

**Hochschule Karlsruhe
Technik und Wirtschaft**
UNIVERSITY OF APPLIED SCIENCES

GENETISCHE ALGORITHMEN ZUR OPTIMIERUNG VON HYPERPARAMETERN EINES KÜNSTLICHEN NEURONALEN NETZES

Fakultät für Maschinenbau und Mechatronik
der Hochschule Karlsruhe
Technik und Wirtschaft

Bachelorarbeit

vom 01.03.2018 bis zum 31.08.2018
vorgelegt von

Christian Heinzmann

geboren am 18.02.1995 in Heilbronn
Matrikelnummer: 52550

Winter Semester 2019

Professor	Prof. Dr.-Ing. habil. Burghart
Co-Professor	Prof. Dr.-Ing. Olawsky
Betreuer FZI:	M. Sc. Kohout

Eigenständigkeitserklärung und Hinweis auf verwendete Hilfsmittel

Eigenständigkeitserklärung und Hinweis auf verwendete Hilfsmittel. Hiermit bestätige ich, dass ich den vorliegenden Praxissemesterbericht selbstständig verfasst und keine anderen als die angegebenen Hilfsmittel benutzt habe. Die Stellen des Berichts, die dem Wortlaut oder dem Sinn nach anderen Werken entnommen sind, wurden unter Angabe der Quelle kenntlich gemacht.

Datum: _____ Unterschrift: _____

Danksagungen

An dieser Stelle möchte ich mich zuerst bei dem Forschungszentrum Informatik bedanken, durch die ich die Möglichkeit bekommen habe, mich in einem innovativen Forschungszentrum auf meiner persönlichen und fachlichen Ebene weiter zu entwickeln. Außerdem bedanke ich mich bei sämtlichen Kollegen der Abteilung Embedded Systems and Sensors Engineering, die mich jederzeit hilfsbereit und kompetent unterstützt haben. Mein besonderer Dank gilt Herrn Lukas Kohout, welcher sich als Betreuer immer die nötige Zeit nahm, Sachverhalte zu erklären und meine Fragen zu beantworten.

Ausschreibung

BACHELORARBEIT

Genetische Algorithmen zur Optimierung von Hyperparametern eines künstlichen neuronalen Netzes

Zur Vermeidung der weiteren Ausbreitung von multiresistenten Keimen in Einrichtungen des Gesundheitswesens (insb. Krankenhäuser) werden am FZI Systeme und Methoden entwickelt, welche helfen sollen die Händehygiene-Compliance von Mitarbeitern dort zu erhöhen. Durch technische Unterstützung bei häufig wiederkehrenden Maßnahmen, soll mehr Zeit geschaffen und gleichzeitig auf die Wichtigkeit von Desinfektionsmaßnahmen aufmerksam gemacht werden. Dies soll mit Hilfe von Augmented-Reality umgesetzt werden. Dabei ist das Ziel, bekannte Prozesse, wie beispielsweise den Wechsel von postoperativen Wundverbänden, visuell zu unterstützen und automatisch zu dokumentieren.

AUFGABEN

Im Kontext der Detektion von Aktionen und Objekten, soll die Optimierung von Neuronalen Netzen mit Genetischen Algorithmen durchgeführt werden. Das Ziel hierbei ist es, ein Framework zu entwickeln und zu implementieren, in welchem künstliche neuronale Netze automatisiert trainiert werden. Unter anderem sollen die Hyperparameter mit Hilfe von Genetischen Algorithmen intelligent angepasst und das trainierte Netz anschließend ausgewertet werden. Diese Aufgaben sollen voll automatisiert ablaufen. Daraus ergeben sich folgende Aufgaben:

- Literaturrecherche über aktuelle Genetische Algorithmen und aktuelle Neuronale Netze
- Einarbeiten in vorhandene Frameworks für Genetische Algorithmen und Neuronale Netze
- Konzeptionierung und Implementierung des ausgewählten Ansatzes zur Optimierung von Hyperparameter des Neuronalen Netzes
- Evaluation und Auswertung speziell unter der Beachtung geringer Datenmengen
- Wissenschaftliche Aufbereitung und Dokumentation des Projekts

WIR BIETEN

- Aktuelle Softwaretools im täglichen wissenschaftlichen Einsatz
- eine angenehme Arbeitsatmosphäre
- konstruktive Zusammenarbeit

WIR ERWARTEN

- Grundkenntnisse in maschinellem Lernen
- Kenntnisse in folgenden Bereichen sind von Vorteil: Python, Tensorflow, C/C++
- selbständiges Denken und Arbeiten
- sehr gute Deutsch- oder Englischkenntnisse
- Motivation und Engagement

ERFORDERLICHE UNTERLAGEN

Wir freuen uns auf Ihre PDF-Bewerbung an Herrn Lukas Kohout, kohout@fzi.de, mit folgenden Unterlagen:

