# TRANSLATING IMAGES TO WORDS : A NOVEL APPROACH FOR OBJECT RECOGNITION

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
THE MIDDLE EAST TECHNICAL UNIVERSITY

BY

PINAR DUYGULU - ŞAHİN

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF

DOCTOR OF PHILOSOPHY

IN

THE DEPARTMENT OF COMPUTER ENGINEERING

FEBRUARY 2003

Approval of the Graduate School of Natural and Applied Sciences.

Prof. Dr. Tayfur Öztürk
Director

I certify that this thesis satisfies all the requirements as a thesis for the degree of Doctor of Philosophy.

Prof. Dr. Ayşe Kiper
Head of Department

This is to certify that we have read this thesis and that in our opinion it is fully adequate, in scope and quality, as a thesis for the degree of Doctor of Philosophy.

Prof. Dr. Fatoş Yarman - Vural
Supervisor

Examining Committee Members

Prof. Dr. Fatoş Yarman - Vural

Prof. Dr. Aytül Erçil

Assoc. Prof. Dr. Volkan Atalay

Assoc. Prof. Dr. Gözde Bozdağı

Assoc. Prof. Dr. Sibel Tarı

# ABSTRACT

TRANSLATING IMAGES TO WORDS : A NOVEL APPROACH FOR OBJECT RECOGNITION

Duygulu - Şahin, Pınar

Ph.D., Department of Computer Engineering

Supervisor: Prof. Dr. Fatoş Yarman - Vural

February 2003, 127 pages

In this thesis, we propose a new approach to the object recognition problem, motivated by the recent availability of large annotated image collections. This approach considers object recognition as the translation of image regions to words, similar to the translation of text from one language to another. The "lexicon" for the translation is learned from large annotated image collections, which consist of images that are associated with text. First, images are segmented into regions, each of which are represented by a pre-specified feature vector. Then the regions (of all the training images) are clustered in the feature space, categorizing the regions into a finite set of blobs. The correspondences between the blobs and the words are learned, using a method based on the Expectation Maximization algorithm. Once learned, these correspondences can be used to predict words corresponding to particular image regions (region naming), or words associated with whole images (auto-annotation).

The method is applied on the Corel data set, a large collection of stock photographs annotated by a set of keywords. A series of experiments are carried out to assess the performance of the method. First, the accuracy of predictions is evaluated on a relatively small number of hand-labeled images. Then the system is evaluated by using annotation performance as a proxy. Annotation performance is evaluated using three

measures: Kullback-Leibler divergence between the predicted and target distributions, normalized classification score and word prediction rate. The results indicate that, the method can predict numerous words with high accuracy. Due to the lack of a ground truth, the performance of the proposed system is compared against two other methods: predictions using empirical word densities and the co-occurrences of blobs and words. The results clearly show that, the proposed method has a better performance than these two methods. Finally, extensions of the basic method to improve the performance of the system are discussed.

Keywords: Object Recognition, Correspondence, Machine translation, Annotated Image Collections, EM algorithm

# ÖZ

GÖRÜNTÜLERDEN KELİMELERE ÇEVİRİ : NESNE TANIMA PROBLEMİNE YENİ BİR YAKLAŞIM

Duygulu - Şahin, Pınar
Doktora, Bilgisayar Mühendisliği Bölümü
Tez Yöneticisi: Prof. Dr. Fatoş Yarman - Vural

Şubat 2003, 127 sayfa

Günümüzde etiketlenmiş görüntü veri tabanlarının artışıyla birlikte, görüntülerin öznitelikleri ve anahtar kelimeler çeşitli amaçlar için birlikte kullanılabilir hale gelmiştir. Bu çalışmada, bölütlenmiş görüntülere kelime yerleştirme yeni bir nesne tanıma yöntemi olarak önerilmektedir. Bu yöntem, nesne tanıma problemini görüntü bölütlerinin kelimelere çevirisi olarak değerlendirir. İşlem bir dilin başka bir dile çevrilmesine benzerdir ve bir çeşit bilgisayarlı çeviri yöntemi olarak tanımlanabilir. İlk işlem görüntülerin bölütlenmesi ve her bölütten önceden belirlenmiş bir öznitelik vektörünün çıkarılmasıdır. Daha sonra bölütler öznitelik uzayında topaklandırılarak, sonlu sayıda bölüt kategorisi oluşturulur. Bölüt kategorileriyle, kelimeler arasındaki uygunluk ilişkisi "Expectation Maximization (EM)" algoritmasının kullanıldığı bir yöntemle öğrenilir. Bu aşamadan sonra, öğrenilen uygunluk ilişkisi kullanılarak, verilen bir görüntü bölgesine karşı gelen kelime ya da bir görüntünün anahtar kelimeleri tahminlenebilir. Bu yöntem bölüt adlandırma, ve otomatik etiketlendirmede kullanılabilir.

Yöntem çok sayıda etiketlenmiş görüntü içeren Corel koleksiyonunda uygulandı ve deneylerle yöntemin başarısı değerlendirildi. Önce yöntemin tahminlerindeki doğruluk başarısı, göreceli olarak ufak sayıdaki elle etiketlenmiş görüntüler üzerinde değerlendirildi. Sonra sistem, görüntülerin etiketleri vekil olarak kullanılarak değerlendirildi.

Etiketleme başarısı üç ölçekle değerlendirildi: tahmin ve hedef dağılımları arasındaki Kullback-Leibler uzaksaması, normalize edilmiş sınıflandırma skoru, ve kelime tahmin oranı. Sonuçlar, yöntemin bir çok kelimeyi doğru olarak tahmin edebildiğini gösterdi. Temel alınabilecek mutlak bir doğrunun yokluğunda, önerilen sistemin başarısı diğer iki yöntemle karşılaştırıldı: ampirik kelime yoğunluguna dayalı önerme, ve bölüt ve kelimelerin birlikte varolma oranları. Sonuçlar önerilen yönte-min kesin olarak diğerlerinden daha iyi başarıya sahip olduğunu göstermekte. Son olarak, temel yöntemin başarısını yükseltebilecek eklemeler tartışıldı.

Anahtar Kelimeler: Nesne tanıma, ilişkilendirme, bilgisayarla çeviri, etiketlenmiş görüntü veri tabanları, EM algoritması

# ACKNOWLEDGMENTS

It is a great pleasure to have the opportunity to express my gratitude to all those who gave me the possibility to complete this thesis.

I would like to give my sincere thankfulness to my thesis supervisor, Prof. Fatoş Yarman Vural, for her support and guidance throughout my study. Without the opportunities that she provided, I would not be able to come to this point. She has been a mentor for not only my research but also my life. In all my academic and personal life I will walk through the way that she taught me.

I am grateful to Prof. David Forsyth, for giving me the opportunity to work with him and for his guidance. He has been very stimulating and encouraging. He is a perfect model for me both as an advisor and as a teacher.

My special thanks are for Kobus Barnard, for all his patience and help. I felt that he was like a big brother for me, protecting and guiding me during my stay at U.C. Berkeley.

It was a pleasure for me to work with Nando de Freitas, Jaety Edwards and Doron Tal. I would like to thank them for all the helpful discussions and joint work.

I also would like to thank to Prof. Jitendra Malik, Prof. Robert Wilensky and all the other members of Computer Vision Group and Digital Library Project in U.C. Berkeley for the valuable discussions and for being very helpful to me. Especially, I would like to thank Okan Arıkan and Alyosha Efros for their friendship which helped me during my hard days at Berkeley.

I am grateful to Prof. Volkan Atalay, for all the guidance he gave me since my third year in undergraduate study. I learned the joy of research from him.

All the members of Image Processing and Pattern Recognition Laboratory at METU: Nafiz, Özge, Turan, Murat, Ulaş, Aykut, Erkut, and Gülşah, thanks a lot for being so friendly and patient. Without your encouragement it would have been

hard to finish this thesis. I also want to thank all the other previous and current research assistants in the department, for their friendship which makes the research a joy. My special thanks are to Ersan, Onur and Erek who helped me a lot.

I am also thankful to the wonderful faculty members and staff at the Dept. of Computer Engineering, METU, for their endless friendship and support. To be with them was one of the most important motivations to start such an hard journey.

I should not forget all my teachers during my whole education, for exposing me to the beauty of learning.

I also want to thank TÜBİTAK for providing the financial support during my stay in U.C. Berkeley.

My endless thanks are to my family, for being a constant source of strength during this endeavor. It is my family who makes my life so wonderful. Not only this thesis but everything in my life was possible because of them. It is not easy to put my thankfulness into words. My mother Filiz, my father Bekir, my brother Özgür and my cats Minnoş and Cingöz, thanks a lot for your love, encouragement, patience. Thanks a lot for always being with me. Life would have no meaning without you.

And finally, I would like to thank my husband Erol for his endless support that made it possible to finish this thesis in time. Without him, neither this thesis nor my life would be complete. Thanks for being the sun of my life.

To my family

# TABLE OF CONTENTS

# LIST OF TABLES

TABLE

# LIST OF FIGURES

FIGURES

# CHAPTER 1

# INTRODUCTION

Vision and language constitute two important aspects of human communicative capabilities. Therefore, it is not surprising to see that text and image have become the most common forms of representing information. Humans often make use of the combination of text and image to utilize the expressive power of these modalities in parallel. It is due to the fact that there exist deep semantic connections between the two forms.

The concept of *object* represents one of the main types of the semantic connections between image and text. Although the object concept is natural to a human observer, the problem of object recognition in images still poses a significant challenge to computer vision systems.

Object recognition is an old and difficult problem in pattern recognition. There are many fundamental questions that need to be answered. The question of *what is an object?* is one of these fundamental questions. It is a conceptually difficult question. For example, it is not clear whether a *face* is a single object or a composite object. If it is composite, what is the level of division: a face has eye as a part, an eye has an eyelid, etc. It is not clear where to stop.

Many different approaches for object recognition have been proposed until now. Most of the existing systems address the question by constructing the model for the object that the system is built to recognize. However, such an approach is not easily applicable to large-scale problems, since constructing generic models, which cover wide variety of objects is not possible.

Instead of using constructed object models for object recognition, an alternative approach is to learn the models from samples. A possible source that can be used for this purpose is collections of annotated images, where the images are associated with some descriptive text. Discovering the semantic connections between the image (which consists of regions that correspond to objects or parts of objects) and the text is a promising approach to the object recognition problem.

There are a wide variety of annotated image collections (e.g. Corel data set [2], most museum image collections [3], the web archive [7] and most collections of news photographs on the web [8]). Although, it is known that the annotation words are associated with the image, the correspondences between the words and the image regions are unknown. Unfortunately, only a few data sets contain the correspondence information, because of the labor-intensive effort required to do this difficult task.

In this study, our aim is to recognize objects by finding the correspondences between regions and words in the large annotated data sets. This study is part of a joint project conducted at University of California at Berkeley, in which joint distribution of image regions and words are learned from large annotated data sets [11, 12, 13, 14, 15, 16, 17, 26, 31].

There are several application areas of using words and images in parallel. The following example applications extensively utilize the joint distribution of images and words:

- **Browsing support:** Museums release their collections partially in the web to attract visitors. Typically users who don't know the collection well, prefer to browse [34]. Therefore, it is attractive to organize the collection to support browsing. Collecting together images that look similar and similarly annotated is a good start. Using image and text together, clustering performance is improved, hence browsing becomes more practical.

- **Auto-illustrate:** A tool that automatically suggests images to illustrate blocks of text (auto-illustration) would be interesting for many users. Auto-illustration is possible if the joint probability of text and image can be learned. Then, to illustrate a text, one can obtain images with high probability given a text.

- **Automated image annotation:** Numerous organizations manage collections of images for internal use [50]. For example, archivists receive pictures and annotate them with words that are likely to be useful keys for retrieving the pictures; journalists then search the collection using these keywords. Annotation is often difficult and uncertain; it is attractive to have a procedure that annotate images automatically. Annotation process can be done automatically, by learning the joint probabilities of words and images. It is possible to auto-annotate the images by predicting words with high posterior probability given an image.

For auto-annotation, words are predicted for a given image. Although, in that sense, auto-annotation can be considered as a suggestive strategy for recognition, it doesn't identify *which* image region corresponds to *which* word. The models proposed for annotation can be adapted for solving the correspondence problem [11]. However, since these models are trained to learn the relationships between the whole image and words, the specific relationships between the image regions and words are not explicitly learned.

The correspondence problem can be considered as a translation from image regions to words. In that sense, there is an analogy between learning a lexicon for machine translation and learning a correspondence model for associating words with image regions. In both cases, a representation of one form (image regions; French) needs to be translated into a representation of another form (words; English).

Learning a lexicon from data is a standard problem in machine translation literature (see [46, 49, 55]). Typically, lexicons are learned from a type of data set known as an **aligned bitext** — a text in two languages, where rough correspondence, perhaps at the paragraph or sentence level, is known. The problem of lexicon acquisition involves determining precise correspondences between words of different languages.

Data sets consisting of annotated images are similar to aligned bitexts – there is an image, consisting of a number of regions, and a set of associated words. Although the annotation words are associated with the image, it is not clear which word refers to which region in the image. Therefore, we propose that the lexicon learning method of machine translation can be applied to the problem of linking image regions with words, i.e. finding the correspondences between image regions and words. This approach can be considered as a form of object recognition since, one can learn the words that

correspond to image regions.

With the proposed approach it is now possible to attack the question of *what counts as an object?*. The answer is all the words in the vocabulary. It becomes possible to recognize a large number of different objects from data sets that are practically available.

## 1.1 Learning the correspondences between words and image regions

The goal of this study is to achieve learning the correspondences between words and image regions as a form of object recognition. Annotated image collections are used for this purpose. There are several large image collections where each image is manually annotated with some descriptive text. Due to the developing technology, besides these annotated image collections, there are many other available sources where image and text occur together: there is a huge amount of data on the web, where images occur with a surrounding text; with the OCR technology it is possible to extract the text from images; and above all, almost all the images have captions which can be used as annotations.

However, these sources are not totally suitable for learning the correspondences between image regions and words, since the data is incomplete. The following four cases may occur in such a data:

- **one-to-one relationship between words and the regions in the image:** all the regions in the image are represented by one of the annotation words,
- **missing words:** some of the regions in the image are not represented with any of the words,
- **missing regions:** some of the annotation words do not correspond to any of the regions in the image,
- **unrelated words and regions:** none of the annotation words correspond to the regions, and none of the regions have a representative annotated word.

In this study, the Corel data set, which is large collection of photographs taken by professionals, is used. In the data set, each image is annotated with about 3-4 keywords. Images are segmented, so that there are 5-10 regions in each image.

Usually, there is not a one-to-one correspondence between the image regions and the annotation words. The number of regions in the image is usually different than the number of keywords. Even if the number of regions and words are equal, there may not be a one-to-one correspondence between the regions and words (e.g. the word `people` may correspond to both face and body of a person; or an object may be divided into more than one region due to the segmentation), or there may be more than one word for a single region (e.g. both `cat` and `tiger` words are used to define a tiger; or synonym words like `train` and `locomotive` are used together to define a locomotive). Also, some words do not correspond to any of the regions in the image. (e.g. the word `Scotland` does not correspond to any region, since it is a general word, not a descriptive word for an object), and some regions do not correspond to any of the words (e.g. the annotators usually don't enter the word `sky`, therefore, in the data set, the words for the sky regions are usually missing; also due to poor segmentation we may have meaningless regions which cannot correspond to any of the words).

Due to the problems mentioned above, it is not possible to obtain a fully annotated data set. However, even with complete annotations, learning the correspondences between words and regions is still a difficult problem. Because, the data set does not provide explicit correspondence. For example, for an image showing a tiger on the grass, and having the annotated keywords `tiger` and `grass`, it is known that tiger and grass are in the image, but it is not known which region is tiger and which region is grass. With a single image, it is not possible to solve this problem. If there were other images, where an orange stripy region (the region corresponding to tiger) occurs with other regions rather than a green region (which correspond to grass), than it would be possible to learn that, the orange stripy region is the `tiger` not the green one.

These problems lead us to use Expectation Maximization (EM) algorithm [27], which is a general method for maximizing the likelihood of an underlying distribution when data is incomplete or has missing values. In our case, the missing variable is the correspondences between words and image regions.

In this study, correspondence problem is attacked by using EM algorithm. First, the similar regions which are coherent according to a set of features are grouped into a single class. We call these classes `blobs`. The problem is then, constructing a probability table which links the blobs with the words. If the correspondences between the blobs and words were estimated, then it would be possible to obtain the probability table easily. Similarly, if these probabilities were estimated then it would be possible to find the correspondences. This suggests the following iterative strategy:

- use an estimate of the probability table to predict correspondences;
- then use the correspondences to refine the estimate of the probability table.

Initially, as an estimate of the probability table, co-occurrences of words and blobs are used. This a rough estimate for the probability table, since orange stripy blob co-occurs with the word `tiger`, and green blob co-occurs with the word `grass`. However, the orange stripy blob also co-occurs with the word `grass` and the green blob also co-occurs with the word `tiger`. Applying the above iterative strategy refines the results, so that the orange stripy blob corresponds to the word `tiger`.

The thesis argues that, the object recognition problem can be considered as the problem of translating images to words. It adapts a lexicon learning method (one that is proposed for machine translation) to learn the correspondences between the words and the regions from a set of annotated images in an "unsupervised" way. The system is "unsupervised", in the sense that, the words are available only for the images, not for the individual regions. The system learns the correspondence between regions and words using the available data set and applying EM algorithm.

The proposed method has some advantages over the existing approaches. It allows us to utilize the large annotated image data sets for constructing a "lexicon" that can then be used to recognize objects in images. The number of objects that can be learned depends on the size of the vocabulary, allowing a large number of objects to be learned. Therefore, the method can be applied to large-scale problems, where the traditional supervised methods fail, since manual-labeling for large quantities is not feasible. Also, it is possible to learn regions from a very diverse set; there is no restriction in the type of images or objects.

## 1.2 Organization of the thesis

The thesis is organized as follows: Chapter 2 presents a literature survey of the relevant studies. In Chapter 3, first, the problem domain is described in detail, and the basic approach for linking words to image regions is explained. Then, different measures for the performance of the system is proposed and discussed. Finally, extensions of the basic approach to improve the performance of the system are presented. In Chapter 4, extensive experimental results obtained from a large annotated image collection are presented and the strengths and weaknesses of the approach are discussed. Chapter 5 concludes with a summary of the proposed approach, and a discussion of possible future directions.

# CHAPTER 2

# RELATED STUDIES

Large amounts of visual material is already being stored in many collections. The organizations which maintain image collections are categorized by Evans [35] as:

- public bodies (such as museums, public libraries and national archives ),
- commercial enterprises (such as press and photo agencies, stock photo libraries and publicity departments of major companies),
- and specialist bodies (such as learned societies and individuals).

With the recent developments in digital imaging technology, in increasing measure, these holdings are being supplemented in electronic form. There is a huge amount of information, but it is not possible to access or make use of this information unless it is efficiently organized for searching and/or browsing.

In this chapter, possible ways of using such a large amount of data will be discussed. The current state of art in image retrieval systems will be reviewed in Section 2.1, by describing some of the current strategies and popular systems. Then, the gap between the user requirements and the available systems will be discussed. In Section 2.2, the advantages of using text and images together will be discussed by describing some of the systems that combine text and image in different ways. Finally, in Section 2.3, the machine translation idea, which is used in this study to link image regions and words, will be presented.

## 2.1 State of art in image databases

Many researchers introduce systems for searching image databases (see [38, 40, 43, 54, 66, 71] for recent surveys on image and video indexing and retrieval technologies). Early work on image retrieval systems are based on text input, in which the images are annotated by text and then text based databases are used for retrieval (see [23] for a survey on text based image retrieval systems). However, two major difficulties are encountered with text based approaches: First, manual annotation, which is a necessary step for these approaches, is labor-intensive and becomes impractical when the collection is large. Second, keyword annotations are subjective; the same image/video may be annotated differently by different annotators.

In order to overcome these difficulties, content-based image retrieval is proposed in the early 1990's. Instead of text-based annotations, images are indexed, searched or browsed by their visual features, such as color, texture or shape.

Recently, many systems are proposed to allow the utilization of simple textual descriptions, or complex visual features to search large databases. The literature in content-based image retrieval systems is broad. In Section 2.1.1, we summarize some of the major content-based image retrieval systems with the underlying methodologies.

In most of the systems, images are matched based on low-level features, like color and/or texture, extracted from the entire image or from image regions. With the exception of systems that can identify faces [68], people [37], pedestrians [63] or cars [68], matching is not usually directed towards object semantics.

However, the users seem to be interested in both the semantics and the appearance. In the user studies literature, authors deal with the disparity between what the users need and what the technology supplies In Section 2.1.2 the user studies will be discussed in detail.

### 2.1.1 Popular content-based image retrieval systems

Many content-based image retrieval systems have been proposed in the last decade. In the following, we describe some of the major systems:

- **QBIC :** QBIC (Query By Image Content) [36] is one of the first content based image retrieval systems developed by IBM. The queries in QBIC are based on sample images, user-constructed sketches and drawings, and selected color and texture patterns. The on-line QBIC demo is at
  `http://wwwqbic.almaden.ibm.com`.

- **RetrievalWare :** RetrievalWare [29] is a content-based image retrieval engine developed by Excalibur Technologies Corp. It searches the images according to their color, shape and texture content, brightness and color structure and aspect ratio. It supports the combination of these features and the weights associated with each feature can be adjusted by the users. More information is available at `http://vrw.excalib.com`.

- **Chabot :** Chabot [61], developed at UC Berkeley, retrieves images based on both their content information and associated meta-data. Chabot support queries by color and by text, in addition to some limited domain concept queries like "sunset" or "snow". Queries based on color are of the form "find me the image that has color mostly blue" and are performed on color histograms. More information can be found at
  `http://elib.cs.berkeley.edu/ ginger/chabot.html`.

- **Photobook and FourEyes :** Photobook is a tool developed at MIT Media Lab [65] for searching and browsing images. Features are compared using the matching algorithms that Photobook provides. Photobook includes FourEyes [56], which is an interactive, power-assisted tool for segmenting and annotating images based on the examples from the user. More information can be found at `http://vismod.www.media.mit.edu/vismod/demos/photobook/`.

- **ImageRover :** ImageRover [69], which is developed at Boston University, combines textual and visual statistics for searching the images from web. The user initializes a search by specifying a few keywords describing the desired images. The user can then refine this initial query through relevance feedback using both visual and textual cues. The on-line demo can be found at
  `http://cs-www.bu.edu/groups/ivc/ImageRover`.

- **Netra :** Netra is a prototype image retrieval system, developed in the UC Santa Barbara [28, 48]. Color, texture, shape and spatial location information of segmented image regions are used to search and retrieve similar regions from the database. It allows the user to compose queries like "retrieve all images that contain regions that have the color of object A, texture of object B, shape of object C, and lie in the upper one-third of the image" where the individual objects could be regions belonging to different images. The on-line demo is at `http://maya.ece.ucsb.edu/Netra`.

- **MARS :** MARS (Multimedia Analysis and Retrieval System) is a system developed at University of Illinois at Urbana-Champaign [67]. MARS organizes various visual features into a meaningful retrieval architecture which can dynamically adapt to different applications and different users. The on-line demo is at `http://www-db.ics.uci.edu/pages/demos/`.

- **Color-WISE and Web-WISE :** Color-WISE [70] is a color based image retrieval system, developed in Wayne State University. Dominant hue and saturation values are determined for different parts of an image through a process of block-based histogram building and peak detection. Web-WISE [82] is designed for content based seeking and retrieval of images on the web. More information can be found at
  `http://www.cs.wayne.edu/ ilc/vision/wise.html`.

- **Surfimage :** Surfimage [58] is a prototype software for the IMEDIA (Image and multimedia indexing, browsing and retrieval) project, developed at INRIA. Surfimage uses the query-by-example approach for retrieving images and integrates advanced features such as image signature combination, classification, multiple queries and query refinement with relevance feedback. The on-line demo can be found at
  `http://www-rocq.inria.fr/cgi-bin/imedia/surfimage.cgi`.

- **PicToSeek :** PicToSeek [39] is an image retrieval system for the web, developed at ISIS Research Group at University of Amsterdam. The system is implemented using photometric color and geometric invariant indices. The basic idea is to extract invariant features (independent of the imaging conditions) from each of

the images in the database, which are subsequently matched with the invariant feature set derived from the query image. The on-line demo can be found at `http://www.science.uva.nl/research/isis/zomax/`.

- **Blobworld :** Blobworld [21], which is developed at UC Berkeley, is a system for image retrieval, based on finding coherent image regions which roughly correspond to objects. Each image is automatically segmented into regions (blobs) with associated color and texture descriptors. Query is based on the attributes of one or two regions of interest, rather than a description of the entire image. The on-line demo can be found at `http://elib.cs.berkeley.edu/vision.html`.

There are many other image retrieval systems in the literature including Virage [62], VisualSEEK [74] and WebSEEK [73] , and WebSeer [79].

### 2.1.2 How people use image collections -user studies

Image databases are used in many areas: art galleries and museum management, architectural and engineering design, interior design, remote sensing and management of earth resources, geographic information systems, medical imaging, scientific database management systems, weather forecasting, retailing, fabric and fashion design, trademark and copyright database management, law enforcement, criminal investigation, picture archiving and communication systems. The use of image databases changes according to the application area. Users may search the art collections for a particular picture painted by a certain artist or in a specific color structure. Medical students or doctors may use medical databases to search for a specific disease. People wish to illustrate an article or book, so they may look for a picture related with the text. Many people use web to find pictures they have in their mind.

There are not many studies in image retrieval literature which analyze user needs. Recent work of Enser [10, 32, 33] deals with the disparity between user needs and technology supplies. He studied the Hulton Getty Collection [4] in U.K. which is the largest picture archive in Europe: size of the collection exceed ten million images. It consists of mostly monochrome prints and negatives, but also color slides, engravings, woodcuts, drawings, cartoons and maps. The requests to search the archive is mostly

by phone, but also with fax or mail. The requirement of the user is elicited by a picture researcher. He gives some example queries for Hulton Getty collection:
*"Dead body on trolley or in morgue"*,
*"Couples dancing charleston"*,
*"5-6 year old boy trampolining, in mid-air,in silhoutte"*,
*"Edwardian girl, aged 10-ish. Middle class, smartish type, doing anything"*,
*"Children at the seaside,traditional,nostalgic(sandcatsles, donkey)"*,
*"Edward VIII looking stupid"*,
*"Charles and Di kissing on balcony after marriage"*
*"North London street scenes, 1950s/60s. Inner city, with and without people"*
*"Danny Kaye on stage - young"*

In a recent study of Ornager [64], the use of digital image collections in a newspaper archive was observed to determine the type of questions that the users ask and the group of categories. The intention of Ornager's research was to define an operational subject indexing strategy for images. She was concerned with enhancing the user interface to deal with aspects of the information querying system. She suggested that indexing must encompass factual description (`of-ness`), expressional content description (`about-ness`) and indication of the context in which the image can be used.

Markkula and Sormunen [50] studied a Finish newspaper's digital photo archive concentrating on journalists as the users. They showed that photographs of named persons and object types were the most common categories of user needs and users defined their needs very often using contextual criteria that cannot be derived directly from the photographs but rather from the assigned textual descriptions. They claimed that content-based image retrieval algorithms were not successful by themselves for retrieval purposes but they could be a potential technology for developing browsing tools for large sets of photographs retrieved by textual queries. In [51], they introduced a test collection for the evaluation of content-based image retrieval algorithms. In the test collection the performance testing is based on photograph similarity perceived by end-users in the context of realistic illustrations tasks and environment. They focused on photographic needs originating from routine illustration tasks in the newsroom (journalistic work).

Jorgenson [44] studied human pictorial image perception on naive image users, and reduced the typical attributes which humans use to describe images into a template consisting of objects, people and other facets. In [45], she tested template of image attribute classes by investigating whether a template for image description would be useful to participants in framing their image descriptions.

Other user studies include the works of Keister [47], O'Connor [60], and Turner [80].

## 2.2 Using image and text

It is a remarkable fact that, while text and images are separately ambiguous, jointly they tend not to be; this is probably because the writers of text descriptions of images tend to leave out what is visually obvious (the colour of flowers, etc.) and to mention properties that are very difficult to infer using vision (the species of the flower, say). Linking image information with text annotations might improve the object based searches.

There are a wide variety of data sets that consist of very large numbers of annotated images. Table 2.1 lists some of the available data sets:

Table 2.1: Available data sets.

| | |
|---|---|
| Corel Image Data [2] | 40,000 images |
| Fine Arts Museum of San Francisco [3] | 83,000 images online |
| Cal-flora [1] | 20,000 images, species information |
| News photos with captions [8] | 1,500 images per day |
| Hulton Getty Archive [4] | 40,000,000 images (only 230,000 online) |
| TV news archives [6, 5] | several terabytes already available |
| Google Image Crawl | $>$ 330,000,000 images with nearby text |

Typically, the annotations refer to the content of the annotated image, more or less specifically and more or less comprehensively. For example, the Corel annotations describe specific image content, but not all of it; museum collections are often annotated with some specific material (the artist, date of acquisition, etc.) but often contain some rather abstract descriptions.

Integrating the semantic information provided by text and the visiual information provided by image features is very helpful for many tasks. One can currently use words to search for pictures (it is often productive to use a sequence of terms and then 'jpg' or 'jpeg' as a query to Google). There are a variety of ways to use words and pictures

simultaneously. The most straightforward is to search using a simple conjunction of keywords and image region features, as provided in Blobworld [21]. Webseer [79] uses similar ideas for query of images on the web, but also indexes the results of a few automatically estimated image features. These include whether the image is a photograph or a sketch and notably the output of a face finder. Going further, Cascia *et. al.* [22] integrate some text and histogram data in the indexing. Others have also experimented with using image features as part of a query refinement process [24, 25]. Srihari *et.al.* have used text information to disambiguate image features, particularly in face finding applications [75, 76, 77].

However, there are not many systems using the joint statistics of text and images. Such a probabilistic approach is more useful than using the boolean queries, since it doesn't require to know exactly the right search terms to get useful results. As discussed in Chapter 1, there are several application areas for the methods that can model the joint probability distributions of text and images. From our perspective, the important point is that one might predict the text, given the images, using the joint probabilities. There are two ways to do this. Firstly, one might attempt to predict annotations of entire images using all information present (**auto-annotation**); secondly, one might attempt to associate particular words with particular image substructures to infer **correspondence**.

In [52, 53], Maron *et. al.* propose automatic annotation of images. They use multiple-instance learning to train classifiers for identifying particular keywords from image data using labeled bags of examples – an image is "positive" if it contains a tiger somewhere amongst all the other stuff and "negative" if it doesn't. Rather than attempt to sort out all correspondences between image structures and words directly, they build classifiers for each word separately.

In the work of Mori *et al.* [57], co-occurrence statistics are collected for words and image areas defined by a fixed grid. However, as discussed before, co-occurences does not provide the exact correspondences.

Barnard and Forsyth [17] proposed a model which cluster image representations and text, to produce a representation of a joint distribution linking images and words. The model is a multi-modal extension of Hofmann's hierarchical model for text [41, 42] and combines the assymetric clustering model which maps documents into clusters and the symmetric clustering model which models the joint distribution of documents and features ("aspect' model).

In their model, images and co-occurring text are generated by nodes arranged in a tree structure, with the leaves of the tree corresponding to clusters. The nodes generate both image regions using a Gaussian distribution, and words using a multinomial distribution. Each cluster is associated with a path from a leaf to the root. Nodes close to the root are shared by many clusters, and nodes closer to leaves are shared by few clusters. Therefore, in a properly fitted model, nodes closer to the root tend to emit items (words or regions) shared by a large number of data elements, and the nodes closer to the leaves each emit items more specific to small numbers of data elements. The documents belonging to a given cluster are modeled as being generated by the nodes along the path from the leaf corresponding to the cluster up to the root node, with each node being weighted on a document and cluster basis. Taking all clusters into consideration, a document is modeled by a sum over the clusters, weighted by the probability that the document is in the cluster

By integrating the two kinds of information during model construction, the system learns links between the image features and semantics which can be exploited for better browsing, better search, and for associating words with images (auto-annotation and auto-illustration) [11, 12, 16, 17]. However, this system does not model the relationships between specific image regions and words explicitly. Correspondence can be encoded to some extent through co-occurrence because there is a advantage of having "topics" collect at the nodes. For example, if the word `tiger` always co-occurs with orange stripy region and never otherwise, then these items will likely be generated by a shared node, as there are far fewer nodes than the observations.

In this thesis, the problem of learning the correspondence between image regions and words is treated as a translation process, similar to the translation of text in two different languages. In the next section, we describe the adapted machine translation idea.

## 2.3 Machine translation

The problem of machine translation has been studied for a long time. Recently, Brown *et.al* [20] suggested that it may be possible to construct automatic machine translation systems by learning from inputs and outputs of examples. They built statistical models of **translational equivalance** which is the relation between two expressions with the same meaning but in different languages. These models act as **lexicons**, devices that predict one expression given another expression.

Learning a lexicon from data is a standard machine translation problem (a good guide is Melamed's book [55]; see also [46], [49]). Lexicons are typically learned from a type of data set, known as an **aligned bitext**, which consists of many small blocks of text in both languages, that are known to correspond to each other (at paragraph or sentence level). A well-known example is the "Hansard Corpus" consisting of debates from the Canadian Parliament[], where each speaker's remarks in the country's two official languages -English and French-, correspond in meaning. Using these aligned bitexts, the problem of lexicon learning is transformed into the problem of finding the correspondences between words of different languages, which can then be tackled by machine learning methods.

Let's assume that the English phrase ``the big house'' and the French phrase ``la grande maison'' are given. It is known that these phrases are the translations of each other, but it is not known which English word is the translation of which French word. Using a single pair, it is not possible to learn the word correspondences. All three English words are equally likely to be the translations of the French words. However, if the English - French pairs ``the big car - la grande voiture'', ``the big tree - la grande arbre'', and ``the big girl - la grande fille'', are available, then the word ``big'' can be linked with the word ``grande'', but not with the words ``maison'' or ``la''.

Assuming an unknown one-to-one correspondence between words, estimating a joint probability distribution linking words in two languages is a missing data problem. Brown *et. al* [20] attack this problem using the Expectation Maximization (EM) algorithm (Dempster et al., [27]), where the missing data is correspondences.

# CHAPTER 3

# LINKING WORDS TO IMAGE REGIONS

In this chapter, we describe the process of linking words and image regions, as a form of machine translation. This is achieved by learning the correspondences between words and image regions using the Expectation Maximization (EM) algorithm. When the correspondences are found, words are predicted by each region allowing two helpful processes: region naming (i.e. predicting words for each region as a form of object recognition) and auto-annotation (predicting words for the image automatically).

Figure 3.1 presents the overview of the proposed system. The data set is separated into training and test data. First, the training images with associated text are tokenized in order to have word and blob tokens. Then, Expectation Maximization algorithm is applied on the training data to learn the probability table that translates blobs to words. On a given test image, words are predicted for each blob using the probability table. Then, the predicted words are used for region naming and auto-annotation.

In the following sections the proposed system will be explained in detail. The tokenization process to obtain word and blobs will be explained in Section 3.1. Then, in Section 3.2, the model for translating blobs to words will be presented and the application of the EM algorithm for the proposed model will be explained. The use of the probability table for word prediction will be described in Section 3.3.

Evaluation of the performance is not a straightforward process, since visually inspecting a large number of images is not practical. A set of alternative measurement strategies that can evaluate a large set of images will be presented in Section 3.4.

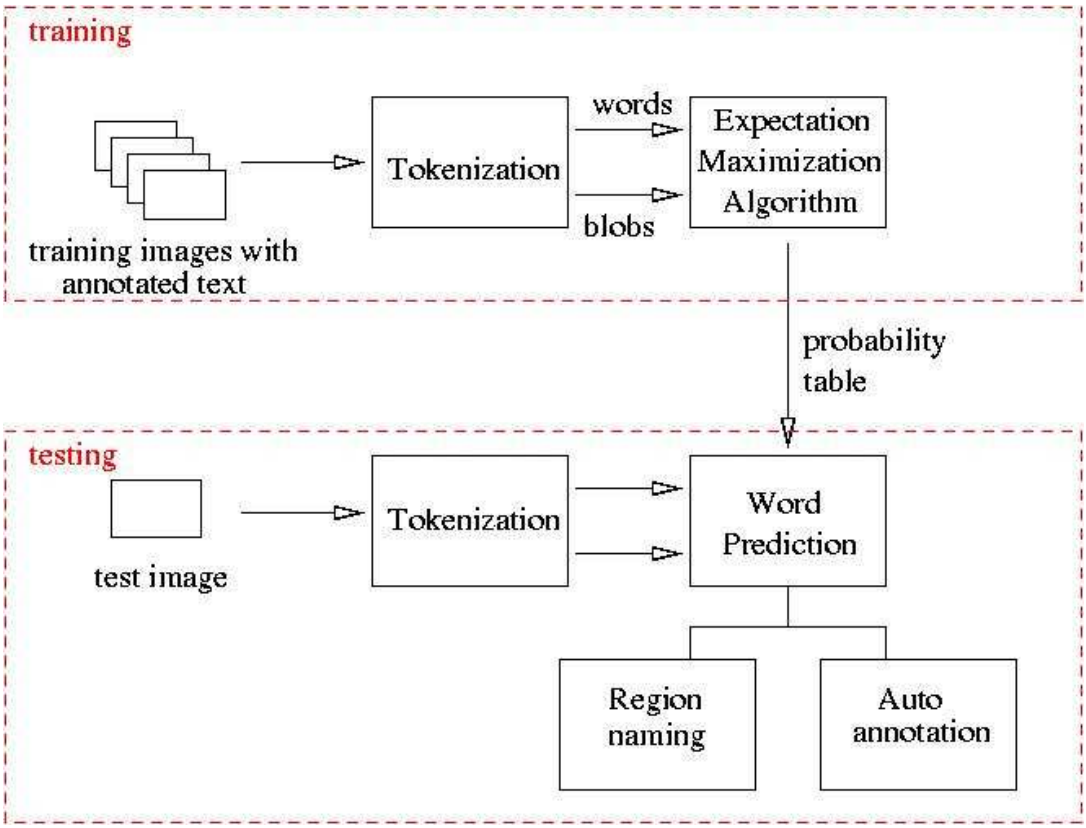



Figure 3.1: Overview of the proposed system.

The data set contain various sources of problems that effect the performance of the proposed system. Some strategies to attack such problems and to improve the performance will be described in Section 3.5.

Although the system is unsupervised, in the sense that the available data sets are used for learning, addition of a small number of supervised input is helpful. In Section 3.6 possible ways of integrating the supervised data into the system will be discussed.

## 3.1 Tokenization

In machine translation, a lexicon links a set of discrete objects (words in one language) onto another set of discrete objects (words in the other language). Therefore, in order to exploit the analogy with machine translation, both the images and the annotations need to be broken up into discrete items, which we call **tokens**.

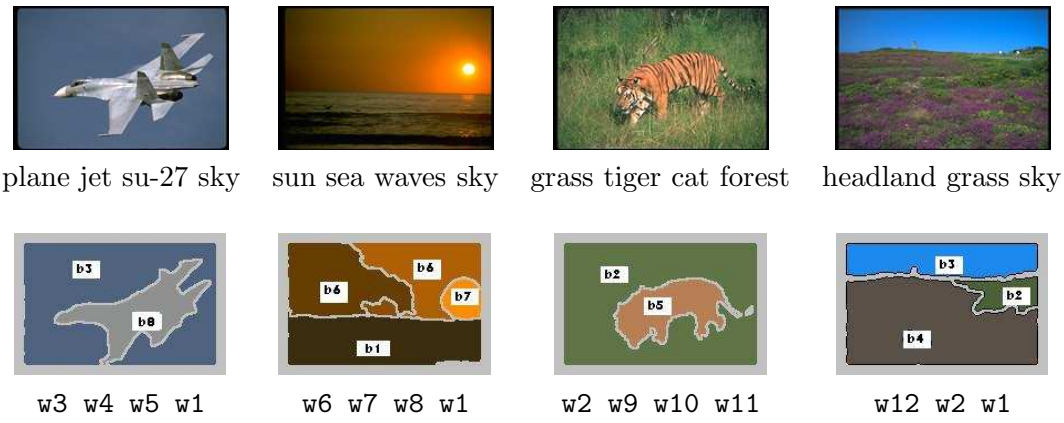



Figure 3.2: **Top:** Annotated images, **bottom** images after segmentation and tokenization. Image regions are replaced by blob tokens and keywords are replaced by word tokens.

In the Corel data set, the annotations consist of individual keywords, which can be directly used as the **word tokens**. For data sets, which are annotated in free text form, an appropriate language processing procedure can be applied to reduce the free text annotation into a set of word tokens.

In order to obtain the tokens for visual data, the images are first segmented into regions. Then, a set of features, such as color, texture, shape, size and position, are computed to represent each region. Finally, the regions are classified into region types (**blob tokens**) by clustering the feature space for all of the regions in the data. The clustering is performed using the k-means algorithm [30]. Each region is then assigned to the label of the cluster that it belongs to, that is to the corresponding **blob token**.

In Figure 3.2, an example for the data obtained after tokenization is shown. The image regions are replaced with the blob tokens, and the keywords are replaced with word tokens. In the rest of the thesis, for clarity, the terms **blobs** and **words** will be used instead of blob tokens and word tokens.

After tokenization, an aligned bitext, consisting of the blobs and the words for each image is obtained. The problem is then, to use the aligned bitext in training to construct a probability table linking blobs with words. The difficulty in learning the probability table is that, the data set does not provide explicit correspondence. This problem can be considered as a missing data problem [20] (where the missing data is the correspondences) and can be tackled using the Expectation Maximization (EM)

algorithm (Dempster et al., [27]). .

## 3.2 Expectation-Maximization algorithm for finding the correspondence between blobs and words

Brown *et. al.* [20] propose a set of models for statistical machine translation. The models aim to maximize the conditional probability density $p(\mathbf{f} \mid \mathbf{e})$, which is called the likelihood of translation $(\mathbf{f}, \mathbf{e})$, where $\mathbf{f}$ is a set of French words, and $\mathbf{e}$ is a set of English words. The simplest model (Model 1), assumes that all connections for each French position are equally likely. Following the analysis performed by de Freitas [31], this model is adapted to translate blobs to words, since there is no order relation among the blobs or words in the data. Figure 3.3 shows the notation used in the rest of the description.

$N$ : Number of items (image and its annotation)
$M_T$ : Number of words in the vocabulary
$L_T$ : Number of all blobs in the data set
$M_n$ : Number of words in the $n$-th annotation
$L_n$ : Number of blobs in the $n$-th image
$w_n$ : Words in the $n$-th annotation, $w_n = (w_{n1}, \ldots, w_{nj}, \ldots, w_{nM_n})$
$b_n$ : Blobs in the $n$-th image, $b_n = (b_{n1}, \ldots, b_{ni}, \ldots, b_{nL_n})$
$w^\star$ : A particular word
$b^\star$ : A particular blob
$t(w^\star \mid b^\star)$ : Probability of obtaining word $w^\star$ given blob $b^\star$
$a$ : Set of alignments
$a_n = a_{n1}, .....a_{nMn}$, such that $a_{nj} = i$ if $b_{ni}$ translates to $w_{nj}$
$p(a_{nj} = i)$ : Assignment probabilities
$\theta$ : Set of model parameters $\theta = (p(a_{nj} = i), t(w_{nj} \| b_{ni}))$

Figure 3.3: Nomenclature.

In our case, the goal is to maximize $p(\mathbf{w} \mid \mathbf{b})$, where $\mathbf{b}$ is a set of blobs and $\mathbf{w}$ is a set of words. Each word is aligned with the blobs in the image as illustrated in Figure 3.4. The alignments (referred as $\mathbf{a}$) provide a correspondence between each word and all the blobs. The model requires the sum over all possible assignments for each pair of aligned sentences, so that $p(\mathbf{w} \mid \mathbf{b})$ can be written in terms of the conditional probability density $p(\mathbf{w}, \mathbf{a} \mid \mathbf{b})$ as

$$p(\mathbf{w} \mid \mathbf{b}) = \sum_{\mathbf{a}} p(\mathbf{w}, \mathbf{a} \mid \mathbf{b}) \tag{3.1}$$

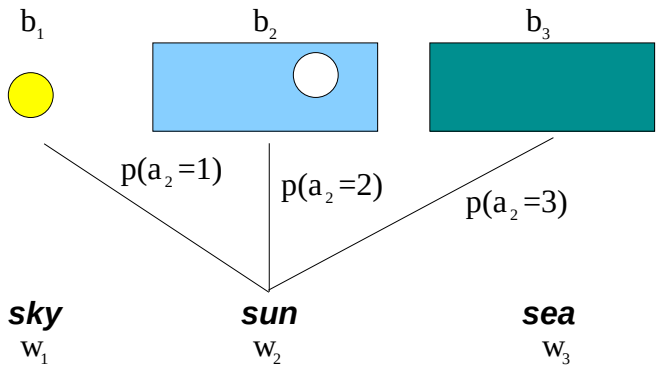



Figure 3.4: Each word is predicted with a certain probability by each blob. The alignments provide the correspondences between the words and the blobs.

If the image has $l$ blobs and $m$ words, the alignment is determined by specifying the values of $a_j$ for j from 1 to $m$. If the $j^{th}$ word is connected to the $i^{th}$ blob, then $a_j = i$. Then, $p(\mathbf{w} \mid \mathbf{b})$ can be written as

$$p(\mathbf{w} \mid \mathbf{b}) = \prod_{n=1}^{N} \sum_{a_{n_1}=1}^{L_n} \cdots \sum_{a_{n_{M_n}}=1}^{L_n} \prod_{j=1}^{M_n} p(a_{nj}) t(w = w_{nj} \mid b = b_{a_{nj}}) \tag{3.2}$$

where $t(w = w_{nj} \mid b = b_{a_{nj}})$ is the translation probability of the word $w_{nj}$ given the blob $b_{a_{nj}}$.

It is equivalent to the following simple mixture of distributions:

$$p(w \mid b) = \prod_{n=1}^{N} \prod_{j=1}^{M_n} \sum_{i=1}^{L_n} p(a_{nj} = i) t(w_{nj} \mid b_{(a_{nj}=i)}) \tag{3.3}$$

Maximizing this likelihood is difficult because of the sum inside the products; the sum represents marginalization over all possible correspondences.

