

BACHELOR THESIS

JAN LAMMEL

BETREUT VON PROF. DR. FRED HAMPRECHT



**Untersuchung des Multicut Problems bei Verwendung einer Variation of
Information Loss Funktion**

September 2015

ABSTRACT

kommt wenn des deutsche als ok befunden wurde...

ZUSAMMENFASSUNG

Es existieren verschiedene Arten, wie die Loss Funktion einer Segmentierung definiert werden kann. Eine bisher oft verwendete Methode ist der Hamming Loss, bei der jede einzelne Edge mit der Ground Truth verglichen wird und bei fehlender Übereinstimmung der Loss steigt. Dies hat verschiedene Nachteile wie die Abhängigkeit 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 Edges, 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 zu einem Overfit der Trainingsdaten, wodurch die erhofften Verbesserungen nicht eintraten.

INHALTSVERZEICHNIS

1	EINLEITUNG	1
1.1	Motivation Variation of Information	1
2	THEORETISCHE GRUNDLAGEN	3
2.1	Graphen Theorie	3
2.2	Feature Space	3
2.3	Das Multicut Problem	3
2.4	Loss Funktionen	4
2.4.1	Partition Hamming	4
2.4.2	Variation of Information	4
2.5	Structured Learning	4
2.5.1	Subgradient Descent	5
2.5.2	Stochastic Gradient	5
3	EXPERIMENTELLES SETUP	7
3.1	Trainings- und Testdaten	7
3.2	Graphical Model Unterbau und Solver	7
3.3	Feature Space	7
4	EXPERIMENTE UND MESSDATEN	9
4.1	BSD-500	9
I	APPENDIX	17
A	APPENDIX TEST	19
	LITERATURVERZEICHNIS	21

EINLEITUNG

Das weite Forschungsfeld der Bildsegmentierung handelt von 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)

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 wird nun die neue Methode Variation of Information zur Quantifizierung der Qualität einer Segmentierung vorgestellt und mit einer bestehenden (Partition Hamming) 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 bei Partition Hamming jede einzelne Kante der Segmentierung überprüft, ob diese in der Ground Truth ebenso vorhanden ist oder nicht. Bei einer leicht verschobenen Segmentierung (siehe Bild) 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.

Mit Variation of Information wird nun versucht, diese Nachteile zu beseitigen. 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.

Bild veranschaulicht dies nun...

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 (Zitat!) in Superpixel unterteilt, dessen Ränder möglichst an den Objektkonturen verlaufen. Das Ergebnis hiervon ist in (Abb verlinken) abgebildet.

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

(hier SLIC-partition Bild und RAG-Bild einfügen)

Bei der letztlichen Segmentierung geht es darum, ein konsistentes Labeling der Superpixel zu erreichen. Dies wird über die Aktivität der Edges erreicht, welche an- oder ausgeschaltet sein können. Für die Aktivität einer Edge y gilt somit: $y \in \{0, 1\}$

Konsistent ist eine Segmentierung genau dann, wenn bei aktiven Edges die zugehörigen Superpixel verschiedene Labels haben und analog bei inaktiven Edges die Superpixel die gleichen Labels. Anschaulich gesehen ist dies der Fall, wenn alle aktiven Edges geschlossene Linien bilden.

2.2 FEATURE SPACE

Der Feature Space $X \in \mathbb{R}^{|E| \times D}$ ordnet jeder Edge D Features zu, die möglichst in Korrelation zur Frage stehen, ob die betrachtete Edge nun aktiv oder inaktiv sein soll.

2.3 DAS MULTICUT PROBLEM

Anhand dieser gewichteten Edges eine konsistente Segmentierung zu erhalten wird als Multicut Problem (MP) bezeichnet. Es wird durch folgendes Minimierungsproblem beschrieben:

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

Hierbei entspricht $w \in \mathbb{R}^D$ den Weights und $\beta_e \in \mathbb{R}^D$ den Funktionswerten der D extrahierten Informationen des Feature Spaces. Die Nebenbedingungen erzwingen die Konsistenz der Segmentierung. $P(y)$

ist hierbei der kürzeste Pfad über inaktive Edges der beiden Superpixel, die benachbart zu y sind. In der Praxis wird das Minimierungsproblem zunächst ohne Constraints gelöst und anschließend solange für Edges 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 neue Methode "Variation of Information" vorgestellt und mit der bestehenden "Partition Hamming" verglichen.

