FAKULTÄT
FÜR !NFORMATIK
Faculty of Informatics

Semi-automatic Annotation on Image Segmentation Hierarchies

DIPLOMARBEIT

zur Erlangung des akademischen Grades

Diplom-Ingenieur

im Rahmen des Studiums

Visual Computing

eingereicht von

Georg M. Zankl
Matrikelnummer 0625388

an der
Fakultät für Informatik der Technischen Universität Wien

Betreuung: Univ.Ass. Dipl.-Ing. Dr.techn. Yll Haxhimusa
Mitwirkung: Ing. Dr. techn. Adrian Ion

Wien, 12.10.2012

(Unterschrift Verfasser)

(Unterschrift Betreuung)

Technische Universität Wien
A-1040 Wien ▪ Karlsplatz 13 ▪ Tel. +43-1-58801-0 ▪ www.tuwien.ac.at

Erklärung zur Verfassung der Arbeit

Georg M. Zankl
Laaer-Berg-Str. 26/361, 1100 Wien

Hiermit erkläre ich, dass ich diese Arbeit selbständig verfasst habe, dass ich die verwendeten Quellen und Hilfsmittel vollständig angegeben habe und dass ich die Stellen der Arbeit - einschließlich Tabellen, Karten und Abbildungen -, die anderen Werken oder dem Internet im Wortlaut oder dem Sinn nach entnommen sind, auf jeden Fall unter Angabe der Quelle als Entlehnung kenntlich gemacht habe.

(Ort, Datum)

(Unterschrift Verfasser)

Acknowledgements

I owe my gratitude to many people for variaious kinds of help with this work. To begin with I would like to thank my supervisor Yll Haxhimusa, who had a sympathetic ear and patience at all times. His helpful comments and scientific insight pushed me and my work forward.

I am grateful to Adrian Ion. This thesis would not have been possible without his guidance and encouragement. We spent many hours discussing details or fundamentals of this work in order for us to converge to the best solution.

Thanks to Walter G. Kropatsch, Head of the Pattern Recognition and Image Processing (PRIP) Group for making it possible to carry out this work in his group. Also, I would like to thank everybody at PRIP for helpful questions and comments during my internal presentations.

Finally, thanks to my family, especially my parents Wilhelm and Hannelore for their continuous love and support.

Abstract

In the field of object recognition in natural images, a variety of established tasks exist, which are focus of attention when it comes to comparing different methods, for example image segmentation, semantic image segmentation or object detection. Image segmentation is the task of grouping pixels in an image that belong to the same region or object. Semantic image segmentation is the task of assigning a semantic label to each pixel of the image. The semantic labels can be objects: for example *car*, *person*, *building*; or classes of areas in an image: *sky*, *floor*, *vertical surface*. Object detection is the task of predicting occurrence and position in an image, for example by determining a bounding box of the object. Traditional object recognition challenges have limitations such as ambiguity in more general contexts. For example for a single natural image, there are often multiple image segmentations a human would consider to be correct, depending on the object that person is particularly interested in. We raise the question: "Is there a different task, that overcomes these limitations?" As an example we propose the task of interactively assigning a semantic label to each segment of a segmentation hierarchy. The result can be represented as a stack of semantic segmentations, with an inclusion-relationship between segments of adjacent segmentations. The focus of this work is to provide a solution to this task and discuss advantages and problems that arise. The main disadvantage is that it is harder to obtain suitable ground-truth that consists of annotated segmentation hierarchies. Also the quality of underlying segmentation methods is, in general, sub-optimal for natural images. The main advantage is that the structure implied by the occurrence of labels in the ground-truth can be used to aid the user in labeling the segments of the hierarchy. We propose a framework that consists of a feedback loop, where a label prediction is provided by the framework and a human user may select one or more misclassified segments and assign the correct label. This process can be repeated until the user is satisfied. The prediction is done using a Conditional Random Field (CRF) that is modified so that we are able to condition the model on the segmentation hierarchy as well as the user input. The framework is evaluated on two distinct datasets by comparing its quality to a straight-forward baseline. The baseline consists of a single prediction step of the proposed framework followed by fully manual correction of the segments without new predictions. The results show a significant difference in quality, after several user interactions. For example after 20 user interactions the baseline adjusts 20 misclassified segments, while the CRF-based framework adjusts about 130 misclassified segments for the two datasets. This experiment illustrates the potential of structured prediction for the proposed task.

Kurzfassung

Wenn es darum geht verscheidene Methoden der Objekterkennung in natürlichen Bildern zu vergleichen, stehen diverse etablierte Aufgaben im Mittelpunkt. Beispiele dafür sind Bildsegmentierung, semantische Bildsegmentierung und Objekterfassung. Bildsegmentierung ist die Aufgabe Bildpixel, die zur selben Region oder zum selben Objekt gehören, zu gruppieren. Semantische Bildsegmentierung ist die Aufgabe jedem Bildpixel eine semantische Bezeichnung zuzuordnen. Eine semantische Bezeichnung kann ein Objekt sein: zum Beispiel *Auto*, *Person*, *Gebäude*; oder eine Klasse von Bereichen in einem Bild: *Himmel*, *Boden*, *vertikale Fläche*. Objekterfassung ist die Aufgabe Aufkommen und Position eines Objektes in einem Bild vorherzusagen, indem zum Beispiel der Rahmen (Bounding Box) eines Objekts bestimmt wird. Traditionelle Aufgaben der Objekterkennung haben gewisse Einschänkungen, wie etwa Mehrdeutigkeit in allgemeinerem Kontext. Zum Beispiel gibt es oft mehrere Bildsegmentierungen für ein natürliches Bild, die ein Mensch als richtig beurteilen würde, abhängig davon welches Objekt besonders interessant für die entsprechende Person ist. Wir stellen die Frage: "Gibt es eine Alternative, die diese Einschränkungen überwinden kann?" Als Beispiel schlagen wir die Aufgabe vor, interaktiv jedem Segment einer hierarchischen Segmentierung eine semantische Bezeichnung zuzuordnen. Das Ergebnis kann dann als ein Stapel semantischer Bildsegmentierungen dargestellt werden, wobei es eine Inklusionsrelation zwischen Segmenten angrenzender Segmentierungen gibt. Der Fokus dieser Arbeit ist es, eine Lösung der vorgeschlagenen Aufgabe vorzustellen und auftretende Vorsowie Nachteile zu diskutieren. Der größte Nachteil ist, dass es schwieriger ist passende Ground Truth zu finden - in unserem Fall besteht diese aus beschrifteten hierarchischen Segmentierungen. Außerdem ist die Qualität der zugrundeliegenden Segmentierung im Allgemeinen suboptimal für natürliche Bilder. Der wesentliche Vorteil ist, dass die Struktur der Beschriftungen in der Ground Truth dazu verwendet werden kann, dem Benutzer zu helfen neue hierarchische Segmentierungen zu beschriften. Wir präsentieren ein Framework, das eine Feedbackschleife beinhaltet, bei der eine Beschriftung vom Framework vorhergesagt wird und der Benutzer einen oder mehrere falsch bezeichnete Segmente selektieren und die korrekte Bezeichnung zuordnen kann. Dieser Vorgang kann wiederholt werden, bis der Nutzer zufrieden mit dem Ergebnis ist. Die Vorhersage der Beschriftung wird mit einem Conditional Random Field (CRF) berechnet, das adaptiert wird, um das Modell sowohl auf die hierarchische Segmentierung als auch auf die Benutzereingaben zu konditionieren. Das Framework wird auf zwei verschiedenen Datensätzen evaluiert, indem die Qualität zu einer einfachen Baseline verglichen wird. Diese Baseline besteht aus einer einzelnen Vorhersage der Beschriftung gefolgt von vollständig manueller Korrektur der Bezeichnungen, ohne erneute Vorhersagen. Ergebnisse zeigen eine wesentliche Differenz in Qualität, nach mehreren Benutzereingaben. Zum Beispiel nach 20 Interaktionen korrigiert

die Baseline 20 falsch bezeichnete Segmente, während das CRF-basierte Framework ungefähr 130 Bezeichnungen auf beiden Datensätzen korrigiert. Das Experiment zeigt das Potential von Structured Prediction für die gegebene Aufgabe.

Contents

CHAPTER 1

Introduction

One of the tasks, currently considered essential to improve on in computer vision, is semantic image segmentation [7]. The success of a method solving this problem relies on the ability to correctly identify objects present in an image and to correctly outline the regions that correspond to the projections of these objects to the image plane – both are non-trivial sub problems. However, considering both compound type objects and their parts, semantic classification of pixels may not be unique - pixels might belong to multiple classes, e.g. in Figure 1.1 a pixel in the highlighted region may belong to class *wheel* as well as class *car*, possibly even *shoe* and *image*. Furthermore, class-independent image segmentation (grouping pixels together such that they represent objects) is ill-posed [17]. These problems motivate our approach to consider the task of semantic labeling of a segmentation hierarchy. We mitigate both aspects by having an inclusion hierarchy for the image regions and a semantic label hierarchy corresponding to the composition of parts into objects.

Current automatic methods for semantic segmentation show impressive performance [6, 9, 13], but in many cases the results are not satisfying enough (e.g. the best results on VOC2011[1] are below 50%). Methods that require annotated ground-truth for their supervised training dominate the state-of-the art. However, high quality annotations are costly to obtain manually. We consider interactive (semi-automatic) methods that include a feedback-loop, thus improving on the automatic results and providing a more efficient way to obtain annotated ground-truth than fully manual approaches.

The focus of this work is the task of interactive semantic labeling, given a segmentation hierarchy. The proposed framework integrates a Conditional Random Field (CRF), whose dependencies are defined by the hierarchy, and feedback provided by a human user (see Fig. 1.2 for an overview of the workflow).

[1] `http://pascallin.ecs.soton.ac.uk/challenges/VOC/voc2011/results/index.html.`

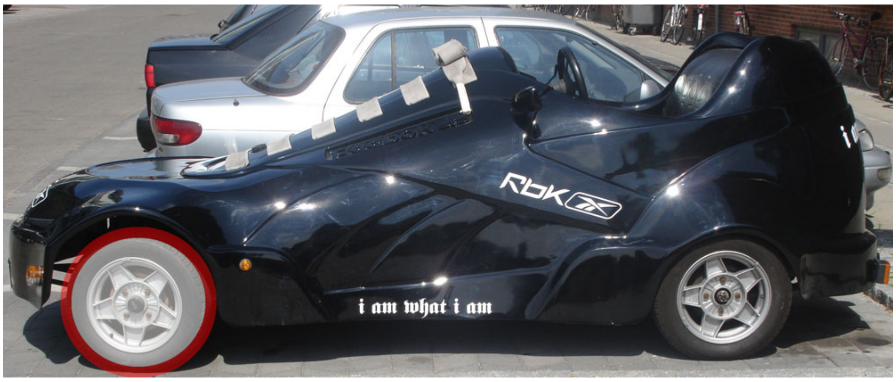


Figure 1.1: Example illustrating the ambiguity of single object classes per pixel. The highlighted wheel may have the classes *wheel*, *car*, *shoe* and *image*. Original Image by Mads Boedker [2].

1.1 Problem Statement

Let $\mathcal{S}$ be a segmentation hierarchy of an image. Let $\mathcal{U}$ be the user provided input in an interactive framework. Given a set of object classes $\mathcal{L}$, we are looking for a model f that computes a labeling

$$y = f(\mathcal{S}, \mathcal{U}, \mathcal{L}) \qquad y \in \mathcal{L}^{|\mathcal{S}|}, \tag{1.1}$$

assigning an object class $y_i \in \mathcal{L}$ to each segment $s_i \in \mathcal{S}$. We want this model to capture structural dependencies between object classes in the segmentation hierarchy, s.t. given $\mathcal{U}$, the computed labeling y is closer to the optimal labeling y^* than for the case of independent classification of segments. In an interactive framework, f can be evaluated after each user interaction in order to provide visual feedback to the user and prevent user interactions that are not necessary, considering the improved labeling based on the structural properties of the segmentation hierarchy.

1.2 Aim of the work

The goal of this work is to provide an interactive framework for generating ground-truth labelings of hierarchical image segmentations. Important factors are fast inference (ideally in linear time), so that the interaction can progress fluently, and high initial quality as well as a high increase in quality of the labeling per user interaction. Note that there is a trade-off between these last two properties, where we prefer high initial quality because it gives the method a head start for the convergence to a classification rate of 100%.

1.3 Definitions

In the following we show a table of commonly used terms and definitions.

Segment s_i is a set of pixels of a connected region i in an image.

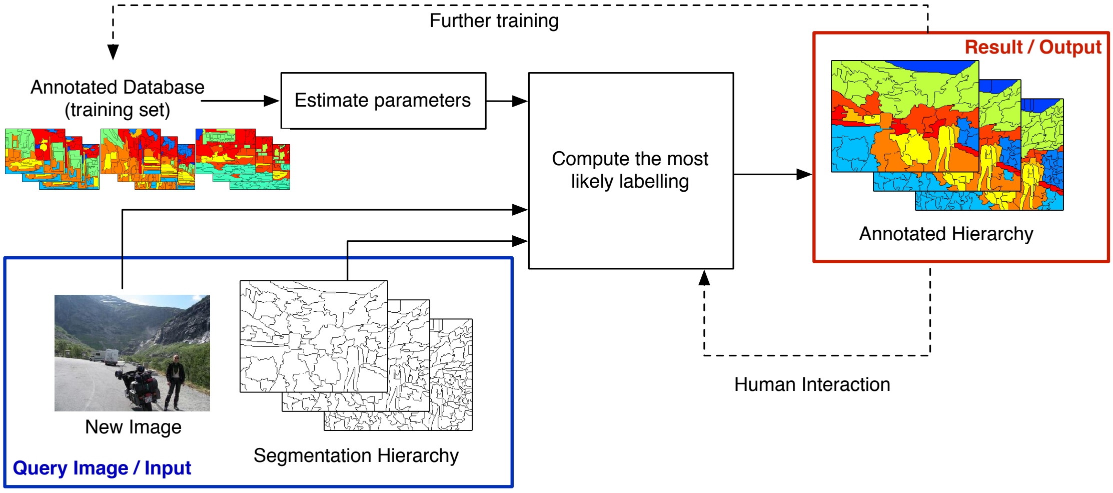


Figure 1.2: Overview of the proposed framework: in a preliminary step we train the parameters of our CRF-based model using a training set consisting of semantically labeled image segmentation hierarchies; at test time the user provides an image and the corresponding segmentation hierarchy. The labeling process alternates between providing a prediction of the labels, and the user correcting a misclassified segment. Segmentations are represented by the outline of the respective segments in the image, semantic segmentations are represented by colored segments, with the color being associated with an object class, and segmentation hierarchies are represented as a stack of respective segmentations.

