

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 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

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 | 4 |
| 2.4 | Loss Funktionen | 4 |
| 2.4.1 | Hamming Loss | 4 |
| 2.4.2 | Variation of Information | 4 |
| 2.5 | Structured Learning | 5 |
| 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 RESULTATE | 9 |
| 4.1 | Stochastic Gradient mit RF Feature | 9 |
| 4.2 | Stochastic Gradient ohne RF Feature | 17 |
| 4.3 | Kreuzvalidierung Messung | 10 |
| 5 | FAZIT | 23 |
| | LITERATURVERZEICHNIS | 25 |

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)

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

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 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

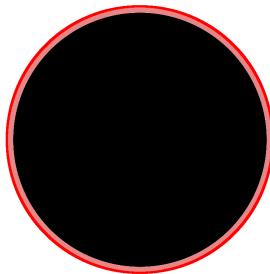
Bisher wird beim Hamming Loss jede einzelne Kante der Segmentierung überprüft, ob diese in der Ground Truth ebenso vorhanden ist

oder nicht. Bei einer leicht verschobenen Segmentierung (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.

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.

Bild 2 a) veranschaulicht dies nun: Der schwarze Bereich entspricht der zu segmentierenden Struktur und die rote Linie wäre die dazugehörige 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.

Bei b) ist ein Beispiel solch eines natürlichen Bildes dargestellt, wo es absolut nicht eindeutig ist, wo genau die Ground Truth jetzt zu zeichnen ist.



(a) Demonstration Loss Variation of Information



(b) Beispielbild mit unklarer Ground Truth

Abbildung 2: Beispielbild mit idealer Segmentierung, die es gilt zu erhalten

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.

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.

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

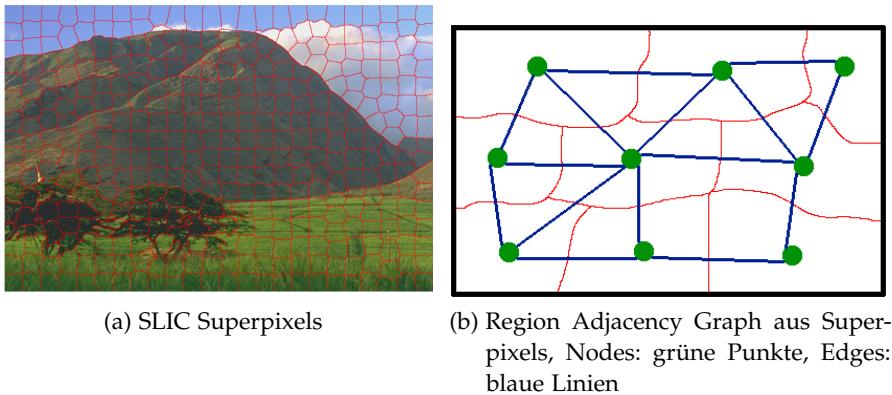


Abbildung 3: Beispiel einer Superpixel Partitionierung mittels SLIC

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

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. Ansprechend 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. Zu den in dieser Arbeit verwendeten Feature wird in [3.3](#) genauer eingegangen.

2.3 DAS MULTICUT PROBLEM

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

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

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. 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 diejenigen 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 Methode Variation of Information vorgestellt und mit der bestehenden des Hamming Loss verglichen.

2.4.1 Hamming Loss

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

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

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, als sie in mehreren Teilen vorzufinden.

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 optimal segmentieren zu können, muss der Parameter w bestimmt werden. "Optimal" bedeutet hier in Bezug auf eine Loss Funktion, die als Qualitätskriterium dient. Da ein niedriger Loss für eine gute Segmentierung 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 hinzugefügt 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 zunächst in #Per-turbs verschiedene normalverteilte Richtungen, vom momentanen Weight Vector aus, der Loss auf dem aktuellen Bild berechnet. An-schließend werden die einzelnen Richtungen nach ihrem Loss ge-wichtet, 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 auf dem gesamten Trainingsset ermittelt und mit dem bisher besten Wert verglichen. Bei Erreichen ei-nen neuen Tiefpunktes, wird direkt dorthin gesprungen, andernfalls wird die Schrittweite mit dem Niedrigsten erreichten Loss gewählt.