The problem can be treated as a missing data problem were the missing data is the correspondences. It leads to the EM formulation which iterates between two steps to obtains the parameters $\theta$ maximizing the log-likelihood $Q^{\mathrm{ML}}$. For our case, $Q^{\mathrm{ML}}$ function, which is given by

$$Q^{\mathrm{ML}} = \sum_{n=1}^{N} \sum_{j=1}^{M_n} \sum_{i=1}^{L_n} p(a_{nj} = i \mid w_{nj}, b_{ni}, \theta^{(\mathrm{old})}) \log \left[ p(a_{nj} = i) t(w = w_{nj} \mid b = b_{(a_{nj}=i)}) \right]. \tag{3.4}$$

is maximized subject to the constraints $\sum_i p(a_{nj} = i) = 1$ for all words $j$ in all images, and $\sum_{w^\star} t(w^\star \mid b^\star) = 1$ for any word $w^\star$ and each blob $b^\star$ (From now on, $t(w = w_{nj} \mid b = b_{(a_{nj}=i)})$ will be written as $t(w_{nj} \mid b_{ni})$). This is accomplished by introducing the following Lagrangian,

$$\mathcal{L} = Q^{\mathrm{ML}} + \sum_{\alpha} \alpha \left( 1 - \sum_{i=1}^{L_n} p(a_{nj} = i) \right) + \sum_{b^\star} \beta_{b^\star} \left( 1 - \sum_{w^\star} t(w^\star \mid b^\star) \right) \tag{3.5}$$

and computing derivatives with respect to the multipliers $(\alpha, \beta)$ and the parameters $(p(a_{nj} = i), t(w^\star \mid b^\star))$. The results lead to the following E and M steps:

**E step:**

1. For each $n = 1, \ldots, N$, $j = 1, \ldots, M_n$ and $i = 1, \ldots, L_n$, compute

$$\widetilde{p}(a_{nj} = i \mid w_{nj}, b_{ni}, \theta^{(\mathrm{old})}) = p(a_{nj} = i) t(w_{nj} \mid b_{ni}) \tag{3.6}$$

2. Normalize $\widetilde{p}(a_{nj} = i \mid w_{nj}, b_{ni}, \theta^{(\mathrm{old})})$ for each image $n$ and word $j$

$$p(a_{nj} = i \mid w_{nj}, b_{ni}, \theta^{(\mathrm{old})}) = \frac{\widetilde{p}(a_{nj} = i \mid w_{nj}, b_{ni}, \theta^{(\mathrm{old})})}{\sum_{i=1}^{L_n} p(a_{nj} = i) t(w_{nj} \mid b_{ni})} \tag{3.7}$$

**M step:**

1. For each different pair $(b^\star, w^\star)$ appearing together in at least one of the images, compute

$$\widetilde{t}(w_{nj} = w^\star \mid b_{ni} = b^\star) = \sum_{n=1}^{N} \sum_{j=1}^{M_n} \sum_{i=1}^{L_n} p(a_{nj} = i \mid w_{nj}, b_{ni}, \theta^{(\mathrm{old})}) \delta_{(w^\star, b^\star)}(w_{nj}, b_{ni}) \tag{3.8}$$

   where $\delta_{(w^\star, b^\star)}(w_{nj}, b_{ni})$ is 1 if $b^\star$ and $w^\star$ appear in image and 0 otherwise.

2. Normalize $\widetilde{t}(w_{nj} = w^\star \mid b_{ni} = b^\star)$ to obtain $t(w_{nj} = w^\star \mid b_{ni} = b^\star)$.

The EM algorithm given above is used for training to learn a joint probability table of the words and blobs. Initially, the probability table is assigned to the co-occurrences of the blobs and words; and it is assumed that all the alignments are equally likely. Then, the above E and M steps are iterated to construct the final probability table that links the blobs and words with certain probabilities. This table consists of word posterior probabilities for each blob (i.e. for each word $w$ in the vocabulary, the conditional probability $p(w \mid b)$, given a blob $b$ ). The probability table is normalized, so that for each blob $b$ the sum of the word posterior probabilities is one. The word posterior probabilities $p(w \mid b)$, is then used to predict words for the test data.

## 3.3 Word prediction based on the probability table

The aim of word prediction is, for a given test image, finding the corresponding words for each blob in the image. For this purpose, first, the given test image is segmented into regions, and features are extracted from each region. The blob tokens are found by the nearest-neighbor method. The features of a region in the test image are compared with the features of cluster centers of the training data using the Euclidean distance. Then, the region is assigned to the closest cluster, and therefore to the corresponding blob token.

The word posteriors for each blob, supplied by the probability table, is then used to predict words for the test data. Based on the application, one can

- predict words for the image regions (region naming), or
- predict words for the whole image (auto-annotation).

Region naming is a model of object recognition; and auto-annotation helps to organize and access large collections of images. In the next subsections, we describe the strategies for region naming and auto-annotation.

### 3.3.1 Region naming strategy

The region naming refers to choosing the "best word" for the region using the posterior probabilities $p(w \mid b)$ for the corresponding blob $b$. In this study, we choose the word with the highest probability given the blob, as the "best word".

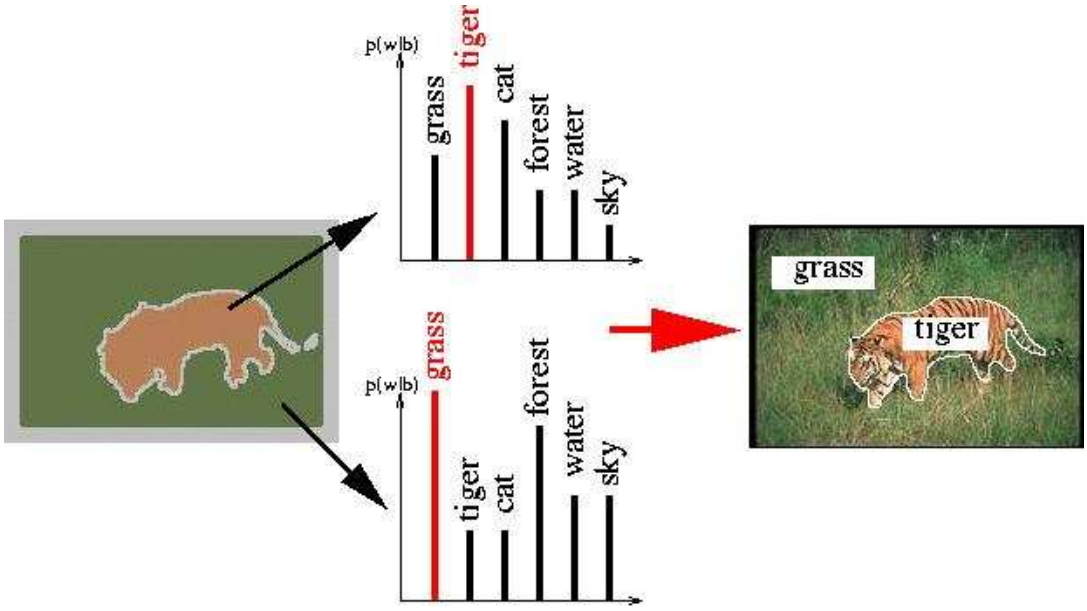



Figure 3.5: Region naming strategy. The word with the highest probability is predicted for each region.

Figure 3.5, shows an example to describe the region naming process. For each blob, the word with the highest probability is chosen and predicted for that blob.

### 3.3.2 Auto-annotation strategy

Manual annotation is labor-intensive and subjective to human. Therefore, auto-annotation of images is attractive, specifically for large image collections. This can be done by predicting words with high posterior probability given the image. In order to obtain the word posterior probabilities for the whole image, the word posterior probabilities of the regions in the image, provided by the probability table, are summed together. For the image I, we can write:

$$p(w \mid I) = \sum_{i=1}^{L} p(w \mid b_i) \tag{3.9}$$

where $b_i$'s are the blobs in the image.

Then, the sum of these word posterior probabilities are normalized to one. Figure 3.6 shows an example for obtaining the word posterior probabilities for the image.

In order to auto-annotate the images we predict $n$ words with the highest probability, where $n$ is a predefined number.

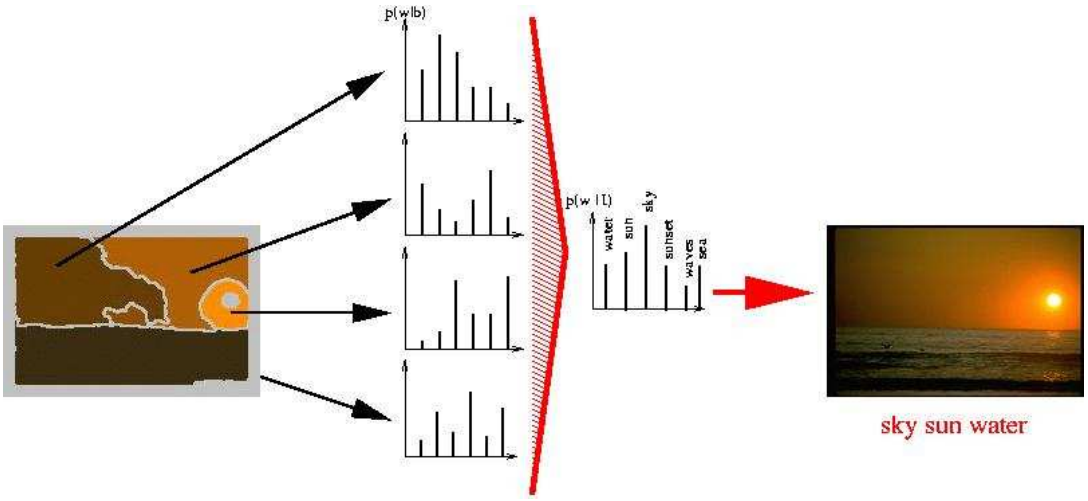



Figure 3.6: Auto-annotation strategy. Word posterior probabilities for the regions of the image are summed, and normalized. Then the best $n$ words with the highest probability are chosen to annotate the image.

## 3.4 Performance evaluation

In order to measure the accuracy of the word predictions some criterion is required. Performance should be evaluated based on the answers of the following questions.

- Does the system predict the correct words?
- Are the words on the right place?

Visual inspection of images provides an answer for both of the above questions. However, it requires human judgment and this form of manual evaluation is not practical to do for large number of images. In the following subsections, some alternative strategies are proposed to evaluate the results.

### 3.4.1 Evaluating the correspondence performance by a hand-labeled set

A hand-labeled set is used for evaluating the correspondence performance. On a relatively small number of images, each region is labeled manually with the vocabulary words defining the object (or part of an object). There are some difficulties in choosing the label words: Some words, like `landscape` and `valley`, normally apply to larger areas than the regions obtained by the system. Therefore, those words should not be used as label words. Some words like `pattern` can arguably be designated as correct

whenever it appears, but it should be scored as incorrect, since recognizing `pattern` is not particularly helpful. Also, regions should have a plausible visual connection to the chosen words. For example, the word `ocean` for `coral` should be judged incorrectly because the `ocean` is transparent. Due to the segmentation errors, some regions do not correspond to any particular object. Thus, those regions should not be linked with any vocabulary term. As a result, producing the labeled data set is clearly a time consuming and error prone process, and can be done only for a modest number of images.

### 3.4.2 Annotation as a proxy

There is a limit in the size of the pool that can be used for evaluating the correspondence performance. A less strict, but nonetheless informative, test is to determine the annotation performance. Using the annotation performance, only the first question can be answered. However, it is an automatic process allowing the evaluation of large number of images. Furthermore, it is reasonably expected that a method that cannot predict annotations accurately, is unlikely to predict correspondences well. Hence, annotation measures offer a plausible proxy.

Annotation performance is measured by comparing the predicted words with the words that actually present as an annotation keyword in the image. We use three measures for comparing the predictions with the actual data:

- Kullback-Leibler (KL) divergence between the predicted and target distributions,
- Normalized classification score,
- Word prediction measure.

In the following subsections, the measures will be explained in detail.

#### 3.4.2.1 Kullback-Leibler divergence between the predicted and target distribution

A measure for computing the difference between the actual and the desired probability distributions is the relative entropy, Kullback Leibler (KL) divergence [30].

In this study, we use the Kullback-Leibler (KL) divergence between the computed predictive distribution $p(W \mid B)$ (where B is the set of blobs in the image), and the target distribution p(W), to measure the "quality" of the word posterior distribution. The error contribution for one image with this measure, $E_{KL}$, is given by:

$$E_{KL} = \sum_{w \in vocabulary} p(w) log \frac{p(w)}{p(w \mid B)} \tag{3.10}$$

Unfortunately, the target distribution is not known. It can be assumed that, in the target distribution, the actual words are predicted uniformly, and all the other words are not predicted. Because division by $p(w \mid B)$ is potentially unstable, a small value (specifically the minimum of the empirical word distribution) is added and then renormalized.

To compute a combined measure for a group of images, we simply average the values for all the images in the set.

#### 3.4.2.2 Normalized classification score

There is a need for a loss function for the omitted words, but traditional zero-one loss is highly misleading, since the number of words that can be predicted is large (the size of the vocabulary). Therefore, the correct and incorrect classifications should be normalized. Specifically, a normalized classification score, which is first defined by Barnard [11], is computed as:

$$E_{NS} = r/n - w/(N - n) \tag{3.11}$$

where $N$ is the vocabulary size, $n$ is the number of actual words for the image, $r$ is the number of words predicted correctly, and $w$ is the number of words predicted incorrectly. This score gives a value of 0 for both predicting everything and predicting nothing, and 1 for predicting exactly the actual word set. The score for predicting exactly the complement of the actual word set is -1.

The number of words predicted, $r + w$, can be determined by the algorithm on a case by case basis. Thus one benefit of this measure over simply counting the number of correct words in a fixed number of guesses is that it can be used to reward a good estimate of how many words to predict.

#### 3.4.2.3 Word prediction measure

Yet another measure is the word prediction measure which is computed based on the best $n$ words as:

$$E_{PR} = r/n \tag{3.12}$$

where $r$ is the number of words predicted correctly. Thus, if there are three keywords, `sky`, `water`, and `sun`, then n=3, and we allow the model to predict 3 words for that image. The range of this score is clearly from 0 to 1.

## 3.5 Improving the performance

There are various sources of problems in Corel data set that affects word prediction performance. Due to poor segmentation, some of the regions do not correspond to objects. The feature selection and clustering steps affect he quality of the blobs, therefore the quality of the word prediction. In some cases, more than one word is used for annotating a single region, or a single word represent more than one region. Also, there are some problems due to the annotations: there are inequalities between the occurrence frequencies of the annotation words (while the word `water` highly occurs in the annotations, the word `tiger` only occurs rarely) causing inequalities in the prediction rates; some words are always used together in the annotations, (such as `cat` and `tiger`) or a compound word is divided into two separate words (such as `polar` and `bear`), therefore it is not possible to distinguish such words; and some words do not correspond to an object (either used to define a larger area, such as `Scotland` or it is not an object such as `pattern`), therefore it is not possible to predict such words.

In order to handle some of the problems mentioned above, we propose the following improvement strategies:

- Refusing to predict,
- Retraining on reduced vocabulary,
- Clustering indistinguishable words.

In the following subsections, the encountered problems and the strategies as a solution to these problems will be explained in detail.

### 3.5.1 Refusing to predict

In most cases, it is not possible to obtain a one-to-one map between regions and words, since the number of regions in the image is different than the number of annotated keywords. We require that all the regions are linked to words; that is, there is no option of deciding that a region corresponds to no word. The problem, in principle, can be handled by appending a special **NULL word** to the set of words, and a special **NULL blob** to the set of blobs for each image. It is a traditional solution in the machine translation literature [55]; the tendency of single words in some languages to generate more than one word in others (a property referred to as "fertility") can be modeled explicitly in this framework. In our limited experience, such models are not easy to fit to our data sets, because of a tendency to link every word with a NULL blob or link every blob with a NULL word.

A simple strategy that offers some benefits of directly modeling NULL words is to refuse to predict an annotation when the annotation with the highest probability given the region is below some predefined threshold. It discourages predictions by regions whose identity is moot. It can be easily performed by requiring that $p(word \mid blob) > threshold$, to make a prediction; which is equivalent to assigning a NULL word to any blob whose best predicted word lies below this threshold.

### 3.5.2 Retraining on a refined vocabulary

Mostly due to the inequalities in the occurrence frequencies of the words some of the words never have the chance to be predicted. Also, the process of refusing to predict, prunes the vocabulary, since some words may never be predicted with sufficiently high probability. In turn, this suggests that once a threshold has been determined, a new lexicon can be fitted using only the reduced vocabulary. In practice, this is advantageous, since re-assigning the probability values "stolen" by the unpredicted words, improves correspondence estimates and therefore the quality of the lexicon.

### 3.5.3 Clustering indistinguishable words

The annotations include some words that are visually indistinguishable, such as `cat` and `tiger`, or `train` and `locomotive`. Also, some words are visually distinguishable *in principle*, but the feature set used in this study does not provide separation. A good example is `eagle` and `jet` where both occur as large dark regions of roughly the same shape in aerial view. Finally, some words always occur together, specifically when one word is a modifier (like `polar` for `bear`) , or because of the intrinsic relation between the concepts ( for example, either `mare` or `foals` often occur with `horses` )

As a result, there are some words which are not distinguishable based on the particular blob data used. This suggests clustering the similar words. The similarity between two words is computed using the Euclidean distance between the conditional probability of blobs, given the words ($p(b \mid w)$). This implies that two words will be similar if they generate similar image blobs at similar frequencies. The similarity matrix of all the words in the vocabulary is then clustered using the graph cut idea, specifically using the Normalized Cuts method proposed by Shi and Malik [72]. Then, the words in a cluster are merged to obtain a new vocabulary. The system is retrained using the merged words.

## 3.6 Integrating labeled data to the system

Until now, the proposed system is purposely designed to learn from unsupervised data. However, there are limits to what can be done without supervision.

Missing correspondence information generates symmetries in the incomplete data log-likelihood. For example, if `horses` and `grass` always appear together, (where green blob appears with brown blob) but not in any other form, then, it is not possible to determine which region is `horses` and which region is `grass`, since the training will not learn further than the co-occurrences. However, labeling a small number of regions manually will break this symmetry, and cause a substantial change in the model.

Therefore, we propose integrating supervised data into the system. It is not only used to break the symmetries but also for the other cases where the available data is not sufficient. Specifically there are limits in feature extraction and clustering for representing visual data without using the available word information, and the supervised data is used to supply such a word knowledge.

The key is to develop strategies which require only a small amount of the supervised data, since it is expensive to collect. For our purposes, supervised data is obtained by manually attaching word labels to image regions. Since, it is not feasible to create such data in large quantities, it needs to be used jointly with unsupervised data.

In the following subsections, we describe the methods to use a small quantity of supervised data for

- selecting appropriate representations of image information, and
- breaking correspondence ambiguities present in unsupervised data.

### 3.6.1 Using labeled data for clustering

Supervised data provides a small set of image regions where the labels are known. Therefore, it is possible to integrate the visual information obtained from the regions with the available word information obtained from the labels.

On a small set of manually labeled data, it is possible to perform the clustering based on the labels. A set of regions share a single label. Therefore, it is possible to construct one cluster per word label. Linear discriminant analysis is applied on the features used to represent regions. This yields a feature space, within which we have a small number of labeled elements.

Unlabeled data is then clustered by finding nearest neighbors in the feature space. This means that an unlabeled image region is assigned to the cluster belonging to the closest labeled image region.

There are two possibilities:

- First, a nearest neighbor classifier, where an unlabeled image region is given the *label* of the closest labeled image region, is built. However, this method suffers from the relatively small amount of labeled data available.
- Second, unlabeled image regions are assigned to the *cluster* of the closest labeled image region, but the joint probability of image clusters and words are still learned from data using EM. This has the considerable advantage that the unsupervised word and image data are still used.

### 3.6.2 Using labeled data for breaking symmetries

It is difficult to learn region-word correspondences from annotated images when the entropy of the annotations is not high. In the extreme case, two words always appear together in annotations, and therefore the incomplete data log-likelihood has a symmetry For example, the `horses` could be green and the `grass` could be brown, or the other way round, since `horses` and `grass` always appear together. Even small amounts of labeled data should break this symmetry.

The goal is to use a small amount of labeled and a large number of unlabeled data together. There are some studies that combine labeled and unlabeled data ([19, 59]) In this study, manual labeling is incorporated into the system by fixing the correspondences that are known between image regions and words, and filling in missing correspondences with EM, as before. This is performed, by setting the alignment probabilities for each of the label word-blob pairs (a labeled blob and the annotation word used to label that blob) and the remaining alignments to zero on the labeled data. Fixing the alignments probabilities in such a way, affects the training performance, since it forces the similar alignments in the unlabeled data to have similar values.

## 3.7 Summary of the chapter

In this chapter, we present the method for linking blobs and words as a process of machine translation; describe two strategies to use the result of the linking process, namely: region naming and auto-annotation; and discuss possible ways of analyzing and improving the performance of the system. In the next chapter, we present the extensive experimental results and discuss the strengths and the weakness of the system.

# CHAPTER 4

# EXPERIMENTS

In this chapter, a range of experiments are performed to asses the strengths and weaknesses of the proposed method. The chapter consists of six sections. First, we describe the image data set used in our experiments. Then, we design a set of experiments and present the results. The performance of the proposed method is assessed against two other strategies: empirical word densities and co-occurrences of blobs and words. The effects of certain parameters of the method on the performance is discussed and analyzed. Then some ideas for improving the performance is proposed. Finally, we discuss and analyze integration of a small number of supervision into the system.

## 4.1 Data set and input representation

### 4.1.1 Data set

In this study, the Corel data set [2], a large collection of stock photographs taken by professional photographers, is used. These photographs capture real-world scenes, creating a diverse set of images.

The Corel data set is commercially available in CD volumes. Some of the volume names and the topics of the CD's in these volumes are:

- **Animals and nature :** African antelopes, underwater reefs, barnyard animals, hawks and falcons, North American wildlife.
- **Leisure, transportation and architecture:** Mediterranean cruise, skiing in Switzerland, commercial construction, doors of San Francisco, WWII planes.
- **Places around the world:** Arizona desert, western Canada, Korea, Czech Republic, Turkey.
- **Scenic sites:** American national parks, old Singapore, The Big Apple, Mayan Aztec ruins.
- **Cities and countries:** Bonny Scotland, Greek isles, Russia, Georgia, and Armenia.
- **Animal life:** African specialty animals, backyard wildlife, wildlife babies, bald eagles.
- **Nature:** Annuals for American gardens, North American wildflowers, perennials in bloom, winter.
- **People, Places and things:** Candy backgrounds, great silk road, people of the world, oil paintings, religious stained glass.
- **Occupations and leisure:** Beautiful women, martial arts, royal military parades, steam trains.
- **Backgrounds, scenery and food:** Barbecue and salads, English pub signs, international fireworks.
- **Land and sea:** Fungi, mountains Of Eurasia, under the Red Sea, flowers close-up.
- **Textures, backgrounds and objects:** Beverages, patterns in stone, sunsets around the world.
- **Travel destinations:** Jersey Channel Islands, beautiful Bali, portrait of Italy.

The complete list of CD's in the data set is available at
`http://www.corel.com/products/clipartandphotos/photos/VOLUMES.HTM`.

Each CD contains 100 photographs on a relatively specific topic such as "aircraft". In this study, 160 CD's that are available with annotations are used. Each image is manually annotated with a set of keywords. Figure 4.1 shows sample images with their annotated keywords.

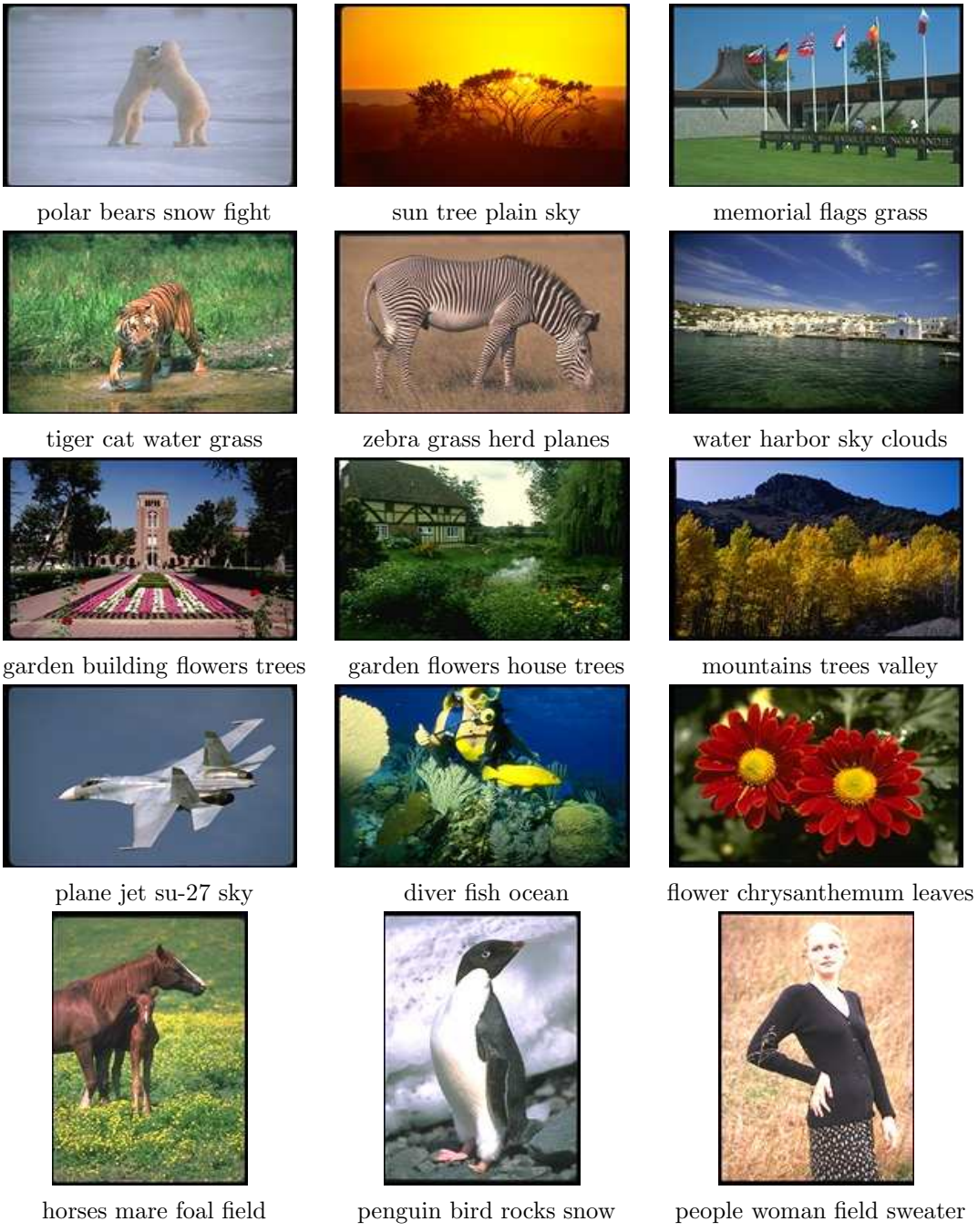



Figure 4.1: Sample annotated images from the Corel data set.

Ten different *experimental data sets* are created by randomly choosing 80 CD's for each set. Each experimental data set is further split up into *training* and *standard test* sets, containing 75% and 25% of the images respectively. The images from the remaining CD's form the *novel test* set for a particular experimental data set.

In the full set (i.e. in the 160 CD's used for the experiments), the vocabulary size (the number of different words used in annotations) is counted as 437. However, none of the experimental data sets contain all the words in this vocabulary, making the vocabulary size different for each set. Table 4.1 lists the number of images in the training, standard test and novel test sets, and the size of the vocabulary for each of the ten experimental data sets. Unless otherwise stated, the first experimental data set (labeled 001) is used throughout the rest of this chapter.

Table 4.1: The number of images in the training, standard test and novel test sets, and the size of the vocabulary are listed for each of the ten experimental data sets.

| set | training | standard test | novel test | vocabulary size |
|---|---|---|---|---|
| 001 | 5188 | 1744 | 6834 | 153 |
| 002 | 5241 | 1783 | 6737 | 164 |
| 003 | 5289 | 1717 | 6754 | 154 |
| 004 | 5287 | 1746 | 6804 | 162 |
| 005 | 5273 | 1741 | 6833 | 160 |
| 006 | 5192 | 1737 | 6930 | 162 |
| 007 | 5266 | 1747 | 6902 | 174 |
| 008 | 5266 | 1724 | 6874 | 168 |
| 009 | 5239 | 1801 | 6844 | 173 |
| 010 | 5197 | 1761 | 6660 | 144 |

### 4.1.2 Word information

Each image in the Corel data set is annotated with 3-5 words. Figure 4.2 plots the occurrence frequencies of the words in the vocabulary of the first experimental data set. The plot shows that the occurrence frequencies of the words lie in a wide range. Some common words, such as `sky`, `water`, `people`, have a high occurrence rate, whereas more specific words, such as `tiger`, `rhino`, `windows`, appear seldom. Table 4.2 shows that the common words tend to have similar frequencies in all the experimental data sets.

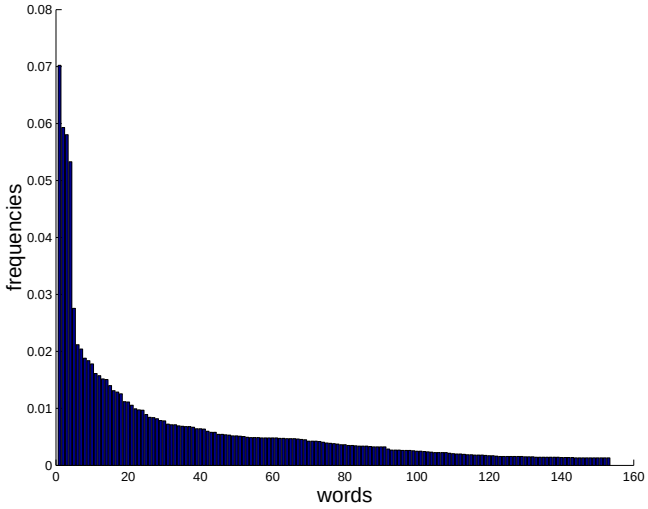



Figure 4.2: Word frequencies in the training set of the first experimental data set. As the plot shows, the occurrence frequencies of the words lie in a wide range, which results in a wide range of prediction performances.

Table 4.2: Occurrence frequencies of some common words in the training sets of ten experimental data sets. The '-' sign indicates the absence of the word. The common words have similar occurrence frequencies in all the experimental data sets.

| word | 001 | 002 | 003 | 004 | 005 | 006 | 007 | 008 | 009 | 010 |
|---|---|---|---|---|---|---|---|---|---|---|
| water | 0.07 | 0.07 | 0.08 | 0.07 | 0.09 | 0.08 | 0.06 | 0.07 | 0.07 | 0.09 |
| sky | 0.06 | 0.06 | 0.05 | 0.05 | 0.06 | 0.06 | 0.05 | 0.07 | 0.06 | 0.07 |
| tree | 0.06 | 0.06 | 0.06 | 0.06 | 0.05 | 0.05 | 0.06 | 0.06 | 0.05 | 0.06 |
| people | 0.05 | 0.04 | 0.05 | 0.05 | 0.05 | 0.06 | 0.06 | 0.05 | 0.05 | 0.04 |
| buildings | 0.03 | 0.03 | 0.02 | 0.02 | 0.03 | 0.03 | 0.03 | 0.03 | 0.02 | 0.02 |
| grass | 0.02 | 0.03 | 0.03 | 0.02 | 0.03 | 0.02 | 0.03 | 0.03 | 0.03 | 0.02 |
| clouds | 0.02 | 0.02 | 0.01 | 0.01 | 0.02 | 0.02 | 0.02 | 0.02 | 0.02 | 0.02 |
| rock | 0.02 | 0.03 | 0.03 | 0.02 | 0.03 | 0.02 | 0.02 | 0.02 | 0.02 | 0.02 |
| birds | 0.02 | 0.01 | 0.01 | 0.02 | 0.00 | 0.01 | 0.02 | 0.02 | 0.01 | 0.01 |
| mountain | 0.02 | 0.02 | 0.03 | 0.02 | 0.02 | 0.02 | 0.02 | 0.02 | 0.02 | 0.03 |
| stone | 0.02 | 0.01 | 0.01 | 0.02 | 0.01 | 0.01 | 0.01 | 0.00 | 0.01 | 0.01 |
| snow | 0.02 | 0.02 | 0.02 | 0.02 | 0.02 | 0.02 | 0.02 | 0.03 | 0.02 | 0.03 |
| street | 0.02 | 0.01 | 0.01 | 0.01 | 0.01 | 0.01 | 0.01 | 0.01 | 0.01 | 0.01 |
| plane | 0.02 | 0.02 | 0.01 | 0.01 | 0.00 | 0.01 | 0.00 | 0.01 | 0.01 | 0.01 |
| flowers | 0.01 | 0.02 | 0.03 | 0.02 | 0.03 | 0.02 | 0.01 | 0.02 | 0.02 | 0.03 |
| pattern | 0.01 | 0.01 | 0.01 | 0.01 | 0.00 | 0.01 | 0.01 | 0.01 | 0.00 | 0.01 |
| jet | 0.01 | 0.01 | 0.00 | 0.00 | 0.00 | 0.01 | - | 0.01 | 0.01 | 0.01 |
| texture | 0.01 | 0.01 | 0.00 | - | 0.00 | 0.01 | 0.00 | 0.01 | 0.00 | 0.01 |
| fish | 0.01 | 0.01 | 0.02 | 0.01 | 0.01 | 0.01 | 0.01 | 0.01 | 0.02 | 0.01 |
| coast | 0.01 | 0.01 | 0.01 | 0.01 | 0.01 | 0.01 | 0.00 | - | 0.00 | 0.01 |
| boats | 0.01 | 0.01 | 0.01 | 0.01 | 0.01 | 0.01 | 0.01 | 0.01 | 0.01 | 0.01 |
| beach | 0.01 | 0.01 | 0.01 | 0.01 | 0.01 | 0.01 | 0.01 | 0.01 | 0.01 | 0.01 |

The occurrence frequencies for all the words in the vocabulary of the first experimental data set are listed in Table 4.3 for the training, standard test and novel test sets. The table shows that the frequencies of the common words are similar in all the sets, but large discrepancies exist between the word frequencies in the training set and the novel test set in the rest of the vocabulary.

Table 4.3: For each word in the vocabulary of the first experimental data set, occurrence frequencies in training, standard and novel test sets are listed. Total number of words used for annotation is, 16006 in the training, 5330 in the standard test, and 18017 in the novel test set. Although, the frequencies are similar for the common words in all of the three sets, for the rest of the vocabulary there is a disparity between the training and novel test sets, since the CD's used in these sets are different. Seldom words in the training set may appear more frequently in the novel test set (see `coral, valley` ), and frequent words in training set may occur rarely (or may not occur) in novel test set (see `vegetables, owl`).

| word | training | standard | novel | word | training | standard | novel |
|---|---|---|---|---|---|---|---|
| water | 0.070 | 0.074 | 0.092 | sky | 0.059 | 0.059 | 0.072 |
| tree | 0.058 | 0.057 | 0.070 | people | 0.053 | 0.057 | 0.064 |
| buildings | 0.028 | 0.024 | 0.028 | grass | 0.021 | 0.024 | 0.042 |
| clouds | 0.020 | 0.020 | 0.015 | rock | 0.019 | 0.022 | 0.029 |
| birds | 0.018 | 0.020 | 0.012 | mountain | 0.018 | 0.018 | 0.035 |
| stone | 0.016 | 0.016 | 0.005 | snow | 0.016 | 0.017 | 0.033 |
| street | 0.015 | 0.011 | 0.009 | plane | 0.015 | 0.015 | 0.003 |
| flowers | 0.014 | 0.014 | 0.040 | pattern | 0.013 | 0.011 | 0.005 |
| jet | 0.013 | 0.012 | 0.000 | texture | 0.013 | 0.010 | 0.000 |
| fish | 0.011 | 0.014 | 0.009 | coast | 0.011 | 0.010 | 0.002 |
| boats | 0.011 | 0.013 | 0.014 | beach | 0.010 | 0.008 | 0.007 |
| vegetables | 0.010 | 0.009 | 0.000 | ground | 0.010 | 0.008 | 0.005 |
| hills | 0.009 | 0.008 | 0.005 | cat | 0.008 | 0.008 | 0.016 |
| leaves | 0.008 | 0.009 | 0.018 | car | 0.008 | 0.008 | 0.002 |
| walls | 0.008 | 0.006 | 0.006 | closeup | 0.008 | 0.008 | 0.010 |
| ocean | 0.007 | 0.010 | 0.010 | ruins | 0.007 | 0.006 | 0.007 |
| close-up | 0.007 | 0.007 | 0.018 | temple | 0.007 | 0.005 | 0.002 |
| house | 0.007 | 0.007 | 0.005 | sand | 0.007 | 0.005 | 0.012 |
| head | 0.007 | 0.005 | 0.006 | plants | 0.007 | 0.007 | 0.022 |
| woman | 0.006 | 0.006 | 0.006 | gardens | 0.006 | 0.007 | 0.009 |
| food | 0.006 | 0.006 | 0.002 | helicopter | 0.006 | 0.002 | 0.000 |
| statues | 0.006 | 0.007 | 0.004 | bridge | 0.006 | 0.006 | 0.003 |
| sun | 0.005 | 0.008 | 0.002 | shore | 0.005 | 0.007 | 0.001 |
| elephants | 0.005 | 0.006 | 0.001 | reefs | 0.005 | 0.008 | 0.004 |
| tracks | 0.005 | 0.005 | 0.002 | field | 0.005 | 0.006 | 0.019 |
| waves | 0.005 | 0.006 | 0.004 | lion | 0.005 | 0.006 | 0.000 |
| Scotland | 0.005 | 0.004 | 0.000 | windows | 0.005 | 0.002 | 0.007 |
| mushrooms | 0.005 | 0.004 | 0.005 | fungus | 0.005 | 0.004 | 0.005 |

Table 4.3: Continued.

| word | training | standard | novel | word | training | standard | novel |
|---|---|---|---|---|---|---|---|
| sunset | 0.005 | 0.006 | 0.006 | pumpkins | 0.005 | 0.004 | 0.000 |
| night | 0.005 | 0.005 | 0.002 | ice | 0.005 | 0.004 | 0.011 |
| cougar | 0.005 | 0.004 | 0.001 | bears | 0.005 | 0.006 | 0.008 |
| animal | 0.005 | 0.006 | 0.001 | insect | 0.005 | 0.005 | 0.005 |
| crystal | 0.005 | 0.004 | 0.001 | branch | 0.005 | 0.006 | 0.002 |
| background | 0.005 | 0.004 | 0.000 | market | 0.005 | 0.004 | 0.002 |
| owl | 0.004 | 0.006 | 0.000 | horizon | 0.004 | 0.004 | 0.002 |
| forest | 0.004 | 0.004 | 0.012 | coral | 0.004 | 0.005 | 0.009 |
| wolf | 0.004 | 0.004 | 0.001 | hawk | 0.004 | 0.003 | 0.000 |
| skis | 0.004 | 0.003 | 0.000 | wildlife | 0.004 | 0.004 | 0.000 |
| sea | 0.004 | 0.004 | 0.003 | textile | 0.004 | 0.003 | 0.000 |
| sculpture | 0.004 | 0.004 | 0.001 | nest | 0.004 | 0.004 | 0.001 |
| trunk | 0.003 | 0.003 | 0.001 | cactus | 0.003 | 0.004 | 0.001 |
| flight | 0.003 | 0.001 | 0.000 | pillars | 0.003 | 0.005 | 0.001 |
| hunter | 0.003 | 0.005 | 0.000 | desert | 0.003 | 0.005 | 0.004 |
| church | 0.003 | 0.004 | 0.003 | mane | 0.003 | 0.004 | 0.000 |
| lizard | 0.003 | 0.002 | 0.003 | costume | 0.003 | 0.002 | 0.001 |
| city | 0.003 | 0.005 | 0.004 | hats | 0.003 | 0.003 | 0.001 |
| shop | 0.003 | 0.002 | 0.002 | face | 0.003 | 0.003 | 0.001 |
| arch | 0.003 | 0.002 | 0.001 | reflection | 0.003 | 0.002 | 0.006 |
| display | 0.003 | 0.003 | 0.001 | cliff | 0.003 | 0.003 | 0.002 |
| palace | 0.003 | 0.003 | 0.003 | town | 0.002 | 0.004 | 0.001 |
| columns | 0.002 | 0.003 | 0.001 | gun | 0.002 | 0.004 | 0.000 |
| river | 0.002 | 0.002 | 0.004 | runway | 0.002 | 0.001 | 0.000 |
| woods | 0.002 | 0.003 | 0.002 | village | 0.002 | 0.001 | 0.001 |
| doors | 0.002 | 0.002 | 0.002 | designs | 0.002 | 0.001 | 0.000 |
| tower | 0.002 | 0.002 | 0.003 | island | 0.002 | 0.001 | 0.001 |
| pyramid | 0.002 | 0.003 | 0.000 | entrance | 0.002 | 0.002 | 0.001 |
| road | 0.002 | 0.003 | 0.006 | dog | 0.002 | 0.002 | 0.001 |
| art | 0.002 | 0.002 | 0.000 | leaf | 0.002 | 0.002 | 0.001 |
| herd | 0.002 | 0.002 | 0.001 | black | 0.002 | 0.002 | 0.001 |
| polar | 0.002 | 0.002 | 0.006 | zebra | 0.002 | 0.002 | 0.000 |
| restaurant | 0.002 | 0.001 | 0.001 | horse | 0.002 | 0.001 | 0.014 |
| turn | 0.002 | 0.001 | 0.000 | snake | 0.002 | 0.002 | 0.002 |
| shadows | 0.002 | 0.003 | 0.002 | museum | 0.002 | 0.001 | 0.001 |
| harbor | 0.002 | 0.002 | 0.001 | fence | 0.002 | 0.002 | 0.003 |
| branches | 0.002 | 0.003 | 0.003 | valley | 0.001 | 0.002 | 0.007 |
| formation | 0.001 | 0.001 | 0.001 | architecture | 0.001 | 0.002 | 0.000 |
| smoke | 0.001 | 0.000 | 0.002 | ships | 0.001 | 0.001 | 0.001 |
| saguaro | 0.001 | 0.001 | 0.000 | roofs | 0.001 | 0.002 | 0.001 |
| perch | 0.001 | 0.001 | 0.000 | courtyard | 0.001 | 0.002 | 0.001 |
| castle | 0.001 | 0.002 | 0.002 | seals | 0.001 | 0.001 | 0.001 |
| prototype | 0.001 | 0.001 | 0.000 | outside | 0.001 | 0.000 | 0.000 |
| detail | 0.001 | 0.002 | 0.000 | tables | 0.001 | 0.002 | 0.003 |
| shrine | 0.001 | 0.001 | 0.001 | paintings | 0.001 | 0.002 | 0.001 |
| light | 0.001 | 0.001 | 0.004 | kauai | 0.001 | 0.002 | 0.000 |
| formula | 0.001 | 0.002 | 0.000 | f-16 | 0.001 | 0.002 | 0.000 |
| dunes | 0.001 | 0.002 | 0.000 | candy | 0.001 | 0.002 | 0.000 |
| bay | 0.001 | 0.001 | 0.001 | | | | |

### 4.1.3 Segmentation, feature extraction and clustering

Each image is segmented using Normalized Cuts algorithm [72]. Figure 4.3 shows the samples from the segmentation results using this segmentation algorithm on the Corel data set. Similar to most other segmentation algorithms, the Normalized Cuts algorithm has the tendency to produce small regions. We choose the 8 largest regions in each image.

For the selected regions, the following set of basic features are computed:

- **Size** is represented by the portion of the image covered by the region.
- **Position** is represented using the coordinates of the region center of mass normalized by the image dimensions.
- **Color** is represented using the average and standard deviation of (R,G,B), (L,a,b) and (r=R/ (R+G+B), g=G/ (R+G+B)) over the region.
- **Texture** is represented using the average and variance of 16 filter responses. We use 4 difference of Gaussian filters with different sigmas, and 12 oriented filters, aligned in 30 degree increments.
- **Shape** is represented by the ratio of the area to the perimeter squared, the moment of inertia (about the center of mass), and the ratio of the region area to that of its convex hull.