- aktueller Notenauszug
- tabellarischer Lebenslauf

WEITERE INFORMATIONEN

- Start: ab sofort

Inhaltsverzeichnis

Eigenständigkeitserklärung und Hinweis auf verwendete Hilfsmittel	2
Danksagungen	2
Ausschreibung	3
1 Einleitung zum Forschungs Zentrum Informatik	6
2 Der FZI-Forschungsbereich ESS	6
3 Einleitung zum Projekt	7
4 Meine Aufgaben	8
5 Grundlagen Neuronaler Netze	9
5.1 Basics	9
5.2 Aufbau eines Neurons	10
5.2.1 Input	10
5.2.2 Bias	10
5.2.3 Activation Function	10
5.2.4 Output	11
5.3 Lossfunction	11
5.4 Backpropagation	11
5.5 Learning Methods	11
5.5.1 Supervised Learning	11
5.5.2 Unsupervised Learning	11
5.5.3 Genetic Algorithmen	12
6 Grundlagen Convolutional Neuronal Network (CNN)	12
6.1 Convolutional Layer	13
6.2 Pooling Layer	13
6.3 Fully Connected Layer	13
6.4 3D Convolutional Network	14
7 Chronologischer Tätigkeitsabriss	15
8 Internetrecherche	15
9 Einarbeiten in vorhandene Frameworks	16
10 Optical Flow	17
11 Erstellen von Trainings- und Testdaten (Datenset)	17

12 Trainieren eines 3D-ConvNets	18
13 Ergebnisse der Actionserkennung	19
14 Verbesserungsvorschläge und Zukunftsaussicht	19
14.0.1 Erweiterung der Trainingsdaten	19
14.0.2 Realtime-Anwendung	19
14.0.3 Optical Flow	19
14.0.4 Dritter Stream mit Hnden	20

Abbildungsverzeichnis

1	Ablaufdiagramm Verbandswechsel [?]	8
2	Neuronales Netz [?]	9
3	Aufbau eines Neurons [?]	10
4	Convolutional Network [?]	12
5	Feature Map [?]	13
6	Pooling [?]	14
7	GANTT-Diagramm [?]	15
8	Two-Stream 3D-ConvNet [?]	16
9	Optical Flow [?]	17
10	Segmentation Hands [?]	20
11	Hand-Pose with RGB-D-Images [?]	20

Abkürzungsverzeichnis

1 Einleitung zum Forschungszentrum Informatik

„Das FZI Forschungszentrum Informatik am Karlsruher Institut für Technologie ist eine gemeinnützige Einrichtung für Informatik-Anwendungsforschung und Technologietransfer. Es bringt die neuesten wissenschaftlichen Erkenntnisse der Informationstechnologie in Unternehmen und öffentliche Einrichtungen und qualifiziert junge Menschen für eine akademische und wirtschaftliche Karriere oder den Sprung in die Selbstständigkeit.“ [?]

2 Der FZI-Forschungsbereich ESS

„Der Forschungsbereich Embedded Systems and Sensors Engineering (ESS) beschäftigt sich mit innovativen Technologien, Entwurfsmethoden und Anwendungen für und von eingebetteten Systemen. Von modellbasierten Entwurfsmethoden und -werkzeugen über technologieorientierte Forschung bis hin zu anwendungsorientierten Forschungs- und Entwicklungsprojekten – wir gestalten und entwickeln praxistaugliche Anwendungen rund um eingebettete Systeme und evaluieren diese.

Die breite Technologie- und Systemkompetenz aus Elektronik, Software-Engineering, Optik und Optoelektronik, Mikrosystemtechnik und Sensorik ist ein Alleinstellungsmerkmal des Bereiches. Schwerpunkte der Arbeiten bilden dementsprechend vor allem stark interdisziplinäre, Technologiebergreifende Forschungsprojekte und Anwendungen von eingebetteten Systemen in der Automobilelektronik, der Industrieautomation und im Gesundheits- und Sozialwesen.

Der Bereich ESS deckt mit seinen verfügbaren Kompetenzen dabei das komplette Spektrum der Entwicklung eingebetteter Systeme und Cyber Physical Systems (CPS) mit heterogenen Komponenten aus Mikroelektronik, Mikrooptik, Mikromechanik und Telematik ab.“ [?]

3 Einleitung zum Projekt

”Nach einer Studie der Charit aus dem Jahr 2015 sterben in Europa jährlich 23.000 Menschen an den Folgen einer Infektion mit multiresistenten Keimen. Die Tendenz ist dabei steigend. Hauptursache für die Ausbreitung dieser Keime, wie beispielsweise MRSA, ist eine mangelnde Hygiene der Angestellten in den Versorgungseinrichtungen beim Umgang mit den Patienten. Gründe dafür liegen im fehlenden Problembewusstsein, der zu hohen Arbeitsdichte und damit verbundenem Zeitmangel und der mangelnden Qualifikation der beteiligten Pflegekräfte.

Ziel des Projekts HEIKE ist es neue, technikgestützte Möglichkeiten zu entwickeln, welche die behandelnden Mitarbeiter im Krankenhausumfeld bei Maßnahmen am Patienten unterstützen und dadurch deren Compliance in Bezug auf die Händedesinfektion erhöhen.

Die Grundlage bilden ein mobiler, vernetzter Desinfektionsspender sowie Augmented-Reality-Technik. Die Technologien werden in dem Projekt weiterentwickelt und in einem System integriert, welches automatisch die durchgeführten Handlungen am Patienten erkennt und basierend darauf zusätzliche Informationen zur Verfügung stellt. Schließlich werden die durchgeführten Maßnahmen automatisch im System dokumentiert, was den Verwaltungsoverhead für das operative Personal verringert.”[?]