3

EXPERIMENTELLES SETUP

3.1 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 unser Trainings- als auch das Testset bestand schließlich 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).

3.2 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. Inferno [7] fand Anwendung beim Zusammenführen aller Daten, Lösen des Multicut Problems und Lernen der Parameter sowohl mit SubGradient bezüglich Hamming Loss, 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

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 [4] mit und ohne Gewichtung auf Länge der Edge
- Structured Forests Kantendetektor [5, Dollár et al.] mit und ohne Gewichtung auf Länge der Edge
- 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 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 Edge, zur Beseitigung des Bias im Feature Space

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

4

EXPERIMENTE UND RESULTATE

Die Experimente bestanden aus Messungen mit unterschiedlichen Konfigurationen, wobei jeweils zuerst der Hamming-Loss \mathcal{L}_{PH} 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.

4.1 STOCHASTIC GRADIENT MIT RF FEATURE

| 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 | Training Set 88.2 ± 6.6 | 1.159 ± 0.048 | 1.267 ± 0.052 |
| 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 | Training Set 88.2 ± 6.6 | 1.159 ± 0.048 | 1.401 ± 0.067 |
| 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 | Training Set 229 ± 11 | 1.165 ± 0.047 | 1.187 ± 0.049 |
| 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 | Training Set 229 ± 11 | 1.165 ± 0.047 | 1.163 ± 0.049 |

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 ± 19 | | 1.166 ± 0.047 1.446 ± 0.065 | |
| | | Training Set 82.3 ± 5.9 41.6 ± 7.5 | | 0.492 ± 0.034 0.256 ± 0.039 | |
| 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 ± 11 249 ± 12 | | 1.166 ± 0.047 1.285 ± 0.056 | |
| | | Training Set 82.3 ± 5.9 42.8 ± 4.4 | | 0.492 ± 0.034 0.258 ± 0.026 | |
| 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 1.328 ± 0.54 | |
| | | Training Set 229 ± 10 159.4 ± 9.4 | | 1.083 ± 0.044 0.861 ± 0.047 | |
| 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 ± 13 | | 1.197 ± 0.049 1.283 ± 0.059 | |
| | | Training Set 229 ± 10 17.9 ± 2.6 | | 1.083 ± 0.044 0.132 ± 0.020 | |

Tabelle 2: Messwerttabelle 2

Es wurde 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 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.

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 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 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 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. 4 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.