Segmentation S_l at level l is a set of pairwise disjoint segments $s_i \in S_l$ covering the whole image. The coarseness of the segmentation is determined by the level $l = 0 \cdots N$, where $l = 0$ is the finest and $l = N$ the coarsest segmentation.

Parent-child mapping $m_{\mathcal{S}} : \mathcal{S} \rightarrow \mathcal{P}(\mathcal{S})$ associates to $s_i \in S_l$ a set of segments $\{s_j\} \subseteq S_{l-1}$ s.t. $s_j \subset s_i$, or the empty set if $l = 0$, with $\mathcal{P}(\mathcal{S})$ being the power set of $\mathcal{S}$. $\{s_j\}$ are the children of s_i. Note that this mapping induces the hierarchy among the segments in $\mathcal{S}$.

Child-parent mapping $m'_{\mathcal{S}} : \mathcal{S} \rightarrow \mathcal{P}_1(\mathcal{S})$ associates to $s_i \in S_l$ its parent segment if it exists,

$$m'_{\mathcal{S}}(s_i) = \begin{cases} \{s_j\} & \text{iff } s_j \in m_{\mathcal{S}}(s_i) \\ \emptyset & \text{otherwise} \end{cases} \quad . \tag{1.2}$$

Segmentation hierarchy $\mathcal{S}$ of an image consists of a segmentation $S_l \subset \mathcal{S}$ per level l and a parent-child mapping m. Every segmentation S_l is coarser than the segmentation S_{l-1} below, s.t. for any segment $s_i \in S_{l-1}$ there exists exactly one segment $s_j \in S_l$ s.t. $s_i \subseteq s_j$. Note that $\mathcal{S}$ also implicitly contains the image data.

Label set $\mathcal{L}$ denotes a pre-defined set of numeric labels with associated object classes.

Labeling $y = (\ldots, y_i, \ldots)$ of a segmentation hierarchy $\mathcal{S}$ associates to each segment $s_i \in \mathcal{S}$ a label $y_i \in \mathcal{L}$. The ground-truth labeling is denoted by y^*.

Possible labelings $\mathcal{Y}(\mathcal{S})$ of a segmentation hierarchy $\mathcal{S}$ is a set containing all labelings of the same cardinality as $\mathcal{S}$, i.e. $\mathcal{Y}(\mathcal{S}) = \{y | y \in \mathcal{L}^{|\mathcal{S}|}\}$.

1.4 Methodological approach

We propose the I^{2}A (Interactive Image Annotation) Model, which is based on a CRF and consists of different components, where various alternatives and parameters are evaluated. The following chapter discusses the CRF and reasons for it's application in this scenario. A qualitative comparison of different structured prediction methods is beyond the scope of this work, though.

The evaluated components and variants contain different features, quality measures, baselines and label sets as well as the option to include hard constraints based on a pre-defined hierarchy of object classes.

The results show that the proposed framework significantly improves the classification rate over the number of user interactions, i.e. the structured model supports the interactive framework for the task of labeling image segmentation hierarchies.

1.5 Original Contribution

To the best of our knowledge this work presents the first approach to interactive semantic labeling of a segmentation hierarchy, using a CRF-based statistical model and a hierarchy of semantic labels. Thus we incorporate methods solving interactive labeling and object-class image segmentation in the context of segmentation hierarchies and a structured label space.

As part of this work, results were published at the joint symposium of the Deutsche Arbeitsgemeinschaft für Mustererkennung (DAGM) and Österreichische Arbeitsgemeinschaft für Mustererkennung (ÖAGM) [28]. In addition, the dataset used in this thesis and evaluation results are available on the authors website[2].

1.6 Structure of this thesis

In Chapter 1 we illustrate the problem addressed in this work and present a rough overview of the chosen approach. Chapter 2 discusses methods related to our approach and their differences. A short summary of CRFs and a more detailed explanation of our approach can be found in Chapter 3. Chapter 4 contains experiments, results and discussion about alternative framework components as well as a comparison to 3 different baselines. Chapter 5 concludes this work by summarizing the proposed framework and discussing open issues as well as further research possibilities.

[2] `http://www.publik.at/gzankl/`.

CHAPTER 2

Related Work

In the following, we describe related methods of object-class image segmentation and interactive labeling, and focus on the differences between the respective method and the interactive annotation framework presented in this work.

2.1 Object-Class Image Segmentation

Object-class image segmentation is the task of predicting object classes per pixel of an image on a single layer of abstraction. This kind of task is related to the problem we are concerned with, but is inherently different, since it is not defined for multiple layers. Several approaches exist that build on CRF models to solve this problem in a non-interactive manner.

Note that several methods solving this task use segmentation hierarchies in order to retrieve more robust segmentations. However, none of them uses a label space with non-trivial object-part relationships (an example is shown in Fig. 1.1).

Nowozin et al. [20] present a method for automatic semantic segmentation on a single level of abstraction. They use a CRF-based model with a dependency graph built from a segmentation hierarchy and instead of computing the potentials from classifier responses, like it is done in this work, they use a high dimensional parameter vector to learn a mapping from the feature vectors to the potentials.

McAuley et al. [18] use a graphical model based on a regular image pyramid. They utilize a simple hierarchy of semantic labels containing the ground-truth labels as well as a generic "multiple" (containing more than one ground-truth classes) and "background" (containing no ground-truth class) labels. They enforce these semantic dependencies with hard constraints in the model. Because of the simple hierarchy of semantic labels, the output of their method still represents a single level of abstraction. The I^{2}A Model uses a similar but more detailed hierarchy of semantic labels: a set of labels with object-part relationship - essentially giving a semantic label to each segment that would otherwise be labeled "multiple".

Lempitsky et al. [15] propose a "Pylon Model" that uses a segmentation hierarchy to compute figure-ground segmentations for a single class. This method can be extended to multiple classes, but a hierarchy of semantic labels is not considered.

Ladicky et al. [14] present a P^N based CRF that uses higher-order relationships. Rather than using a segmentation hierarchy, they compute single image segmentations and use the model to capture relationships between pixels and segments as well as the relationships between segments.

Gonfaus et al. [9] introduce the "Harmony Potential", a generalization of Ladicky's work that penalizes labels that do not match a global random variable of the model, while Ladicky's work also penalizes the case where the global random variable does not match the local labels. This way the method allows for a more relaxed labeling of segments.

Plath et al. [24] propose a method that constructs a hierarchical segmentation without the restriction of an inclusion relationship between parent and child. The prediction is performed on the whole hierarchy, like in our model, but only the labels on the finest segmentation are used for the result.

2.2 Interactive Labeling

Interactive structured prediction is a less common, but very important task in the context of this work.

Branson et al. [3] use interactive training for deformable part models. The proposed interaction model is functionally equivalent to the adjustments of the energy function presented in this work. However, this model uses continuous labels (position of points) and performs online training using stochastic gradient descent after the user finished his data input.

Mensink et al. [19] propose a CRF-based model for learning annotation hierarchies of images as a whole. Here agglomerative clustering is used in order to create a hierarchy of clusters of semantic labels. This hierarchy is then directly used as a model, where in each cluster the prediction determines wether a label exists or not. Annotation is performed interactively by asking the user yes/no questions that determine the existence of labels with low confidence.

CHAPTER 3

Interactive Image Annotation Model

Standard regression and classification tasks are meant to predict the value of a single variable. Predicting multiple variables while considering their relation to each other is a structured prediction task [27, 21]. A Conditional Random Field (CRF) is a method of structured prediction that models the probability function $p(y|x)$, where y is a labeling and x is the data. A popular alternative is the Markov Random Field (MRF), which models the joint probability $p(x, y)$ instead.

For the task of predicting the labels in a segmentation hierarchy we use a CRF because it is a statistical framework that can solve the task by distinguishing between data space and label space, which allows training with less parameters or samples for the same level of generalizability and computes a theoretically optimal solution given the specific model and training set used [21].

3.1 The Probability Function

The I^{2}A Model is a CRF with the probability distribution over all possible labelings defined as:

$$p(y|\mathcal{S}, \theta) = \frac{1}{Z(\mathcal{S}, \theta)} e^{-E(y,\mathcal{S},\theta)} \tag{3.1}$$

with $\theta = (\theta_u, \theta_p)$ being a learned parameter vector containing the unary θ_u and pairwise parameters θ_p, $y = (\dots, y_i, \dots)$ being a labeling of all segments $s_i \in \mathcal{S}$, and

$$Z(\mathcal{S}, \theta) = \sum_{y \in \mathcal{Y}(\mathcal{S})} e^{-E(y,\mathcal{S},\theta)} \tag{3.2}$$

being the partition function (used to normalize the term to retrieve a probability function). $\mathcal{Y}(\mathcal{S})$ is the set of all possible labelings of the segments in $\mathcal{S}$. The probability function $p(y|\mathcal{S}, \theta)$ is of the family of exponential distributions. More specifically we define E to be linear in θ, meaning that we have a log-linear model. Thus training our model is a convex optimization problem (every local extrema is a global extrema).

We define the energy function E for a set of segments $\mathcal{Q} \subseteq \mathcal{S}$ and corresponding labels y as:

$$\begin{aligned} E(y, \mathcal{Q}, \theta) &= \sum_{s_i \in \mathcal{Q}} \Phi(s_i, y_i, \theta_u) + \sum_{s_i \in \mathcal{Q}} \sum_{s_j \in \mathcal{Q} \cap m_{\mathcal{S}}(s_i)} \Psi(y_i, y_j, \theta_p) & (3.3) \\ \Phi(s_i, y_i, \theta_u) &= \theta_u(y_i)^\top \phi(s_i, y_i) & (3.4) \\ \Psi(y_i, y_j, \theta_p) &= \theta_p(y_i, y_j)^\top \psi(y_i, y_j) & (3.5) \end{aligned}$$

where ϕ is a unary meta-feature obtained as output of a classifier for each segment s_i and label y_i, ψ is a feature computed using the co-occurrence of the ground-truth labels and $m_{\mathcal{S}}$ is the parent-child mapping for $\mathcal{S}$. ψ does not have to be symmetric $\psi(y_i, y_j) \neq \psi(y_j, y_i)$. Thus, the features and parameters encode which labels correspond to the parent and child segments, respectively. This allows the framework to distinguish between child and parent when computing the pairwise potentials, e.g. child *wheel* and the parent *car* will have a higher score than labeling the child *car* and the parent *wheel.*

3.2 Inference and the Feedback Loop

The workflow for labeling an image is as follows: The user provides an image and the corresponding segmentation hierarchy $\mathcal{S}$ and receives an initial prediction of the labels of all segments. An interactive process (feedback loop) follows that alternates between the user selecting one or more segments and specifying their label(s), and the system providing a new prediction based on this input and providing visual feedback to the user. Labels provided by the user are considered correct and will not be predicted anymore. In the following we discuss the case of the user selecting and labeling a single segment per iteration, without the loss of generality.