4 Meine Aufgaben

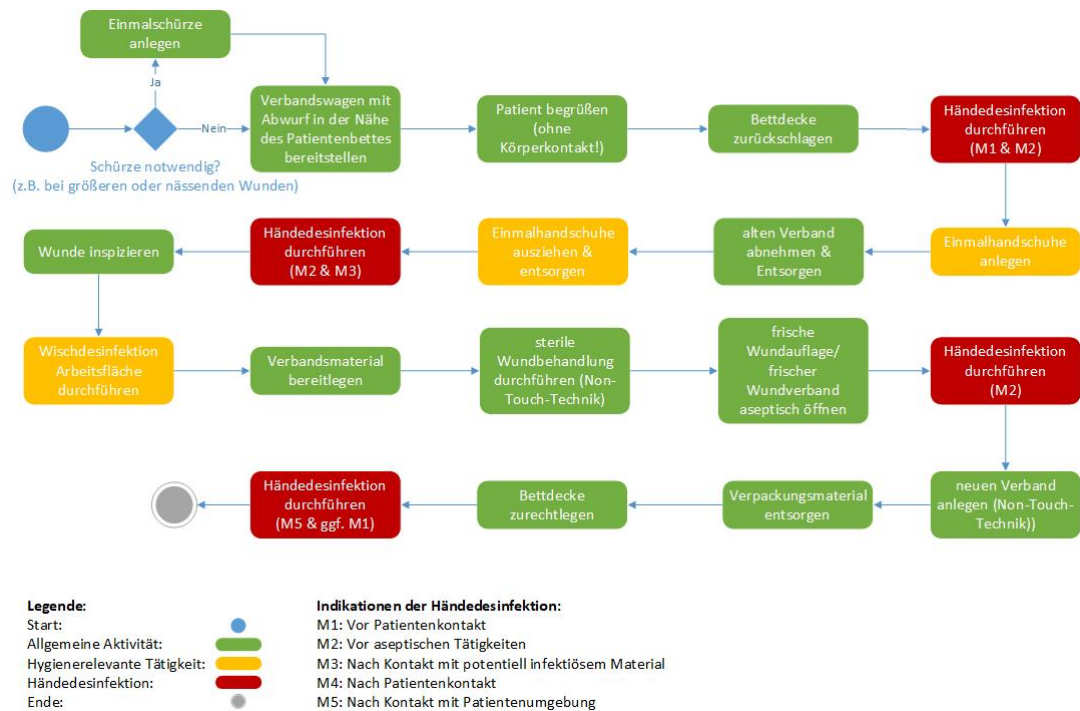


Abbildung 1: Ablaufdiagramm Verbandswechsel [?]

Nun war die Aufgabe herauszufinden, welche Möglichkeiten es gibt, diese Aktionen, die per EGO-Perspektiv-Video der Hololens aufgenommen wurden, zu erkennen. Die Wichtigsten Aktionen sind hierbei das Desinfizieren und anlegen der Handschuhe. 1 Hierfür soll ein Künstliches Neuronales Netz (KNN) zuhulfe gezogen werden, welches die wichtigen Aktionen erkennt und somit die Fehlerquote, welche durch Vergessen oder Zeitstress entstehen, zu reduzieren. Dazu muss erst ein Netz, welches für unser Anwendung passt, gefunden und für die oben genannten Anwendungen getestet werden, desweiteren muss hierfür ein Datenset mit diesen Aktionen erstellt werden, welche zum Trainieren und Testen des Netzes notwendig sind. Anschließend soll das Netz noch evaluiert und Verbesserungen vorgenommen werden.

5 Grundlagen Neuronaler Netze

5.1 Basics

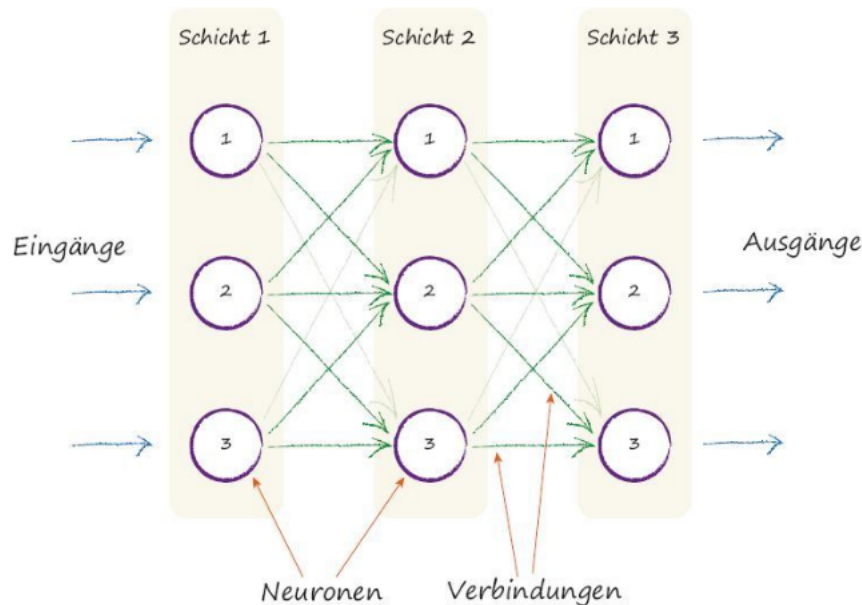


Abbildung 2: Neuronales Netz [?]

