Word Assignment to Image Parts

Armin Fritsche

Betreuer: Daniel Keysers

Inhalt

1 Einleitung **2**

2 Objekterkennung und Wortvorhersage **2**
2.1 Der EM-Algorithmus . . . 2
2.2 Kontrolle des Vokabulars . . . 5
2.2.1 Streichung nicht verwendeter Wörter . . . 6
2.2.2 Einführung eines null-Schwellwertes . . . 6
2.2.3 Clusterbildung . . . 7
2.3 Hinzunahme von Supervised Data . . . 7

3 Einfluss der Merkmale **9**
3.1 Messung der Performance . . . 9
3.2 Wahl der Merkmale . . . 10
3.3 Auswertung der Merkmale . . . 11

4 Einfluss des Segmenters **12**
4.1 Normalized Cuts . . . 13
4.2 Segmenter und Wortvorhersage . . . 14

5 Zusammenfassung **16**

Zusammenfassung

Um leichten Zugriff auf ein Bild zu bekommen werden Bilder oft mit deren Inhalt beschriftet. Allerdings besteht bei diesen Wörtern meist keine Verknüpfung mit einzelnen Teilen des Bildes. Diese Ausarbeitung beschäftigt sich damit, dass diese Wörter auch noch mit bestimmten Regionen im Bild assoziiert werden und somit eine Beschriftung für einzelne Bildregionen gemacht werden kann. Damit lassen sich dann auch neue Bilder automatisch beschriften. Dazu wird zunächst ein stochastisches Modell aufgestellt, welches den Regionen bestimmte Wörter zuweist. Des weiteren wird untersucht, wie man dieses Verfahren verfeinern kann. Außerdem wird betrachtet welche Seiteneffekte auf das Verfahren einwirken und wie diese zur Verbesserung der Wortvorhersage benutzt werden können.

1 Einleitung

Bei mehreren Anwendungen wie z.B. Image Retrieval will man inhaltsbasierten Zugriff auf eine Bilddatenbank zu haben. Zu diesem Zweck sollen die Bilder mit einer dem Inhalt entsprechenden Beschriftung versehen werden. Häufig ist es nun so, dass man bei einer gegebener Bildermenge schon eine grobe Inhaltsangabe des Bildes findet. So stehen zum Beispiel im Internet schon solche Bildarchive zur Verfügung (z.B. `http://www.archive.org`). Auf der anderen Seite gibt es bei vielen Anwendungen schon eine gewisse Wort-Vorauswahl. In der Medizin etwa weiß der Radiologe im Allgemeinen was geröngt wird. Diese Wortvorauswahl, beziehungsweise Wörter für das komplette Bild werden als Vokabular bezeichnet. Diese Beschriftung kann noch insofern verbessert werden, dass die Wörter aus dem Vokabular zwar mit den Bildern verknüpft sind, aber noch nicht einzelnen Bildteilen zugewiesen wurden. Des weiteren sollen neue Bilder auch automatisch beschriftet werden, ohne dass vorher eine Wortauswahl getroffen wird.
Aus diesem Grunde beschäftigt sich diese Ausarbeitung mit der Beschriftung von einzelnen Bildteilen durch gegebene Terme. Die einzelnen Bildteile sind dabei schon durch einen Segmentierungsalgorithmus erzeugt worden. Die Beschriftungen der Segmente werden direkt im Bild vorgenommen. Das Verfahren läßt nach dem einem Training auf Bildern, mit Wortvorauswahl, auch die Beschriftung von Bildern ohne Wortvorauswahl zu. Abbildung 1 zeigt ein Beispiel dessen, was erreicht werden soll. Die Grundlage dieser Ausarbeitung sind dabei haupsächlich die Arbeiten [1] und [2] von K. Barnard, P. Duygulu, J. de Freitas, R. Guru und D. Forsyth. Diese Arbeiten nehmen als Bilderdatenmenge das Corel Data Set, welches verschiedene Bilder unterschiedlichster Inhalte hat. Aus diesem Grund befasst sich diese Ausarbeitung zunächst auch nicht speziell mit medizinischen Bildern.
Die besten Ergebnisse werden natürlich durch manuelle Beschriftung durch einen Experten gemacht. Dies ist aber aufgrund der großen Datenmengen nicht immer möglich. Deshalb wird Objekterkennung und Termzuweisung automatisiert. Diese Ausarbeitung wird zunächt einen Algorithmus zur Wortvorhersage für Bildsegmente präsentieren. Anschließend werden verschiedene Möglichkeiten untersucht, um mit dem Algorithmus bessere Ergebnisse zu erreichen. Des weiteren stellt sich die Frage, wie die Regionen in einem Bild repräsentiert werden und was eine „gute“ Segmentierung eines Bildes ist. Auch diese Fragen werden in dieser Ausarbeitung behandelt.

2 Objekterkennung und Wortvorhersage

Nachdem ein Segmenter ein Bild in verschiedene Teile zerlegt hat, wird eine adäquate Methode benötigt, mit welcher auf diese Segmente zugegriffen werden kann. Zu diesem Zweck wird jede Region einer Vektorquantisierung unterzogen. D.h. jede Region wird mit ihren Merkmalen wie z.B. ihrer Form oder ihrer Farbe identifiziert. In Abschnitt 3 wird noch genauer betrachtet, wie die Wahl dieser Merkmale in die Performance der Wortvorhersage einfließen und wie diese Performance gemessen werden kann. Ein Merkmalsvektor eines Segments wird als „Blob“ bezeichnet. Bevor aber die Erstellung dieser Blobs betrachtet wird, sollte man sich zunächst mit dem Problem der Wortvorhersage für Bildregionen beschäftigen.

2.1 Der EM-Algorithmus

Um Blobs mit Wörtern zu beschriften, wird hier mit einem stochastischen Modell gearbeitet. Ob ein Blob b mit einem Wort w beschriftet wird, entscheidet sich also über die Wahrscheinlichkeit $p(w|b)$. Die

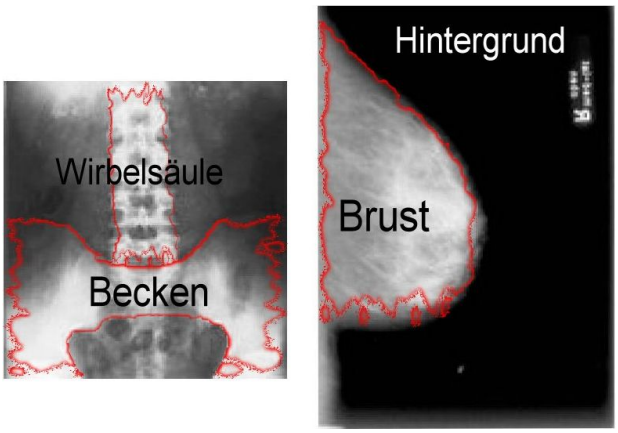

Abbildung 1: Hier werden zwei medizinische Bilder gezeigt, welche jeweils segmentiert und beschriftet wurden. Die Bilder wurden in diesem Beispiel von einem Menschen „per Hand“ segmentiert und beschriftet. Ziel ist jedoch die Segmentierung und Beschriftung solcher Bilder automatisch vornehmen zu lassen.

N : Anzahl der Bilder in der Trainingsmenge
M_n : Anzahl der Wörter im n-ten Bild — L_n : Anzahl der Blobs im n-ten Bild
w_{nj} : j-tes Wort im n-ten Bild — b_{ni} :i-ter Blobs im n-ten Bild

$p(w = w_{nj}|b = b_{ni})$: Wahrscheinlichkeit, dass $w = w_{nj}$ falls b und b_{ni} im selben Cluster liegen
$p(a_{nj} = i)$: Wahrscheinlichkeit, dass das j-te Wort im n-ten Bild dem Blob i zugeordnet wird
$p_{l,m}(a_j = i)$: Wahrscheinlichkeit, dass in einem Bild mit m Wörtern und l Blobs das j-te Wort dem n-ten Blob zugewiesen wird

$p(w_{nj}|b_{ni}) := p(w = w_{nj}|b = b_{ni})$

Abbildung 2: Notation

Beschriftung wird dann schließlich mit dem Wort w_{max} vorgenommen, welches die höchste Wahrscheinlichkeit $p(w_{max})$ aufweist. Demnach ist

$$w_{max} = \arg\max_{w} p(w|b) \tag{1}$$

Die Bedingung über die Blobs b macht noch Probleme, da die Blobs zunächst durch einen reellwertigen Merkmalsvektor repräsentiert werden. Deshalb wird der Merkmalsraum in Blobcluster mit dem k-means Verfahren unterteilt. Hierfür werden zunächst k zufällige Punkte im Vektorraum über die Merkmale gewählt. Diese Punkte definieren den Mittelwert ihrer Cluster. Dann werden alle Blobs per nearest-neighbour Suche einem Cluster zugeordnet und danach der Mittelpunkt der Cluster neu bestimmt. Die Iteration dieses Verfahrens konvergiert gegen eine Partitionierung des Vektorraums, welch die Cluster der Blobs darstellt.
Um nun das Wort w_{max} zu erhalten, wird eine Wahrscheinlichkeitstabelle über alle $p(w|b)$ aufgestellt. Da die Notation in [1] teilweise unklar ist, wird sie hier angepasst und in Abbildung 2 gezeigt. Um nun damit die Tabelle über die Wahrscheilnlichkeiten $p(w|b)$ aufzustellen wird ein Modell von Brown *et al.* [3] angewendet. Dieses Modell stammt ursprünglich aus der maschinellen Übersetzung zweier verschiedensprachiger Texte. Hier wird die „Wort-zu-Wort-Übersetzung“ in eine „Blob-zu-Wort-Übersetzung“ überführt. Das Training der Übersetzungswahrscheinlichkeit beruht auf der Maximierung der Likelihood auf den Trainingsdaten:

$$\prod_{n=1}^{N} \prod_{j=1}^{M_n} \sum_{i=1}^{L_n} p(a_{nj} = i) p(w = w_{nj}|b = b_{ni}) \tag{2}$$

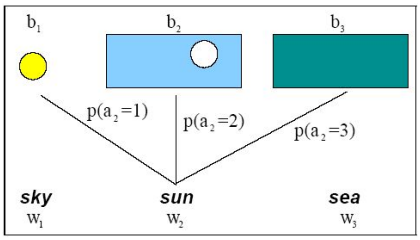

Abbildung 3: Ein Beispiel aus [1] für die Notation der Wahrscheinlichkeiten $p(a_j = i)$. Unter der Annahme, dass dieses Bild das erste Bild der Trainingsmenge ist, wird die Wahrscheinlichkeit, dass der linke Blob mit dem Wort „sun“ assoziert wird als $p(a_{1,2} = 1)$ notiert.