2.4.1 Partition Hamming

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

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

Es werden direkt die Edges 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.

2.4.2 Variation of Information

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

H_x ist hierbei die Entropie der Segmentierung x . Jede Segmentierung besitzt eine individuelle Entropie.

$I(x, x^*)$ 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

Um später mithilfe des Multicut Algorithmus Bilder möglichst gut segmentieren zu können, muss der Parameter w bestimmt werden. "Möglichst gut" bedeutet hier in Bezug auf eine Loss Funktion, die als Qualitätskriterium dient. Da ein niedriger Loss für eine gute Segmentation steht ist also das folgende Minimierungsproblem zu lösen:

$$\hat{w} = \arg \min_w \mathcal{L}(y, y^*) \quad (5)$$

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 dem weight Vector w hinzugeaddiert werden. Für nähere Details siehe [1].

Die Minimierung des Partition Hamming Losses in dieser Arbeit wird hiermit realisiert.

2.5.2 Stochastic Gradient

Der hier verwendete Stochastic Gradient ist eine Variante des in [1] näher erläuterten Algorithmus. 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 Wector
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 also zunächst in n Perturbs verschiedene normalverteilte Richtungen, vom momentanen Weight Vector aus, der Loss auf dem aktuellen Bild berechnet. 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 for-loop
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 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

EXPERIMENTELLES SETUP

3.1 TRAININGS- UND TESTDATEN

Zum einen wurden Experimente an kleinen synthetisch erzeugten Bildern durchgeführt um die prinzipiellen Vorteile von Variation of Information zu demonstrieren.

Zum Anderen diente für die umfassenderen Experimente zur praktischen Anwendung das Berkeley Segmentation Dataset (BSD-500) [2], welches aus natürlichen Bildern besteht. Es wurden hiervon die Test-Bilder genommen, da hierfür State of the Art Kantendetektoren als Feature zur Verfügung standen. Sowohl das Trainings- als auch das Testset bestand aus 100 Bildern.

3.2 GRAPHICAL MODEL UNTERBAU UND SOLVER

Zur Generierung des Region Adjacency Graphs, des Random Forests und der Filter wurde VIGRA [5] verwendet. Inferno [6] zum Zusammenführen aller Daten, Lösen des Multicut Problems und Lernen der Parameter sowohl mit SubGradient bezüglich Partition Hamming, als auch mit Stochastic Gradient bezüglich Variation of Information. Beide Bibliotheken basieren auf C++, welche allerdings über Python angesteuert werden können. Daher wurde das komplette Programm für diese Arbeit in Python realisiert.

3.3 FEATURE SPACE

Für die synthetischen Daten wurden folgende Feature gewählt:

- Gaussian Gradient Magnitude mit $\sigma = 1$
- Hessian of Gaussian Eigenvalues mit $\sigma = 1$
- Structure Tensor Eigenvalues

Beim BSD hat sich der Feature Space folgendermaßen zusammengesetzt:

- Gaussian Gradient Magnitude mit $\sigma = \{1, 2, 5\}$
- Hessian of Gaussian Eigenvalues mit $\sigma = 2$
- Laplacian of Gaussian
- Structure Tensor Eigenvalues
- Canny Filter
- N⁴-Fields Kantendetektor [3]

- Structured Forests Kantendetektor [4, Dollár et al.]
- Statistische Kenndaten in variablen Bereichen \bar{u} und \bar{v} um eine Edge an Superpixeln u und v
(seperat angewandt auf alle 3 Farbkanäle des eigentlichen Bildes, als auch auf den N^4 -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 Edge, zur Beseitigung des Bias im Feature Space

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

4

EXPERIMENTE UND MESSDATEN

4.1 BSD-500

Die Experimente bestanden aus Messungen mit unterschiedlichen Konfigurationen, wobei jeweils zuerst der Hamming-Loss \mathcal{L}_{PH} mittels Subgradient minimiert wurde. Der resultierende Weight Vector dient anschließend als Startpunkt um mittels Stochastic Gradient den Variation of Information Loss \mathcal{L}_{VOI} zu optimieren.

