

Fakultät für Physik und Astronomie

Universität Heidelberg

Bachelor Thesis im Fach Physik
vorgelegt von

Jan Lammel

geboren in Heidelberg

2015

**Vergleich von Hamming- und Variation of Information-Loss
basiertem strukturiertem Parameterlernen beim Multicut Problem**

Diese Bachelorarbeit wurde ausgeführt von Jan Lammel am
Heidelberg Collaboratory for Image Processing (HCI) in Heidelberg
unter Betreuung von
Prof. Dr. Fred Hamprecht

ABSTRACT

There are several ways of segmenting images. In this paper we solve the Multicut Problem in order to do that, which involves structured learning of the parameters. For that task there are different ways to define the loss function of a segmentation. The Hamming Loss is a commonly used one, which checks every edge in comparison to the ground truth. If there is no match in activity the loss increases. There are several disadvantages about that like the dependence on a very good ground truth and the discontinuity (a little shifted segmentation is rated very bad).

Therefore the calculation of the loss via Variation of Information is analyzed within this work, which does not consider the individual edges anymore. Rather one examines the segmentation's labels and penalizes area-dependent dissimilarities to the ground truth.

The experiments were done on the BSD-500 dataset, comparing minimization of Hamming- and Variation of Information-Loss. Against expectation, latter leads to an overfit of the training data or there is no significant difference, depending on the feature space. So the expected improvements did not occur. Though a possible explanation of this behaviour has been found, which is based on the use of SLIC-Superpixels.

ZUSAMMENFASSUNG

Es existieren verschiedene Wege zur Segmentierung von Bildern. In dieser Arbeit wird dies über das Lösen des Multicut Problems realisiert, was ein strukturelles Lernen der Parameter mit sich bringt. Hierzu gibt es verschiedene Arten, wie die Loss Funktion einer Segmentierung definiert werden kann. Eine bisher oft verwendete Methode ist der Hamming Loss, bei der jede einzelne Kante mit der Ground Truth verglichen wird und bei fehlender Übereinstimmung der Loss steigt. Dies hat verschiedene Nachteile, wie die Abhängigkeit von einer sehr guten Ground Truth, sowie die Unstetigkeit (eine minimal verschobene Segmentierung wird als extrem schlecht eingestuft).

Daher wird in dieser Arbeit die Berechnung des Losses mittels Variation of Information untersucht. Hierbei betrachtet man nicht mehr die einzelnen Kanten, sondern die Labels der Segmentierung und bestraft flächenabhängig Unterschiede zur Ground Truth.

Die Experimente wurden auf dem BSD-500 Datensatz durchgeführt, wobei die Minimierung des Hamming- und Variation of Information-Losses verglichen wurde. Wider den Erwartungen führt Letzteres allerdings je nach Feature Space entweder zu einem Overfit der Trainingsdaten oder zu keiner signifikanten Veränderung, wodurch die erhofften Verbesserungen nicht eintraten. Es konnte allerdings eine mögliche Erklärung für dieses Verhalten gefunden werden, die auf der Verwendung der SLIC-Superpixel beruht.

INHALTSVERZEICHNIS

1	EINLEITUNG	1
1.1	Motivation Variation of Information	2
2	THEORETISCHE GRUNDLAGEN	3
2.1	Graphen Theorie	3
2.2	Feature Space	4
2.3	Das Multicut Problem	4
2.4	Loss Funktionen	4
2.4.1	Hamming Loss	4
2.4.2	Variation of Information	5
2.5	Structured Learning	5
2.5.1	Subgradient Descent	6
2.5.2	Stochastic Gradient	6
3	RELATED WORK	9
4	EXPERIMENTELLES SETUP	11
4.1	Trainings- und Testdaten	11
4.2	Graphical Model Unterbau und Solver	12
4.3	Feature Space	12
5	EXPERIMENTE UND RESULTATE	17
5.1	Stochastic Gradient mit RF Feature	17
5.2	Stochastic Gradient ohne RF Feature	25
5.3	Kreuzvalidierung Messung 10	29
6	FAZIT	31
	LITERATURVERZEICHNIS	33

EINLEITUNG

Das weite Forschungsfeld der Bildsegmentierung beschäftigt sich mit der Problemstellung, Bilder automatisiert in einzelne semantisch sinnvolle Segmente zu unterteilen. Anwendungsgebiete finden sich unter anderem in der Objekterkennung, Biologie, Medizin und allgemein Bildanalyse-Methoden, wobei die Segmentierung als Vorstufe zur weiteren Bearbeitung dient.



(a) Beispielbild



(b) Segmentierung des Beispielbildes
(hier Ground Truth)

Abbildung 1: Beispielbild mit idealer Segmentierung, die es zu erhalten gilt

Dieser Prozess soll lern-basiert sein, d.h. es werden dem Algorithmus Trainingsdaten mit Beispielbildern inklusive Soll-Segmentierung (Ground Truth) übergeben. Anhand dieser werden Parameter optimiert um möglichst allgemeingültig, den Trainingsdaten ähnliche, Bilder ebenso in einzelne Segmente gliedern zu können.

In dieser Arbeit handelt es sich um strukturelles Lernen da nicht jede Kante des Graphs für sich genommen betrachtet wird, sondern das Bild als Ganzes. Dies wirkt sich auch auf die Möglichkeiten von Loss-Funktionen aus, die Meta-Informationen der Bilder nutzen können, die beim unstrukturierten Lernen nicht zur Verfügung stehen.

In diesem Sinne wird nun die Loss-Funktion Variation of Information (VOI) zur Quantifizierung der Qualität einer Segmentierung vorgestellt und mit einer bestehenden (Hamming Loss) verglichen. Da der Lern-Algorithmus auf diesem Kriterium aufbaut, ist dies elementar für die Güte der resultierenden Segmentierung.

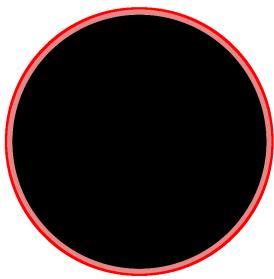
1.1 MOTIVATION VARIATION OF INFORMATION

Bisher wird beim Hamming Loss jede einzelne Kante der Segmentierung überprüft, ob diese in der Ground Truth ebenso vorhanden ist oder nicht. Bei einer leicht verschobenen Segmentierung führt dies zu einem extrem hohen Loss, obwohl die Segmentierung nicht viel schlechter ist als die der Ground Truth. Dieser Fall kann beispielsweise eintreten wenn der Ground Truth Ersteller unsauber gearbeitet hat, was durchaus vorkommen kann wenn man davon etliche erstellt. Außerdem ist es oft Interpretationssache, wo nun genau eine Kante eines Objektes verläuft, wenn sich der zugehörige Gradient des Bildes über eine gewisse Fläche streckt.

Mittels Variation of Information sollen nun diese Nachteile beseitigt werden. Man betrachtet hierbei nicht mehr den Zustand jeder einzelnen Kante, ob diese nun an oder aus ist, sondern die einzelnen segmentierten Flächen. Die exakte mathematische Berechnung folgt in den theoretischen Grundlagen (2.4.2). Anschaulich gesehen werden allerdings nur die Flächen bestraft, die ein unterschiedliches Label als die Ground Truth haben.

Abb. 2a veranschaulicht dies: Der schwarze Bereich entspricht der zu segmentierenden Struktur und die rote Linie wäre die zugehörige Ground Truth. Beim Hamming Loss wären sowohl die äußeren Kanten der schwarzen Struktur, als auch die Kanten der roten Linie nicht korrekt, was einen hohen Loss zur Folge hätte. Bei Variation of Information hingegen wird die rosa Fläche als Loss gezählt und ist infolgedessen deutlich geringer da die Segmentierung fast der Ground Truth entspricht.

In Abb. 2b ist ein Beispiel solch eines natürlichen Bildes dargestellt, wo es absolut nicht eindeutig ist, wo genau die Ground Truth zu zeichnen ist.



(a) Demonstration Loss Variation of Information



(b) Beispielbild mit unklarer Ground Truth

Abbildung 2

2

THEORETISCHE GRUNDLADEN

2.1 GRAPHEN THEORIE

Die Grundlage aller weiteren Betrachtungen ist ein Region Adjacency Graph (RAG). Um diesen zu erstellen, wird das Bild zunächst mithilfe des SLIC-Algorithmus [2] in Superpixel unterteilt, dessen Ränder möglichst an den Objektkonturen verlaufen. Die grundsätzliche Idee ist die, Schwerpunkte bezüglich Distanz und Farbe für jeden Superpixel zu bilden und iterativ Pixel einem Schwerpunkt mit möglichst kleiner Farb-Raum-Distanz zuzuordnen.

Die Betrachtung auf Superpixel- statt auf Pixelebene ist notwendig da das zu lösende Multicut Problem NP-hard ist und deshalb die Anzahl der Kanten des Graphs mithilfe von Superpixeln reduziert werden muss.

Der Region Adjacency Graph G baut sich aus Knoten V und Kanten E auf. In unserem Fall entsprechen die Knoten den Superpixeln. Die Kanten bestehen nur zwischen denjenigen Knoten, bei denen die zugehörigen Superpixel direkt angrenzen und somit eine gemeinsame Kante besitzen.

Sowohl die resultierende Superpixel-Partitionierung mittels SLIC, als auch eine Demonstration, wie daraus der Region Adjacency Graph entsteht, ist in Abb. 3 dargestellt.