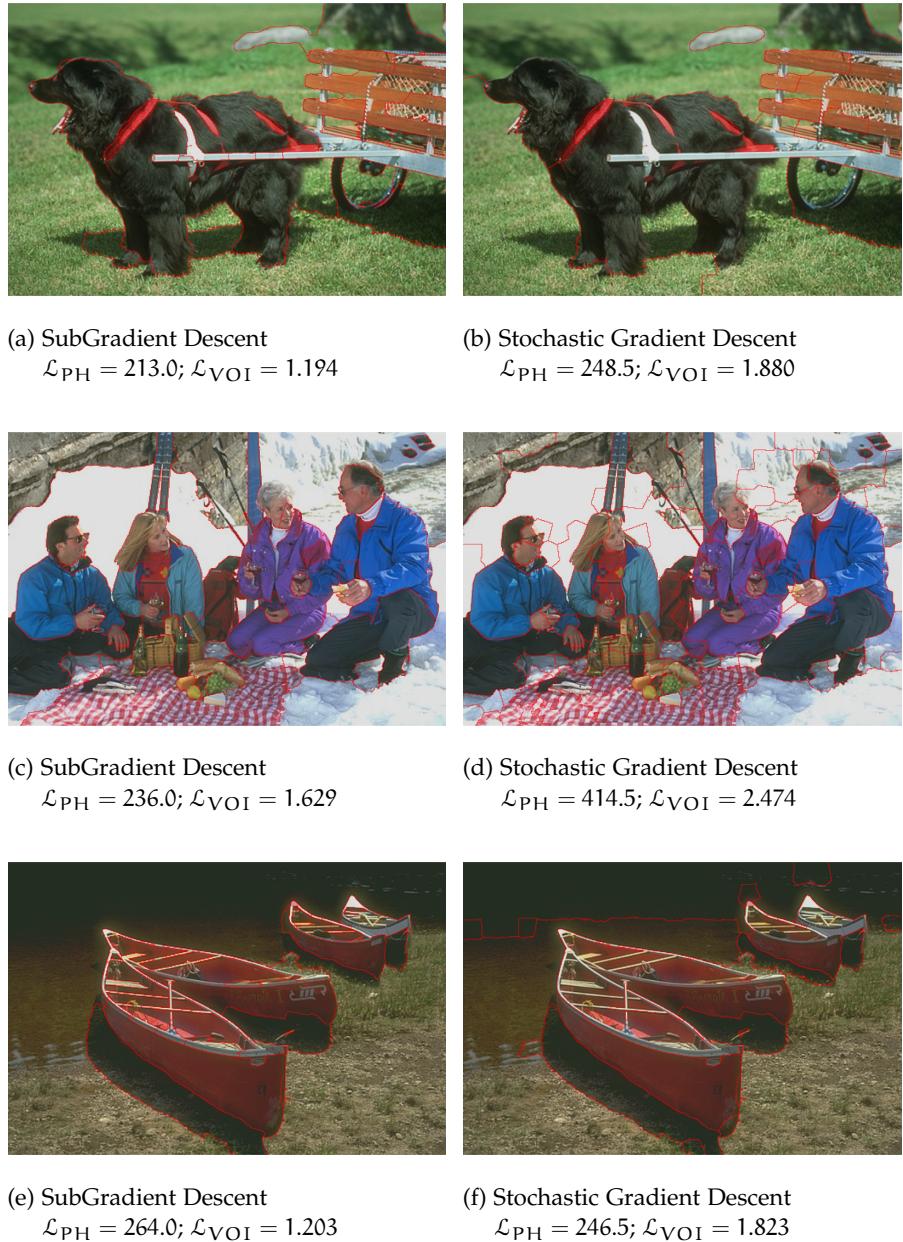


Abbildung 4: Vergleich der Segmentierungen nach Optimierung auf \mathcal{L}_{PH} mittels Subgradient Descent bzw. \mathcal{L}_{VOI} mittels Stochastic Gradient Descent; Resultate aus Messung 2 (a&b)/ Messung 8 (c)



Abbildung 5: Beispiel gutes/schlechtes Loss-Maß VOI

In Abb. 5 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 liegende Hauswand, beim Vogelbild der Stock und der fehlende Teil des

Vogels. Dieses Verhalten hat nun 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.