SubGrad	StochGrad	\mathcal{L}_{PH}		\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 ± 9.8	251.34	1.159 ± 0.048	1.2667
Training Set 88.2 ± 6.6	26.24	0.519 ± 0.038	0.1726		
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 ± 9.8	280.51	1.159 ± 0.048	1.4009
Training Set 88.2 ± 6.6	24.95	0.519 ± 0.038	0.1460		
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 ± 10	227.73	1.165 ± 0.047	1.1871
Training Set 229 ± 11	225.03	1.068 ± 0.047	1.0239		
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 ± 10	225.32	1.165 ± 0.047	1.1628
Training Set 229 ± 11	222.86	1.068 ± 0.047	1.0270		

Tabelle 1: Messwerttabelle 1

		\mathcal{L}_{PH}		\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 $\# \text{Perturbs} = 3$	Test Set 234 ± 11	306.41	1.166 ± 0.047	1.4457
Messung ⑥; mit RF Feature ohne Constraint 80 Iterationen $\eta = 0.1$	Feature $\in [0, 1]$ mit RF Feature ohne Constraint 1 Iteration $\# \text{Perturbs} = 3$	Test Set 234 ± 11	248.85	1.166 ± 0.047	1.2847
Messung ⑦; ohne RF Feature ohne Constraint 80 Iterationen $\eta = 0.1$	Feature $\in [0, 1]$ mit RF Feature Constraint auf RF 3 Iterationen $\# \text{Perturbs} = 3$	Test Set 237 ± 11	260.13	1.197 ± 0.049	1.3279
Messung ⑧; ohne RF Feature ohne Constraint 80 Iterationen $\eta = 0.1$	Feature $\in [0, 1]$ mit RF Feature ohne Constraint 3 Iterationen $\# \text{Perturbs} = 3$	Test Set 237 ± 11	247.16	1.197 ± 0.049	1.2830
		Training Set 82.3 ± 5.9	41.58	0.492 ± 0.034	0.2558
		Training Set 82.3 ± 5.9	42.83	0.492 ± 0.034	0.2585
		Training Set 229 ± 10	159.44	1.083 ± 0.044	0.8607
		Training Set 229 ± 10	17.87	1.083 ± 0.044	0.1321

Tabelle 2: Messwerttabelle 2

Bisherige Analyse beschränkt sich auf Messung 1-8....

Es wurde bei den Messungen (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 angewendet. Da dieser Prozess bei Stochastic Gradient allerdings statistisch geschieht, war hier eine enorme Varianz der Ergebnisse zu beobachten. Hier besteht somit definitiv noch Verbesserungspotenzial, um schneller ein möglichst niedriges Minimum im Loss zu finden.

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 PH-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 sehr stark vermindert werden, wie in den anderen Fällen. Außerdem ist die Absenkung 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 PH- 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. 1 sind Beispiele des Test-Sets 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
 $\mathcal{L}_{\text{PH}} = 213.0; \mathcal{L}_{\text{VOI}} = 1.1938$

(b) Stochastic Gradient Descent
 $\mathcal{L}_{\text{PH}} = 248.5; \mathcal{L}_{\text{VOI}} = 1.8802$



(c) SubGradient Descent
 $\mathcal{L}_{\text{PH}} = 236.0; \mathcal{L}_{\text{VOI}} = 1.6293$

(d) Stochastic Gradient Descent
 $\mathcal{L}_{\text{PH}} = 414.5; \mathcal{L}_{\text{VOI}} = 2.4742$



(e) SubGradient Descent
 $\mathcal{L}_{\text{PH}} = 189.0; \mathcal{L}_{\text{VOI}} = 0.7880$

(f) Stochastic Gradient Descent
 $\mathcal{L}_{\text{PH}} = 137.5; \mathcal{L}_{\text{VOI}} = 0.4722$

Abbildung 1: Vergleich der Segmentierungen nach Optimierung auf \mathcal{L}_{PH} mittels Subgradient Descent bzw. \mathcal{L}_{VOI} mittels Stochastic Gradient Descent; Resultate aus Messung 2 (a&b) / Messung 8 (c)



Abbildung 2: Beispiel gutes/schlechtes Loss-Maß VOI

In Abb. 2 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 a)&b) beispielsweise der Bereich rechts neben der Treppe oder die linke weiter hinten lie-

gende Hauswand. Beim Vogelbild der Stock und der fehlende Teil des Vogels. Es hängt nun