The length of the feature vector is 30. It consists of:

- area, x, y, boundary/area, convexity, moment-of-inertia (6)
- average RGB (3)
- RGB stdev (3)
- average L*a*b (3)
- L*a*b* stdev (3)
- mean oriented energy, 30 degree increments (12)

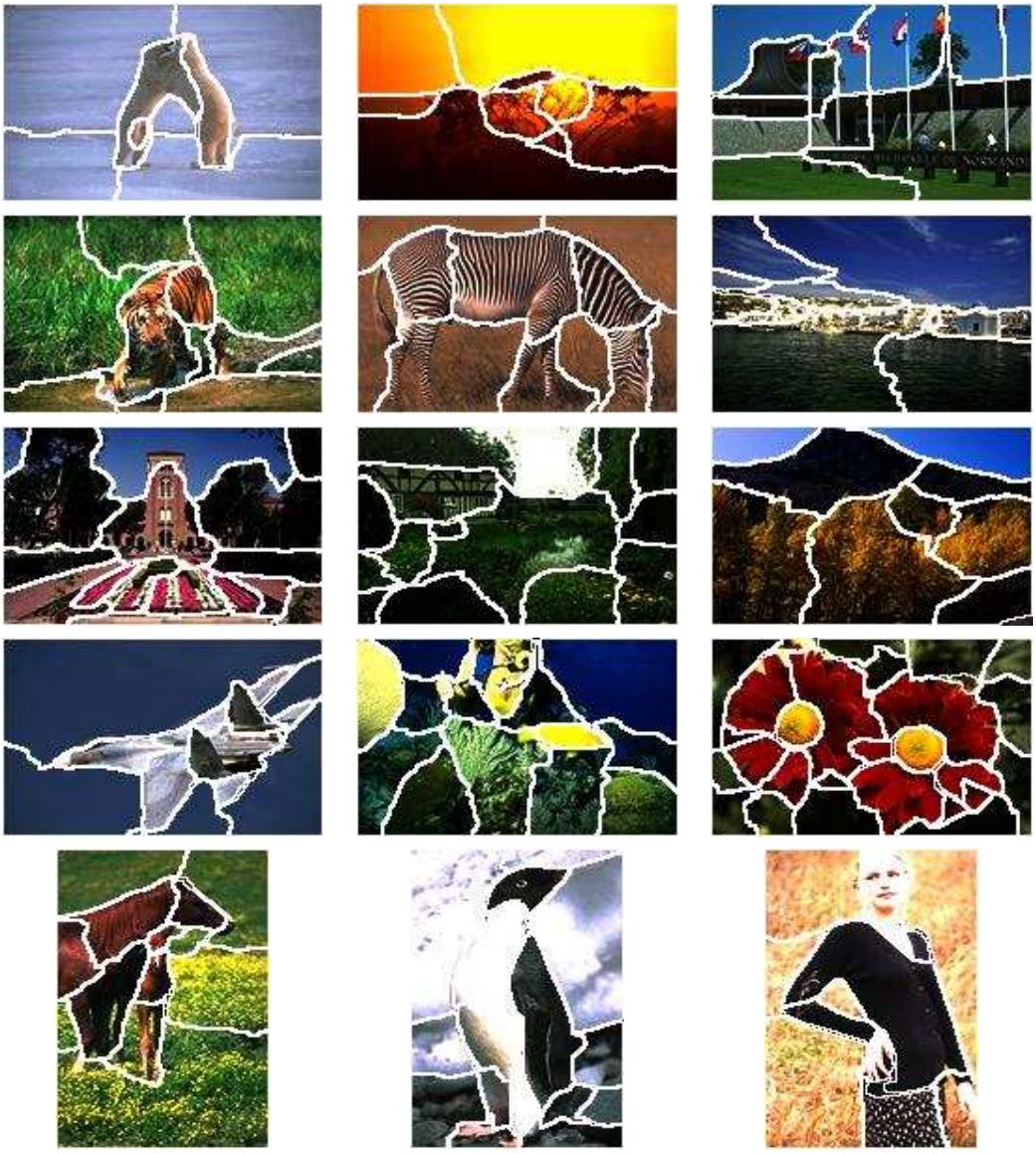


Figure 4.3: Sample outputs of Normalized Cuts segmentation.

In order to map the feature vectors onto a finite number of blob tokens, first the feature vectors of the regions, obtained from all the images in the training set are shifted and scaled to have zero mean and unit variance. Then these vectors are clustered using the k-means algorithm[30], with the total number of clusters $k = 500$. Figure 4.4 shows the distribution of blobs for the training and standard test sets of the first experimental data set. In Figure 4.5 some examples of blobs are given. As the figures show, in most of the cases k-means groups the similar regions into the same cluster. However, different regions corresponding to different type of objects may be put into the same cluster, or visually similar regions may be put into different clusters. The result of clustering affects the prediction performance as will be discussed in Section 4.4.2.

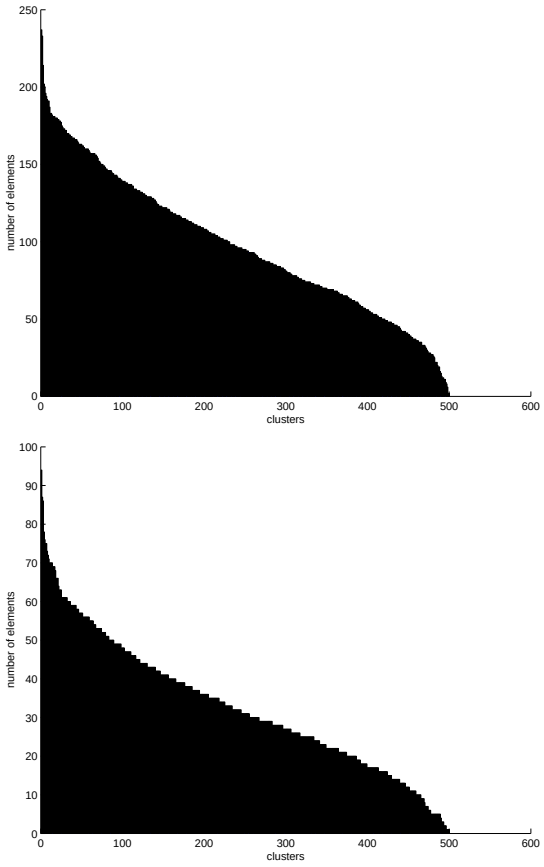



Figure 4.4: Occurrence counts for the blobs in training and standard test sets of the first experimental data set, for $k = 500$. As the graphs show, the frequencies of blobs lie in a large range, and they are similar for the training and standard test tests.

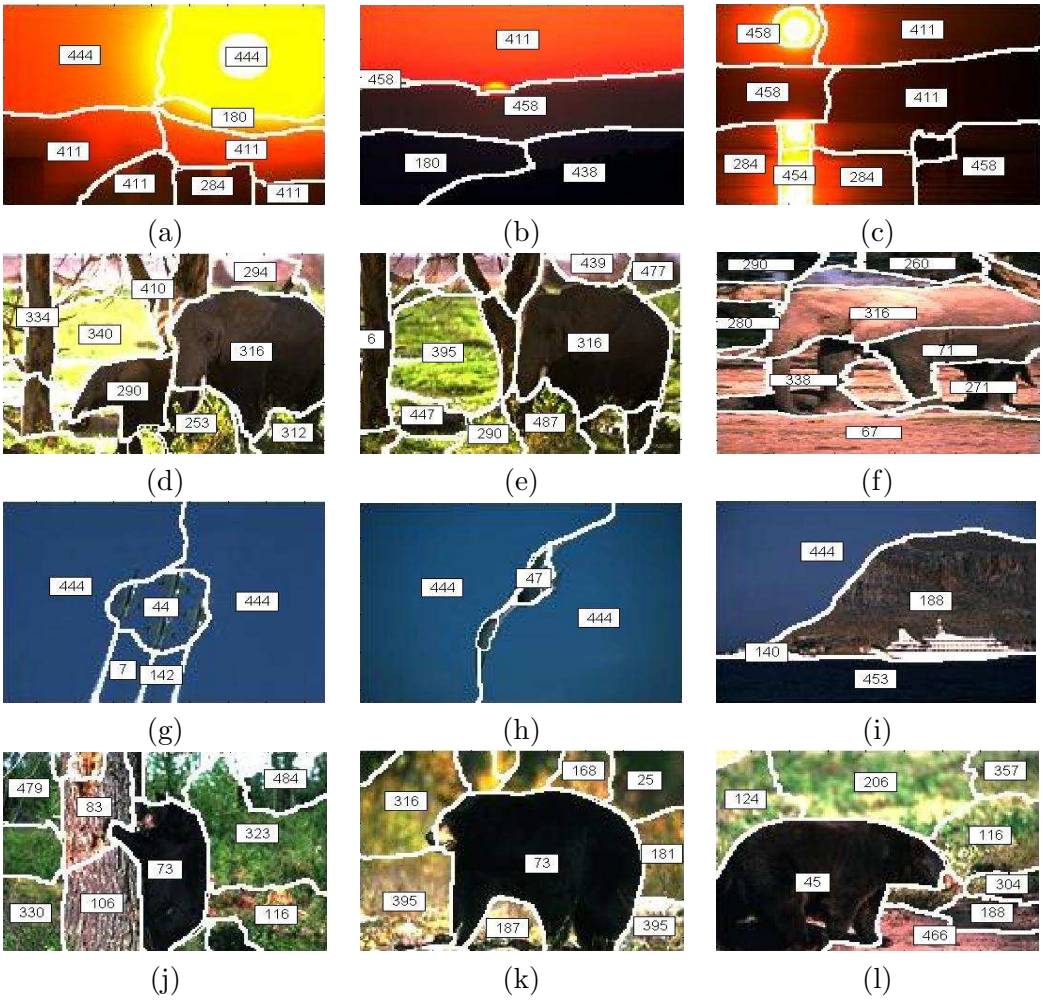



Figure 4.5: Some results from clustering. In images (a), (b), (c), the regions with similar colors are clustered together under the clusters 411 and 458. Note that, the regions grouped under the cluster 458 correspond to both sky, as in (b), and to water, as in (c). This is a degrading factor for word prediction. In images (j) and (k), the bears are labeled with the same cluster 73, but in (l) a similar region - according to color and texture - is labeled with another cluster number. Images (d) and (e) are very similar in content, and the elephants which are almost same are labeled with the same cluster number which is 316. The elephant in the image (f) is also labeled with the same cluster. Although, the images in the last row are from different CD's, the sky is labeled with the same cluster in all of these images. However, the cluster number 444 is also used in image (a) for the red sky which indicates that features other than color can also be dominant factors for clustering.

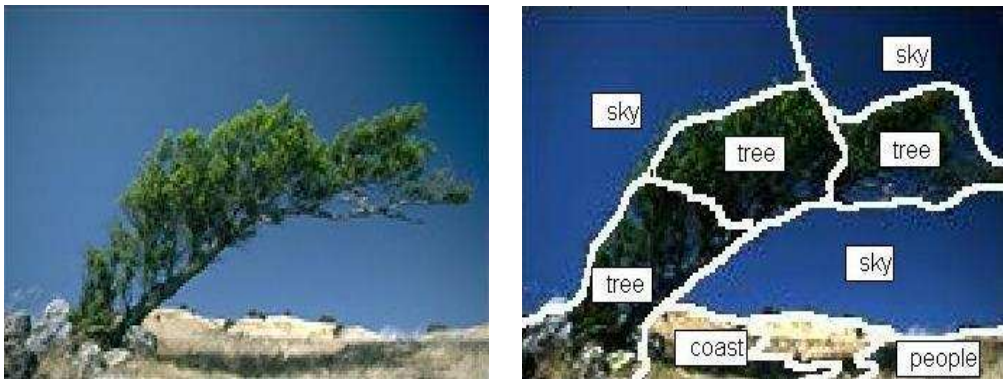
## 4.2 Experimental results

Initially, all the alignments between the words and blobs are set to uniform weights, and the initial probability table is constructed using the co-occurrences of the words and blobs in the training set. The EM algorithm, described in detail in the previous chapter, is iterated 50 times over the data sets to construct the probability table. Unless otherwise stated, all the results are obtained using the first experimental data set. However, in order to show that the results are fairly independent of the choice of the data set, some experiments are carried on all of the sets.

The results presented in this section are referred to as the results of the **base case**. In the following section, we investigate how the performance of the method is affected by different choices of the parameters, and how the performance can be improved.

### 4.2.1 Visual evaluation

Each blob is labeled by the word with the highest probability of prediction for that blob. This allows one to visually inspect the images and evaluate the results. Figure 4.6 and 4.7 present some exemplary results obtained from the training and the standard test sets respectively. On the images from the training set, most of the predicted words are correct: the words `tree, sky, buildings, gardens, mountains, hills,` and `helicopter` are correctly predicted. The errors are mostly created by the high frequency words like `water` and `people`. This is due to the fact that, when a correspondence is not learned properly, the highest probability word predicted for a blob tends to be one of the high frequency words. As can be seen in the figure, the predictions made for the images from the standard test set are satisfactory: the words `plane, sky, waves, lion, buildings, people, fish` and `rock` are predicted correctly. The problems pointed out above are also valid for this set.



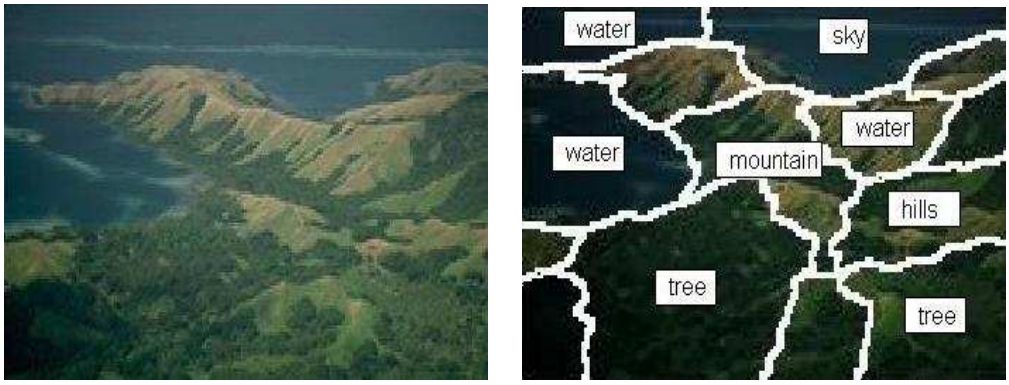

hills sky tree


sea sky sun waves


beach sky tree water

Figure 4.6: Sample images and the word prediction results on the training set. For each blob, the word with the highest probability is chosen. In the top image the words `tree`, and `sky`; in the middle image the words `sky` and `water`; and in the bottom image the words `sky, water, tree` and `beach` are predicted correctly. Although, the word `sun`, which is also an annotation word is predicted in the middle image, due to the segmentation error it is not on the right blob.

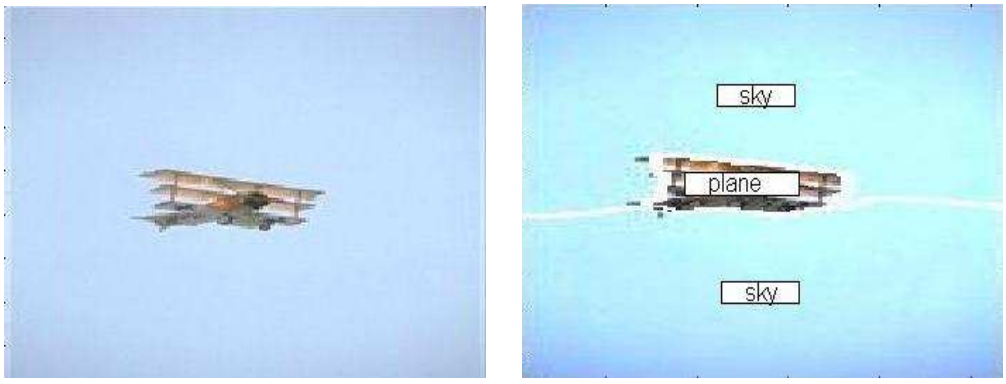

gardens house tree



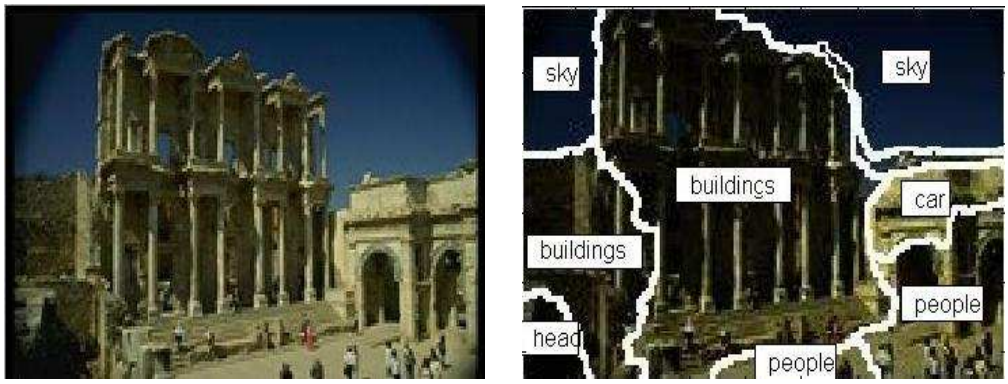

mountain tree water


coast helicopter water

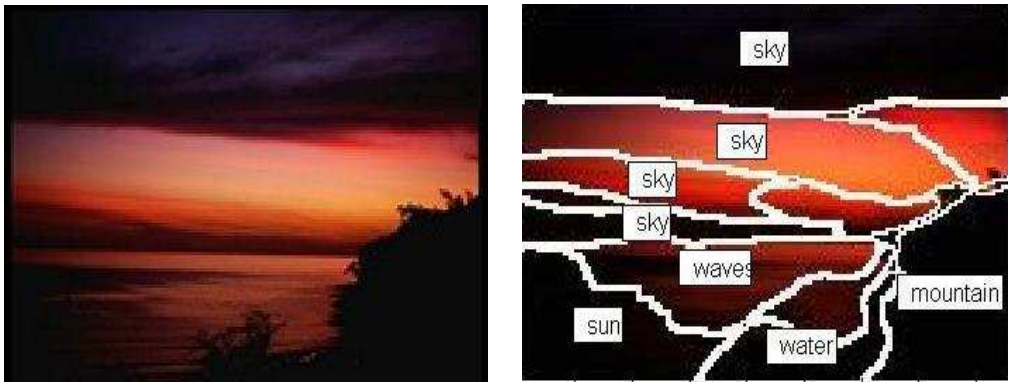
Figure 4.7: Sample images and the word prediction results on the training set (continued). In the top image the words `buildings, gardens`, and `tree`; in the middle image the words `water, mountain, hills` and `tree`; and in the bottom image the words `helicopter, water` and `sky` are predicted correctly on the right blobs. Some words are predicted correctly on the right blobs, although they are not used as the annotation keywords: `buildings` in the top image, `hills` in the middle image, and `sky` in the bottom image.




plane sky




people pillars ruins stone




horizon sunset tree water

Figure 4.7: Sample images and the word prediction results for the standard test set. In the top image the words `plane`, and `sky`; in the middle image the words `buildings, people` and `sky`; and in the bottom image the words `waves, sky` and `water` are predicted correctly.

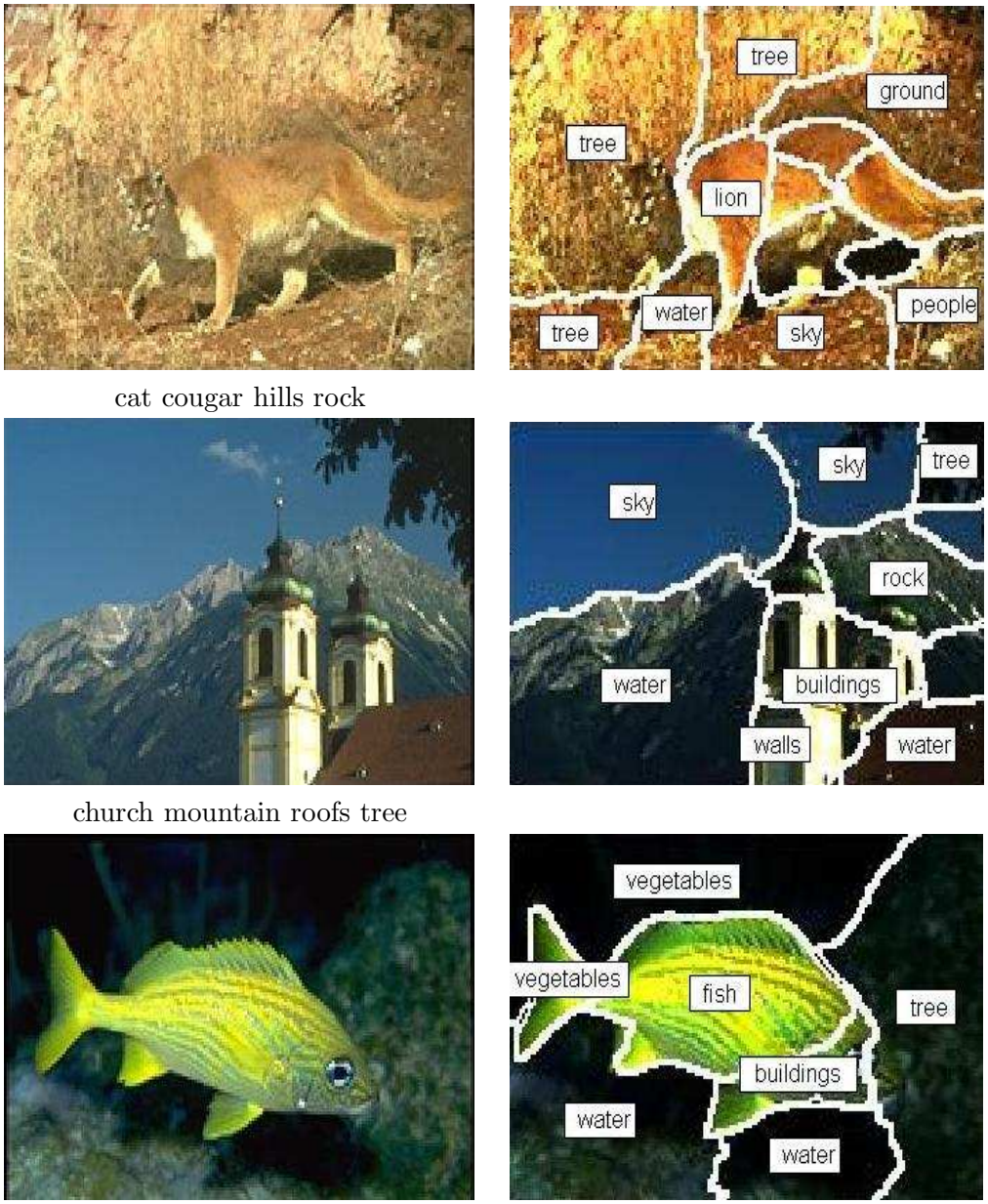



Figure 4.8: Sample images and the word prediction results for the standard test set (continued). In the top image the words `lion`, and `ground`; in the middle image the words `buildings, walls, rock, sky` and `tree`; are predicted correctly. The word `water` is predicted wrongly in the first two images, since the system has a tendency to predict high frequency words where the correspondence is not learned properly. In the last image the word `fish` is predicted correctly. Although `water` is an annotation word, since it is transparent, it is not clear whether it should be counted as a correct prediction or not. All the other words are incorrect.

### 4.2.2 Scoring correspondences by using hand-labeled data

In this section, a hand-labeled set is used for evaluating the correspondence performance. On a small number of images, regions are labeled by hand with the vocabulary words defining the object (or part of an object). Then, the words are predicted for the labeled regions (for each region the word with the highest probability is predicted). The correct predictions are found by comparing the predicted word with the label words.

Producing the labeled data set is a time consuming and error prone process as discussed in Section 3.4. Thus, we label 450 images, with 1-4 words per region. Only the regions that have coherent visual properties are labeled (e.g. if the region is not a part of an object -due to poor segmentation- it is not labeled).

In Table 4.4, the correspondence scores (i.e. number of correct predictions) on the hand-labeled set for all the words which are either predicted or used as a label word are listed. Two methods are used for comparing the predicted and label words to count the correct predictions:

- In the first one, only the first label word is compared with the predicted word (referred as **first**).
- In the second one all of the label words are used for comparison (referred as **all**).

For labeling, 117 words are selected among 153 words in the vocabulary and 77 words are predicted by the system. Only 23 words are correctly predicted when the comparison is with the first label word, and 26 words are correctly predicted when the comparison is with all of the label words. Usually, the first label is a specific word and the others are more general words. There are more words correctly predicted when the comparison is with all the label words. However, when the prediction rates are analyzed, the performance for predicting the first label is better. That is, the system is able to predict words, even if they are specific words. This indicates the success of the system in learning the correspondence between the words and blobs.

Table 4.4: Correspondence scores using the hand-labeled set. 77 words are predicted by the system. 23 words are correctly predicted when the comparison is with the first label word, and 26 words are correctly predicted when the comparison is with all of the label words. For each word: (i) number of times that the word is predicted, (ii) number of times that the word is used as the first label word (first), (iii) number of correct predictions when predicted word is compared with the first label word, (iv) number of times that the word is used as one of the label words (all), and (v) number of times the predicted word is compared with all of the label words, are listed. For some high frequency words, such as `water, sky` and `people` the prediction rate for the labeled samples are over 30%. For some rare words, such as `sea` and `windows`, almost all the predictions are correct.

| word | pred | first | correct(first) | all | correct(all) |
|---|---|---|---|---|---|
| animal | 0 | 28 | 0 | 57 | 0 |
| arch | 0 | 10 | 0 | 19 | 0 |
| architecture | 2 | 0 | 0 | 0 | 0 |
| art | 0 | 3 | 0 | 12 | 0 |
| bay | 0 | 2 | 0 | 9 | 0 |
| beach | 9 | 37 | 0 | 40 | 0 |
| bears | 0 | 13 | 0 | 35 | 0 |
| birds | 117 | 61 | 6 | 71 | 7 |
| black | 0 | 0 | 0 | 5 | 0 |
| boats | 26 | 29 | 2 | 30 | 2 |
| branch | 13 | 6 | 0 | 21 | 0 |
| bridge | 0 | 25 | 0 | 25 | 0 |
| buildings | 120 | 130 | 21 | 190 | 27 |
| cactus | 0 | 29 | 0 | 29 | 0 |
| car | 8 | 10 | 1 | 10 | 1 |
| castle | 0 | 13 | 0 | 14 | 0 |
| cat | 0 | 55 | 0 | 81 | 0 |
| church | 1 | 8 | 0 | 10 | 0 |
| city | 0 | 7 | 0 | 22 | 0 |
| cliff | 0 | 16 | 0 | 28 | 0 |
| close-up | 3 | 0 | 0 | 0 | 0 |
| closeup | 8 | 0 | 0 | 0 | 0 |
| clouds | 75 | 26 | 5 | 194 | 13 |
| coast | 12 | 13 | 0 | 22 | 0 |
| columns | 3 | 1 | 0 | 6 | 0 |
| coral | 5 | 19 | 1 | 19 | 1 |
| costume | 0 | 9 | 0 | 9 | 0 |
| cougar | 0 | 6 | 0 | 6 | 0 |
| courtyard | 0 | 6 | 0 | 8 | 0 |
| crystal | 1 | 5 | 0 | 5 | 0 |
| desert | 0 | 2 | 0 | 12 | 0 |
| dog | 0 | 5 | 0 | 5 | 0 |
| doors | 1 | 3 | 0 | 3 | 0 |
| dunes | 0 | 1 | 0 | 1 | 0 |

Table 4.4: Continued.

| word | predicted | first | correct(first) | all | correct(all) |
|---|---|---|---|---|---|
| elephants | 0 | 21 | 0 | 21 | 0 |
| entrance | 3 | 0 | 0 | 0 | 0 |
| f-16 | 7 | 0 | 0 | 0 | 0 |
| face | 9 | 0 | 0 | 3 | 0 |
| fence | 0 | 2 | 0 | 2 | 0 |
| field | 5 | 53 | 0 | 168 | 0 |
| fish | 68 | 17 | 1 | 17 | 1 |
| flight | 0 | 0 | 0 | 15 | 0 |
| flowers | 49 | 96 | 8 | 116 | 9 |
| food | 4 | 0 | 0 | 0 | 0 |
| forest | 4 | 20 | 0 | 39 | 0 |
| fungus | 0 | 3 | 0 | 7 | 0 |
| gardens | 8 | 6 | 0 | 51 | 1 |
| grass | 110 | 239 | 21 | 331 | 28 |
| ground | 38 | 67 | 1 | 124 | 3 |
| gun | 4 | 0 | 0 | 0 | 0 |
| harbor | 2 | 0 | 0 | 0 | 0 |
| hats | 1 | 1 | 0 | 1 | 0 |
| hawk | 0 | 0 | 0 | 3 | 0 |
| head | 4 | 0 | 0 | 1 | 0 |
| helicopter | 7 | 10 | 0 | 10 | 0 |
| hills | 11 | 19 | 1 | 56 | 2 |
| horizon | 1 | 0 | 0 | 0 | 0 |
| horse | 0 | 26 | 0 | 26 | 0 |
| house | 1 | 9 | 0 | 31 | 0 |
| hunter | 0 | 4 | 0 | 4 | 0 |
| ice | 0 | 21 | 0 | 57 | 0 |
| insect | 10 | 4 | 0 | 4 | 0 |
| island | 0 | 5 | 0 | 10 | 0 |
| jet | 24 | 24 | 0 | 25 | 0 |
| leaf | 1 | 37 | 0 | 75 | 0 |
| leaves | 4 | 0 | 0 | 0 | 0 |
| light | 0 | 1 | 0 | 9 | 0 |
| lion | 6 | 20 | 1 | 20 | 1 |
| lizard | 0 | 10 | 0 | 10 | 0 |
| mane | 0 | 0 | 0 | 6 | 0 |
| mountain | 31 | 49 | 0 | 65 | 0 |
| museum | 0 | 2 | 0 | 2 | 0 |
| mushrooms | 9 | 5 | 0 | 8 | 0 |
| nest | 0 | 11 | 0 | 11 | 0 |
| night | 1 | 11 | 0 | 12 | 0 |
| ocean | 8 | 37 | 0 | 46 | 0 |
| owl | 5 | 10 | 0 | 10 | 0 |
| paintings | 0 | 5 | 0 | 6 | 0 |
| palace | 0 | 1 | 0 | 5 | 0 |
| pattern | 14 | 0 | 0 | 5 | 0 |
| people | 292 | 41 | 13 | 73 | 21 |
| pillars | 9 | 6 | 1 | 7 | 1 |
| plane | 26 | 1 | 0 | 25 | 1 |
| plants | 11 | 14 | 0 | 115 | 2 |
| polar | 0 | 22 | 0 | 22 | 0 |

Table 4.4: Continued.

| word | predicted | first | correct(first) | all | correct(all) |
|---|---|---|---|---|---|
| pumpkins | 0 | 6 | 0 | 6 | 0 |
| pyramid | 0 | 7 | 0 | 7 | 0 |
| reefs | 5 | 3 | 0 | 15 | 0 |
| reflection | 0 | 1 | 0 | 11 | 0 |
| restaurant | 0 | 3 | 0 | 3 | 0 |
| river | 0 | 0 | 0 | 9 | 0 |
| road | 0 | 9 | 0 | 12 | 0 |
| rock | 81 | 82 | 6 | 130 | 9 |
| roofs | 0 | 0 | 0 | 2 | 0 |
| ruins | 0 | 5 | 0 | 9 | 0 |
| runway | 0 | 5 | 0 | 5 | 0 |
| saguaro | 0 | 0 | 0 | 6 | 0 |
| sand | 0 | 19 | 0 | 63 | 0 |
| sculpture | 0 | 2 | 0 | 5 | 0 |
| sea | 4 | 3 | 2 | 48 | 2 |
| seals | 3 | 0 | 0 | 2 | 0 |
| shadows | 0 | 12 | 0 | 16 | 0 |
| ships | 0 | 1 | 0 | 3 | 0 |
| shore | 0 | 5 | 0 | 6 | 0 |
| shrine | 4 | 0 | 0 | 0 | 0 |
| sky | 352 | 382 | 119 | 451 | 134 |
| smoke | 3 | 7 | 0 | 7 | 0 |
| snake | 3 | 5 | 1 | 5 | 1 |
| snow | 17 | 127 | 2 | 152 | 3 |
| statues | 1 | 21 | 0 | 25 | 0 |
| stone | 45 | 3 | 0 | 29 | 0 |
| street | 36 | 17 | 0 | 17 | 0 |
| sun | 2 | 12 | 0 | 15 | 0 |
| sunset | 0 | 33 | 0 | 52 | 0 |
| temple | 17 | 6 | 0 | 7 | 0 |
| textile | 7 | 0 | 0 | 0 | 0 |
| texture | 40 | 0 | 0 | 5 | 0 |
| tower | 0 | 4 | 0 | 13 | 0 |
| tracks | 12 | 0 | 0 | 0 | 0 |
| tree | 430 | 230 | 65 | 268 | 69 |
| trunk | 0 | 0 | 0 | 1 | 0 |
| turn | 2 | 0 | 0 | 0 | 0 |
| vegetables | 27 | 0 | 0 | 3 | 0 |
| walls | 23 | 25 | 1 | 35 | 1 |
| water | 459 | 229 | 92 | 283 | 103 |
| waves | 2 | 3 | 0 | 45 | 0 |
| wildlife | 3 | 0 | 0 | 0 | 0 |
| windows | 4 | 3 | 1 | 3 | 1 |
| wolf | 0 | 8 | 0 | 8 | 0 |
| woman | 39 | 16 | 0 | 16 | 0 |
| woods | 5 | 3 | 0 | 5 | 0 |
| zebra | 0 | 12 | 0 | 12 | 0 |

Recall and precision graphs are popular tools to represent performance in information retrieval literature. For correspondence scoring, the recall and precision can be defined as:

- **Recall:** the number of correct predictions of the word over the number of times that the word is a label word.
- **Precision:** the number of correct predictions of the word over the number of times that the word is predicted.

Figure 4.8 shows the recall and precision values for the words which have non-zero recall values, when the predicted word is compared with the first label word and Figure 4.9 shows the results when the comparison is with all the label words. For a decent number of words the recall and precision rates are over 30%. The recall values in Figure 4.9 are higher than the recall values in Figure 4.8, while the precision values are very similar.

### 4.2.3 Using annotation as a proxy

In this section, instead of evaluating the results with correspondence scores, we use annotation scores as a proxy, in order to score the results on the large scale.

The easiest way to obtain the auto-annotations for an image is summing up all the word posterior probabilities for the blobs that occur in the image, to obtain a single word posterior probability for the whole image. We measure annotation performance by comparing the words predicted for the image with the words that actually present as a keyword in the image. In the following experiments, we allow to predict $n$ words with the highest probabilities given the image, where $n$ is the number of actual keywords of the image.

When image annotations are used for scoring, a word is assumed to be correctly predicted if it is one of the actual keywords. In this way, it is not possible to investigate whether the words are predicted on the correct places, but it is known whether the correct words are predicted or not.

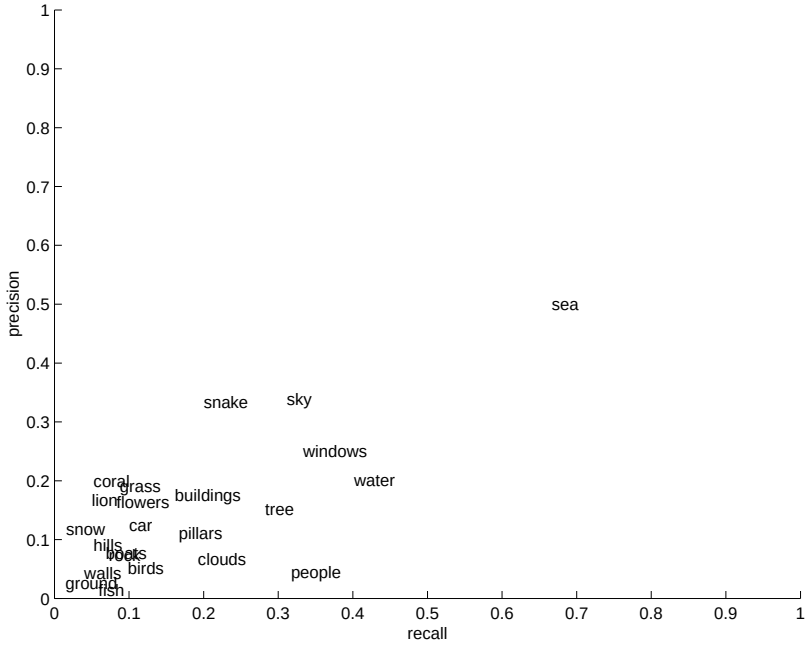



Figure 4.8: Recall versus precision graphs for the words which have non-zero recall values, when the first label word is used for comparison. **Recall** is defined as the number of correct predictions over number of times that the word is used as a label word and **precision** is defined as the number of correct predictions over number of times that the word is predicted.

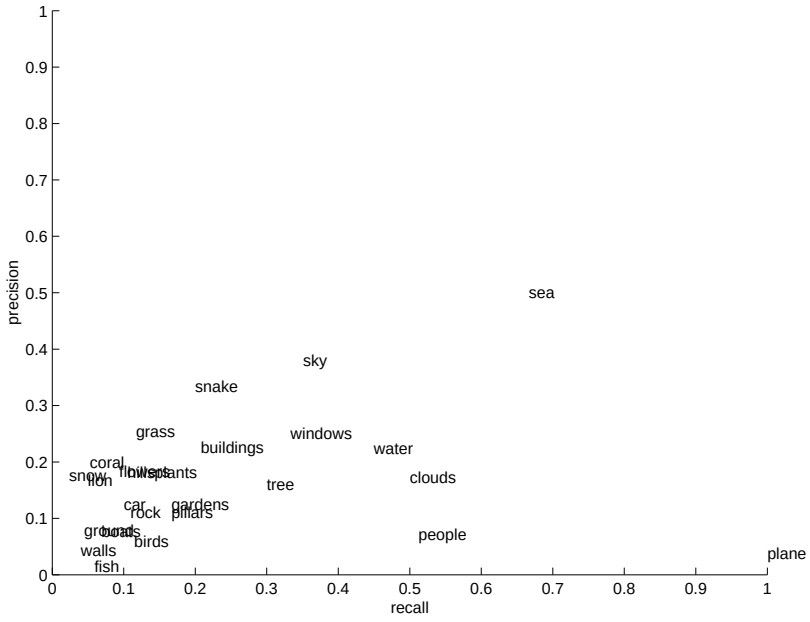



Figure 4.9: Recall versus precision graphs for the words which have non-zero recall values, when all the label words are used for comparison.

Recall and precision are defined as an annotation score as follows:

- **Recall:** the number of correct predictions over number of actual occurrence of the word in the data as a keyword,
- **Precision:** the number of correct predictions over number of all predictions.

In Figure 4.10, the recall versus precision values are shown for the hand-labeled set when auto-annotation performance is used as a proxy. When compared with Figure 4.8, it is seen that for some words, correspondence and annotation scores yield similar results (e.g. snow, flowers, clouds, grass). It is not possible to understand the complete relation between annotation and correspondence results, since the hand-labeled data only includes 450 images (the number of images in the standard test set is 1744), and hence few words.

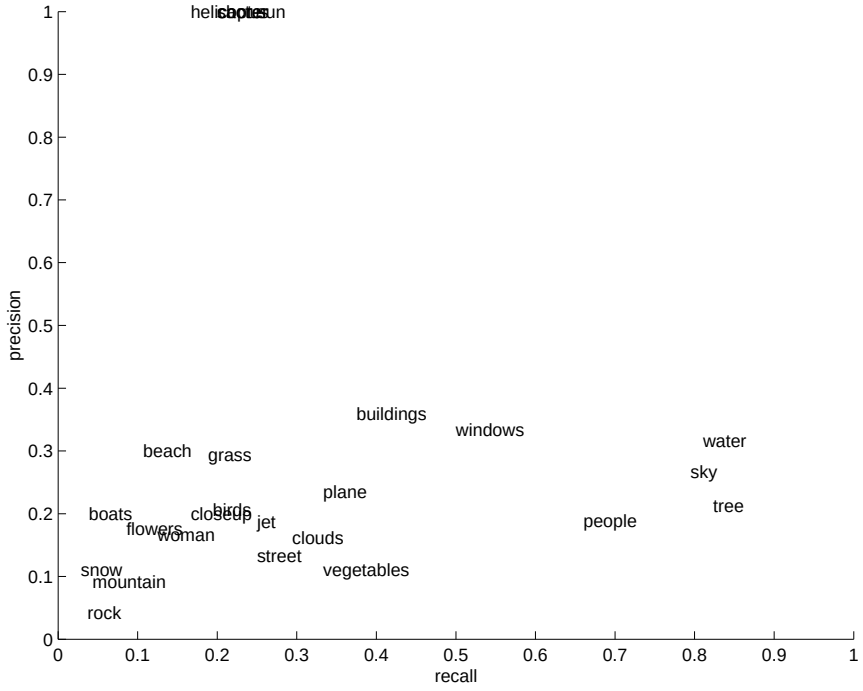



Figure 4.10: Recall versus precision values on the hand-labeled set for annotation scores. **Recall** is defined as the number of correct predictions over number of actual occurrence of the word in the data as a keyword and **precision** is defined as the number of correct predictions over number of all predictions.

We also plot the recall versus precision values for the training, standard test and novel test sets of the first experimental data set, as shown in Figure 4.11, Figure 4.12 and Figure 4.13, respectively. There are total of 153 words in the vocabulary of this particular set. However, 76 words on the training set have nonzero values. Similarly, there are 36 words on the standard test set and 40 words on the novel test set with nonzero values.

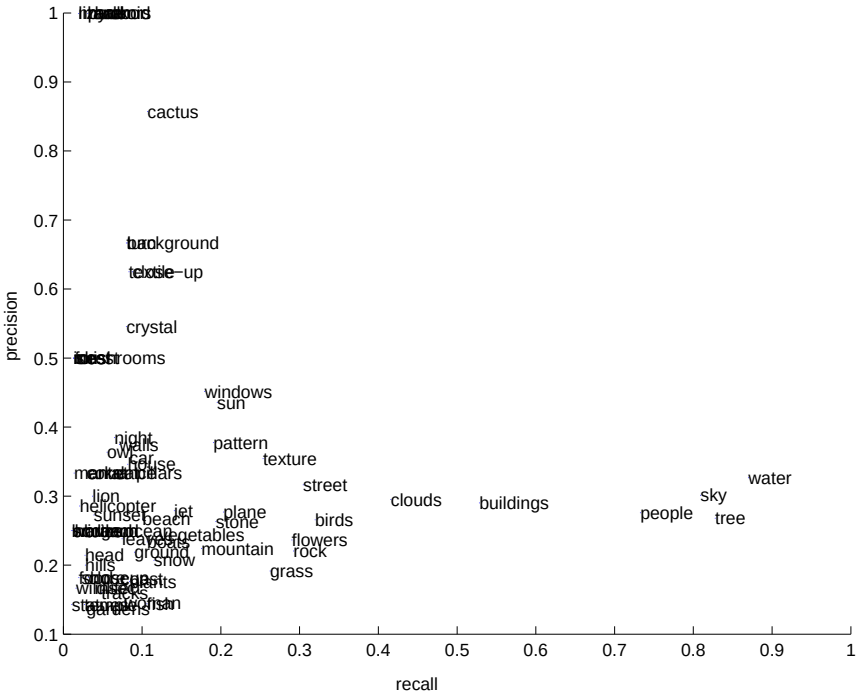



Figure 4.11: Recall versus precision values for the training set. 76 words have nonzero values. The system predict some high frequency words, such as `water, people, sky` and `tree` with high recall values. However, the precision values are not satisfactory. It means that, the system predicts these words many times, but less than half of these predictions are correct. However, for some words, such as `hawk, doors, pyramid, cactus` and `background`, although the recall values are low, the precision values are very high. It means that, whenever the system predicts these words, almost always it is correct.

The goal is to obtain high recall and precision values for the words in the vocabulary. For a decent number of words, the recall and/or precision values are very satisfactory. The recall values for some words are very high (close to 1), meaning that they are predicted almost always when they occur in the data. However, their precision values are less than 0.5, which means that we are predicting those words many times, but less than half of those predictions are correct. However, there are some words with low recall but high precision values, which means that although they are not predicted often, the predictions are usually correct.

The results for the training and standard test sets are similar for most of the words, but in the test set we predict less words. Prediction rates for the novel test set is similar to the the predictions for the training set only for some common words, but different for the rest of the vocabulary, since the vocabulary of the novel test set is usually different than the vocabulary of the trained data.

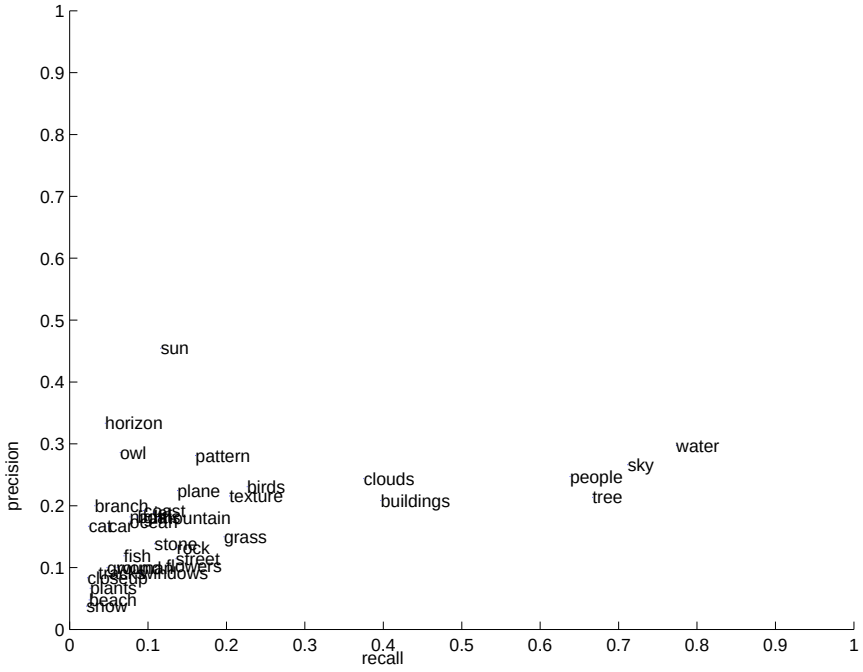



Figure 4.12: Recall versus precision values for the standard test set. 36 words have nonzero values. When compared to Figure 4.11 it is observed that, the results are similar for most of the words, but the system predicts less words for the test data.

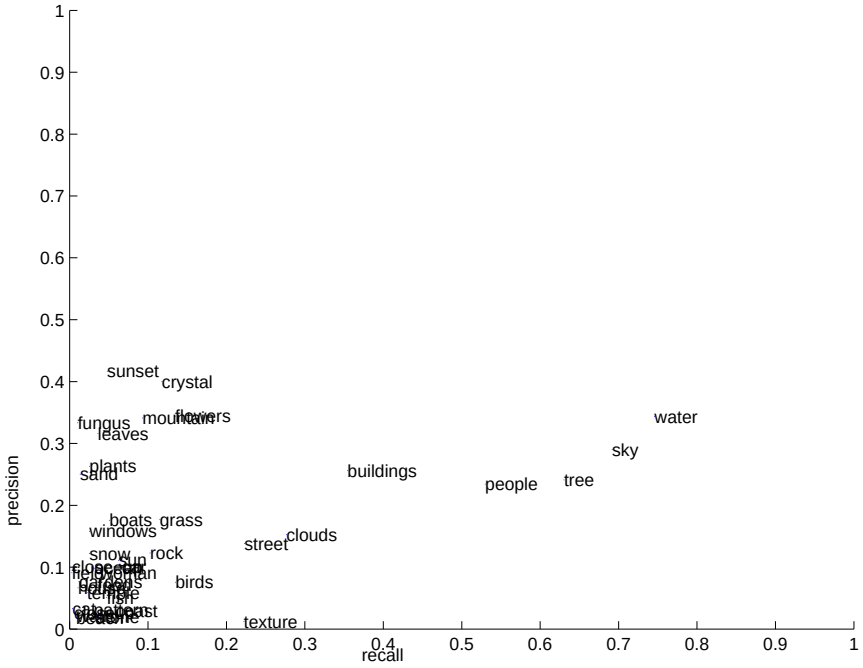



Figure 4.13: Recall versus precision values for the novel test set. 40 words have nonzero values. When compared to Figure 4.11 it is observed that, the results for the novel test set is similar to the the results for the training set only for some common words, but different for the rest of the vocabulary, since the vocabulary of the novel test set is usually different than the vocabulary of the trained data.