Zu sehen ist ein Künstliches Neuronales Netz mit drei Schichten. Dies wurde dem natürlichen Vorbild der neuronalen Netze im Gehirn nachempfunden. Die Kreise nennt man Neuronen, mehrere Neuronen zusammen ergeben eine Schicht oder auch Layer genannt. Die Verbindungen sind die Gewichte, so kann einem Netz verschiedene Zusammenhänge von Input und Output antrainiert bzw. angelernt werden. Zum Training werden viele Daten benötigt, aus welchen das Netz „Lernt“. Dafür ist es wichtig, viele aufbereitete Daten zu besitzen, denn diese Netze brauchen viele Trainingsiterationen, bis das gewünschte Ergebnis zustande kommt. Ein Neuron besteht aus Eingängen, Gewichten und einer Aktivierungsfunktion sowie einem Ausgang. Die Vernetzung mehrerer Neuronen lässt ein Neuronales Netz entstehen.

5.2 Aufbau eines Neurons

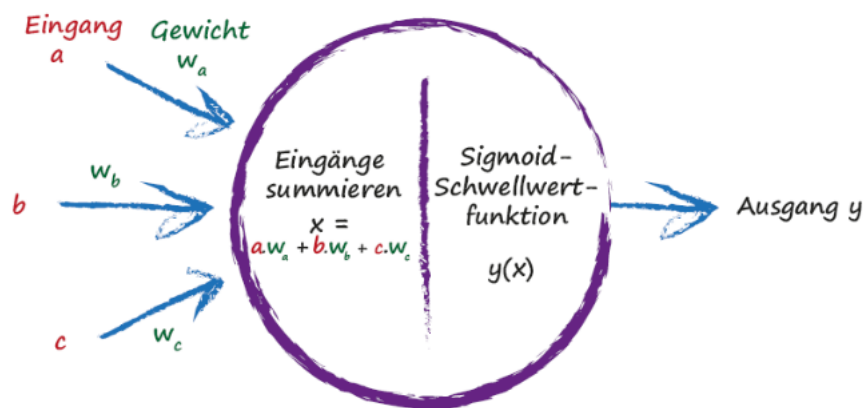


Abbildung 3: Aufbau eines Neurons [?]

5.2.1 Input

Der Input wird mit den einzelnen Gewichten verrechnet, anschließend werden diese zusammen gebracht. Diese Werte werden zufällig initialisiert und per Training verbessert, somit handelt es sich um einen angelerten Werte, welche durch die Backproagation (Fehlerrckfhrung) verbessert werden.

5.2.2 Bias

Auf diesen Input wird anschlieend ein Bias gerechnet, dieser fhrt zu einem besseren Verhalten beim Trainieren. Bei diesen Werten handelt es sich um angelerte Werte, die per Backpropagation verbessert werden und die Flexibitlitt der Netze erhht.

5.2.3 Activation Function

Die Aktivierungsfunktion kann man sich als Schwellwert vorstellen, ab wann das Neuron den Input weiter gibt. Es gibt verschiedene Funktionen, um diesen Schwellwert zu definieren. Bei Klassifizierungen werden heute meist ReLu-Layer oder ein Weakly-ReLu Layer benutzt, diese verhindern das Vanishing- bzw. Exploding- gradientproblem beim Trainieren.

5.2.4 Output

Wenn der Schwellwert berschritten wird, wird am Output durchgeschaltet. Von Input nach Output nennt sich ein Single-Forward-Pass. Wie hier beschrieben wird, kann ein Netz verschieden viele Layer besitzen mit verschiedenen Anzahlen von Neuronen.

5.3 Lossfunction

Die Verlustfunktion stellt ein ausgesuchtes Ma der Diskrepanz zwischen den beobachteten und den vorhergesagten Daten dar. Sie bestimmt die Leistungsfhigkeit des neuronalen Netzes whrend des Trainings und der Ausfhrung. Ziel ist es, im laufenden Prozess der Modellanpassung, die Verlustfunktion zu minimieren.

5.4 Backpropagation

Um die Fehlerfunktion zu minimieren wird als Tool Gradienten Abstieg benutzt. Im Grunde werden dabei die Gewichte so angepasst, dass ein besseres Ergebnis entsteht und dadurch die Fehlerfunktion verringert wird. Wie das Wort Backpropagation schon sagt, wird von hinten nach vorne verbessert. Es gibt verschiedene Variationen von Gradientenabstiegen, welche verschiedene Vor- und Nachteile haben. Bei dem Trainieren des Netzes wurde der Momentum-Optimizer, welcher aus einem Gradientenabstieg mit Momentum aufgebaut ist.

5.5 Learning Methods

5.5.1 Supervised Learning

berwachtes Lernen wird dafr benutzt, eine Funktion zu finden, Daten einem Wert zuzuweisen. Dennoch mssen dafr alle Daten vorverarbeitet werden und einem Label zugeordnet werden. Damit das Netz auch eine Aussage ber das Ergebnis, whrend des Trainings, geben kann. Anwendungsfle sind Regression, Klassifikation und Empfehlungen.