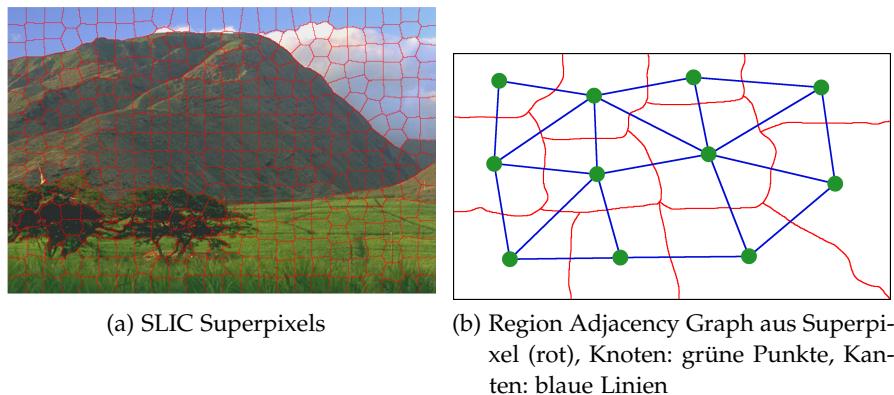


Abbildung 3: Beispiel einer Superpixel Partitionierung mittels SLIC

Bei der Segmentierung geht es darum, ein konsistentes Labeling der Superpixel zu erreichen. Die Menge aller Superpixel mit dem selben Label entspricht einem Gebiet, welches semantisch möglichst einheitlich sein soll. Dies wird über die Aktivität der Kanten erreicht, welche an- oder ausgeschaltet sein können. Für die Aktivität einer Kante y gilt somit: $y \in \{0, 1\}$, wobei 1 aktiv und 0 inaktiv bedeutet.

Konsistent ist eine Segmentierung genau dann, wenn bei aktiven Kanten die zugehörigen Superpixel verschiedene Labels haben und ana-

log bei inaktiven Kanten die Superpixel die gleichen Labels. Anschaulich gesehen ist dies der Fall, wenn alle aktiven Kanten geschlossene Linien bilden.

2.2 FEATURE SPACE

Der Feature Space $X \in \mathbb{R}^{|E| \times D}$ ordnet jeder Kante D Features zu, die möglichst in Korrelation zur Frage stehen, ob die betrachtete Kante aktiv oder inaktiv sein soll. Zu den in dieser Arbeit verwendeten Feature wird in [4.3](#) genauer eingegangen.

2.3 DAS MULTICUT PROBLEM

Die Aufgabe, eine konsistente Segmentierung anhand der durch den Feature Space beschriebenen gewichteten Kanten zu erstellen, wird als Multicut Problem (MP) bezeichnet. Es wird durch folgendes Minimierungsproblem beschrieben:

$$\begin{aligned} \arg \min_y \quad & \sum_{y_i \in E} \langle w, \beta_e \rangle \cdot y_i \\ \text{s.t.} \quad & y - \sum_{y_i \in P(y)} y_i \leq 0 \quad \forall y \in E \end{aligned} \quad (1)$$

Hierbei entspricht $w \in \mathbb{R}^D$ den Weights der einzelnen Feature und $\beta_e \in \mathbb{R}^D$ den Funktionswerten der D extrahierten Informationen des Feature Spaces bezüglich Kante e. Die Nebenbedingungen erzwingen die Konsistenz der Segmentierung. $P(y)$ ist hierbei der kürzeste Pfad über inaktive Kanten der beiden Superpixel, die benachbart zu y sind. In der Praxis wird das Minimierungsproblem zunächst ohne Constraints gelöst und anschließend solange für diejenigen Kanten hinzugefügt, die die Konsistenzbedingung verletzen, bis Konsistenz erreicht ist.

2.4 LOSS FUNKTIONEN

Mithilfe einer Loss Funktion $\mathcal{L}(y, y^*)$ wird quantifiziert, wie gut eine Segmentierung y mit derjenigen der Ground Truth y^* übereinstimmt. In dieser Arbeit wird die Methode Variation of Information vorgestellt und mit der bestehenden des Hamming Loss verglichen.

2.4.1 Hamming Loss

$$\mathcal{L}_i(y_i, y_i^*) = \begin{cases} \mathbb{I}[y_i \neq y_i^*] \cdot \alpha_{\text{over}} & \text{if } y_i^* = 0 \\ \mathbb{I}[y_i \neq y_i^*] \cdot \alpha_{\text{under}} & \text{if } y_i^* = 1 \end{cases} \quad \forall y_i \in E \quad (2)$$

$$\mathcal{L}_H(y, y^*) = \sum_{y_i \in E} \mathcal{L}_i(y_i, y_i^*) \quad (3)$$

Es werden direkt die Kanten der Segmentierung y und der Ground Truth y^* verglichen und bei fehlender Übereinstimmung erhöht sich

der Loss. Meist ist $\alpha_{\text{under}} > \alpha_{\text{over}}$ um Übersegmentierung zu bevorzugen, da es tragischer ist Objekte nicht zu erfassen, als sie in mehreren Teilen vorzufinden.

2.4.2 Variation of Information

$$\mathcal{L}_{\text{VOI}}(y, y^*) = H_y + H_{y^*} - 2 \cdot I(y, y^*) \quad (4)$$

$$H_y = \mathbb{E}[\hat{I}(y)] = - \sum_{l \in y} p(l) \cdot \log_e(p(l)) \quad (5)$$

$$I(y, y^*) = \sum_{l_1 \in y} \sum_{l_2 \in y^*} p(l_1, l_2) \cdot \log_e \left(\frac{p(l_1, l_2)}{p(l_1)p(l_2)} \right) \quad (6)$$

H_y ist hierbei die Entropie des Labelings y , welche als Erwartungswert der Information \hat{I} definiert ist. Jede Segmentierung besitzt eine individuelle Entropie.

$I(y, y^*)$ bezeichnet die Transinformation, anschaulich gesehen entspricht diese der Schnittmenge der Ist- und Soll-Segmentierung. Es werden also die Labels der Superpixel untersucht und bei fehlender Übereinstimmung beim Vergleich mit der Ground Truth erhöht sich der Loss.

2.5 STRUCTURED LEARNING

Allgemein wird beim Lernprozess eine Abbildung $f : X \rightarrow Y$ gesucht, welche die Feature X in einen Output Y überführt. Der Hauptunterschied zum unstrukturierten Lernen besteht in der Form dieses Outputs. Je nach Klassifikations- oder Regressionsproblem besteht beim unstrukturierten Lernen der Output entweder aus Klassen oder reellen Zahlen. Beim strukturierten Lernen hingegen existiert eine zulässige Menge an Outputs, welche einer gewissen Struktur genügen. In unserem Fall ist diese Struktur die konsistente Segmentierung. Man kann unser Problem auch als Klassifikationsproblem der Kanten mit Nebenbedingungen betrachten. Da somit die einzelnen Outputs $y \in Y$ miteinander in Verbindung stehen, muss beim Lernprozess der Loss gemeinsam für alle y berechnet werden.

Beim Multicut Problem (1) wird f durch die Weights w spezifiziert, welche je nach Abstiegsverfahren und Loss unterschiedlich erlangt werden. Die konkreten Methoden werden in den folgenden Kapiteln erläutert.

Ein weiterer Unterschied besteht in der Wahl der Loss-Funktion, welche in Kapitel 2.4 vorgestellt wurden. Beim unstrukturierten Lernen wird bei der Klassifikation oft der 0-1-Loss genommen, bei dem der Output entweder der Ground Truth entspricht, oder eben nicht. Aufgrund der Struktur der Lösung ist dies hier allerdings nicht sinnvoll, da es offensichtlich bessere und schlechtere Segmentierungen gibt. Diese Abstufungen sollten unterschieden werden, was beispielsweise durch die genannten Loss Funktionen ermöglicht wird.

2.5.1 Subgradient Descent

Der Subgradient Descent Algorithmus basiert auf der Berechnung der Differenz der akkumulierten Feature der Segmentierung y und der Ground Truth y^* , welche gewichtet den Weights w hinzu addiert werden. In [1] wird der Algorithmus ausführlich ausgeführt.

Die Berechnung des Loss findet hier beim Lösen des Multicut Problems statt, sodass dieses wie folgt aussieht:

$$\begin{aligned} \arg \min_y & \sum_{y_i \in E} \langle w, \beta_e \rangle \cdot y_i + \mathcal{L}_H(y, y^*) \\ \text{s.t.} & y - \sum_{y_i \in P(y)} y_i \leq 0 \quad \forall y \in E \end{aligned} \quad (7)$$

Die Minimierung des Hamming Loss wird in dieser Arbeit hiermit realisiert.

2.5.2 Stochastic Gradient

Der hier verwendete Stochastic Gradient ist eine Variante des in [1] näher erläuterten Algorithmus. Anders als beim Subgradient Descent wird hier zunächst das Multicut Problem ohne Loss (1) gelöst und anschließend der Loss der resultierenden Segmentierung berechnet. Im folgenden wird die hier angewandte Methode zur Ermittlung der Gradientenrichtung (Alg. 1), sowie der Liniensuche (Alg. 2), also der Schrittweite pro Iterationsschritt beschrieben:

Algorithm 1 Get Gradient Descent Direction

```

1: procedure GETGRADIENTDESCENTDIRECTION(#Perturbs,  $\sigma$ ,  $w$ )
2:    $\sigma$ : Noise standard deviation
3:    $w$ : Current Weight Vector
4:
5:    $\Delta x = 0$ 
6:   for  $n = 1 \dots \#Perturbs$  do
7:     Generate Noise  $\in \mathcal{N}(0, \sigma^2)$  und add to  $w$ 
8:     Calculate Loss on current Training Sample
9:      $\Delta x = \Delta x + \text{Noise} * \text{Loss}$ 
10:     $\Delta x = -\Delta x / \#Perturbs$ 
11:   return  $\Delta x$ 

```

Um die Gradientenrichtung zu bestimmen, wird zunächst für #Perturbs verschiedene normalverteilte Richtungen der Loss auf dem aktuellen Bild berechnet, ausgehend von den momentanen Weights. Anschließend werden die einzelnen Richtungen nach ihrem Loss gewichtet, wodurch man eine Richtung starken Anstieges ermittelt hat. Daher ist am Ende noch ein Vorzeichenwechsel nötig.