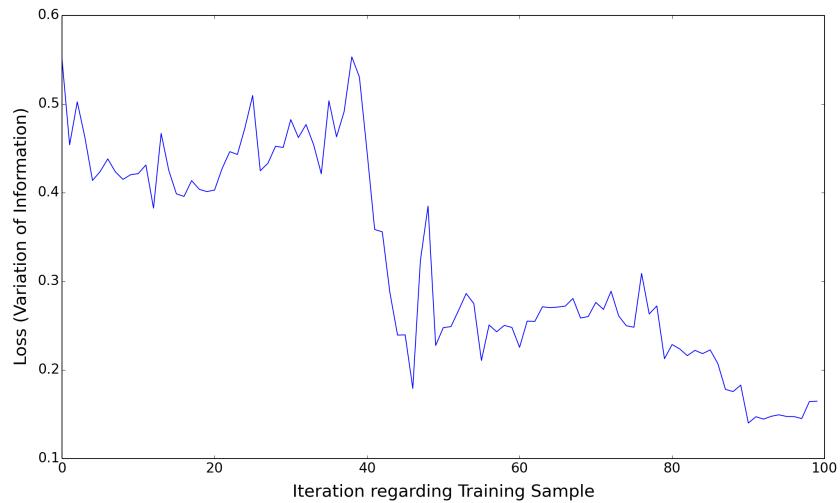


Abbildung 3: Average Loss Verlauf beim Optimieren der Weights bezüglich Variation of Information mithilfe von Stochastic Gradient Descent; **Messung ②**

		\mathcal{L}_{PH}		\mathcal{L}_{VOI}	
SubGrad	StochGrad	SubGrad	StochGrad	SubGrad	StochGrad
Messung ⑨; Feature $\in [-1, 1]$ ohne RF Feature ohne Constraint 80 Iterationen $\eta = 0.1$	Feature $\in [-1, 1]$ ohne RF Feature Constraint auf \mathbb{N}^4 3 Iterationen $\# \text{Perturbs} = 3$	Test Set 237.66 228 ± 11		1.1653 1.148 ± 0.049	
		Training Set 229 ± 11 225 ± 10		1.068 ± 0.047 1.026 ± 0.045	
Messung ⑩; Feature $\in [-1, 1]$ ohne RF Feature ohne Constraint 80 Iterationen $\eta = 0.1$	Feature $\in [-1, 1]$ ohne RF Feature ohne Constraint 7 Iterationen $\# \text{Perturbs} = 6$	Test Set 237.66 228.2 ± 9.9		1.165 1.179 ± 0.048	
		Training Set 229 ± 11 225 ± 11		1.068 ± 0.047 1.015 ± 0.046	
Messung ⑪; Feature $\in [0, 1]$ ohne RF Feature ohne Constraint 80 Iterationen $\eta = 0.1$	Feature $\in [0, 1]$ ohne RF Feature Constraint auf \mathbb{N}^4 4 Iterationen $\# \text{Perturbs} = 6$	Test Set 237 ± 11 246 ± 12		1.196 ± 0.049 1.196 ± 0.050	
		Training Set 229 ± 10 232 ± 11		1.083 ± 0.044 1.048 ± 0.044	
Messung ⑫; Feature $\in [0, 1]$ ohne RF Feature ohne Constraint 80 Iterationen $\eta = 0.1$	Feature $\in [0, 1]$ ohne RF Feature ohne Constraint 4 Iterationen $\# \text{Perturbs} = 6$	Test Set 237 ± 11 228 ± 10		1.196 ± 0.049 1.147 ± 0.049	
		Training Set 229 ± 10 223 ± 10		1.083 ± 0.044 1.034 ± 0.045	

Tabelle 3: Messwerttabelle 3

Teil I
APPENDIX

A

APPENDIX TEST

LITERATURVERZEICHNIS

- [1] Sebastian Nowozin, Christoph H. Lampert, *Structured Learning and Prediction in Computer Vision*, 2011.
- [2] 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
- [3] aroslav Ganin and Victor Lempitsky, *N⁴-Fields: Neural Network Nearest Neighbor Fields for Image Transforms*, Skolkovo Institute of Science and Technology, 2014
- [4] Piotr Dollár and C. Lawrence Zitnick, *Structured Forests for Fast Edge Detection*, ICCV 2013
- [5] Ullrich Köthe, *Vision with Generic Algorithms*, Image Processing and Analysis Library, Version 1.10.0
- [6] Thorsten Beier

DECLARATION

Put your declaration here.

Saarbrücken, September 2015

Jan Lammel