We perform inference by selecting the maximum a-posteriori configuration y^* (MAP inference). For the initial ($\mathcal{Q} = \mathcal{S}$), fully automatic prediction y^* is given by solving

$$y^* = \operatorname*{argmax}_{y \in \mathcal{Y}(\mathcal{S})} \{p(y|\mathcal{S}, \theta)\} = \operatorname*{argmin}_{y \in \mathcal{Y}(\mathcal{S})} \{E(y, \mathcal{S}, \theta)\}. \tag{3.6}$$

For two disjoint subsets $\mathcal{S}^{(v)}, \mathcal{S}^{(f)} \subseteq \mathcal{S}$ with $\mathcal{S}^{(v)} \cup \mathcal{S}^{(f)} = \mathcal{S}$ we can decompose the energy function $E(y, \mathcal{S}, \theta)$ in Eq. 3.3 as:

$$\begin{aligned} E(y, \mathcal{S}, \theta) =& E(y^{(v)}, \mathcal{S}^{(v)}, \theta) + E(y^{(f)}, \mathcal{S}^{(f)}, \theta) + \\ &+ \sum_{s_i \in \mathcal{S}^{(v)}} \sum_{s_j \in \mathcal{S}^{(f)} \cap m_{\mathcal{S}}(s_i)} \Psi(y_i^{(v)}, y_j^{(f)}, \theta) + \sum_{s_i \in \mathcal{S}^{(f)}} \sum_{s_j \in \mathcal{S}^{(v)} \cap m_{\mathcal{S}}(s_i)} \Psi(y_i^{(f)}, y_j^{(v)}, \theta), \end{aligned} \tag{3.7}$$

with $y^{(v)}, y^{(f)}$ representing the restrictions of y to the labels corresponding to segments in $\mathcal{S}^{(v)}, \mathcal{S}^{(f)}$, respectively. Note that the superscript (v) stands for variable and (f) for fixed (or user-specified) variables, i.e. $y^{(f)}$ is constant. We start with $\mathcal{S}^{(v)} = \mathcal{S}$ and $\mathcal{S}^{(f)} = \emptyset$ and model user interaction by moving the manually labeled segment s_k from $\mathcal{S}^{(v)}$ to $\mathcal{S}^{(f)}$ as well as adding the given corresponding label to $y^{(f)}$. Then, inference is performed only over the labels in $y^{(v)}$ corresponding to the segments in $\mathcal{S}^{(v)}$:

$$y^{(v)*} = \underset{y^{(v)} \in \mathcal{Y}(\mathcal{S}^{(v)})}{\operatorname{argmin}} \left\{ \sum_{s_i \in \mathcal{S}^{(v)}} \Phi'(s_i, y_i^{(v)}, \theta) + \sum_{s_i \in \mathcal{S}^{(v)}} \sum_{s_j \in \mathcal{S}^{(v)} \cap m_{\mathcal{S}}(s_i)} \Psi(y_i^{(v)}, y_j^{(v)}, \theta_p) \right\}, \quad (3.8)$$

where Φ', the modified unary potential, is defined as:

$$\Phi'(s_i, y_i^{(v)}, \theta) = \Phi(s_i, y_i^{(v)}, \theta_u) + \sum_{s_j \in \mathcal{S}^{(f)} \cap m_{\mathcal{S}}(s_i)} \Psi(y_i^{(v)}, y_j^{(f)}, \theta_p) + \sum_{s_j \in \mathcal{S}^{(f)} \cap m'_{\mathcal{S}}(s_i)} \Psi(y_j^{(f)}, y_i^{(v)}, \theta_p)$$

$$\text{with } m'(s_i) = \begin{cases} \{s_j\} & \text{iff } s_j \in m(s_i) \\ \emptyset & \text{otherwise} \end{cases} \quad (3.9)$$

Eq. 3.8 represents the inference function $f(\mathcal{S}, \mathcal{U}, \mathcal{L})$ from Eq. 1.1, where the user input $\mathcal{U}$ contains the user selected segments $\mathcal{S}^{(f)}$ and the specified labels $y^{(f)}$. The set of segments that have to be inferred is $\mathcal{S}^{(v)} = \mathcal{S} \setminus \mathcal{S}^{(f)}$ and the inferred labeling y is a combination of $y^{(v)*}$ and $y^{(f)}$.

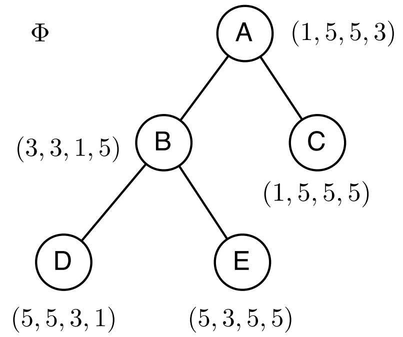
It can be seen in Eq. 3.7 that the pairwise interaction between segments $s_i \in \mathcal{S}^{(v)}$ and $s_j \in \mathcal{S}^{(f)}$ only depend on $y_i^{(v)}$, since $y_j^{(f)}$ is constant during inference. Thus the pairwise terms can be included in the unary potential (shown in Eq. 3.9). User-provided labels change the MAP configuration and other misclassified segments may be corrected automatically if the features are well suited for the corresponding image segments, i.e. ϕ has a high response for non ground-truth labels of these segments.

The Markov Property for MRFs states that a random variable is independent of all other variables, given its neighbors. In our model, the neighbors of a variable y_k consist of the corresponding segment s_k, the labels of child segments y_i with $s_i \in m_{\mathcal{S}}(s_k)$ and the label of the parent y_j with $s_j \in m'_{\mathcal{S}}(s_k)$. The Markov Property also holds for CRFs and in our case means that the labels of children and parents of a segment s_k are conditionally independent of each other, given y_k and their respective segments s_i. Thus, during inference each user selected segment s_k and the specified label $y_k^{(f)}$ splits the graphical model at each s_k. The resulting graphical model is a forest where inference can be done independently for each tree. We perform exact inference efficiently using the Belief Propagation (BP) algorithm [23].

Example Inference

Fig. 3.1 shows a constructed example of potentials of a small hierarchy with 5 segments. Inference is performed by minimizing the energy function. Tbl. 3.1 shows several labelings and the corresponding value of the energy function. The values of the potential functions Φ and Ψ in this example are artificial, the computation will be explained in Sec. 3.4 and the respective features and experiments will be presented in Sec. 4.4.

Let $y = (Car, Wheel, Wheel, Rim, Tire)$ be the correct labeling (the order given by the name of the node: A to E). An inference algorithm will choose $y = (Wheel, Rim, Rim, Rim)$



		parent			
	Ψ	*Wheel*	*Tire*	*Rim*	*Car*
child	*Wheel*	3	5	5	1
	Tire	2	1	5	5
	Rim	1	5	1	5
	Car	5	5	5	1

Figure 3.1: Example of unary (Φ, to the left) and pairwise (Ψ, to the right) potentials for a label set $\mathcal{L} = (Wheel, Tire, Rim, Car)$. Unary potentials are represented by a 4-tuple per node, containing the potential for each label in the order given by $\mathcal{L}$.

y_A	y_B	y_C	y_D	y_E	Energy	Classification Rate
Wheel	*Rim*	*Wheel*	*Car*	*Tire*	15	0.40
Car	*Wheel*	*Wheel*	*Rim*	*Tire*	13	1.00
Wheel	*Rim*	*Wheel*	*Rim*	*Rim*	14	0.40

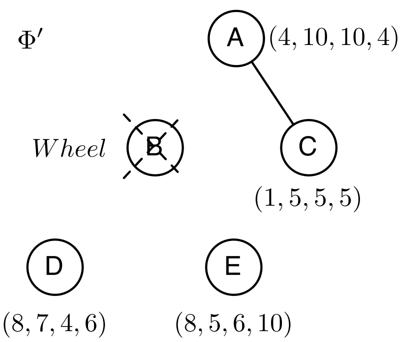
Table 3.1: Example labelings y and the respective values of the energy function based on the potentials from Fig. 3.1. The last column shows the classification rate assuming the correct labeling is $y = (Car, Wheel, Wheel, Rim, Tire)$.

over the correct assignment, since its energy value is lower. Fig. 3.2 shows what happens if the user selects node B and sets its label to *Wheel*. All pairwise potentials of this node only depend on the respective neighbor, thus they are added to the modified potential function of children and parent.

By incorporating the user interaction we split the model into a forest where inference can be done independently for each tree, e.g. we can immediately infer the most likely label of node E to be *Tire*, since it is now conditionally independent of all other nodes in the model and we can simply minimize the modified potential function. In fact this example is set up, so that this single user interaction makes the correct labeling also the most likely labeling, as the model has minimal energy for the correct labeling, as long as it is condition by $y_B = Wheel$.

3.3 Parameter Estimation

Given a set of N training images, the corresponding hierarchical segmentations $\{\mathcal{S}_k\}$ and ground-truth labelings $\{y^{(k)}\}$, for the k-th image, we perform training by assuming i.i.d. samples and maximizing the likelihood (the probability of the ground-truth, given our model):



		parent			
	Ψ	*Wheel*	*Tire*	*Rim*	*Car*
child	*Wheel*	3^*_+	5^*	5^*	1^*
	Tire	2_+	1	5	5
	Rim	1_+	5	1	5
	Car	5_+	5	5	1

Figure 3.2: Example of unary (Φ', to the left) and pairwise (Ψ, to the right) potentials for a label set $\mathcal{L} = (Wheel, Tire, Rim, Car)$. Here the label of node B is specified by the human user as *Wheel*. Pairwise potentials related to the selected node are added to the unary potentials of adjacent nodes (numbers marked with '$*$' are added to the potentials parents of B, while numbers marked with '$+$' are added to the potentials children of B). The graph also represents the new model that is used for inference, where B and incident edges are removed.

$$\begin{aligned} \theta^* &= \operatorname*{argmax}_{\theta} \prod_{k=1}^{N} p(y^{(k)} | \mathcal{S}_k, \theta) && (3.10)\\ &= \operatorname*{argmin}_{\theta} \sum_{k=1}^{N} \Big[E(y^{(k)}, \mathcal{S}_k, \theta) + \ln Z(\mathcal{S}_k, \theta) \Big] && (3.11)\\ &= \operatorname*{argmin}_{\theta} \sum_{k=1}^{N} L_k(\theta_u, \theta_p) && (3.12) \end{aligned}$$

We solve this optimization efficiently by using a second-order gradient descent. The loss L_k for each sample k is differentiated with respect to the unary parameters θ_u:

$$
\begin{aligned}
\frac{\delta}{\delta\theta_u} L_k &= \frac{\delta}{\delta\theta_u} E - \frac{1}{Z}\frac{\delta}{\delta\theta_u} Z && (3.13)\\
&= \frac{\delta}{\delta\theta_u} E - \sum_{y\in\mathcal{Y}(\mathcal{S}_k)} \left[\frac{1}{Z} e^{-E(y,\mathcal{S}_k,\theta)} \frac{\delta}{\delta\theta_u} E\right] && (3.14)\\
&= \sum_{s_i\in\mathcal{S}_k} \phi(s_i, y_i^{(k)}) - \sum_{s_i\in\mathcal{S}_k} \sum_{y\in\mathcal{Y}(\mathcal{S}_k)} p(y|\mathcal{S}_k,\theta)\phi(s_i, y_i) && (3.15)\\
&= \sum_{s_i\in\mathcal{S}_k} \left[\phi(s_i, y_i^{(k)}) - \sum_{y_i\in\mathcal{L}} \sum_{\substack{y'\in\mathcal{Y}(\mathcal{S}_k)\\ y'_i=y_i}} p(y'|\mathcal{S}_k,\theta)\phi(s_i, y_i)\right] && (3.16)\\
&= \sum_{s_i\in\mathcal{S}_k} \left[\phi(s_i, y_i^{(k)}) - \sum_{y_i\in\mathcal{L}} p(y_i|s_i,\theta)\phi(s_i, y_i)\right], && (3.17)
\end{aligned}
$$

where $p(y_i|\mathcal{S}_k, \theta) = p(y_i|s_i, \theta)$ is the marginal belief for label y_i at segment s_i. Note that ϕ and ψ here are lowercase and do not depend on the parameter vector θ (see Eq. 3.4 and 3.5).

Analogously we differentiate with respect to θ_p:

$$
\begin{aligned}
\frac{\delta}{\delta\theta_p} L_k &= \frac{\delta}{\delta\theta_p} E - \sum_{y\in\mathcal{Y}(\mathcal{S}_k)} \left[\frac{1}{Z} e^{-E(y,\mathcal{S}_k,\theta)} \frac{\delta}{\delta\theta_p} E\right] && (3.18)\\
&= \sum_{s_i\in\mathcal{Q}} \sum_{s_j\in\mathcal{Q}\cap m(s_i)} \psi(y_i^{(k)}, y_j^{(k)}) - \sum_{s_i\in\mathcal{Q}} \sum_{s_j\in\mathcal{Q}\cap m(s_i)} \sum_{y\in\mathcal{Y}(\mathcal{S}_k)} p(y, \mathcal{S}_k, \theta)\psi(y_i, y_j) && (3.19)\\
&= \sum_{s_i\in\mathcal{Q}} \sum_{s_j\in\mathcal{Q}\cap m(s_i)} \left[\psi(y_i^{(k)}, y_j^{(k)}) - \sum_{(y_i,y_j)\in\mathcal{L}^2} p(y_i, y_j|s_i, s_j, \theta)\psi(y_i, y_j)\right], && (3.20)
\end{aligned}
$$

where $p(y_i, y_j|s_i, s_j, \theta)$ is the pairwise belief for labels y_i, y_j. The marginal and pairwise beliefs $p(y_i|s_i, \theta)$ and $p(y_i, y_j|s_i, s_j, \theta)$ are computed efficiently using the BP algorithm implemented by the UGM toolbox[1].

3.4 Potential Functions

We want the meta-features ϕ and ψ to have low responses for segment and label combinations that represent the ground-truth because we minimize the energy. For example a segment s_i containing a *wheel* may have the low values for $\phi(s_i, wheel)$ and high values for $\phi(s_i, street)$. Because of similar appearance $\phi(s_i, rim)$ and $\phi(s_i, tire)$ may also provide low responses, but they should be higher than $\phi(s_i, wheel)$.

[1] by Mark Schmidt 2011, http://www.di.ens.fr/ mschmidt/Software/UGM.html

Unary Potentials

The meta-feature ϕ is computed using the output of a classification or regression method based on feature vectors per segment of the image. For a linear model, this means that ϕ is a function of $w^\top v(s_i)$, with w being a trained weight vector and $v(s_i)$ the feature vector of segment s_i. The final function definition will be presented in the next chapter along with experiments on alternative feature vectors and classification or regression methods. Another way to use the feature vectors is to define $\phi(s_i) = v(s_i)$ and use the unary parameters θ_u as a matrix, such that the regression is implicitly done by the CRF framework [20]. However, this would increase the number of unary parameters to $|\theta_u| = |v||\mathcal{L}|$, creating a less generalizable model that needs more training images than our approach.

Pairwise Potentials

Similarly ψ is supposed to have low responses for meaningful parent-child label combinations. For example $\psi(wheel, rim)$ may have low values, because a rim is part of a $wheel$, while $\psi(rim, wheel)$ or $\psi(street, rim)$ have high responses, since these combinations are unlikely or unobserved (they do not occur in the training data).

