

Praktikum Medizinische Informationssysteme (Pflichtenheft)

Marvin Teichman, Martin Thoma

16. Mai 2016

1 Projekttitle

Semantische Segmentierung von medizinischen Instrumenten mit Deep Learning Techniken

2 Fragestellung

Wie gut kann man im Kontext minimalinvasiver Operationen die Klassen „Hintergrund“ und „Medizinisches Instrument“ in Bildern automatisch segmentieren?

Kann man zugleich eine hohe Segmentierungsqualität und geringe Latenz erreichen?

3 Stand der Forschung

CNNs werden seit einigen Jahren in allen Bereichen der Computer Vision eingesetzt: Traditionellerweise für Bildklassifikation, für Objektdetektion und auch für semantische Segmentierung.

[Moh14] erreicht mit einem 7-schichtigem CNN für den KITTI-Datensatz [FKG13] den zweiten Platz, [?] erreicht auf MSRC-21 den ersten Platz mit Conditional Random Fields.

Für weitere Techniken zur semantischen Segmentierung: [Tho]

4 Ziele

4.1 Sollkriterien

- Es wird ein Programm erstellt, welches als Input ein Bild einer Operation erhält und als Output ein Bild gleicher Größe liefert, welches die Segmentierung darstellt.
- Das Programm liefert für ein $640 \text{ px} \times 480 \text{ px}$ Bild innerhalb von spätestens 5 s auf einem ThinkPad T460p mit einer Nvidia 940MX-Grafikkarte eine Segmentierung.
- Das Programm liefert ein Segmentierungsergebnis, welches nach einem noch festzulegendem Maß (vgl. AP2) mindestens eine relative Verbesserung von 10 % bzgl. dem Basis-Segmentierer (vgl. AP3) bringt. Diese Segmentierungsqualität wird über eine Testmenge berechnet.¹
- Es wird für mindestens 5 Modelle die Qualität berechnet und miteinander verglichen.

4.2 Wunschkriterien

- Das Programm liefert für ein $640 \text{ px} \times 480 \text{ px}$ Bild innerhalb von spätestens 0.06 s auf einem ThinkPad T460p mit einer Nvidia 940MX-Grafikkarte ein Ergebnis.
- Das Programm liefert ein Segmentierungsergebnis, welches nach dem in AP2 festgelegten Maß um mindestens 20 % relativ besser ist.

¹Also: $\frac{Q_{\text{Best}} - Q_{\text{Basis}}}{1 - Q_{\text{Basis}}} \geq 0.1$ mit einem Maß $Q \in [0, 1]$

5 Arbeitspakete

5.1 AP1: Input Pipeline

Bauen eines Input Producers, welcher nebenläufig auf der CPU Daten einlieft und verschiedene Standardtechniken zur Data Augmentation anwendet. Die Daten werden bereits als Batch zusammengefasst und auf die Grafikkarte geladen. Verwaltet wird das ganze von einer nebenläufigen Queue, welche neue Daten anfordert, wenn die Anzahl der bereitgestellten Batches einen Schwellwert unterschreitet.

5.2 AP2: Daten, Test- und Evaluierungsskripte

Viele der folgenden Arbeitspakete beinhalten das Überprüfen verschiedener Modelle auf zwei Kriterien:

- **Geschwindigkeit:** Wie schnell wird ein einzelnes Bild segmentiert?
- **Qualität:** Wie brauchbar ist das Ergebnis?

Dabei wird in diesem Arbeitspaket ein Test-Datensatz festgelegt, welcher konsistent für alle Modelle ist. Dieser wird niemals zum Training verwendet. Die Qualität wird konsistent für alle Modelle berechnet werden. Das Maß der Qualität ist noch festzulegen.

Die Geschwindigkeit ist die reine Zeit zum Segmentieren. Sie beinhaltet weder die Zeit zum Laden des Modells noch zum schreiben des Ergebnisses auf die Festplatte.

