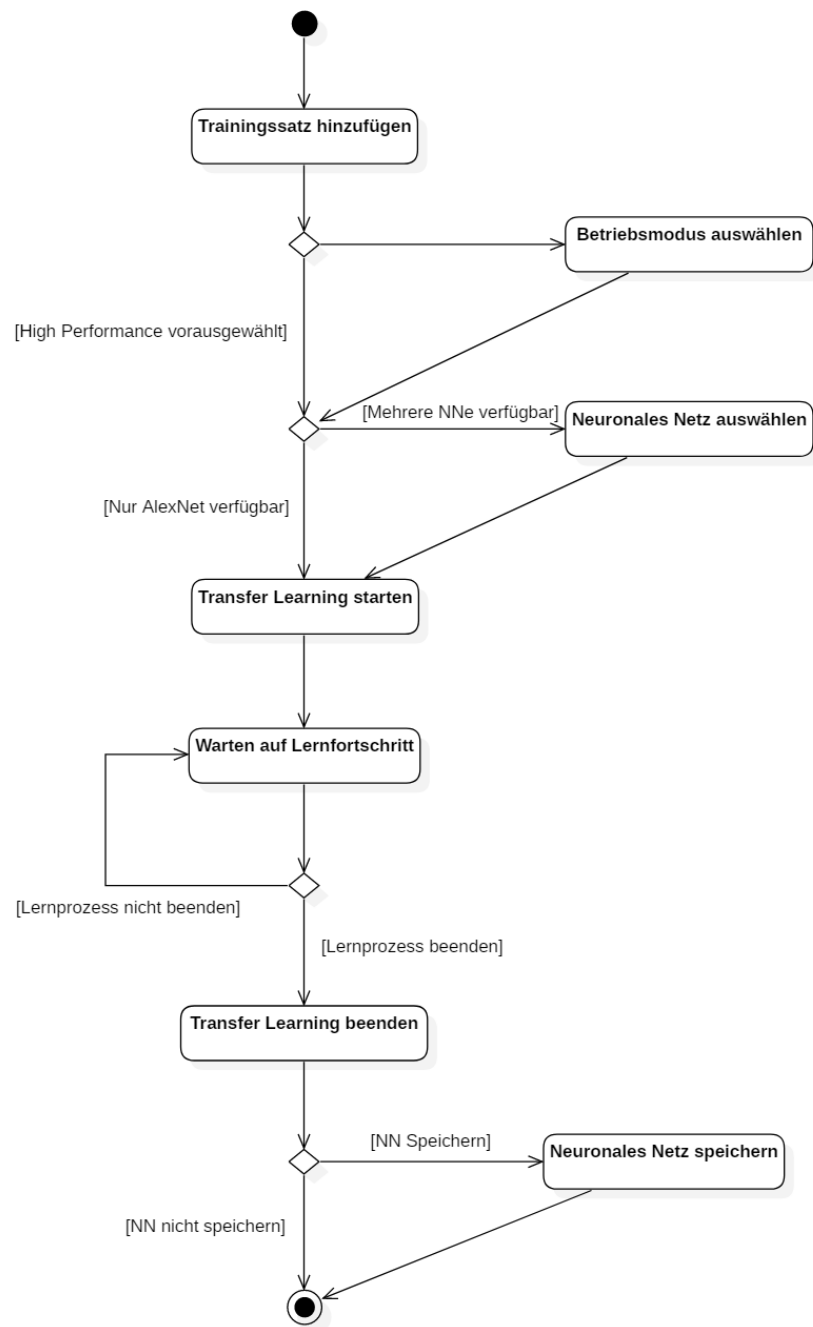


Abbildung 8: Das Aktivitätsdiagramm zeigt die Nutzerinteraktion beim Lernen mittels Transfer Learning

