

Aufgaben 1-3 sind für das erfolgreiche Bestehen des Praktikums vollständig zu bearbeiten. Die Ergebnisse der Aufgaben sollen in einem **20 minütigen Gruppenvortrag** dargelegt werden. Bei allen Aufgaben soll eine kurze theoretische Einführung in den genutzten Algorithmus gegeben werden. Hierzu gehört in Aufgaben 2 und 3 eine genaue Beschreibung des genutzten Modells (inklusive ggf. zu wählender (Hyper-)Parameter, gewählten Train/Validations Splits, etc.). Außerdem sollen die erzielten Ergebnisse (Trainingsverlauf, Validation Ergebnisse, etc.) in geeigneter Form dargestellt und Vorteile und Nachteile der drei Kontur-Erkennungs-Methoden gegenüber gestellt werden. Hierbei sollen alle Entscheidungen begründet werden. Die Note des Praktikums basiert auf der Präsentation und einer anschließenden prüfungsrelevanten Fragerunde.

Die Bearbeitung der Aufgaben erfolgt in Zweiergruppen. Gängige Packages dürfen für die Bearbeitung der Aufgaben genutzt werden. Sie können den Zugang zum Palma-II Cluster der Universität nutzen, um die Aufgaben zu lösen - schreiben Sie uns dann bitte eine Email und informieren sich im Confluence der Uni MS IT¹ (nur aus dem Uni-Netz erreichbar).

Das Ziel der Aufgaben ist Kontur-Erkennung auf dem BSDS500² mit drei verschiedenen Methoden, sowie einer sinnvollen Auswertung bei schwer zu bestimmender ground truth. Beachten Sie, dass ein Vergleich der Lösungen aller Verfahren untereinander ebenso zur Aufgabe des Praktikums gehört.

Zur einfacheren Verarbeitung können die Bilder etwas heruntergesampelt werden. Es sollte mindestens eine Auflösung von 100 Pixeln pro Dimension erhalten bleiben. Für Aufgabe A3 muss eine Unterteilung des Datensatzes in Trainings-, Test- und Validierungsdaten vorliegen. Nehmen Sie zur Auswertung von den Aufgaben A1 und A2 nur jene Bilder, die in den Testdaten enthalten sind, um die Ergebnisse vergleichbar zu machen.

Aufgabe 1 [Klassische Kantenfilter]

Implementieren Sie Canny Edge Detection wie in Vorlesung 4 einmal mit und einmal ohne Bildglättung. Benutzen Sie für die Gradientenberechnung im Canny-Algorithmus einen Operator Ihrer Wahl (z. B. Sobel). Überlegen Sie sich eine Metrik und werten Sie die Qualität des Canny-Edge Detectors sinnvoll anhand der im Datensatz vorhandenen Konturmasken aus.

Aufgabe 2 [Kombination von Features mit lernendem Algorithmus: MLP]

In dieser Aufgabe soll ein kleines Multi-Layer Perceptron trainiert werden, das eine pixelweise Klassifikation der Konturen aus Featurevektoren vornimmt, die aus den einzelnen Pixeln generiert werden.

- Erstellen Sie ein geeignetes MLP zum Lernen eines Klassifikationsproblems mit 2 output Klassen und einem Feature Vector als Input. Die Wahl der Features steht frei.
- Trainieren Sie pro Bild ein MLP, wobei der Input des MLP jeweils ein Pixel ist. Als Trainingsdaten können Sie bis zu 10% der Bildpunkte nutzen. Führen Sie nach dem abgeschlossenen Training auf den restlichen Pixel eine Inference durch und erhalten so eine vollständige Kontur-Maske.

Werten Sie die Qualität des MLP sinnvoll anhand der im Datensatz vorhandenen Konturmasken aus.

¹<https://confluence.uni-muenster.de/display/HPC/Getting+started>

²<https://github.com/BIDS/BSDS500>

Aufgabe 3 [End-to-end Kontur-Erkennung: U-Net]

Trainieren Sie ein U-Net wie in der Vorlesung vorgestellt, das eine Segmentierung der Bilder durch Erzeugen einer Kontur-Maske generiert. Übernehmen Sie die gegebene Aufteilung des Datensatzes in Trainings-, Validierungs- und Testdaten. Werten Sie die Qualität des U-Net sinnvoll anhand der im Datensatz vorhandenen Konturmasken aus.