It is possible to define $\psi(s_i, y_i, s_j, y_j)$ and use features of the segments s_i and s_j for these pairwise potentials. However, a common definition of pairwise interactions is $\psi(y_i, y_j)$. For example the Potts Model defined as $\psi(y_i, y_j) = 1 - 1_{y_i = y_j}$, with 1_B being the indicator function, which is 1 if B is true and 0 otherwise. Thus the model encourages neighbors in the graphical model to have the same label.

Rather than encouraging the same labels for neighbors, we want to encourage parent and child labels, that are consistent with the observed ground-truth. Thus we define $\psi(y_i, y_j)$ as a $|\mathcal{L}| \times |\mathcal{L}|$ matrix and automatically compute it from the training data. The following chapter provides more details on how ϕ and ψ are computed.

CHAPTER 4

Evaluation and Component Comparison

In this chapter we present how the framework is evaluated followed by several experiments with alternative components and evaluation setups and discuss the results.

4.1 Datasets

The experiments are performed on a generated ground-truth based on the Stanford Background dataset [10] as well as the CamVid dataset [4, 5]. Both datasets provide ground-truth object class segmentations on a single level of coarseness. To evaluate the proposed framework it is necessary to create a hierarchy of semantic labels as well as segmentation hierarchies for the dataset.

Several videos from a camera mounted to a moving car are contained in the CamVid dataset[1] [4, 5]. For a total of 701 images from these videos there exist manually labeled segmentations (every 30th frame of the videos) with 32 classes representing objects and backgrounds commonly occurring in street scenes. There is no stationary background and motion information is not used for further computation.

The Stanford Background dataset[2] [10] consists of 715 single images and two ground-truths with different label sets: the geometric classes {*sky, horizontal, vertical*} and the semantic classes {*sky, tree, road, grass, water, building, mountain, foreground object*}.

The samples of each dataset were divided into training set (3/4 of the images) and test set (1/4 of the images). The training set is further divided for the feature evaluation in order to provide short training and test times, s.t. the parameter space can be thoroughly investigated.

[1] http://mi.eng.cam.ac.uk/research/projects/VideoRec/CamVid/.

[2] http://dags.stanford.edu/projects/scenedataset.html.

4.2 Experiments Setup

A desired property of the evaluation is that the influence of the segmentation quality is minimal, so that only the proposed method is evaluated. It is not the goal of this thesis to evaluate segmentation methods [29]. We compute hierarchical segmentations $\mathcal{S}_k$ for all images and consider them to be part of the ground-truth.

The semantic labels of the datasets (further called base labels L_B) only capture one level of abstraction and the straight-forward solution to this is to use all label combinations $\mathcal{L} = \mathcal{P}(L_B) \setminus \emptyset$. We start with $|L_B| = 3$ (the smallest set of base labels with label combinations that contain more than one, but not all of the base labels) and perform experiments with bigger label sets $\mathcal{L}$. To complete a ground-truth that can be used by the proposed framework it is necessary to assign a label $y_i^* \in \mathcal{L}$ to each segment $s_i \in \mathcal{S}_k$ of all training samples $\mathcal{S}_k$ by determining the occurrence of object classes of the dataset inside the segments of the hierarchy.

We compute the segmentation hierarchy $\mathcal{S}$ of an image, using the minimum spanning tree-based pyramid [11][3]. The dataset provides a flat ground-truth segmentation $\mathcal{S}^{(gt)}$ and a corresponding ground-truth labeling with object classes from L_B for the image. Ground-truth labelings y^* of the segmentation hierarchy are obtained by determining the occurrence of object classes of $\mathcal{S}^{(gt)}$ inside the segments of the hierarchy. This is done by computing the overlap of $s_i \in \mathcal{S}$ with segments from $\mathcal{S}^{(gt)}$. Is the overlap with segments of a class bigger than $\frac{1}{|L_B|}$ (in our case 1/3), then that class is added to the label combination y_i^* of the segment. The threshold $\frac{1}{|L_B|}$ is chosen so that it is minimal without allowing segments to have no class.

The quality measures used are the number of corrected regions $|\{y^n = y^*\}| - |\{y^0 = y^*\}|$ and the classification rate

$$q_1 = \frac{|\{y^n = y^*\}|}{|\mathcal{S}|}, \tag{4.1}$$

with y^n denoting the inferred labeling after n user interactions. The evaluation is performed for 100 simulated user interactions using a top-down strategy of correcting segments, i.e. in each iteration a virtual annotator selects the top-most misclassified segment (starting at the coarsest level of the segmentation hierarchy) and assigns the correct label. The measurements are then averaged over the test set. For the feature evaluation, which does not consider user interaction, the classification rate over all images is used.

4.3 Framework Evaluation

For the full evaluation the hierarchy of semantic labels is created by using $|L_B| = 3$ base labels and all label combinations $\mathcal{L} = \mathcal{P}(L_B) \setminus \emptyset$. For the Stanford dataset L_B represents the geometric classes and for the CamVid database L_B consists of classes representing roads, cars and background (everything else).

The meta-features ϕ and ψ discussed in Sec. 3.4 are fixed in this experiment. Here ϕ is a function of the Mahalanobis distance of the feature vectors to each class, with the feature vectors being computed using the Geometric Context framework by Hoiem et al. [12]. Details on the best of the evaluated functions can be found in the next section.

[3] Although any hierarchical segmentation method, e.g. [1], can be used.

For our baseline we use an initial prediction without pairwise interactions, i.e. using an energy function without the second term in Eq. 3.3, followed by a simulation of repeated manual correction of labels (with a top-down strategy as well). Thus the initial classification rate y^0 is slightly lower than for the full framework.

The baseline is outperformed by our interactive system as seen in Fig. 4.1a. Because misclassified segments are corrected independently in the baseline, it has a constant quality-slope. We can reach an average precision of 80% after a single user interaction on the CamVid dataset, compared to 13 for the baseline. On the Stanford Background dataset we need 9 user interactions for the full framework and 63 for the baseline to reach the same precision. Fig. 4.1b shows that after 20 user interactions, while the baseline corrected 20 segment labels, the interactive system corrected an average of 122 labels on the Stanford and 118 labels on the CamVid dataset. A lower number of misclassified regions also means that less regions can be corrected with a single interaction. Thus all curves converge to the point where no misclassified segments are left.

Comparison of different Baselines

Alternative baselines that can be used are fully manual approaches. The most basic approach is to label each segment independently. It provides the same improvement rate $\frac{\Delta q_1}{\Delta n}$ as the baseline, but starts at 0 correctly classified regions. This can be further improved by starting with a default label (e.g. the most common label in the training set). However, the initial prediction still has a higher classification rate as this trivial labeling. A more efficient strategy is to ask the user to label in a top-down order, only the segments that do not have the same label as their parent. Fig. 4.2a and 4.2b show a qualitative comparison of the baseline and these two alternatives. While the top-down method has the highest improvement rate, the classification rates during the first 40 user interactions are significantly lower, which is also the interval at which the most improvement of the interactive method comes into play.

Another alternative is to start with a single prediction step of the full framework (using the pairwise term in Eq. 3.3), followed by fully manual corrections. This variant has the same initial classification rate as the full interactive framework, but the slope of the proposed baseline with its independent classification of segments. We decided on the independent classification of segments because it is a more straight-forward approach that still works in the feedback loop of the proposed system. Also by comparing the framework to independent classification we can show the advantages of structured prediction in an interactive setting.

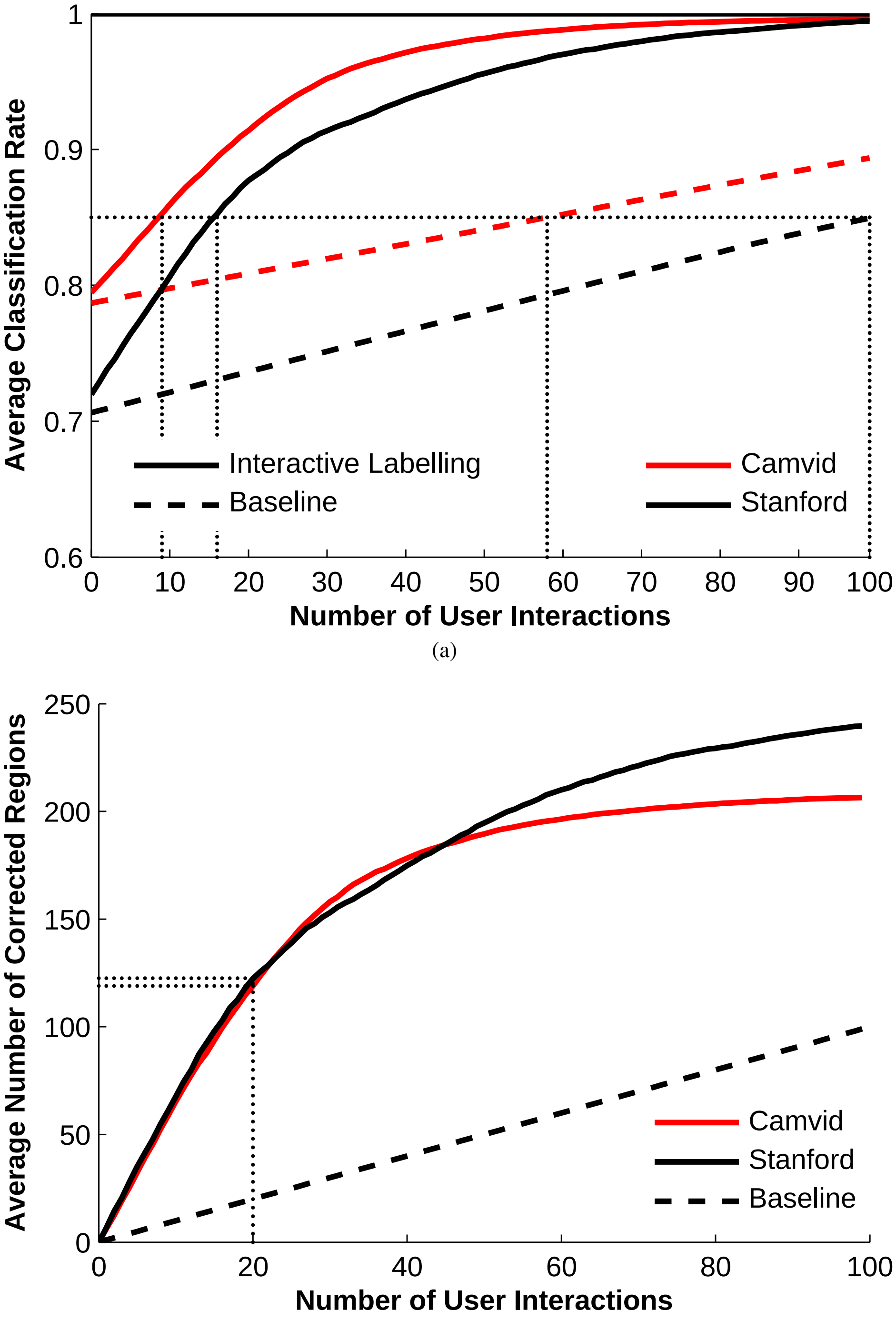
All of the proposed manual approaches require significantly lower average classification rates than the proposed baseline as well as the interactive method.

Comparison of Evaluation and Simulation Methods

It was taken into consideration to evaluate the framework based on custom quality measures that favor specific segments. Two alternatives were evaluated:

- The normalized sum of the area of correctly classified segments, more formally

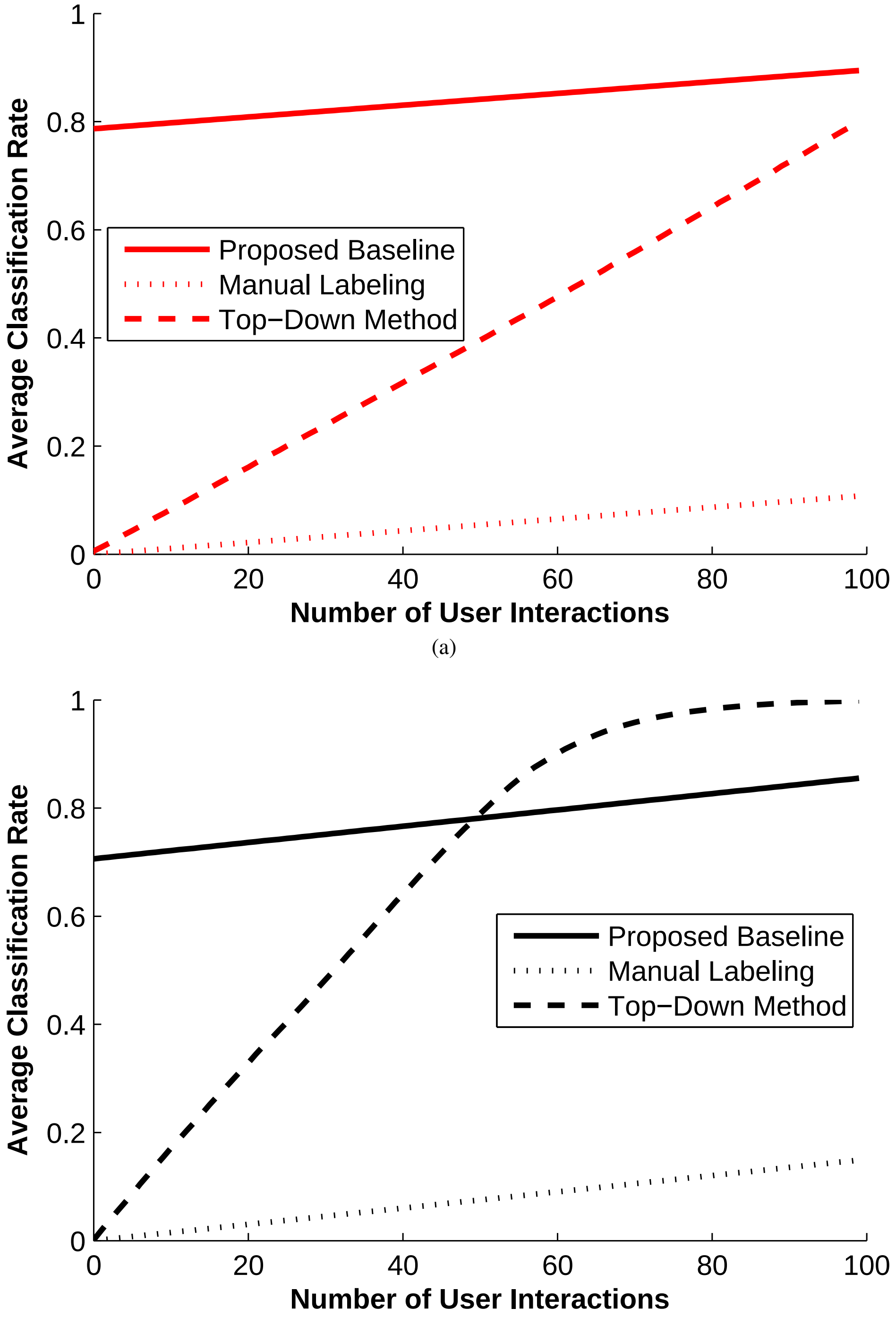
$$q_2 = \frac{1}{\sum_{s_i \in \mathcal{S}} A(s_i)} \sum_{s_i \in \mathcal{S}} A(s_i) 1_{y_i = y_i^*}, \tag{4.2}$$



(a)

(b)

Figure 4.1: (a) Main evaluation showing average classification rates of the interactive framework and the baseline on the Camvid and Stanford Background datasets; (b) illustration of the average number of corrected regions per user interaction. Note that the curves in the bottom graph converge to different maxima, so a comparison between the datasets has to be performed in the top graph. The evaluation is done with a total of 7 labels for each dataset.