9.4.3 Sequenzdiagramm

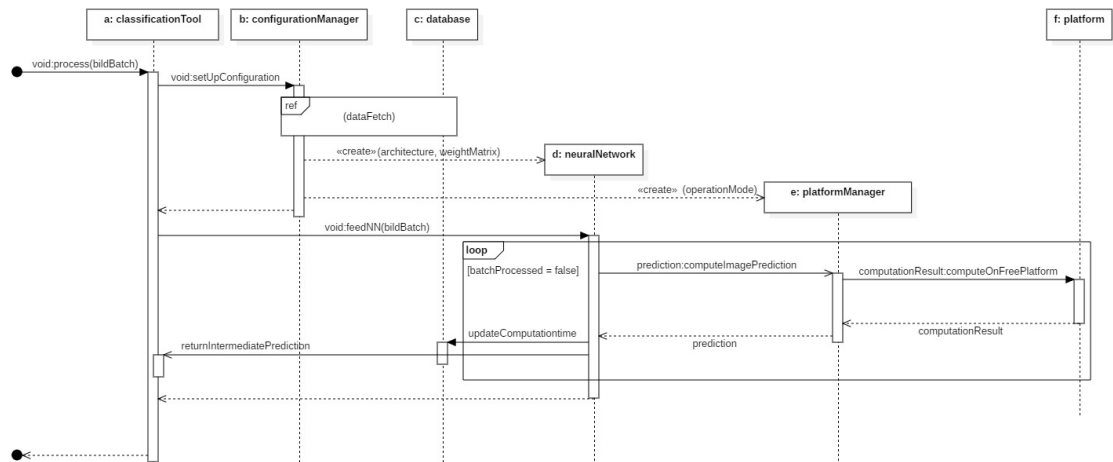


Abbildung 9: Das Sequenzdiagramm zeigt die interne Modellierung der Bildklassifizierung

Glossar

AlexNet AlexNet ist der Name eines konvolutionellen Neuronalen Netzwerks, das von Alex Krizhevsky entworfen und mit Ilya Sutskever und Krizhevskys PhD-Berater Geoffrey Hinton veröffentlicht wurde. AlexNet nahm am 30. September 2012 an der ImageNet Large Scale Visual Recognition Challenge teil¹.

CDNC Chair of Dependable Nano Computing (CDNC) at KIT - Karlsruhe Institute of Technology, Karlsruhe, Germany ².

CNN Ein Convolutional Neural Network, zu Deutsch etwa „faltendes Neuronales Netzwerk“, ist ein künstliches Neuronales Netz. Es handelt sich um ein von biologischen Prozessen inspiriertes Konzept im Bereich des maschinellen Lernens³.

CPU Central Processing Unit – englisch für zentrale Verarbeitungseinheit.

Cropping Cropping (englisch für abschneiden, stutzen) ist das Beschneiden von Bildrändern, um das Bildformat in ein anderes Format zu übernehmen, ohne dass dabei schwarze Ränder entstehen oder das Bild verzerrt werden muss⁴.

Cross Validation Cross Validation (englisch für Kreuzvalidierung) ist eine spezielle Methode, um die Qualität eines Modells zu bewerten. Sie basiert auf einer Aufteilung von dem verfügbaren Datensatz in Trainings-und-Validierungsteile. Die Validierungsteile werden von dem Modell erst zur Testzeit gesehen. Ihr Ziel ist Probleme wie overfitting (englisch für Überanpassung) oder selection bias (englisch für Stichprobenverzerrung) zu signalisieren⁵.

Data Augmentation Data Augmentation (englisch für Datensatzerweiterung) ist das Erweitern des Trainingssatzes, durch Verändern der Bilder ohne wesentliche Reduktion des Informationsgehalts z.B. durch Drehen oder Spiegeln⁶.

Drag and Drop Drag and Drop, deutsch „Ziehen und Ablegen“, ist eine Methode zur Bedienung grafischer Benutzeroberflächen von Rechnern durch das Bewegen grafischer Elemente mittels eines Zeigegerätes. Ursprünglich ist Drag and Drop eine Visualisierung von Kopieren und Einfügen für diverse Datenelemente⁷.

¹<https://en.wikipedia.org/wiki/AlexNet>

²<http://cdnc.itec.kit.edu/>

³https://de.wikipedia.org/wiki/Convolutional_Neural_Network

⁴<https://de.wikipedia.org/wiki/Cropping>

⁵[https://en.wikipedia.org/wiki/Cross-validation_\(statistics\)](https://en.wikipedia.org/wiki/Cross-validation_(statistics))

⁶<https://medium.com/nanonets/how-to-use-deep-learning-when-you-have-limited-data-part-2-data-augmentation>

⁷https://de.wikipedia.org/wiki/Drag_and_Drop

Epoche Beim Trainieren wird der Trainingsdatensatz mehrfach durchlaufen. Einen Durchlauf bezeichnet man auch als Epoche.

FLOPS Floating Point Operations Per Second (kurz FLOPS; englisch für Gleitkommaoperationen pro Sekunde) ist ein Maß für die Leistungsfähigkeit von Computern[1] oder Prozessoren und bezeichnet die Anzahl der Gleitkommazahl-Operationen (Additionen oder Multiplikationen), die von ihnen pro Sekunde ausgeführt werden können ⁸.