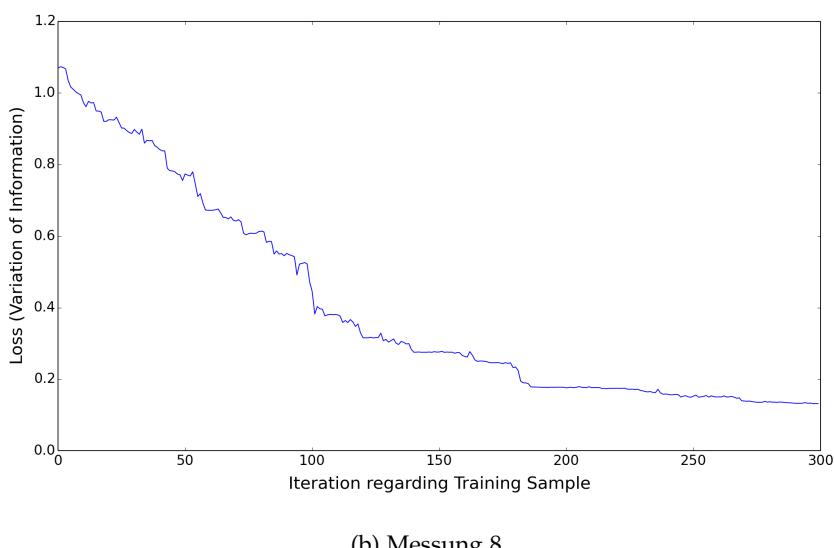
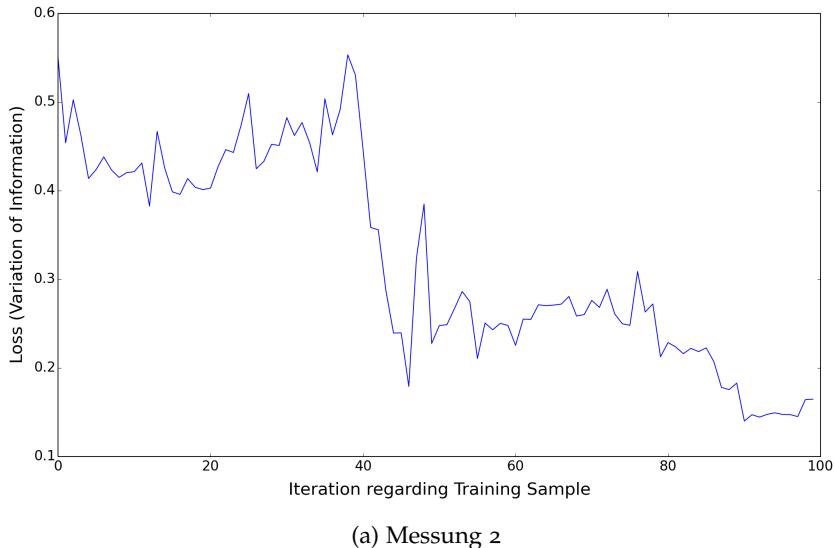
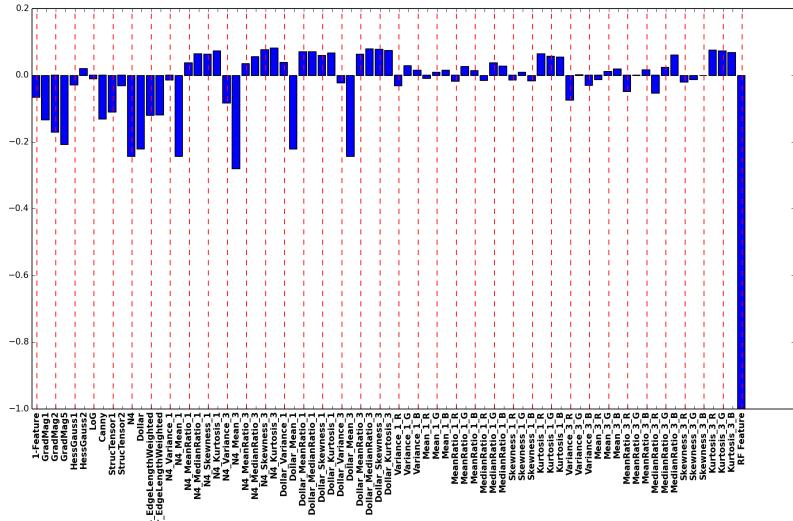
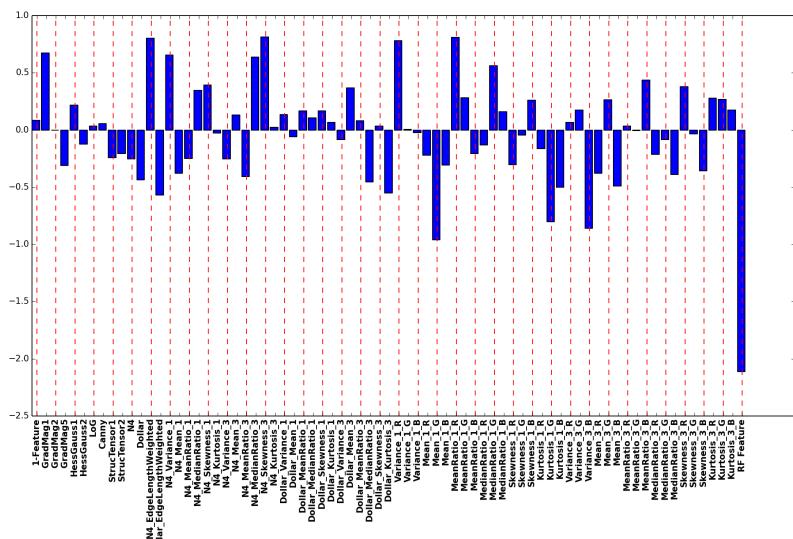


Abbildung 6: Average Loss Verlauf beim Optimieren der Weights bezüglich Variation of Information mithilfe von Stochastic Gradient Descent

In Abb. 6 ist nun 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 auf einen Niedrigen Loss zu gelangen, sieht man speziell hier bei b) sehr schön das konvergente Verhalten.