Dabei ist $p(a_{nj} = i)$ die Wahrscheinlichkeit, dass das j-te Wort im n-ten Bild dem i-ten Blob im n-ten Bild zugeordnet wird. Der Wert $p(w = w_{nj}|b = b_{ni})$ entspricht der Wahrscheinlichkeit, dass $w = w_{nj}$ unter der Bedingung, dass b im selben Cluster liegt, wie b_{ni}. Um die Notation zu erleichtern, wird im Folgenden $p(w_{nj}|b_{ni})$ für $p(w = w_{nj}|b = b_{ni})$ verwendet. Der Wert $p(a_{nj} = i)$ ist die Wahrscheinlichkeit, dass das j-te Wort im n-ten Bild dem i-ten Blob im n-ten Bild zugeordnet wird. Ein Beispiel dieser Wahrscheinlichkeit ist in Abbildung 3 gegeben. Auf die Herleitung von (2) wird hier nicht weiter eingegangen, da die Formel aus der Maschinenübersetzung stammt und sich diese Ausarbeitung auf die Bildverarbeitungskomponente konzentriert. Um eine Beschriftung der Regionen zu berechnen, werden also die Wahrscheinlichkeiten $p(w_{nj}|b_{ni})$ und $p(a_{nj} = i)$ gebraucht. Jeder dieser Werte ist für sich genommen aus dem anderen zu berechnen. Das Problem besteht darin, dass für die Berechnung des einen Wertes jeweils der andere Wert benötigt wird. Ein solches Problem wird in der Literatur als „Missing Data Problem“ bezeichnet. Dafür hat bereits 1977 Dempster *et al.* den EM-Algorithmus als Lösung [4, 6] angeboten. Hierfür wird der zunächt eine Näherung für $p(a_{nj} = i)$ aufgestellt und damit der Wert $p(w_{nj}|b_{ni})$ berechnet. Hiermit kann man wiederum eine bessere Näherung für $p(a_{nj} = i)$ bestimmen. Diese beiden Schritte nennen sich Expectation- und Maximization-Schritt. Iteriert konvergieren sie gegen ein lokales Maximum der Zielfunktion (2). Der Beweis der Konvergenz wird hier mit Verweis auf [4] ausgelassen. Um den EM-Algorithmus für die Wortzuweisung zu formalisieren, wird als Zwischenergebniss der Wert $p_{l,m}(a_j = i)$ gebraucht. Dieser Wert ist die Wahrscheinlichkeit, dass in einem Bild mit m Wörtern und l Blobs das j-te Wort dem n-ten Blob zugewiesen wird. Für die Berechnung von $p_{l,m}(a_j = i)$ im Maximization-Schritt wurden von Duygulu *et al.* alle Bilder n mit der selben Anzahl von Wörtern und selber Anzahl von Blobs betrachtet. Dann berechnet sich $p_{l,m}(a_j = i)$ als

$$p_{l,m}(a_j = i) = \frac{1}{N_{l,m}} \sum_{n}^{N_{l,m}} p(a_{nj} = i) \tag{3}$$

wobei $N_{l,m}$ die Anzahl aller Wörter ist mit l Blobs und m Wörtern ist. Das heißt, dass die bildunabhängige Wahrscheinlichkeit einer Beschriftung gleich dem arithmetischen Mittel aller Beschriftungswahrscheinlichkeiten aller Bilder ist.
Als letzter Wert braucht nun nur noch $p(w_{nj}|b_{ni})$ berechnet zu werden. Dafür werden alle Paare (b, w) von Blobs und Wörtern betrachtet, die in mindestens einem Bild zusammen auftreten. Eine andere Zuordnung würde keinen Sinn machen, da sonst jegliche Informationen über den Zusammenhang von Regionen und Wörtern unbeachtet blieben. Dafür wird die Indikatorfunktion $\delta_{w,b}(w_{nj}, b_{ni})$ eingeführt, welche genau dann gleich 1 ist, wenn w und b im Bild n an der Stelle i bzw. j auftreten und sonst den Wert 0 annimmt. Dann berechnen Duygulu *et al.* $p(w|b)$ durch

$$\tilde{p}(w|b) = \sum_{n=1}^{N} \sum_{j=1}^{M_n} \sum_{i=1}^{L_n} p(a_{nj} = i)\delta_{w,b}(w_{nj}, b_{ni}) \tag{4}$$

und normieren anschließend. Nach der erfolgten Berechnung von $p_{l,m}(a_j = i)$ aus (3) und $t(w|b)$ aus (4) kann nun wieder in dem Expectation-Schritt gegangen werden und eine bessere Näherung für $p(a_{nj} = i)$ berechnet werden. Das komplette Verfahren ist noch einmal zur besseren Übersicht in Abbildung 4 zusammengefasst.
Mit dem EM-Algorithmus werden die Werte $p(a_j = i)$, $p(w|b)$ und $p(a_{nj} = i)$ schrittweise trainiert. Die Beschriftung eines beliebigen Blobs b wird dann mit dem Wort vorgenommen, welches die höchste

Initialisierung
E Step:

1. Für alle $n = 1 \dots N$, $j = 1 \dots M_n$ und $i = 1 \dots L_n$ berechne:

$$\tilde{p}(a_{nj} = i) = p_{l,m}(a_j = i) p(w_{nj}|b_{ni}) \tag{5}$$

2. Normalisieren von $\tilde{p}(a_{nj} = i)$ für alle Bilder n und Wörter j

$$p(a_{nj} = i) = \frac{\tilde{p}(a_{nj} = i)}{\sum_{i'=1}^{L_n} \tilde{p}(a_{nj} = i')} \tag{6}$$

M Step:

1. Berechne die gemeinsame Wahrscheinlichkeit für alle j und alle Bilder n der selben Größe (d.h. Anzahl von Wörtern = m und Anzahl von Blobs = l)

$$p_{l,m}(a_j = i) = \frac{1}{N_{l,m}} \sum_{n}^{N_{l,m}} p(a_{nj} = i) \tag{7}$$

   mit $N_{l,m}$ ist die Anzahl der Bilder mit m Wörtern und l Blobs

2. Berechne für jedes Paar (b, w), die zusammen in mindestens einem Bild auftreten

$$\tilde{p}(w|b) = \sum_{n=1}^{N} \sum_{j=1}^{M_n} \sum_{i=1}^{L_n} p(a_{nj} = i) \delta_{w,b}(w_{nj}, b_{ni}) \tag{8}$$

   mit $\delta_{w,b}(w_{nj}, b_{ni})$ ist gleich 1 wenn b und w im selben Bild auftreten und sonst 0.

3. Normalisiere $\tilde{p}(w|b)$ und erhalte $p(w|b)$

Iteriere E Step und M Step bis Konvergenz auftritt

Abbildung 4: Der EM (Expectation Maximization) Algorithmus

Wahrscheinlichkeit $p(w|b)$ aufweist. In den folgenden Abschnitten soll nun betrachtet werden, als wie gut sich das hier vorgestellte Verfahren erweist bzw. wie man es noch weiter verbessern kann.

2.2 Kontrolle des Vokabulars

Mit dem EM-Algorithmus ist man jetzt in der Lage eine Wahrscheinlichkeitstabelle aufzustellen, und mit ihrer Hilfe genau das Wort auszuwählen, das die größte Wahrscheinlichkeit für einen gegebenen Blob hat. So ist schon jetzt eine Wortvorhersage für einzelne Bildregionen möglich. Betrachtet man aber größere Bildmengen, so stellt man fest, dass das Ergebnis noch nicht zufriedenstellend ist. Es existieren einige wenige „gute“ Wörter, die nur sehr wenig vorhergesagt werden. Aber wenn ein Blob mit ihnen beschriftet wird, so sind diese Beschriftungen meist richtig. Bei den meisten Wörtern gibt es jedoch den Zusammenhang, dass häufiger mit ihnen richtig beschriftet wurde, desto öfter sie für einen Blob vorhergesagt wurden. Diese beiden Werte (richtige und häufige Vorhersagen) werden durch die Bewertungskriterien „Präzision“ und „Recall“ beschrieben. Die Präzision eines Wortes errechnet sich durch die Anzahl der korrekten Beschriftungen mit diesem Wort geteilt durch die Gesamtanzahl der Beschriftungen mit diesem Wort. Damit gibt die Präzision an, wie gut ein Wort bezüglich falsch-positiver Vorhersagen ist. Der Recall eines Wortes ist bestimmt durch die Anzahl der korrekten Beschriftungen mit diesem Wort geteilt durch die Anzahl der Beschriftungen, die mit dem Wort richtig wären. Somit kann der Recall eines Wortes als die relative Häufigkeit, mit der das Wort vorhergesagt wurde, betrachtet werden. Ein Ideales Verfahren würde solche Beschriftungen liefern, bei denen der Recall und die Präzision jedes Wortes gleich eins sind. Untersucht man eine durch den EM-Algorithmus beschriftete Datenmenge mit der Präzision und dem Recall, so kommt man zu der Beobachtung, dass es wenige Wörter mit einer hohen Präzision von etwa

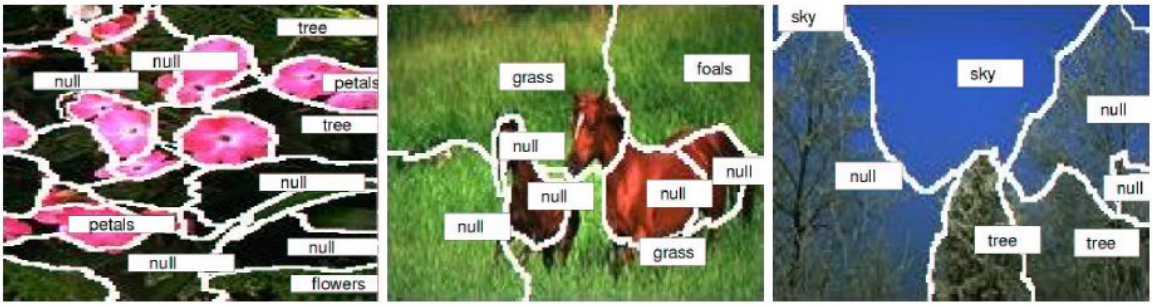

Abbildung 5: Durch den null-Schwellwert werden die schwierigen Regionen nur noch mit „null“ beschriftet. Wählt man den Schwellwert zu hoch (mitte) , so gibt es keine bzw. kaum noch verwertbare Beschriftungen. Die Bilder stammen aus [1].