(a)

(b)

Figure 4.2: Average classification rates of the proposed baseline compared to top-down and fully manual approaches, evaluated on the (a) CamVid and (b) Stanford Background datasets.

where $A(s_i)$ is the area of segment s_i. The indicator function 1_B equals 1 if B is true and 0 otherwise.

- Additionally this can be weighted by a function of the number of base labels $|y_i|$ of each segment s_i:
$$q_3 = \frac{1}{\sum_{s_i \in \mathcal{S}} A(s_i) e^{-|y_i|}} \sum_{s_i \in \mathcal{S}} A(s_i) e^{-|y_i|} 1_{y_i = y_i^*}, \tag{4.3}$$
This represents the desire to have specific labels (few base labels) in larger regions too, because for q_2 the most important segment would be the whole image, which has one of the most generic labels (e.g. the label “image”, containing all object classes).

Both alternatives favor specific strategies to select and correct nodes. For q_2 one would want to always choose the biggest misclassified segment in order to maximize the slope in the quality-user interactions graph. The same approach can be used to maximize the slope for q_3 by choosing the segment with the highest score $A(s_i)e^{-|y_i|}$.

More complex quality measures and node selection strategies defeat the purpose of the evaluation, because the user may or may not follow such strategies. It would require more elaborate experiments to find a quality measure that is adapted to human perception.

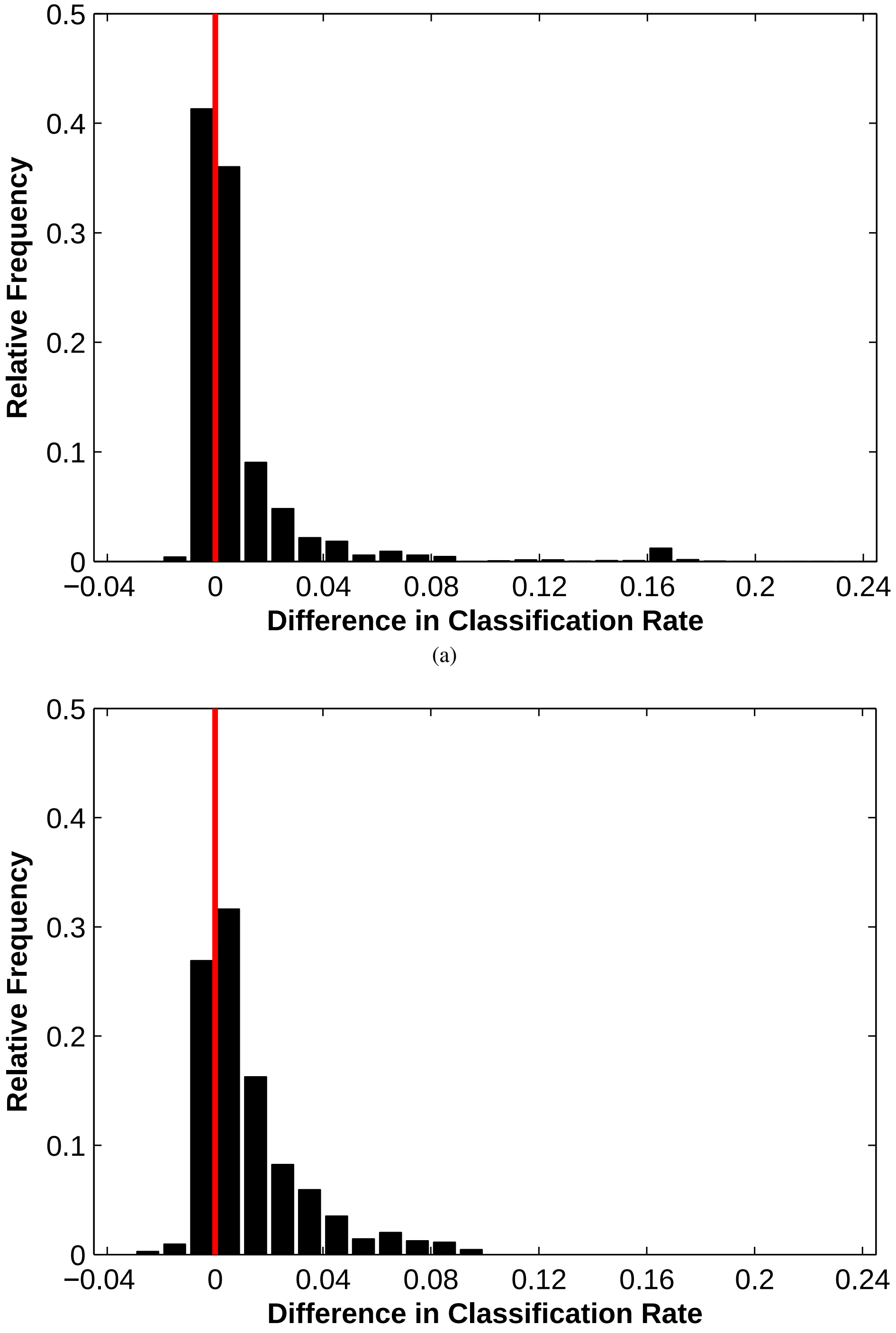
Also note that quality is subjective and there is no single perfect measurement to capture the quality of the labeling, since it depends on the application and the segments the user is interested in.

Hard Constraints

The asymmetric feature function ψ implicitly encodes the ordering of semantic labels that represents the aggregation of objects from parts. The labels of the descendants of a segment labeled by the user can be constrained using the hierarchy of semantic labels, e.g. when a segment is labeled *human*, its descendants labels should only correspond to body parts.

While ψ implicitly encodes this ordering, it only represents a “soft-constraint” that makes a parent-child combination of *human-wheel* less likely, but does not enforce the children of *human* to be body-parts. Using the human interventions, since the human-labeled segments are considered ground-truth, we can incrementally include hard-constraints on subtrees of the segmentation hierarchy. This marginally improves the classification rates.

Fig. 4.3a and 4.3b show the distribution of differences in classification rates on the CamVid and Stanford dataset, respectively. These illustrations represent histograms of the signed difference between the method that uses these hard constraints and the originally proposed method. Positive values mean that using the hard constraints provide higher classification rates. Note that the computation is done per image, not based on the average classification rates. Both graphs show that using these hard constraints, the classification rates are higher for most images. And the absolute value of differences on the positive side tends to be larger, indicating that it is more likely that hard constraints improve the classification rates. When they do, the absolute difference between classification rates with and without hard constraints is higher than in the case of quality decline. Because this only affects the feedback loop and there is no additional train-



(a)

(b)

Figure 4.3: Histogram of the signed difference in classification rates between the using hard constraints and the default proposed approach, evaluated on the (a) CamVid and (b) Stanford Background datasets.

ing necessary, it is possible for the user to decide to (de-)activate these hard-constraints during runtime, for the next prediction in the feedback loop.

4.4 Feature Alternatives

The unary and pairwise potential functions are evaluated based on different features and different mappings. In the following, we discuss options, parameters and results of these evaluations.

Features for Unary Potentials

Three steps are necessary in order to find the unary potential function $\phi(s_i, y_i)$, each introducing new parameters:

Step 1: Base Features

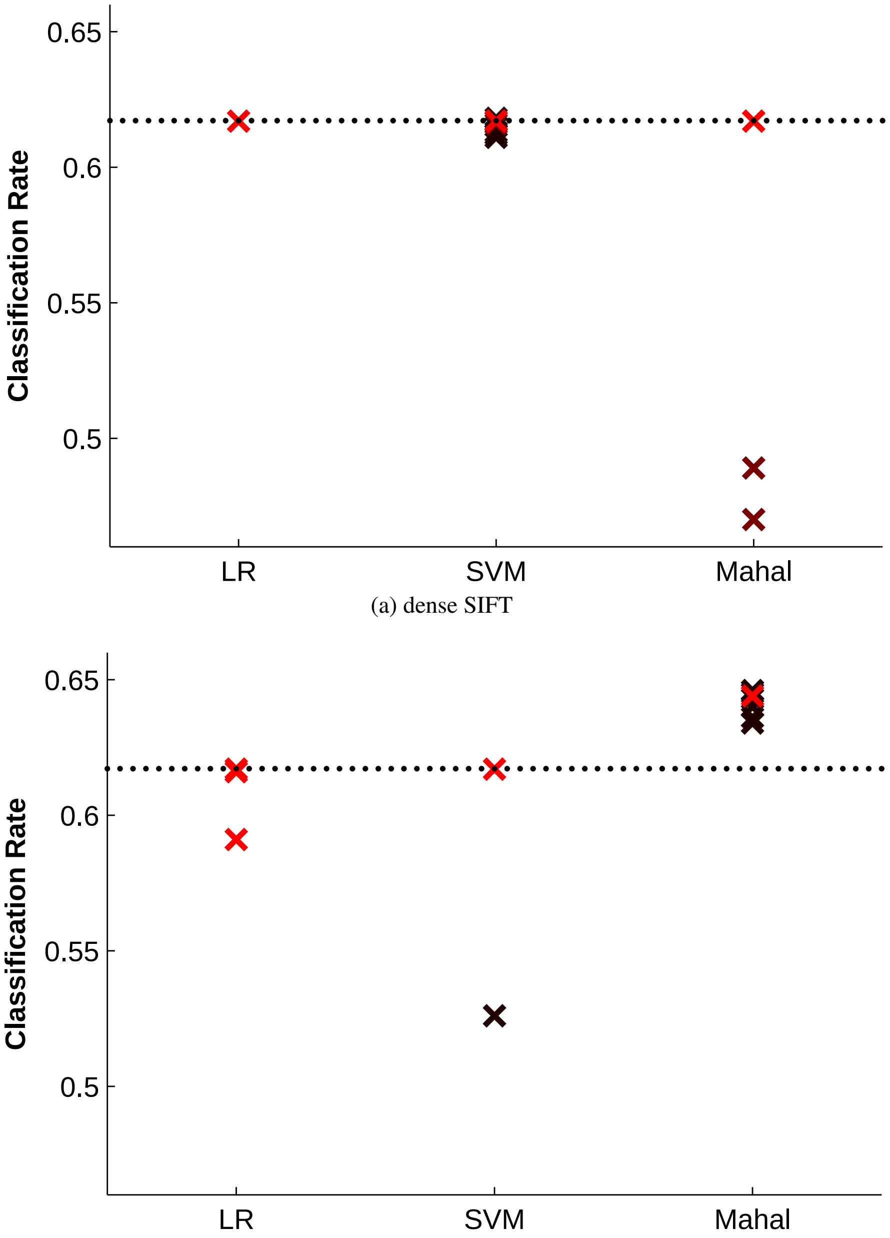
The features evaluated are the Bag of Words [26] histogram of dense SIFT [16] (dSIFT) features sampled every 6 pixels using 20 code words and the region features from the Geometric Context (GC) method [12]. Sampling parameters are determined on the validation set with a simple k-NN classifier, maximizing the classification rate. The GC features are computed per image segment and contain several color, texture, shape, spatial and geometric features.

Using the *randFeat* framework[4], the transformation of features into the Hilbert space of skewed χ^2 and intersection kernels was approximated, using different kernel parameters. Only a few Experiments with these transformations were performed, because the dimension of the transformed features is set to 2000, which only slightly improves results at a high computational cost. The tested parameters (c, d) for the Stanford dataset with Linear Regression (LR) are $\{(1, 200), (0.5, 200\}, (0.01, 200), (1, 1000), (1, 2000)$, with c being the kernel parameter and d being the dimension of the transformed features. In the case of the SVM, only $c = 1$ with $d = 200$ was tested, since high dimensional SVM training takes more time than LR training.