Algorithm 2 Line Search and update Weights

```

1: procedure LINESEARCHANDTAKESTEP( $w, \eta, \Delta x$ )
2:    $\eta$ : Stepwidth
3:    $w$ : Current weight vector
4:    $\Delta x$ : Gradient Descent Direction
5:
6:   for  $n = \{0.1, 0.5, 1.0, 5.0, 10.0\}$  do
7:     Varied Weight Vector  $w_{var} = w + \Delta x \cdot n$ 
8:     Calculate mean Loss  $\mathcal{L}$  on entire Training Set
9:     from  $w_{var}$ 
10:    if  $\mathcal{L} < \mathcal{L}_{best}$  then
11:       $\mathcal{L}_{best} = \mathcal{L}$ 
12:      Save  $w_{best} = w$ 
13:      Break
14:    Memorize  $\mathcal{L}$  and associated varied weight vector
15:   $w = w_{var}$ , where regarding Loss is minimal
16:  return  $w$ 

```

Für äquidistant gewählte Schrittweiten über 2 Größenordnungen wird jeweils der Mittelwert des Losses auf dem gesamten Trainingsset ermittelt und mit dem bisher besten Wert verglichen. Bei Erreichen eines neuen Tiefpunktes, wird direkt dorthin gesprungen, andernfalls wird die Schrittweite mit dem Niedrigsten erreichten Loss gewählt.

3

RELATED WORK

Elementar für die Bildsegmentierung sind Algorithmen zur Detektion von Kantenkonturen, die als Feature dienen. Als klassischen Ansatz ist hierbei die Arbeit von Arbeláez et al. [8] zu nennen, welches auf der Berechnung der lokalen posteriori Wahrscheinlichkeit von Kanten mit anschließender globaler Zusammenführung beruht. Ein anderes Beispiel ist der Kantendetektor von Dollár et al. [5], der auf einen strukturierten Random Forest setzt. Moderne Methoden, die aktuell die besten Ergebnisse erzielen, nutzen neuronale Netze. Zu nennen ist hierbei N⁴-Fields [4], sowie Holistically-Nested Edge Detection [9], welches auf ein deep convolutional neural network setzt und momentan State-of-the-Art darstellt.

Es existieren verschiedene Wege zur Segmentierung eines Bildes. Die Methode des direkten Labelings der Knoten des Graphs beim Multicut Problem basiert auf der Arbeit von Nowozin et al. [11]. Im Gegensatz dazu steht das Labeling der Kanten, welches in dieser Arbeit genutzt wird und [10] als Grundlage dient.

Ein anderer Ansatz ist ebenfalls in [8] beschrieben, welches auf der dort beschriebenen Konturdetektion aufbaut. Hierbei wird die Konsistenz der Kanten mittels einer Variation des Watershed Algorithmus [12] erzielt, was eine starke Übersegmentierung liefert. Ein Abgleich der Orientierung der Kanten der Konturdetektion liefert gute Informationen über die Stärke jeder Kante. Eine gröbere Segmentierung wird nun erreicht indem eine Hierarchie der Kanten bezüglich der Kantenstärke aufgebaut wird. Je nachdem wie fein die gewünschte Segmentierung ist, werden unterschiedlich tief in der Hierarchie die Kanten abgeschaltet und somit Gebiete zusammengeführt. Auf diesem Verfahren aufbauend liefert [13] momentan State-of-the-Art Ergebnisse bei der Segmentierung von natürlichen Bildern.

4

EXPERIMENTELLES SETUP

4.1 TRAININGS- UND TESTDATEN

Die Trainings- und Testbilder stammten vom Berkeley Segmentation Dataset (BSD-500) [3], welches aus natürlichen Bildern besteht. Der Datensatz ist gegliedert in einen Trainings-, Test- und Validierungsbereich, wobei die Test-Bilder genommen wurden, da hierfür State-of-the-Art Kantendetektoren als Feature zur Verfügung standen. Sowohl das in dieser Arbeit verwendete Trainings- als auch das Testset bestand daher aus 100 Bildern.

Die Ground Truth der verwendeten Bilder stammen ebenfalls aus dem BSD-500 Datensatz und lagen in Form eines Pixel-Labelings vor, d.h. jedem Pixel ist eine Zahl zugeordnet, zu welchem Segment es gehört. Letztendlich will man allerdings das Labeling der zuvor berechneten Superpixel haben. Hierzu wird durch alle Superpixel iteriert und jeweils die Anzahl der Label auf Pixelebene gezählt. Dem Superpixel wird nun dasjenige Label zugeordnet, welches am häufigsten auf Pixelebene vorkommt (Majority Vote).

Ein Beispiel findet sich in Abb. 4: Die farbigen Bereiche entsprechen dem Pixel-Labeling der Ground Truth, außerdem ist das Bild zuvor durch den SLIC-Algorithmus in 4 Superpixel eingeteilt worden. Für jeden Superpixel wird nun geprüft, welches Label am Häufigsten vorkommt. Für die zwei linken Superpixel ist dies der violette, oben rechts offensichtlich der blaue und unten rechts der grüne Bereich. Bei der Ground Truth ist somit nur die linke Kante inaktiv.

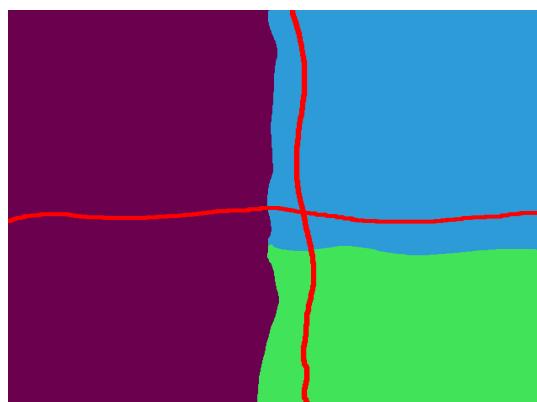


Abbildung 4: Ground Truth Labeling (farbige Bereiche) und Kanten der Superpixel vom SLIC-Algorithmus in rot

4.2 GRAPHICAL MODEL UNTERBAU UND SOLVER

Zur Generierung des Region Adjacency Graphs, des Random Forests und der Filter wurde VIGRA [6], eine Bibliothek zur Bildanalyse- und Bearbeitung, verwendet. Eine weitere genutzte Bibliothek ist Inferno [7], welche unter anderem zum Lösen von grafischen Modellen dient. Diese fand Anwendung beim Zusammenführen aller Daten, Lösen des Multicut Problems und Lernen der Parameter sowohl mit Sub-Gradient bezüglich Hamming Loss, als auch mit Stochastic Gradient bezüglich Variation of Information. Beide Bibliotheken basieren auf C++, welche allerdings über Python angesteuert werden können. Darauf wurde das komplette Programm für diese Arbeit in Python realisiert.

4.3 FEATURE SPACE

Der Feature Space wurde folgendermaßen zusammengesetzt:

- Gaussian Gradient Magnitude mit $\sigma = \{1, 2, 5\}$
- Hessian of Gaussian Eigenwerte mit $\sigma = 2$
- Laplacian of Gaussian
- Structure Tensor Eigenwerte
- Canny Filter
- N⁴-Fields Kantendetektor [4] mit und ohne Gewichtung auf Länge der Kante
- Structured Forests Kantendetektor [5, Dollár et al.] mit und ohne Gewichtung auf Länge der Kante
- Statistische Kenndaten in variablen Bereichen \bar{u} und \bar{v} um eine Kante an Superpixeln u und v (siehe Abb. 6)
(separat angewandt auf alle 3 Farbkanäle des eigentlichen Bildes, als auch auf den N⁴-Fields- und Dollár-Kantendetektor)
 - Mean($\bar{u} + \bar{v}$)
 - Variance($\bar{u} + \bar{v}$)
 - $\frac{\max\{\text{Mean}(\bar{u}), \text{Mean}(\bar{v})\}}{\min\{\text{Mean}(\bar{u}), \text{Mean}(\bar{v})\}}$
 - $\frac{\max\{\text{Median}(\bar{u}), \text{Median}(\bar{v})\}}{\min\{\text{Median}(\bar{u}), \text{Median}(\bar{v})\}}$
 - Skewness($\bar{u} + \bar{v}$)
 - Kurtosis($\bar{u} + \bar{v}$)
- Konstantes Feature für jede Kante, zur Beseitigung des Bias im Feature Space

In Abb. 7 und 8 sind einige Filter dargestellt.

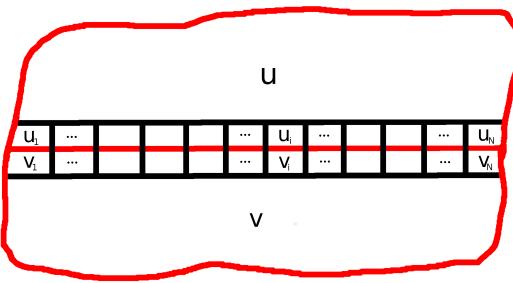


Abbildung 5: Pixel und Kanten auf Pixelebene (u_i und v_i), als auch auf Superpixelebene (u und v)

Um aus den Pixelinformationen der Filter die Gewichtungen der Superpixelkanten zu bekommen (veranschaulicht in Abb. 5) werden zunächst die Kantengewichte auf Pixelebene berechnet. Bei Pixel u_i und v_i beträgt das zugehörige Kantengewicht:

$$E_i = \frac{u_i + v_i}{2} \quad (8)$$

Anschließend wird der Mittelwert der Pixelkanten, welche zu einer Superpixelkante gehören, gebildet.

$$E_{uv} = \frac{1}{N} \sum_{i=1}^N E_i \quad (9)$$

Dies ist die gewünschte Gewichtung der Superpixelkante.

Zusätzlich wurde aus den Feature Spaces aller Trainingsdaten ein Random Forest aufgebaut und dieser zur Generierung eines Weiteren (RF Feature) verwendet.