The number of times that the predicted word is one of the actual keywords (number of **true positives** is also a useful measure to understand the prediction performance, however it is not sufficient by itself. Therefore, we also investigate the number of **false positives** and the number of **false negatives**, which are defined as:

- **false positives:** number of times that the word is predicted but not one of the keywords
- **false negatives:** number of times that the word is a keyword but not predicted.

Table 4.5 and Table 4.6 show the true positive, false positive and false negative values for the predicted words on training and standard test sets respectively.

Table 4.5: Prediction results for the words in the training set. 98 words are predicted over 153 words. For each word: (i) number of predictions (pred.), (ii) number of occurrence in the set (occ.), (iii) number of true positives (tp), (iv) number of false positives (fp) and (v) number of false negatives (fn), are shown. The results are in the sorted order according to the frequency of the predictions.

| word | pred. | occ. | tp | fp | fn | word | pred. | occ. | tp | fp | fn |
|---|---|---|---|---|---|---|---|---|---|---|---|
| water | 2995 | 1124 | 978 | 2017 | 146 | tree | 2870 | 929 | 769 | 2101 | 160 |
| sky | 2549 | 949 | 768 | 1781 | 181 | people | 2265 | 853 | 625 | 1640 | 228 |
| buildings | 803 | 441 | 233 | 570 | 208 | grass | 465 | 339 | 89 | 376 | 250 |
| clouds | 461 | 327 | 136 | 325 | 191 | rock | 399 | 301 | 88 | 311 | 213 |
| birds | 354 | 294 | 94 | 260 | 200 | flowers | 275 | 224 | 65 | 210 | 159 |
| mountain | 223 | 285 | 50 | 173 | 235 | street | 234 | 243 | 74 | 160 | 169 |
| stone | 190 | 258 | 50 | 140 | 208 | plane | 177 | 241 | 49 | 128 | 192 |
| texture | 144 | 201 | 51 | 93 | 150 | snow | 140 | 252 | 29 | 111 | 223 |
| fish | 132 | 179 | 19 | 113 | 160 | pattern | 106 | 210 | 40 | 66 | 170 |
| vegetables | 78 | 156 | 19 | 59 | 137 | jet | 104 | 206 | 29 | 75 | 177 |
| boats | 77 | 169 | 18 | 59 | 151 | coast | 72 | 178 | 13 | 59 | 165 |
| ground | 64 | 155 | 14 | 50 | 141 | beach | 60 | 159 | 16 | 44 | 143 |
| woman | 55 | 103 | 8 | 47 | 95 | plants | 51 | 107 | 9 | 42 | 98 |
| leaves | 42 | 134 | 10 | 32 | 124 | sun | 39 | 87 | 17 | 22 | 70 |
| windows | 31 | 78 | 14 | 17 | 64 | ocean | 36 | 116 | 9 | 27 | 107 |
| car | 31 | 131 | 11 | 20 | 120 | cat | 27 | 135 | 9 | 18 | 126 |
| house | 26 | 110 | 9 | 17 | 101 | tracks | 25 | 83 | 4 | 21 | 79 |
| walls | 24 | 126 | 9 | 15 | 117 | gardens | 22 | 103 | 3 | 19 | 100 |
| closeup | 22 | 125 | 4 | 18 | 121 | temple | 21 | 111 | 3 | 18 | 108 |
| hills | 20 | 143 | 4 | 16 | 139 | insect | 18 | 75 | 3 | 15 | 72 |
| close-up | 16 | 114 | 10 | 6 | 104 | pillars | 15 | 54 | 5 | 10 | 49 |
| head | 14 | 109 | 3 | 11 | 106 | night | 13 | 77 | 5 | 8 | 72 |
| sunset | 11 | 77 | 3 | 8 | 74 | shore | 11 | 87 | 2 | 9 | 85 |
| owl | 11 | 72 | 4 | 7 | 68 | food | 11 | 102 | 2 | 9 | 100 |
| crystal | 11 | 75 | 6 | 5 | 69 | lion | 10 | 81 | 3 | 7 | 78 |
| background | 9 | 74 | 6 | 3 | 68 | reefs | 9 | 85 | 0 | 9 | 85 |
| textile | 8 | 60 | 5 | 3 | 55 | sand | 8 | 109 | 0 | 8 | 109 |
| branch | 8 | 75 | 2 | 6 | 73 | statues | 7 | 93 | 1 | 6 | 92 |
| helicopter | 7 | 96 | 2 | 5 | 94 | cactus | 7 | 56 | 6 | 1 | 50 |
| wildlife | 6 | 62 | 1 | 5 | 61 | coral | 6 | 68 | 2 | 4 | 66 |
| field | 5 | 83 | 0 | 5 | 83 | wolf | 4 | 67 | 0 | 4 | 67 |
| waves | 4 | 82 | 1 | 3 | 81 | Scotland | 4 | 79 | 1 | 3 | 78 |
| horizon | 4 | 68 | 1 | 3 | 67 | hats | 4 | 46 | 0 | 4 | 46 |
| bridge | 4 | 93 | 1 | 3 | 92 | woods | 3 | 36 | 0 | 3 | 36 |
| turn | 3 | 25 | 2 | 1 | 23 | market | 3 | 73 | 1 | 2 | 72 |
| leaf | 3 | 29 | 0 | 3 | 29 | entrance | 3 | 32 | 1 | 2 | 31 |
| bears | 3 | 76 | 0 | 3 | 76 | animal | 3 | 76 | 0 | 3 | 76 |
| skis | 2 | 63 | 1 | 1 | 62 | nest | 2 | 58 | 1 | 1 | 57 |
| mushrooms | 2 | 78 | 1 | 1 | 77 | ice | 2 | 77 | 1 | 1 | 76 |
| hawk | 2 | 65 | 2 | 0 | 63 | forest | 2 | 68 | 1 | 1 | 67 |
| doors | 2 | 36 | 2 | 0 | 34 | village | 1 | 36 | 0 | 1 | 36 |
| ruins | 1 | 114 | 0 | 1 | 114 | pyramid | 1 | 32 | 1 | 0 | 31 |
| pumpkins | 1 | 77 | 0 | 1 | 77 | palace | 1 | 41 | 0 | 1 | 41 |
| mane | 1 | 52 | 1 | 0 | 51 | lizard | 1 | 52 | 1 | 0 | 51 |
| harbor | 1 | 25 | 1 | 0 | 24 | fungus | 1 | 78 | 0 | 1 | 78 |

Table 4.6: Prediction results for the words in the standard test set. 74 words are predicted over 153 words. For each word: (i) number of predictions (pred.), (ii) number of occurrence in the set (occ.), (iii) number of true positives (tp), (iv) number of false positives (fp) and (v) number of false negatives (fn), are shown. The results are in the sorted order according to the frequency of the predictions.

| word | pred. | occ. | tp | fp | fn | word | pred. | occ. | tp | fp | fn |
|---|---|---|---|---|---|---|---|---|---|---|---|
| water | 1022 | 393 | 304 | 718 | 89 | tree | 946 | 303 | 202 | 744 | 101 |
| sky | 834 | 312 | 222 | 612 | 90 | people | 785 | 304 | 194 | 591 | 110 |
| buildings | 240 | 126 | 50 | 190 | 76 | grass | 167 | 127 | 25 | 142 | 102 |
| clouds | 160 | 104 | 39 | 121 | 65 | rock | 121 | 117 | 16 | 105 | 101 |
| birds | 104 | 106 | 24 | 80 | 82 | flowers | 88 | 73 | 9 | 79 | 64 |
| street | 70 | 59 | 8 | 62 | 51 | stone | 65 | 83 | 9 | 56 | 74 |
| mountain | 61 | 97 | 11 | 50 | 86 | snow | 53 | 93 | 2 | 51 | 91 |
| texture | 51 | 54 | 11 | 40 | 43 | plane | 49 | 80 | 11 | 38 | 69 |
| fish | 42 | 72 | 5 | 37 | 67 | jet | 37 | 65 | 7 | 30 | 58 |
| boats | 33 | 69 | 6 | 27 | 63 | pattern | 32 | 56 | 9 | 23 | 47 |
| coast | 26 | 53 | 5 | 21 | 48 | ocean | 23 | 52 | 4 | 19 | 48 |
| beach | 21 | 40 | 1 | 20 | 39 | woman | 20 | 33 | 2 | 18 | 31 |
| ground | 20 | 42 | 2 | 18 | 40 | vegetables | 19 | 50 | 0 | 19 | 50 |
| leaves | 16 | 50 | 0 | 16 | 50 | plants | 15 | 38 | 1 | 14 | 37 |
| walls | 12 | 33 | 0 | 12 | 33 | insect | 12 | 24 | 0 | 12 | 24 |
| closeup | 12 | 44 | 1 | 11 | 43 | car | 12 | 40 | 2 | 10 | 38 |
| windows | 11 | 11 | 1 | 10 | 10 | tracks | 11 | 27 | 1 | 10 | 26 |
| sun | 11 | 43 | 5 | 6 | 38 | night | 11 | 26 | 2 | 9 | 24 |
| head | 8 | 24 | 0 | 8 | 24 | reefs | 7 | 41 | 0 | 7 | 41 |
| owl | 7 | 31 | 2 | 5 | 29 | hills | 7 | 40 | 0 | 7 | 40 |
| cat | 6 | 41 | 1 | 5 | 40 | statues | 5 | 35 | 0 | 5 | 35 |
| pillars | 5 | 28 | 0 | 5 | 28 | house | 5 | 36 | 0 | 5 | 36 |
| food | 5 | 32 | 0 | 5 | 32 | close-up | 5 | 37 | 0 | 5 | 37 |
| branch | 5 | 31 | 1 | 4 | 30 | temple | 4 | 26 | 0 | 4 | 26 |
| background | 4 | 23 | 0 | 4 | 23 | waves | 3 | 33 | 0 | 3 | 33 |
| sunset | 3 | 33 | 0 | 3 | 33 | shore | 3 | 39 | 0 | 3 | 39 |
| horizon | 3 | 22 | 1 | 2 | 21 | helicopter | 3 | 13 | 0 | 3 | 13 |
| coral | 3 | 29 | 0 | 3 | 29 | wildlife | 2 | 22 | 0 | 2 | 22 |
| textile | 2 | 18 | 0 | 2 | 18 | skis | 2 | 18 | 0 | 2 | 18 |
| sand | 2 | 28 | 0 | 2 | 28 | ruins | 2 | 30 | 0 | 2 | 30 |
| lion | 2 | 33 | 0 | 2 | 33 | hawk | 2 | 15 | 0 | 2 | 15 |
| field | 2 | 32 | 0 | 2 | 32 | nest | 1 | 20 | 0 | 1 | 20 |
| gardens | 1 | 38 | 0 | 1 | 38 | fungus | 1 | 22 | 0 | 1 | 22 |
| flight | 1 | 7 | 0 | 1 | 7 | entrance | 1 | 13 | 0 | 1 | 13 |
| display | 1 | 15 | 0 | 1 | 15 | crystal | 1 | 21 | 0 | 1 | 21 |
| columns | 1 | 17 | 0 | 1 | 17 | cactus | 1 | 19 | 0 | 1 | 19 |
| bridge | 1 | 32 | 0 | 1 | 32 | animal | 1 | 31 | 0 | 1 | 31 |

### 4.2.4 Measuring annotation performance

As discussed in Section3.4.2, three measures are used to evaluate the performance of annotation:

- Kullback-Leibler (KL) divergence,
- Normalized classification score,
- Word prediction measure.

We use these measures, to compare ten experimental data sets. The results are shown in Table 4.7 – 4.9 for KL-divergence, normalized classification score and word prediction measure respectively. For KL divergence, small values indicates better performance, since it indicates the distance between the predicted distribution and the target distribution. For normalized classification score and for word prediction measure, bigger values indicate better performance.

As the tables indicate, the performances for each of ten sets are close to each other on training and standard test sets. The differences are mostly for the novel test sets. It is due to the fact that, the vocabulary of the training set and novel test set can be very different for some set, and hence affects the performance.

Table 4.7: KL divergence results for each of the ten experimental data sets on training, standard test and novel test sets.

| set | training | standard test | novel test |
|---|---|---|---|
| 001 | 3.5602 | 5.2089 | 5.6769 |
| 002 | 3.4932 | 4.9387 | 4.3696 |
| 003 | 3.5322 | 4.9982 | 5.4598 |
| 004 | 3.6355 | 5.3491 | 5.7723 |
| 005 | 3.5123 | 5.0050 | 5.5352 |
| 006 | 3.5206 | 5.1052 | 5.9007 |
| 007 | 3.7002 | 5.2544 | 4.3680 |
| 008 | 3.5643 | 5.1617 | 5.5048 |
| 009 | 3.6573 | 5.2011 | 4.4484 |
| 010 | 3.4594 | 4.9578 | 5.4725 |

Table 4.8: Normalized classification scores for each of the ten experimental data sets on training, standard test and novel test sets.

| set | training | standard test | novel test |
|---|---|---|---|
| 001 | 0.2560 | 0.2012 | 0.2102 |
| 002 | 0.2657 | 0.2111 | 0.2053 |
| 003 | 0.2616 | 0.2129 | 0.1968 |
| 004 | 0.2449 | 0.1771 | 0.2048 |
| 005 | 0.2713 | 0.2222 | 0.1933 |
| 006 | 0.2636 | 0.2046 | 0.2037 |
| 007 | 0.2501 | 0.1895 | 0.2097 |
| 008 | 0.2664 | 0.2220 | 0.1978 |
| 009 | 0.2527 | 0.2082 | 0.1990 |
| 010 | 0.2659 | 0.2131 | 0.1854 |

Table 4.9: Word prediction measures for each of the ten experimental data sets on training, standard test and novel test sets.

| set | training | standard test | novel test |
|---|---|---|---|
| 001 | 0.2708 | 0.2171 | 0.2236 |
| 002 | 0.2799 | 0.2262 | 0.2173 |
| 003 | 0.2763 | 0.2288 | 0.2095 |
| 004 | 0.2592 | 0.1925 | 0.2172 |
| 005 | 0.2853 | 0.2370 | 0.2059 |
| 006 | 0.2776 | 0.2198 | 0.2163 |
| 007 | 0.2632 | 0.2036 | 0.2217 |
| 008 | 0.2799 | 0.2363 | 0.2102 |
| 009 | 0.2659 | 0.2223 | 0.2114 |
| 010 | 0.2815 | 0.2297 | 0.1991 |

### 4.2.5 Log-likelihood and mutual information

Another way of evaluating the performance is checking the log-likelihoods and mutual information of the probability table. In Table 4.10 log-likelihood and mutual information values are shown for each of ten experimental data sets.

Table 4.10: Log-likelihood and mutual information values for the ten experimental data sets.

| set | log-likelihood | mutual info |
|---|---|---|
| 001 | -5.64e+004 | 3.1853 |
| 002 | -5.95e+004 | 3.1651 |
| 003 | -5.79e+004 | 3.1205 |
| 004 | -5.89e+004 | 3.2016 |
| 005 | -5.79e+004 | 3.1100 |
| 006 | -5.71e+004 | 3.2116 |
| 007 | -5.82e+004 | 3.2970 |
| 008 | -5.80e+004 | 3.1954 |
| 009 | -5.85e+004 | 3.2112 |
| 010 | -5.62e+004 | 3.0732 |

### 4.2.6 Blob based results

In this section, we analyze the results on the blob basis. In Table 4.11 the prediction rates for each of 500 blobs on training and standard test sets are indicated. Prediction rates are computed by dividing the number of correct predictions of the highest probability word with the number of occurrence of the blob in the data.

If the word posterior probability of the "best" word is not significantly higher than the other words for a given a blob, then choosing the word with the highest probability may not be the best choice. In order to test the effect of predicting other higher probability words, for each blob we allow to predict 3 words. In Table 4.12, the number of times that the predicted word is one of the keywords for each of three words are shown. There are some overlaps between the numbers since we may predict the first and the second word correctly if both of them are in the keyword list. We compare the number of correct predictions with the number of times that the blob occurs in the data.

Table 4.11: Blob based prediction rates.(i) Blob, (ii)the word with the highest probability given that blob (pred. word), prediction rates (the number of correct predictions of the highest probability word over the number of occurrence of the blob in the data) (iii) for the training set (training), (iv) for the standard test set (test). As it is seen, one word may be predicted with different blobs, with different prediction rates. For some blobs, the prediction rates are very high even the predicted word is not a high frequency word (e.g. blob 380 which predicts `zebra`, blobs 441 and 190 which predict `windows`, blob 217 which predicts `nest` and blob 411 which predicts `sun`).

| blob | pred. word | training | test | blob | pred. word | training | test | blob | pred. word | training | test |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | people | 0.42 | 0.42 | 2 | people | 0.21 | 0.25 | 3 | architecture | 0.08 | 0.13 |
| 4 | closeup | 0.09 | 0.08 | 5 | buildings | 0.18 | 0.15 | 6 | tree | 0.30 | 0.09 |
| 7 | textile | 0.10 | 0.00 | 8 | people | 0.18 | 0.30 | 9 | water | 0.56 | 0.43 |
| 10 | buildings | 0.20 | 0.13 | 11 | coast | 0.13 | 0.13 | 12 | birds | 0.15 | 0.04 |
| 13 | windows | 0.27 | 0.10 | 14 | pillars | 0.13 | 0.13 | 15 | plane | 0.18 | 0.18 |
| 16 | tree | 0.26 | 0.35 | 17 | buildings | 0.29 | 0.07 | 18 | tree | 0.19 | 0.23 |
| 19 | cliff | 0.17 | 0.00 | 20 | house | 0.19 | 0.00 | 21 | rock | 0.15 | 0.00 |
| 22 | birds | 0.19 | 0.19 | 23 | people | 0.26 | 0.05 | 24 | flowers | 0.18 | 0.07 |
| 25 | reefs | 0.09 | 0.00 | 26 | sky | 0.33 | 0.11 | 27 | rock | 0.16 | 0.30 |
| 28 | pattern | 0.53 | 0.60 | 29 | birds | 0.15 | 0.27 | 30 | beach | 0.17 | 0.00 |
| 31 | street | 0.15 | 0.00 | 32 | textile | 0.11 | 0.12 | 33 | sky | 0.29 | 0.27 |
| 34 | street | 0.10 | 0.06 | 35 | birds | 0.26 | 0.15 | 36 | mountain | 0.15 | 0.09 |
| 37 | texture | 0.21 | 0.10 | 38 | water | 0.30 | 0.31 | 39 | tree | 0.28 | 0.28 |
| 40 | tree | 0.23 | 0.19 | 41 | people | 0.46 | 0.40 | 42 | temple | 0.14 | 0.00 |
| 43 | woman | 0.10 | 0.05 | 44 | mushrooms | 0.06 | 0.03 | 45 | tree | 0.29 | 0.20 |
| 46 | jet | 0.11 | 0.09 | 47 | people | 0.21 | 0.20 | 48 | people | 0.23 | 0.18 |
| 49 | water | 0.39 | 0.40 | 50 | sky | 0.18 | 0.09 | 51 | people | 0.22 | 0.35 |
| 52 | stone | 0.10 | 0.07 | 53 | tree | 0.31 | 0.16 | 54 | tree | 0.21 | 0.18 |
| 55 | people | 0.25 | 0.14 | 56 | pattern | 0.19 | 0.06 | 57 | mountain | 0.16 | 0.13 |
| 58 | texture | 0.19 | 0.29 | 59 | sky | 0.22 | 0.15 | 60 | water | 0.41 | 0.29 |
| 61 | tree | 0.27 | 0.23 | 62 | tree | 0.23 | 0.19 | 63 | fish | 0.12 | 0.08 |
| 64 | people | 0.27 | 0.11 | 65 | street | 0.44 | 0.25 | 66 | face | 0.05 | 0.07 |
| 67 | water | 0.40 | 0.36 | 68 | people | 0.26 | 0.29 | 69 | buildings | 0.21 | 0.16 |
| 70 | field | 0.11 | 0.00 | 71 | people | 0.32 | 0.45 | 72 | lion | 0.13 | 0.03 |
| 73 | tree | 0.29 | 0.26 | 74 | coral | 0.09 | 0.04 | 75 | vegetables | 0.13 | 0.03 |
| 76 | water | 0.36 | 0.35 | 77 | people | 0.19 | 0.14 | 78 | water | 0.37 | 0.46 |
| 79 | people | 0.27 | 0.27 | 80 | water | 0.43 | 0.48 | 81 | people | 0.30 | 0.36 |
| 82 | display | 0.07 | 0.10 | 83 | water | 0.17 | 0.13 | 84 | turn | 0.29 | 0 |
| 85 | tree | 0.23 | 0.14 | 86 | coast | 0.10 | 0.00 | 87 | sky | 0.24 | 0.19 |
| 88 | sky | 0.38 | 0.26 | 89 | ocean | 0.11 | 0.11 | 90 | gun | 0.07 | 0.00 |
| 91 | water | 0.32 | 0.20 | 92 | rock | 0.15 | 0.03 | 93 | plane | 0.10 | 0.10 |
| 94 | buildings | 0.19 | 0.13 | 95 | tree | 0.27 | 0.23 | 96 | sky | 0.30 | 0.46 |
| 97 | flowers | 0.14 | 0.07 | 98 | birds | 0.12 | 0.14 | 99 | ground | 0.13 | 0.13 |
| 100 | buildings | 0.19 | 0.09 | 101 | clouds | 0.18 | 0.00 | 102 | texture | 0.13 | 0.09 |
| 103 | water | 0.25 | 0.14 | 104 | pattern | 0.13 | 0.10 | 105 | doors | 0.33 | 0.00 |
| 106 | temple | 0.10 | 0.03 | 107 | water | 0.28 | 0.25 | 108 | vegetables | 0.10 | 0.09 |
| 109 | sky | 0.26 | 0.14 | 110 | tree | 0.29 | 0.17 | 111 | water | 0.20 | 0.27 |
| 112 | tree | 0.20 | 0.14 | 113 | sea | 0.14 | 0.00 | 114 | leaf | 0.16 | 0.00 |
| 115 | texture | 0.14 | 0.06 | 116 | rock | 0.14 | 0.12 | 117 | people | 0.23 | 0.15 |
| 118 | sky | 0.33 | 0.43 | 119 | nest | 0.14 | 0.00 | 120 | helicopter | 0.15 | 0.00 |
| 121 | buildings | 0.32 | 0.29 | 122 | walls | 0.11 | 0.04 | 123 | harbor | 0.10 | 0.00 |
| 124 | rock | 0.14 | 0.20 | 125 | people | 0.34 | 0.29 | 126 | grass | 0.14 | 0.11 |
| 127 | sky | 0.31 | 0.29 | 128 | gardens | 0.10 | 0.00 | 129 | sky | 0.23 | 0.27 |
| 130 | rock | 0.09 | 0.19 | 131 | buildings | 0.11 | 0.00 | 132 | night | 0.11 | 0.00 |
| 133 | forest | 0.08 | 0.06 | 134 | grass | 0.14 | 0.05 | 135 | buildings | 0.24 | 0.10 |
| 136 | ground | 0.21 | 0.00 | 137 | street | 0.29 | 0.20 | 138 | plane | 0.20 | 0.07 |
| 139 | fish | 0.07 | 0.11 | 140 | people | 0.27 | 0.19 | 141 | food | 0.07 | 0.00 |
| 142 | sky | 0.27 | 0.19 | 143 | sky | 0.25 | 0.14 | 144 | entrance | 0.08 | 0.00 |
| 145 | snow | 0.12 | 0.24 | 146 | water | 0.29 | 0.25 | 147 | wildlife | 0.07 | 0.20 |
| 148 | texture | 0.23 | 0.06 | 149 | flowers | 0.21 | 0.15 | 150 | buildings | 0.26 | 0.19 |
| 151 | grass | 0.12 | 0.00 | 152 | boats | 0.11 | 0.17 | 153 | rock | 0.16 | 0.08 |
| 154 | people | 0.34 | 0.33 | 155 | people | 0.22 | 0.15 | 156 | turn | 0.20 | 0.00 |
| 157 | art | 0.09 | 0.00 | 158 | background | 0.15 | 0.25 | 159 | water | 0.28 | 0.24 |
| 160 | water | 0.33 | 0.20 | 161 | sky | 0.21 | 0.14 | 162 | water | 0.23 | 0.27 |
| 163 | people | 0.37 | 0.33 | 164 | birds | 0.19 | 0.10 | 165 | water | 0.31 | 0.28 |
| 166 | water | 0.28 | 0.33 | 167 | people | 0.23 | 0.35 | 168 | temple | 0.06 | 0.00 |
| 169 | background | 0.86 | 0 | 170 | people | 0.28 | 0.46 | 171 | shrine | 0.06 | 0.00 |
| 172 | water | 0.40 | 0.37 | 173 | sky | 0.44 | 0.23 | 174 | owl | 0.11 | 0.00 |
| 175 | people | 0.38 | 0.35 | 176 | insect | 0.14 | 0.00 | 177 | horizon | 0.17 | 0.40 |
| 178 | people | 0.38 | 0.27 | 179 | tree | 0.28 | 0.11 | 180 | sky | 0.28 | 0.28 |
| 181 | people | 0.28 | 0.37 | 182 | ground | 0.09 | 0.05 | 183 | tree | 0.30 | 0.16 |
| 184 | water | 0.49 | 0.33 | 185 | beach | 0.09 | 0.00 | 186 | waves | 0.12 | 0.06 |
| 187 | people | 0.24 | 0.38 | 188 | clouds | 0.16 | 0.06 | 189 | church | 0.08 | 0.06 |
| 190 | windows | 0.44 | 0 | 191 | jet | 0.11 | 0.10 | 192 | pattern | 0.11 | 0.02 |
| 193 | tree | 0.22 | 0.26 | 194 | water | 0.42 | 0.39 | 195 | entrance | 0.12 | 0.00 |
| 196 | flowers | 0.08 | 0.05 | 197 | woman | 0.14 | 0.00 | 198 | buildings | 0.18 | 0.04 |
| 199 | hills | 0.09 | 0.04 | 200 | people | 0.29 | 0.34 | 201 | jet | 0.09 | 0.05 |
| 202 | tree | 0.26 | 0.23 | 203 | sky | 0.20 | 0.15 | 204 | fish | 0.12 | 0.10 |
| 205 | rock | 0.13 | 0.19 | 206 | water | 0.38 | 0.20 | 207 | people | 0.30 | 0.42 |
| 208 | columns | 0.09 | 0.00 | 209 | clouds | 0.13 | 0.00 | 210 | sky | 0.36 | 0.37 |
| 211 | water | 0.27 | 0.15 | 212 | buildings | 0.34 | 0.24 | 213 | hawk | 0.14 | 0.00 |
| 214 | sky | 0.39 | 0.27 | 215 | sky | 0.29 | 0.37 | 216 | street | 0.37 | 0.10 |
| 217 | nest | 1.00 | 0.00 | 218 | plants | 0.16 | 0.04 | 219 | buildings | 0.27 | 0.17 |
| 220 | ground | 0.08 | 0.02 | 221 | water | 0.20 | 0.11 | 222 | water | 0.41 | 0.43 |
| 223 | mountain | 0.20 | 0.05 | 224 | sky | 0.31 | 0.20 | 225 | walls | 0.07 | 0.04 |
| 226 | sky | 0.14 | 0.11 | 227 | street | 0.28 | 0.06 | 228 | people | 0.19 | 0.18 |

Table 4.11: Continued.

| no | pred. word | training | test | no | pred. word | training | test | no | pred. word | training | test |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 229 | water | 0.42 | 0.50 | 230 | sky | 0.38 | 0.00 | 231 | rock | 0.14 | 0.10 |
| 232 | tree | 0.28 | 0.34 | 233 | stone | 0.13 | 0.00 | 234 | people | 0.35 | 0.39 |
| 235 | tree | 0.24 | 0.21 | 236 | tree | 0.31 | 0.29 | 237 | mountain | 0.18 | 0.05 |
| 238 | water | 0.40 | 0.40 | 239 | water | 0.35 | 0.30 | 240 | jet | 0.17 | 0.07 |
| 241 | mountain | 0.17 | 0.10 | 242 | snow | 0.11 | 0.00 | 243 | tracks | 0.14 | 0.00 |
| 244 | clouds | 0.15 | 0.28 | 245 | water | 0.24 | 0.12 | 246 | sky | 0.26 | 0.43 |
| 247 | people | 0.21 | 0.11 | 248 | stone | 0.11 | 0.03 | 249 | boats | 0.11 | 0.06 |
| 250 | tree | 0.24 | 0.19 | 251 | street | 0.26 | 0.17 | 252 | people | 0.32 | 0.27 |
| 253 | tree | 0.23 | 0.23 | 254 | textile | 0.31 | 1.00 | 255 | tree | 0.22 | 0.23 |
| 256 | buildings | 0.30 | 0.09 | 257 | clouds | 0.20 | 0.13 | 258 | water | 0.49 | 0.26 |
| 259 | seals | 0.06 | 0.00 | 260 | tree | 0.27 | 0.12 | 261 | street | 0.23 | 0.10 |
| 262 | texture | 0.14 | 0.07 | 263 | street | 0.22 | 0.16 | 264 | crystal | 0.15 | 0.00 |
| 265 | people | 0.29 | 0.29 | 266 | smoke | 0.12 | 0.00 | 267 | tree | 0.36 | 0.42 |
| 268 | walls | 0.11 | 0.11 | 269 | clouds | 0.13 | 0.04 | 270 | ground | 0.14 | 0.00 |
| 271 | people | 0.37 | 0.48 | 272 | pillars | 0.11 | 0.15 | 273 | water | 0.32 | 0.40 |
| 274 | sky | 0.48 | 0.33 | 275 | water | 0.44 | 0.57 | 276 | tree | 0.29 | 0.19 |
| 277 | sky | 0.34 | 0.33 | 278 | people | 0.26 | 0.33 | 279 | people | 0.31 | 0.17 |
| 280 | grass | 0.14 | 0.11 | 281 | water | 0.34 | 0.29 | 282 | grass | 0.12 | 0.10 |
| 283 | street | 0.31 | 0.20 | 284 | sky | 0.19 | 0.11 | 285 | sky | 0.47 | 0.32 |
| 286 | temple | 0.33 | 0.00 | 287 | water | 0.63 | 0.00 | 288 | vegetables | 0.27 | 0.00 |
| 289 | stone | 0.15 | 0.21 | 290 | buildings | 0.18 | 0.04 | 291 | grass | 0.17 | 0.22 |
| 292 | sky | 0.34 | 0.46 | 293 | water | 0.34 | 0.47 | 294 | tree | 0.25 | 0.22 |
| 295 | sky | 0.36 | 0.38 | 296 | water | 0.30 | 0.32 | 297 | f-16 | 0.06 | 0.00 |
| 298 | sky | 0.28 | 0.19 | 299 | street | 0.27 | 0.17 | 300 | sky | 0.14 | 0.06 |
| 301 | mountain | 0.10 | 0.00 | 302 | fish | 0.10 | 0.04 | 303 | fish | 0.10 | 0.02 |
| 304 | water | 0.40 | 0.40 | 305 | snow | 0.15 | 0.04 | 306 | birds | 0.29 | 0.14 |
| 307 | buildings | 0.20 | 0.18 | 308 | windows | 0.16 | 0.00 | 309 | jet | 0.19 | 0.11 |
| 310 | sky | 0.29 | 0.22 | 311 | street | 0.18 | 0.05 | 312 | people | 0.23 | 0.28 |
| 313 | birds | 0.12 | 0.03 | 314 | sky | 0.24 | 0.16 | 315 | pattern | 0.20 | 0.07 |
| 316 | tree | 0.26 | 0.19 | 317 | sky | 0.28 | 0.46 | 318 | buildings | 0.21 | 0.14 |
| 319 | insect | 0.11 | 0.00 | 320 | sky | 0.26 | 0.17 | 321 | buildings | 0.23 | 0.20 |
| 322 | hawk | 0.07 | 0.00 | 323 | tree | 0.28 | 0.22 | 324 | birds | 0.13 | 0.17 |
| 325 | street | 0.18 | 0.00 | 326 | tree | 0.26 | 0.20 | 327 | birds | 0.09 | 0.06 |
| 328 | tree | 0.28 | 0.15 | 329 | water | 0.37 | 0.35 | 330 | tree | 0.25 | 0.19 |
| 331 | woman | 0.08 | 0.03 | 332 | woman | 0.08 | 0.04 | 333 | tree | 0.28 | 0.32 |
| 334 | tree | 0.29 | 0.30 | 335 | stone | 0.21 | 0.17 | 336 | tracks | 0.12 | 0.05 |
| 337 | sky | 0.18 | 0.19 | 338 | car | 0.09 | 0.00 | 339 | plane | 0.17 | 0.09 |
| 340 | water | 0.20 | 0.20 | 341 | water | 0.31 | 0.19 | 342 | birds | 0.14 | 0.02 |
| 343 | plants | 0.09 | 0.02 | 344 | water | 0.36 | 0.23 | 345 | buildings | 0.23 | 0.27 |
| 346 | plane | 0.18 | 0.10 | 347 | boats | 0.29 | 0.20 | 348 | sky | 0.25 | 0.21 |
| 349 | stone | 0.11 | 0.05 | 350 | woman | 0.12 | 0.09 | 351 | walls | 0.11 | 0.07 |
| 352 | grass | 0.12 | 0.09 | 353 | flowers | 0.12 | 0.00 | 354 | sky | 0.28 | 0.35 |
| 355 | close-up | 0.16 | 0.00 | 356 | cactus | 0.08 | 0.00 | 357 | water | 0.29 | 0.24 |
| 358 | buildings | 0.26 | 0.21 | 359 | water | 0.39 | 0.24 | 360 | buildings | 0.25 | 0.26 |
| 361 | insect | 0.13 | 0.00 | 362 | water | 0.32 | 0.29 | 363 | people | 0.28 | 0.45 |
| 364 | people | 0.27 | 0.26 | 365 | people | 0.22 | 0.22 | 366 | sky | 0.20 | 0.13 |
| 367 | snake | 0.07 | 0.08 | 368 | stone | 0.14 | 0.04 | 369 | tree | 0.21 | 0.25 |
| 370 | head | 0.10 | 0.04 | 371 | branch | 0.07 | 0.07 | 372 | birds | 0.09 | 0.07 |
| 373 | night | 0.13 | 0.00 | 374 | people | 0.28 | 0.00 | 375 | clouds | 0.18 | 0.10 |
| 376 | jet | 0.17 | 0.06 | 377 | people | 0.26 | 0.17 | 378 | tree | 0.24 | 0.31 |
| 379 | coast | 0.13 | 0.07 | 380 | zebra | 0.50 | 0.00 | 381 | fish | 0.09 | 0.02 |
| 382 | jet | 0.40 | 0.00 | 383 | people | 0.28 | 0.41 | 384 | people | 0.38 | 0.32 |
| 385 | water | 0.35 | 0.31 | 386 | water | 0.45 | 0.57 | 387 | texture | 0.13 | 0.09 |
| 388 | street | 0.16 | 0.22 | 389 | vegetables | 0.13 | 0.25 | 390 | statues | 0.07 | 0.06 |
| 391 | buildings | 0.23 | 0.04 | 392 | buildings | 0.26 | 0.06 | 393 | water | 0.46 | 0.33 |
| 394 | tree | 0.25 | 0.19 | 395 | buildings | 0.19 | 0.31 | 396 | grass | 0.13 | 0.07 |
| 397 | gardens | 0.17 | 0.05 | 398 | boats | 0.06 | 0.00 | 399 | water | 0.39 | 0.27 |
| 400 | sky | 0.25 | 0.16 | 401 | rock | 0.14 | 0.10 | 402 | car | 0.17 | 0.04 |
| 403 | vegetables | 0.13 | 0.04 | 404 | helicopter | 0.09 | 0.00 | 405 | head | 0.09 | 0.06 |
| 406 | people | 0.29 | 0.27 | 407 | birds | 0.17 | 0.25 | 408 | clouds | 0.28 | 0.28 |
| 409 | closeup | 0.10 | 0.00 | 410 | sky | 0.32 | 0.20 | 411 | sun | 0.40 | 0.40 |
| 412 | water | 0.45 | 0.28 | 413 | people | 0.32 | 0.23 | 414 | flowers | 0.17 | 0.09 |
| 415 | clouds | 0.14 | 0.05 | 416 | food | 0.09 | 0.00 | 417 | sky | 0.28 | 0.26 |
| 418 | water | 0.26 | 0.20 | 419 | woman | 0.13 | 0.10 | 420 | leaves | 0.12 | 0.00 |
| 421 | temple | 0.13 | 0.04 | 422 | sky | 0.25 | 0.18 | 423 | ground | 0.17 | 0.00 |
| 424 | flowers | 0.15 | 0.10 | 425 | tree | 0.26 | 0.25 | 426 | hats | 0.09 | 0.06 |
| 427 | tree | 0.24 | 0.21 | 428 | tree | 0.30 | 0.21 | 429 | tracks | 0.18 | 0.03 |
| 430 | car | 0.17 | 0.00 | 431 | people | 0.23 | 0.23 | 432 | water | 0.27 | 0.15 |
| 433 | water | 0.46 | 0.36 | 434 | buildings | 0.28 | 0.24 | 435 | texture | 0.13 | 0.06 |
| 436 | sky | 0.20 | 0.09 | 437 | birds | 0.11 | 0.07 | 438 | sky | 0.30 | 0.43 |
| 439 | tree | 0.25 | 0.18 | 440 | woods | 0.12 | 0.00 | 441 | windows | 0.35 | 0.10 |
| 442 | buildings | 0.16 | 0.12 | 443 | water | 0.41 | 0.14 | 444 | sky | 0.44 | 0.47 |
| 445 | insect | 0.11 | 0.00 | 446 | hills | 0.11 | 0.10 | 447 | buildings | 0.20 | 0.16 |
| 448 | gun | 0.09 | 0.05 | 449 | woman | 0.12 | 0.00 | 450 | water | 0.54 | 0.49 |
| 451 | buildings | 0.22 | 0.13 | 452 | water | 0.47 | 0.41 | 453 | clouds | 0.22 | 0.17 |
| 454 | people | 0.36 | 0.18 | 455 | walls | 0.09 | 0.03 | 456 | people | 0.51 | 0.19 |
| 457 | people | 0.32 | 0.13 | 458 | sun | 0.39 | 0.42 | 459 | texture | 0.24 | 0.00 |
| 460 | people | 0.32 | 0.16 | 461 | people | 0.23 | 0.19 | 462 | stone | 0.14 | 0.00 |
| 463 | sky | 0.28 | 0.36 | 464 | texture | 0.13 | 0.03 | 465 | street | 0.16 | 0.04 |
| 466 | rock | 0.10 | 0.11 | 467 | water | 0.40 | 0.31 | 468 | boats | 0.26 | 0.00 |
| 469 | clouds | 0.14 | 0.00 | 470 | tree | 0.25 | 0.13 | 471 | birds | 0.13 | 0.15 |
| 472 | water | 0.30 | 0.13 | 473 | people | 0.59 | 0.40 | 474 | stone | 0.10 | 0.06 |
| 475 | ground | 0.13 | 0.13 | 476 | water | 0.19 | 0.22 | 477 | sky | 0.20 | 0.19 |
| 478 | tree | 0.40 | 0.25 | 479 | tree | 0.29 | 0.19 | 480 | pillars | 0.19 | 0.00 |
| 481 | grass | 0.17 | 0.11 | 482 | water | 0.29 | 0.36 | 483 | closeup | 0.13 | 0.00 |
| 484 | buildings | 0.16 | 0.13 | 485 | people | 0.41 | 0.26 | 486 | beach | 0.11 | 0.00 |
| 487 | tree | 0.30 | 0.15 | 488 | water | 0.33 | 0.33 | 489 | flowers | 0.16 | 0.10 |
| 490 | people | 0.36 | 0.43 | 491 | boats | 0.13 | 0.07 | 492 | tree | 0.23 | 0.30 |
| 493 | coast | 0.11 | 0.00 | 494 | birds | 0.22 | 0.19 | 495 | water | 0.44 | 0.45 |
| 496 | skis | 0.08 | 0.00 | 497 | tree | 0.27 | 0.23 | 498 | insect | 0.13 | 0.00 |
| 499 | birds | 0.10 | 0.19 | 500 | plants | 0.10 | 0.04 | | | | |

Table 4.12: For each blob in the standard test set: number of times that the blob occurs in the data, and the number of correct predictions for the first three words with the highest probabilities given the blob.