Step 2: Label-dependent Features

Because we need potential functions providing values for each region and label, an intermediate step provides label dependent features. One of the tested methods for this step is a linear regression to the overlap of segment s_i with image regions of the original ground-truth that represent the label combination y_i. Another method is to use the signed distance of the test feature to the hyperplane of a linear Support Vector Machine (SVM) for each class (which is positive if the point is classified as a negative sample, in order to get low values if the points are similar). The last method uses the Mahalanobis distance of the feature vector of s_i towards class y_i. The covariance matrices are computed independently for each class and Principal Component Analyisis (PCA) is used in order to prevent them from being becoming singular. We remove all dimensions with eigenvalues $\lambda_i < \xi \max_j \lambda_j$, introducing the new parameter ξ.

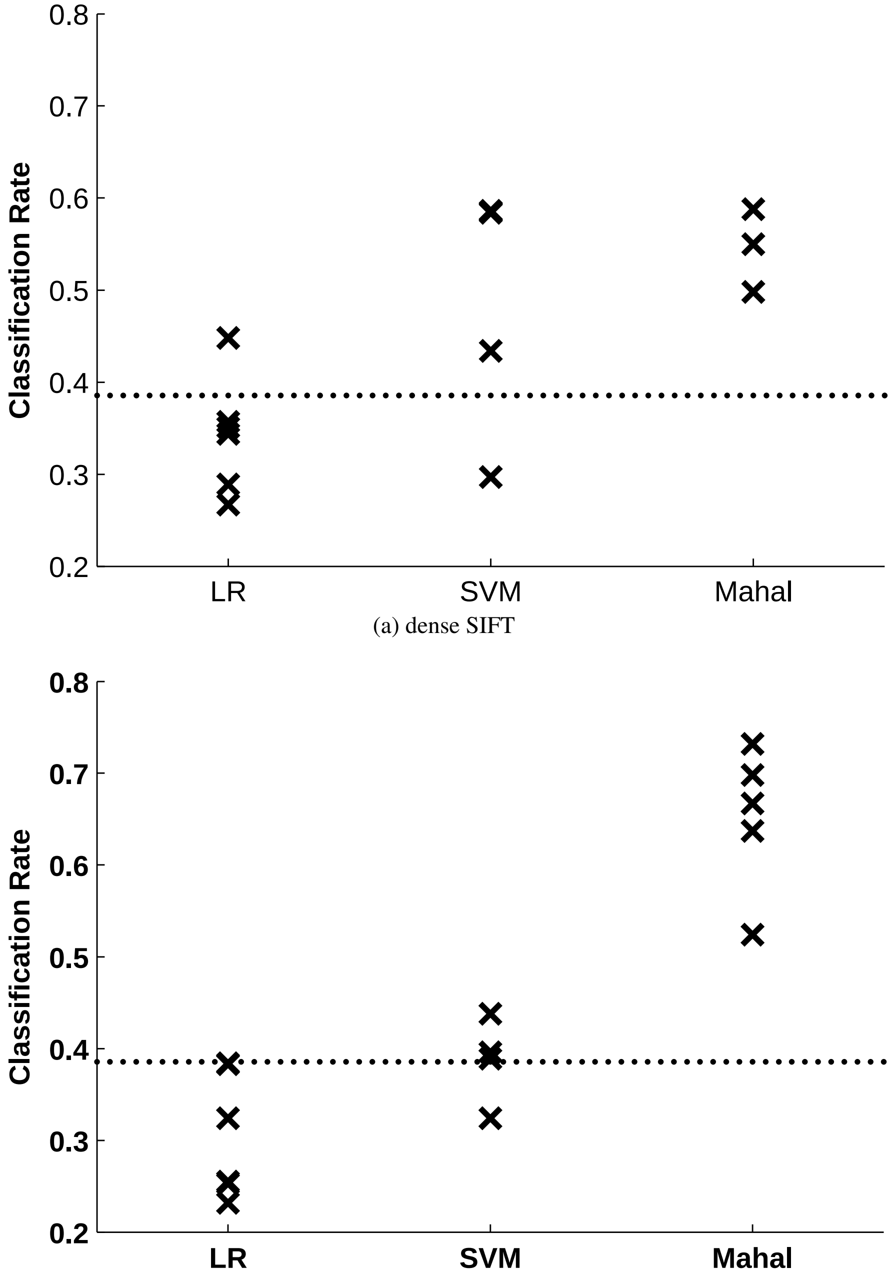
[4] `http://sminchisescu.ins.uni-bonn.de/code/randfeat/`



(a) dense SIFT

(b) Geometric Context

Figure 4.4: Feature evaluation on the CamVid dataset, based on (a) Bag of Words histogram of dense SIFT features and (b) Geometric Context features by Hoiem et al. [12]. The evaluated potential functions are based on (LR) a linear regression to the overlap of image regions; (SVM) the distance to the hyperplane of an SVM per class; (Mahal) the Mahalanobis distance. Brighter symbols represent multiple occurrence of the same classification rate. Dotted lines represent a trivial classifier that always chooses the class that most frequently occurs in the training set.



(a) dense SIFT

(b) Geometric Context

Figure 4.5: Feature evaluation on the Stanford Background dataset, based on (a) Bag of Words histogram of dense SIFT features and (b) Geometric Context features by Hoiem et al. [12]. The evaluated potential functions are based on (LR) a linear regression to the overlap of image regions; (SVM) the distance to the hyperplane of an SVM per class; (Mahal) the Mahalanobis distance. Dotted lines represent a trivial classifier that always chooses the class that most frequently occurs in the training set.

Step 3: Mapping Function

The final step consists of a mapping function, which is supposed to be closer to 0 the better the segment fits to the specified object class. This not necessary for the linear regression. For the other two options (SVM, Mahalanobis distance) we use the exponential function and introduce a scaling parameter δ:

$$\phi(s_i, y_i) = e^{-\delta d(s_i, y_i)}, \tag{4.4}$$

where $d(s_i, y_i)$ is the label-dependent feature.

Pairwise Potentials

The pairwise potential function $\psi(y_i, y_j)$ is computed using the co-occurrence of labels in the generated ground-truth and is independent of the actual segments. This is a simple way to capture the structure of the label space and use it in the form of soft constraints that do not enforce, but encourage a reasonable labeling with respect to the parent-child relationship.

Evaluated alternatives are different mapping parameters (as with the unary potentials, step 3) and normalization of the contribution of each image to the co-occurrence matrix. In addition two different types of parameter tying are used, effectively reducing the number of pairwise parameters to $|\theta_p| \in \{1, 2\}$. A single parameter means all combinations of labels y_i and y_j have the same weight, so that θ_p represents the weight of the pairwise term in comparison to the weights of the unary term. In the case of $|\theta_p| = 2$, one of the weights is used iff $y_i = y_j$ and the second weight is used otherwise. This means that the case where parent and child segments have the same label is scaled differently than where they do not have the same label.

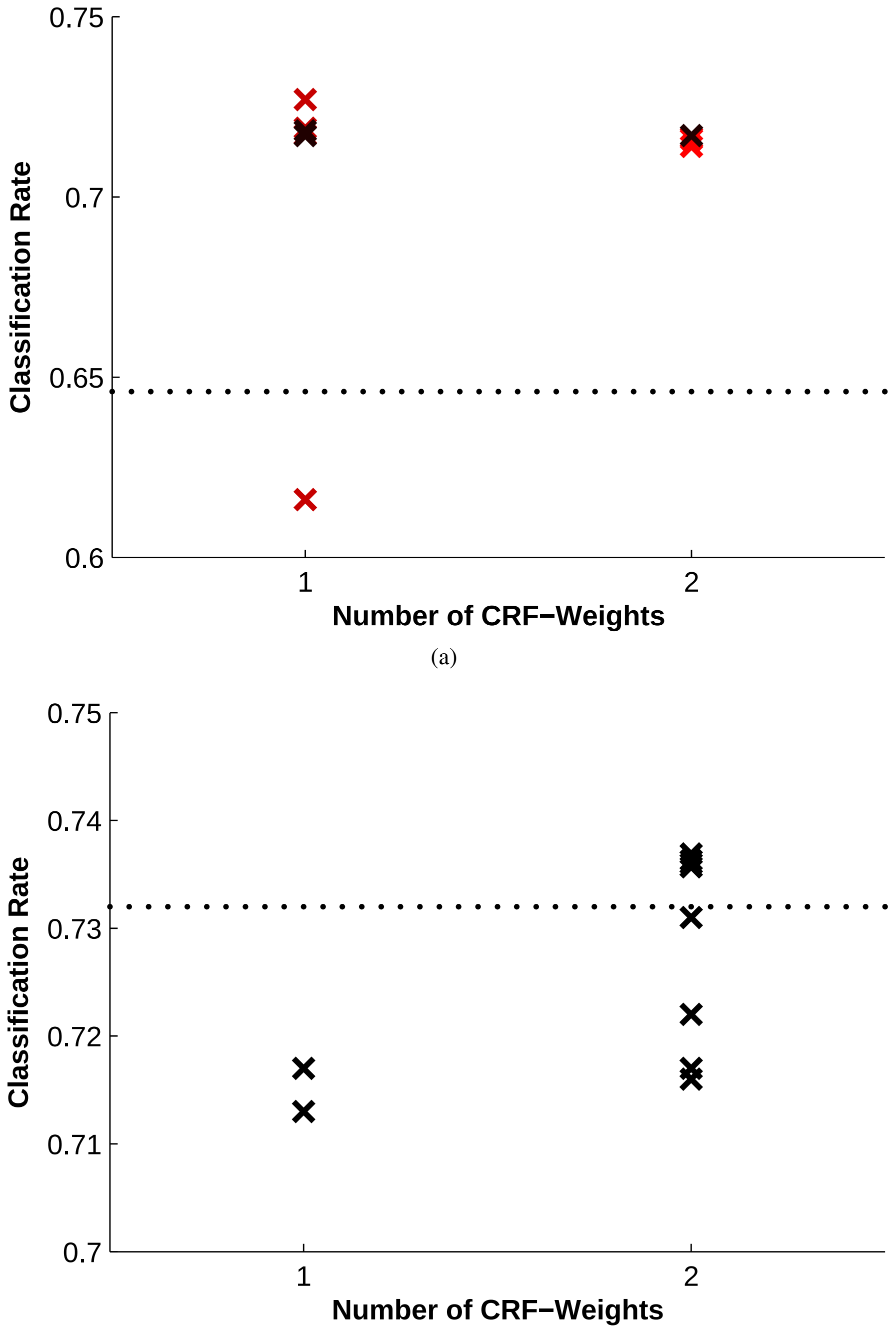
This definition of ψ allows for a model with a small number of parameters, high generalizability and training with a small number of samples. The trade-off between these variables remains, though.

Results

The results of the unary potential evaluations are illustrated in Fig. 4.4 and 4.5 for the CamVid and the Stanford Background dataset, respectively. The experiments shows that using the Geometric Context features with the Mahalanobis distance provide consistently better results than all other combinations. The best results are obtained using $\delta = 0.02$ and $\xi = 10^{-8}$. Note that the number of samples is unevenly distributed because promising setups were tested more thoroughly, due to limited time for the feature evaluation.

The evaluation results of the pairwise potential functions are depicted in Fig. 4.6a and 4.6b for the CamVid and the Stanford Background dataset, respectively. Based on these results we chose to use 2 weights and focus on the Stanford Background dataset for further evaluation. Best results for this case were achieved using

$$\psi(y_i, y_j) = e^{-\frac{50}{N} C(y_i, y_j)}, \tag{4.5}$$



(a)

(b)

Figure 4.6: Evaluation of pairwise potential functions on the (a) CamVid and (b) Stanford Background dataset, using 1 or 2 model parameters (CRF-Weights). Brighter symbols represent multiple occurrence of the same classification rate. Dotted lines represent the best classification rates achieved without pairwise potentials (see Fig. 4.4 and 4.5).

where N is the number of sample images and $C(y_i, y_j)$ is the co-occurrence of labels in the training set, with the contribution of image k to C being L_1-normalized, i.e.

$$C(y_i, y_j) = \sum_{k=1}^{N} \frac{\left|\left\{s_p, s_q \in \mathcal{S}_k : s_q \in m_{\mathcal{S}_k}(s_p) \wedge y_p^{(k)} = y_i \wedge y_q^{(k)} = y_j\right\}\right|}{|\{s_p, s_q \in \mathcal{S}_k : s_q \in m_{\mathcal{S}_k}(s_p)\}|}. \tag{4.6}$$

4.5 Extending the Label Set

In the following, we discuss results obtained by using more than 3 base labels on the Stanford Background dataset. Previously 3 geometric classes and all of their combinations $\mathcal{L}_7$ were used ($\mathcal{L}_7 = \{(1), (2), (3), (1, 2), (1, 3), (2, 3), (1, 2, 3)\}$, with the numbers $1, 2, 3$ representing the object classes; this makes a total of 7 label combinations). We now compare this to a bigger set of label combination, based on the 8 semantic classes of the data set. Let $\mathcal{L}_9$, $\mathcal{L}_{12}$ and $\mathcal{L}_{45}$ be new label sets, where $\mathcal{L}_x$ contains x combinations of base labels. Label combinations in $\mathcal{L}_9$ and $\mathcal{L}_{45}$ contain $\{1, 8\}$ and $\{1, 2, 7, 8\}$ base labels, respectively. That means in the case of $\mathcal{L}_9$ we use all base labels (combinations containing 1 label) and one combination containing all 8 base labels. The set $\mathcal{L}_{12}$ contains all 8 base labels and 4 manually defined label combinations with a associated semantics: *flora/fauna* = {*tree, grass, fg.obj*}, *nature* = {*tree, grass, water, mountain, fg.obj*}, *city* = {*road, building, fg.obj*} and the combination *image*, containing all base labels. An example of the generated ground-truth for the label set $\mathcal{L}_{12}$ is illustrated in Fig. 4.9. Note that this label set has a relevant flaw: it does not consider the foreground-object to be part of a simple scenery, e.g. one can see in Fig. 4.9 that there is a *fg.obj* on a *road*, each segment containing parts of both objects is labeled as *city*. In the specific case of the Stanford Background dataset, where the class *fg.obj* represents many different real-world object classes it may make sense to ignore this base label. However, the goal of this work is not to optimize the hierarchy of semantic labels, therefore the straight-forward method of base labels and combinations of them is used for all cases.