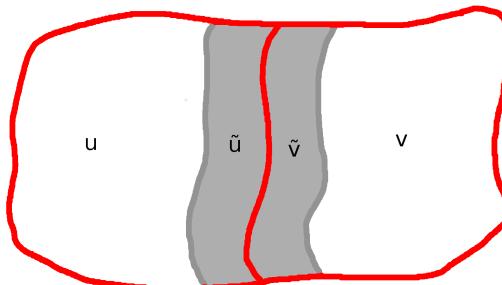


Abbildung 6: Variable Bereiche um Kante der Superpixel u und v

In Abb. 7 sind Beispiele für den State-of-the-Art Kantendetektor N^4 -Fields. Man sieht deutlich, dass Objektgrenzen registriert werden obwohl der Gradient aufgrund der Farbunterschiede nicht sehr hoch ist, beispielsweise am oberen Kopfende des Eisbären. Allerdings arbeiten auch diese Filter nicht perfekt, wie man am unteren Bereich der Hose des Mannes bei 7c sieht, diese Objektgrenze wird nicht sehr scharf registriert.

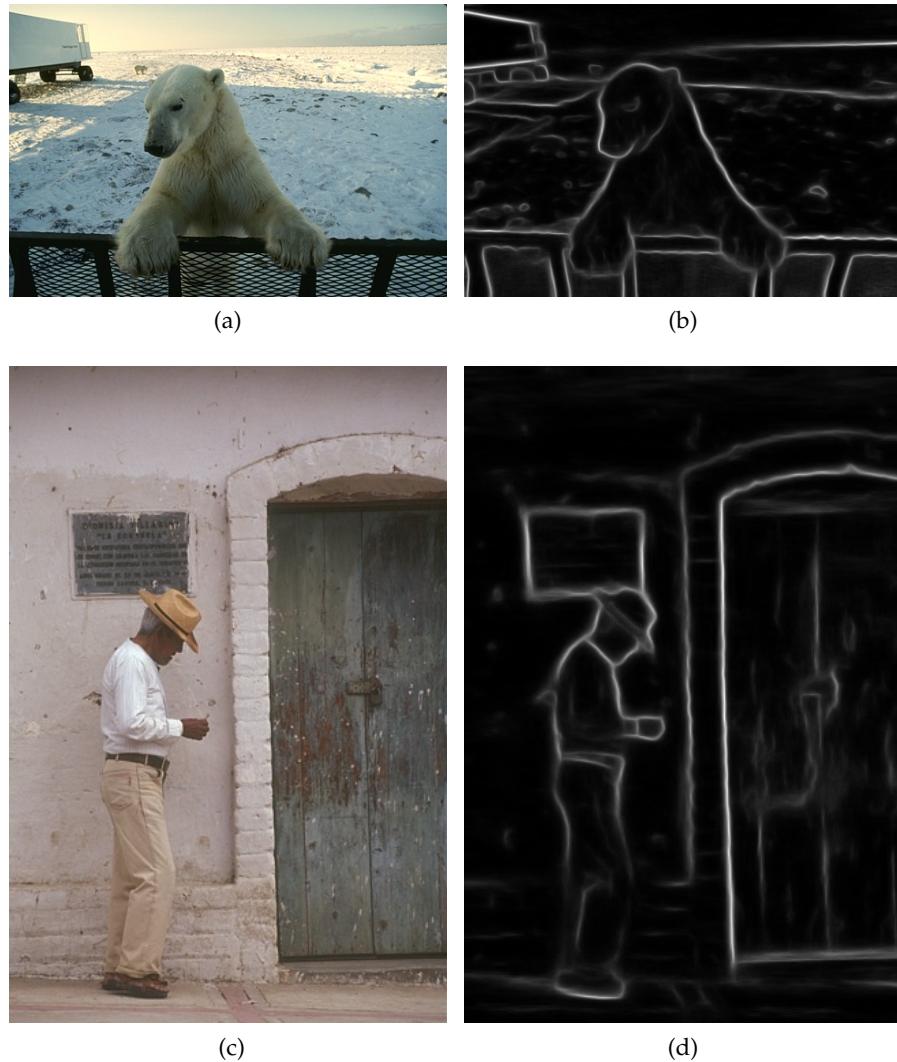


Abbildung 7: Beispiele N^4 -Fields Kantendetektor

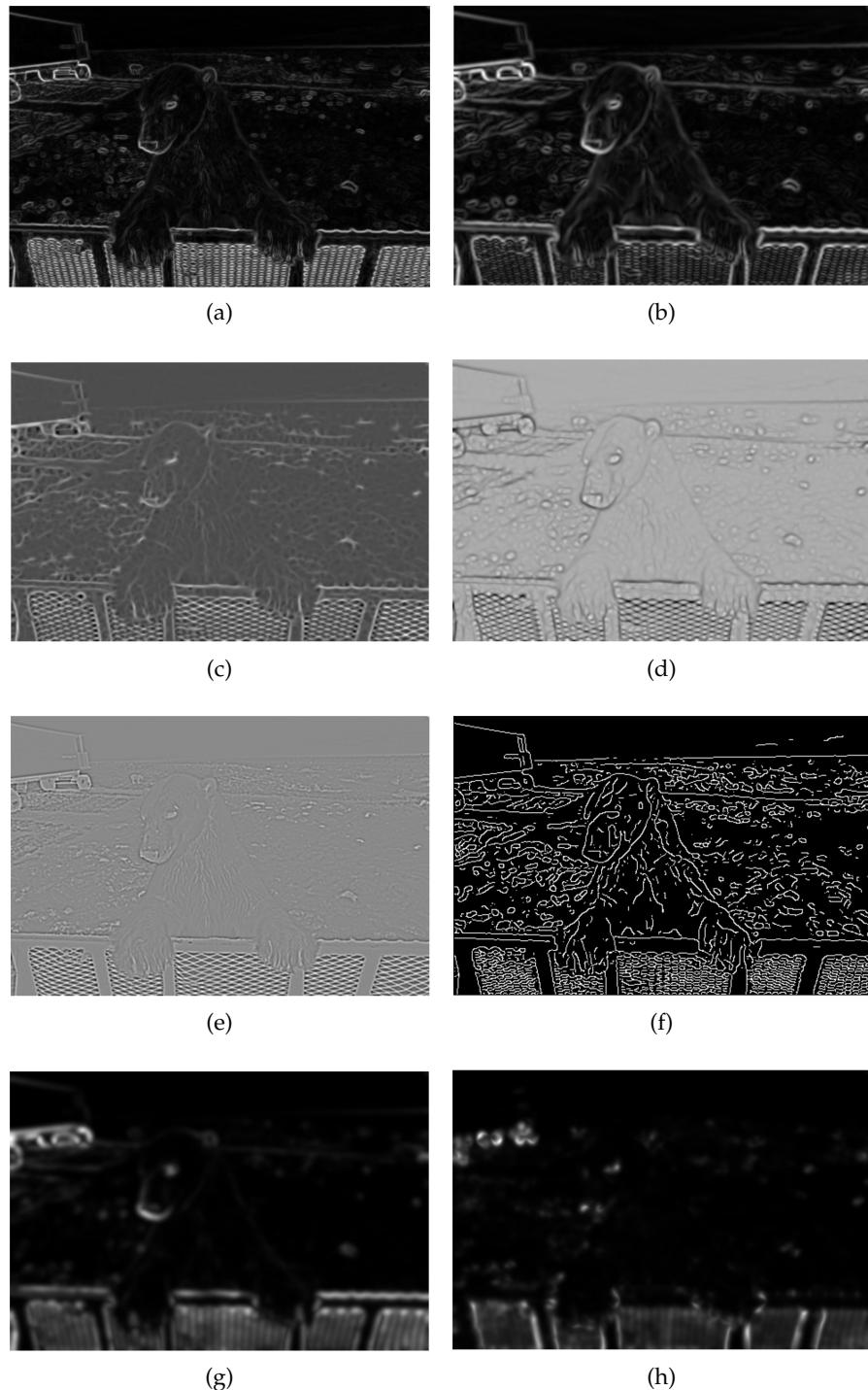


Abbildung 8: Output verschiedener Filter auf Pixelebene: (a)&(b) Gradient Magnitude mit $\sigma = \{1, 2\}$, (c)&(d) Hessian of Gaussian Eigenvalues, (e) Laplacian of Gaussian, (f) Canny, (g)&(h) Structure Tensor Eigenvalues

5

EXPERIMENTE UND RESULTATE

Die Messungen wurden mit unterschiedlichen Konfigurationen durchgeführt, wobei jeweils zuerst der Hamming Loss \mathcal{L}_H mittels Subgradient minimiert wurde. Die resultierenden Weights dienten anschließend als Startpunkt um mittels Stochastic Gradient den Variation of Information Loss \mathcal{L}_{VOI} zu optimieren und beide Ergebnisse miteinander zu vergleichen.

5.1 STOCHASTIC GRADIENT MIT RF FEATURE

SubGrad	StochGrad	\mathcal{L}_H		\mathcal{L}_{VOI}	
		SubGrad	StochGrad	SubGrad	StochGrad
Messung ①; mit RF Feature ohne Constraint 80 Iterationen $\eta = 0.1$	Feature $\in [-1, 1]$ mit RF Feature Constraint auf RF 1 Iteration #Perturbs = 3	Test Set 225.7 \pm 9.8	251.34 \pm 11	1.159 \pm 0.048	1.267 \pm 0.052
		Training Set 88.2 \pm 6.6	26.2 \pm 3.1	0.519 \pm 0.038	0.173 \pm 0.020
Messung ②; mit RF Feature ohne Constraint 80 Iterationen $\eta = 0.1$	Feature $\in [-1, 1]$ mit RF Feature ohne Constraint 1 Iteration #Perturbs = 3	Test Set 225.7 \pm 9.8	281 \pm 14	1.159 \pm 0.048	1.401 \pm 0.067
		Training Set 88.2 \pm 6.6	25.0 \pm 4.2	0.519 \pm 0.038	0.146 \pm 0.022
Messung ③; ohne RF Feature ohne Constraint 80 Iterationen $\eta = 0.1$	Feature $\in [-1, 1]$ mit RF Feature Constraint auf RF 3 Iterationen #Perturbs = 3	Test Set 238 \pm 10	227.7 \pm 9.9	1.165 \pm 0.047	1.187 \pm 0.049
		Training Set 229 \pm 11	225 \pm 10	1.068 \pm 0.047	1.024 \pm 0.045
Messung ④; ohne RF Feature ohne Constraint 80 Iterationen $\eta = 0.1$	Feature $\in [-1, 1]$ mit RF Feature ohne Constraint 3 Iterationen #Perturbs = 3	Test Set 238 \pm 10	225 \pm 10	1.165 \pm 0.047	1.163 \pm 0.049
		Training Set 229 \pm 11	223 \pm 11	1.068 \pm 0.047	1.027 \pm 0.046