| count | 1st word | 2nd word | 3rd word | count | 1st word | 2nd word | 3rd word |
|---|---|---|---|---|---|---|---|
| 81 | sky : 22 | water : 25 | snow : 4 | 80 | sky : 37 | water : 32 | clouds : 12 |
| 78 | tree : 14 | sky : 12 | water : 16 | 74 | sky : 32 | water : 24 | snow : 4 |
| 70 | tree : 13 | hills : 3 | water : 18 | 68 | grass : 5 | ocean : 10 | mountain : 3 |
| 68 | sky : 26 | clouds : 8 | water : 35 | 67 | sky : 29 | mountain : 9 | tree : 11 |
| 64 | tree : 14 | water : 17 | walls : 0 | 63 | tree : 14 | grass : 6 | closeup : 1 |
| 63 | grass : 7 | tree : 13 | fish : 7 | 63 | people : 9 | grass : 8 | tree : 16 |
| 62 | birds : 9 | hawk : 3 | snow : 5 | 62 | ocean : 7 | rock : 4 | fish : 4 |
| 62 | sky : 16 | clouds : 11 | tracks : 5 | 60 | tree : 14 | ocean : 3 | water : 10 |
| 60 | woman : 2 | leaves : 8 | nest : 1 | 59 | water : 14 | sky : 12 | clouds : 4 |
| 59 | rock : 7 | tree : 11 | snake : 1 | 58 | water : 13 | people : 4 | reefs : 0 |
| 58 | water : 21 | sky : 18 | mountain : 8 | 58 | tree : 11 | grass : 5 | cat : 3 |
| 57 | water : 20 | house : 1 | tree : 4 | 57 | people : 8 | texture : 6 | costume : 0 |
| 56 | tree : 13 | sky : 12 | ruins : 0 | 55 | grass : 6 | sky : 13 | hills : 2 |
| 55 | grass : 3 | lion : 1 | people : 16 | 55 | buildings : 5 | tree : 11 | people : 6 |
| 54 | water : 15 | clouds : 3 | sky : 13 | 54 | texture : 5 | people : 6 | tree : 4 |
| 54 | tree : 17 | flowers : 5 | gardens : 4 | 54 | grass : 6 | vegetables : 2 | leaves : 5 |
| 54 | tree : 14 | street : 1 | flowers : 3 | 53 | people : 10 | pumpkins : 1 | coast : 2 |
| 53 | clouds : 15 | sky : 19 | sun : 8 | 53 | rock : 10 | ocean : 6 | formation : 0 |
| 53 | sky : 15 | mountain : 5 | water : 18 | 52 | tree : 11 | people : 8 | water : 10 |
| 52 | plants : 1 | texture : 6 | reefs : 3 | 52 | water : 10 | sky : 14 | mountain : 11 |
| 52 | fish : 2 | texture : 1 | grass : 8 | 52 | tree : 11 | fish : 4 | coral : 2 |
| 52 | water : 13 | people : 5 | house : 2 | 52 | tree : 10 | water : 8 | hills : 3 |
| 51 | tree : 6 | grass : 3 | sky : 7 | 51 | water : 14 | hats : 1 | head : 0 |
| 50 | sky : 18 | clouds : 6 | jet : 3 | 50 | flowers : 5 | tree : 9 | people : 8 |
| 50 | people : 14 | close-up : 3 | pattern : 4 | 50 | tree : 7 | people : 3 | pattern : 2 |
| 49 | sky : 23 | plane : 7 | jet : 5 | 49 | rock : 5 | sky : 4 | grass : 3 |
| 49 | grass : 11 | water : 9 | birds : 1 | 49 | tree : 8 | people : 3 | forest : 2 |
| 49 | people : 10 | grass : 3 | tree : 8 | 48 | water : 16 | sky : 16 | coast : 4 |
| 47 | water : 21 | sky : 9 | plane : 2 | 47 | tree : 10 | sky : 5 | flowers : 4 |
| 47 | fish : 1 | ocean : 3 | reefs : 1 | 47 | water : 14 | grass : 5 | hills : 2 |
| 47 | water : 13 | sky : 10 | mountain : 4 | 47 | coral : 2 | reefs : 2 | flowers : 7 |
| 47 | tree : 11 | vegetables : 2 | bears : 1 | 47 | water : 19 | plane : 2 | runway : 1 |
| 46 | tree : 14 | cat : 1 | wolf : 3 | 46 | grass : 4 | mountain : 1 | hills : 1 |
| 46 | people : 17 | tower : 1 | bears : 2 | 46 | rock : 9 | cougar : 0 | cat : 0 |
| 45 | birds : 1 | rock : 5 | nest : 0 | 45 | texture : 3 | pattern : 4 | clouds : 7 |
| 45 | tree : 5 | mountain : 3 | grass : 6 | 45 | water : 15 | close-up : 1 | people : 5 |
| 44 | rock : 5 | tree : 10 | water : 12 | 44 | tree : 11 | fish : 2 | buildings : 3 |
| 44 | stone : 2 | desert : 0 | ruins : 1 | 44 | water : 9 | sky : 8 | rock : 4 |
| 44 | sky : 5 | church : 0 | water : 9 | 44 | tree : 10 | statues : 0 | sand : 0 |
| 44 | water : 19 | coast : 4 | sea : 0 | 44 | ground : 1 | lizard : 0 | face : 1 |
| 44 | pattern : 1 | birds : 6 | texture : 0 | 44 | people : 8 | tree : 9 | village : 1 |
| 43 | water : 21 | jet : 5 | plane : 5 | 43 | birds : 3 | texture : 0 | plants : 5 |
| 43 | sky : 8 | water : 6 | birds : 8 | 43 | tree : 8 | snow : 2 | forest : 0 |
| 43 | water : 20 | rock : 3 | mountain : 3 | 43 | tree : 10 | snow : 0 | vegetables : 4 |
| 43 | mountain : 2 | sky : 15 | jet : 6 | 42 | people : 11 | house : 0 | close-up : 2 |
| 42 | fish : 1 | grass : 4 | bears : 0 | 42 | people : 14 | tree : 10 | beach : 1 |
| 42 | clouds : 0 | sky : 6 | texture : 1 | 42 | tree : 11 | ground : 0 | water : 8 |
| 42 | plane : 4 | tree : 7 | ice : 1 | 41 | flowers : 4 | ground : 4 | vegetables : 1 |
| 41 | tree : 6 | people : 9 | grass : 1 | 41 | water : 6 | people : 9 | grass : 3 |
| 41 | tree : 13 | leaves : 1 | stone : 3 | 41 | water : 12 | tree : 11 | mountain : 5 |
| 41 | sky : 15 | water : 19 | clouds : 9 | 41 | water : 8 | sand : 0 | rock : 4 |
| 41 | face : 3 | close-up : 2 | animal : 2 | 41 | tree : 8 | rock : 3 | coast : 2 |
| 41 | buildings : 6 | tree : 15 | harbor : 0 | 40 | water : 13 | coast : 2 | cliff : 1 |
| 40 | flowers : 3 | plants : 1 | vegetables : 2 | 40 | tree : 9 | flowers : 2 | buildings : 2 |
| 40 | lion : 1 | cat : 2 | mane : 1 | 40 | sky : 6 | birds : 5 | cactus : 4 |
| 39 | birds : 1 | tree : 9 | fungus : 1 | 39 | water : 10 | Scotland : 1 | snow : 3 |
| 39 | water : 6 | snow : 1 | buildings : 5 | 39 | rock : 3 | sky : 7 | coast : 2 |
| 39 | flowers : 6 | gardens : 1 | leaves : 1 | 39 | people : 14 | temple : 1 | street : 2 |
| 38 | people : 6 | buildings : 2 | street : 2 | 38 | water : 12 | people : 2 | sky : 6 |
| 38 | tree : 13 | ocean : 2 | coral : 2 | 38 | people : 7 | water : 15 | sky : 6 |
| 37 | tree : 7 | bears : 1 | snow : 2 | 37 | sky : 5 | elephants : 0 | sand : 1 |
| 37 | fish : 4 | ocean : 4 | tree : 10 | 37 | fish : 3 | texture : 3 | ocean : 3 |
| 36 | stone : 2 | statues : 1 | tree : 9 | 36 | walls : 1 | head : 0 | closeup : 1 |
| 36 | birds : 2 | water : 11 | buildings : 2 | 36 | water : 14 | clouds : 5 | sky : 7 |
| 36 | water : 9 | tree : 8 | stone : 3 | 36 | rock : 7 | helicopter : 0 | insect : 3 |
| 36 | people : 4 | stone : 5 | beach : 1 | 36 | water : 11 | stone : 4 | clouds : 0 |
| 36 | birds : 7 | owl : 4 | night : 4 | 35 | water : 11 | stone : 3 | sky : 9 |
| 35 | people : 9 | snow : 1 | walls : 0 | 35 | jet : 4 | plane : 4 | coast : 4 |
| 35 | stone : 1 | gardens : 2 | reflection : 0 | 35 | sky : 4 | people : 8 | gardens : 1 |
| 35 | water : 4 | people : 13 | street : 2 | 35 | clouds : 2 | snow : 2 | ice : 1 |
| 35 | water : 7 | tree : 6 | sky : 8 | 35 | coast : 0 | water : 7 | rock : 2 |
| 35 | sky : 3 | water : 5 | ocean : 1 | 34 | water : 14 | beach : 1 | horizon : 2 |
| 34 | tree : 6 | mountain : 5 | water : 9 | 34 | flowers : 3 | snow : 2 | leaves : 4 |
| 34 | temple : 0 | rock : 4 | ocean : 4 | 34 | people : 12 | tree : 8 | buildings : 4 |
| 34 | people : 5 | woman : 0 | church : 1 | 34 | sky : 10 | clouds : 10 | birds : 0 |
| 34 | vegetables : 3 | grass : 2 | animal : 0 | 34 | water : 10 | rock : 1 | boats : 1 |
| 34 | people : 12 | tree : 5 | rock : 1 | 33 | tracks : 1 | car : 1 | polar : 0 |
| 33 | water : 8 | rock : 5 | tree : 3 | 33 | buildings : 6 | sky : 2 | coast : 0 |
| 33 | water : 4 | sand : 0 | bears : 1 | 33 | sky : 5 | tree : 8 | buildings : 2 |
| 33 | water : 12 | hills : 0 | Scotland : 2 | 32 | buildings : 5 | prototype : 0 | tracks : 4 |

Table 4.12: Continued.

| count | 1st word | 2nd word | 3rd word | count | 1st word | 2nd word | 3rd word |
|---|---|---|---|---|---|---|---|
| 32 | sky : 3 | gardens : 3 | buildings : 5 | 32 | tree : 8 | buildings : 2 | hills : 1 |
| 32 | buildings : 10 | tree : 4 | water : 8 | 32 | tree : 6 | vegetables : 0 | pumpkins : 0 |
| 32 | ground : 0 | hills : 0 | sky : 6 | 32 | clouds : 9 | ground : 0 | plane : 0 |
| 32 | texture : 3 | pattern : 1 | water : 10 | 32 | sky : 6 | tree : 10 | buildings : 5 |
| 32 | buildings : 5 | boats : 2 | plane : 1 | 32 | mountain : 4 | water : 5 | clouds : 11 |
| 32 | street : 2 | people : 5 | flowers : 1 | 32 | buildings : 4 | people : 12 | street : 1 |
| 31 | birds : 6 | branch : 2 | leaves : 5 | 31 | buildings : 4 | people : 6 | forest : 0 |
| 31 | texture : 1 | pattern : 1 | water : 12 | 31 | sky : 8 | clouds : 1 | jet : 0 |
| 31 | tree : 6 | buildings : 3 | boats : 1 | 31 | sky : 5 | grass : 1 | water : 11 |
| 31 | boats : 2 | tree : 6 | water : 4 | 31 | tree : 5 | buildings : 3 | people : 6 |
| 31 | people : 9 | stone : 2 | snow : 2 | 31 | water : 4 | fish : 0 | people : 7 |
| 31 | water : 15 | rock : 3 | coast : 3 | 30 | tree : 4 | people : 9 | birds : 2 |
| 30 | people : 7 | tree : 5 | buildings : 5 | 30 | water : 17 | sky : 10 | ice : 1 |
| 30 | branch : 2 | tree : 4 | people : 10 | 30 | car : 0 | tracks : 1 | hills : 0 |
| 30 | stone : 5 | buildings : 5 | street : 2 | 30 | tree : 6 | water : 6 | boats : 2 |
| 30 | water : 12 | shore : 1 | gardens : 0 | 30 | water : 17 | plane : 1 | shore : 3 |
| 30 | water : 12 | sky : 5 | desert : 0 | 30 | sky : 6 | texture : 0 | flowers : 0 |
| 30 | sky : 11 | plane : 8 | helicopter : 1 | 30 | water : 8 | fungus : 3 | mushrooms : 3 |
| 30 | temple : 1 | sky : 2 | sculpture : 0 | 30 | buildings : 4 | tree : 8 | arch : 0 |
| 30 | rock : 1 | people : 7 | boats : 3 | 30 | vegetables : 1 | food : 1 | branch : 2 |
| 29 | hills : 3 | coast : 1 | sky : 7 | 29 | buildings : 7 | street : 2 | entrance : 0 |
| 29 | water : 9 | plane : 5 | sky : 9 | 29 | plane : 3 | jet : 3 | grass : 3 |
| 29 | birds : 5 | ocean : 0 | plants : 0 | 29 | people : 14 | shrine : 0 | tree : 2 |
| 29 | buildings : 5 | stone : 4 | tree : 5 | 29 | fish : 3 | mountain : 3 | mushrooms : 0 |
| 29 | people : 10 | tree : 5 | water : 7 | 29 | jet : 3 | shore : 0 | water : 10 |
| 29 | tree : 5 | water : 0 | people : 5 | 29 | mushrooms : 1 | closeup : 0 | water : 3 |
| 28 | plants : 1 | leaves : 1 | ocean : 3 | 28 | people : 12 | tree : 4 | display : 0 |
| 28 | sky : 6 | water : 14 | mountain : 3 | 28 | sky : 13 | shore : 1 | water : 9 |
| 28 | people : 3 | sky : 1 | buildings : 2 | 28 | snow : 0 | water : 11 | sky : 9 |
| 28 | tree : 8 | cat : 0 | mane : 0 | 28 | water : 13 | clouds : 4 | tree : 7 |
| 27 | birds : 5 | owl : 4 | night : 5 | 27 | sky : 5 | wolf : 1 | clouds : 1 |
| 27 | temple : 1 | statues : 0 | arch : 0 | 27 | tree : 4 | snow : 3 | rock : 3 |
| 27 | sky : 5 | rock : 2 | flight : 0 | 27 | buildings : 1 | people : 12 | bridge : 1 |
| 27 | water : 10 | coast : 1 | sand : 0 | 27 | sky : 5 | windows : 2 | cactus : 1 |
| 27 | tree : 5 | birds : 3 | branch : 1 | 27 | flowers : 2 | plants : 3 | people : 7 |
| 26 | street : 1 | car : 1 | people : 8 | 26 | people : 6 | woods : 0 | grass : 3 |
| 26 | people : 7 | buildings : 5 | street : 1 | 26 | vegetables : 1 | water : 9 | reefs : 1 |
| 26 | head : 1 | grass : 3 | ground : 0 | 26 | woman : 1 | people : 4 | lion : 0 |
| 26 | clouds : 1 | water : 5 | river : 0 | 26 | walls : 1 | stone : 2 | ruins : 1 |
| 26 | plants : 1 | flowers : 1 | leaves : 2 | 26 | people : 11 | buildings : 3 | street : 3 |
| 26 | people : 7 | costume : 1 | palace : 0 | 26 | people : 7 | birds : 1 | ground : 0 |
| 26 | water : 9 | rock : 1 | boats : 3 | 26 | birds : 4 | sky : 5 | night : 0 |
| 26 | closeup : 2 | food : 2 | woman : 3 | 25 | water : 9 | cat : 0 | head : 0 |
| 25 | clouds : 0 | stone : 0 | car : 2 | 25 | car : 1 | tracks : 1 | street : 0 |
| 25 | sky : 4 | water : 6 | people : 1 | 25 | stone : 1 | house : 0 | buildings : 1 |
| 25 | sky : 8 | clouds : 3 | jet : 1 | 25 | water : 6 | sky : 2 | palace : 0 |
| 25 | walls : 1 | architecture : 0 | bridge : 0 | 25 | tree : 7 | flowers : 0 | trunk : 1 |
| 25 | textile : 3 | background : 3 | pattern : 5 | 24 | water : 3 | snow : 2 | beach : 0 |
| 24 | clouds : 4 | ice : 0 | beach : 5 | 24 | buildings : 3 | people : 10 | temple : 0 |
| 24 | birds : 6 | branch : 2 | flight : 0 | 24 | buildings : 1 | boats : 0 | statues : 1 |
| 24 | people : 4 | mountain : 2 | water : 5 | 24 | snow : 1 | ice : 0 | sun : 2 |
| 24 | sky : 11 | clouds : 4 | mountain : 1 | 24 | clouds : 3 | water : 9 | mountain : 3 |
| 24 | hills : 1 | sun : 4 | water : 9 | 24 | buildings : 1 | rock : 1 | water : 7 |
| 24 | beach : 0 | branch : 1 | house : 1 | 24 | boats : 4 | buildings : 5 | market : 0 |
| 24 | reefs : 0 | head : 1 | wildlife : 0 | 24 | birds : 1 | gardens : 1 | stone : 0 |
| 23 | ground : 3 | sky : 1 | mushrooms : 2 | 23 | stone : 0 | people : 8 | fence : 0 |
| 23 | people : 3 | ocean : 0 | beach : 1 | 23 | buildings : 6 | people : 9 | temple : 1 |
| 23 | woman : 2 | vegetables : 1 | buildings : 1 | 23 | tree : 7 | elephants : 0 | trunk : 0 |
| 23 | sky : 4 | tables : 0 | shop : 1 | 23 | people : 4 | street : 2 | tree : 4 |
| 23 | people : 9 | pillars : 1 | buildings : 5 | 23 | jet : 2 | plane : 2 | stone : 1 |
| 23 | tree : 8 | flowers : 2 | food : 0 | 23 | pillars : 3 | stone : 4 | palace : 0 |
| 23 | water : 10 | coast : 2 | waves : 2 | 23 | tree : 2 | sky : 4 | buildings : 5 |
| 22 | beach : 0 | snake : 0 | birds : 1 | 22 | people : 4 | woman : 1 | closeup : 1 |
| 22 | woman : 0 | people : 1 | hats : 0 | 22 | water : 6 | buildings : 2 | stone : 0 |
| 22 | people : 10 | gardens : 0 | pillars : 0 | 22 | street : 1 | car : 4 | shop : 0 |
| 22 | mountain : 1 | grass : 0 | people : 3 | 22 | sky : 6 | water : 7 | hills : 2 |
| 22 | jet : 1 | plane : 1 | sea : 0 | 22 | ground : 1 | head : 0 | food : 1 |
| 22 | sky : 5 | wildlife : 2 | hawk : 0 | 22 | tree : 3 | stone : 3 | ruins : 2 |
| 22 | water : 3 | fish : 1 | rock : 0 | 22 | birds : 3 | head : 1 | mushrooms : 0 |
| 22 | field : 0 | water : 4 | tree : 3 | 22 | woman : 1 | people : 5 | sky : 1 |
| 22 | people : 1 | tree : 6 | stone : 1 | 22 | tree : 5 | lion : 0 | people : 7 |
| 21 | water : 3 | grass : 4 | tree : 6 | 21 | clouds : 1 | plane : 0 | ruins : 1 |
| 21 | gardens : 1 | flowers : 1 | statues : 0 | 21 | mountain : 2 | clouds : 2 | sky : 7 |
| 21 | rock : 2 | birds : 2 | crystal : 0 | 21 | people : 8 | boats : 1 | street : 0 |
| 21 | sky : 3 | formula : 0 | pyramid : 0 | 21 | snow : 5 | helicopter : 0 | hills : 1 |
| 21 | sky : 9 | ice : 0 | snow : 1 | 21 | sky : 3 | walls : 0 | street : 1 |
| 21 | texture : 6 | detail : 0 | pattern : 6 | 21 | texture : 2 | house : 0 | pattern : 2 |
| 20 | gun : 1 | windows : 0 | cliff : 0 | 20 | clouds : 2 | runway : 0 | sky : 4 |
| 20 | buildings : 4 | walls : 1 | sky : 5 | 20 | insect : 0 | sand : 0 | pattern : 4 |
| 20 | street : 4 | sky : 3 | windows : 1 | 20 | grass : 2 | tree : 2 | birds : 0 |
| 20 | people : 7 | skis : 0 | mountain : 0 | 20 | people : 3 | snow : 2 | sunset : 2 |
| 20 | pattern : 2 | texture : 3 | jet : 0 | 20 | water : 4 | snow : 2 | owl : 1 |
| 20 | people : 9 | closeup : 3 | owl : 1 | 20 | rock : 6 | water : 6 | mountain : 2 |
| 20 | people : 5 | ocean : 0 | beach : 1 | 19 | people : 6 | sunset : 0 | tree : 6 |

Table 4.12: Continued.

| count | 1st word | 2nd word | 3rd word | count | 1st word | 2nd word | 3rd word |
|---|---|---|---|---|---|---|---|
| 19 | tracks : 1 | car : 1 | water : 6 | 19 | stone : 4 | walls : 0 | grass : 2 |
| 19 | walls : 2 | shop : 0 | outside : 0 | 19 | street : 3 | car : 3 | buildings : 3 |
| 19 | flowers : 1 | hills : 0 | snow : 0 | 19 | street : 0 | branch : 1 | birds : 2 |
| 19 | textile : 0 | entrance : 1 | designs : 0 | 18 | food : 0 | ground : 0 | sky : 3 |
| 18 | statues : 1 | tree : 4 | night : 0 | 18 | jet : 1 | plane : 2 | runway : 1 |
| 18 | sky : 4 | buildings : 4 | tree : 2 | 18 | sky : 1 | pattern : 1 | leaves : 2 |
| 18 | street : 1 | buildings : 3 | flowers : 0 | 18 | water : 6 | coast : 0 | sea : 0 |
| 18 | owl : 0 | night : 0 | birds : 2 | 18 | people : 6 | flowers : 1 | plants : 0 |
| 18 | forest : 1 | people : 4 | stone : 2 | 18 | gardens : 0 | tree : 5 | street : 0 |
| 18 | texture : 1 | close-up : 2 | field : 0 | 18 | temple : 0 | people : 5 | stone : 0 |
| 17 | buildings : 2 | windows : 1 | stone : 0 | 17 | texture : 1 | close-up : 1 | pattern : 1 |
| 17 | head : 1 | rock : 1 | woman : 3 | 17 | helicopter : 0 | windows : 0 | stone : 4 |
| 17 | people : 7 | textile : 0 | columns : 0 | 17 | water : 5 | city : 1 | street : 2 |
| 17 | sky : 6 | plane : 0 | helicopter : 0 | 17 | hawk : 0 | snow : 1 | birds : 0 |
| 17 | buildings : 4 | street : 2 | people : 8 | 17 | church : 1 | statues : 0 | people : 7 |
| 17 | grass : 0 | fence : 1 | palace : 1 | 17 | rock : 0 | shore : 1 | head : 1 |
| 16 | people : 3 | sky : 5 | windows : 0 | 16 | hats : 1 | rock : 1 | night : 0 |
| 16 | buildings : 1 | people : 7 | tree : 3 | 16 | sky : 2 | market : 0 | leaves : 1 |
| 16 | waves : 1 | snow : 2 | boats : 2 | 16 | shrine : 0 | windows : 0 | clouds : 1 |
| 16 | buildings : 3 | water : 9 | boats : 3 | 16 | texture : 1 | background : 1 | pattern : 0 |
| 16 | people : 3 | bridge : 1 | beach : 1 | 16 | pattern : 1 | texture : 1 | boats : 3 |
| 16 | beach : 0 | tree : 3 | island : 0 | 16 | architecture : 2 | bridge : 2 | stone : 0 |
| 15 | walls : 1 | pillars : 0 | house : 2 | 15 | boats : 3 | sunset : 0 | waves : 0 |
| 15 | pattern : 1 | textile : 1 | harbor : 0 | 15 | people : 4 | sky : 2 | costume : 0 |
| 15 | jet : 1 | plane : 2 | cliff : 0 | 15 | water : 6 | temple : 0 | sky : 3 |
| 15 | people : 5 | buildings : 2 | street : 0 | 15 | buildings : 0 | texture : 0 | vegetables : 1 |
| 15 | ground : 2 | lizard : 1 | mushrooms : 1 | 15 | stone : 1 | close-up : 1 | crystal : 1 |
| 14 | boats : 1 | water : 4 | statues : 0 | 14 | birds : 1 | clouds : 1 | dunes : 0 |
| 14 | coast : 1 | water : 2 | ice : 0 | 14 | buildings : 3 | sun : 0 | tables : 0 |
| 14 | flowers : 0 | gardens : 0 | leaves : 0 | 14 | people : 4 | buildings : 4 | water : 3 |
| 14 | seals : 0 | woods : 0 | rock : 1 | 14 | stone : 0 | water : 3 | tree : 0 |
| 14 | woman : 0 | fish : 1 | close-up : 2 | 14 | insect : 0 | head : 0 | shore : 0 |
| 14 | food : 0 | people : 4 | nest : 0 | 14 | plane : 1 | jet : 1 | fish : 1 |
| 14 | buildings : 4 | skis : 0 | people : 3 | 14 | helicopter : 0 | water : 6 | boats : 0 |
| 14 | people : 4 | snow : 0 | temple : 0 | 14 | buildings : 1 | street : 0 | house : 0 |
| 13 | ground : 0 | pyramid : 0 | plants : 0 | 13 | boats : 0 | rock : 0 | people : 4 |
| 13 | water : 3 | car : 6 | turn : 1 | 13 | pillars : 2 | stone : 2 | forest : 0 |
| 13 | people : 6 | display : 0 | elephants : 1 | 12 | sun : 5 | sunset : 3 | beach : 0 |
| 12 | leaves : 0 | pyramid : 0 | ocean : 0 | 12 | vegetables : 3 | bridge : 0 | hats : 0 |
| 12 | night : 0 | birds : 0 | owl : 0 | 12 | snake : 1 | branch : 1 | crystal : 1 |
| 12 | mountain : 0 | tree : 1 | animal : 0 | 12 | sky : 4 | plane : 1 | f-16 : 0 |
| 12 | tree : 5 | people : 5 | snow : 0 | 12 | columns : 0 | stone : 1 | sculpture : 0 |
| 12 | night : 0 | birds : 1 | owl : 0 | 12 | people : 5 | shop : 0 | street : 1 |
| 11 | closeup : 0 | insect : 0 | leaf : 0 | 11 | sky : 2 | tree : 0 | beach : 0 |
| 11 | closeup : 0 | food : 0 | river : 0 | 11 | buildings : 3 | house : 1 | car : 0 |
| 11 | plane : 1 | bridge : 0 | boats : 1 | 11 | buildings : 1 | display : 1 | street : 0 |
| 11 | sky : 0 | temple : 0 | bridge : 0 | 11 | ground : 0 | plane : 1 | insect : 0 |
| 11 | sky : 3 | architecture : 0 | car : 0 | 11 | sea : 0 | beach : 1 | sun : 0 |
| 11 | mountain : 1 | textile : 0 | pattern : 0 | 11 | birds : 3 | buildings : 1 | food : 0 |
| 11 | plane : 2 | jet : 1 | kauai : 0 | 10 | boats : 0 | sea : 0 | water : 3 |
| 10 | windows : 1 | house : 0 | tables : 0 | 10 | woman : 1 | saguaro : 0 | doors : 1 |
| 10 | water : 2 | windows : 0 | flowers : 0 | 10 | street : 0 | horizon : 0 | entrance : 0 |
| 10 | street : 1 | doors : 0 | house : 1 | 10 | street : 1 | people : 3 | entrance : 1 |
| 10 | birds : 1 | sky : 2 | night : 0 | 10 | buildings : 1 | people : 2 | Scotland : 0 |
| 10 | display : 1 | temple : 2 | food : 1 | 10 | people : 4 | buildings : 2 | street : 1 |
| 10 | windows : 1 | flowers : 0 | buildings : 3 | 10 | people : 3 | wildlife : 1 | designs : 0 |
| 9 | street : 2 | columns : 1 | cactus : 0 | 9 | people : 2 | clouds : 0 | flight : 1 |
| 9 | f-16 : 0 | hills : 0 | beach : 1 | 9 | crystal : 0 | hats : 0 | jet : 0 |
| 9 | tracks : 0 | buildings : 2 | car : 0 | 9 | hawk : 0 | branch : 0 | stone : 1 |
| 9 | entrance : 0 | buildings : 1 | palace : 0 | 9 | sky : 1 | temple : 0 | elephants : 0 |
| 8 | pillars : 0 | architecture : 0 | sculpture : 0 | 8 | texture : 0 | pattern : 0 | market : 0 |
| 8 | insect : 0 | flowers : 1 | plants : 0 | 8 | woods : 0 | columns : 0 | beach : 0 |
| 8 | car : 0 | market : 0 | temple : 0 | 8 | people : 0 | bridge : 0 | street : 0 |
| 8 | water : 4 | tree : 1 | tracks : 0 | 8 | background : 2 | leaves : 0 | plants : 0 |
| 8 | leaf : 0 | insect : 0 | leaves : 1 | 8 | coast : 1 | boats : 1 | stone : 0 |
| 7 | buildings : 1 | street : 1 | shop : 0 | 7 | windows : 0 | buildings : 2 | walls : 0 |
| 7 | birds : 1 | stone : 0 | night : 1 | 7 | art : 0 | textile : 0 | pattern : 0 |
| 7 | harbor : 0 | skis : 0 | ruins : 0 | 6 | close-up : 0 | crystal : 0 | birds : 2 |
| 6 | street : 1 | car : 0 | buildings : 1 | 6 | vegetables : 0 | food : 0 | insect : 0 |
| 6 | sky : 2 | harbor : 0 | plane : 1 | 6 | street : 1 | windows : 1 | palace : 0 |
| 5 | coast : 0 | crystal : 0 | close-up : 0 | 5 | people : 2 | water : 1 | skis : 1 |
| 5 | sun : 2 | sunset : 0 | texture : 2 | 5 | sky : 1 | jet : 0 | plane : 0 |
| 5 | cactus : 0 | pillars : 0 | columns : 0 | 5 | smoke : 0 | sky : 2 | hills : 0 |
| 5 | horizon : 2 | people : 1 | prototype : 0 | 5 | turn : 0 | harbor : 0 | tables : 0 |
| 5 | wildlife : 1 | plane : 1 | fish : 1 | 5 | street : 1 | sunset : 0 | pumpkins : 0 |
| 5 | clouds : 0 | shore : 0 | island : 0 | 5 | gun : 0 | helicopter : 0 | tree : 2 |
| 5 | pattern : 3 | texture : 3 | crystal : 2 | 5 | house : 0 | water : 0 | doors : 0 |
| 4 | tree : 1 | fungus : 0 | mushrooms : 0 | 4 | insect : 0 | boats : 0 | walls : 0 |
| 4 | street : 1 | people : 1 | house : 1 | 3 | water : 0 | fish : 0 | plane : 0 |
| 3 | entrance : 0 | textile : 0 | background : 0 | 3 | nest : 0 | smoke : 0 | jet : 0 |
| 3 | cliff : 0 | ships : 1 | dunes : 0 | 2 | skis : 0 | outside : 0 | perch : 0 |
| 2 | jet : 0 | plane : 0 | boats : 0 | 2 | zebra : 0 | animal : 0 | birds : 0 |
| 2 | temple : 0 | branch : 0 | designs : 0 | 1 | insect : 0 | roofs : 0 | hunter : 0 |
| 1 | textile : 1 | hills : 0 | pattern : 1 | 1 | nest : 0 | birds : 1 | zebra : 0 |
| 1 | doors : 0 | house : 0 | hats : 0 | 0 | windows : 0 | house : 0 | entrance : 0 |
| 0 | background : 0 | textile : 0 | texture : 0 | 0 | turn : 0 | car : 0 | tracks : 0 |

## 4.3 Evaluating the results

There is no ground truth result that can be used for an objective evaluation of the performance of the proposed method. However, the performance of the method can be measured to a certain extent by comparing it with

- Empirical word densities,
- Co-occurrences of words and blobs.

### 4.3.1 Predicting empirical word densities

In Corel data set, the annotators typically use common words, such as `sky`, `water`, `people`, and fewer less common words such as `tiger`. As a null hypothesis, one can predict the most common words for all the images in the set. The correct prediction rates can then be used as a baseline for evaluating the performance of the proposed method. Note that, although this baseline will already be quite high, it would be significantly lower if the empirical density (shown in Figure 4.2) were flatter. Therefore, better prediction results will provide us a reason to reject the hypothesis that the performance of the method is merely a product of chance factors. Thus, for the Corel data set, the increment of performance over the empirical density is a sensible indicator.

Table 4.13 shows the recall and precision values for the most frequent four words. Both recall and precision values for the other words are zero. When we compare the results with the Figures 4.11, 4.12 and 4.13, it is easy to see that, although recall values are higher, precision values are lower than the base results. That is, when the empirical word densities are used for prediction the performance (the percentage of correct predictions) is worse than the performance of the proposed method.

Table 4.13: Recall and precision values for the first four words with the highest occurrence frequencies.

| | 'water' | 'sky' | 'tree' | 'people' |
|---|---|---|---|---|
| training | 1.0000 - 0.2167 | 0.9926 - 0.1899 | 0.8934 - 0.2083 | 0.3658 - 0.1676 |
| standard test | 1.0000 - 0.2253 | 0.9936 - 0.1873 | 0.8944 - 0.2045 | 0.3487 - 0.1758 |
| novel test | 1.0000 - 0.2419 | 0.9723 - 0.2082 | 0.7370 - 0.2394 | 0.1480 - 0.1408 |

Table 4.14: KL divergence for each of ten experimental data sets on training, standard test and novel test sets, using empirical word density.

| set | training | standard test | novel test |
|---|---|---|---|
| 001 | 4.8500 | 4.8425 | 4.8395 |
| 002 | 4.7754 | 4.7957 | 5.0540 |
| 003 | 4.7782 | 4.7230 | 4.9863 |
| 004 | 4.9256 | 4.9672 | 4.8187 |
| 005 | 4.7701 | 4.7888 | 5.0684 |
| 006 | 4.8627 | 4.8951 | 5.0041 |
| 007 | 5.0310 | 4.9844 | 4.9499 |
| 008 | 4.8519 | 4.8155 | 5.0124 |
| 009 | 4.9238 | 4.9250 | 4.9968 |
| 010 | 4.6895 | 4.6788 | 4.9150 |

Table 4.15: Normalized classification score for each of ten experimental data sets on training, standard test and novel test sets, using empirical word density.

| | training | standard test | novel test |
|---|---|---|---|
| 001 | 0.1686 | 0.1701 | 0.1857 |
| 002 | 0.1743 | 0.1739 | 0.1760 |
| 003 | 0.1744 | 0.1905 | 0.1699 |
| 004 | 0.1643 | 0.1618 | 0.1970 |
| 005 | 0.1803 | 0.1800 | 0.1603 |
| 006 | 0.1775 | 0.1729 | 0.1628 |
| 007 | 0.1529 | 0.1583 | 0.2018 |
| 008 | 0.1784 | 0.1848 | 0.1730 |
| 009 | 0.1742 | 0.1689 | 0.1666 |
| 010 | 0.1875 | 0.1925 | 0.1595 |

Table 4.16: Word prediction measures for each of ten experimental data sets on training, standard test and novel test sets, using empirical word density.

| | training | standard test | novel test |
|---|---|---|---|
| 001 | 0.1851 | 0.1864 | 0.1993 |
| 002 | 0.1903 | 0.1896 | 0.1883 |
| 003 | 0.1908 | 0.2067 | 0.1829 |
| 004 | 0.1801 | 0.1773 | 0.2093 |
| 005 | 0.1960 | 0.1955 | 0.1733 |
| 006 | 0.1931 | 0.1886 | 0.1759 |
| 007 | 0.1676 | 0.1730 | 0.2138 |
| 008 | 0.1934 | 0.1996 | 0.1857 |
| 009 | 0.1888 | 0.1835 | 0.1793 |
| 010 | 0.2048 | 0.2094 | 0.1735 |

Table 4.17: Comparison of the prediction performances using empirical word densities with the proposed method for training, standard test, and novel test sets. The values are obtained by averaging the results of ten sets. **Left:** results for predictions using empirical word densities, **right :** results for proposed method. KL refers to KL divergence values where smaller is better, NS refers to normalized classification score where larger is better, and PR refers to word prediction rate where larger is better.

| | KL | NS | PR |
|---|---|---|---|
| training | 4.8458 - 3.5635 | 0.1732 - 0.2598 | 0.1894 - 0.2740 |
| standard test | 4.8416 - 5.1180 | 0.1754 - 0.2062 | 0.1914 - 0.2211 |
| novel test | 4.9645 - 5.2508 | 0.1753 - 0.2006 | 0.1881 - 0.2132 |

Table 4.14 – 4.16 show the results for KL-divergence, normalized classification score and word prediction measure respectively for each of ten experimental data sets, on training, standard test and novel test sets for the predictions using empirical word densities. In Table 4.17, the results are compared with the results of the proposed method. The comparison is based on the values obtained by averaging ten experimental data sets. For KL divergence (KL), smaller values represent better prediction, since it means that the predicted word distribution is closer to the target distribution. For normalized classification score (NS) and for word prediction rate (PR) larger values are better. As it can be seen, the proposed method is better than the predictions using empirical word densities (about 50% for NS, 45% for PR and 26% for KL on the training set).

### 4.3.2 Co-occurrences as the probability table

It is possible to model the joint probability of words and blobs using the co-occurrences of words and blobs in the data, and compare the performance against the proposed method. In that case, the predictions are based on the probability table consisting of the co-occurrences of words and blobs.

Table 4.18 – 4.20 show the results for KL-divergence, normalized classification score and word prediction measure respectively for each of ten experimental data sets, on training, standard test and novel test sets using co-occurrences as the probability table.

Table 4.18: KL divergence for each of ten experimental data sets on training, standard test and novel test sets, using co-occurrences as the probability table.

| set | training | standard test | novel test |
|---|---|---|---|
| 001 | 4.0456 | 4.5402 | 4.6444 |
| 002 | 3.9705 | 4.4811 | 4.7923 |
| 003 | 3.9994 | 4.4438 | 4.7796 |
| 004 | 4.1201 | 4.6884 | 4.6631 |
| 005 | 3.9847 | 4.5012 | 4.8926 |
| 006 | 4.0066 | 4.5541 | 4.8290 |
| 007 | 4.2016 | 4.6772 | 4.7516 |
| 008 | 4.0456 | 4.5179 | 4.8094 |
| 009 | 4.1329 | 4.6650 | 4.7494 |
| 010 | 3.9201 | 4.3591 | 4.7401 |

Table 4.19: Normalized classification scores for each of ten experimental data sets on training, standard test and novel test sets, using co-occurrences as the probability table.

| set | training | standard test | novel test |
|---|---|---|---|
| 001 | 0.2153 | 0.2071 | 0.2247 |
| 002 | 0.2223 | 0.2046 | 0.2131 |
| 003 | 0.2212 | 0.2129 | 0.2031 |
| 004 | 0.2003 | 0.1787 | 0.2249 |
| 005 | 0.2329 | 0.2143 | 0.1931 |
| 006 | 0.2267 | 0.1928 | 0.2034 |
| 007 | 0.2106 | 0.1971 | 0.2309 |
| 008 | 0.2263 | 0.2153 | 0.2057 |
| 009 | 0.2193 | 0.2098 | 0.2017 |
| 010 | 0.2254 | 0.2152 | 0.1825 |

Table 4.20: Prediction measures for each of ten experimental data sets on training, standard test and novel test sets, using co-occurrences as the probability table.

| set | training | standard test | novel test |
|---|---|---|---|
| 001 | 0.2310 | 0.2229 | 0.2379 |
| 002 | 0.2374 | 0.2198 | 0.2249 |
| 003 | 0.2369 | 0.2287 | 0.2156 |
| 004 | 0.2155 | 0.1940 | 0.2368 |
| 005 | 0.2478 | 0.2293 | 0.2057 |
| 006 | 0.2415 | 0.2082 | 0.2160 |
| 007 | 0.2245 | 0.2113 | 0.2425 |
| 008 | 0.2405 | 0.2297 | 0.2179 |
| 009 | 0.2333 | 0.2238 | 0.2140 |
| 010 | 0.2419 | 0.2317 | 0.1961 |

The results are compared with the results of the proposed system. Table 4.21 shows the values obtained by averaging ten sets, for using co-occurrences as the probability table and for the propose method. The results clearly show that the proposed method has a higher performance than using co-occurrences. This indicates that, the EM algorithm fixes the correspondence ambiguities by iterating over the data.

Table 4.21: Comparison of prediction performances for using co-occurrences of words and blobs with the proposed method, for training, standard test, and novel test sets. The values are obtained by averaging the results of ten sets. **Left:** the results when co-occurrences are used as the probability table, and **right:** results of the proposed method. KL refers to KL divergence values where smaller is better, NS refers to normalized classification score where larger is better, and PR refers to word prediction rate where larger is better.

| | KL | NS | PR |
|---|---|---|---|
| training | 4.0427 - 3.5635 | 0.2200 - 0.2598 | 0.2350 - 0.2740 |
| standard test | 4.5428 - 5.1180 | 0.2048 - 0.2062 | 0.2199 - 0.2211 |
| novel test | 4.7651 - 5.2508 | 0.2083 - 0.2006 | 0.2206 - 0.2132 |

Table 4.22 shows the log-likelihood and mutual information when co-occurrences are used as the probability table. The results show that, the probability table learned by the proposed method is better than the co-occurrences.

Table 4.22: Log-likelihood and mutual information values for each of ten experimental data sets on training, standard test and novel test sets, using co-occurrences as the probability table.

| set | log-likelihood | mutual info |
|---|---|---|
| 001 | -6.18e+004 | 0.8490 |
| 002 | -6.51e+004 | 0.8712 |
| 003 | -6.34e+004 | 0.8345 |
| 004 | -6.45e+004 | 0.8663 |
| 005 | -6.34e+004 | 0.8207 |
| 006 | -6.26e+004 | 0.8935 |
| 007 | -6.40e+004 | 0.9044 |
| 008 | -6.36e+004 | 0.8631 |
| 009 | -6.40e+004 | 0.8661 |
| 010 | -6.15e+004 | 0.8034 |

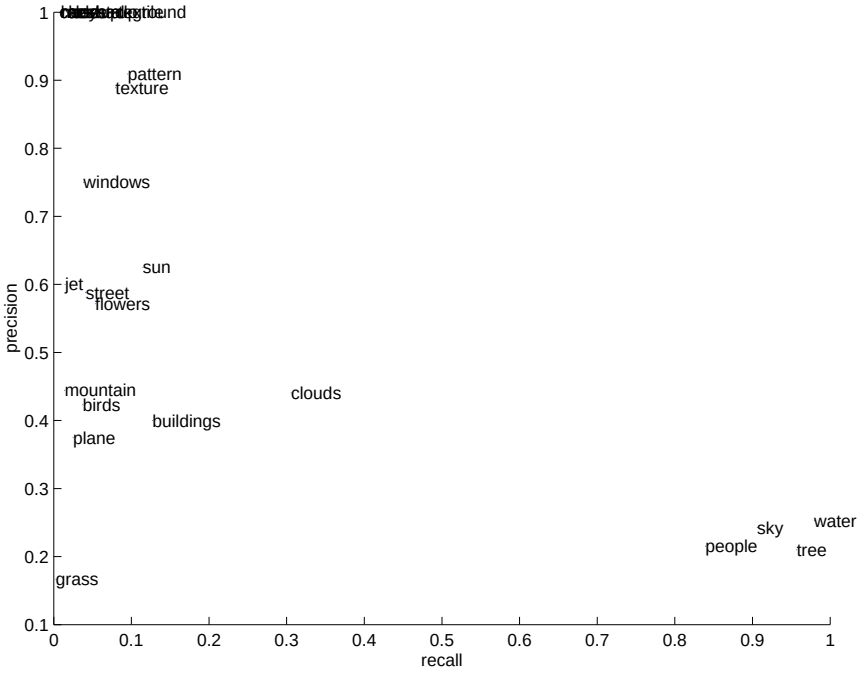



Figure 4.14: Recall versus precision values for the training set, using co-occurrences as the probability table. Number of words which has non-zero recall and precision values is 26.

Figure 4.14 – 4.16 show the recall and precision values for the training, standard test and novel test sets for the first experimental data set, using the co-occurrences as the probability table. When compared with the results with the proposed method (see Figures 4.11, 4.12 and 4.13), it is seen that the number of predicted words are smaller and for the predicted words recall values are lower. For the high frequency words, although recall values are higher, precision is lower than the values of the proposed method.

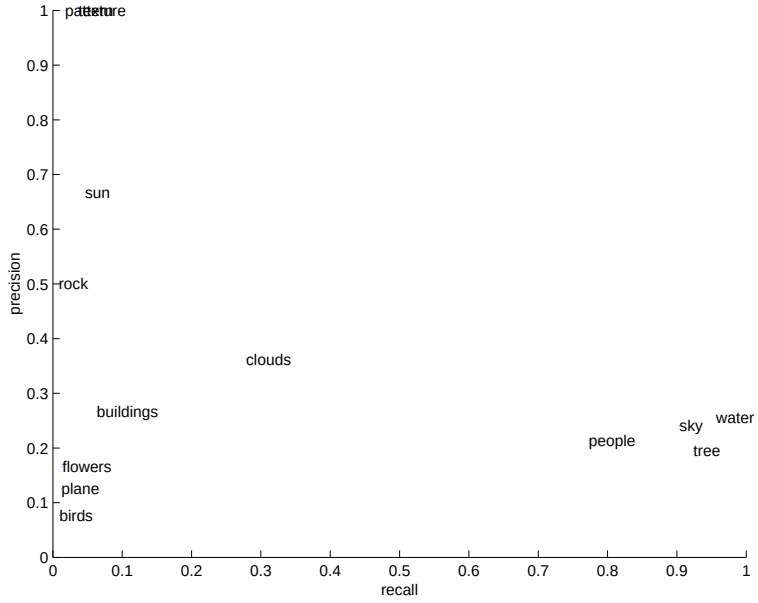



Figure 4.15: Recall versus precision values for the standard test set, using co-occurrences as the probability table. Number of words which has non-zero recall and precision values is 13.

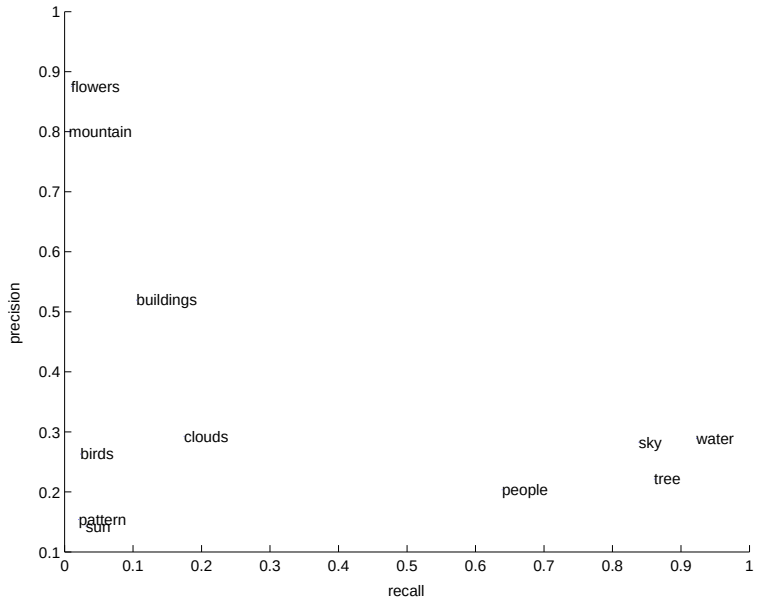



Figure 4.16: Recall versus precision values for the novel test set using co-occurrences as the probability table. Number of words which has non-zero recall and precision values is 11.

## 4.4 Parameters effecting the performance

Until now, the experiments are carried out with the fixed parameter set explained in Section 4.2. In this section, we analyze the effect of changing the parameters on the performance. The major parameters which have an impact on the performance of the system are:

- initialization and number of iterations in EM,
- the number of clusters in k-means, and
- feature selection.

Let us investigate the stability and vulnerability of the proposed system with respect to the above parameters.

### 4.4.1 Effect of initialization and number of iterations in EM

For the experiments, probability table is initialized with the co-occurrences of the blobs and words in the data. Brown *et. al.* prove that Model 1 has a unique local maximum so that parameters derived for it in a series of EM iterations do not depend on the starting point. The following experiments supports this proof.

Table 4.23 shows the results for (i) initialization by co-occurrences, (ii) uniform initialization and (iii) three different random initializations, on the training set. As it is seen, EM always converges and the values are close to each other for all types of initialization.

Table 4.23: Effect of initializations in EM. Probability table is initialized (i) using co-occurrences of blobs and words, (ii) uniformly, and (iii) randomly. Log-likelihood, mutual information and prediction measure values are used for comparison.

| initialization type | log-likelihood | mutual information | prediction measure |
|---|---|---|---|
| using co-occurrences | -5.64e+004 | 3.1853 | 0.2708 |
| uniform initialization | -5.64e+004 | 3.1836 | 0.2707 |
| random initialization-1 | -5.64e+004 | 3.1844 | 0.2706 |
| random initialization-2 | -5.64e+004 | 3.1842 | 0.2709 |
| random initialization-2 | -5.64e+004 | 3.1853 | 0.2707 |

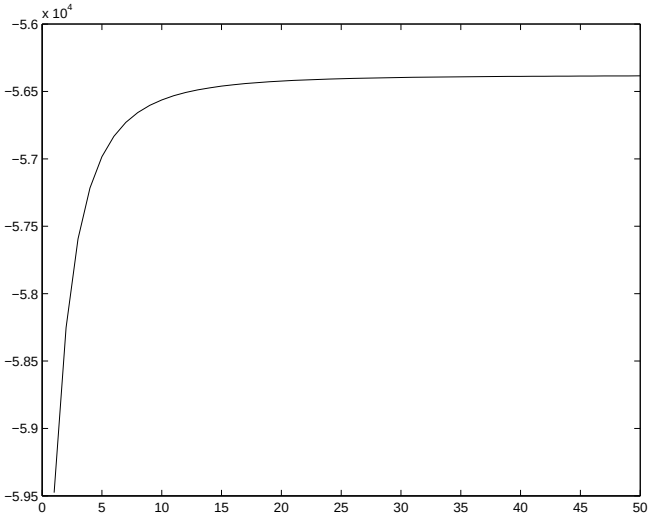



Figure 4.17: Log-likelihood during 50 EM iterations.

During the Expectation maximization algorithm E and M steps are iteratively repeated. For the experiments, 50 iterations are used to guarantee the convergence for any cases. Figures 4.17 – 4.19 show the log-likelihood, mutual information, and word prediction values during 50 EM iterations.