$0.6 - 1.0$ gibt. Diese Wörter machen nur ca. $2 - 6\%$ des Vokabulars aus. Der Recall dieser Wörter ist aber eher niedrig. Für alle anderen Wörter macht man die Beobachtung, dass die Präzision zwischen 0.0 und 0.4 schwankt und umso höher ist, je höher auch der Recall ist. Diesen Zusammenhang kann man nutzen und die Wortvorhersage verbessern, indem man möglichst viele Wörter erreicht, die einen möglichst hohen Recall haben. Den Ergebnissen, die hier vorgestellt werdenn, liegt der Versuch von Duygulu *et al.* in [1] zugrunde.

2.2.1 Streichung nicht verwendeter Wörter

Es ist natürlich, dass nicht alle gegebenen Worte vorhergesagt werden, da zum Beispiel ein bestimmtes Wort zu keinem Blob die höchste Wahrscheinlichkeit aufweist. Streicht man diese Wörter aus dem Vokabular und macht anschließend ein Retraining durch nochmaliges Anwenden des EM-Algorithmus, so ist man in der Lage die Vorhersageergebnisse zu verbessern. Im Versuch aus [1] wurde allein durch diese Methode das Vokabular erheblich (fast 80%) vermindert. Nach einem Retraining mit dem EM-Algorithmus hat sich die Anzahl der guten Wörter nicht signifikant geändert. Für alle anderen Wörter (also die Mehrzahl) ist die Präzision aber leicht gestiegen.

2.2.2 Einführung eines null-Schwellwertes

Jetzt können durch den EM-Algorithmus und Retraining nach der Beschneidung des Vokabular schon teilweise brauchbare Vorhersagen für Bildregionen getroffen werden. Allerdings sind noch nicht alle Vorhersagen zufriedenstellend. Daher wird das Vokabular noch weiter für die Bedürfnisse der Wortvorhersage angepasst. Und zwar sollen nun die „guten“ Vorhersagen von den „schlechten“ Vorhersagen unterschieden werden. Um die schlechten Vorhersagen zu eliminieren wird zunächst ein Schwellwert eingeführt, der die mindeste Wahrscheinlichkeit darstellt, die als gut genug betrachtet wird. Das heißt, wenn es eine potentielle Vorhersage für einen Blob b mit mit dem Wort w gibt, für die nicht gilt

$$p(w|b) > Schwellwert$$

so wird diese Vorhersage gar nicht erst getroffen. Die entsprechenden Blobs werden stattdessen mit „null“ beschriftet. Mit steigendem Schwellwert werden natürlich immer weniger Wörter tatsächlich vorhergesagt und somit immer weniger Blobs beschriftet. Ab einer gewissen Höhe führt das dazu, dass gar kein Wort mehr vorhergesagt wird und alle Blobs mit „null“ beschriftet werden. Der Schwellwert kann daher nicht pauschal bestimmt werden, sondern muss unter Betrachtung der Präzision und des Recalls experimentell bestimmt werden. Als guter Wert für das Corel Data Set hat sich ein Wert zwischen 0.1 und 0.2 erwiesen. Abbildung 5 zeigt Beispiele für null-Beschriftungen. Alle Wörter, die nach der Einführung des Schwellwertes nicht mehr vorhergesagt werden können, werden nun auch noch aus dem Vokabular gestrichen. Da mit steigendem null-Schwellwert immer mehr Wörter mit null beschriftet werden, sinkt der Recall. Aber da grade die schlechten Beschriftungen wegfallen, steigt die Anzahl der verlässlichen Wörter. Das heißt es gibt nun mehr Wörter mit hoher Präzision, was angestrebt wird.

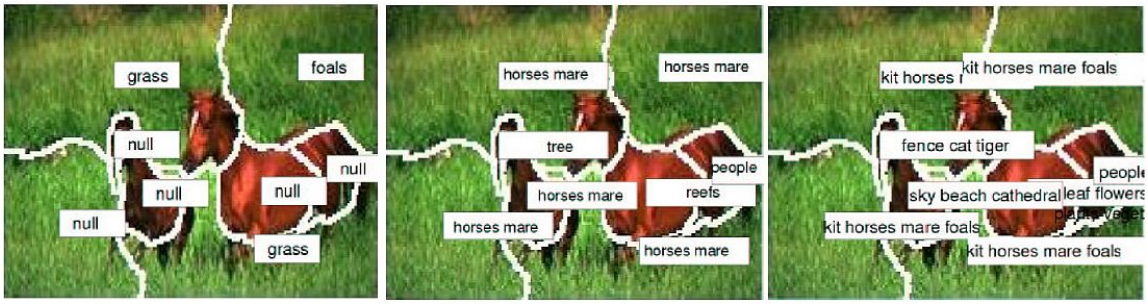

Abbildung 6: Die Bilder aus [1] zeigen die Beschriftungen, die nach verschiedenen Clusteriterationen entstanden sind. Im linken Bild ist noch keine Wortclusterung vorhanden. In mittleren Bild wurde das Wortcluster-Verfahren einmal und im rechten Bild zweimal angewendet. Da in den Bildern mit Pferden auch fast immer Gras vorhanden ist, hat das Lernverfahren Probleme bei der Zuweisung dieser Wörter und es passiert häufig, dass Regionen mit Gras das Wort (bzw. Cluster) „horse“ zugewiesen wird.

2.2.3 Clusterbildung

Trotz des inzwischen akzeptablen Ergebnisses wird unsere Wortvorhersage weiter verfeinert. Dazu nimmt man den ganz intuitiven Ansatz, dass einige Wörter sich sehr ähnlich sind. Nimmt man zum Beispiel die Wörter „horse“, „mare“ und „foal“ oder auch „train“ und „locomotive“, so werden unter Umständen schon zwei verschiedene menschliche Betrachter zu unterschiedlichen Ergebnissen kommen. Solche Wörter mit starken semantischen oder visuellen Zusammenhang nennen wir „ununterscheidbar“. Dazu zählen aber auch nicht so offensichtliche Wörter wie „jet“ und „eagle“, weil beide ein dunkler Klecks auf blauen Hintergrund sind und sie damit nicht mit den hier vorgestellten Mitteln unterschieden werden können. Wie man es schaffen kann, möglichst wenig ununterscheidbare Wörter zu bekommen wird in den Abschnitten 3 und 4 beschrieben. Die ununterscheidbaren Wörter werden nun zu Clustern zusammengefasst. Die Cluster werden vor der Wortvorhersage so definiert, dass sie solche Worte wie z.B. „horse“ und „mare“ zusammenfassen. Um dies zu erreichen, sollen alle Wörter, die sich in einem gewissen Maße ähneln, in einem Cluster zusammengefasst werden. Zu diesem Zweck wird zunächst eine Ähnlichkeitsmatrix aufgestellt. Als Ähnlichkeitsmaß wird hier die Kullback-Leibler Divergenz (KL-Divergenz) gewählt. Die KL-Divergenz zwischen zwei Worten berechnet sich auf die Anwendung der Wortvorhersage bezogen wie folgt:

$$d_{KL}(w_1, w_2) = \sum_b p(w_1|b) \log \frac{p(w_1|b)}{p(w_2|b)} \tag{9}$$

Ist nun die Ähnlichkeitsmatrix aufgestellt, wird der Algorithmus „normalized cuts“ (siehe Abschnitt 4.1) auf diese Matrix angewendet und somit Wortcluster definiert. Abbildung 6 zeigt ein Bild, bei dem das Vokabular mit diesem Verfahren geclustert wurde. Nimmt man nun für jeden Cluster nur einen Repräsentanten, so führt dies wieder zu einer Beschneidung des Vokabulars um ca. 25%. Eine Beschriftung ist korrekt, wenn sie mit einem Wort aus dem Cluster gemacht wurde. Dadurch erhöht sich sich die Häufigkeit der Vorhersage eines Clusters zu den Häufigkeiten der einzelnen Wörter. Außerdem, steigt auch die Anzahl der korrekten Vorhersagen, da die meisten ununterscheidbaren Wörter geclustert wurden und so nun diese Wörter mit höherer Wahrscheinlich vorhergesagt werden können. Also steigt der Recall leicht an. Durch das Ansteigen der korrekten Vorhersagen, wird zusätzlich die Präzission der meisten Wörter erhöht.

2.3 Hinzunahme von Supervised Data

Mit den oben vorgestellten Mitteln wird die Worterkennung auf Basis des Vokabulars verbessert. Wenn man aber noch mal betrachtet, wie die Beschriftung zugewiesen wurde, erkennt man, dass der EM-Argorithmus eine eindeutige Beschriftung aufgrund der höchsten Wahrscheinlichkeit erzeugt. Der semantische Zusammenhang zwischen den Regionen und den Wörtern wird aber nur aus der Trainingsmenge gelernt. Dies führt zu dem Problem, dass, falls zwei Wörter immer im selben Bild auftauchen, der Algorithmus nicht zwischen diesen beiden unterscheiden kann. Zum Beispiel tritt in den meisten Bildern mit dem Wort „horse“ auch das Wort „grass“ auf. Das macht es für das Lernverfahren schwer, diese Worte zu einer Region zuzuordnen (Abbildung 6). Hätte man eine Referenz, welches dieser Wörter z.B.