Tabelle 1: Messwerttabelle 1

		\mathcal{L}_H		\mathcal{L}_{VOI}	
SubGrad	StochGrad	SubGrad	StochGrad	SubGrad	StochGrad
Messung ⑤; mit RF Feature ohne Constraint 80 Iterationen $\eta = 0.1$	Feature $\in [0, 1]$ mit RF Feature Constraint auf RF 1 Iteration $\#Perturbs = 3$	Test Set 234 ± 11	Training Set 82.3 ± 5.9	1.166 ± 0.047	1.446 ± 0.065
Messung ⑥; mit RF Feature ohne Constraint 80 Iterationen $\eta = 0.1$	Feature $\in [0, 1]$ mit RF Feature ohne Constraint 1 Iteration $\#Perturbs = 3$	Test Set 234 ± 11	Training Set 82.3 ± 5.9	1.166 ± 0.047	1.285 ± 0.056
Messung ⑦; ohne RF Feature ohne Constraint 80 Iterationen $\eta = 0.1$	Feature $\in [0, 1]$ mit RF Feature Constraint auf RF 3 Iterationen $\#Perturbs = 3$	Test Set 237 ± 11	Training Set 229 ± 10	1.197 ± 0.049	1.328 ± 0.54
Messung ⑧; ohne RF Feature ohne Constraint 80 Iterationen $\eta = 0.1$	Feature $\in [0, 1]$ mit RF Feature ohne Constraint 3 Iterationen $\#Perturbs = 3$	Test Set 237 ± 11	Training Set 229 ± 10	1.197 ± 0.049	1.283 ± 0.059

Tabelle 2: Messwerttabelle 2

Es wurden bei den Messungen 1-8 (siehe Messwerttabellen 1 & 2) die Fälle, Subgradient Descent mit bzw. ohne Random Forest Feature, ob ein Weight bei Stochastic Gradient konstant gehalten wird, und in welchem Wertebereich der Feature Space liegt, unterschieden. Die gewählten Werte der Schrittweite $\eta = 0.1$ und Noise $\sigma = 0.3$ bei Stochastic Gradient wurden in vorigen Experimenten empirisch auf den Abfall des Losses optimiert und bei allen Messungen angewandt. Da dieser Prozess bei Stochastic Gradient allerdings statistisch geschieht, war hier eine enorme Varianz der Ergebnisse zu beobachten. Hier besteht somit noch Verbesserungspotenzial, um schneller ein möglichst niedriges Minimum im Loss zu finden.

Die ermittelten Werte für den Loss ergaben sich immer aus dem Mittelwert der 100 Trainings- bzw. Testdaten und der 1σ -Fehlerbereich ergab sich als Fehler des Mittelwertes.

Erfreulich ist zunächst die Beobachtung, dass der VOI Loss mithilfe vom Stochastic Gradient Descent bei allen Messungen auf dem Trainingsset reduziert werden konnte. Der Grad, wie schnell der Abfall stattfindet hängt allerdings deutlich von der Messkonfiguration ab. Ohne Random Forest Feature beim Subgradient Descent war der Abfall beispielsweise langsamer, was 3 Iterationen im Stochastic Gradient notwendig gemacht hat. Nichtsdestotrotz ist der Abfall bei Messung 3 & 4 nur sehr gering. Zu bemerken ist hier, dass in allen Fällen damit auch eine Senkung des Hamming Loss auf dem Trainingsset einhergeht.

Betrachtet man nun jedoch den resultierenden Loss auf dem Testset, verschlechtert sich dieser in allen Messungen, außer Messung 4. Hier konnte der VOI Loss auf dem Trainingsset jedoch nicht so stark vermindert werden, wie in den anderen Fällen. Außerdem ist die Absenkung auf dem Testset äußerst minimal und ist daher eher dem Zufall geschuldet. Die Erhöhung in allen anderen Fällen ist im Vergleich deutlich stärker und trifft sowohl auf den Hamming- als auch den VOI-Loss zu.

Die Beobachtung von sinkendem Loss auf dem Trainingsset und steigendem Loss auf dem Testset, lässt somit auf einen Overfit der Daten schließen.

Auf der nächsten Seite in Abb. 9 sind Beispiele des Testsets der resultierenden Segmentierung mittels Subgradient Descent bzw. Stochastic Gradient Descent zu sehen. Die Erhöhung des Loss ist deutlich in Form einer schlechteren Segmentierung zu erkennen.



(a) SubGradient Descent
 $\mathcal{L}_H = 213.0; \mathcal{L}_{VOI} = 1.194$



(b) Stochastic Gradient Descent
 $\mathcal{L}_H = 248.5; \mathcal{L}_{VOI} = 1.880$



(c) SubGradient Descent
 $\mathcal{L}_H = 236.0; \mathcal{L}_{VOI} = 1.629$



(d) Stochastic Gradient Descent
 $\mathcal{L}_H = 414.5; \mathcal{L}_{VOI} = 2.474$



(e) SubGradient Descent
 $\mathcal{L}_H = 264.0; \mathcal{L}_{VOI} = 1.203$



(f) Stochastic Gradient Descent
 $\mathcal{L}_H = 246.5; \mathcal{L}_{VOI} = 1.823$

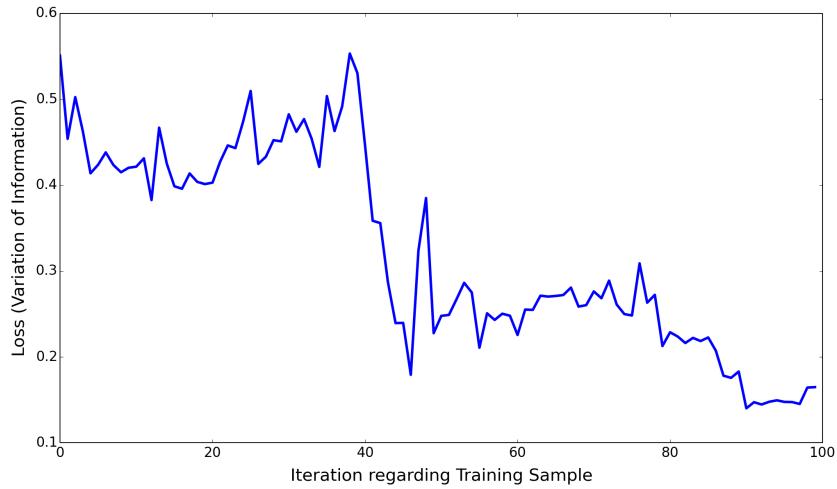
Abbildung 9: Vergleich der Segmentierungen nach Optimierung auf \mathcal{L}_H mittels Subgradient Descent bzw. \mathcal{L}_{VOI} mittels Stochastic Gradient Descent; Resultate aus Messung 2 (a&b)/ Messung 8 (c)



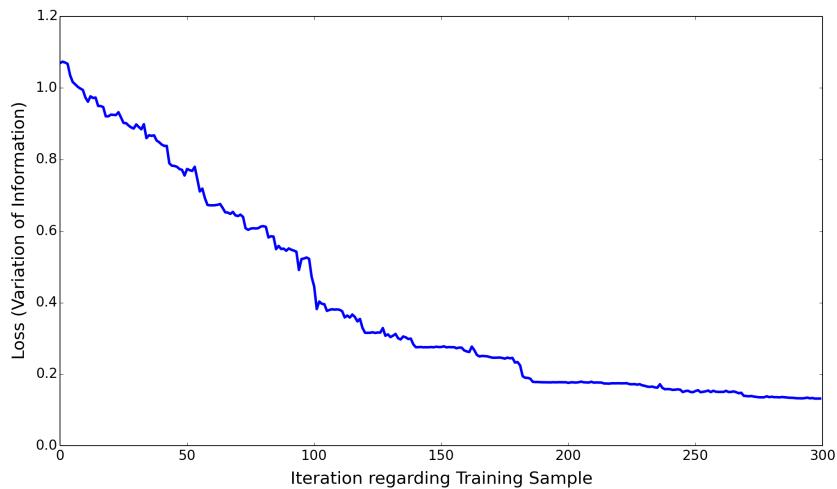
Abbildung 10: Beispiel gutes/schlechtes Loss-Maß VOI

In Abb. 10 sind nun Segmentierungen inklusive Hamming- bzw. Variation of Information-Loss in Bezug zur Ground Truth dargestellt. In beiden Fällen ist der VOI-Loss recht niedrig, was die anschauliche Interpretation von VOI bestätigt, denn in beiden Bildern ist die Fläche von falschen Segmentierungen recht klein. Bei 10a&10b beispielweise der Bereich rechts neben der Treppe oder die linke weiter hinten liegende Hauswand, beim Vogelbild der Stock und der fehlende Teil

des Vogels. Dieses Verhalten hat Vor- und Nachteile. Zu den Vorteilen gehört, dass wenigstens eher ein gewisser Teil eines zu erfassenden Objektes registriert wird, falls das Objekt nur groß genug ist. Nachteilig ist allerdings, dass kleine Objekte eher vernachlässigt werden und die Form des Objekts eher verfälscht wird.