### 4.4.2 Effect of number of clusters in k-means on the performance

For the experiments, the default value for the number of blobs is chosen as $k = 500$ in k-means algorithm. The choice of $k$ is arbitrary. We assume that each word is represented 3-4 region types (for example `sky` may be in different colors: red, blue, dark blue, etc.). Therefore, we choose $k$ as 500, since the number of words in the vocabulary is about 150. In this section, we analyze the results for different $k$ values.

Figure 4.20 and 4.21, plot the log-likelihood and prediction values for different $k$ values. As the figures indicate, the values are increasing with the increased $k$ value. Due to the time and memory limits, the results are tested only up to 500 clusters.

K-means algorithm is not the best way for clustering. In Section 4.6, some alternative strategies will be discussed. It remains an open problem, to choose the best clustering method for the proposed approach.

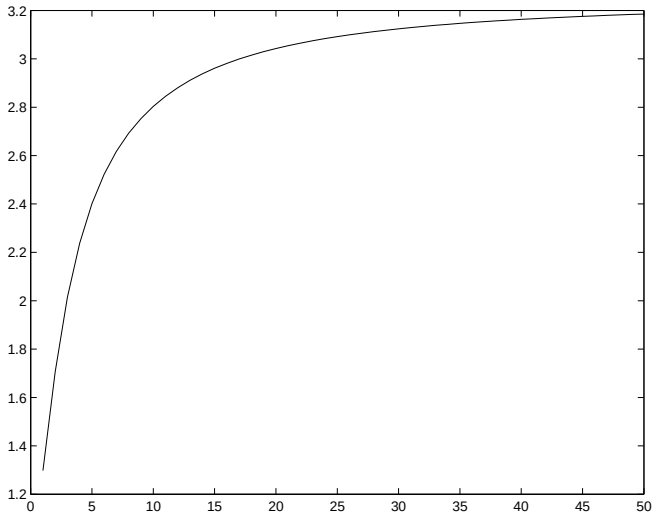



Figure 4.18: Mutual information during 50 EM iterations.

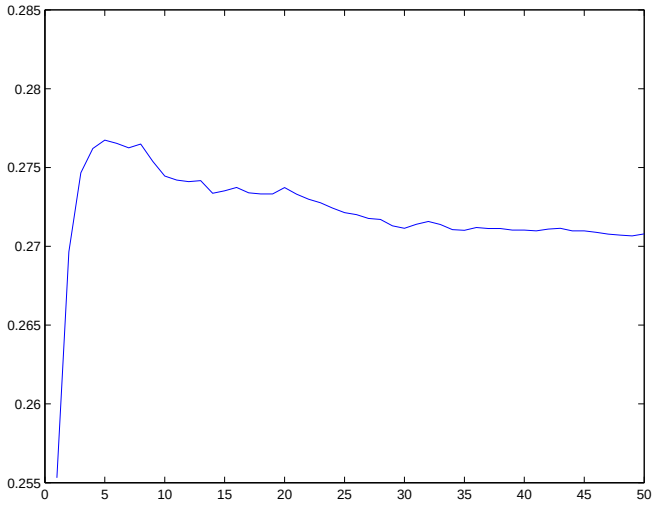



Figure 4.19: Prediction measure during 50 EM iterations.

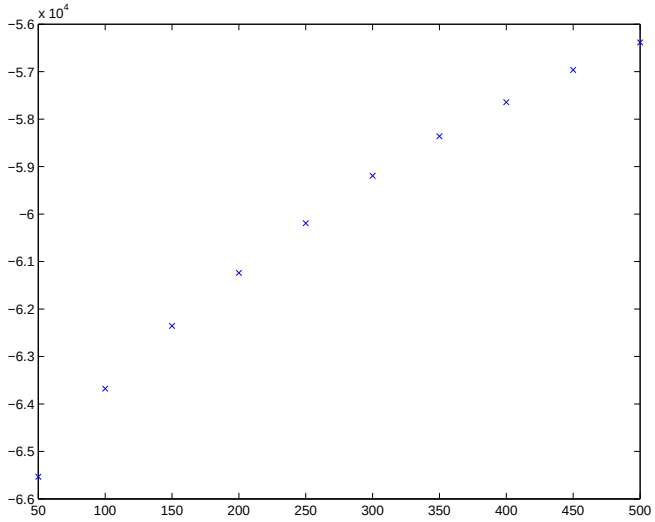



Figure 4.20: Log-likelihoods for different number of clusters for training set of the first experimental data set.

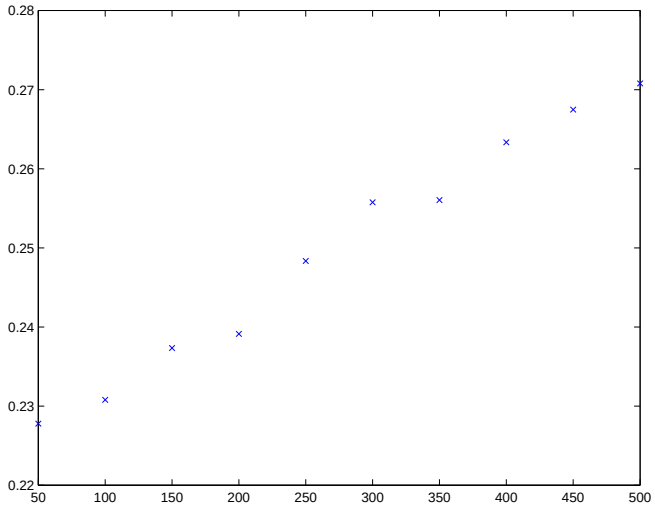



Figure 4.21: Word prediction measure for different number of clusters for the training set of the first experimental data set.

### 4.4.3 The selection of feature set

Feature selection is an important preliminary step for most of the computer vision tasks. Our system also highly depends on the selected features, since it affects the clustering phase and therefore the learning phase.

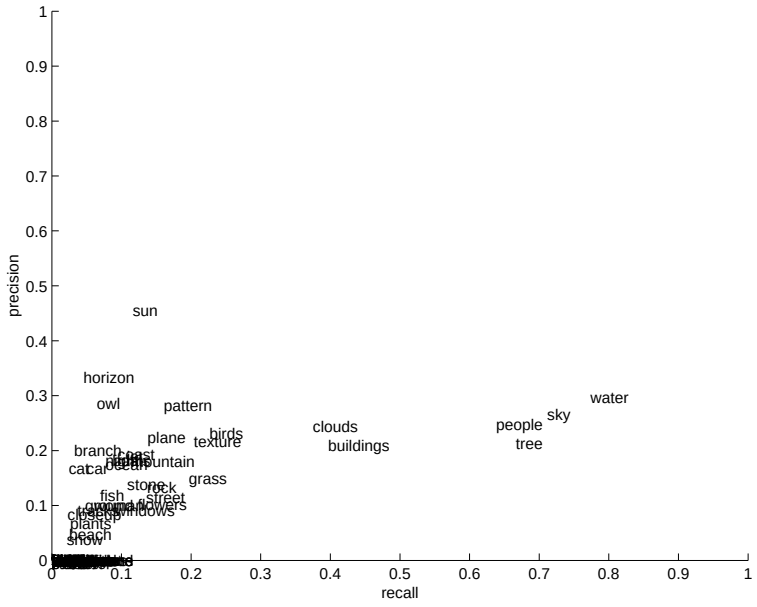
The performance of different feature sets are evaluated by using the word prediction measures as a function of the selected features. Since it is impractical to evaluate all combinations of features, we break them into subgroups. We evaluate the results for the following feature sets;

- all of the original features, mentioned in Section4.1.3, are used (the size of the feature set is 30),
- PCA is applied on the original feature set (the largest 11 eigenvalues are taken, since the number of nonzero eigenvalues is 11),
- only color and texture features are taken (the size of the feature set is 24),
- only color features are taken (the size of the feature set is 12),

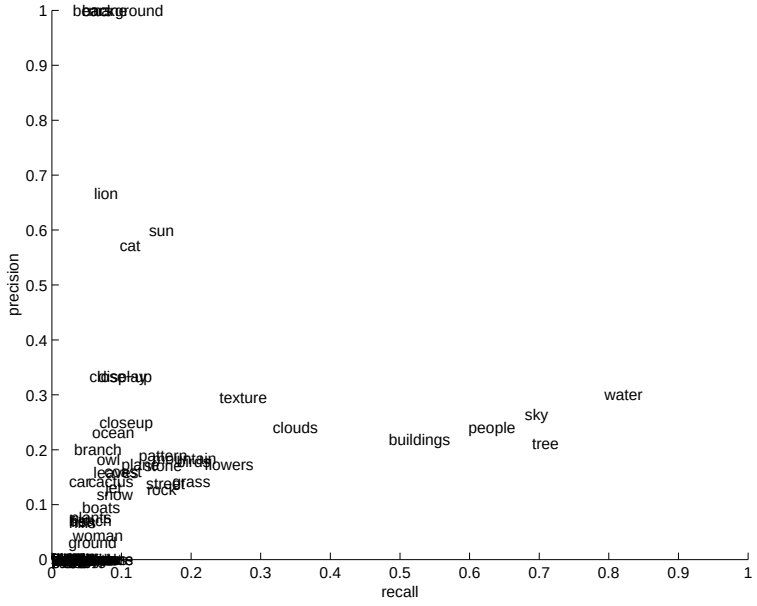
In Figure 4.22, the recall and precision values are used to compare the results as a function of words. The selected sets are also compared using KL divergence, normalized classification score, word prediction rate, and correspondence scores in Table 4.24. Not surprisingly, for the Corel data, color is the most important feature among the features that are used in the experiments.

Table 4.24: Results as a function of selected features. KL divergence (KL), normalized classification rate (NS), word prediction performance(PR) and correspondence sores, using the first label (corr-first) and using all the labels (corr-all) for comparison, are used for evaluating the results.

| | original | PCA | color+texture | color |
|---|---|---|---|---|
| KL | 5.2089 | 5.1967 | 5.1915 | 5.0018 |
| NS | 0.2012 | 0.2006 | 0.2008 | 0.2175 |
| PR | 0.2171 | 0.2164 | 0.2166 | 0.2330 |
| corr-first | 315 | 285 | 283 | 271 |
| corr-all | 444 | 422 | 408 | 457 |

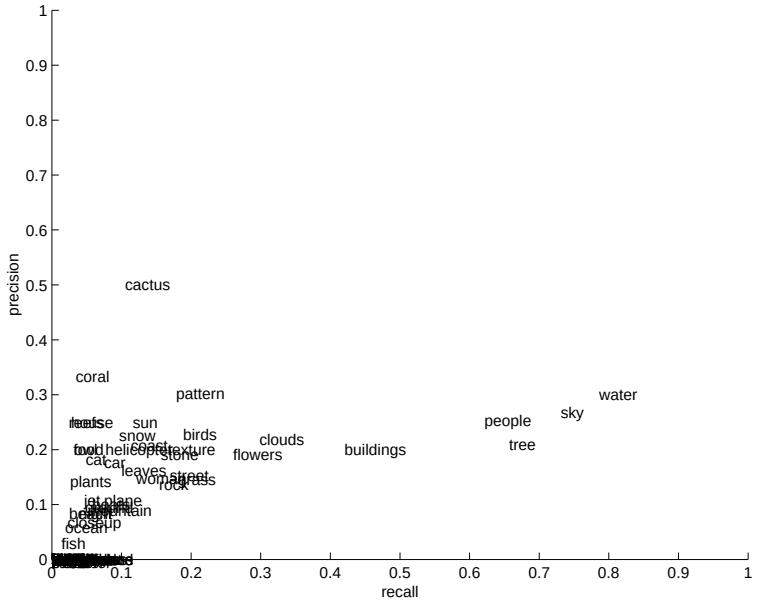



(a)

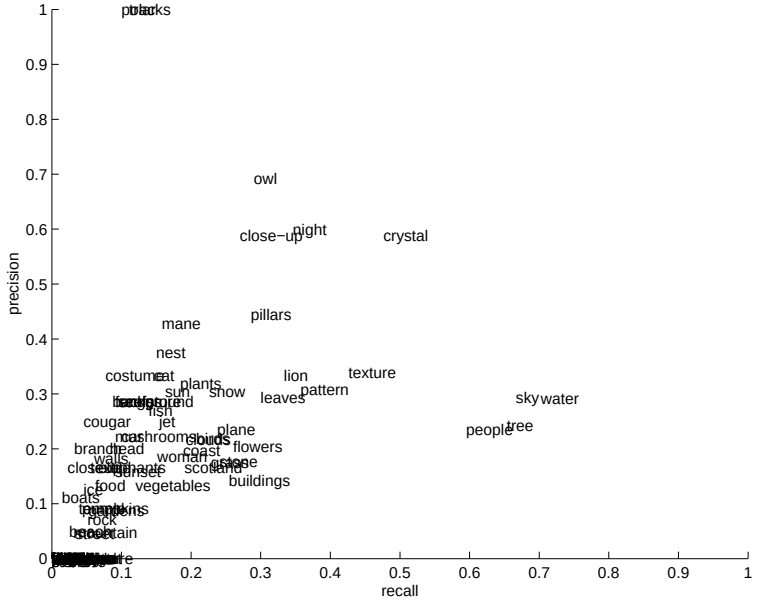



(b)

Figure 4.22: Recall and precision values as a function of selected features: (a) using all of the original features, (b) when PCA is applied on the original set.




(c)




(d)

Figure 4.22: Recall and precision values as a function of selected features (continued): (c) using only color and texture features, (d) using only color features (both RGB and lab).

## 4.5 Improving the system

As discussed before, the annotation words used in the Corel data set create some problems for the proposed approach. The occurrence frequencies of the words are in a large range, causing an unstable prediction rates between the common and rare words. There are some words which do not represent the visual properties, or representing a larger area than a region, thus it is not possible to learn such words. Also, there are some compound words or some words which always occur together in the annotations, making hard to distinguish them. In this section we present the following improvement strategies to solve mentioned problems:

- Refusing to predict,
- Retraining on refined vocabulary,
- Merging indistinguishable words.

### 4.5.1 Refusing to predict - NULL prediction

In the proposed method, for each blob in an image, the word with the highest probability is chosen as the predicted word. For some blobs, however, even the probability of the maximum word is not high to have a certain prediction.

Table 4.25 shows the relationship between the prediction probabilities and the success of the prediction. For each of 500 blobs (in the sorted order according to the prediction probabilities of the predicted words), the probability for the predicted word, and the prediction rate, which is computed as the number of correct predictions over number of occurrence of the blob, are given. The highest prediction probability for the predicted words is 0.59 and the lowest value is 0.19. As we expect, there is a relationship between the prediction probability and success of the prediction. Therefore, if we remove the words, that have low prediction probabilities, and keep only the words with high prediction probabilities, the performance of the system should improve.

Table 4.25: Highest probability words for each blob (in the sorted order according to the prediction probability): (i) Probability of the predicted word (prob), (ii) predicted word (word), (iii) prediction rate (rate). The values indicate the relationship between the prediction probability and success of the prediction.

| prob | word | rate | prob | word | rate | prob | word | rate | prob | word | rate |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 0.59 | nest | 1.00 | 0.54 | water | 0.49 | 0.52 | people | 0.41 | 0.46 | people | 0.51 |
| 0.44 | water | 0.44 | 0.44 | water | 0.46 | 0.42 | water | 0.39 | 0.41 | tree | 0.28 |
| 0.41 | people | 0.34 | 0.41 | water | 0.49 | 0.41 | sky | 0.44 | 0.41 | people | 0.32 |
| 0.40 | people | 0.38 | 0.40 | water | 0.45 | 0.40 | people | 0.32 | 0.40 | water | 0.34 |
| 0.40 | street | 0.37 | 0.39 | windows | 0.35 | 0.39 | people | 0.59 | 0.38 | street | 0.44 |
| 0.38 | sky | 0.47 | 0.38 | sky | 0.29 | 0.37 | boats | 0.29 | 0.37 | jet | 0.40 |
| 0.36 | water | 0.56 | 0.36 | people | 0.32 | 0.35 | people | 0.38 | 0.35 | sky | 0.38 |
| 0.35 | water | 0.46 | 0.35 | buildings | 0.32 | 0.34 | sky | 0.28 | 0.34 | water | 0.40 |
| 0.34 | tree | 0.36 | 0.33 | people | 0.28 | 0.33 | water | 0.43 | 0.33 | tree | 0.31 |
| 0.33 | street | 0.28 | 0.33 | buildings | 0.34 | 0.33 | water | 0.30 | 0.33 | buildings | 0.27 |
| 0.33 | water | 0.42 | 0.32 | water | 0.40 | 0.32 | tree | 0.30 | 0.32 | sky | 0.39 |
| 0.32 | people | 0.30 | 0.32 | people | 0.26 | 0.32 | people | 0.27 | 0.32 | sky | 0.38 |
| 0.32 | tree | 0.29 | 0.32 | sun | 0.39 | 0.32 | sky | 0.34 | 0.32 | stone | 0.21 |
| 0.32 | people | 0.30 | 0.31 | water | 0.41 | 0.31 | people | 0.36 | 0.31 | people | 0.29 |
| 0.31 | people | 0.34 | 0.31 | water | 0.47 | 0.31 | pattern | 0.53 | 0.31 | people | 0.35 |
| 0.31 | street | 0.31 | 0.30 | people | 0.32 | 0.30 | people | 0.37 | 0.30 | water | 0.39 |
| 0.30 | plane | 0.20 | 0.30 | water | 0.31 | 0.30 | sky | 0.26 | 0.30 | people | 0.36 |
| 0.30 | people | 0.23 | 0.30 | water | 0.38 | 0.29 | people | 0.19 | 0.29 | tracks | 0.18 |
| 0.29 | people | 0.22 | 0.29 | water | 0.28 | 0.28 | tree | 0.30 | 0.28 | sky | 0.30 |
| 0.28 | sky | 0.26 | 0.28 | people | 0.28 | 0.28 | boats | 0.26 | 0.28 | tree | 0.29 |
| 0.28 | water | 0.37 | 0.28 | sky | 0.26 | 0.28 | turn | 0.29 | 0.28 | people | 0.25 |
| 0.28 | tree | 0.28 | 0.28 | doors | 0.33 | 0.28 | people | 0.27 | 0.28 | tree | 0.26 |
| 0.27 | water | 0.23 | 0.27 | pattern | 0.20 | 0.27 | people | 0.37 | 0.27 | sky | 0.48 |
| 0.27 | clouds | 0.28 | 0.27 | flowers | 0.21 | 0.27 | birds | 0.22 | 0.27 | fish | 0.12 |
| 0.27 | water | 0.35 | 0.27 | background | 0.86 | 0.27 | water | 0.42 | 0.27 | buildings | 0.26 |
| 0.27 | water | 0.54 | 0.26 | tree | 0.27 | 0.26 | street | 0.16 | 0.26 | people | 0.29 |
| 0.26 | people | 0.29 | 0.26 | birds | 0.15 | 0.26 | sky | 0.34 | 0.26 | sky | 0.44 |
| 0.26 | people | 0.26 | 0.26 | house | 0.19 | 0.26 | sky | 0.28 | 0.26 | jet | 0.17 |
| 0.26 | sun | 0.40 | 0.26 | beach | 0.17 | 0.26 | people | 0.32 | 0.25 | windows | 0.44 |
| 0.25 | water | 0.40 | 0.25 | tree | 0.29 | 0.25 | buildings | 0.20 | 0.25 | street | 0.22 |
| 0.25 | sky | 0.32 | 0.25 | water | 0.41 | 0.25 | woman | 0.14 | 0.25 | field | 0.11 |
| 0.25 | sky | 0.28 | 0.25 | tree | 0.26 | 0.25 | woods | 0.12 | 0.25 | tree | 0.24 |
| 0.25 | tree | 0.29 | 0.25 | birds | 0.26 | 0.25 | ground | 0.21 | 0.25 | tree | 0.29 |
| 0.25 | textile | 0.31 | 0.24 | people | 0.28 | 0.24 | closeup | 0.13 | 0.24 | sky | 0.29 |
| 0.24 | tree | 0.28 | 0.24 | people | 0.38 | 0.24 | leaf | 0.16 | 0.24 | ground | 0.13 |
| 0.24 | plants | 0.16 | 0.24 | people | 0.24 | 0.24 | sky | 0.27 | 0.24 | water | 0.41 |
| 0.24 | grass | 0.17 | 0.24 | water | 0.40 | 0.24 | windows | 0.27 | 0.24 | street | 0.23 |
| 0.24 | mountain | 0.18 | 0.24 | tree | 0.24 | 0.24 | tree | 0.26 | 0.24 | zebra | 0.50 |
| 0.24 | tree | 0.29 | 0.24 | rock | 0.16 | 0.24 | buildings | 0.19 | 0.24 | tree | 0.24 |
| 0.24 | mountain | 0.20 | 0.24 | sky | 0.36 | 0.23 | vegetables | 0.27 | 0.23 | buildings | 0.18 |
| 0.23 | water | 0.34 | 0.23 | tree | 0.28 | 0.23 | pillars | 0.11 | 0.23 | people | 0.46 |
| 0.23 | sky | 0.31 | 0.23 | buildings | 0.16 | 0.23 | water | 0.40 | 0.23 | buildings | 0.19 |
| 0.23 | tree | 0.25 | 0.23 | flowers | 0.14 | 0.23 | sky | 0.36 | 0.23 | water | 0.29 |
| 0.23 | water | 0.30 | 0.22 | water | 0.33 | 0.22 | tree | 0.30 | 0.22 | buildings | 0.20 |
| 0.22 | texture | 0.24 | 0.22 | jet | 0.19 | 0.22 | night | 0.13 | 0.22 | sky | 0.24 |
| 0.22 | sky | 0.22 | 0.22 | flowers | 0.17 | 0.22 | rock | 0.14 | 0.22 | rock | 0.16 |
| 0.22 | pillars | 0.19 | 0.22 | people | 0.23 | 0.22 | people | 0.23 | 0.22 | buildings | 0.29 |
| 0.22 | tree | 0.27 | 0.22 | windows | 0.16 | 0.22 | buildings | 0.28 | 0.22 | tree | 0.31 |
| 0.22 | sky | 0.33 | 0.22 | helicopter | 0.15 | 0.22 | street | 0.29 | 0.21 | sky | 0.20 |
| 0.21 | ground | 0.13 | 0.21 | clouds | 0.18 | 0.21 | grass | 0.17 | 0.21 | water | 0.26 |
| 0.21 | boats | 0.11 | 0.21 | water | 0.30 | 0.21 | sky | 0.30 | 0.21 | rock | 0.15 |
| 0.21 | clouds | 0.16 | 0.21 | street | 0.26 | 0.21 | texture | 0.21 | 0.21 | people | 0.31 |
| 0.21 | grass | 0.14 | 0.21 | tree | 0.21 | 0.21 | sky | 0.31 | 0.21 | plane | 0.17 |
| 0.21 | flowers | 0.18 | 0.21 | water | 0.36 | 0.21 | sky | 0.20 | 0.21 | buildings | 0.26 |
| 0.21 | water | 0.28 | 0.21 | water | 0.35 | 0.21 | tree | 0.23 | 0.20 | tree | 0.21 |
| 0.20 | tree | 0.23 | 0.20 | buildings | 0.26 | 0.20 | woman | 0.12 | 0.20 | people | 0.22 |
| 0.20 | lion | 0.13 | 0.20 | waves | 0.12 | 0.20 | buildings | 0.25 | 0.20 | temple | 0.13 |
| 0.20 | insect | 0.11 | 0.20 | ocean | 0.11 | 0.20 | horizon | 0.17 | 0.20 | people | 0.26 |
| 0.20 | sky | 0.24 | 0.20 | tree | 0.25 | 0.20 | temple | 0.33 | 0.20 | stone | 0.11 |

Table 4.25: Continued.

| prob | word | rate | prob | word | rate | prob | word | rate | prob | word | rate |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 0.20 | tree | 0.27 | 0.20 | boats | 0.11 | 0.20 | tree | 0.22 | 0.20 | smoke | 0.12 |
| 0.20 | buildings | 0.21 | 0.20 | temple | 0.14 | 0.20 | walls | 0.11 | 0.20 | coast | 0.10 |
| 0.20 | pillars | 0.13 | 0.20 | sky | 0.19 | 0.20 | vegetables | 0.13 | 0.20 | snow | 0.11 |
| 0.20 | water | 0.63 | 0.20 | close-up | 0.16 | 0.20 | people | 0.23 | 0.20 | columns | 0.09 |
| 0.19 | street | 0.18 | 0.19 | gardens | 0.10 | 0.19 | plane | 0.18 | 0.19 | water | 0.39 |
| 0.19 | street | 0.27 | 0.19 | beach | 0.11 | 0.19 | turn | 0.20 | 0.19 | buildings | 0.23 |
| 0.19 | people | 0.18 | 0.19 | people | 0.28 | 0.19 | water | 0.27 | 0.19 | jet | 0.11 |
| 0.19 | tree | 0.22 | 0.19 | buildings | 0.23 | 0.19 | coast | 0.11 | 0.19 | insect | 0.11 |
| 0.19 | car | 0.17 | 0.19 | water | 0.25 | 0.19 | tree | 0.24 | 0.19 | water | 0.29 |
| 0.19 | water | 0.17 | 0.19 | hats | 0.09 | 0.19 | people | 0.28 | 0.19 | rock | 0.14 |
| 0.19 | tree | 0.25 | 0.19 | fish | 0.12 | 0.19 | buildings | 0.19 | 0.19 | entrance | 0.12 |
| 0.19 | mountain | 0.16 | 0.19 | people | 0.27 | 0.19 | water | 0.20 | 0.19 | helicopter | 0.09 |
| 0.18 | water | 0.37 | 0.18 | snow | 0.12 | 0.18 | tree | 0.28 | 0.18 | water | 0.29 |
| 0.18 | tree | 0.26 | 0.18 | birds | 0.15 | 0.18 | water | 0.20 | 0.18 | buildings | 0.20 |
| 0.18 | sky | 0.25 | 0.18 | tracks | 0.12 | 0.18 | street | 0.16 | 0.18 | ground | 0.14 |
| 0.18 | birds | 0.14 | 0.18 | sky | 0.25 | 0.18 | plane | 0.18 | 0.18 | birds | 0.29 |
| 0.18 | woman | 0.08 | 0.18 | people | 0.42 | 0.18 | sky | 0.25 | 0.18 | people | 0.22 |
| 0.18 | walls | 0.11 | 0.18 | texture | 0.13 | 0.18 | insect | 0.14 | 0.18 | people | 0.27 |
| 0.18 | buildings | 0.22 | 0.18 | walls | 0.11 | 0.18 | car | 0.17 | 0.18 | tree | 0.30 |
| 0.18 | tree | 0.28 | 0.18 | night | 0.11 | 0.18 | water | 0.45 | 0.18 | sky | 0.28 |
| 0.18 | sky | 0.33 | 0.18 | gardens | 0.17 | 0.18 | texture | 0.19 | 0.18 | plants | 0.09 |
| 0.17 | buildings | 0.21 | 0.17 | clouds | 0.20 | 0.17 | water | 0.36 | 0.17 | buildings | 0.30 |
| 0.17 | buildings | 0.24 | 0.17 | temple | 0.06 | 0.17 | pattern | 0.19 | 0.17 | texture | 0.23 |
| 0.17 | flowers | 0.16 | 0.17 | grass | 0.14 | 0.17 | clouds | 0.13 | 0.17 | woman | 0.12 |
| 0.17 | stone | 0.14 | 0.17 | head | 0.10 | 0.17 | flowers | 0.15 | 0.17 | texture | 0.14 |
| 0.17 | birds | 0.19 | 0.17 | rock | 0.14 | 0.17 | water | 0.32 | 0.17 | walls | 0.09 |
| 0.17 | sky | 0.20 | 0.17 | buildings | 0.18 | 0.17 | clouds | 0.14 | 0.17 | water | 0.32 |
| 0.17 | water | 0.24 | 0.17 | water | 0.27 | 0.17 | birds | 0.19 | 0.17 | owl | 0.11 |
| 0.17 | background | 0.15 | 0.17 | people | 0.19 | 0.17 | jet | 0.09 | 0.17 | stone | 0.10 |
| 0.17 | jet | 0.17 | 0.17 | buildings | 0.18 | 0.17 | people | 0.21 | 0.16 | sky | 0.29 |
| 0.16 | birds | 0.09 | 0.16 | street | 0.10 | 0.16 | mountain | 0.17 | 0.16 | fish | 0.09 |
| 0.16 | texture | 0.14 | 0.16 | architecture | 0.08 | 0.16 | wildlife | 0.07 | 0.16 | stone | 0.14 |
| 0.16 | birds | 0.13 | 0.16 | tree | 0.25 | 0.16 | clouds | 0.13 | 0.16 | tree | 0.27 |
| 0.16 | clouds | 0.18 | 0.16 | textile | 0.11 | 0.16 | clouds | 0.15 | 0.16 | grass | 0.12 |
| 0.16 | cliff | 0.17 | 0.16 | hawk | 0.14 | 0.16 | sea | 0.14 | 0.16 | woman | 0.10 |
| 0.16 | car | 0.09 | 0.16 | pattern | 0.11 | 0.16 | tree | 0.40 | 0.16 | boats | 0.13 |
| 0.16 | statues | 0.07 | 0.16 | head | 0.09 | 0.16 | clouds | 0.14 | 0.16 | water | 0.44 |
| 0.16 | people | 0.23 | 0.16 | leaves | 0.12 | 0.16 | vegetables | 0.13 | 0.16 | stone | 0.13 |
| 0.16 | branch | 0.07 | 0.16 | pattern | 0.13 | 0.16 | forest | 0.08 | 0.16 | church | 0.08 |
| 0.16 | tree | 0.20 | 0.15 | art | 0.09 | 0.15 | rock | 0.15 | 0.15 | coast | 0.13 |
| 0.15 | street | 0.18 | 0.15 | sky | 0.25 | 0.15 | water | 0.28 | 0.15 | sky | 0.23 |
| 0.15 | buildings | 0.16 | 0.15 | insect | 0.13 | 0.15 | tree | 0.26 | 0.15 | crystal | 0.15 |
| 0.15 | grass | 0.13 | 0.15 | grass | 0.12 | 0.15 | people | 0.21 | 0.15 | people | 0.26 |
| 0.15 | hawk | 0.07 | 0.15 | rock | 0.14 | 0.15 | rock | 0.10 | 0.15 | reefs | 0.09 |
| 0.15 | snow | 0.15 | 0.15 | gun | 0.07 | 0.15 | birds | 0.12 | 0.15 | birds | 0.13 |
| 0.15 | hills | 0.11 | 0.15 | flowers | 0.12 | 0.15 | sky | 0.18 | 0.15 | grass | 0.12 |
| 0.15 | texture | 0.13 | 0.15 | coral | 0.09 | 0.15 | food | 0.09 | 0.15 | closeup | 0.10 |
| 0.15 | texture | 0.13 | 0.15 | fish | 0.10 | 0.15 | rock | 0.13 | 0.15 | tracks | 0.14 |
| 0.15 | grass | 0.14 | 0.15 | water | 0.31 | 0.15 | water | 0.33 | 0.14 | tree | 0.23 |
| 0.14 | tree | 0.23 | 0.14 | gun | 0.09 | 0.14 | sky | 0.28 | 0.14 | vegetables | 0.13 |
| 0.14 | f-16 | 0.06 | 0.14 | ground | 0.09 | 0.14 | closeup | 0.09 | 0.14 | people | 0.23 |
| 0.14 | mountain | 0.15 | 0.14 | textile | 0.10 | 0.14 | sky | 0.28 | 0.14 | insect | 0.13 |
| 0.14 | stone | 0.15 | 0.14 | walls | 0.07 | 0.14 | birds | 0.12 | 0.14 | street | 0.15 |
| 0.14 | water | 0.32 | 0.14 | stone | 0.11 | 0.14 | sky | 0.20 | 0.14 | nest | 0.14 |
| 0.14 | tree | 0.25 | 0.14 | shrine | 0.06 | 0.14 | seals | 0.06 | 0.13 | coast | 0.13 |
| 0.13 | sky | 0.14 | 0.13 | temple | 0.10 | 0.13 | skis | 0.08 | 0.13 | water | 0.19 |
| 0.13 | sky | 0.18 | 0.13 | display | 0.07 | 0.13 | woman | 0.08 | 0.13 | woman | 0.13 |
| 0.13 | people | 0.21 | 0.13 | jet | 0.11 | 0.13 | fish | 0.10 | 0.13 | birds | 0.17 |
| 0.13 | harbor | 0.10 | 0.13 | face | 0.05 | 0.13 | snake | 0.07 | 0.13 | clouds | 0.22 |
| 0.13 | beach | 0.09 | 0.13 | birds | 0.11 | 0.13 | plane | 0.10 | 0.13 | boats | 0.06 |
| 0.13 | food | 0.07 | 0.12 | sky | 0.14 | 0.12 | ground | 0.17 | 0.12 | fish | 0.07 |
| 0.12 | buildings | 0.11 | 0.12 | vegetables | 0.10 | 0.12 | sky | 0.21 | 0.12 | texture | 0.13 |
| 0.12 | water | 0.20 | 0.12 | entrance | 0.08 | 0.12 | rock | 0.09 | 0.12 | tree | 0.19 |
| 0.11 | birds | 0.09 | 0.11 | tree | 0.23 | 0.11 | mushrooms | 0.06 | 0.11 | birds | 0.10 |
| 0.11 | mountain | 0.10 | 0.11 | buildings | 0.23 | 0.11 | cactus | 0.08 | 0.11 | stone | 0.10 |
| 0.10 | hills | 0.09 | 0.10 | plants | 0.10 | 0.10 | flowers | 0.08 | 0.10 | ground | 0.08 |

In order to analyze this idea, we set a threshold and refuse to predict the words that have lower probabilities than the threshold. For a blob, if the predicted word has a lower probability than the threshold, we assume that NULL word is predicted.

In Figure 4.22, the result of NULL prediction when we set the threshold to 0.2 are shown. As it is seen from the images, when we allow the system to predict words for all the blobs, most of the blobs predict unreliable words. However, when the system refuse to predict for the blobs which have low prediction probabilities, then the unrelated words are removed, and only the "good" words remain in the predictions.

The recall and precision values are also a good indicator for understanding the effect of the threshold. In order to obtain the predicted words for an image, the union of the words predicted by the individual blobs are taken. Then, recall and precision values are computed using these annotations. Figures 4.23, indicates the change in the recall and precision values as a function of increasing null threshold.

As the figures indicate, with the increasing null threshold, less number of words are predicted, since for most of the words prediction probabilities are lower than the specified threshold. Recall values decrease, since the number of predictions decrease, but precision values increase for the remaining words, since the remaining predictions are more reliable.

Table 4.26 shows the change in the number of blobs that predict words as a function of increasing threshold. With the increasing threshold, less blobs predict words, and after 0.59 (which the highest prediction probability) none of the blobs can predict words.

Table 4.26: For increasing thresholds (0.00 to 0.60), number of blobs (num blobs), and number of regions corresponding to those blobs remained in the training (training - num regions ) and the standard test (test - num regions) sets, and number of words predicted by those blobs whose predicted word's probability is larger than the threshold (num words).

| threshold | num blobs | training - num regions | test - num regions | num words |
|---|---|---|---|---|
| 0.00 | 500 | 40357 | 13590 | 86 |
| 0.05 | 500 | 40357 | 13590 | 86 |
| 0.10 | 499 | 40199 | 13546 | 86 |
| 0.15 | 411 | 33312 | 11266 | 71 |
| 0.20 | 228 | 19168 | 6444 | 44 |
| 0.25 | 122 | 10133 | 3401 | 24 |
| 0.30 | 63 | 5410 | 1816 | 13 |
| 0.35 | 28 | 2222 | 788 | 9 |
| 0.40 | 13 | 1161 | 411 | 5 |
| 0.45 | 4 | 246 | 77 | 3 |
| 0.50 | 3 | 187 | 61 | 3 |
| 0.55 | 1 | 2 | 1 | 1 |
| 0.60 | 0 | 0 | 0 | 0 |

The effect of null threshold is experimented by changing it between 0 and 0.5. In Figure 4.24, the effect of null threshold on the recall and precision values of some selected "good" words are shown. A word is regarded as "good" if it has high recall and precision values. As the figure indicates increasing the null threshold, the recall decreases. The increase in the precision values shows that the correct prediction rate is increasing. When the null threshold is increased sufficiently, some words cannot be predicted at all, since their highest prediction rate is lower than the null threshold. Therefore, both recall and precision values become 0 after some threshold. The results are very similar for both training and the standard test sets. Figure 4.25 shows the effect of null threshold on the recall and precision values of some selected "bad" words, which have low recall and precision values. As the figure indicates, the values immediately decreases to 0. Note that scales are different.

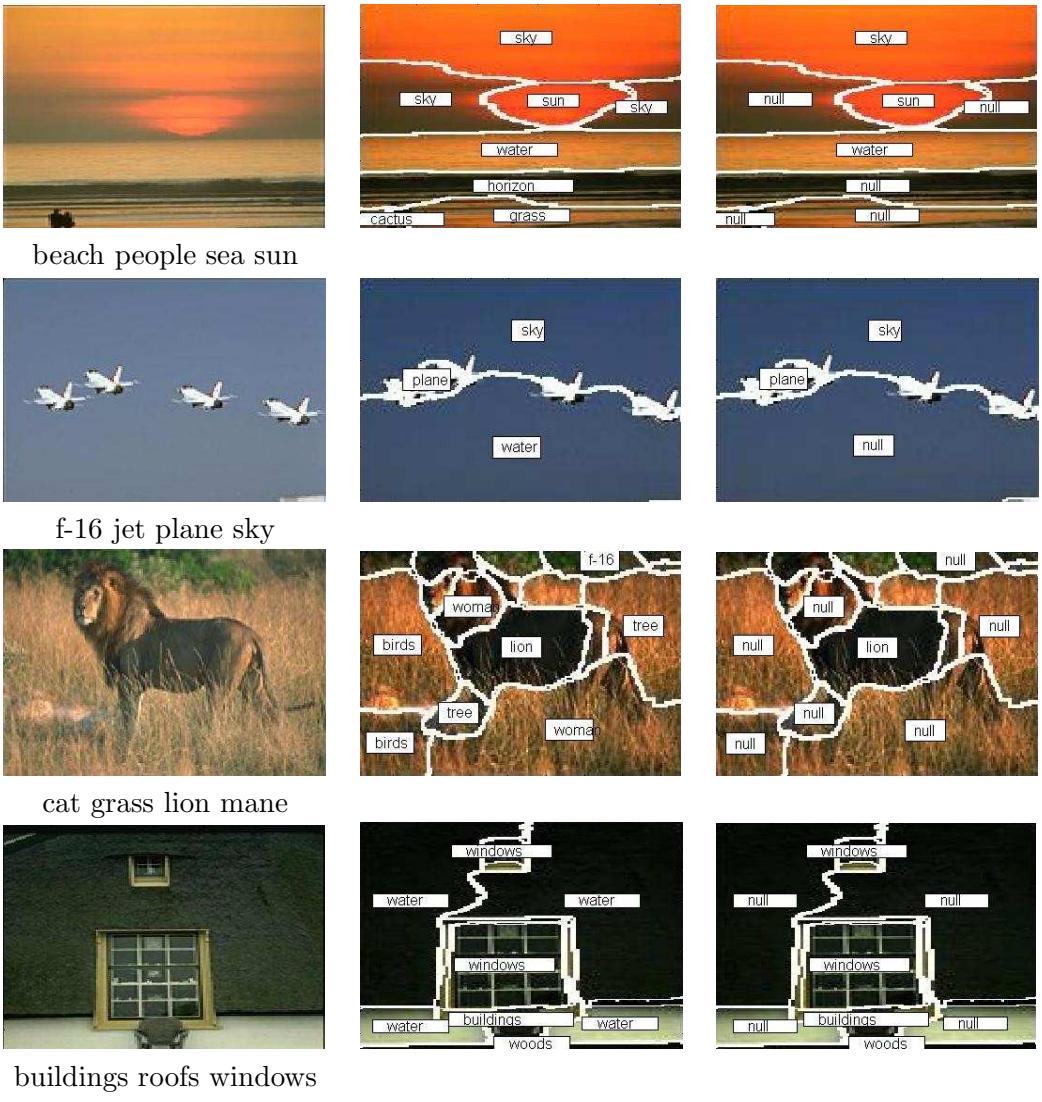



Figure 4.22: Results of NULL prediction on the training set. First column : original image with annotated keywords, second column : results without null prediction and third column : results for null threshold 0.2. As it is seen from the images, when the system refuse to predict for the blobs which have low prediction probabilities, then the unreliable words are removed, and only the "good" words remain in the predictions. Examples: the word `sun, sky` and `water` in the first image, `plane` and `sky` in the second image, only `lion` in the third image, and `windows, buildings` and `woods` in the last image.

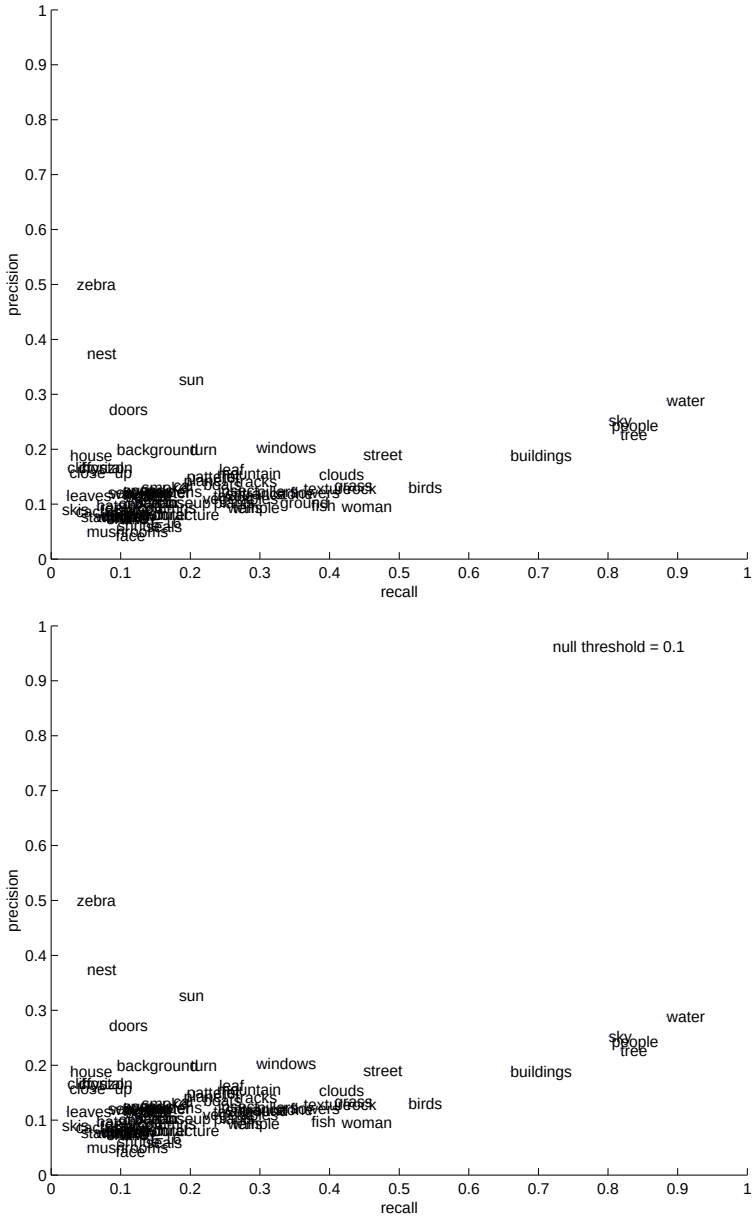



Figure 4.23: Recall versus precision values for the training set for the null thresholds 0 and 0.1

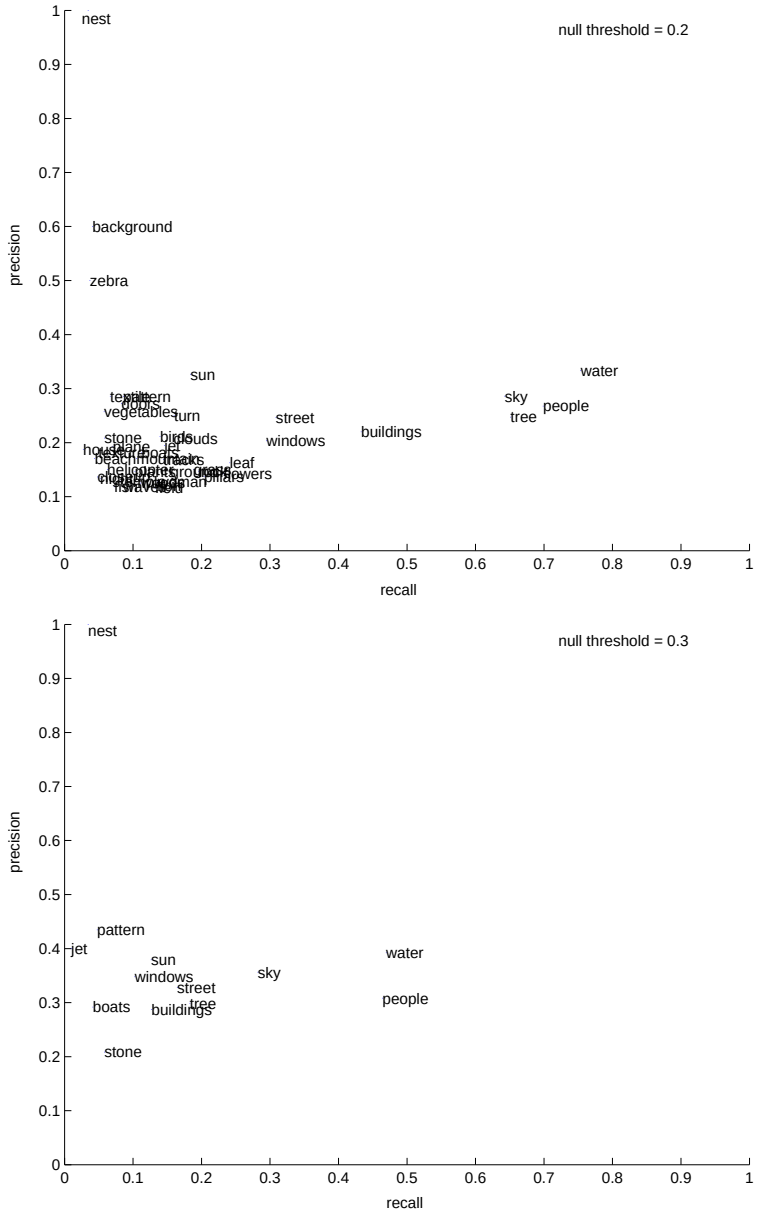



Figure 4.24: Recall versus precision values for the training set for the null thresholds 0.2 and 0.3.