(a) Subgradient Descent, RF Feature dominiert stark



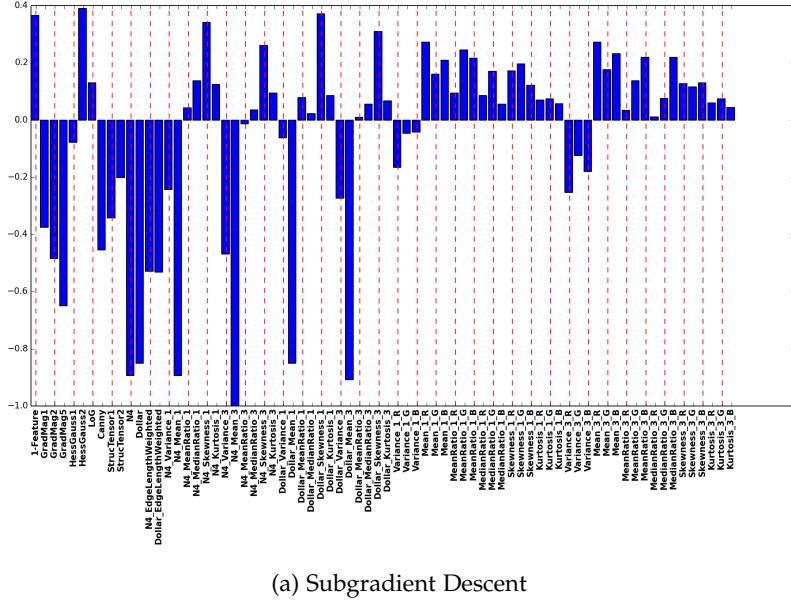
(b) Stochastic Gradient Descent

Abbildung 7: Resultierende Weights nach Optimierung auf Hamming Loss
 (a) und Variation of Information (b), Messung 2

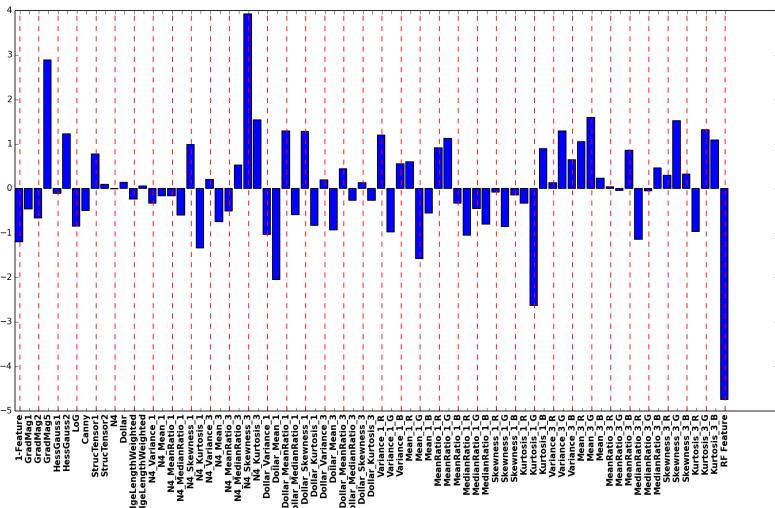
Betrachtet man nun die Weights nach den jeweiligen Lernprozessen fällt bei Subgradient Descent mit RF Feature (Abb. 7) dessen Dominanz auf. Die anderen Feature wirken nur als kleine Korrekturen. Die anschließend durch Stochastic Gradient aktualisierten Weights weisen diese Dominanz nicht mehr auf. Dies 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. 8 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 8: Resultierende Weights nach Optimierung auf Hamming Loss
 (a) und Variation of Information (b), Messung 8