Tabelle 1: Die Tabelle mit Werten aus [2] gibt die Beschriftungsperformance (Spalten 1-3) und Korrespondenz (Spalte 4) für die verschiedenen Methoden zur Beschriftung der Bildteile an. Die maximal erreichbare Korrespondenz liegt in diesem Versuch bei 301 und wird von der Nearest Neighbour Methode erreicht.

Verfahren	Unsupervised-Traingsmenge	Supervised-Traingsmenge	Testmenge	Korrespondenz
EM-Algorithmus	0.0692	0.1431	0.0597	15
EM-Algorithmus + Supervised Data	0.0782	0.1736	0.0811	202
Nearest Neighbour	0.0687	0.1486	0.1025	301

Tabelle 2: Die Werte der Tabelle stammen aus [2] und vergleichen die Nearest Neighbour Methode (N-N) mit der Beschriftung durch den EM-Algorithmus mit Supervised Data (EM + Sup). Die Tabelle gibt in den ersten beiden Spalten die Anzahl der korrekten Vorhersagen, die Gesamtanzahl der Vorhersagen sowie den prozentualen Anteil korrekter Vorhersagen mit einem Wort wieder. In der dritten und vierten Spalte sind die relativen falsch-positiven Vorhersagen der Wörter mit den beiden Verfahren eingetragen.

	Vorhersagen (EM + Sup)	Vorhersagen (N-N)	fasch positiv (EM + Sup)	fasch positiv (N-N)
eagle	0/0 (-)	4/63 (6%)	0.00	0.84
elephant	5/30 (17%)	4/30 (13%)	0.77	0.77
field	6/54(11%)	6/54 (11%)	0.85	0.85
forest	0/0 (-)	0/5 (0%)	0.00	0.95
grass	10/31 (32%)	19/54 (35%)	0.77	0.78
horses	5/42 (12%)	5/37 (14%)	0.82	0.84
lion	2/35 (6%)	2/23 (9%)	0.75	0.76
plane	9/40 (23%)	9/40 (23%)	0.70	0.70

grüne Regionen beschreibt, so ließe sich dieses Problem besser lösen. Deswegen haben Barnard *et al.* in [2] untersucht, welche Auswirkung die Hinzunahme von Supervised Data zu der Trainingsmenge hat. In dem Zusammenhang von Wortzuweisung auf einzelne Regionen spricht man bei „Supervised Data“ von Bilddaten, bei denen die Wortzuweisung „per Hand“, also von einem Menschen, vorgenommen wurde. Diese Beschriftung entspricht genau der Zuweisung, die erreicht werden soll. Die Anwendung von ausschließlich dieser Methode scheitert aber meist am großen Umfang, der zu beschriftenden Datenmenge. Also wird nur eine verhältnismäßig kleine beschriftete Datenmenge hinzu genommen und versucht, aus dieser für das Problem mit der großen Datenmenge zu lernen.
Für die Untersuchung der Auswirkungen von Hinzunahme von Supervised Data wurde die Repräsentation einer Region (siehe Abschnitt 3) auf 11-dimensionale Merkmalsvektoren begrenzt. Diese Einschränkung soll nach Barnard *et al.* eine höhere Stabilität bewirken. Die Merkmalsvektoren werden dann noch so skaliert, dass sie den Mittelwert 0 und die Varianz 1 haben. Die Vektoren spannen einen Vektorraum auf. In diesem Vektorraum liegen die Blobs aus der beschrifteten Datenmenge. Jeder dieser Blobs ist durch die Supervised Data mit einem Wort assoziert und definiert in dem Vektorraum seinen eigenen Cluster. Nimmt man jetzt noch das null-Wort auf, so kann man die Beschriftung der Regionen über eine Nearest-Neighbour Suche realisieren. Jeden Blob ordnet man also dem Cluster zu, welches die kürzeste Distanz zu dem gegebenen Blob hat. Dies hat allerdings den Nachteil, dass der Algorithmus nicht lernfähig ist, wie zum Beispiel der EM-Algorithmus aus Abschnitt 2.1. Deshalb werden die beiden Verfahren miteinander verbunden. Das heißt zunächt wird jedem Blob der nächste Nachbar im Supervised Vektorraum zugeordnet. Diese Zuornung wird dann als Initialisierung für den EM-Algorithmus verwendet und der EM-Algorithmus nimmt als Trainingsmenge die Supervised Data und die ursprüngliche Unsupervised Trainingsmenge. Damit wird dann die gemeinsame Wahrscheinlichkeit von Blobs und Wörtern, unter Benutung von Supervised und Unsupervised Data, hergestellt. Diese Methode wurde in [2] angewendet und ausgewertet. Dafür wurde die automatische Beschriftung mit und ohne Supervised Data mit der Methode, die in 3.1 vorgestellt wird, bewertet. Zusätzlich wurde die reine Nearest Neighbour Methode verglichen, bei welcher die Zuweisung nur durch die Nearest Neighbour Suche in der Supervised Data vorgenommen wird. Als erstes Maß für eine gute Beschriftung wir die Performancemessung aus Abschnitt 3.1 verwendet.

Als zweites Maß wird die Korrespondenz definiert. Für die Korrespondenz wurde dem Algorithmus auch erlaubt, mehrere Wörter für einen Blob vorherzusagen (vgl. Abschnitt 3.1). Die Korrespondenz ist die absolute Anzahl der Blobs, die mit einem Wort korrekt beschriftet wurden. Die Auswertung der Korrespondenz wurde per Hand von Menschen durchgeführt. Die Ergebnisse der Auswertung sind in Tabelle 1 aufgelistet. Die Verknüpfung von EM-Algorithmus mit Hinzunahme von Supervised Data zeigt in allen getesteten Mengen eine höhere Beschriftungsperformance und hat einen besseren Korrespondenzwert als die Beschriftung mittels reinem EM-Algorithmus. Die Messung für das Nearest Neighbour Verfahren zeigt sich uneinheitlich. Zwar zeigt die Nearest-Neighbour Methode die beste Beschriftungsperformance in der Testmenge und in der Korrespondenz, aber angewendet auf die Trainings- oder Supervised Menge, zeigt dieses Verfahren eine relativ schlechte Performance. Die schlechten Ergebnisse in den Trainingsmengen erklären Barnard *et al.* damit, dass die Supervised Data Menge relativ klein ist und somit die Nearest Neighbour Methode nicht hinreichend trainiert werden kann. Die Frage weswegen dann in der Testmenge so gute Ergebnisse erzielt werden bleibt jedoch unbeantwortet. Um ein eindeutiges Ergebnis zwischen der Nearest Neighbour Methode und dem EM-Algorithmus mit Supervised Data zu erreichen, werden diese Verfahren weiter verglichen. Hierfür werden, für die beiden Verfahren, in Tabelle 2 die Anzahl der Vorhersagen, korrekten Vorhersagen, sowie die relative Anzahl der falsch-positiver Vorhersagen ausgewählter Worte gegenübergestellt. Eine falsch-positive Vorhersage liegt vor, wenn ein Wort für eine Region falsch vorhergesagt wurde. Man sieht, dass bei jedem der dort gezeigten Worte die falsch-positiv-Rate bei dem Nearest Neigbour Verfahren gleich hoch oder höher liegt. Weiter erkennt man, dass einerseits die beiden Verfahren zwar prozentual die Wörter gleich gut vorhersagen, andererseits aber das Nearest Neigbour Verfahren die Wörter häufiger vorhersagt. In diesem Zusammenhang wird von Over-Prediction der Wörter geredet. Die Over-Prediction geben Barnard *et al.* als signifikanten Fehler an und kommen so zu dem Schluss, dass das Verfahren des EM-Algorithmus unter Zuhilfenahmen von Supervised Data das beste der vorgestellten Verfahren ist.
Bevor sich der Verbesserung der Wortvorhersage für einzelne Bildteile von einer anderen Seite genähert wird, soll nun noch einmal die bisher vorgestellten Ergebnisse zusammengefasst werden. Um Bildteile zu beschriften wird hier ein stochastisches Modell benutzt. Der EM-Algorithmus ist ein geeignetes Mittel eine solche Beschriftung zu erreichen. Die Performance dieser Wortvorhersage kann aber noch erhöht werden, indem man alle „schlechten“ Wörter, also die mit niedriger Wahrscheinlichkeit aus dem eigentlichen Vokabular streicht und den EM-Algorithmus erneut anwendet. Desweiteren gibt es in dem Vokabular noch ununterscheidbare Wörter, die in einem Cluster zusammengefasst werden. Dies bringt eine erneute Performancesteigerung mit sich. Um den semantischen Zusammenhang zwischen den Regionen zu erhöhen kann man Supervised Data in den EM-Lernalgorithmus hinzu ziehen. Dies bringt schon bei Hinzunahme von kleinen Mengen eine Performancesteigerung.

3 Einfluss der Merkmale

Für die Beschriftung von Bildteilen mit Wörtern werden zunächst die einzelnen Bildsegmente in so genannte Blobs umgewandelt. Diese werden wiederum durch ihre Merkmalsvektoren repräsentiert. Mit dem EM-Algorithmus lassen sich dann diese Blobs mit dem Wort beschriften, welches die höchste Wahrscheinlichkeit für den jeweiligen Blob aufweist. Um die Wortvorhersage weiter zu verbessern haben Barnard *et al.* in [2] untersucht, ob eine geeignete Repräsentation der Blobs die Performance der Wortvorhersage steigern kann. Dies wurde am Beispiel des Corel Data Sets untersucht. Das Corel Data Set enthält Bilder aus verschiedenen Bereichen, wie z.B. Natur, Gebäude, Tiere. Für das Experiment wurde eine Trainingsmenge von 6000 Bildern für den EM-Algoritmus und eine Testmenge von 6000 Bildern verwendet. Supervised Data lag nicht vor.

