

DeepXRay

Themenbereiche:	Anwendung von Deep Learning in der Rheumatologie
Studierender:	Patrick Bucher
Betreuungsperson:	Daniel Pfäffli
Experte:	Jeremy Callner, APG SGA
Auftraggebende:	Dr. Tobias Reinhard, Seantis GmbH
Keywords:	Deep Learning, Medizinische Forschung, Xray-Scoring, Bilderkennung

1 Aufgabenstellung

Die rheumatoide Arthritis ist eine Autoimmunerkrankung, bei der gesunde Gelenke aufgrund andauernder Entzündungen irreversibel geschädigt werden. Die fortschreitende Erosion des Gelenkgewebes kann anhand von Röntgenaufnahmen durch medizinisches Fachpersonal beurteilt werden. Dies nimmt pro Patient mehrere Minuten in Anspruch.

In einer Vorarbeit haben Janick Rohrbach (ZHAW), Tobias Reinhard (Seantis GmbH), Beate Sick (Universität Zürich) und Oliver Dürr (HTWG Konstanz) diesen Vorgang mithilfe von *Deep Convolutional Neuronal Networks* automatisiert, wodurch eine gleichbleibende Bewertungsqualität gewährleistet wird. Das Machine-Learning-Modell wurde mit zehntausenden von klassifizierten Gelenken aus der SCQM-Datenbank (*Swiss Clinical Quality Management in Rheumatic Disease*) trainiert, validiert und getestet. Es kann die Erosion des Gelenkgewebes automatisch für zehn verschiedene Gelenke der linken Hand (siehe Abbildung 1, Seite 2) bewerten. Dieser Vorgang wird als *Scoring* bezeichnet.

Das Ziel des Projekts *DeepXRay* ist es, auf Basis dieses und zweier weiterer Machine-Learning-Modelle (Erkennung von Körperteilen und Extraktion von Gelenken auf Röntgenbildern) einen Prototyp für einen Webservice zu entwickeln, womit der beschriebene Vorgang des Scorings automatisiert wird, und somit schneller und in konstanter Qualität erfolgen kann. Dies ist gerade bei langfristig angelegten und grösseren Studien zum Verlauf der rheumatoiden Arthritis hilfreich.

Weiter soll die Qualität der automatisch erstellten Bewertungen mithilfe verschiedener Metriken überprüft werden, sodass sich Qualitätsverbesserungen beim Austausch von einzelnen Modellkomponenten besser einschätzen lassen. Die notwendigen Schritte zur Vereinheitlichung der Machine-Learning-Modelle und zur Überführung des erstellten Prototyps in die Produktivumgebung des Auftraggebers sollen in einem Ausblick behandelt werden.

2 Ergebnisse

Das Ergebnis der Arbeit ist ein Prototyp, der einen Webservice (RESTful API) anbietet. Das Röntgenbild wird in den folgenden drei Schritten verarbeitet: Zunächst wird das auf dem Röntgenbild dargestellte Körperteil erkannt. Handelt es sich dabei um eine linke Hand, kann der Vorgang fortgesetzt werden. Im zweiten Schritt werden vom Röntgenbild die Metacarpophalangealgelenke (MCP) und proximalen Interphalangealgelenke extrahiert. Die Bildausschnitte werden an den nächsten Verarbeitungsschritt weitergereicht. Im dritten Schritt wird die Erosion der Gelenke auf einer Skala von 0 (keine Schädigung) bis 5 (Schädigung von 80 bis 100 Prozent des Gelenkgewebes) ermittelt. Die Scoring-Ergebnisse werden gesammelt und an den Client zurückgeliefert.

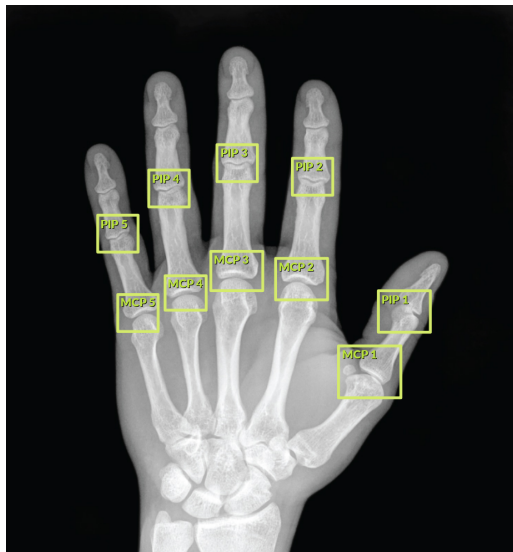


Abbildung 1: Röntgenbild einer linken Hand mit markierten Metacarpophalangealgelenken (MCP) und proximalen Interphalangealgelenken (PIP), Quelle: Wikimedia Commons (Lizenz: CC BY 4.0)

Dem Webservice kann via HTTP ein Röntgenbild einer linken Hand übergeben werden, worauf er in Sekundenschnelle mit einem Scoring der Metacarpophalangealgelenke (MCP) und proximalen Interphalangealgelenke im JSON-Format antwortet:

```
$ curl -k -X POST -F "xray=@hand-left.jpg" https://localhost:8080/score | jq
{
  "scores": {
    "mcp1": 1, "mcp2": 0, "mcp3": 1, "mcp4": 1, "mcp5": 1,
    "pip1": 1, "pip2": 3, "pip3": 2, "pip4": 5, "pip5": 5
  }
}
```