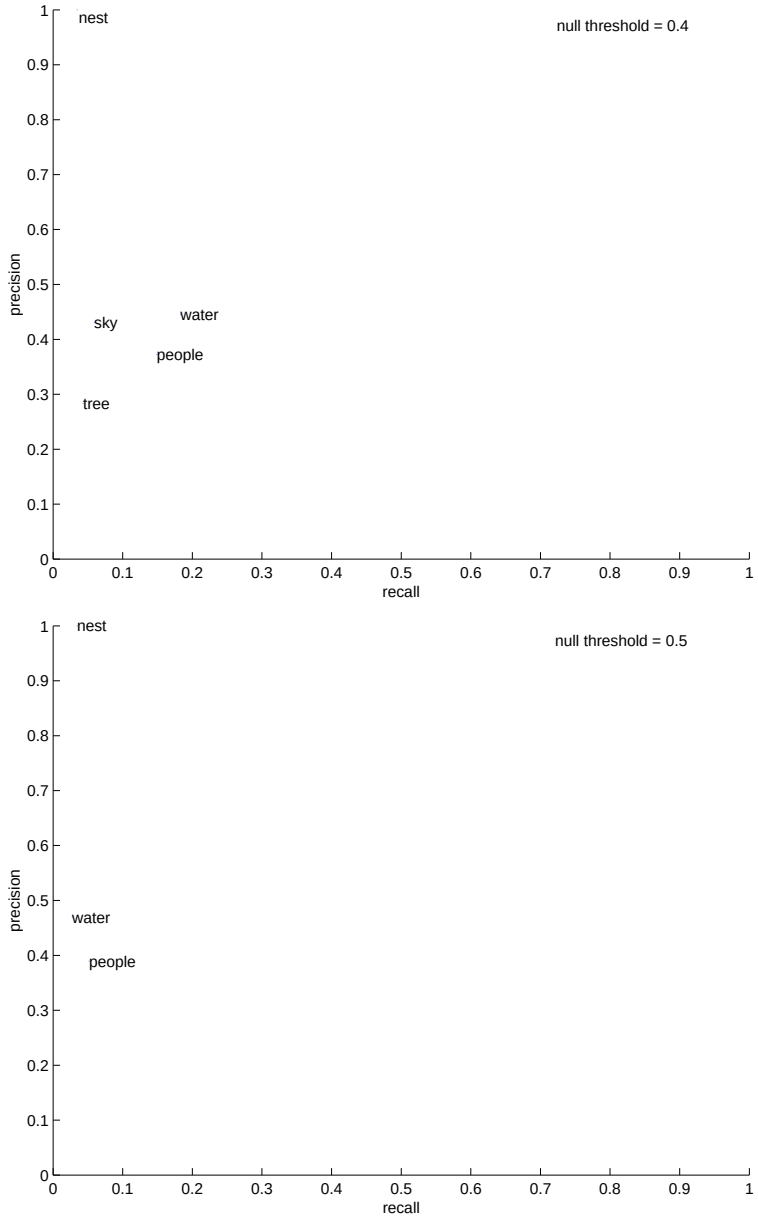



Figure 4.24: Recall versus precision values for the training set for the null thresholds 0.4 and 0.5.

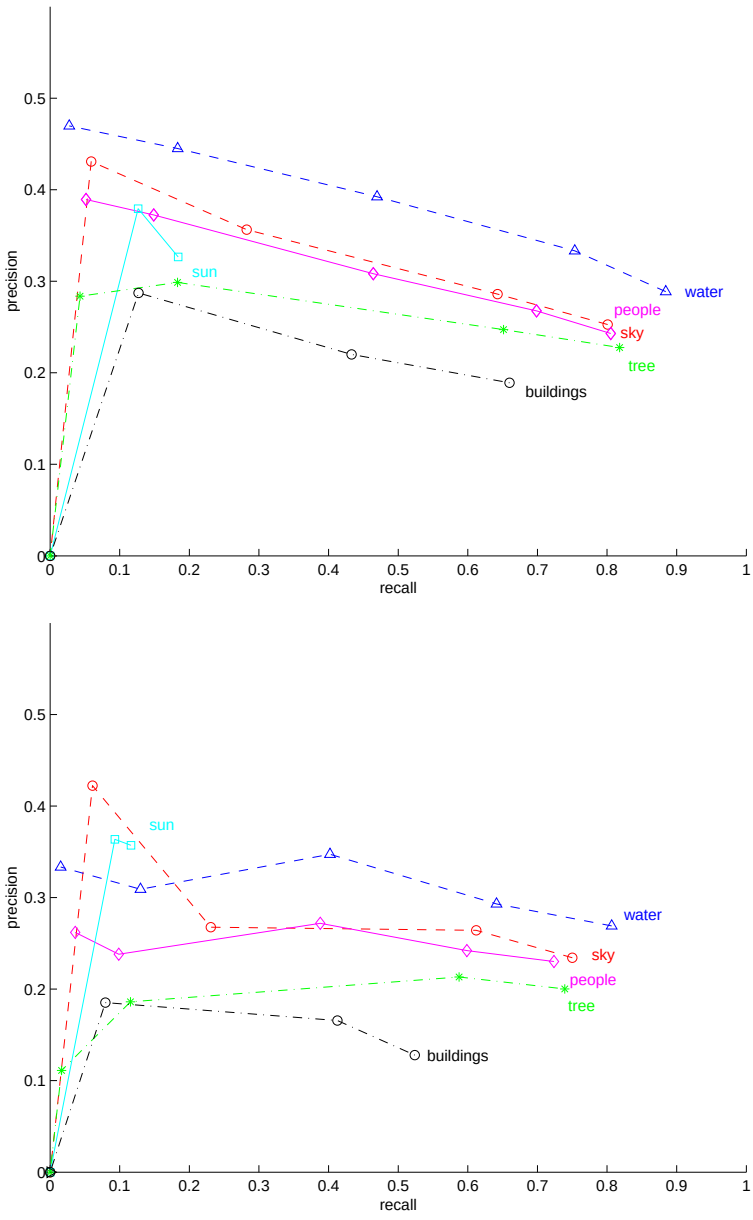



Figure 4.24: Recall versus precision for selected good words with increasing null threshold values (0-0.5) : The top line shows the results for training and bottom line shows the results for the standard test sets. The results are very similar both for the training and test sets. Recall values decrease by increasing null threshold, but usually precision increase since the correct prediction rate increase. After a threshold value, all precision and recall may go to 0 since we cannot predict the words anymore.

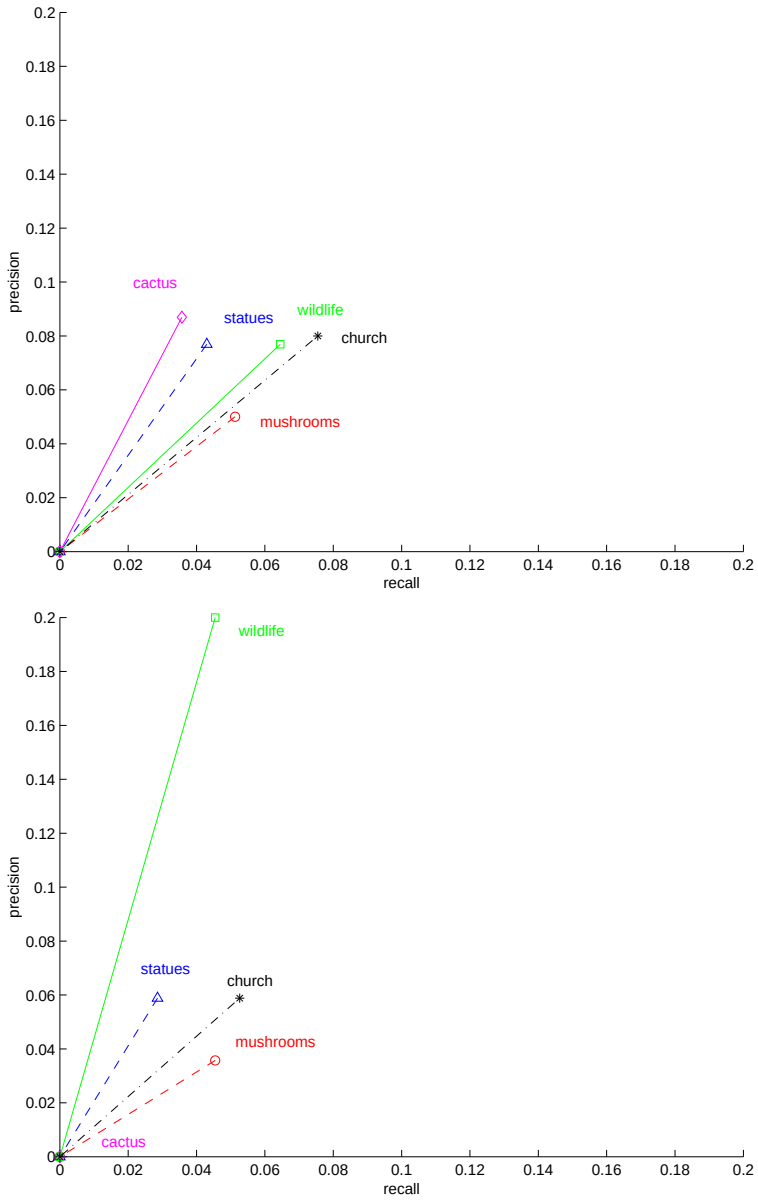



Figure 4.25: Recall versus precision for selected bad words with increasing null threshold values (0-0.5) : The top line shows the results for training and bottom line shows the results for the standard test set. The results are very similar both for the training and test sets. Since the words have very low recall and precision values, after setting the null threshold to 0.1 none of these "bad" words can be predicted. Note that scales are different than the scales for the good words.

### 4.5.2 Effect of retraining on refined vocabulary

The vocabulary is refined by choosing a threshold and allowing only the words which have higher prediction probabilities than the threshold to remain in the vocabulary. Then, the system is retrained on the refined vocabulary.

Table 4.27 show the number of words remained in the refined vocabularies: originally the number of words in the vocabulary was 153. When the words which have nonzero prediction probabilities are chosen the number of words in the vocabulary decreases to 86. When the null threshold is decreased further the number of words that remains in the vocabulary decreases.

Table 4.28 show the prediction probabilities for the words as we increase the threshold and refine the vocabulary. As it is observed, we predict the words with higher probabilities.

Retraining on a refined vocabulary increases the performance of the system. Table 4.29 compares the recall and precision values as a function of null threshold that is used for refining the vocabulary.

Table 4.27: Effect of retraining with refined vocabulary on the training set. In the original data there are 153 words. Vocabulary is refined by setting the threshold to 0.0, 0.1, 0.2 and 0.3, and choosing the words whose prediction probability is larger than the threshold. Performance is compared using KL divergence (KL), normalized classification score (NS) and word prediction measure (PR). The performance increases with the increasing null threshold.

| | num words | KL | NS | PR |
|---|---|---|---|---|
| original | 153 | 3.5602 | 0.2560 | 0.2708 |
| $> 0.0$ | 86 | 3.3132 | 0.2820 | 0.2936 |
| $> 0.1$ | 80 | 3.2878 | 0.2853 | 0.2966 |
| $> 0.2$ | 65 | 3.1968 | 0.2950 | 0.3054 |
| $> 0.3$ | 41 | 2.9685 | 0.3235 | 0.3320 |

Table 4.28: Word prediction probabilities after training on refined vocabulary. Since, the system learns the remaining words better than the original vocabulary which includes many words that are noisy, the prediction probabilities are higher meaning that the predictions are more reliable.

| word | org | $> 0.0$ | $> 0.1$ | $> 0.2$ | $> 0.3$ |
|---|---|---|---|---|---|
| architecture | 0.163 | 0.215 | 0.219 | 0.227 | 0.000 |
| art | 0.155 | 0.185 | 0.189 | 0.000 | 0.000 |
| background | 0.269 | 0.207 | 0.207 | 0.267 | 0.000 |
| beach | 0.256 | 0.329 | 0.330 | 0.343 | 0.379 |
| birds | 0.273 | 0.309 | 0.318 | 0.352 | 0.453 |
| boats | 0.372 | 0.497 | 0.498 | 0.612 | 0.715 |
| branch | 0.156 | 0.195 | 0.318 | 0.000 | 0.000 |
| buildings | 0.345 | 0.447 | 0.493 | 0.504 | 0.607 |
| cactus | 0.108 | 0.165 | 0.166 | 0.000 | 0.000 |
| car | 0.190 | 0.251 | 0.294 | 0.317 | 0.382 |
| church | 0.155 | 0.180 | 0.191 | 0.000 | 0.000 |
| cliff | 0.161 | 0.000 | 0.000 | 0.000 | 0.000 |
| close-up | 0.195 | 0.237 | 0.241 | 0.266 | 0.000 |
| closeup | 0.244 | 0.293 | 0.296 | 0.335 | 0.433 |
| clouds | 0.273 | 0.351 | 0.352 | 0.368 | 0.439 |
| coast | 0.197 | 0.243 | 0.247 | 0.263 | 0.000 |
| columns | 0.195 | 0.249 | 0.254 | 0.286 | 0.000 |
| coral | 0.148 | 0.176 | 0.179 | 0.000 | 0.000 |
| crystal | 0.152 | 0.208 | 0.207 | 0.210 | 0.000 |
| display | 0.133 | 0.000 | 0.000 | 0.000 | 0.000 |
| doors | 0.280 | 0.336 | 0.343 | 0.382 | 0.513 |
| entrance | 0.186 | 0.265 | 0.268 | 0.282 | 0.000 |
| f-16 | 0.143 | 0.172 | 0.174 | 0.000 | 0.000 |
| face | 0.129 | 0.000 | 0.000 | 0.000 | 0.000 |
| field | 0.249 | 0.303 | 0.307 | 0.321 | 0.363 |
| fish | 0.270 | 0.318 | 0.320 | 0.406 | 0.476 |
| flowers | 0.273 | 0.314 | 0.312 | 0.333 | 0.442 |
| food | 0.148 | 0.237 | 0.239 | 0.281 | 0.000 |
| forest | 0.155 | 0.185 | 0.189 | 0.000 | 0.000 |
| gardens | 0.195 | 0.242 | 0.244 | 0.268 | 0.000 |
| grass | 0.241 | 0.303 | 0.306 | 0.341 | 0.400 |
| ground | 0.246 | 0.367 | 0.375 | 0.418 | 0.483 |
| gun | 0.150 | 0.196 | 0.212 | 0.230 | 0.000 |
| harbor | 0.129 | 0.163 | 0.174 | 0.000 | 0.000 |
| hats | 0.188 | 0.231 | 0.236 | 0.260 | 0.000 |
| hawk | 0.160 | 0.204 | 0.208 | 0.273 | 0.000 |
| head | 0.172 | 0.253 | 0.255 | 0.276 | 0.000 |
| helicopter | 0.216 | 0.264 | 0.276 | 0.289 | 0.000 |
| hills | 0.149 | 0.174 | 0.176 | 0.000 | 0.000 |
| horizon | 0.199 | 0.261 | 0.266 | 0.275 | 0.000 |
| house | 0.258 | 0.330 | 0.334 | 0.348 | 0.377 |

Table 4.28: Continued.

| word | org | $> 0.0$ | $> 0.1$ | $> 0.2$ | $> 0.3$ |
|---|---|---|---|---|---|
| insect | 0.200 | 0.238 | 0.247 | 0.272 | 0.000 |
| jet | 0.371 | 0.400 | 0.396 | 0.470 | 0.501 |
| leaf | 0.242 | 0.276 | 0.279 | 0.294 | 0.000 |
| leaves | 0.157 | 0.204 | 0.209 | 0.227 | 0.000 |
| lion | 0.201 | 0.307 | 0.311 | 0.366 | 0.508 |
| mountain | 0.239 | 0.302 | 0.305 | 0.328 | 0.348 |
| mushrooms | 0.110 | 0.134 | 0.137 | 0.000 | 0.000 |
| nest | 0.590 | 0.619 | 0.621 | 0.633 | 0.679 |
| night | 0.222 | 0.349 | 0.352 | 0.368 | 0.473 |
| ocean | 0.199 | 0.257 | 0.259 | 0.274 | 0.000 |
| outside | 0.133 | 0.000 | 0.000 | 0.000 | 0.000 |
| owl | 0.169 | 0.208 | 0.212 | 0.243 | 0.000 |
| pattern | 0.308 | 0.340 | 0.346 | 0.397 | 0.544 |
| people | 0.520 | 0.604 | 0.611 | 0.627 | 0.744 |
| pillars | 0.232 | 0.283 | 0.285 | 0.320 | 0.441 |
| plane | 0.300 | 0.358 | 0.362 | 0.392 | 0.350 |
| plants | 0.242 | 0.292 | 0.297 | 0.318 | 0.451 |
| reefs | 0.150 | 0.182 | 0.184 | 0.000 | 0.000 |
| rock | 0.237 | 0.312 | 0.319 | 0.350 | 0.408 |
| sea | 0.160 | 0.176 | 0.177 | 0.000 | 0.000 |
| seals | 0.137 | 0.000 | 0.000 | 0.000 | 0.000 |
| shrine | 0.137 | 0.167 | 0.177 | 0.000 | 0.000 |
| sky | 0.408 | 0.492 | 0.500 | 0.522 | 0.578 |
| smoke | 0.197 | 0.240 | 0.243 | 0.284 | 0.000 |
| snake | 0.129 | 0.000 | 0.000 | 0.000 | 0.000 |
| snow | 0.197 | 0.230 | 0.233 | 0.253 | 0.000 |
| statues | 0.158 | 0.192 | 0.197 | 0.000 | 0.000 |
| stone | 0.319 | 0.373 | 0.375 | 0.400 | 0.442 |
| street | 0.395 | 0.498 | 0.516 | 0.551 | 0.630 |
| sun | 0.321 | 0.396 | 0.400 | 0.443 | 0.479 |
| temple | 0.200 | 0.312 | 0.252 | 0.426 | 0.990 |
| textile | 0.246 | 0.306 | 0.314 | 0.353 | 0.546 |
| texture | 0.222 | 0.302 | 0.302 | 0.314 | 0.392 |
| tracks | 0.291 | 0.391 | 0.393 | 0.406 | 0.507 |
| tree | 0.414 | 0.489 | 0.493 | 0.504 | 0.579 |
| turn | 0.281 | 0.408 | 0.413 | 0.416 | 0.435 |
| vegetables | 0.235 | 0.265 | 0.276 | 0.306 | 0.444 |
| walls | 0.197 | 0.244 | 0.253 | 0.279 | 0.000 |
| water | 0.545 | 0.653 | 0.657 | 0.701 | 0.836 |
| waves | 0.201 | 0.274 | 0.279 | 0.289 | 0.000 |
| wildlife | 0.163 | 0.192 | 0.000 | 0.000 | 0.000 |
| windows | 0.394 | 0.491 | 0.495 | 0.513 | 0.586 |
| woman | 0.249 | 0.298 | 0.309 | 0.335 | 0.409 |
| woods | 0.247 | 0.282 | 0.283 | 0.313 | 0.391 |

Table 4.29: Recall and precision values for the predicted words on training set after training with refined vocabulary. With the increasing null threshold, recall and precision increase.

| word | org | > 0.0 | > 0.1 | > 0.2 | > 0.3 |
|---|---|---|---|---|---|
| architecture | 0.125 - 0.081 | 0.125 - 0.086 | 0.125 - 0.086 | 0.125 - 0.086 | 0.000 - 0.000 |
| art | 0.100 - 0.091 | 0.100 - 0.091 | 0.100 - 0.091 | 0.000 - 0.000 | 0.000 - 0.000 |
| background | 0.095 - 0.200 | 0.054 - 0.143 | 0.054 - 0.148 | 0.041 - 0.600 | 0.000 - 0.000 |
| beach | 0.107 - 0.113 | 0.107 - 0.114 | 0.107 - 0.115 | 0.176 - 0.109 | 0.151 - 0.117 |
| birds | 0.514 - 0.131 | 0.510 - 0.132 | 0.524 - 0.128 | 0.541 - 0.131 | 0.575 - 0.142 |
| boats | 0.219 - 0.137 | 0.225 - 0.138 | 0.225 - 0.138 | 0.219 - 0.140 | 0.243 - 0.142 |
| branch | 0.080 - 0.075 | 0.187 - 0.071 | 0.107 - 0.077 | 0.000 - 0.000 | 0.000 - 0.000 |
| buildings | 0.660 - 0.189 | 0.617 - 0.194 | 0.633 - 0.192 | 0.612 - 0.194 | 0.644 - 0.195 |
| cactus | 0.036 - 0.087 | 0.107 - 0.118 | 0.107 - 0.118 | 0.000 - 0.000 | 0.000 - 0.000 |
| car | 0.176 - 0.134 | 0.176 - 0.135 | 0.221 - 0.142 | 0.244 - 0.145 | 0.344 - 0.137 |
| church | 0.075 - 0.080 | 0.075 - 0.080 | 0.075 - 0.080 | 0.000 - 0.000 | 0.000 - 0.000 |
| cliff | 0.024 - 0.167 | 0.000 - 0.000 | 0.000 - 0.000 | 0.000 - 0.000 | 0.000 - 0.000 |
| close-up | 0.026 - 0.158 | 0.088 - 0.091 | 0.088 - 0.092 | 0.088 - 0.093 | 0.000 - 0.000 |
| closeup | 0.152 - 0.104 | 0.152 - 0.105 | 0.152 - 0.105 | 0.088 - 0.121 | 0.256 - 0.103 |
| clouds | 0.385 - 0.154 | 0.398 - 0.161 | 0.398 - 0.161 | 0.398 - 0.163 | 0.437 - 0.162 |
| coast | 0.118 - 0.116 | 0.101 - 0.117 | 0.101 - 0.117 | 0.140 - 0.104 | 0.000 - 0.000 |
| columns | 0.125 - 0.094 | 0.125 - 0.096 | 0.125 - 0.096 | 0.125 - 0.096 | 0.000 - 0.000 |
| coral | 0.147 - 0.081 | 0.147 - 0.082 | 0.147 - 0.082 | 0.000 - 0.000 | 0.000 - 0.000 |
| crystal | 0.040 - 0.167 | 0.080 - 0.136 | 0.080 - 0.136 | 0.107 - 0.114 | 0.000 - 0.000 |
| display | 0.071 - 0.079 | 0.000 - 0.000 | 0.000 - 0.000 | 0.000 - 0.000 | 0.000 - 0.000 |
| doors | 0.083 - 0.273 | 0.083 - 0.300 | 0.083 - 0.300 | 0.083 - 0.300 | 0.083 - 0.300 |
| entrance | 0.250 - 0.119 | 0.250 - 0.119 | 0.250 - 0.119 | 0.250 - 0.123 | 0.000 - 0.000 |
| f-16 | 0.143 - 0.067 | 0.143 - 0.068 | 0.143 - 0.068 | 0.000 - 0.000 | 0.000 - 0.000 |
| face | 0.093 - 0.043 | 0.000 - 0.000 | 0.000 - 0.000 | 0.000 - 0.000 | 0.000 - 0.000 |
| field | 0.133 - 0.116 | 0.133 - 0.117 | 0.133 - 0.117 | 0.133 - 0.118 | 0.133 - 0.121 |
| fish | 0.374 - 0.098 | 0.441 - 0.102 | 0.441 - 0.102 | 0.380 - 0.102 | 0.346 - 0.114 |
| flowers | 0.344 - 0.122 | 0.344 - 0.124 | 0.344 - 0.124 | 0.393 - 0.123 | 0.438 - 0.130 |
| food | 0.078 - 0.077 | 0.108 - 0.092 | 0.108 - 0.092 | 0.167 - 0.085 | 0.000 - 0.000 |
| forest | 0.103 - 0.083 | 0.103 - 0.083 | 0.103 - 0.083 | 0.000 - 0.000 | 0.000 - 0.000 |
| gardens | 0.136 - 0.123 | 0.136 - 0.124 | 0.204 - 0.109 | 0.204 - 0.111 | 0.000 - 0.000 |
| grass | 0.407 - 0.134 | 0.442 - 0.138 | 0.451 - 0.138 | 0.484 - 0.139 | 0.475 - 0.145 |
| ground | 0.329 - 0.103 | 0.329 - 0.105 | 0.329 - 0.105 | 0.329 - 0.106 | 0.355 - 0.105 |
| gun | 0.128 - 0.081 | 0.077 - 0.086 | 0.077 - 0.086 | 0.128 - 0.083 | 0.000 - 0.000 |
| harbor | 0.120 - 0.103 | 0.120 - 0.107 | 0.120 - 0.107 | 0.000 - 0.000 | 0.000 - 0.000 |
| hats | 0.065 - 0.097 | 0.217 - 0.055 | 0.217 - 0.055 | 0.217 - 0.057 | 0.000 - 0.000 |
| hawk | 0.092 - 0.088 | 0.092 - 0.091 | 0.092 - 0.091 | 0.092 - 0.091 | 0.000 - 0.000 |
| head | 0.110 - 0.097 | 0.110 - 0.098 | 0.110 - 0.098 | 0.165 - 0.099 | 0.000 - 0.000 |
| helicopter | 0.104 - 0.120 | 0.177 - 0.105 | 0.177 - 0.106 | 0.104 - 0.122 | 0.000 - 0.000 |
| hills | 0.091 - 0.078 | 0.056 - 0.087 | 0.056 - 0.087 | 0.000 - 0.000 | 0.000 - 0.000 |
| horizon | 0.044 - 0.167 | 0.044 - 0.176 | 0.044 - 0.176 | 0.044 - 0.176 | 0.000 - 0.000 |
| house | 0.027 - 0.188 | 0.027 - 0.188 | 0.027 - 0.188 | 0.027 - 0.188 | 0.027 - 0.200 |
| insect | 0.240 - 0.124 | 0.227 - 0.123 | 0.227 - 0.124 | 0.240 - 0.129 | 0.000 - 0.000 |
| jet | 0.248 - 0.151 | 0.223 - 0.156 | 0.223 - 0.156 | 0.223 - 0.156 | 0.257 - 0.165 |
| leaf | 0.241 - 0.163 | 0.241 - 0.167 | 0.241 - 0.167 | 0.241 - 0.171 | 0.000 - 0.000 |
| leaves | 0.022 - 0.115 | 0.022 - 0.115 | 0.022 - 0.115 | 0.052 - 0.132 | 0.000 - 0.000 |
| lion | 0.136 - 0.121 | 0.136 - 0.124 | 0.136 - 0.124 | 0.136 - 0.125 | 0.136 - 0.129 |

Table 4.29: Continued.

| word | org | > 0.0 | > 0.1 | > 0.2 | > 0.3 |
|---|---|---|---|---|---|
| mountain | 0.239 - 0.155 | 0.182 - 0.133 | 0.182 - 0.133 | 0.284 - 0.136 | 0.249 - 0.143 |
| mushrooms | 0.051 - 0.050 | 0.051 - 0.050 | 0.051 - 0.050 | 0.000 - 0.000 | 0.000 - 0.000 |
| nest | 0.052 - 0.375 | 0.052 - 0.375 | 0.052 - 0.375 | 0.052 - 0.429 | 0.052 - 0.500 |
| night | 0.104 - 0.118 | 0.104 - 0.119 | 0.104 - 0.121 | 0.052 - 0.143 | 0.143 - 0.124 |
| ocean | 0.121 - 0.104 | 0.181 - 0.100 | 0.181 - 0.100 | 0.121 - 0.108 | 0.000 - 0.000 |
| outside | 0.000 - 0.000 | 0.000 - 0.000 | 0.000 - 0.000 | 0.000 - 0.000 | 0.000 - 0.000 |
| owl | 0.097 - 0.101 | 0.097 - 0.103 | 0.097 - 0.104 | 0.097 - 0.104 | 0.000 - 0.000 |
| pattern | 0.195 - 0.151 | 0.224 - 0.146 | 0.224 - 0.147 | 0.233 - 0.142 | 0.238 - 0.143 |
| people | 0.805 - 0.243 | 0.829 - 0.244 | 0.817 - 0.247 | 0.818 - 0.253 | 0.825 - 0.257 |
| pillars | 0.296 - 0.123 | 0.296 - 0.125 | 0.296 - 0.125 | 0.296 - 0.127 | 0.296 - 0.132 |
| plane | 0.191 - 0.142 | 0.191 - 0.144 | 0.216 - 0.141 | 0.266 - 0.134 | 0.203 - 0.134 |
| plants | 0.234 - 0.103 | 0.234 - 0.105 | 0.234 - 0.105 | 0.234 - 0.105 | 0.280 - 0.118 |
| reefs | 0.071 - 0.094 | 0.071 - 0.094 | 0.071 - 0.094 | 0.000 - 0.000 | 0.000 - 0.000 |
| rock | 0.425 - 0.128 | 0.498 - 0.129 | 0.498 - 0.129 | 0.472 - 0.131 | 0.502 - 0.134 |
| sea | 0.082 - 0.122 | 0.082 - 0.122 | 0.082 - 0.125 | 0.000 - 0.000 | 0.000 - 0.000 |
| seals | 0.136 - 0.059 | 0.000 - 0.000 | 0.000 - 0.000 | 0.000 - 0.000 | 0.000 - 0.000 |
| shrine | 0.095 - 0.061 | 0.095 - 0.061 | 0.095 - 0.061 | 0.000 - 0.000 | 0.000 - 0.000 |
| sky | 0.801 - 0.253 | 0.793 - 0.259 | 0.788 - 0.261 | 0.786 - 0.265 | 0.797 - 0.270 |
| smoke | 0.130 - 0.130 | 0.130 - 0.130 | 0.130 - 0.130 | 0.130 - 0.130 | 0.000 - 0.000 |
| snake | 0.080 - 0.074 | 0.000 - 0.000 | 0.000 - 0.000 | 0.000 - 0.000 | 0.000 - 0.000 |
| snow | 0.103 - 0.124 | 0.103 - 0.125 | 0.103 - 0.125 | 0.103 - 0.125 | 0.000 - 0.000 |
| statues | 0.043 - 0.077 | 0.108 - 0.061 | 0.043 - 0.080 | 0.000 - 0.000 | 0.000 - 0.000 |
| stone | 0.322 - 0.119 | 0.229 - 0.131 | 0.256 - 0.121 | 0.271 - 0.125 | 0.337 - 0.133 |
| street | 0.449 - 0.190 | 0.449 - 0.192 | 0.424 - 0.194 | 0.416 - 0.200 | 0.407 - 0.205 |
| sun | 0.184 - 0.327 | 0.184 - 0.327 | 0.184 - 0.327 | 0.184 - 0.333 | 0.184 - 0.348 |
| temple | 0.261 - 0.094 | 0.297 - 0.096 | 0.297 - 0.097 | 0.243 - 0.103 | 0.243 - 0.109 |
| textile | 0.233 - 0.116 | 0.283 - 0.136 | 0.283 - 0.137 | 0.267 - 0.110 | 0.300 - 0.138 |
| texture | 0.363 - 0.128 | 0.363 - 0.130 | 0.363 - 0.130 | 0.363 - 0.131 | 0.403 - 0.133 |
| tracks | 0.265 - 0.140 | 0.265 - 0.143 | 0.229 - 0.152 | 0.265 - 0.143 | 0.265 - 0.150 |
| tree | 0.818 - 0.228 | 0.815 - 0.234 | 0.815 - 0.235 | 0.818 - 0.237 | 0.846 - 0.241 |
| turn | 0.200 - 0.200 | 0.200 - 0.200 | 0.200 - 0.200 | 0.200 - 0.200 | 0.160 - 0.250 |
| vegetables | 0.218 - 0.110 | 0.141 - 0.138 | 0.141 - 0.138 | 0.090 - 0.179 | 0.122 - 0.153 |
| walls | 0.254 - 0.093 | 0.246 - 0.102 | 0.246 - 0.102 | 0.310 - 0.098 | 0.000 - 0.000 |
| water | 0.884 - 0.289 | 0.875 - 0.294 | 0.875 - 0.295 | 0.885 - 0.294 | 0.902 - 0.304 |
| waves | 0.085 - 0.119 | 0.085 - 0.119 | 0.085 - 0.119 | 0.085 - 0.119 | 0.000 - 0.000 |
| wildlife | 0.065 - 0.077 | 0.065 - 0.082 | 0.000 - 0.000 | 0.000 - 0.000 | 0.000 - 0.000 |
| windows | 0.295 - 0.204 | 0.295 - 0.211 | 0.295 - 0.211 | 0.321 - 0.181 | 0.333 - 0.170 |
| woman | 0.417 - 0.095 | 0.408 - 0.099 | 0.350 - 0.099 | 0.417 - 0.099 | 0.359 - 0.102 |
| woods | 0.111 - 0.125 | 0.194 - 0.085 | 0.194 - 0.085 | 0.194 - 0.085 | 0.250 - 0.084 |

### 4.5.3 Effect of merging words

There are some words that are hard to distinguish, since either they are compound words, like `polar` and `bear`, or they are always used together in the annotations, like `jet` and `plane`, or it is hard to distinguish them using our feature set. In order to handle such problems, the similar words are merged, and the system is retrained on the new vocabulary consisting of word groups.

The blob posterior probabilities for each words ($p(b \mid w)$ ) are computed as discussed in Section 3.5, and similarity matrix between the words is constructed , using the Euclidean distance between the blob posterior probabilities. Then, the graph cut idea, specifically Normalized Cuts, is applied for clustering the words similar words. In the experiments, the number of word clusters (number of words in the new vocabulary) is defined as 100. Some example word combinations, obtained by merging words are: `coral-ocean, leaves-plants, sun-sunset, jet-plane-waves, beach-water, entrance-museum-pillars` and `arch-city-sculpture-walls`. This shows that, it is possible to obtain meaningful word combinations using the proposed method. After merging words, we use the new vocabulary for training, and we obtain a new probability table.

Figure 4.27 shows sample images where the words are predicted from new vocabulary which consists of merged words.

Table 4.30 shows the correspondence scores for word clusters using the hand-labeled set. We assume that, a word cluster is correctly predicted for the given blob, if one of the words in the cluster occurs as a label word.

Similarly, for computing the annotation scores, we assume that the word clusters are predicted correctly, if one of the words in the cluster is an annotation word. Table 4.31 shows the recall and precision values, for the word clusters and Table 4.32 shows true positive, false positive and false negative results as a function of word clusters.

The overall annotation and correspondence performances for the original vocabulary and for the word clusters are compared in Table 4.33.

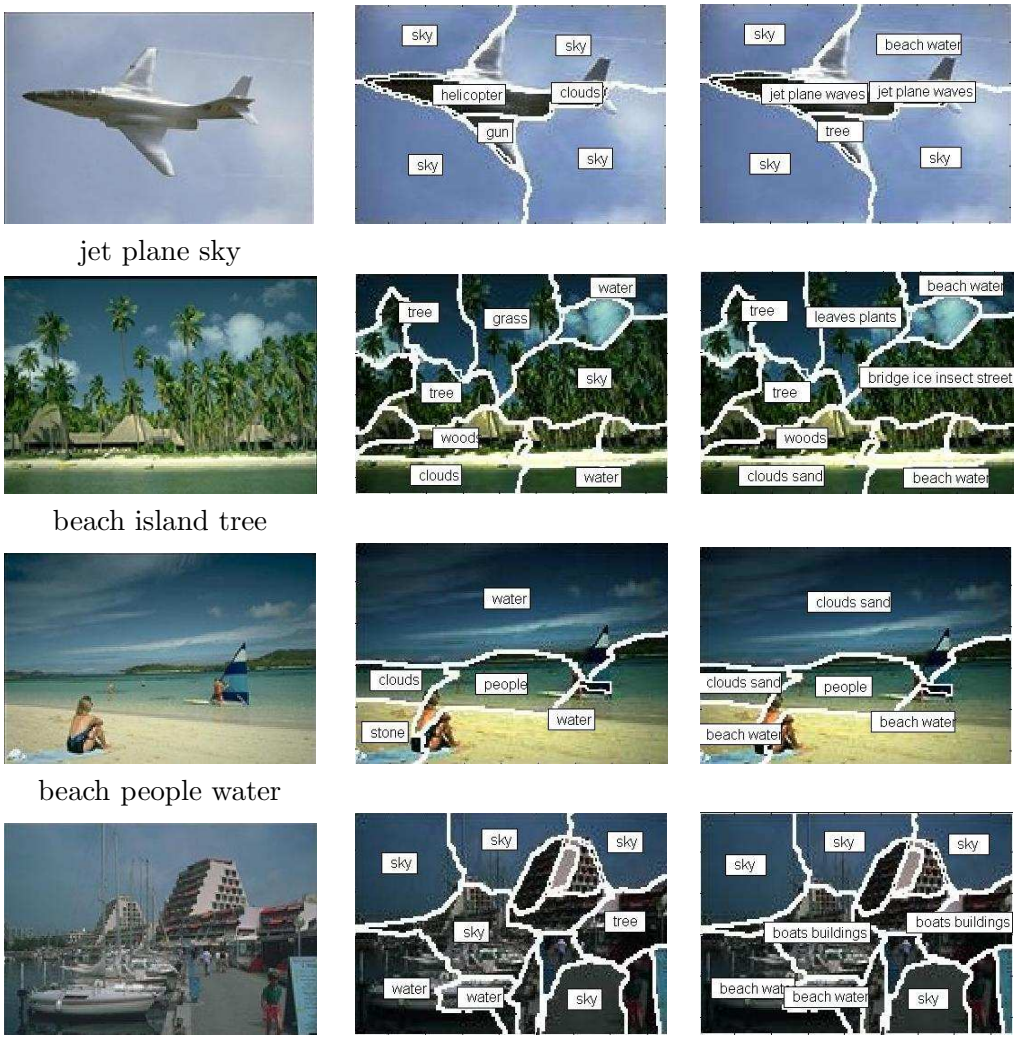



Figure 4.26: Results of merging words. First column is the original image, with annotated keywords, second column is the results for original vocabulary, third column is the results after merging indistinguishable words and retraining on the new vocabulary. When the words are merged, the performance is increased both for the new word clusters, and for the other words that remain unclustered. On the top image the words `jet,plane`,and `waves` are put into the same word cluster. The body of the plane, which was assigned to `helicopter` with the original vocabulary is assigned to the correct word cluster after retraining with merged words. Similar results occur in the other images: after merging, in most cases, one of the words in the word cluster is the correct word (e.g. `clouds-sand` and `beach-water` in the second and third images and `boats-buildings` in the last image)

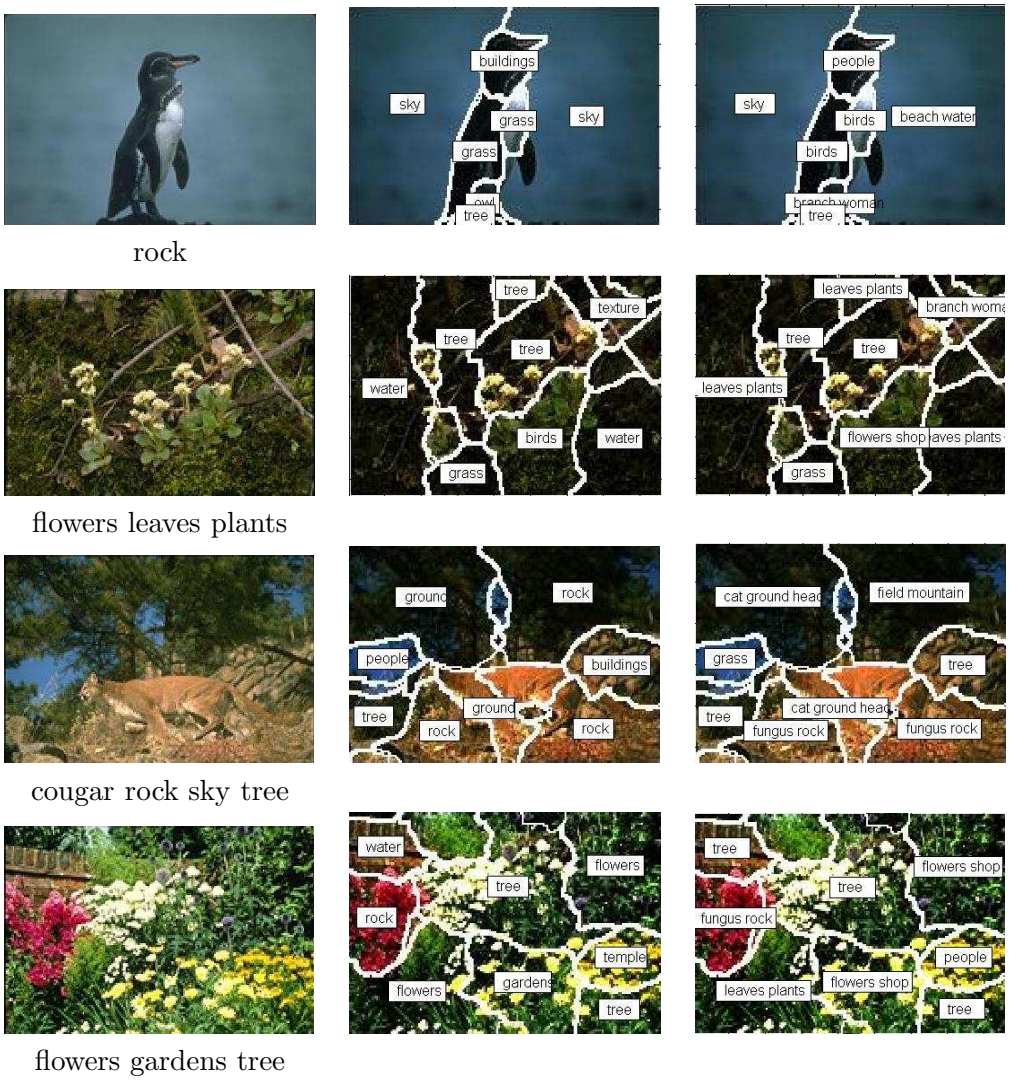



Figure 4.27: Results of merging words (continued). Prediction performance is better when the words are merged and the system is retrained: `birds` in the first image, `leaves-plants` in the second image, `cat-ground-had` in the third image, `leaves-plants` and `flowers-shop` in the last image are the examples of the correct predictions.