3.1 Messung der Performance

Bevor man sich Gedanken über die richtige Wahl der Merkmale macht, sollte man sich zunächst überlegen, wie später getestet werden kann, ob die vorgeschlagenen Merkmale auch wirklich den erwarteten Erfolg bringen, oder ob diese die Performance sogar verschlechtern. Dieses Problem wurde auch bereits von Barnard *et al.* in [5] untersucht. Die Performancemessung ist keine triviale Aufgabe. Der beste Ansatz ist ein Vergleich zwischen der maschinellen Wortzuweisung und einer menschlichen Wortzuweisung. Dies ist aber in dem Fall einer großen Datenmenge schwer zu erreichen. Desweiteren neigen die Gutachter dazu, bestimmte Worte wie zum Beispiel „sky“, „water“ oder „people“ häufiger zu gebrauchen als nicht so allgemeine Worte wie zum Beispiel „tiger“. Das hieße, dass mit der Strategie alle Bilder mit „sky“,

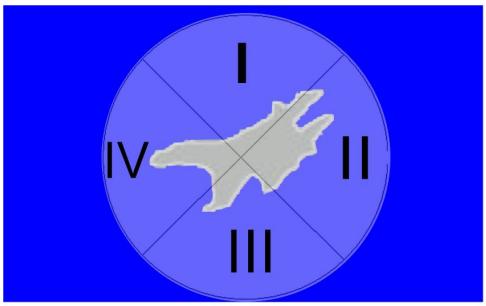

Abbildung 7: Der Farbkontext wird für die vier Quartale als die Menge der hellblauen Punkte bestimmt. Das heisst genau die Punkte, die zum Himmel gehören, aber noch im Kreis sind. Der Kreis den Radius des doppelten, mittleren Abstandes der Grenze der Region zu ihrem Schwerpunkt. Das urprüngliche Bild der Form des Jets stammt aus [1].

„water“ oder „people“ zu beschriften schon eine hohe Performance erreicht würde.
Es wird dem EM-Algorithmus erlaubt, für jedes Bild die M besten Wörter vorherzusagen. M ist hier die Anzahl der Wörter aus der Wortvorgabe für das komplette Bild. In der untersuchten Datenmenge variiert M von 1 bis 5. Für ein Bild n wird dann die Performance $P_n^{(Algorithmus)}$ definiert als

$$P_n^{(Algorithmus)} := r_n/M_n, \tag{10}$$

wobei r_n die Anzahl der Wörter ist, die im Bild n richtig vorhergesagt wurden und M_n die Anzahl der möglichen Wörter für das Bild n. Die Performance für die komplette Datenmenge ist dann das arithmetische Mittel aus allen bewerteten Bildern der Datenmenge. Also:

$$P^{(Algorithmus)} = \sum_{n=1}^{N} \frac{P_n}{N} \tag{11}$$

Dabei ist N die Anzahl aller bewerteten Bilder. Diese Performancemessung trägt aber immer noch ein Problem mit sich. Falls es nämlich es einige Wörter gibt, die sehr viel häufiger benutzt werden als andere, so ist es, mit dieser Performancemessung, eine gute Strategie alle Blobs nur mit diesen Wörtern zu beschriften. Im Corel Data Set sind die Wörter „sky“, „water“ oder „grass“ Beispiele für häufige Wörter. Das heißt in die Performancemessung sollte noch die empirische Verteilung der Wörter in der Trainingsmenge berücksichtigt werden. Dafür wird die Performance $P^{(Wortverteilung)}$ wie $P^{(Algorithmus)}$ in den Gleichungen 10 und 11, nur, dass jeweils mit den M_n häufigsten Wörtern aus der Trainingsmenge beschriftet wird. Dann ergibt sich die Endgültige Performance P als

$$P = P^{(Algorithmus)} - P^{(Wortverteilung)} \tag{12}$$

Damit ist ein für diese Zwecke brauchbares Performance-Modell konstruiert. Im nächsten Abschnitt soll dann mit hilfe dieser Perfpomancemessung gezeigt werden, wie gut oder schlecht bestimmte Merkmale sich auf die Wortvorhersage auswirken.

3.2 Wahl der Merkmale

Um den Bezug zwischen einzelnen Merkmalen und die Auswirkung einzelner Merkmale auf die Perfomance zu betrachten, werden die Merkmale in Gruppen eingeteilt. Barnard *et al.* haben die Forderung an die Merkmale gestellt, dass diese Gruppen möglichst unabhängig von einander sind. Damit wollten sie eine getrennte und möglichst aussagekräftige Auswerung erreichen. Zuerst wird eine Menge von Basismerkmalen festgelegt, die hier auf jeden Fall gebraucht werden um eine Region durch Merkmalsvektor zu beschreiben. Dazu zählen die Regiongröße, die Platzierung der Region im Bild und zwei einfache Formfunktionen. Die erste ist die Lage des Schwerpunktes in der Region und die zweite ist die Fläche der Region geteilt durch das Quadrat der Länge ihrer Grenze. Das führt allein durch die Basismerkmale schon auf einen 4-dimensionalen Vektor. Diese Merkmale haben Barnard und Duygulu schon in früheren Arbeiten (z.B. [5]) benutzt und haben damit gute Ergebnisse erziehlt. Trotzdem soll die Merkmalsmenge um möglichst aussagekräftige Merkmale erweitert werden. Die im Folgenden gewählten Eigenschaften

werden in Abschnitt 3.3 hinsichtlich Performancesteigerung der Wortvorhersage ausgewertet.
Zunächst wird die Farbe der Region betrachtet und duch drei unterschliedliche Standardmethoden kodiert, um diese später miteinander zu vergleichen. Von jeder Form der Kodierung wird jeweils der Mittelwert der Region und die Varianz gespeichert. Als erste und wohl offensichtlichste Farbdarstellung wird das Tripel (R, G, B) gewählt, wobei R die Rot-, G die Grün- und B die Blauanteile des Bildes beschreiben. Die zweite Darstellung ist die $L * a * b$ -Darstellung. Dafür werden die drei Werte L, a und b gespeichert. Dabei steht L für die Luminanz (Helligkeit) a sind die Grün- und Rot-Anteile und b die Blau- und Gelb-Anteile in der Region. Das heisst $L * a * b$ gibt Auskunft über die Helligkeit und Chominanz (= Farbe) einer Region. Als dritte Farbdarstellung wird das Tripel (r, g, S) gewählt. Bei dieser Zerlegung gilt $S = R + G + B$, $r = R/S$ und $g = G/S$. Diese drei Farbmerkmale werden der Merkmalsmenge hinzugefügt und im nächsten Kapitel miteinander bezüglich ihrer Performance verglichen.
Als nächstes wird die Textur einer Region mit in die Merkmalsmenge aufgenommen. Die Textur sollte auch ein sinnvolles Maß sein, da sie z.B. eine glatte Schneefläche von einer verputzten Wand unterscheiden lässt. Hier wird die Textur als das mittlere Ergebnis von Anwendungen von 12 Filtern mit verschiedener Orientierung dargestellt. Dazu kommt noch das mittlere Ergebnis der Differenz von vier verschiedenen Kombinationen von zwei Gaussfiltern.
Jetzt wird eine neue Formfunktion mit in die Merkmalmenge aufgenommen. Zwar sind in den Basismerkmalen schon zwei Formfunktionen enthalten, diese sollen nun aber bezüglich ihrer Aussagekraft untersucht und mit der neuen Funktion verglichen werden. In [2] sollte die Dimension der Merkmale „handlich“ gehalten werden. Dies ist im Allgemeinen für gute Formdetektoren schwer zu realisieren. Barnard *et al.* haben zu diesem Zweck eine Formrepräsentation gewählt, die auf 30 Dimensionen beschränkt ist. Zur Formulierung dieser Formfunktion wird die äußere Genzlinie jeder Region betrachtet und über die Länge der Grenzlinie aller Regionen normalisiert. Diese wird dann parametrisiert durch den Abstand zum Schwerpunkt der Region. Das Ergebnis wird wiederum geglättet und an 30 Punkten abgetastet. Dabei soll der erste Punkt per Definition oben links liegen. Das heißt der Punkt der zu der oberen linken Ecke des Bildes den kürzesten Abstand hat. Damit ist diese Formfunktion absichtlich nicht drehungsinvariant, da man annehmen kann, dass die Orientierung der Achsen noch wertvolle oder wenigstens brauchbare Informationen bietet.
Als letztes Merkmal wird der Farbkontext einer Bildregion eingeführt. Dies mag sinnvoll erscheinen, wenn man sich überlegt, wie man einen Vogel von einem Schmutzfleck unterscheiden kann. Dies scheint zunächst zwar eine triviale Aufgabe, aber mit unserem bisherigen Merkmalen sind Vogel wie auch Schmutzfleck nur ein kleiner dunkler Klecks. Wenn man aber die Umgebung mitbetrachtet, kann man sagen, das der Klecks mit höherer Wahrscheinlichkeit ein Vogel ist, wenn er von einer hellblauen Fläche (= Himmel) umgeben ist. Um den Farbkontext zu berechnen, wird zunächst der mittlere Abstand des Schwerpunktes einer Region zu deren äußerer Grenze berechnet. Nun stellt man einen Kreis um den Schwerpunkt auf mit dem doppelten des eben berechneten Abstandes als Radius. Alle Punkte dieses Kreises, die nicht in der Region selber liegen, werden in vier Quadranten eingeteilt, deren Grenzen im 45 Gradwinkel zu den Bildachsen sind und erhalten so die Quadranten oben, rechts, unten und links. Abbildung 7 zeigt dies am Beispiel eines Jets am Himmel. Für jeden der Quadranten der mehr als 100 Punkte enthält, wird nun der farbliche Mittelwert der Punkte berechnet. Sollte ein Quadrant weniger als 100 Punkte enthalten, so wird er mit der mittleren Farbe der Region selber beschriftet. Die fabliche Kodierung des Farbkontextes wurde hierbei nur in der RGB-Darstellung vorgenommen. Somit erhält man mit dem Farbkontext eine 12-dimensionales Merkmal zu der Merkmalmenge.
In diesem Abschnitt wurden nun einige Merkmale vorgestellt. Die Merkmale im einzelnen sind Größe, Lage im Bild, verschiedene Formfunktionen, Farbe in drei verschiedenen Kodierungen, Farbkontext und Textur. Im folgenden Abschnitt soll nun gezeigt werden, wie weit sich diese Merkmale auf die Performance für die Wortvorhersage für einzelne Bildregionen auswirken.