5.5.2 Unsupervised Learning

Im Vergleich zum bewachten Lernen liegen hier keine Labelinformationen vor. Weshalb dieser Ansatz eher zum Erkennen von Mustern und Ableiten von Regeln da ist. Fr unsu-

pervised Learning Algorithmen sind in der Regel sehr viele Daten ntig. Anwendungsgebiete sind das Clustering und die Dimensionsreduktion.

5.5.3 Genetic Algorithmen

Man kombiniert Genetic Algorithmen mit KNNs, um den Vorteil aus beiden Modulen zu bekommen. Dennoch ist es nicht der praktischste Ansatz. Durch die Genetic Algorithmen mssen die Hyperparameter der KNN nicht vom Programmierer bestimmt werden, sondern werde durch das Training einer ganzen Population von Netzen natrlich ausgewhlt. Es gibt viele verschiedene Mglichkeiten, um GA umzusetzen. Eine Mglichkeit ist es, die Netze mit den besten Ergebnissen herauss zu filtern, um sie dann fr die nchste Epoche mit einer kleinen zuflligen Mutation zu versehen oder die besten zwei Netze zu kreuzen um somit die nchste Polupation zubekommen. Diese Learning Methoden werden zum Beispiel fr Path-Finding-Aufgaben benutzt.

6 Grundlagen Convolutional Neuronal Network (CNN)

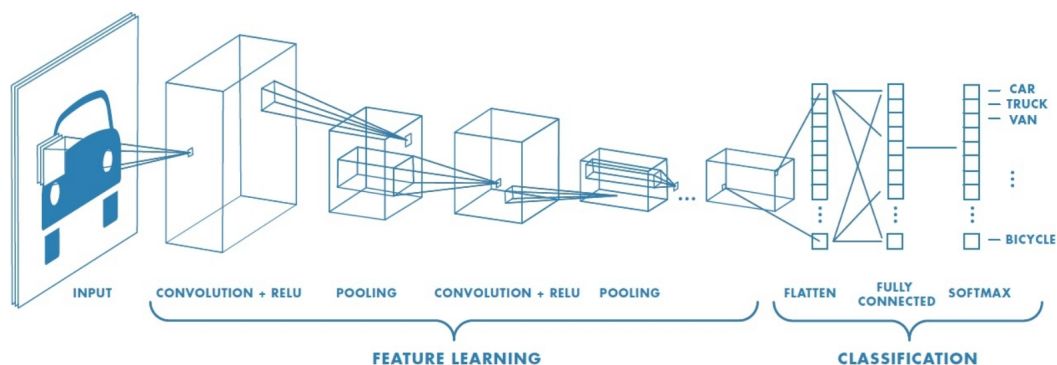


Abbildung 4: Convolutional Network [?]

Ein Convolutional Neuronal Network hat als Eingabeschicht das Bild als Input. Anschließend folgt je nach Netz und Funktion verschiedene Anordnung von Convolutional und ReLU Layern, sowie den Pooling Layern. Die Ausgabeschicht besteht bei Klassifikationsaufgaben immer aus einem Fully Connected Layer, welcher die Convolutional Layer mit den Ausgangsklassen verbindet und somit die eigentliche Zusammenhänge ermittelt.

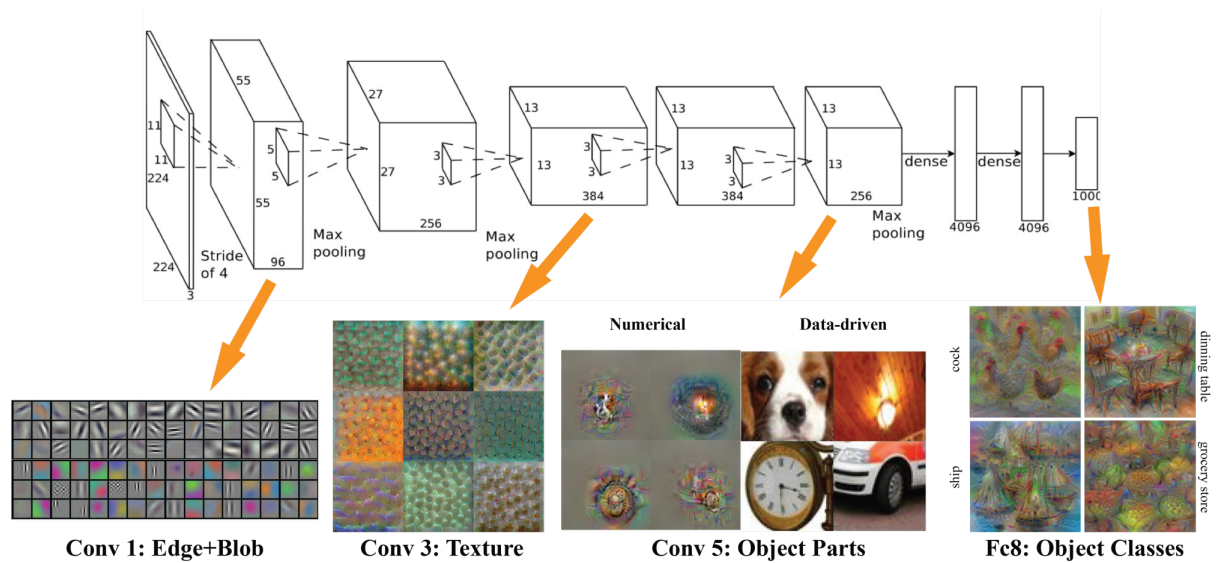


Abbildung 5: Feature Map [?]