Es wird angegeben, wie gut die Qualität ist, wenn man einfach die häufigste Klasse schätzt. Damit wird ein Vergleich geschaffen, wie stark die Verbesserung der verschiedenen Segmentierer ist.

5.3 AP3: Basic Segmenter

Diese Modelle sollen eine Grundlage für die Beurteilung der Qualität und des Wertes von weiteren Verbesserungen bieten.

- Modell 301: Klassifiziere Pixel rein anhand der Farbe.
- Modell 302: Klassifiziere Pixel rein anhand der Farbe und Position.
- Modell 303: Nutze die Klassifikation der anderen Pixel um eine schlüssige Gesamtsegmentierung zu erhalten. Dafür können z.B. morphologische Operationen eingesetzt werden.

5.4 AP4: Shallow CNN Segmenter

5.4.1 AP 4.1: SST-Modell

Modell 401: Trainieren eines Segmentierers, welcher den Sliding-Window-Ansatz benutzt und nur den Pixel im Zentrum klassifiziert. Es wird die folgende Topologie haben:

Layer	Type	Shape
0	Input	51
1	Convolution	10 filter each 5×5
2	Convolution	10 filter each 5×5
3	Pooling	2×2
4	Output	1

Als Trainingsdaten werden 51×51 -Patches mit einem Stride von $s = 10$ aus den verfügbaren Bildern deterministisch ausgeschnitten.

Zur Klassifikation der Randpixel wird 0-Padding verwendet.

Diese Einstellungen wurden so gewählt, da sie bereits in [BKTT15] vernünftige Ergebnisse für eine semantische Segmentierungsaufgabe geliefert haben.

5.4.2 AP 4.2: Weitere Patch-Segmenter

Trainieren von Segmentierern, welche den Sliding-Window-Ansatz benutzt und mehr Pixel als nur das Zentrum klassifiziert. Variationen sind durch die Window-Größe des Inputs und den Stride (welcher damit die Ausgabe-Größe bestimmt) gegeben.

5.5 AP6: FCN-Architectures

Modell 6XX: Experimentiere mit FCN-Architekturen. Diese wurden in [LSD15] vorgestellt und haben für PASCAL VOC sehr gute Segmentierungsergebnisse geliefert.

5.6 AP7: Fine-Tuning

Modell 7XX: Experimentiere mit Fine-Tuning Ansätzen. Aufgrund der geringen Anzahl an verfügbaren Bildern zum trainieren könnte es sinnvoll sein ein bereits auf Bildern trainiertes Netz zu nutzen und nur die letzten Schichten des Netzwerks auf den medizinischen Daten anzupassen. In den ersten Schichten werden typischerweise allgemeine Features trainiert, wohingegen die letzten Schichten sehr problemspezifisch sind.

5.7 AP8: Evaluation

Vergleich der Ergebnisse und Schreiben des Abschluss-Papers. Das Paper wird auf Englisch geschrieben und wird mindestens 3 Seiten lang sein. Es werden die Ergebnisse der vorherigen Arbeitspakete vorgestellt, verglichen und bewertet.

Literatur

- [BKTT15] S. Bittel, V. Kaiser, M. Teichmann, and M. Thoma, “Pixel-wise segmentation of street with neural networks,” *arXiv preprint arXiv:1511.00513*, 2015.
- [FKG13] J. Fritsch, T. Kuehnl, and A. Geiger, “A new performance measure and evaluation benchmark for road detection algorithms,” in *International Conference on Intelligent Transportation Systems (ITSC)*, 2013.
- [LSD15] J. Long, E. Shelhamer, and T. Darrell, “Fully convolutional networks for semantic segmentation,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 3431–3440. [Online]. Available: <https://arxiv.org/abs/1411.4038>
- [Moh14] R. Mohan, “Deep deconvolutional networks for scene parsing,” 2014. [Online]. Available: <http://arxiv.org/abs/1411.4101>
- [Tho] M. Thoma, “A survey of semantic segmentation.” [Online]. Available: <http://arxiv.org/abs/1602.06541v1>