3.3 Auswertung der Merkmale

Nachdem nun einige Merkmale vorgeschlagen wurden, haben Barnard *et al.* die Güte der Merkmalsmenge überprüft. Einige Ergebnisse sind in Tabelle 3 aufgelistet. Am wichtigsten scheint die Farbeigenschaft zu sein. Dabei schneidet die rgS Zerlegung am besten ab. Die RGB Zerlegung hingegen zeigt im Vergleich eine ziemlich schlechte Performance. So sind, nach Aussage der Autoren, die rgS- und L*a*b-Werte sogar signifikant, das heißt statistisch nicht zufällig, besser als RGB. Da die Merkmale nach Barnard *et al.* voneinander unabhängig seien sollten, sollte man sich darauf beschränken die rgS-Farbinformation als einziges Farbmerkmal zu speichern. Ein Wert für die Performance, wenn man alle drei Farbcodierungen

Tabelle 3: Die Tabelle mit Werten aus [2] zeigt die Performance für die Wortvorhersage unter Verwendung verschiedener Merkmalsvektoren

Merkmalmenge	P(Trainingsmenge)	P(Testmenge)
Basismerkmale	0.019	0.020
Basismerkmale, RGB	0.076	0.057
Basismerkmale, L*a*b	0.097	0.085
Basismerkmale, rgS	0.109	0.092
Basismerkmale, rgS, Farbkontext	0.134	0.094
Basismerkmale, Textur	0.092	0.048
Basismerkmale, rgS, Textur	0.109	0.072
Basismerkmale, Form	0.053	0.016
Basismerkmale, rgS, Form	0.065	0.029
Basismerkmale, Textur, Form	0.083	0.043
Alle Merkmale	0.097	0.055

in die Merkmalvektoren aufnimmt, wurde in [2] zur Untermauerung nicht angegeben.
Als nächstes wird hier der der Einfluss der Textureigenschaften auf die Performance betrachtet. Die Hinzunahme der Texturinformationen bringt eine signifikante Performancesteigerung, wenn man es ausschlißlich mit den Basismerkmalen anwendet. Wendet man es aber zusammen mit den rgS-Farbinformationen an, so erreicht man keine weitere Verbesserug. Barnard *et al.* versuchen dies damit zu erklären, dass in den Farbinformationen schon Texturinformationen enthalten sind und diese damit nicht mehr unabhängig sind. Sie gestehen aber auch ein, dass noch weitere Experimente notwendig seien, um diese These zu untermauern.
Auch der Farbkontext bringt eine Steigerung der Performance mit sich. Allerdings fällt diese Steigerung verhältnismäßig klein aus. Dies lässt zwei Schlüsse auf den Nutzen des Farbkontextes für die Wortvorhersage zu. Die erste Möglichkeit ist, dass die Kontextinformationen für die gegebene Datenmenge nicht soviel Performance bringen, wie davon erhofft wurde. Als zweite Möglichkeit sollte betrachtet werden, dass in Barnards Experiment für den Farbkontext nur die RGB Zerlegung gewählt wurde. Wie sich der Farbkontext auf die Performance auswirkt, wenn man ihn in L*a*b oder in rgS berechnet, wurde nicht geklärt.
Die Auswertung der Formfunktion ist widersprüchlich. Tabelle 3 zeigt, dass ihre Verwendung im Trainingsset einen Performanceanstieg bringt. Allerdings zeigt sich in der Testmenge, dass sich diese erhebliche Verbesserung nicht gut verallgemeinern lässt. Das heißt, dass die hier vorgestellte Formfunktion die Wortvorhersage für Bildregionen nicht verbessert, sondern eher zu einer Verschlechterung der Performance führt. Man spricht in diesem Zusammenhang von einem Übertraining für die Trainingsmenge, da diese Formfunktion die Wortvorhersage zu sehr auf die Trainingsmenge spezialisiert. Ein weiterer Gedanke ist, dass die hinzugenommene Formfunktion nicht mehr unabhängig von den Formfunktionen in den Basismerkmalen ist. Auf diesen Punkt wird in [2] aber nicht eingegengen. Hingegen wurde erwähnt, dass gerade die Formfunktion stark von der Qualität der Segmentierung abhängt. Ein möglicher Vorschlag um die Segmentierung zu verbessern wird in Abschnitt 4.2 erörtert. Aus der Auswertung der Merkmale kann man den Schluss ziehen, dass die Blobs angemessen repräsentiert werden durch einen 22-dimensionalen Merkmalsvektor. Die Komponenten dieses Vektors sind die BasisMerkmale (4 Dimensionen) die rgS-Farbinformationen (6 Dimensionen) und der Farbkontext (12 Dimensionen). Die Texturinformationen haben keine weitere Verbesserung herbeigeführt. Desweiteren scheint die Form schon ausreichend durch die Basismerkmale repräsentiert zu sein, da die Hinzunahme der hier vorgestellten Formfunktion zu einem schlechterem Ergebnis führt. Die Ergebnisse deuten nach Barnard *et al.* an, dass eine weitere Vergrößerung der Merkmalmenge zu Übertraining führt, obwohl mehr Informationen zur Objekterkennung in die Merkmalmenge mit einfließt.

4 Einfluss des Segmenters

Nachdem mit dem EM-Algorithmus ein Verfahren vorgestellt wurde, wie sich einzelne Bildteile mit Wörtern aus einem Vokabular beschriften lassen und wie man diese Bildregionen für dieses Problem gut repräsentieren kann, soll als nächstes auf die Herstellung der Regionen eingegangen werden. Es existieren

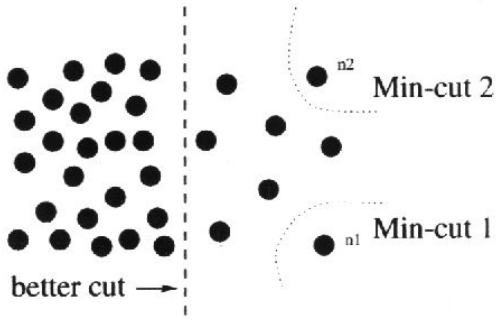

Abbildung 8: In diesem Bild aus [7] ist die Ähnlichkeit zwischen zwei Knoten über die Entfernung dargestellt. Ein minimaler *cut* ergibt sich indem man den Knoten n_1 oder n_2 als Komponente heraus schneidet. Eine bessere Zerlegung wäre aber etwa die, welche durch die senkrechte gestrichelte Linie dargestellt ist.

zahlreiche Segmentierungsalgorithmen, mit verschiedenen Ansätzen. Das „Normalized Cuts“ Verfahren soll im Folgendem exemplarisch vorgestellt werden. Anschließend wird betrachtet, wie gut sich dieses Verfahren für die Wortbeschriftung der Segmente mittels des EM-Algorithmus eignet. Abschließend wird untersucht, ob man hergestellte Segmente weiter verbessern kann, indem man die Wortbeschriftung zu Hilfe nimmt.

4.1 Normalized Cuts

Normalized Cuts segmentiert ein Bild durch graphentheoretisches Clustern. Das heißt als erstes muss das zu segmentierende Bild in einen Graphen $G = (V, E)$ umgewandelt werden. Hierbei soll der Graph ungerichtet und gewichtet sein. Die Knotenmenge entspricht der Menge der Pixel. Dann werden alle paarweise verschiedenen Knoten mit Kanten verbunden. Das Kantengewicht $w_{a,b}$ zwischen zwei Knoten a und b bestimmt sich aus der Ähnlichkeit der Pixel, die mit a bzw. b assoziiert sind. Die Ähnlichkeit kann durch verschiedene Eigenschaften bestimmt werden. So können unter anderem ihr Abstand, die Farbe, oder Textur in das Ähnlichkeitsmaß miteinbezogen werden. Zur Segmentierung kann nun der mit dem Bild assozierte Graph in mehrere zusammenhängende Komponenten unterteilt werden, die sich gegenseitig möglichst wenig ähneln. Auf der anderen Seite soll jede Komponente in sich eine möglichst hohe Ähnlichkeit enthalten. Das heißt, der Graph soll so geschnitten werden, dass bezüglich der Gewichtung der Kanten ein minimaler Schnitt entsteht. Das Gewicht eines Schnitts in zwei nicht leere Komponenten A und B ist in der Graphentheorie definiert durch die Summe der Gewichte der Kanten, die geschnitten wurden:

$$cut(A, B) = \sum_{a \in A, b \in B} w_{a,b} \tag{13}$$

Mit alleine dieser Definition stößt man aber auf das Problem, dass ein minimaler Schnitt bezüglich der *cut*-Funktion nicht immer die optimale Ergebnisse bezüglich der Segmentierung liefert. Dies ist eine Folge daraus, dass im Allgemeinen die *cut*-Funtion wächst, wenn mehr Kanten geschnitten werden. Daraus folgt, dass meist sehr kleine Komponenten aus der Knotenmenge gelöst werden. Abbildung 8 zeigt ein Beispiel dafür. Um bessere Schnitte erreichen zu können setzten Shi und Malik in [7] die *cut*-Funktion in Bezug zu den Kanten des kompletten Graphen. Diese Messung des Schnitts nennt sich „Normalized Cut“ und berechnet sich durch

$$ncut(A, B) = \frac{cut(A, B)}{cut(A, V)} + \frac{cut(A, B)}{cut(B, V)} \tag{14}$$

D.h. der *cut* wird also jeweils in Beziehung gesetzt zu dem Gewicht aller Knoten, die in A bzw. B mindestens ein Ende haben. Mit einem Verfahren, welches einen minimalen Normalized Cut findet, lässt sich dann ein Bild durch wiederholtes Anwenden segmentieren. Um den minimalen *ncut* zu berechnen, wird zunächst der Graph in die Adjazenzmatrix W überführt, mit

$$W_{ij} = w_{i,j} \tag{15}$$

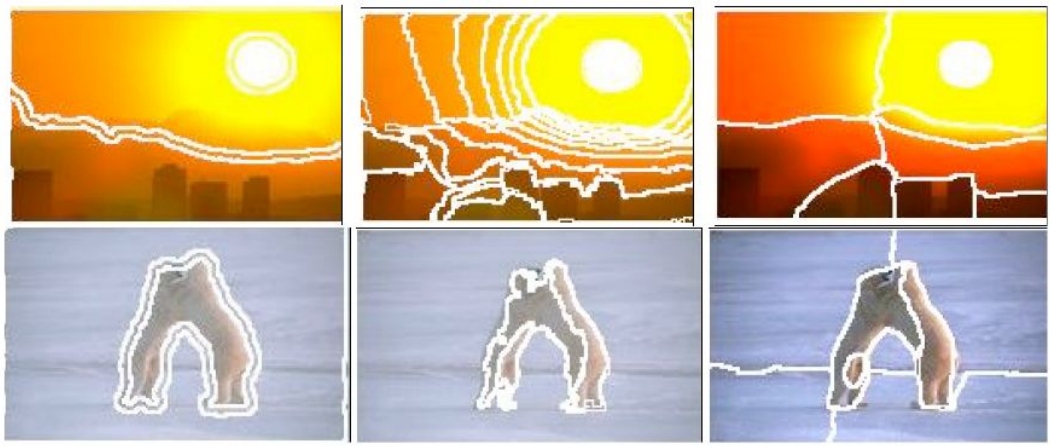

Abbildung 9: Beispielsegmentierungen aus [2] für EM for Blobworld (links), Mean Shift (mitte) und Normalized Cuts (rechts). In den linken Bildern ergeben sich zwischen zwei Regionen Doppellinien als Grenze, da EM for Blobworld die Pixel im Grenzbereich keiner der Regionen zuweist.