The number of possible label combinations for a base label set L_B is $2^{L_B} - 1$. Generally having a larger label set implies a bigger classification error (assuming the frequency of all labels in the ground truth is greater than 0). On the other hand, the stronger the correlation between specific labels and the features in the training set, the more accurate the model. Fig. 4.7 illustrates this behavior. A smaller label set with less meaningful label combinations $\mathcal{L}_9$ has very low initial classification rate. Note that having a lower initial classification also means that there are more regions to correct and the slope of the classification rate tends to be steeper. In order to fully assess the results, it is necessary to at least relate the graph to the initial classification rate (shown in parentheses in the legend).

Another relevant observation is that $\mathcal{L}_{12}$ has better performance than $\mathcal{L}_{45}$ in the interactive framework and has a slightly higher initial classification rate (44% compared to 45%). This shows that a small set of manually defined label combinations with semantic meaning fits better to the model than a larger set containing most of these combinations.

Fig. 4.8 shows the distribution of label combinations over the number of base labels for the 3 new label sets. These graphs agree with the assumption that combinations containing a low amount of base labels are more distinctive and also show that there are label combinations that

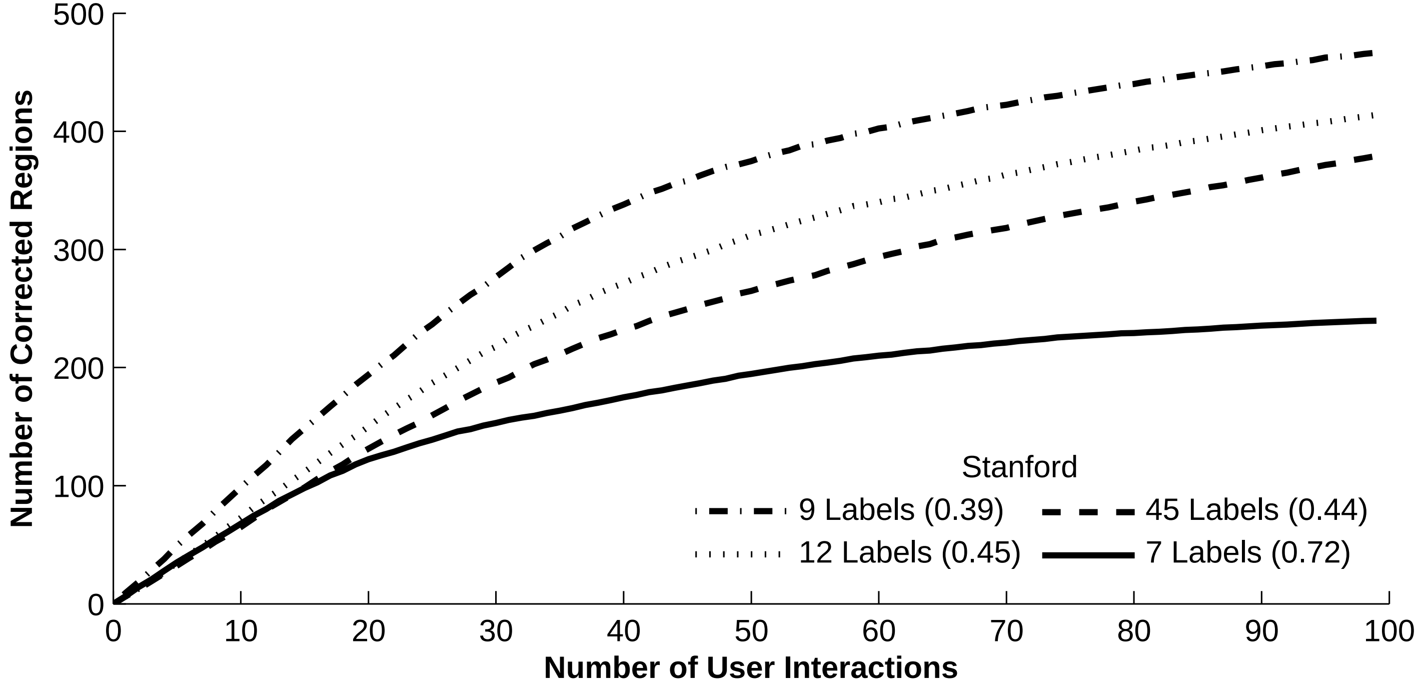


Figure 4.7: Illustration of the improvement in the number of correctly classified segments based on the number of user interactions for different label sets. It shows the the average number of corrected regions over the number of user interactions on the Stanford Background dataset. In parenthesis of the legend the average classification rates at 0 interactions are shown, since this initial handicap still affects the curves.

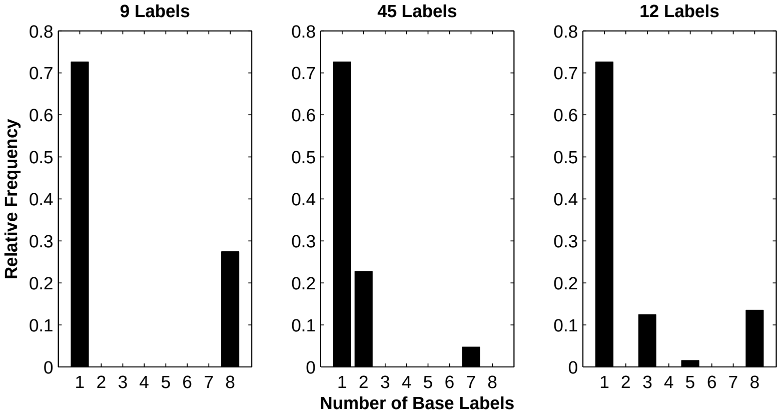


Figure 4.8: Histogram of label combinations over the number of base labels for different label sets on the Stanford Background dataset.

do not occur in the ground-truth (e.g. the combination containing all 8 base labels in $\mathcal{L}_{45}$). In such cases the CRF training will eliminate the influence of such classes by setting the $\theta_u(y_i)$ to a significantly lower value than other parameters.

We further show examples of initial predictions in Fig. 4.10, 4.11 and 4.12, depicting results with worst, median and best classification rates respectively. For the purpose of illustration we only show a single level of the segmentation hierarchy each. Results indicate that our framework works best on man-made structures like roads and buildings. This is expected as the GC features are optimized for such scenes. Another observation is that the results are smooth (neighboring regions tend to have the same label). This is also expected because children of the same parent that have similar features tend to have the same label, thus indirectly smoothing the semantic segmentation. Please note that there is no interaction involved in the creation of these examples.

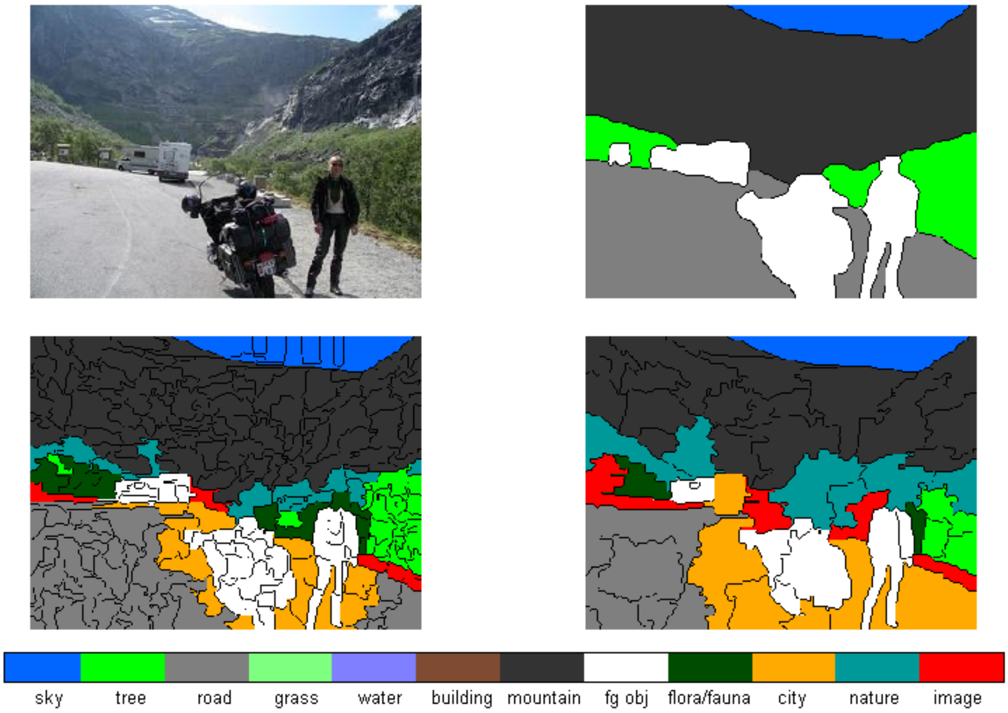


Figure 4.9: Visualization of ground-truths of an image (*top left*) from the Stanford Background dataset. It shows the original ground-truth of the dataset (*top right*) and the computed ground-truth of a fine (*bottom left*) and a coarse (*bottom right*) segmentation of the segmentation hierarchy. The illustration is done using the semantic label set $\mathcal{L}_{12}$ (semantic labels of the Stanford Background dataset, one label combination $image$ containing all labels and the following manually defined combinations: $flora/fauna$, $nature$ and $city$). Image best viewed in color.

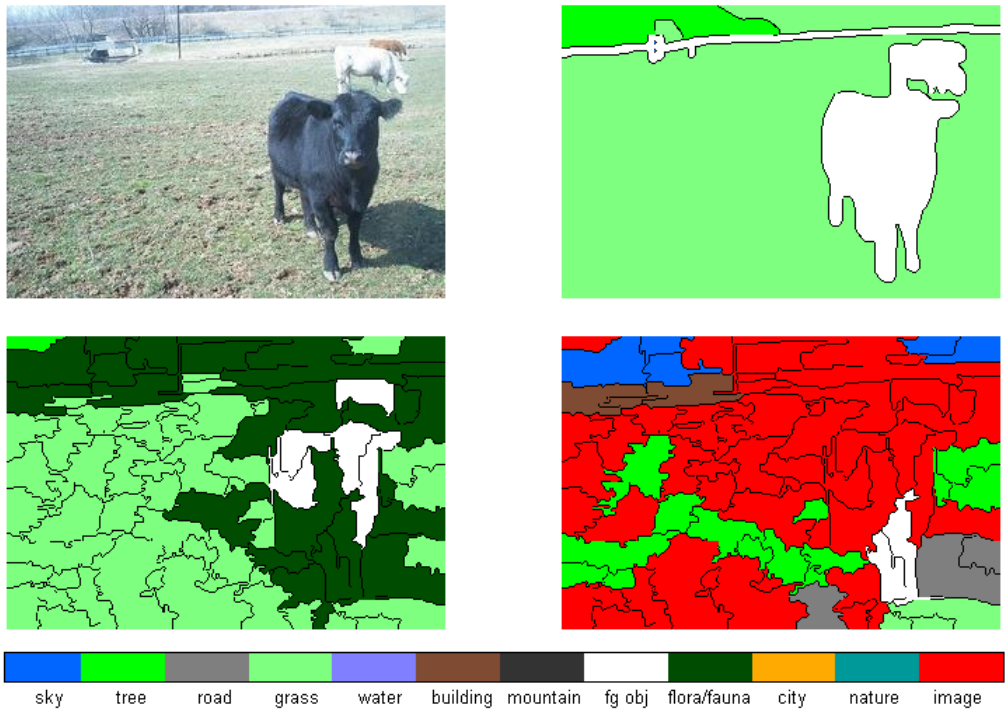


Figure 4.10: Semantic segmentation result with lowest classification rate, using an image (*top left*) from the Stanford Background dataset. It shows the original ground-truth of the dataset (*top right*), the computed ground-truth (*bottom left*) and the initially predicted labeling (*bottom right*), both showing only level 2 of the segmentation hierarchy. The computation is done using the semantic label set $\mathcal{L}_{12}$. Image best viewed in color.

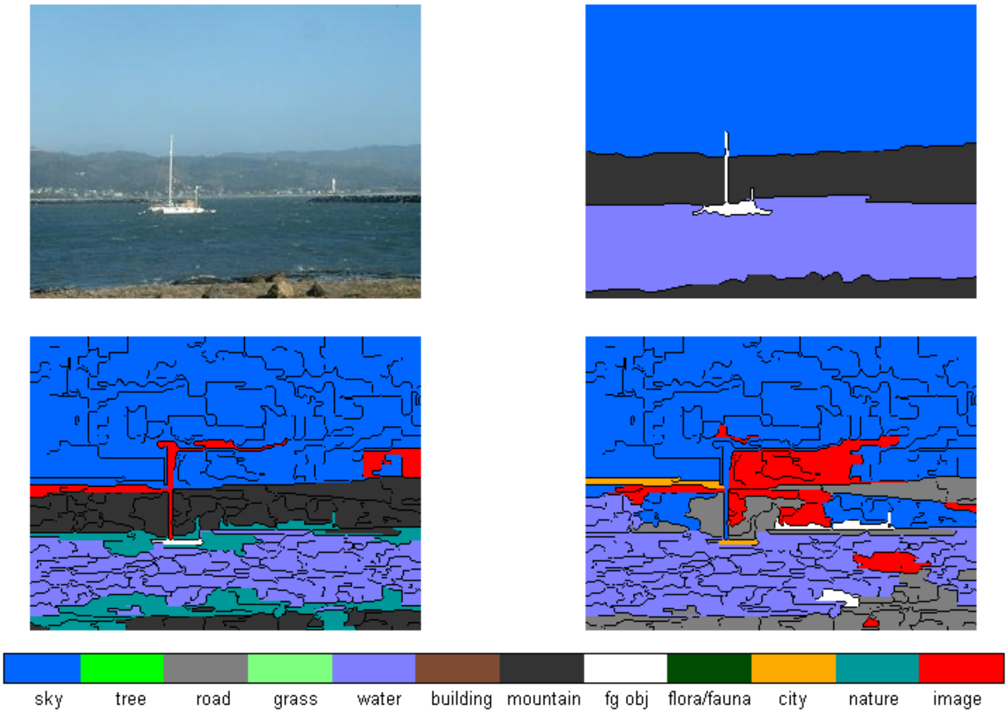


Figure 4.11: Semantic segmentation result with median classification rate, using an image (*top left*) from the Stanford Background dataset. It shows the original ground-truth of the dataset (*top right*), the computed ground-truth (*bottom left*) and the initially predicted labeling (*bottom right*), both showing only level 2 of the segmentation hierarchy. The computation is done using the semantic label set $\mathcal{L}_{12}$. Image best viewed in color.