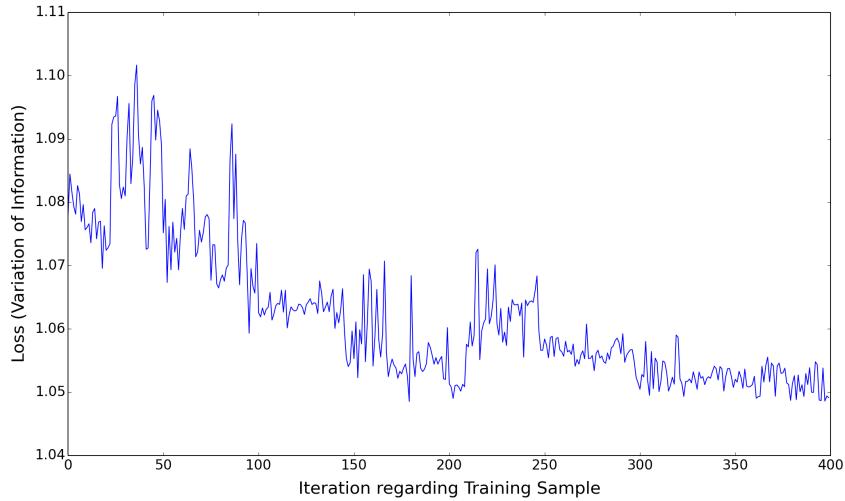
4.2 STOCHASTIC GRADIENT OHNE RF FEATURE

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.

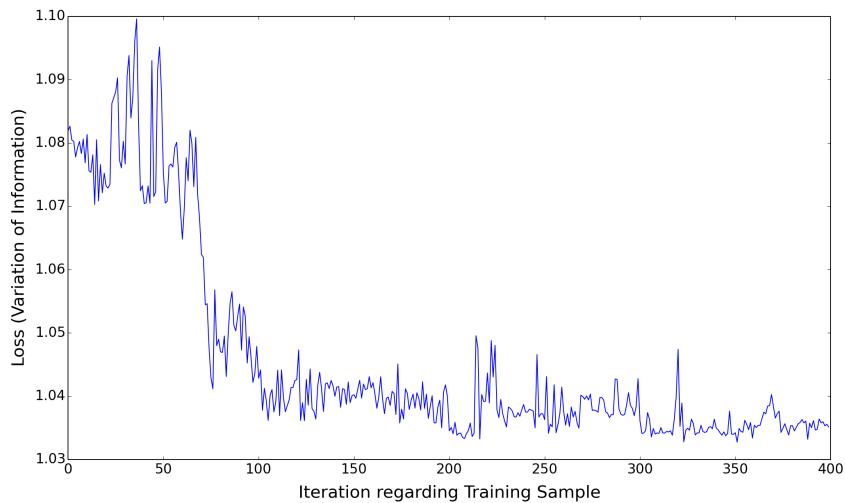
| SubGrad | StochGrad | \mathcal{L}_{PH} | | \mathcal{L}_{VOI} | |
|---|---|-------------------------------------|-------------------------------------|--|--|
| | | SubGrad | StochGrad | SubGrad | StochGrad |
| Messung ⑨; ohne RF Feature ohne Constraint 80 Iterationen $\eta = 0.1$ | Feature $\in [-1, 1]$ 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 $\#Perturbs = 6$ | Training Set 229 ± 11 | Training Set 225 ± 10 | Training Set 1.068 ± 0.047 | Training Set 1.026 ± 0.045 |
| 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 238 ± 10 | Test Set 228.2 ± 9.9 | Test Set 1.165 ± 0.047 | Test Set 1.179 ± 0.048 |
| Messung ⑫; ohne RF Feature ohne Constraint 80 Iterationen $\eta = 0.1$ | Feature $\in [0, 1]$ ohne RF Feature ohne Constraint 4 Iterationen $\#Perturbs = 6$ | Training Set 229 ± 11 | Training Set 225 ± 11 | Training Set 1.068 ± 0.047 | Training Set 1.015 ± 0.046 |
| Messung ⑬; 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 246 ± 12 | Test Set 1.196 ± 0.049 | Test Set 1.196 ± 0.050 |
| Messung ⑭; ohne RF Feature ohne Constraint 80 Iterationen $\eta = 0.1$ | Feature $\in [0, 1]$ ohne RF Feature ohne Constraint 4 Iterationen $\#Perturbs = 6$ | Training Set 229 ± 10 | Training Set 232 ± 11 | Training Set 1.083 ± 0.044 | Training Set 1.048 ± 0.044 |
| Messung ⑮; 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 |
| Messung ⑯; ohne RF Feature ohne Constraint 80 Iterationen $\eta = 0.1$ | Feature $\in [0, 1]$ ohne RF Feature ohne Constraint 4 Iterationen $\#Perturbs = 6$ | Training Set 229 ± 10 | Training Set 223 ± 10 | Training Set 1.083 ± 0.044 | Training Set 1.034 ± 0.045 |