3 Lösungskonzept

Die genannten Schritte sollen in einer für den Client nützlichen Frist ausgeführt werden. Wird erst mit dem Scoring begonnen, wenn *alle* Gelenke extrahiert worden sind, dauert dies recht lange. Darum arbeitet der Prototyp asynchron, sodass mit dem Scoring bereits begonnen werden kann, wenn das *erste* Gelenk extrahiert worden ist. Hierzu werden die verschiedenen Komponenten (siehe Abbildung 2, Seite 3) lose mittels Messaging gekoppelt, sodass diese mit mehreren Instanzen ausgeführt werden können. Dadurch kann die Extraktion und das Scoring mehrerer Gelenke parallel ausgeführt werden, was sich positiv auf die Performance auswirkt.

Eine Orchestrator-Komponente koordiniert den ganzen Ablauf und die Kommunikation mit dem Client. Kann auf einem gegebenen Röntgenbild eine linke Hand erkannt werden, gibt der Orchestrator die Extraktion der Gelenke in Auftrag. Diese Anfragen werden mit einem *Correlation Identifier* ausgestattet. Die extrahierten Gelenke werden via Queue, d.h. am Orchestrator vorbei an das Scoring geschickt. Der Orchestrator sammelt die Scores über eine weitere Queue ein, und kann diese anhand ihres Correlation Identifiers dem richtigen Client zuordnen.

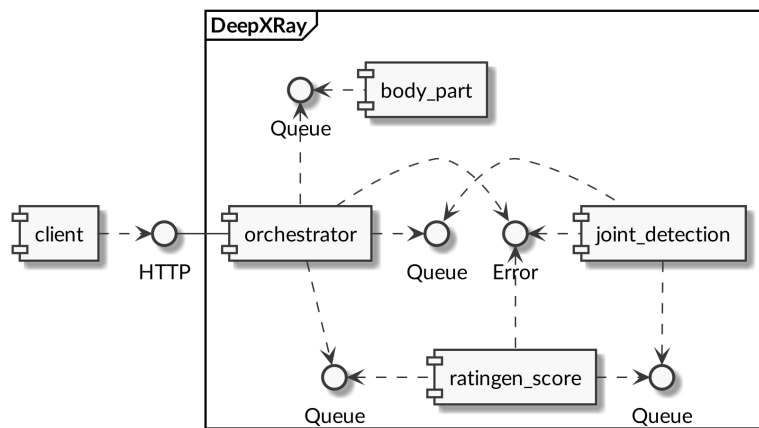


Abbildung 2: Die Architektur des Prototyps, die eine performante Verarbeitung von Röntgenbildern ermöglicht. Die einzelnen Komponenten sind lose via Messaging gekoppelt.

4 Spezielle Herausforderungen

Die Machine-Learning-Modelle waren mit verschiedenen Frameworks in unterschiedlichen Versionen entwickelt worden (TensorFlow 0.12, TensorFlow 1.4, Keras usw.). Eine Konvertierung der Modelle für neuere Versionen war für den gegebenen Projektrahmen zu aufwändig. Dadurch ist die Ausführung der Modelle nicht in einer einheitlichen Laufzeitumgebung möglich. Die Modelle werden darum isoliert in Containern ausgeführt.

Der Prototyp muss gegen aussen eine synchrone API anbieten, welche die Röntgenbilder in einer nützlichen Frist, d.h. in Sekundenschnelle verarbeiten kann. Hierzu muss das System intern asynchron arbeiten, um eine nebenläufige Abarbeitung der einzelnen Schritte zu ermöglichen, und um so die nötige Performance gewährleisten zu können.

Weiter mussten für das Spezialgebiet der Rheumaforschung sinnvolle Evaluationsmetriken gefunden und umgesetzt werden, womit die Qualität des Gesamtsystems sinnvoll bewertet werden kann. Hierzu mussten Testdaten gesammelt, aufbereitet und manuell aussortiert werden.

5 Ausblick

Um den Prototyp in den Produktivbetrieb überführen zu können, sind noch einige Aufgaben zu erledigen. Die verschiedenen Modellkomponenten müssen auf performanten Servern (idealerweise mit vielen CPUs) bereitgestellt werden. Die RESTful API des Orchestrators benötigt Schutz vor unbefugtem Zugriff, wozu geeignete Mechanismen (wie z.B. OAuth 2.0) einzurichten sind. Die Kommunikation zwischen dem Client und dem Orchestrator muss mit einem «richtigen», d.h. nicht selbst signierten TLS-Zertifikat abgesichert werden. Schliesslich muss der Service in andere Anwendungen integriert werden.

Die Machine-Learning-Modelle sollten längerfristig auf die aktuelle Version von TensorFlow migriert werden. Mit dem weit verbreiteten *SavedModel*-Format kann sichergestellt werden, dass die Modelle auch in Zukunft unterstützt und in verschiedenen Umgebungen ausgeführt werden können. Der Einfluss der aktualisierten Modelle auf die Qualität des Gesamtsystems kann mithilfe der bereitgestellten Evaluationsmechanismen überprüft werden. Längerfristig sollen nebst linken Händen auch andere Körperteile verarbeitet werden können, wozu die Modelle entsprechend zu erweitern sind.