Des weiteren wird eine Diagonalmatrix D definiert, mit der Summe aller Kantengewichte eines Knotens auf der Diagonalen:

$$D_{ij} := \begin{cases} \sum_n^{|V|} W_{in} & \text{für } i = j \\ 0 & \text{für } i \neq j \end{cases} \tag{16}$$

Eine Näherung für den minimalen *ncut* wird dann gegeben durch die Berechnung des verallgemeinerten Eigenvektors y aus

$$(D - W)y = \lambda Dy \tag{17}$$

Eine Herleitung dieser Formel wird hier, mit Verweis auf [7] nicht gegeben. Man wählt als y den Eigenvektor zum zweitkleinsten verallgemeinerten Eigenwert, da der kleinste verallgemeinerte Eigenwert immer null ist. Sei $y = (y_1, \ldots, y_n)$ mit $n = |V|$, dann gehören alle Knoten i mit $y_i > s$ zu dem Segment A und alle anderen zu Segment B. Der Schwellwert s kann auf verschiedene Weisen gewählt werden. In [6] wird $s = y_i$ für alle $i = 1, \ldots, n$ gesetzt und jeweils der Normalized Cut bezüglich dieser Zerlegung berechnet. Anschließend wird die Zerlegung mit dem minimalen Normalized Cut gewählt.

4.2 Segmenter und Wortvorhersage

Nachdem am Beispiel von Normalized Cuts ein kleiner Einblick in die Segmentierung von Bildern gegeben wurde, wird als nächstes auf den Bezug der Segmentierung zur Beschriftung von Bildteilen eingegangen. Es soll untersucht werden, welchen Einfluss die Wahl des Segmenters auf die Wortvorhersage hat. Zu diesem Zweck haben Barnard *et al.* in [2] Normalized Cuts mit zwei anderen Segmentierungsverfarhen verglichen. Die beiden anderen Verfahren sind: „Expectation-Maximum for Blobworld“ und der „Mean Shift“-Algorithmus. EM for Blobworld ist ein stochastisches Modell und verwendet eine abgewandelte Form des EM-Algorithmus aus Abschnitt 2.1. Der Mean Shift-Algorithmus arbeitet im Vektorraum über die Merkmale. Eine dort berechnete Partitionierung wird auf die Pixel im Bild übertragen. Ein Segmentierungsbeispiel für die drei Verfahren ist in Abbildung 9 gegeben. Diese drei Verfahren werden jeweils auf die Test- und Trainingsmenge angewendet. Nach Training mit dem EM-Algorithmus, werden die Beschriftungen für die Bildmengen durchgeführt.
Die Testmenge für diesen Versuch beträgt insgesamt 10,000 Bilder des Corel Data Sets. Für die Auswertung wurden jeweils die 2, 4, 6, 8, 10 und 12 größten Regionen beschriftet. In Abbildung 10 ist die Performance über der Anzahl der Segmente für die verschiedenen Segmenter aufgetragen. Man erkennt, dass sich Normalized Cuts besser bewährt als die beiden anderen Verfahren. Die Performance des Mean Shift-Algorithmus liegt in diesem Versuch am zweiter Stelle und überbietet noch die Performance des EM-Blobworld Segmenters. In der ersten Menge liefert Normalized Cuts sogar ein signifikant bessers Ergebnis als die anderen beiden Verfahren. Das lässt darauf schliessen, dass sich für die Beschriftung von Bildteilen das Normalized Cuts Verfahren gut eignet und den anderen hier verwendeten Verfahren vorgezogen werden sollte.

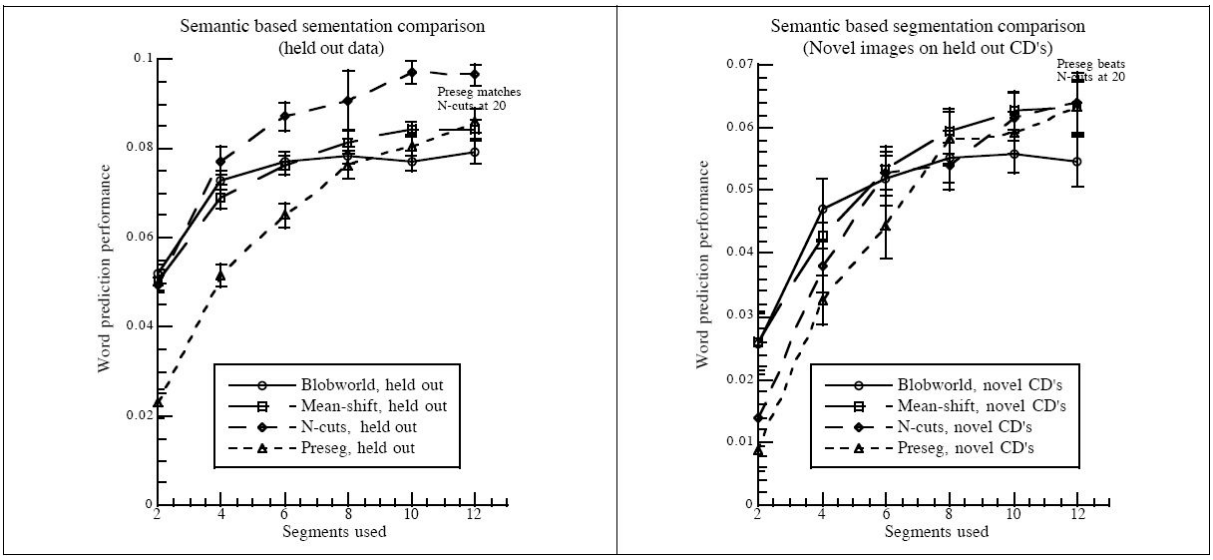

Abbildung 10: Die Performance der drei Segmenter ist für 2 Testmengen gegen die Anzahl der grössten Segmente, die für die Wortvorhersage herangezogen werden aufgetragen. Zusätzlich ist noch die Initialausgabe (preseg) von Normalized cuts aufgetragen. Diese Ausgabe beruht auf der Implementierung von Normalized Cuts und liefert ein übersegentiertes Ergebnis. Die linke Grafik zeigt die Performance auf einer Testmenge, mit ähnlichen Bildern wie in der Trainingmenge. Die rechte Grafik zeigt die Performance auf einer Testmenge, in der Bilder aus anderen Kategorien als die Trainingsmenge. Assymtotisch ist in beiden Testmengen die Performance von Normalized Cuts am größten un von Blobworld am kleinsten. Beide Grafiken stammen aus [2].

Mit Normalized Cuts ist jetzt ein Verfahren gewählt, welches für die Beschriftung von Bildteilen gute Ergebnisse liefert. Dennoch teilt es sich den Nachteil mit allen anderen Segmentern, dass es ausschließlich auf der Pixelebene arbeitet. Bei solchen Verfahren spricht man von Low-Level-Verfahren. Das heißt, dass diese Verfahren keinen semantischen Zusammenhang zwischen einzelnen Regionen erkennen. So gibt es z.B. keinen Segmenter, der die schwarze und die weiße Hälfte eines Pinguins als eine einzige Region erkennt. Von Barnard *et al.* wurde versucht, die Wortvorhersage als High-Level-Konstrukt zu verwenden und somit die Segmentierung zu verbessern. Dafür wurde die Annahme aufgestellt, dass Segmente, die nebeneinander liegen und im Sinne der Wortvorhersage ähnlich sind, miteinander verschmolzen werden können. Wenn z.B. zwei braune Regionen direkt nebeneinander liegen und beide werden mit z.B. „animal“ oder „bear“ beschriftet, so ist eine Verschmelzung attraktiver als eine Verschmelzung mit einer anders beschrifteten Region (Abbildung 11). Als Indiz für die Attraktivität der Verschmelzung zweier Regionen b_1 und b_2 wurde das skalare Produkt der Wahrscheinlichkeiten $p(w|b)$ gewählt.
Die Auswertung dieser Verschmelzung muss unter Zuhilfenahme von menschlich bewerteten Daten ausgeführt werde. Zwei Experten geben in einem Datenset per Hand an, ob die maschinell berechneten

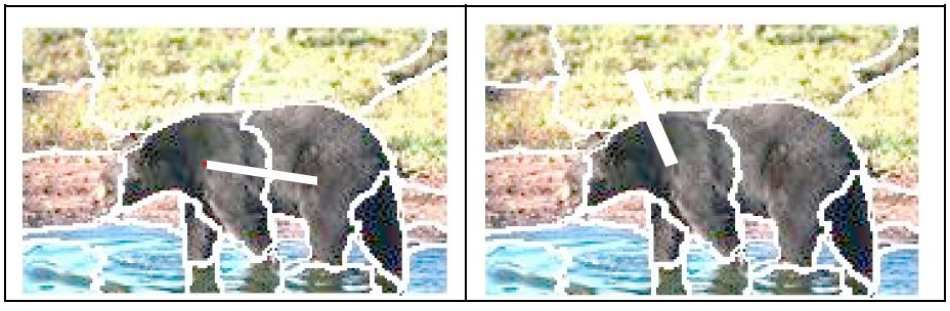

Abbildung 11: In diesem Beispiel aus [2] wird die Attraktivität der Verschmelzung der Regionen überprüft, die hier mit einer Linie verbunden sind. Hier wird die Verschmelzung linken Bild als besser bewertet, als die im rechten Bild

Tabelle 4: Normalisierte Performance der maschinellen Verschmelzung. Die Wahrscheinlichkeit, dass eine Verschmelzung richtig ist, steigt mit der Güter der maschinellen Performance. Die Werte der Tabelle sind aus [2] übernommen.