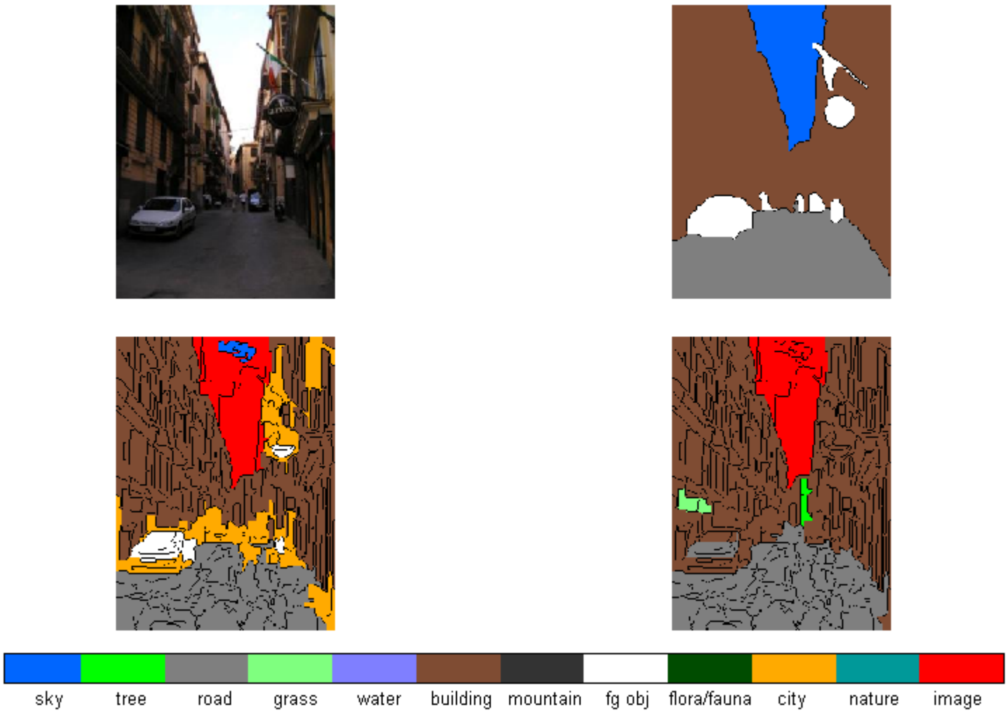


Figure 4.12: Semantic segmentation result with highest classification rate, using an image (*top left*) from the Stanford Background dataset. It shows the original ground-truth of the dataset (*top right*), the computed ground-truth (*bottom left*) and the initially predicted labeling (*bottom right*), both showing only level 2 of the segmentation hierarchy. The computation is done using the semantic label set $\mathcal{L}_{12}$. Image best viewed in color.

CHAPTER 5

Summary and future work

5.1 Summary

In this work we present a probabilistic framework for interactively labeling hierarchical image segmentations, including the relevant theory of this CRF based method.

The approach is then compared to a baseline consisting of independent prediction of segments, which shows the advantages of structured prediction in this setup. The main evaluation is done by using a virtual annotator, that corrects misclassified segments in top-down manner. Other experiments include comparison of different framework components and behavior for larger label set. One of the most exciting observations is that a large set of labels (45 labels) provides worse classification rates than a subset containing a few semantic labels (12 labels).

Also the most frequent label combinations contain 1-3 base labels (see Fig. 4.8) and several label combinations do not occur. While it is be possible to learn distinguishable label combinations by clustering, thus improving overall performance, this defeats the purpose of the framework. The method is supposed to predict object classes defined by a human user based on a training set.

5.2 Open Issues

One of the main goals of this work is to quickly generate ground-truth for the specified task. However, the proposed method is supervised, which creates a chicken-egg problem: we need ground-truth to generate ground-truth. We deal with this issue by generating a custom ground-truth that does not necessarily represent the objects in the scene.

With this custom ground-truth comes the computed hierarchy of (semantic) labels, which do not have distinct semantic meaning but rather depict combinations of base classes. Even when manually defining specific label combinations with a semantic meaning these combinations may be ambiguous, e.g. in the experiment with 12 labels, we have many segments labeled *city* in the ground-truth, even though there is no city in the image. A suitable manually produced ground-truth with carefully chosen object classes may solve this problem.

We keep the number of model parameters in order to encourage generalizability and allow training on small data sets, so that the time spend on a manually produced ground-truth is minimized. Initially this assumption was supposed to be analyzed by repeatedly learning the model on training data of increasing size. However, at that point in time a single training iteration took several days because the optimization in Eq. 3.12 was done with a smaller error tolerance. Repeated training seemed infeasible for the time plan of this work. Thus the focus shifted to the comparison between structured and independent prediction in this interactive framework.

Relative geometric features are not considered in our experiments, the Geometric Context features normalized position and shape features, but the position relative to its parent may be more relevant. We do not rely on a segment fully covering an object or part, so the use of such relative geometric features has yet to be evaluated.

5.3 Outlook

There are several options to possibly improve the performance by giving up generalizability or using approximate inference. Aside from obvious modifications and possibilities mentioned above, an evaluation of different kinds of models can be very helpful. For example, the model could be extended in order to include adjacency relations of the regions on one level of coarseness or use higher order energy terms. Also alternatives to CRF like Structured SVM [21] or Decision Tree Fields [22] may improve the quality of labelings.

Even with alternative models one of the big problems for real-life use of this framework is the underlying segmentation. There is a method for object class segmentation that incorporates both segmentation and labeling in a joint probabilistic model [13], which may be a solution in this context, which does not rely on another suboptimal image segmentation method. However, this would result in a more complex model, likely needing more parameters and/or a larger training set to achieve similar results.

There is also the option of using the general idea of the framework for a different task. For example one could use a deformable parts model [8] and extend it to include semantics, thus detecting and identifying objects and its parts without relying on a segmentation method. Once could also increase the level of abstraction further and look at the relation between detected objects, which may improve confidence in object labels - essentially this would be an interactive hierarchical version of the method by Rabinovich et al. [25]. On the other hand interactivity is not always desired and it depends on the requirements of the application whether or not such a solution may be useful.

Another possible research topic is to find a quality measure for labelings on segmentation hierarchies that are adapted to human perception, e.g. by finding correlations between human assessment of importance of regions or labels and a specific weight for the quality score.

Bibliography

[1] Pablo Arbelaez, Michael Maire, Charless Fowlkes, and Jitendra Malik. Contour detection and hierarchical image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 33(5):898–916, 2011.

[2] Mads Boedker. Shoe car image taken using a sony dsc-p200. http://www.flickr.com/photos/boedker/174418993/, 2006.

[3] Steve Branson, Pietro Perona, and Serge Belongie. Strong supervision from weak annotation: Interactive training of deformable part models. In *Proceedings of the 13th International Conference on Computer Vision (ICCV)*, pages 1832 –1839, 2011.

[4] Gabriel J. Brostow, Julien Fauqueur, and Roberto Cipolla. Semantic object classes in video: A high-definition ground truth database. *Pattern Recognition Letters*, 30(2):88–97, 2009.

[5] Gabriel J. Brostow, Jamie Shotton, Julien Fauqueur, and Roberto Cipolla. Segmentation and recognition using structure from motion point clouds. In *Proceedings of the 10th European Conference on Computer vision (ECCV)*, pages 44–57, 2008.

[6] João Carreira, Fuxin Li, and Cristian Sminchisescu. Object Recognition by Sequential Figure-Ground Ranking. *International Journal of Computer Vision (IJCV)*, pages 1–20, 2011.

[7] Mark Everingham, Luc Van Gool, Christopher K. I. Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *International Journal of Computer Vision (IJCV)*, 88(2):303–338, June 2010.

[8] Pedro F. Felzenszwalb, Ross B. Girshick, David McAllester, and Deva Ramanan. Object Detection with Discriminatively Trained Part-Based Models. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 32(9):1627–1645, 2010.

[9] Josep M. Gonfaus, Xavier Boix, Joost van de Weijer, Andrew D. Bagdanov, Joan Serrat, and Jordi Gonzàndlez. Harmony potentials for joint classification and segmentation. In *23rd Conference on Computer Vision & Pattern Recognition (CVPR)*, pages 3280–3287, 2010.

[10] Stephen Gould, Richard Fulton, and Daphne Koller. Decomposing a scene into geometric and semantically consistent regions. In *Proceedings of the 12th International Conference on Computer Vision (ICCV)*, pages 1–8, 2009.

[11] Yll Haxhimusa and Walter G. Kropatsch. Hierarchy of partitions with dual graph contraction. In *Proceedings of the 25th German Association for Pattern Recognition Symposium (DAGM)*, pages 338–345, 2003.

[12] Derek Hoiem, Alexei A. Efros, and Martial Hebert. Geometric context from a single image. In *Proceedings of the 10th International Conference on Computer Vision (ICCV)*, pages 654–661, 2005.

[13] Adrian Ion, Joao Carreira, and Cristian Sminchisescu. Probabilistic joint image segmentation and labeling. In *Advances in Neural Information Processing Systems (NIPS)*, pages 1827–1835, 2011.

[14] L'ubor Ladický, Chris Russell, Pushmeet Kohli, and Philip H.S. Torr. Associative hierarchical crfs for object class image segmentation. In *Proceedings of the 12th International Conference on Computer Vision (ICCV)*, pages 739 –746, 2009.

[15] Victor Lempitsky, Andrea Vedaldi, and Andrew Zisserman. A pylon model for semantic segmentation. In *Advances in Neural Information Processing Systems (NIPS)*, pages 1485–1493, 2011.

[16] David G. Lowe. Object recognition from local scale-invariant features. In *Proceedings of the 7th International Conference on Computer Vision (ICCV)*, volume 2, pages 1150–1157, 1999.

[17] Tomasz Malisiewicz and Alexei A. Efros. Improving spatial support for objects via multiple segmentations. In *Proceedings of the 18th British Machine Vision Conference (BMVC)*, 2007.

[18] Julian J. McAuley, Teofilo de Campos, Gabriela Csurka, and Florent Perronnin. Hierarchical image-region labeling via structured learning. In *Proceedings of the 20th British Machine Vision Conference (BMVC)*, 2009.

[19] Thomas Mensink, Jakob Verbeek, and Gabriela Csurka. Learning structured prediction models for interactive image labeling. In *Proceedings of the 24th Conference on Computer Vision & Pattern Recognition (CVPR)*, pages 833–840, 2011.

[20] Sebastian Nowozin, Peter V. Gehler, and Christoph H. Lampert. On parameter learning in crf-based approaches to object class image segmentation. In *Proceedings of the 11th European Conference on Computer vision (ECCV)*, pages 98–111, 2010.

[21] Sebastian Nowozin and Christoph H. Lampert. Structured learning and prediction in computer vision. *Foundations and Trends in Computer Graphics and Vision*, 6(3-4):185–365, 2011.

[22] Sebastian Nowozin, Carsten Rother, Shai Bagon, Toby Sharp, Bangpeng Yao, and Pushmeet Kohli. Decision Tree Fields. In *IEEE 13th International Conference on Computer Vision (ICCV)*, pages 1668–1675, 2011.

[23] Judea Pearl. Fusion, propagation, and structuring in belief networks. *Artificial Intelligence*, 29(3):241–288, 1986.

[24] Nils Plath, Marc Toussaint, and Shinichi Nakajima. Multi-class image segmentation using conditional random fields and global classification. In *Proceedings of the 26th Annual International Conference on Machine Learning (ICML)*, pages 817–824, 2009.

[25] Andrew Rabinovich, Andrea Vedaldi, Carolina Galleguillos, Eric Wiewiora, and Serge Belongie. Objects in context. In *IEEE 11th International Conference on Computer Vision (ICCV)*, pages 1 –8, oct. 2007.

[26] Josef Sivic and Andrew Zisserman. Video google: A text retrieval approach to object matching in videos. In *Proceedings of the 9th IEEE International Conference on Computer Vision (ICCV)*, volume 2, pages 1470–1477, 2003.

[27] Martin J. Wainwright and Michael I. Jordan. Graphical models, exponential families, and variational inference. *Foundations and Trends in Machine Learning*, 1(1-2):1–305, January 2008.

[28] Georg Zankl, Yll Haxhimusa, and Adrian Ion. Interactive labeling of image segmentation hierarchies. In *Pattern Recognition*, volume 7476 of *Lecture Notes in Computer Science*, pages 11–20. 2012.

[29] Hui Zhang, Jason E. Fritts, and Sally A. Goldman. Image segmentation evaluation: A survey of unsupervised methods. *Computer Vision and Image Understanding*, 110(2):260–280, May 2008.