Table 4.30: Correspondence scores for word clusters that has nonzero prediction rates. For each word cluster: number of times that the word cluster is predicted, number of times that the word is used as the first label word (first), number of correct predictions when the first label is used for comparison, number of times that the word is used as one of the label words (all), and number of times the predicted word is one of the words used to label that region. Compare the results with Table 4.4.

| word | pred | first | correct(first) | all | correct(all) |
|---|---|---|---|---|---|
| arch city sculpture walls | 23 | 44 | 0 | 81 | 2 |
| art | 3 | 3 | 0 | 12 | 0 |
| background horse road | 8 | 35 | 0 | 38 | 0 |
| beach water | 421 | 266 | 69 | 323 | 100 |
| bears columns forest | 10 | 34 | 0 | 80 | 0 |
| birds | 95 | 61 | 4 | 71 | 4 |
| boats buildings | 104 | 159 | 36 | 220 | 36 |
| branch woman | 47 | 22 | 0 | 37 | 2 |
| bridge ice insect street | 69 | 67 | 3 | 103 | 3 |
| cactus perch seals | 18 | 29 | 0 | 31 | 0 |
| car | 7 | 10 | 1 | 10 | 1 |
| cat ground head | 63 | 122 | 6 | 206 | 6 |
| cliff horizon Scotland | 4 | 16 | 0 | 28 | 0 |
| clouds sand | 58 | 45 | 12 | 257 | 12 |
| coast helicopter hills | 80 | 42 | 4 | 88 | 5 |
| coral ocean | 57 | 56 | 2 | 65 | 2 |
| costume cougar | 2 | 15 | 0 | 15 | 0 |
| crystal | 4 | 5 | 0 | 5 | 0 |
| desert owl | 7 | 12 | 0 | 22 | 0 |
| display temple | 28 | 6 | 0 | 7 | 0 |
| doors pyramid | 1 | 10 | 0 | 10 | 0 |
| entrance museum pillars | 11 | 8 | 0 | 9 | 0 |
| fence paintings | 6 | 7 | 0 | 8 | 0 |
| field mountain | 60 | 102 | 4 | 233 | 4 |
| fish | 21 | 17 | 1 | 17 | 1 |
| flight house ruins shore | 67 | 19 | 0 | 61 | 0 |
| flowers shop | 38 | 96 | 7 | 116 | 7 |
| fungus rock | 85 | 85 | 6 | 137 | 11 |
| gardens town | 15 | 6 | 1 | 51 | 3 |
| grass | 90 | 239 | 13 | 331 | 29 |
| hunter night | 1 | 15 | 0 | 16 | 0 |
| jet plane waves | 154 | 28 | 3 | 95 | 7 |
| leaves plants | 62 | 14 | 0 | 115 | 8 |
| lion | 6 | 20 | 0 | 20 | 1 |
| palace shadows windows | 16 | 16 | 0 | 24 | 1 |
| people | 261 | 41 | 17 | 73 | 21 |
| reefs texture | 64 | 3 | 0 | 20 | 0 |
| sky | 238 | 382 | 65 | 451 | 97 |
| snow | 18 | 127 | 2 | 152 | 2 |
| statues | 1 | 21 | 0 | 25 | 0 |
| stone | 20 | 3 | 0 | 29 | 0 |
| sun sunset | 4 | 45 | 0 | 67 | 1 |
| tree | 385 | 230 | 45 | 268 | 62 |
| woods | 5 | 3 | 0 | 5 | 0 |

Table 4.31: Recall and precision for the standard test set, by clustering the words. Compare the results with Figure 4.12. to see that word clusters have higher recall and precision values. Examples: `water` has 0.870 recall and 0.326 precision, and `beach` has 0.025 recall and 0.047 precision, whereas `beach-water` has 0.988 recall and 0.333 precision; `leaves` has 0 recall and precision and `plants` has 0.026 recall and 0.067 precision, whereas `leaves-plants` has 0.125 recall and 0.125 precision.

| words | recall | precision |
|---|---|---|
| arch city sculpture walls | 0.089888 | 0.222222 |
| beach water | 0.988453 | 0.333074 |
| bears columns forest | 0.013889 | 0.142857 |
| birds | 0.141509 | 0.254237 |
| boats buildings | 0.482051 | 0.254743 |
| branch woman | 0.109375 | 0.218750 |
| bridge ice insect street | 0.116788 | 0.132231 |
| cat ground head | 0.149533 | 0.102564 |
| cliff horizon Scotland | 0.016949 | 0.333333 |
| close-up pattern | 0.225806 | 0.241379 |
| closeup | 0.022727 | 0.200000 |
| clouds sand | 0.219697 | 0.245763 |
| coast helicopter hills | 0.169811 | 0.166667 |
| coral ocean | 0.086420 | 0.120690 |
| entrance museum pillars | 0.090909 | 0.190476 |
| field mountain | 0.100775 | 0.250000 |
| fish | 0.027778 | 0.166667 |
| flight house ruins shore | 0.053571 | 0.085714 |
| flowers shop | 0.146341 | 0.179104 |
| fungus rock | 0.143885 | 0.152672 |
| grass | 0.094488 | 0.160000 |
| hunter night | 0.039216 | 0.333333 |
| jet plane waves | 0.303371 | 0.198529 |
| leaves plants | 0.125000 | 0.125000 |
| palace shadows windows | 0.047619 | 0.100000 |
| people | 0.532895 | 0.278830 |
| reefs texture | 0.147368 | 0.179487 |
| sky | 0.570513 | 0.296173 |
| stone | 0.060241 | 0.227273 |
| sun sunset | 0.078947 | 0.545455 |
| tree | 0.547855 | 0.234795 |

Table 4.32: Word prediction results when annotation is used as a proxy for the word clusters. For each word cluster: (i) number of predictions, (ii) number of occurrence of the word cluster, (iii) number of true positives, (iv) number of false positives, and (v) number of false negatives, are given. Compare the results with Table 4.6. Example: With the original vocabulary, `water` occurs 393 times in the data, and predicted 1022 times. It has 304 true positives, 718 false positives, and 89 false negatives. Similarly `beach` occurs 40 times, predicted 21 times. It has 1 true positives, 20 false positives and 39 false negatives. After training with merged words, `beach-water` has more true positive and less false positive and false negative rates.

| word cluster | pred. | occ. | tp | fp | n |
|---|---|---|---|---|---|
| arch city sculpture walls | 36 | 89 | 8 | 28 | 11 |
| beach water | 1285 | 433 | 428 | 642 | 4 |
| bears columns forest | 7 | 72 | 1 | 6 | 32 |
| birds | 59 | 106 | 15 | 44 | 91 |
| boats buildings | 369 | 195 | 94 | 236 | 44 |
| branch woman | 32 | 64 | 7 | 25 | 27 |
| bridge ice insect street | 121 | 137 | 16 | 105 | 26 |
| cat ground head | 156 | 107 | 16 | 126 | 35 |
| cliff horizon Scotland | 3 | 59 | 1 | 2 | 16 |
| close-up pattern | 87 | 93 | 21 | 59 | 30 |
| closeup | 5 | 44 | 1 | 4 | 43 |
| clouds sand | 118 | 132 | 29 | 86 | 82 |
| coast helicopter hills | 108 | 106 | 18 | 85 | 43 |
| coral ocean | 58 | 81 | 7 | 42 | 24 |
| entrance museum pillars | 21 | 44 | 4 | 13 | 13 |
| field mountain | 52 | 129 | 13 | 37 | 30 |
| fish | 12 | 72 | 2 | 10 | 70 |
| flight house ruins shore | 70 | 112 | 6 | 58 | 6 |
| flowers shop | 67 | 82 | 12 | 48 | 64 |
| fungus rock | 131 | 139 | 20 | 100 | 19 |
| grass | 75 | 127 | 12 | 63 | 115 |
| hunter night | 6 | 51 | 2 | 4 | 25 |
| jet plane waves | 272 | 178 | 54 | 147 | 44 |
| leaves plants | 88 | 88 | 11 | 64 | 42 |
| palace shadows windows | 20 | 42 | 2 | 15 | 16 |
| people | 581 | 304 | 162 | 419 | 142 |
| reefs texture | 78 | 95 | 14 | 57 | 36 |
| sky | 601 | 312 | 178 | 423 | 134 |
| stone | 22 | 83 | 5 | 17 | 78 |
| sun sunset | 11 | 76 | 6 | 4 | 39 |
| tree | 707 | 303 | 166 | 541 | 137 |

Table 4.33: Comparison of the prediction results obtained from the probability tables constructed from the original vocabulary (original) and from the merged vocabulary (merged). For comparison normalized classification scores (NS) and prediction measures(PR) are computed on the training and standard test sets, and correspondence scores are obtained by comparing the predicted word with the first label word (first) and with all of the label words (all) in the hand-labeled set.

| | original | merged |
|---|---|---|
| NS - standard test | 0.2012 | 0.2242 |
| PR - standard test | 0.2171 | 0.2395 |
| NS - training | 0.2708 | 0.2490 |
| PR - training | 0.2560 | 0.2616 |
| correspondence(first) | 315 | 301 |
| correspondence(all) | 444 | 429 |

## 4.6 Integrating labeled data

Creating a hand-labeled data is very labor intensive. However, a small number of carefully created labeled data can be used for many purposes. In this section, two strategies are proposed to make use of hand-labeled data

- Clustering based on the labeled data,
- Integrating supervision to the system to fix the correspondence errors.

### 4.6.1 Data sets

6 CD's from the Corel data set is used to test the effect of labeled data. Ten images from each CD are labeled by hand (each CD contains 100 images). Since each CD represents a specific topic such as **tigers**, **planes**, etc., only a few keywords are sufficient to describe a CD. The vocabulary is reduced to only the label words required to describe each CD, and it is assumed that words are represented differently in different CD's (i.e. that the word `sky` in the **tiger** CD is a different word than `sky` in the **planes** CD — in particular, it may be depicted differently). The word `NULL` is added to the vocabulary, and to the annotations of each image.

### 4.6.2 Using labeled data for clustering

In order to use the labeled data for better clustering, first, PCA is applied to our feature set (there are 30 features in the original set) and the largest 11 values are chosen. Then, Linear Discriminant Analysis is performed on the labeled set. The blobs that share the same label word are assigned to the same class, (if there are two sky words in the vocabulary, then there are two different sky classes) and the unlabeled blobs are assigned to an **outlier** class. Therefore, there are 22 classes: one for each word and one for the outliers.

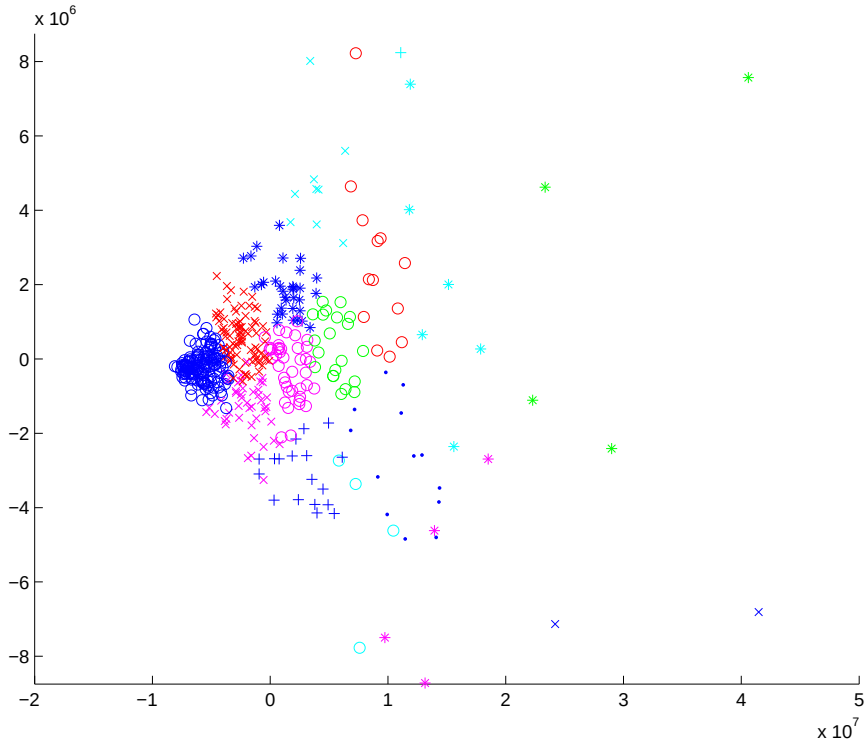
In order to find the classes for the blobs in the remaining images, nearest neighbor method is applied on the new feature space. In Figure 4.27, the blob clusters for labeled data are shown for LDA results and for k-means(it is projected onto 2D space using the multidimensional scaling method). In the LDA method, although the elements of an individual class (features of the blobs corresponding to a single word) is usually close together, there are many overlaps between the elements of different classes. Since, k-means only uses visual features, but not the word information the distribution is very different

### 4.6.3 Using labeled data for fixing the correspondence errors

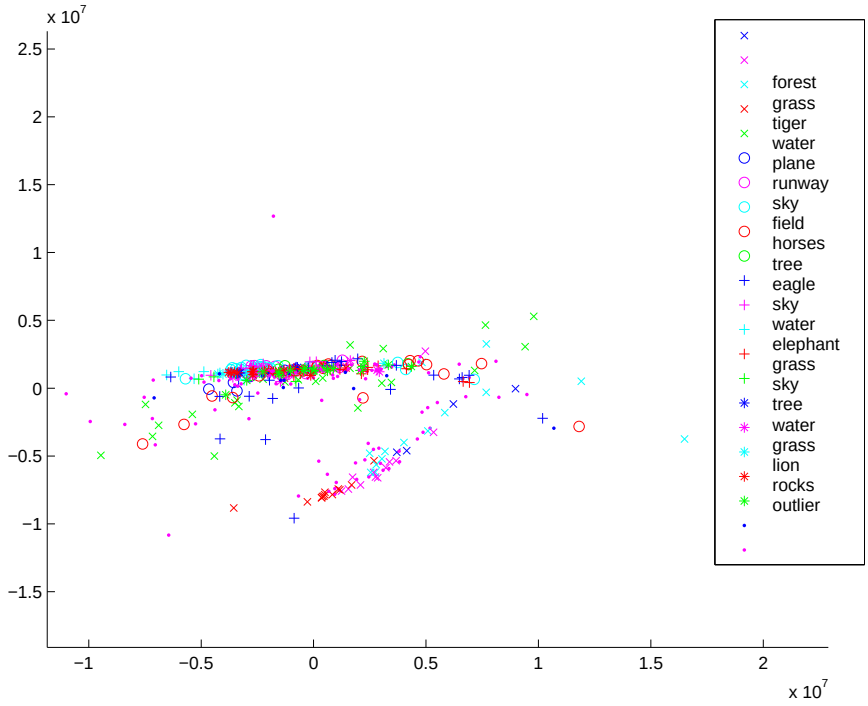
There are some limits in learning the correspondences, when only the unlabeled data is used. For example, if `horses` always occur with `grass`, then it is not possible to learn which blob is `horse` and which blob is `grass`. However, if the system is supervised by assigning the brown blob to `horse`, then it is possible to solve the correspondence problem between `horses` and `grass`. Therefore, we integrate the labeled data into the system for supervision by fixing the correspondences on the labeled set.

Instead of starting with uniform alignments and trying to estimate the correct alignment probabilities, on the labeled data, we insist that the alignments between the blobs and their label words are 1, but the alignments with the other words are 0. Therefore, the system is forced to learn the correct alignments for the rest of the data.

Each CD is split into a 70 image unlabeled training set, a 10 image labeled training set (where word region correspondences are manually identified) and 20 image test set. To initialize the system, for the 10 labeled images, the correspondences are fixed, and




(a)




(b)

Figure 4.27: Blob clusters for labeled data : (a) using k-means with 22 centers, (b) using label words for clustering by applying linear discriminant analysis.

for the rest of the training set, it is assumed that there is a uniform distribution. Then, the system is retrained with EM while keeping the correspondences fixed for the 10 labeled images during the training process.

### 4.6.4 Strategies for improving the system with labeled data

In order to test the effect of integrating labeled data, the following strategies are compared (the strategies are summarized in Table 4.34).

1. The basic approach, where only unlabeled data are used.

2. A method where labeled data are used to produce clusters of image regions, but where the joint probability table between clusters of image regions and words is learned using unlabeled data.

3. A method where labeled data are used to produce clusters of image regions and where the joint probability table between clusters and words is learned with a combination of labeled and unlabeled data.

4. A method where labeled data is used to produce a nearest neighbor classifier — image regions are assigned the label of the nearest labeled example.

Table 4.34: Summary of the strategies to use labeled data: There are four methods compared. **Clustering strategy** is either based on k-means or using labeled data. **Training strategy** can be EM on only unlabeled data (unlabeled + EM); or both on unlabeled data and labeled data where the correspondences are fixed (labeled data + unlabeled data + EM). When the labeled data is used for clustering another alternative is using the nearest neighbor classifier.

| method | clustering strategy | training strategy |
|---|---|---|
| method 1 | k-means | unlabeled data + EM |
| method 2 | labeled data | unlabeled data + EM |
| method 3 | labeled data | labeled data + unlabeled data + EM |
| method 4 | labeled data | nearest neighbor classifier |

In Figure 4.28, examples from the results are shown for each of the four methods. It is seen that using labeled data for clustering improves the performance over k-means, and using labeled data for training has a better performance than not using it.

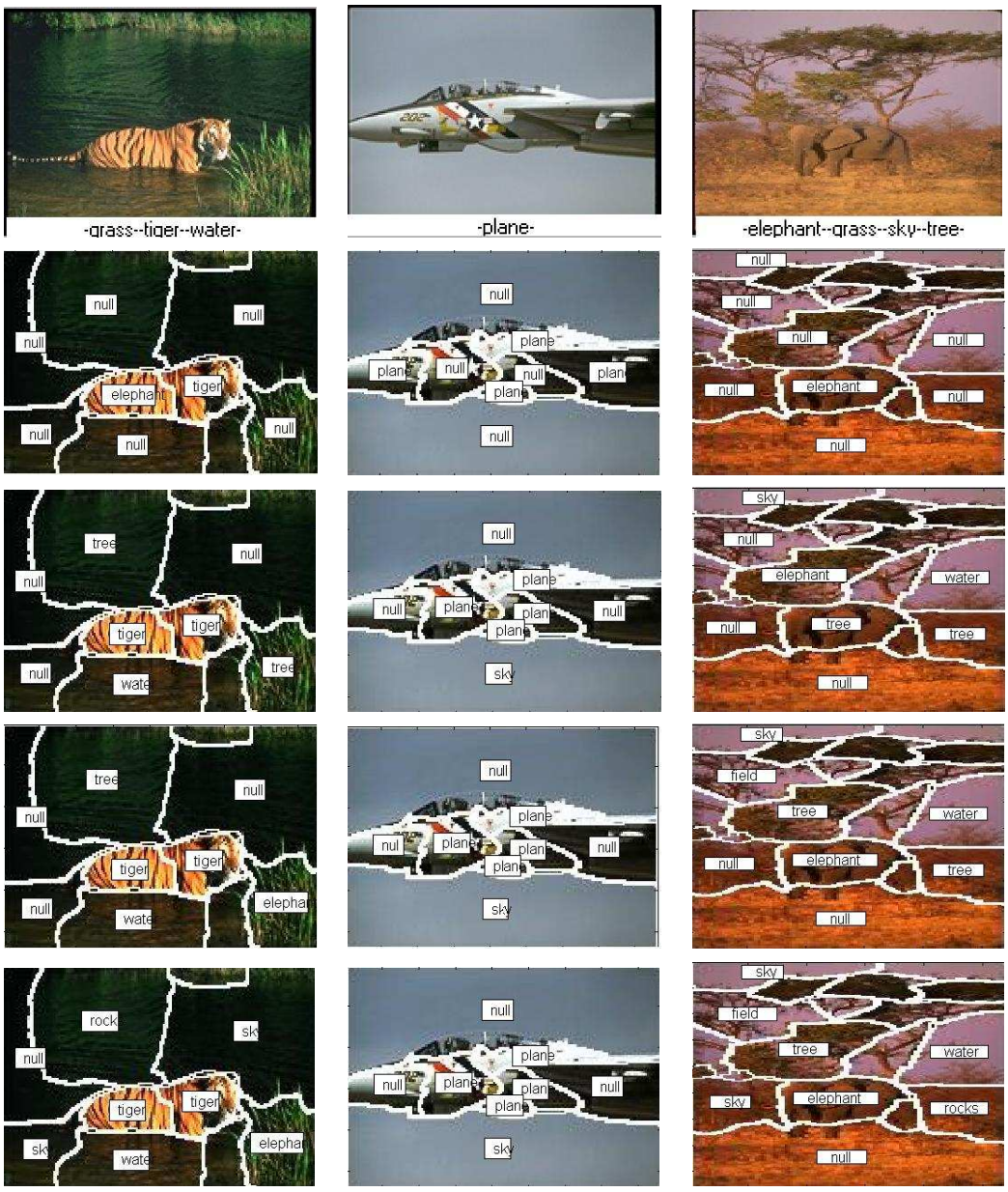



Figure 4.28: Sample images: from top to bottom, original image with the annotated keywords, results for method1(k-means), method2(unsupervised), method3(supervised), method4(nearest neighbor classifier). The results show that, using labeled data for clustering improves the performance over k-means, and using labeled data for training improves the performance over the methods that use only unlabeled data.

Table 4.35: The mutual information for joint probability tables linking words and blobs for 15 words and 22 blobs, in bits, constructed using three different methods. The maximum possible value is 3.72. Notice that supervisory information on image region clusters alone appears to make little difference, but supervising both clustering and correspondence results in a significant difference.

| method | mutual information |
|---|---|
| method 1 | 1.25 |
| method 2 | 1.24 |
| method 3 | 1.32 |

In Table 4.35, mutual information is used to compare the first three methods. It is seen that, supervisory information on image region clusters alone appears to make little difference, but supervising both clustering and correspondence results in a significant difference.

In Table 4.36, we list the first three words with the highest probability for each blob. The predictions are performed using method 1, 2 and 3. We compare the predicted words, with the words used for labeling that blob. It is seen that, when labeled data is used both for clustering and for fixing, the word prediction performance is increased.

In order to compare *method 3* which uses EM with fixed correspondences and *method 4*, we check the results visually on the test data as shown in Table 4.37. We count the number of correct correspondences by checking whether we are predicting the correct word on the correct place. We look for the ratio of *correct count / number of all predictions* for each label word for the **supervised** and **nearest neighbor classifier** methods. It can be seen that although the correct counts are higher for label based method, when we look at the number of all predictions the ratios are lower than supervised method.

We also check the false positive and false negative results. For this experiment, the auto-annotation words for an image are found by taking the union of the words predicted by each blob in the image. Table 4.38 shows the false positive and false negative rates for method 3 and method 4. As seen from the results, false positive rates are smaller when EM is used with labeled data (compared to nearest neighbor classifier method). The nearest neighbor classifier method predicts more words making more mistakes. The proposed method is more reliable in that sense.

Table 4.36: The first 3 words with the highest probability for the given blob cluster: **k-means:**for k-means clustering (method1), **unsupervised:** for using labeled data in clustering, but correspondences are not fixed(method2) and **supervised:** for using labeled data both for clustering and fixing correspondences (method3). The words in parenthesis are the original label words for the blobs in that cluster. The supervised method where labeled data is used both for clustering and for fixing correspondences is the best over all. In most of the cases the word predicted with the highest probability is the correct label word. It is also seen that, using labeled data for clustering improves the performance over using k-means for clustering.

| class label | k-means | unsupervised | supervised |
|---|---|---|---|
| 1 (forest) | null eagle sky | horses field forest | horses field forest |
| 2 (grass) | null water elephant | grass tiger water | grass tiger water |
| 3 (tiger) | elephant horses field | tiger null water | tiger null water |
| 4 (water) | null grass horses | water eagle grass | water eagle grass |
| 5 (plane) | sky plane forest | plane sky null | plane null sky |
| 6 (runway) | null sky eagle | runway plane eagle | runway plane eagle |
| 7 (sky) | null lion rocks | null sky eagle | null sky eagle |
| 8 (field) | plane null sky | null horses field | field null elephant |
| 9 (horses) | tiger null forest | null tiger tree | horses null tiger |
| 10 (tree) | null water tiger | horses lion null | lion tree null |
| 11 (eagle) | plane null runway | null eagle sky | null eagle tiger |
| 12 (sky) | forest sky tiger | sky eagle null | sky null eagle |
| 13 (water) | null field horses | null eagle water | null water eagle |
| 14 (elephant) | sky null grass | tree elephant null | elephant null tree |
| 15 (grass) | horses null plane | grass horses null | grass horses field |
| 16 (sky) | null elephant horses | sky elephant tree | sky tree field |
| 17 (tree) | plane sky runway | elephant horses null | tree field horses |
| 18 (water) | tiger plane water | water null sky | water null sky |
| 19 (grass) | plane sky null | null lion grass | null grass lion |
| 20 (lion) | tiger null plane | grass lion tiger | lion grass tiger |
| 21 (rocks) | null eagle tiger | tree elephant null | tree elephant water |
| 22 (null) | plane sky null | null horses sky | null horses sky |

Table 4.37: Correspondence scores computed by visually inspecting the results for method 3 (EM with labeled data) and method 4 (nearest neighbor classifier).

| word | EM with labeled data | nearest neighbor classifier |
|---|---|---|
| eagle | 0 / 0 | 4 / 63 |
| elephant | 5 / 30 | 4 / 30 |
| field | 6 / 54 | 6 / 54 |
| forest | 0 / 0 | 0 / 5 |
| grass | 10 / 31 | 19 / 54 |
| horses | 5 / 42 | 5 / 37 |
| lion | 2 / 35 | 2 / 23 |
| plane | 9 / 40 | 9 / 40 |
| rocks | 0 / 0 | 1 / 28 |
| runway | 2 / 8 | 2 / 8 |
| sky | 13 / 48 | 29 / 92 |
| tiger | 8 / 50 | 9 / 50 |
| tree | 6 / 48 | 5 / 32 |
| water | 3 / 40 | 6 / 70 |

Table 4.38: False positive and false negative rates as a function of words for method 3 (EM with labeled data) and method 4 ( nearest neighbor classifier ).

| word | EM with labeled data | nearest neighbor classifier |
|---|---|---|
| eagle | 0.000000-1.000000 | 0.848684-0.671429 |
| elephant | 0.775701-0.657143 | 0.775701-0.657143 |
| field | 0.851351-0.541667 | 0.851351-0.541667 |
| forest | 0.000000-1.000000 | 0.952381-0.904762 |
| grass | 0.773585-0.636364 | 0.780749-0.378788 |
| horses | 0.823129-0.628571 | 0.849558-0.757143 |
| lion | 0.752066-0.571429 | 0.758242-0.685714 |
| plane | 0.702381-0.615385 | 0.702381-0.615385 |
| rocks | 0.000000-1.000000 | 0.988372-0.909091 |
| runway | 0.764706-0.428571 | 0.764706-0.428571 |
| sky | 0.663043-0.720721 | 0.681319-0.477477 |
| tiger | 0.736842-0.500000 | 0.736842-0.500000 |
| tree | 0.866667-0.513514 | 0.935484-0.837838 |
| water | 0.803279-0.661972 | 0.804734-0.535211 |

As seen from the results, using labeled data for clustering instead of using k-means clusters improves the performance. Although assigning the label word to each class seems better in some cases than training with EM, when we look at the false positive and false negative rates, translation method is better than a nearest neighbor classifier. Also, using labeled data for fixing the correspondences gives better results than not using supervision.

# CHAPTER 5

# CONCLUSIONS, DISCUSSIONS AND FUTURE DIRECTIONS

In this thesis, we propose a new approach to the object recognition problem, motivated by the recent availability of large annotated image data sets. This approach formalizes object recognition problem as the translation of image regions to words, similar to the translation of text from one language to another. The "lexicon" for the translation is learned from large annotated image collections, which consist of images that are associated with text. A machine translation method proposed by Brown *et. al.* [20] is adapted to discover the correspondences between image regions and words.

The proposed method can be summarized as follows: First, images are segmented into regions, each of which are represented by a pre-specified feature vector. Then, the regions (of all the training images) are clustered in the feature space, categorizing the regions into a finite set of blobs. The correspondences between the blobs and the words are learned, using a method based on the Expectation Maximization method. This process is analogous with learning a lexicon from an aligned bitext.

Once, the correspondences between words and image regions are learned from the training set, the system can be used to;

- predict words corresponding to particular image regions (region naming), and
- predict words associated with whole images (auto-annotation).

This study is inspired from the work of Barnard and Forsyth [17], where the images are linked to words, based on the joint probabilities that are learned using a variant

of Hoffman's hierarchical aspect model [41]. However, unlike the proposed method, their work do not explicitly model the relationships between specific image regions and words.

The method is applied on the Corel data set, a large collection of stock photographs annotated by a set of keywords. A series of experiments are carried out to evaluate the performance of the method. First, the accuracy of predicted words is evaluated on a relatively small number of hand-labeled images. Then, the method is evaluated using annotation performance as a proxy. Annotation performance is evaluated using three measures: KL divergence between the predicted and target distributions, normalized classification score and word prediction rate. There is no ground truth that can be used for comparing the performance of the proposed method. Two methods are used for comparison: predictions using empirical word densities and the co-occurrences of blobs and words. The results clearly show that, the proposed method has a better performance than these two methods.

On a large test set, the method predicts numerous words with high accuracy. Simple methods are proposed to identify words that are not predicted well and the system is retrained on a reduced vocabulary consisting of words with better prediction rates. Individual words are grouped into word clusters to improve the performance for the words that cannot be distinguished using the current set of features.

Although the system is purposely designed as unsupervised, for some cases integration of supervisory input improves the performance. A small number of images are labeled manually to use supervised data for better clustering and to fix the correspondence errors.

The proposed method is attractive, because it allows us to attack a variety of otherwise inaccessible problems in object recognition: There has been little work to address object recognition at a broad scale. For example, not much is known on how to recognize thousands of different objects from data sets that are practically available. Little can be said about what is easy and what is hard to recognize using a particular set of features. These questions become possible to discuss when recognition is considered as a process that learns the correspondences between words and image regions using a large data set.

With the proposed system it is possible to learn a large number of objects. The

size of the objects that can be learned is related with the size of the vocabulary. It is not required to specify a list of objects to be learned. The system learns the type of objects that can be recognized by deciding on the prediction performance. Many types of objects from a diverse set of images can be recognized.

## 5.1 Future directions

The work presented in this thesis should be considered as a proof-of-concept. Therefore, it leaves a number of issues open-ended for future research;

- The segmented regions of the images are represented by a set of simple basic features. We make no claim that the image features adopted are canonical. They are chosen to be computable for any image region, and be independent of any recognition hypothesis. Construction of a feature set that can offer a better performance for the proposed set remains an open question.

- The feature vectors of the regions are clustered using the k-means algorithm, where number of classes is set to a predefined value. It is likely that better clustering can improve the performance of the system, and hence needs to be investigated.

- The proposed system can be a useful tool in evaluating segmentation and feature extraction algorithms on the large-scale. In [15], a number of features and segmentation algorithms are compared based on their word prediction performances.

- The annotations in the Corel data set are relatively simple in the sense that they consist of individual keywords and the vocabulary is relatively small. Many data sets, mentioned above, contain free text annotations. In such data sets, natural language processing is required to identify candidate annotations that appear to refer to the picture.

- The annotations of Corel data set creates various problems for the proposed system. First, the range of the occurrences of words are large: while some words such as occur frequently, others appear rarely. This causes a poor prediction rate for words that do not occur often. NULL word prediction, and retraining

on a reduced vocabulary were proposed to handle such problems. Improvement of the performance of the word predictions using the word distributions in the training set, remains an interesting issue that needs to be tackled.

- In this study, we propose a strategy for merging the indistinguishable words. Melamed [55] proposes a greedy algorithm for grouping some elements of the lexicon to deal with compound words. A similar approach can be employed for our system.

- Typically, we expect our system to predict the same word for different parts of an object. Grouping the neighboring regions that predicts the same word can allow us to obtain better segments, where an object is represented by a single region. Learning how to group segments simultaneously with the construction of the lexicon is an interesting problem that needs to be considered.

- The regions are clustered without the use of the word information. The methods that cluster regions (rather than quantizing their representation) to ensure that region clusters are improved by word information should give better blobs, therefore better prediction performance.

As seen, there are many research issues that are uncovered by the proposed approach, as well as many possible ways of improving the system.

# REFERENCES


[1] Calflora. http://www.calflora.org.

[2] Corel Data Set. http://www.corel.com/products/clipartandphotos.

[3] Fine Arts Museum of San Francisco. http://www.thinker.org/fam/thinker.html.

[4] Hulton Getty Archive. http://search.hultongetty.com/.

[5] Informedia Project. http://informedia.cs.cmu.edu.

[6] TV archive. http://televisionarchive.org.

[7] Web archive. http://www.archive.org.

[8] Yahoo News. http://news.yahoo.com.

[9] L.-M. Albiges. Remote public access to picture databanks. *Audiovisual Librarian*, 18(1):22–27, 1992.

[10] L.H. Armitage and P.G.B. Enser. Analysis of user need in image archives. *Journal of Information Science*, 23(4):287–299, 1997.

[11] K. Barnard, P. Duygulu, N. de Freitas, D. A. Forsyth, D. Blei, and M. Jordan. Matching words and pictures. *Journal of Machine Learning Research*, 3:1107–1135, 2003.

[12] K. Barnard, P. Duygulu, and D. A. Forsyth. Clustering art. In *IEEE Conf. on Computer Vision and Pattern Recognition*, volume 2, pages 434–441, 2001.

[13] K. Barnard, P. Duygulu, and D. A. Forsyth. Modeling the statistics of image features and associated text. In *Document Recognition and Retrieval IX, Electronic Imaging*, 2002.

[14] K. Barnard, P. Duygulu, and D. A. Forsyth. Recognition as translating images into text. In *Internet Imaging IX, Electronic Imaging*, 2003.

[15] K. Barnard, P. Duygulu, R. Guru, P. Gabbur, and D. A. Forsyth. The effects of segmentation and feature choice in a translation model of object recognition. In *IEEE Conf. on Computer Vision and Pattern Recognition*, 2003.

[16] K. Barnard and D. A. Forsyth. Exploiting image semantics for picture libraries. In *The First ACM/IEEE-CS Joint Conference on Digital Libraries*, page 469, 2001.

[17] K. Barnard and D. A. Forsyth. Learning the semantics of words and pictures. In *Int. Conf. on Computer Vision*, pages 408–15, 2001.

[18] J. Bilmes. A gentle tutorial on the em algorithm and its application to parameter estimation for gaussian mixture and hidden markov models. Technical Report ICSI-TR-97-021, University of Berkeley, 1997.

[19] A. Blum and T. Mitchell. Combining labeled and unlabeled data with co-training. In *Proceedings of the Workshop on Computational Learning Theory, Morgan Kaufmann Publishers*, 1998.

[20] P.F. Brown, S. A. Della Pietra, V. J. Della Pietra, and R. L. Mercer. The mathematics of statistical machine translation: Parameter estimation. *Computational Linguistics*, 19(2):263–311, 1993.

[21] C. Carson, S. Belongie, H. Greenspan, and Jitendra Malik. Blobworld: Color and texture based image segmentation using em and its application to image querying and classification. *PAMI*, 24(8):1026–1038, 2002.

[22] M. La Cascia, S. Sethi, and S. Sclaroff. Combining textual and visual cues for content-based image retrieval on the world wide web. 1998.

[23] S. Chang and A. Hsu. Image information systems: Where do we go from here? *IEEE Trans. on Knowledge and Data Enginnering*, 4(5):431–442, October 1992.

[24] F. Chen, U. Gargi, L. Niles, and H. Schutze. Multi-modal browsing of images in web documents. In *SPIE Document Recognition and Retrieval*, 1999.

[25] J.-Y. Chen, C.A. Bouman, and J.C. Dalton. Hierarchical browsing and search of large image databases. In *IEEE Transactions on Image Processing*, volume 9, pages 442–455, 2000.

[26] N. de Freitas, K. Barnard, P. Duygulu, and D. Forsyth. Bayesian models for massive multimedia databases: a new frontier. In *7th Valencia International Meeting on Bayesian Statistics/2002 ISBA International Meeting*, June 2-6, 2002.

[27] A. P. Dempster, N. M. Laird, and D.B. Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society, Series B*, 1(39):1–38, 1977.

[28] Y. Deng and B. S. Manjunanth. An efficient low-dimensional color indexing scheme for region-based image retrieval. In *Proc. of. IEEE Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP)*, 1999.

[29] J. Dowe. Content-based retrieval in multimedia imaging. In *Proceedings SPIE Storage and Retrieval For Image and Video Databases*, 1993.

[30] R. O. Duda, P. E. Hart, and D. G. Stork. John Wiley and Sons, Inc., New York, 2001.

[31] P. Duygulu, K. Barnard, N.d. Freitas, and D. A. Forsyth. Object recognition as machine translation: learning a lexicon for a fixed image vocabulary. In *Seventh European Conference on Computer Vision (ECCV)*, volume 4, pages 97–112, 2002.

[32] P.G.B. Enser. Query analysis in a visual information retrieval context. *Journal of Document and Text Management*, 1(1):25–39, 1993.

[33] P.G.B. Enser. Progress in documentation pictorial information retrieval. *Journal of Documentation*, 51(2):126–170, June 1995.

[34] C. O. Frost et. al. Browse and search patterns in a digital image database. volume 1, pages 287–313, 2000.

[35] H. Evans. *Practical picture research: a guide to current practice, procedure, techniques and resources.* London:Blueprint, 1992.

[36] C. Faloutsos, R. Barber, M. Flickner, J. Hafner, W. Niblack, D. Petkovic, and W. Equitz. Efficient and effective querying by image content. *Journal of Intelligent Information Systems*, 3(3-4):231–262, 1994.

[37] M.M. Fleck, D.A. Forsyth, and C.Bregler. Finding naked people. In *4th European Conference on Computer vision*, volume 2, pages 591–602, 1996.

[38] D. A. Forsyth and J. Ponce. *Computer Vision: a modern approach.* Prentice-Hall, 2001.

[39] T. Gevers and A.W.M. Smeulders. Pictoseek:combining color and shape invariant features for image retrieval. *IEEE Trans. on Image Processing*, 9(1):102–119, January 2000.

[40] A. A. Goodrum. Image information retrieval: An overview of current research. *Informing Science*, 3(2):63–66, 2000.

[41] T. Hofmann. Learning and representing topic. a hierarchical mixture model for word occurrence in document databases. In *Workshop on learning from text and the web, CMU*, 1998.

[42] T. Hofmann and J. Puzicha. Statistical models for co-occurrence data. Technical Report 1635, Massachusetts Institute of Technology, 1998.

[43] F. Idris and S. Panchanathan. Review of image and video indexing techniques. *Journal of Visual Communication and Image Representation*, 8(2):146–166, 1997.

[44] C. Jorgenson. *Image attributes: An investigation.* PhD thesis, Syracuse University, 1995.

[45] C. Jorgenson. Indexing images: Testing an image description template. In *ASIS 1996 Annual Conference Proceedings*, October 1996.

[46] D. Jurafsky and J. H. Martin. *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics and Speech Recognition.* Prentice-Hall, 2000.

[47] L.H. Keister. User types and queries: impact on image access systems, challenges in indexing electronic text and images. *Learned Information*, 1994.

[48] W. Y. Ma and B. S. Manjunanth. Netra: A toolbox for nevigating large image databases. In *Proc. of International Conference On Image Processing*, 1997.

[49] C. D. Manning and H. Schutze. *Foundations of Statistical Natural Language Processing*. MIT Press, 1999.

[50] M. Markkula and E. Sormunen. End-user searching challenges indexing practices in the digital newspaper photo archive. *Information retrieval*, 1:259–285, 2000.

[51] M. Markkula, M. Tico, B. Sepponen, K. Nirkkonen, and E. Sormunen. A test collection for the evaluation of content-based image retrieval algorithms - a user and task-based approach. *Information retrieval*, 4(3/4):275–294, 2001.

[52] O. Maron. *Learning from Ambiguity*. PhD thesis, MIT, 1998.

[53] O. Maron and A. L. Ratan. Multiple-instance learning for natural scene classification. In *The Fifteenth International Conference on Machine Learning*, 1998.

[54] M. De Marsicoi, L. Cinque, and S. Levialdi. Indexing pictorial documents by their content: A survey of current techniques. *Image and Vision Computing*, 15(2):119–141, 1997.

[55] I. D. Melamed. *Empirical Methods for Exploiting Parallel Texts*. MIT Press, 2001.

[56] T. P. Minka and R. W. Picard. Interactive learning using a society of models. *Pattern Recognition*, 30(4):565–581, 1997.

[57] Y. Mori, H. Takahashi, and R. Oka. Image-to-word transformation based on dividing and vector quantizing images with words. In *First International Workshop on Multimedia Intelligent Storage and Retrieval Management*, 1999.

[58] C. Nastar, M. Mitschke, C. Meilhac, and N. Boujemaa. Surfimage: a flexible content-based image retrieval system. In *Proceedings of the 6th ACM International Conference on Multimedia*, pages 339–344, September 1998.

[59] K. Nigam, A. K. McCallum, S. Thrun, and T. M. Mitchell. Learning to classify text from labeled and unlabeled documents. In *Proceedings of AAAI-98, 15th Conference of the American Association for Artificial Intelligence*, pages 792–799, Madison, US, 1998. AAAI Press, Menlo Park, US.

[60] B. C. O'Connor. Access to moving image documents: Background concepts and proposals for surrogates for film and video works. *Journal of Documentation*, 41(4):209–220, 1985.

[61] V.E. Ogle and M. Stonebraker. Chabot: Retrieval from relational database of images. *Computer*, 28(9):40–48, 1995.

[62] C. Okon. Image recognition meets content management for authoring, editing and more. *Advanced Imaging*, 10(7), 1995.

[63] M. Oren, C. Papageorgiou, P. Sinha, and E. Osuna. Pedestrian detection using wavelet templates. In *Computer vision and pattern recognition*, 1997.

[64] S. Ornager. View a picture, theoretical image analysis and empirical user studies on indexing and retrieval. *Swedis Library Research*, 2-3:31–41, 1996.

[65] A. Pentland, R. W. Picard, and S. Sclaroff. Photobook: Content-based manipulation of image databases. *International Journal on Computer Vision*, 18(3):233–254, 1996.

[66] Y. Rui, T. S. Huang, and S. Chang. Image retrieval: Current techniques, promising directions, and open issues. *Journal of Visual Communication and Image Representation*, 10(4):39–62, 1999.

[67] Y. Rui, T. S. Huang, M. Ortega, and S. Mehrotra. Relevance feedback: A power tool in interactive content-based image retrieval. *IEEE Transactions On Circ. Sys. Video Tech*, September 1998.

[68] H. Schneiderman and T.Kanade. A statistical approach to 3d object recognition applied to faces and cars. In *IEEE Conference on Computer Vision and Pattern Recognition*, page 100, 2000.

[69] S. Sclaroff, L. Taycher, and M. La Cascia. Imagerover: A content-based image browser for the world wide web. In *Proc. IEEE Workshop on Content-based Access of Image and Video Libraries*, 1997.

[70] I. K. Sethi, I. Coman, B. Day, F. Jiang, D. Li, J. Segovia-Juarez, G. Wei, and B. You. Color-wise: A system for image similarity retrieval using color. In *Proceedings of SPIE, Storage and Retrieval for Image and Video Databases VI*, volume 3312, 1998.

[71] L. Shapiro and G. Stockman. *A new Computer Vision Textbook*. Prentice-Hall, 2001.

[72] J. Shi and J. Malik. Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):888–905, 2000.

[73] J. R. Smith and S. F. Chang. Visually searching the web for the content. *IEEE Multimedia Magazine*, 4(3):12–20, 1997.

[74] J. R. Smith and S.F. Chang. Visualseek: A fully automated content-based image query system. In *Proceedings of ACM Multimedia 96*, 1996.

[75] R. Srihari. *Extracting Visual Information from Text: Using Captions to Label Human Faces in Newspaper Photographs*. PhD thesis, SUNY at Buffalo, 1991.

[76] R. K. Srihari and D.T Burhans. Visual semantics: Extracting visual information from text accompanying pictures. In *AAAI 94*, 1994.

[77] R.K. Srihari, R. Chopra, D. Burhans, M. Venkataraman, and V. Govindaraju. Use of collateral text in image interpretation. In *ARPA Image Understanding Workshop*, 1994.

[78] Swain and Ballard. Color indexing. *International Journal of Computer Vision*, 7, 1991.

[79] M. J. Swain, C. Frankel V., and Athitsos. Webseer: An image search engine for the world wide web. Technical Report TR-96-14, Computer Science Department, University of Chicago, 1996.

[80] J. M. Turner. Indexing "ordinary" pictures for storage and retrieval. *Visual Resources*, X:265–273, 1994.

[81] J. Z. Wang and J. Li. Mining digital imagery data for automatic linguistic indexing of pictures. In *Proc. NSF Workshop on Next Generation Data Mining*.

[82] G. Wei, D. Li, and I. K. Sethi. Web-wise: Compressed image retrieval over the web. In *Proc. of Multimedia Information Analysis and Retrieval, IAPR International Workshop*, 1998.

# VITA

Pınar Duygulu-Şahin received her B.S. and M.S. degrees from the Department of Computer Engineering, Middle East Technical University, Turkey in 1996 and 1998 respectively. She worked as a visiting scholar in Digital Library Project at the University of California, Berkeley, USA, between February 2001 and May 2002. She received the "Best paper in Cognitive Vision" award in European Conference on Computer Vision (ECCV 2002). Her research interests include computer vision and machine learning; specifically object recognition, browsing and retrieval in image collections and document analysis. She is a member of IEEE Computer Society and Turkish Pattern Recognition and Image Analysis Society.

**Publications**

Matching Words and Pictures. Kobus Barnard, Pınar Duygulu, David Forsyth, Nando de Freitas, David Blei, Michael Jordan. Journal of Machine Learning Research, Vol. 3, pp.1107-1135, 2003.

Object Based Image labeling through Learning-by-Example and Multi-level Segmentation. Yaowu Xu, Pınar Duygulu, Eli Saber, Murat Tekalp, Fatoş Yarman Vural. Pattern Recognition, Vol. 36, no. 6, pp.1407-1423, June 2003.

A Hierarchical Representation of Form Documents for Identification and Retrieval. Pınar Duygulu, Volkan Atalay, International Journal on Document Analysis and Recognition. IJDAR 5 (2002) 1, 17-27.

The effects of segmentation and feature choice in a translation model of object recognition. Kobus Barnard, Pınar Duygulu, Raghavendra Guru, Prasad Gabbur, David Forsyth. Computer Vision and Pattern Recognition, CVPR, 2003.

Recognition as translating images into text. Kobus Barnard, Pınar Duygulu, David Forsyth. Internet Imaging IX, Electronic Imaging 2003.

Object Recognition as Machine Translation: Learning a lexicon for a fixed image vocabulary. Pınar Duygulu, Kobus Barnard, Nando de Freitas, David Forsyth. European Conference on Computer Vision (ECCV) Copenhagen, 2002, (also published in Lecture Notes in computer Science, Volume 2353, pp.97)

Bayesian Models for Massive Multimedia DatabasesL a new frontier. Nando de Freitas, Kobus Barnard, Pınar Duygulu, David Forsyth. 7th Valencia International Meeting on Bayesian Statistics/2002 ISBA International meeting, June 2-6, 2002, Spain.

Modeling the statistics of image features and associated text. Kobus Barnard, Pınar Duygulu, David Forsyth. SPIE Electronic Imaging 2002, Document Recognition and Retrieval IX, 20-25 January 2002, San Jose, California, USA.

Clustering Art. Kobus Barnard, Pınar Duygulu, David Forsyth. Computer Vision and Pattern Recognition (CVPR 2001), December 9-14, 2001, Hawaii.

Multi-Level Image Segmentation and Object Representation for Content Based Image Retrieval. Pınar Duygulu, Fatoş Yarman Vural. SPIE Electronic Imaging 2001, Storage and Retrieval for Media Databases, January 21-26, 2001, San Jose, CA.

Object Based Image Retrieval Based On Multi-level Segmentation. Yaowu Xu, Pınar Duygulu, Eli Saber, Murat Tekalp, Fatoş Yarman Vural. IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP2000), June 5-9, 2000, Istanbul, Turkey.

Multi-level Object Description:Color or Texture. Pınar Duygulu, Abdurrahman Çarkacıoğlu, Fatoş Yarman Vural. First IEEE Balkan Conference on Signal Processing, Communications, Circuits, and Systems, June 1-3, 2000, Istanbul, Turkey.

A Hierarchical Representation of Form Documents for Identification and Retrieval. Pınar Duygulu, Volkan Atalay, Document Recognition and Retrieval, SPIE Electronic Imaging 2000.

Iki Boyutlu Goruntulerden Uc Boyutlu Nesne Olusturulmasina Eniyileme Yaklasimi. (in Turkish) Pınar Duygulu, Kemal Leblebicioğlu, Volkan Atalay, Veysi İşler. IEEE

1999 Sinyal İşleme ve Uygulamaları Kurultayı (SIU'99), Ankara, Turkey, June 1999.

A Heuristic Algorithm For Hierarchical Representation of Form Documents. Pınar Duygulu, Volkan Atalay, Ebru Dincel. 14th Int'l Conf. Pattern Recognition (ICPR98), Brisbane, Australia, August 1998.

Form Document Representation and Identification. Pınar Duygulu, Volkan Atalay, Ebru Dincel. The Seventh Turkish Symposium on Artificial Intelligence and Neural Networks, Ankara, Turkey, June 1998.

A Form Document Image Parser. Ebru Dincel, Volkan Atalay, Pınar Duygulu. The Seventh Turkish Symposium on Artificial Intelligence and Neural Networks, Ankara, Turkey, June 1998.

Logical Structure Representation Of Form Documents Based on Line Information. Pınar Duygulu, Volkan Atalay, Ebru Dincel. Technical Report, TR-97-6, Dept. of Computer Engineering, METU.