FPGA Ein Field Programmable Gate Array (FPGA) ist ein integrierter Schaltkreis (IC) der Digitaltechnik, in welchen eine logische Schaltung geladen werden kann ⁹.

Gewicht Das Gewicht drückt die Stärke der Verbindung zwischen zwei Neuronen aus, die durch eine reelle Zahl dargestellt wird.

GPU Graphics Processing Unit – englisch für Grafikprozessor.

GUI Eine GUI - englisch für graphical user interface - ist die graphische Benutzeroberfläche eines Programms.

Host PC Der Host PC ist ein von CDNC Institut bereitgestellter PC, auf dem die Xubuntu 16.04 Desktopumgebung installiert ist.

Intel Movidius Neural Compute Stick Der Intel Movidius Neural Compute Stick (NCS) ist ein winziges lüfterloses Deep Learning-USB-Laufwerk, das zum Erlernen der AI-Programmierung entwickelt wurde. Der NCS wird von der stromsparenden und leistungsstarken Movidius Visual Processing Unit (VPU) angetrieben ¹⁰.

JPG Weit verbreitetes Dateiformat für Bilder und Grafiken.

Layer Ein Layer ist eine schichtweise Anordnung der Neuronen eines neuronalen Netzes.

Model View Controller Model View Controller (MVC, englisch für Modell-Präsentation-Steuerung) ist ein Muster zur Unterteilung einer Software in die drei Komponenten Datenmodell, Präsentation und Programmsteuerung. Ziel des Musters ist ein flexibler Programmentwurf, der eine spätere Änderung oder Erweiterung erleichtert und eine Wiederverwendbarkeit der einzelnen Komponenten ermöglicht ¹¹.

⁸https://de.wikipedia.org/wiki/Floating_Point_Operations_Per_Second

⁹https://de.wikipedia.org/wiki/Field_Programmable_Gate_Array

¹⁰<https://www.mouser.de/new/Intel/intel-movidius-stick/>

¹¹https://de.wikipedia.org/wiki/Model_View_Controller

Neuronales Netz Ein (künstliches) Neuronales Netz oder (künstliches) Neuronales Netzwerk ist ein Netzwerk aus einfachen (üblicherweise adaptiven) Elementen (Neuronen) in hierarchischer Anordnung oder Organisation (Layers), die mit der Welt in der selben Art wie biologische Nervensysteme interagieren sollen ¹².

Nutzer Eine Person, die die Anwendung benutzt.

PNG Weit verbreitetes verlustfreies Dateiformat für Bilder und Grafiken.

Supervised Learning Das Supervised Learning (englisch für überwachtes Lernen) ist ein Oberbegriff von Lernverfahren für Neuronale Netze. Die Trainingsdaten sind bereits gelabelt und im Gegensatz zu Transfer Learning alle Schichten des Netzes trainiert werden. ¹³.

Transfer Learning Unter Transfer Learning versteht man das Übertragen der Ergebnisse eines fertig trainierten Neuronalen Netzes auf eine neue Aufgabe. Die fertig trainierten Layer werden dabei entweder konstant gehalten und nur am Output-Layer nachtrainiert, oder es werden einige oder alle Layer auf Basis des aktuellen Trainings-Stands weiter trainiert. ¹⁴.

UML Die Unified Modeling Language (vereinheitlichte Modellierungssprache), ist eine grafische Modellierungssprache zur Spezifikation, Konstruktion und Dokumentation von Software-Teilen und anderen Systemen. UML ist heute die dominierende Sprache für die Softwaresystem-Modellierung¹⁵.

Xfce Xfce (gesprochen: "eks-ef-si-i") ist eine recht bunte und gut konfigurierbare Desktop-Umgebung. Besonders auf älteren Rechnern ist Xfce eine Alternative zu GNOME oder KDE, vor allem wegen der höheren Geschwindigkeit bzw. des geringeren Speicherbedarfs ¹⁶.

Xubuntu Xubuntu ist eine offizielle Variante von Ubuntu mit der Desktop-Umgebung Xfce ¹⁷.

¹²Kohonen 1984

¹³<http://www.datenbanken-verstehen.de/lexikon/supervised-learning/>

¹⁴<https://jaai.de/transfer-learning-1739/>

¹⁵https://de.wikipedia.org/wiki/Unified_Modeling_Language

¹⁶<https://wiki.ubuntuusers.de/Xfce//>

¹⁷<https://wiki.ubuntuusers.de/Xubuntu/>