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

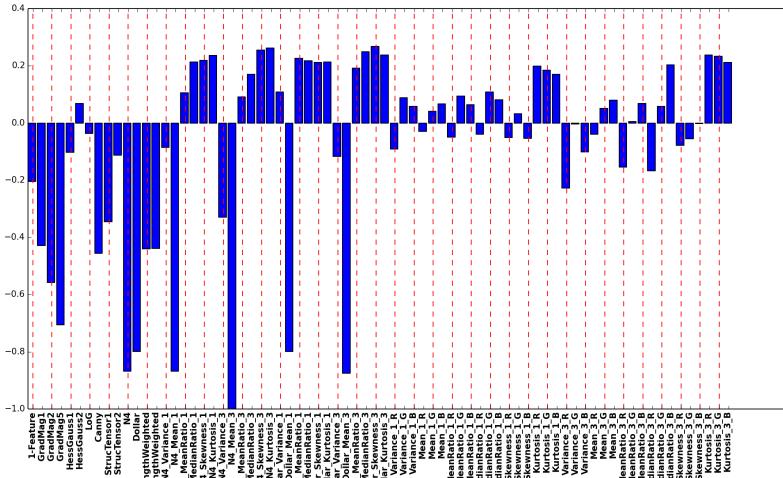
Abbildung 9: Average Loss Verlauf beim Optimieren der Weights bezüglich Variation of Information mithilfe von Stochastic Gradient Descent

In Abb. 9 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

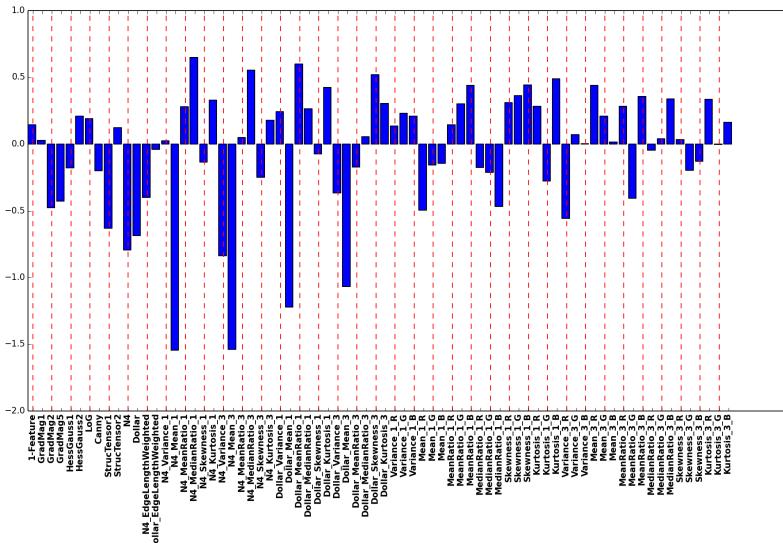
Verhalten erkennbar, was zur Annahme führt, dass das globale Minimum des Loss in etwa erreicht wurde. Der starke Abfall bei Messung 12 Iteration 80 bietet zwei Interpretationsmöglichkeiten über die Struktur des Loss-Raums. Entweder es existiert ein räumlich eng begrenztes Minimum, in das der Stochastic 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 so ähnlich sind. Dies lässt sich allerdings durch die Vorauswahl der Edges 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 nun 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 10: Resultierende Weights nach Optimierung auf Hamming Loss
 (a) und Variation of Information (b), Messung 10

Bemerkenswert ist die starke Veränderung der Weights beim Stochastic Gradient (siehe Abb. 10), trotz der Tatsache dass der Loss auf dem Trainingsset nicht sonderlich gesenkt werden konnte. Dies spricht, zusammen mit Abb. 7 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.