6.1 Convolutional Layer

Die Convolutional Layer sind im Grunde Feature Extractoren. Sprich, sie erkennen Muster, je nach Größe des Filters (Kernels). Je tiefer der Layer, umso detaillierter ist das Muster der Featuremap. Je weiter am Anfang der Layer sich befindet, umso grober werden die Features. Als Aktivierungsfunktion wird hier der Relu-Funktion verwendet.

6.2 Pooling Layer

Der Pooling Layer reduziert die Sensibilität des neuronalen Netzwerks im Bezug auf kleine Ortsveränderungen der Merkmale. Außerdem wird die Größe der Feature Map und somit die Komplexität der Struktur reduziert. Average-pooling ermittelt den Mittelwert, Max-pooling das Maximum jedes betrachteten Ausschnittes und ersetzt diesen durch den ermittelten Wert.

6.3 Fully Connected Layer

Die letzte Schicht des KNN ist immer ein Fully Connected Layer. Wie der Name Fully Connected schon sagt, ist dieser Layer mit allen Outputs der vorherigen Layer verbunden. Dieser Layer entscheidet somit über die Ausgabe des Netzes.

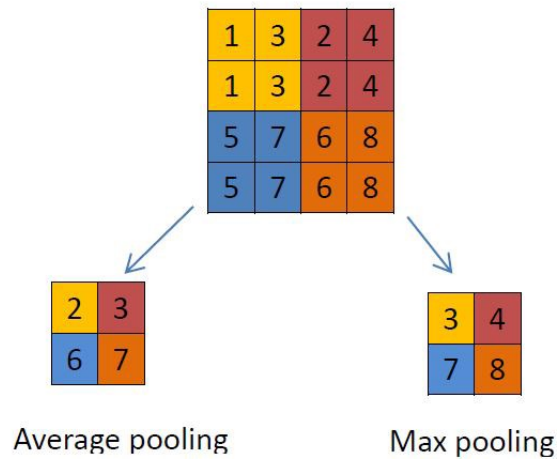


Abbildung 6: Pooling [?]

6.4 3D Convolutional Network

Ein 3D Convolutional Network ist im Grunde das gleiche wie ein 2D Convolutional Network, nur wurde dieses um eine Dimension ergänzt. Diese Dimension ist bei uns die Zeitachse. Somit ist es möglich nicht nur einzelne Fotos zu klassifizieren, sondern auch Videos von beliebiger Länge, begrenzt nur durch die Rechenleistung des Computers. Dennoch wurde empfohlen die Trainingsvideos zwischen 3 und 4 Sekunden zu halten, um ein sinnvolles Training des Netzes zu ermöglichen.

7 Chronologischer Tätigkeitsabriss

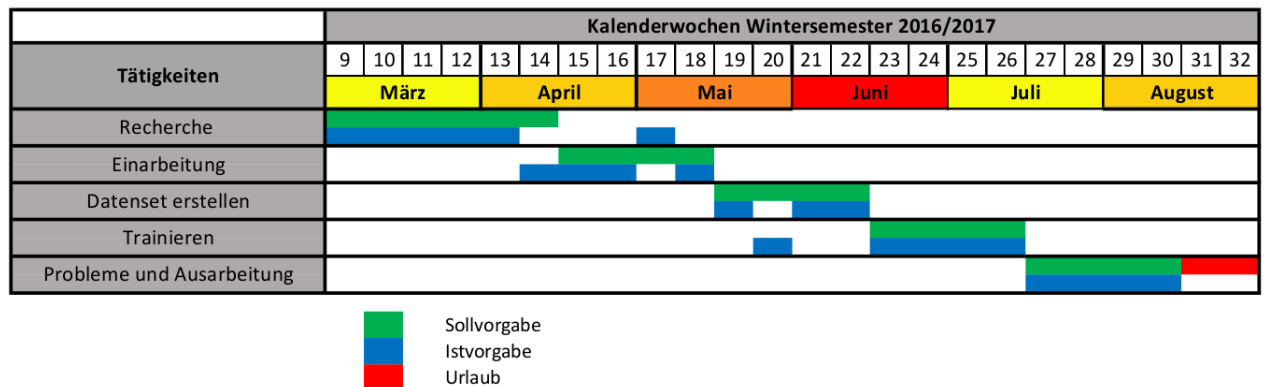


Abbildung 7: GANTT-Diagramm [?]

8 Internetrecherche

Zu Beginn des Praktikums hatte ich wenige Vorkenntnisse in Action-Recognition und mit welchen Künstlichen Neuronalen Netzen man dies umsetzen kann. Deshalb habe ich eine recht lange Recherchephase eingeplant, um mich mit den Grundlagen des Themas ausführlich zu befassen. Es stellte sich heraus, dass es verschiedene Ansätze zur Action-Recognition gibt:

- LSTM (Long- Short- Time Memory)
- 3D Convolutional Networks
- Two Stream 3D Convolutional Networks

Es gab viele Informationen und etliche Papers zu diesem Thema. Im Paper *Quo Vadis, Action Recognition? A New Model and the Kinetics Dataset* [?] von Google DeepMind. Sie stellten verschiedene Ansätze nebeneinander und verglichen, welche Möglichkeiten sich als effektiv herausstellten. Das beste Netz dieses Papers war das Two-Stream 3D-Convolutional Netz, welches die besten Ergebnisse lieferte. Von Deep Mind wurde das Framework für dieses Netz bei Github frei zugänglich gemacht.