Einteilung der Scores aus Wortvorhersage	Durschschnittliche Bewertung durch Gutachter
0-20%	0.420
20%-40%	0.436
40%-60%	0.471
60%-80%	0.484
80%-100%	0.525

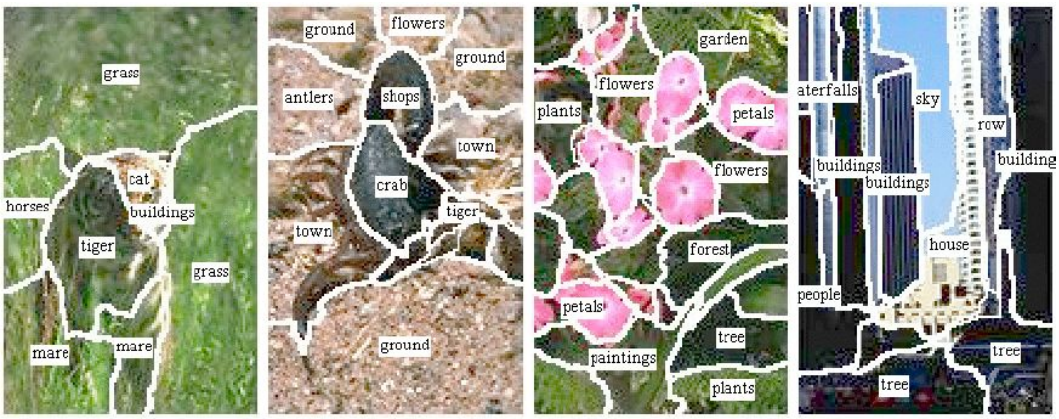

Abbildung 12: Im segmentierten Bild werden die automatischen Beschriftungen direkt im Bild plaziert. Das hier gezeigte Ergebnis zeigt trotz einiger Fehler (z.B. „mare“ oder „horses“ im ganz linken Bild) gute Ergebnisse. Allerdings wurden diese Bilder ohne Hinzunahme von Supervised Data beschriftet. Daher ist Anzunehmen, dass die Beschriftung weiter verbessert werden kann, indem man die Initialisierung des EM-Algorithmus gemäß Abschnitt 2.3 durchführt und Supervised Data mit in die Trainingsmenge aufnimmt. Die hier gezeigten Bilder stammen aus dem Corel Data Set und die Beschriftungen wurden in [5] hergestellt.

Verschmelzungen gut oder schlecht sind. Die maschinell vergebenen Scores werden eingeteilt in die besten 20%, die nächst besten 20% usw. und dann mit den Scores der Gutachter verglichen. Tabelle 4 zeigt, dass ansteigende maschinelle Scores auch bessere menschliche Beurteilungen mit sich bringen. Allerdings ist mit diesen Scores nur ein erster Schritt in Richtung der semantischen Verschmelzung getan. Es besteht noch immer Forschungsbedarf um High-Level Methoden auf Segmentierung anzuwenden.
Es wurde in diesem Abschnitt gezeigt, dass es einen Zusammenhang zwischen der Wahl des Segmenters und der Güte der Wortvorhersage gibt. Das graphentheoretische Segmentierungsverfahren Normalized Cuts hat sich hier als guter Segmenter erwiesen. Durch Hinzunahme von Wortvorhersage als High-Level-Konstrukt für Regionen lässt sich die Segmentierung noch verbessern.

5 Zusammenfassung

Hier wurde ein Modell vorgestellt, um Wortvorhersagen für einzelne Bildsegmente zu erstellen. Zunächst werden die zu beschriftenden Bilder segmentiert. Dazu hat sich der Normalized Cuts Algorithmus als gut erwiesen. Normalized Cuts wandelt jedes Bild zunächst in einen gewichteten Graphen um. Dieser Graph soll nun so in Komponenten unterteilt werden, dass die Gewichtung des Schnittes möglichst klein ist. Zu diesem Zweck wird der Graph in eine Blockdiagonalmatrix umgewandelt, für die der Schnitt leichter berechnet werden kann. Die Schnitte in dieser Matrix lassen sich auf die Pixel im Bild übertragen. Hat man ein so segmentiertes Bild vorliegen, so wird jede Region durch ihren Merkmalsvektor beschrieben. Als gute Wahl für die Merkmale haben sich Lage-, Größen-, Form-, Farbkontext- und rgS-Farbinformationen herausgestellt.
Die so repräsentierten Blobs werden in Cluster eingeteilt und diskretisiert. Durch den EM-Algorithmus lässt sich dann für jeden Blob b das Wort w_{max} mit der höchsten Wahscheinlichkeit $p(w_{max}|b)$ bestimmen.

Werden des weiteren noch die Worte mit der geringsten Wahrscheinlichkeit aus dem Vokabular gestrichen, so erhöht sich Recall und Präzision nach einem erneuten Training durch den EM-Algorithmus. Durch die Verwerfung von schlechten Vorhersagen durch die Einführung eines null-Schwellwertes und semantisches Clustern wird die Wortvorhersage weiter verbessert. Eine weitere signifikante Verbesserung erhält man durch Verwendung von Supervised Data und zusätlichem Lernen aus der Datenmenge. Allerdings wurde nur ansatzweise gezeigt, dass dieses Verfahren besser ist, als reines lernen aus der Supervised Menge mittels Nearest Neighbour. Dieser Ansatz lieferte auf der Testmenge eine bessere Performance. Ein weiterer Test mit verschiedenen Trainingsmengen, in denen das Verhältnis von Supervised zu Unsupervised Data variiert, würde wahrscheinlich mehr Klarheit bringen.
Wendet man diese Methoden auf eine Bildmenge an, so kann man schon gute Wortvorhersagen für die Bildregionen treffen. In Abbildung 12 wurde schon ein Beispiel gezeigt, zu welchen Ergebnissen man mit diesem Verfahren kommen kann. Trotzdem bleiben einige Fragen vorerst unbeantwortet. So wurde nicht untersucht ob man die Wahl der Merkmale noch verbessern kann, indem man den Farbkontext durch z.B. eine rgS-Zerlegung berechnet oder eine andere Formfunktion verwendet.
Ein anderer Ansatzpunkt ist die Benutzung der Wortvorhersage zur Verbesserung der Segmentierung, indem man ähnliche Regionen miteinander verschmelzen lässt. Eine Untersuchung, ob man mit der Segmentierung nach Verschmelzung der Regionen und anschließendem Retraining mit dem EM-Algorithmus wiederum eine Verbesserung der Wortvorhersage bekommt wurde nicht durchgeführt. Außerdem wurde für die Verschmelzung der Regionen nur ein erster Ansatz geliefert. Dieser kann wahrscheinlich durch weitere Forschungsarbeiten noch verbessert werden.
Eine weitere Frage ist, inwieweit sich die hier gezeigte Vektorquantisierung und Wortvorhersage auf medizinische Bilddatenbanken anwenden lässt. Für eine Anwendung in der Medizin müsste man wahrscheinlich die Merkmale der Regionen anders wählen, da in der Medizin größtenteils schwarz-weiß Bilder verwendet werden. Des weiteren bleibt zu untersuchen, wie weit die Wortclusterung geführt werden darf, um noch brauchbare Informationen zu erhalten. Clustert man nämlich zu viele Wörter zusammen, so kommt man nur noch auf Beschriftungen der Form „Knochen“, „Weichteile“ und „Luft“, welche nicht mehr als brauchbar anzusehen sind. Eng mit diesem Problem verwandt ist das Problem, wie weit man die Segmentierung laufen lässt. Ob man z.B. die komplette Wirbelsäule oder einzelne Wirbel als Segmente braucht, und ob es dann nach der Segmentierung noch sinnvoll ist einzelne Regionen wieder verschmelzen zu lassen. Diese Probleme können mit den hier bearbeiteten Papern nicht gelöst werden und machen weitere Forschungen notwendig.

Literatur

[1] Duygulu P, Barnard K, de Freitas JFG, Forsyth DA: Object Recognition as Machine Translation: Learning a Lexicon for a Fixed Image Vocabulary. Proc. Seventh European Conference on Computer Vision, Copenhagen, Denmark, pp. IV: 97-112, 2002

[2] Barnard K, Duygulu P, Guru R, Forsyth DA: The Effects of Segmentatin and Feature Choice in a Translation Model Of Object Recognition. Proc. Computer Vision and Pattern Recognition, pp. II: 675-682, 2003

[3] Brown P, Della Pietra SA, Della Pietra VJ, Mercer RL. The mathematics of statistical machine translation: Parameter estimation Computal Linguistics, 32(2):263-311, 1993

[4] Dempster AP, Larid NM, Rubin DB: Maximum Likelihood from Incomplete Data via the EM-Algorithm. Journal of the Royal Statistical Society, Series B vol. 39, pp. 1-38, 1977

[5] Barnard B, Duygulu P, de Freitas N, Forsyth D, Blei D, Jordan MI: Matching Word and Pictures. Journal of Machine Learning Research, vol 3, pp. 1107-1135, 2003

[6] Forsyth DA, Ponce J: Computer Vision: A modern approach. Upper Saddle River, NJ: Prentice Hall, 2003

[7] Shi J, Malik J: Normalized Cuts and Image Segmentation. IEEE Transactions on Pattern Analysis and Machine Intelligence, vol 22, pp. 888-905, 2000