4.3 KREUZVALIDIERUNG MESSUNG 10

Da wegen der großen Fehlerbereiche keine Aussage möglich war, ob es nun 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 gleiche große Abschnitte aufgeteilt, so dass das jeweilige Trainingsset immer aus 160 und das Testset aus 40 Bildern bestand.

| # | Hamming-Loss | | | VOI-Loss | | |
|---|--------------|-----------|---|---------------------|-----------|---|
| | 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 ± 0.005 | | | |
| | | | | | | |
| | | | | 1.0025 ± 0.0084 | | |

Tabelle 4: Messwerttabelle Testset Kreuzvalidierung

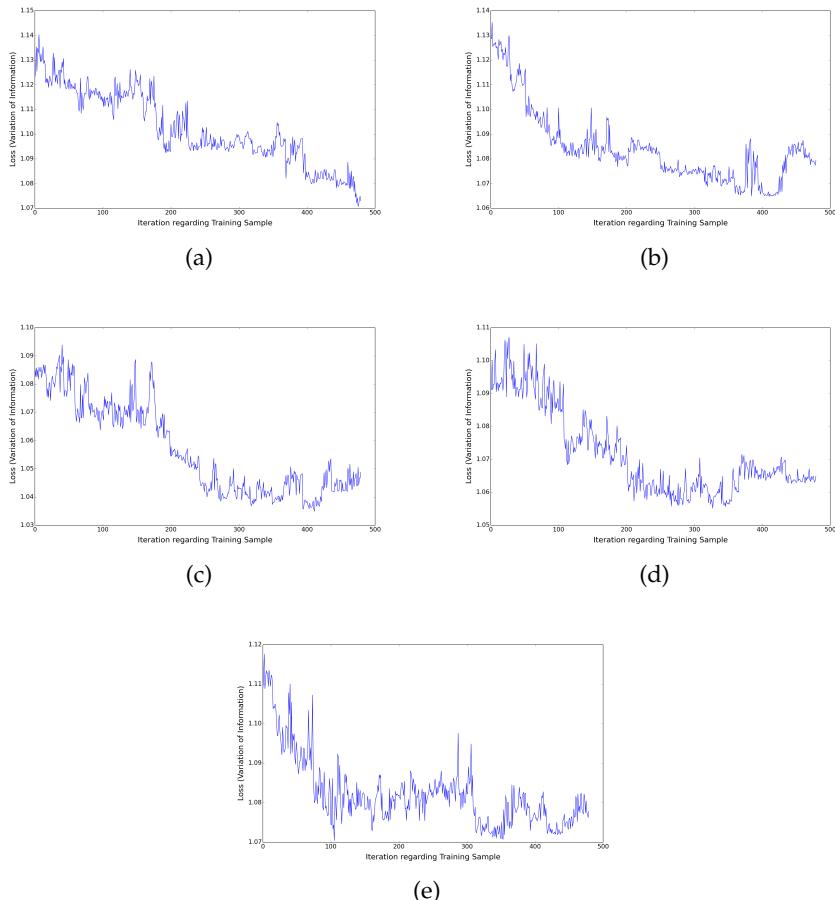


Abbildung 11: Verlauf des Loss beim Stochastic Gradient Descent
Aufgrund des eher kleinen Testsets variiert der erreichte Loss von Messung zu Messung recht stark. Daher wurde das Verhältnis der Änderung individuell für jede Messung berechnet und daraus der Mittelwert bzw. der Fehler des Mittelwerts genommen. 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. 11 sind noch die Trends des Loss beim Stochastic Gradient Descent zu sehen. Abgesehen von a) 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 Nutzung des Variation of Information, statt des Hamming Loss.

5

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 aufgetrennen minimalen Verbesserungen auf dem Testset nun zufällig waren oder nicht.

Anhand einzelner Beispiele wurde die anschauliche Bedeutung vom Variation of Information Loss verdeutlicht und konnte so auf seine Vor- und Nachteile eingegangen werden.

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

DECLARATION

Put your declaration here.

Saarbrücken, September 2015

Jan Lammel