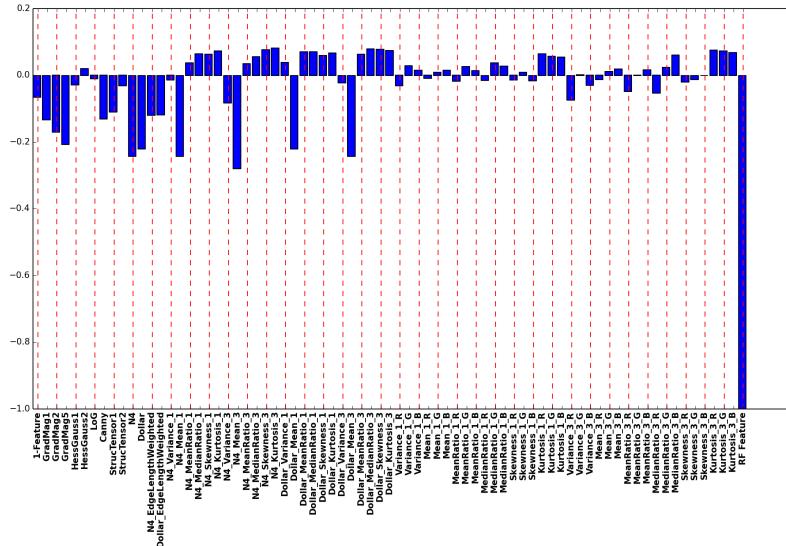
(a) Messung 2



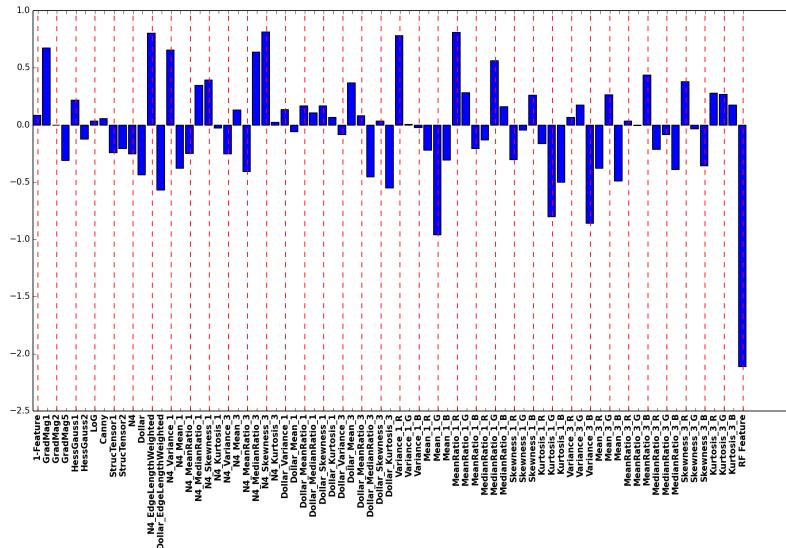
(b) Messung 8

Abbildung 11: Average Loss Verlauf beim Optimieren der Weights bezüglich Variation of Information mithilfe von Stochastic Gradient Descent

In Abb. 11 ist der Loss beim Lern-Prozess des Stochastic Gradient Descent für Messungen 2&8 zu sehen. Da bei den Messungen ohne RF Feature beim Subgradient Descent zuvor mehr Iterationen nötig waren um den Loss entsprechend abzusenken, sieht man speziell hier bei b) sehr deutlich das konvergente Verhalten.



(a) Subgradient Descent, RF Feature dominiert stark



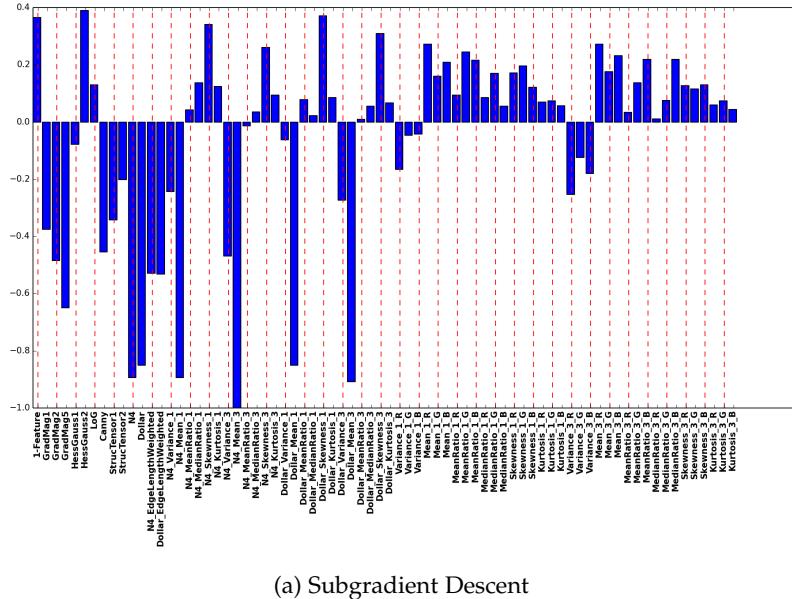
(b) Stochastic Gradient Descent

Abbildung 12: Resultierende Weights nach Optimierung auf Hamming Loss
 (a) und Variation of Information (b), Messung 2

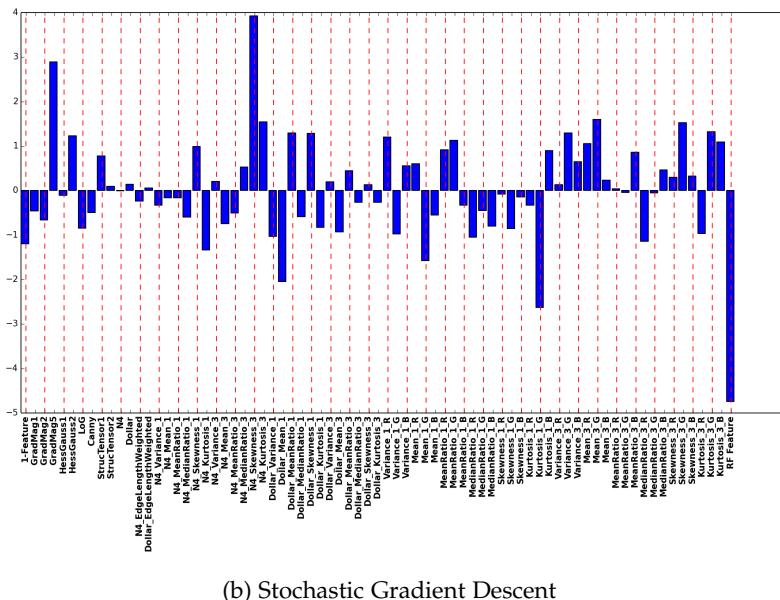
Betrachtet man die Weights nach den <jeweiligen Lernprozessen fällt beim Subgradient Descent mit RF Feature (Abb. 12) dessen Dominanz auf. Die anderen Feature wirken nur als kleine Korrekturen. Die anschließend durch Stochastic Gradient aktualisierten Weights weisen diese Dominanz nicht mehr auf. Dies ist bei allen Messungen mit RF Feature beim Subgradient Descent (Messung 1,2,5 und 6) zu beobachten und somit unabhängig vom Constraint.

Die Veränderungen bei der Konfiguration Subgradient Descent ohne RF Feature ist in Abb. 13 abgebildet. Das RF Feature wird interessanterweise im Verhältnis zu den Übrigen ähnlich stark wie bei Messung

1, die übrige Verteilung weist jedoch deutliche Unterschiede auf, trotz des ähnlich erreichten Losses im Trainingsset (0.146 bei Messung 2 zu 0.132 bei Messung 8). Dies kann allerdings auch an den unterschiedlichen Wertebereichen des Feature Space liegen.



(a) Subgradient Descent



(b) Stochastic Gradient Descent

Abbildung 13: Resultierende Weights nach Optimierung auf Hamming Loss
 (a) und Variation of Information (b), Messung 8

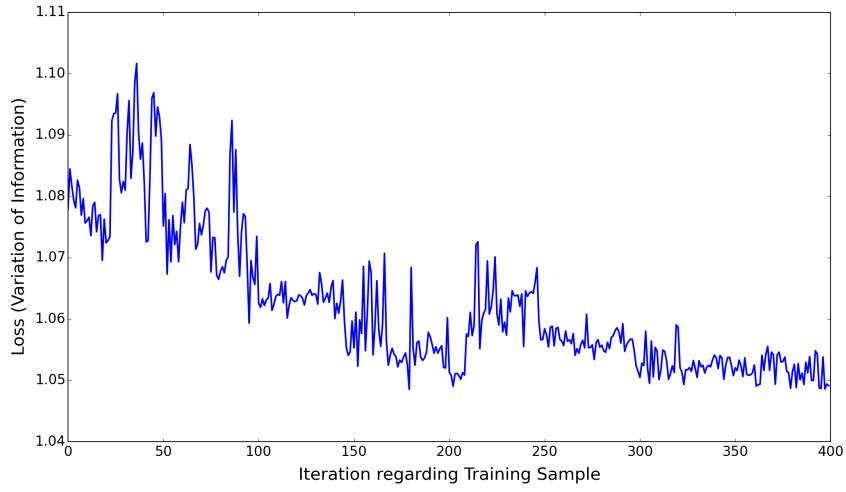
5.2 STOCHASTIC GRADIENT OHNE RF FEATURE

Die in Messwerttabelle 3 aufgeführten Experimente wurden gänzlich ohne das RF Feature durchgeführt, da sich in vorigen kleineren Tests gezeigt hat, dass so der Loss nicht so stark abfällt und man somit möglicherweise den Overfit vermeiden kann. Die Messkonfiguration wurde hinsichtlich der Normierung des Feature Space, sowie bei der Wahl, ob mit oder ohne Constraint, variiert. Bei vorhandenem Constraint wurde das N^4 -Fields Feature auf den Wert fixiert den es nach Subgradient Descent hatte, da es dort eins der Dominierenden war. Aufgrund der geringeren Konvergenzgeschwindigkeit im Vergleich zu den vorigen Messungen war eine höhere Anzahl an Iterationen notwendig und außerdem hat sich gezeigt, dass eine leicht erhöhte Anzahl an Perturbationen im Stochastic Gradient Algorithmus sich positiv auf die Konvergenzgeschwindigkeit auswirkt.