9 Einarbeiten in vorhandene Frameworks

Die Einarbeitung gestaltete sich schwieriger als zuerst angenommen, da keine Funktionen zum Trainieren und Testen mitgeliefert wurden. Als Programmiersprache wurde Python benutzt und wie fr DeepMind klassisch wurde, als Front-End Sonnet, welches on top on Tensorflow aufbaut. Desweiteren habe ich ein Opensource Projekt der University of Science and Technology of China gefunden, welche ein Github Reposetorie zur Verfugung stellten, welches es ermoglichte dieses Netz zu trainieren und zu testen und von mir fr unseren Anwendungsfall angepasst werden konnte.

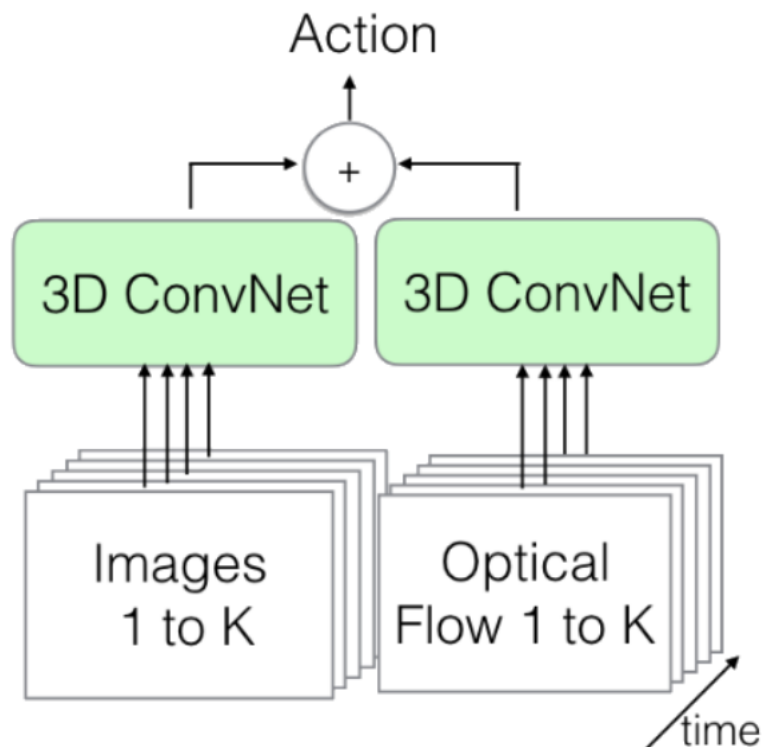


Abbildung 8: Two-Stream 3D-ConvNet [?]

Was heit Two-Stream 3D-ConvNet?

Es handelt sich um zwei 3D-ConvNets. Eins fr die Verarbeitung von RGB-Videos und eins fr die Verarbeitung von Optical Flow Videos mit einer anschlieenden Zusammenfhrung der Ergebnisse. Damit hat man stabilere Ergebnisse als nur einzelne Streams.

10 Optical Flow

Optical Flow ist die Bewegung von einzelnen Pixeln in X und Y Achse von zwei aufeinander folgende Frames eines Videos. Dieser Optical Flow wurde anfangs beiseite gelassen, um einen einfacheren Einstieg zu ermöglichen. Jedoch, zu einem späteren Zeitpunkt mit integriert. Um diesen Optical Flow zu extrahieren wurde von der Bibliothek Open-CV2 der Dense Optical Flow Algorithmus verwendet. Ich habe auch ein Künstliches Neuronales Netz das FlowNet2.0 [?] versucht für die Generierung des Optical Flow Images zu verwenden. Dies lieferte bessere Ergebnisse, dennoch war die Umsetzung zu Rechenaufwendig und somit für spätere Echtzeitanwendungen nicht brauchbar.



Abbildung 9: Optical Flow [?]

11 Erstellen von Trainings- und Testdaten (Datenset)

Ursprünglich wurde angedacht die Trainingsdaten mit der Microsoft Hololens zu erstellen. Da die Aktionserkennung auch für das Projekt mit der Hololens gedacht war. Dennoch stellte sich nach einigen Versuchen heraus, dass der Sichtwinkel der Hololens ungünstig ist und die Hände nur bei ausgestreckten Armen zu sehen sind. Dies ist aber keine natürliche Haltung. Um trotzdem ein Datenset zusammenzustellen wurde eine GoPro mit Kopfgurt benutzt. Welches mir es ermöglichte die Arbeiten unter normalen Umständen aufzunehmen. Es war wichtig, an verschiedenen Orten und mit verschiedenen Personen diese Videos aufzunehmen, um gute aber gewollte Varianz in den Trainingsdatenset zu bekommen. 80% der Videos wurden im FZI aufgenommen und 20% der Videos wurden in der Uni-Clinic Heidelberg aufgenommen. Natürlich wäre es am Besten gewesen Videos aus realen Verbandswechsel mit Patienten zu haben, dies war aber aus rechtlichen Gründen (Datenschutz) noch nicht möglich.