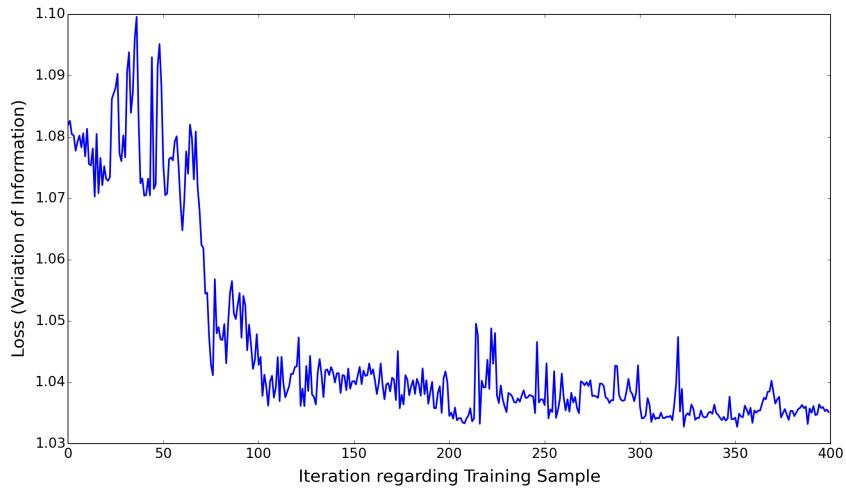
SubGrad	StochGrad	\mathcal{L}_H		\mathcal{L}_{VOI}	
		SubGrad	StochGrad	SubGrad	StochGrad
Messung ⑨; ohne RF Feature ohne Constraint 80 Iterationen $\eta = 0.1$	Feature $\in [-1, 1]$ ohne RF Feature Constraint auf N^4 3 Iterationen $\#Perturbs = 3$	Test Set 238 ± 10	Test Set 228 ± 11	Test Set 1.165 ± 0.047	Test Set 1.148 ± 0.049
Messung ⑩; ohne RF Feature ohne Constraint 80 Iterationen $\eta = 0.1$	Feature $\in [-1, 1]$ ohne RF Feature ohne Constraint 7 Iterationen $\#Perturbs = 6$	Test Set 238 ± 10	Test Set 228.2 ± 9.9	Test Set 1.165 ± 0.047	Test Set 1.179 ± 0.048
Messung ⑪; ohne RF Feature ohne Constraint 80 Iterationen $\eta = 0.1$	Feature $\in [0, 1]$ ohne RF Feature Constraint auf N^4 4 Iterationen $\#Perturbs = 6$	Test Set 237 ± 11	Test Set 246 ± 12	Test Set 1.196 ± 0.049	Test Set 1.196 ± 0.050
Messung ⑫; ohne RF Feature ohne Constraint 80 Iterationen $\eta = 0.1$	Feature $\in [0, 1]$ ohne RF Feature ohne Constraint 4 Iterationen $\#Perturbs = 6$	Test Set 237 ± 11	Test Set 228 ± 10	Test Set 1.196 ± 0.049	Test Set 1.147 ± 0.049

Tabelle 3: Messwerttabelle 3

Es konnte zwar bei allen Messungen eine Verminderung des VOI-Loss beim Training erreicht werden, diese war allerdings zu klein um eine statistisch relevante Verbesserung im Test-Set zu beobachten. Der erreichte Loss mittels Subgradient liegt in allen Fällen im 1σ -Bereich des mittels Stochastic Gradient erreichten Wertes. Positiv zu Bemerken ist hierbei, dass kein Overfit wie in der ersten Testreihe mehr auftritt. Außerdem bedeutet dies, dass die Ergebnisse per Hamming- bzw. VOI-Loss im Mittel durchaus vergleichbar sind.



(a) Messung 11



(b) Messung 12

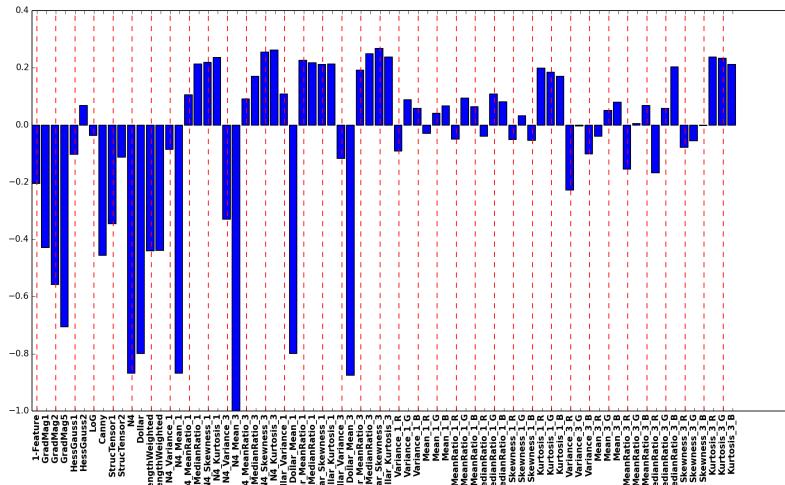
Abbildung 14: Average Loss Verlauf beim Optimieren der Weights bezüglich Variation of Information mithilfe von Stochastic Gradient Descent

In Abb. 14 ist nun erneut beispielhaft für Messung 11 und 12 der Abfall des Variation of Information Loss beim Parameter lernen mittels Stochastic Gradient zu sehen. Auch hier ist das asymptotische

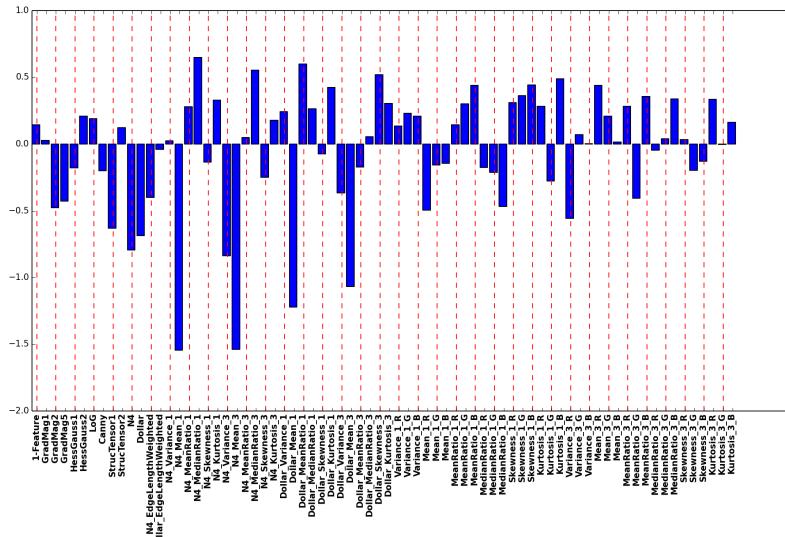
Verhalten erkennbar, was zur Annahme führt, dass das globale Minimum des Loss nahezu erreicht wurde. Der starke Abfall bei Messung 12 Iteration 80 bietet zwei Interpretationsmöglichkeiten über die Struktur des Loss-Raums. Entweder es existiert ein räumlich eng begrenztes Minimum, in das der Stochastic Gradient fällt und anschließend nicht mehr heraus kommt. Oder aber der Loss-Raum ist teilweise nicht sonderlich stetig und fällt an dieser Stelle in eine tiefere Ebene.

Interessant ist, dass der Loss Verlauf bei Messung 9/11 und 10/12 sehr ähnlich sind. Vor allem ist sowohl in Messung 10, als auch in Messung 12 der "Knick" vorhanden, was offenbar auf das fehlende Constraint zurückzuführen ist.

Weiter ist es bemerkenswert und nicht offensichtlich, dass die Ergebnisse der Minimierung des Hamming- und Variation of Information-Loss sich so ähnlich sind. Dies lässt sich allerdings durch die Vorauswahl der Kanten durch den SLIC Algorithmus erklären, wodurch diese in den meisten Fällen schon sehr gut um die Objektkanten herum gelegt werden. Dies reduziert die Vieldeutigkeit, welche Kanten an oder aus sein sollen, deutlich. Es bleiben dabei natürlich noch Fälle übrig, die in Kapitel 1.1 erwähnt wurden.



(a) Subgradient Descent



(b) Stochastic Gradient Descent

Abbildung 15: Resultierende Weights nach Optimierung auf Hamming Loss
 (a) und Variation of Information (b), Messung 10

Erwähnenswert ist die starke Veränderung der Weights beim Stochastic Gradient (siehe Abb. 15), trotz der Tatsache dass der Loss auf dem Trainingsset nicht stark gesenkt werden konnte. Dies spricht, zusammen mit Abb. 12 dafür, dass es einen recht großen Bereich gibt, in welchem der Loss minimal ist und es keine einzigartige Konfiguration der Weights existiert, um dieses zu erreichen.

5.3 KREUZVALIDIERUNG MESSUNG 10

Da wegen der großen Fehlerbereiche keine Aussage möglich war, ob es tatsächlich zu einer Verbesserung oder Verschlechterung kam, wurde noch eine Kreuzvalidierung einer Konfiguration durchgeführt. Diese Konfiguration ist die gleiche, wie in Messung 10, da hier der VOI-Loss auf dem Trainingsset am stärksten gesenkt werden konnte. Es wurden beim Stochastic Gradient jeweils 3 Iterationen durchgeführt.

Der Datensatz aus insgesamt 200 Bildern wurde hierbei in 5 gleich große Abschnitte aufgeteilt, so dass das jeweilige Trainingsset immer aus 160 und das Testset aus 40 Bildern bestand.

#	\mathcal{L}_H			\mathcal{L}_{VOI}		
	SubGrad	StochGrad	$\frac{\text{StochGrad}}{\text{SubGrad}}$	SubGrad	StochGrad	$\frac{\text{StochGrad}}{\text{SubGrad}}$
1	227.9	229.9	1.0088	1.0343	1.0646	1.0293
2	215.4	210.0	0.9749	1.0576	1.0416	0.9849
3	245.2	241.2	0.9837	1.1922	1.1671	0.9790
4	239.6	238.5	0.9954	1.1745	1.1787	1.0036
5	224.5	220.6	0.9826	1.1182	1.1357	1.0157
				0.989 \pm 0.005		
				1.0025 \pm 0.0084		

Tabelle 4: Messwerttabelle Testset Kreuzvalidierung

Aufgrund des eher kleinen Testsets variiert der erreichte Loss von Messung zu Messung recht stark. Daher wurde das Verhältnis der Änderung individuell für jede Messung berechnet und daraus der Mittelwert bzw. der Fehler des Mittelwerts gebildet. Interessanterweise kam es nur bezüglich des Hamming-Loss zu einer Verbesserung und der Wert 1.0, welcher keiner Veränderung entspricht, liegt nur knapp außerhalb des 2σ -Fehlerbereiches. Beim Variation of Information-Loss hingegen kam es praktisch zu keiner Veränderung.

In Abb. 16 sind die Trends des Loss beim Stochastic Gradient Descent dargestellt. Abgesehen von 16a ist deutlich das konvergente Verhalten zu erkennen, was ausschließt, dass schlicht nicht lange genug iteriert wurde.

Es konnte also weiter verfestigt werden, dass es nicht zu signifikanten Veränderungen kommt bei der Optimierung des Variation of Information, statt des Hamming Loss.

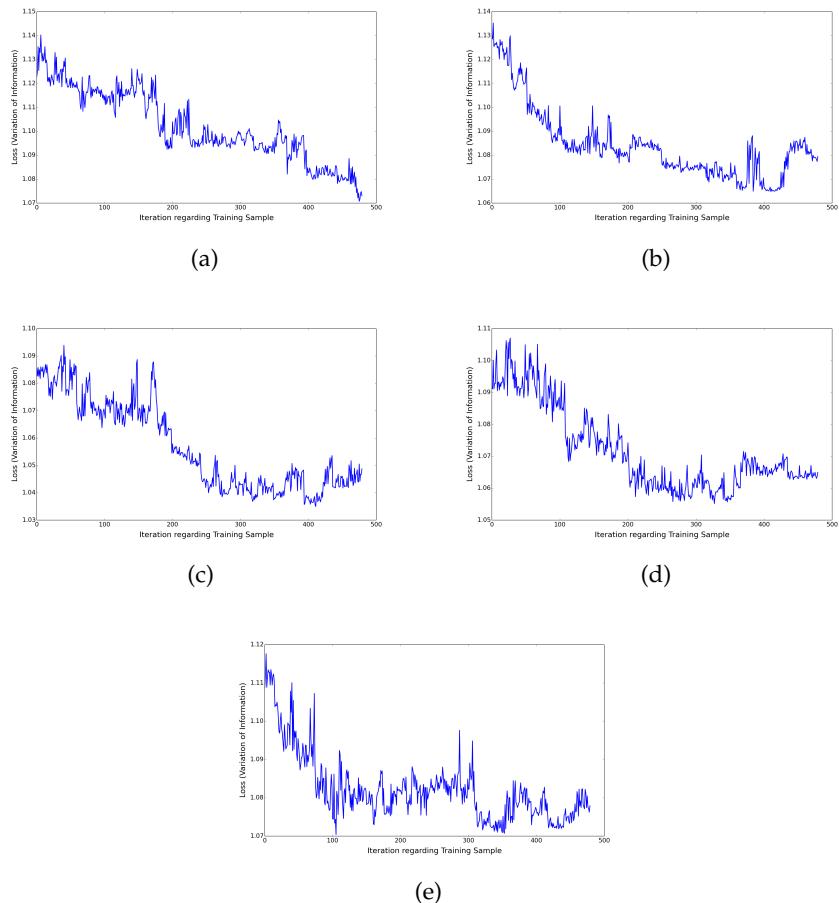


Abbildung 16: Verlauf des Loss beim Stochastic Gradient Descent

FAZIT

Es konnte nicht bestätigt werden, dass eine Nachiteration zur Optimierung des Variation of Information Loss zu einer Verbesserung der Segmentierung führt. Mit RF Feature kam es zu einem Overfit der Trainingsdaten und das Ergebnis auf den Testdaten hat sich rapide verschlechtert. Ohne RF hingegen konnte der VOI Loss auf dem Trainingsset nur geringfügig minimiert werden und aufgrund des hohen Fehlers der Messung war keine Aussage darüber möglich ob die teilweise aufgetrennten minimalen Verbesserungen auf dem Testset nun zufällig waren oder nicht.

Auch bei der nachträglichen Kreuzvalidierung hat sich keine signifikante Veränderung ergeben, obwohl der Fehlerbereich verkleinert werden konnte. Dies ist bemerkenswert, da die zwei Arten der Betrachtung über den Hamming- bzw. VOI-Loss doch recht unterschiedlich sind.

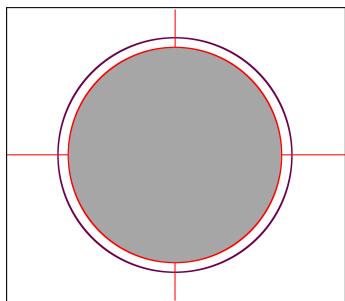


Abbildung 17: Kompensation schlechter GT durch SLIC

Eine mögliche Erklärung für dieses Verhalten liefert der recht gut funktionierende SLIC-Algorithmus, welcher nur die relevanten Kanten für den Region Adjacency Graph liefert. Die genannten möglichen Probleme beim Hamming Loss wie eine leicht verschobene Ground Truth könnten durch die recht großen Superpixel kompensiert werden da das Ground Truth Labeling auf einem Majority Vote basiert. Ein Beispiel liefert Abb. 17: Hierbei ist grau die zu segmentierende Struktur, violett die zu-

gehörige schlechte Ground Truth auf Pixel-Ebene und rot die Kanten der Superpixel, welche aufgrund des SLIC Algorithmus an der Objektgrenze liegen. Nun wird für jeden Superpixel das am meisten vorherrschende Label ausgewählt um die Ground Truth auf Superpixel-Ebene zu erzeugen. Diese beinhaltet den ursprünglichen Fehler allerdings nicht mehr.

Interessant für die Zukunft wäre eine Untersuchung des Problems auf Pixel- statt auf Superpixel-Ebene, was allerdings approximative Methoden voraussetzt da das Multicut Problem NP-Hard ist, oder aber man ist gezwungen auf Fortschritte bei Quantencomputern zu warten.

Es wurde der Verlauf des Variation of Information Loss beim Stochastic Gradient Descent aufgezeigt, an welchem das konvergente Verhalten meist sehr deutlich zu sehen war. Dies schließt die Erklärung einer unzureichenden Anzahl an Iterationen aus.

Da allerdings sehr wenig über die Struktur der Funktion bekannt ist,

welche die Weights auf den Variation of Information Loss abbildet, ist es nicht auszuschließen, dass ein Minimum existiert, welches auf den Testdaten bessere Resultate liefert. Durch den Startpunkt der Weights auf dem Ergebnis des Subgradient Descent ist das Minimum, in das der Stochastic Gradient läuft, schon vorbestimmt und möglicherweise nicht optimal.

Anhand einzelner Beispiele wurde die anschauliche Bedeutung vom Variation of Information Loss verdeutlicht und konnte so auf seine theoretischen Vorteile eingegangen werden, wie die größere Unabhängigkeit vom genauen Ground Truth Verlauf. Ebenso wurden die Nachteile aufgezeigt, wie die Vernachlässigung kleiner Objekte sowie die Verfälschung der Objektform, was insbesondere die Aufgabe des Zuordnens erschwert, um welches Objekt es sich handelt.

LITERATURVERZEICHNIS

- [1] Sebastian Nowozin, Christoph H. Lampert, *Structured Learning and Prediction in Computer Vision*, 2011.
- [2] Radhakrishna Achanta, Appu Shaji, Kevin Smith, Aurelien Lucchi, Pascal Fua, and Sabine Süsstrunk, *SLIC Superpixels Compared to State-of-the-art Superpixel Methods*, 2011
- [3] D. Martin, C. Fowlkes, D. Tal and J. Malik, *A Database of Human Segmented Natural Images and its Application to Evaluating Segmentation Algorithms and Measuring Ecological Statistics*, Proc. 8th Int'l Conf. Computer Vision, July 2001
- [4] Yaroslav Ganin and Victor Lempitsky, *N⁴-Fields: Neural Network Nearest Neighbor Fields for Image Transforms*, Skolkovo Institute of Science and Technology, 2014
- [5] Piotr Dollár and C. Lawrence Zitnick, *Structured Forests for Fast Edge Detection*, ICCV 2013
- [6] Ullrich Köthe, *Vision with Generic Algorithms*, Image Processing and Analysis Library, Version 1.10.0
- [7] Thorsten Beier, *Inferno*, <https://github.com/DerThorsten/inferno>
- [8] P. Arbeláez, M. Maire, C. Fowlkes and J. Malik, *Contour Detection and Hierarchical Image Segmentation*, IEEE TPAMI, Vol. 33, No. 5, pp. 898-916, May 2011
- [9] Saining Xie and Zhuowen Tu, *Holistically-Nested Edge Detection*, CoRR, abs/1504.06375, 2015
- [10] Bjoern Andres, Joerg H. Kappes, Thorsten Beier, Ullrich Köthe and Fred A. Hamprecht, *Probabilistic Image Segmentation with Closedness Constraints* ICCV, Proceedings, (2011), 2611 - 2618
- [11] Sebastian Nowozin and Christoph H. Lampert, *Global Connectivity Potentials for Random Field Models*, IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2009)
- [12] S. Beucher and F. Meyer, *Mathematical Morphology in Image Processing*, Marcel Dekker, 1992, ch. 12
- [13] Jordi Pont-Tuset and Pablo Arbelaez and Jonathan T. Barron and Ferran Marqués and Jitendra Malik, *Multiscale Combinatorial Grouping for Image Segmentation and Object Proposal Generation*, CoRR, abs/1503.00848, 2015

ERKLÄRUNG

Ich versichere, dass ich diese Arbeit selbstständig verfasst und keine anderen Hilfsmittel und Quellen als die angegebenen verwendet habe.

Heidelberg, Januar 2016

Jan Lammel