Es wurden ein Datenset zu folgenden Klassen erstellt:

- Gloves on (Anziehen von Handschuhen)

- Clean (Putzen/Desinfizieren der Ablage)
- unpacking (Auspacken der Verschlussenen Materialien wie Plaster und Verband)
- Gloves off (Ausziehen von Handschuhen)
- Desinfekt (Desinfizieren der Hnde)
- Other (Verschiedene Actionen aus der Egoperspektive)

12 Trainieren eines 3D-ConvNets

Um ein Netzwerk dieser Gre zu trainieren werden groe Rechenleistungen bentigt. Deshalb wurde das Netz von Googles Deepmind auf das ImageNet Datenset vor trainiert. Dies brauchte auf 11 Tesla GPUS mehrer Wochen Training. Dieses Netz kann dann fine-getuned werden, also auf unsere Anwendungen angepasst werden. Das Finetunen funktioniert wie folgt: man schneidet die letzten Fully Connectet Layer ab und trainiert diese dann auf seine eigenen Daten neu. Somit ist es mglich, ein Netz in kurzer Zeit mit wenig Daten zu trainieren und zustzlich noch gute Ergebnisse zu erhalten.

Das Training mit den RGB-Videos funktionierte gut. Und ergab nach den ersten Versuchen gute Ergebnisse. Das Ganze wurde dann mit Optical Flow wiederholt, somit hat man zwei Netze die einmal RGB Images und Optical Flow Images Verarbeiten knnen. Diese Ergebnisse werden anschlieden zu einer Aussage zusammengefasst. Diese kann man dann mit einem Evaluierungstest auswerten.

13 Ergebnisse der Actionserkennung

Es wurde ein Datenset erstellt welches aus 80% selbst aufgenommen Videos und 20% Videos aus der Uni Cline Heidelberg.

Es wurde ein Neuronales Netz zur Action Erkennung gefunden. Fr dieses Netz wurden Python Skripte erstellt, welche die Trainingsdaten so vor verarbeiten, dass damit ein Netz trainiert werden konnte. Anschließend wurde das 3D-ConvNet mit 6 Klassen: Gloves on, cleaning, unpacking, Gloves off, desinfekt, others mit jeweils 40 Videos trainiert. Die Evaluation gab gute Ergebnisse aus, das Netz hat eine Precision von 90% erreichen. Mit diesen Ergebnissen konnte man zeigen, dass es möglich ist diese Netzstruktur fr Ego-Action-Recognition zu verwenden.

14 Verbesserungsvorschläge und Zukunftsaussicht

14.0.1 Erweiterung der Trainingsdaten

Die Trainerdaten knnen noch erweitert werden, einmal mit Realen Daten und weiteren Klassen. Oder mit Data-Augmentation, in welcher man durch Drehen oder einbringen von Farben knstlich mehr Daten erstellt und somit eine bessere Erkennung zubekommen.

14.0.2 Realtime-Anwendung

Die Schnelligkeit muss verbessert werden und die API muss angepasst werden, um es als Realtimeanwendung umsetzen zuknnen. Dies bersteigt aber die Zeit in meinem Praktikum und ist deswegen noch zu entwickeln.

14.0.3 Optical Flow

Der Optical Flow wurde umgesetzt, dennoch ist es möglich, diesen zu Verbessern, durch verschiedene Vorverarbeitungen z.b. Bildstabilisierungen. Dies ist nicht getestet worden, knnte aber zu Verbesserungen fhren, um Hintergrundstrungen aus den einzelnen Images zu filtern.

14.0.4 Dritter Steam mit Hnden

Des Weiteren knnte man das Netz mit einem dritten Steam erweitern, um so den Focus auf spezielle Details zu legen.

- **Segmentation der Hnde:** Mit der Segmentation der Hnde wre es mglich speziellen Focus auf die Hnde zulegen oder auf Video abschnitte mit Hnden.



Abbildung 10: Segmentation Hands [?]

- **Hand-Pose:** Mit der Pose Estimation der Hnde wre eine Weiterentwicklung der Segmentation. Mit der Pose Estimation der Hnde wre eine Weiterentwicklung der Segmentation. Dennoch wurde dies meist mit RGB-D Images umgesetzt. RGB-D Images enthalten noch weiter Tiefeninformation welche die GoPro nicht mit liefert.

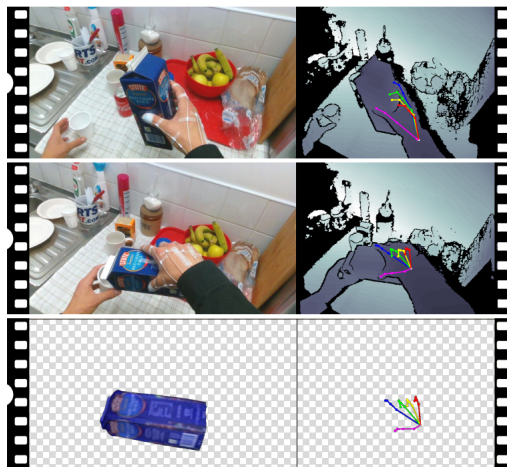


Abbildung 11: Hand-Pose with RGB-D-Images [?]