

# Automatic Image Tagging

*Sean Moran*



Master of Science  
School of Informatics  
University of Edinburgh  
2009

## **Abstract**

Automatic Image Tagging seeks to assign relevant words (e.g. “jungle”, “boat”, “trees”) to images that describe the actual content found in the images without intermediate manual labelling. Current approaches are largely based on categorization, and treat the tags independently, so an annotation (jungle,trees) is just as plausible as (jungle,snow). The goal of this dissertation was to develop a probabilistic model (the Continuous Relevance Model) to take into account the dependencies between keywords so as to provide more precise annotations. The main findings suggest that, under certain conditions, taking into account keyword correlation, coupled with an efficient method (beam search) to search over sets of tags is an effective method to increase annotation accuracy.

## **Acknowledgements**

Many thanks to my supervisor Victor Lavrenko for his direction and advice throughout the dissertation. I also wish to express my gratitude to the School of Informatics for funding my MSc degree through the Collaborative Training Account (CTA) Studentship.

## **Declaration**

I declare that this thesis was composed by myself, that the work contained herein is my own except where explicitly stated otherwise in the text, and that this work has not been submitted for any other degree or professional qualification except as specified.

*(Sean Moran)*

# Table of Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	Why Automatic Image Tagging? . . . . .	1
1.2	Limitations of Automatic Image Tagging . . . . .	5
1.3	Problem Identification . . . . .	7
1.4	Aims and Objectives . . . . .	8
1.5	Summary of Main Results . . . . .	8
1.6	Dissertation Structure . . . . .	9
<b>2</b>	<b>Background</b>	<b>11</b>
2.1	Automatic Image Tagging Challenges . . . . .	11
2.2	Existing Image Tagging Models . . . . .	13
2.3	Capturing keyword correlation . . . . .	18
2.4	The Continuous Relevance Model (CRM) . . . . .	21
2.4.1	Overview . . . . .	21
2.4.2	Image representation . . . . .	22
2.4.3	Annotation Model . . . . .	22
2.4.4	Image Annotation and Retrieval . . . . .	26
2.5	Evaluating Image Tagging Performance . . . . .	26
2.5.1	Annotation Performance . . . . .	27
2.5.2	Retrieval Performance . . . . .	28
2.6	Reducing computational complexity through Beam Search . . . . .	30
<b>3</b>	<b>Methodology</b>	<b>33</b>
3.1	Software & Architecture . . . . .	33
3.2	Image Pre-processing . . . . .	35
3.2.1	COREL Dataset . . . . .	35
3.2.2	PASCAL Dataset . . . . .	37
3.3	Feature Pre-processing . . . . .	43
3.4	CRM Model Implementation . . . . .	44

3.4.1	Original CRM Model . . . . .	44
3.4.2	Annotating Images . . . . .	47
3.4.3	Adding Keyword Correlation . . . . .	49
3.4.4	The BS-CRM Model . . . . .	50
3.5	Evaluation Framework . . . . .	55
3.5.1	Cross-Validation . . . . .	55
3.5.2	Integration with Trec Eval . . . . .	55
3.5.3	Custom Evaluation Functions . . . . .	56
<b>4</b>	<b>Evaluation</b>	<b>57</b>
4.1	Experimental Methodology . . . . .	57
4.1.1	COREL Dataset . . . . .	58
4.1.2	PASCAL Dataset . . . . .	58
4.2	COREL: N-CRM Model . . . . .	59
4.2.1	Image Annotation Performance . . . . .	59
4.2.2	Ranked Retrieval Performance . . . . .	73
4.3	COREL: Dirichlet Model . . . . .	75
4.3.1	Image Annotation Performance . . . . .	75
4.4	COREL: Multinomial Model . . . . .	79
4.4.1	Image Annotation Performance . . . . .	79
4.5	COREL: Bernoulli Model . . . . .	82
4.5.1	Image Annotation Performance . . . . .	82
4.6	PASCAL: N-CRM Model . . . . .	85
4.6.1	Ranked Retrieval Performance . . . . .	85
4.6.2	Image Annotation Performance . . . . .	95
<b>5</b>	<b>Conclusions and Future Work</b>	<b>101</b>
5.1	Overview . . . . .	101
5.2	Summary of dissertation achievements . . . . .	101
5.3	Limitations . . . . .	102
5.4	Future Work . . . . .	103
<b>A</b>		<b>106</b>
A.1	Example Source Code Listing . . . . .	106
A.1.1	Image annotation algorithm . . . . .	106
A.1.2	Beam search algorithm . . . . .	109
A.1.3	Non-parametric kernel density estimation . . . . .	112
A.1.4	Image feature extraction . . . . .	113

A.1.5 Cross validation framework . . . . .	115
<b>Bibliography</b>	<b>119</b>

# List of Figures

1.1	Growth in images stored online at Flickr.com . . . . .	2
1.2	The Google image labelling website . . . . .	3
1.3	Query by Sketch . . . . .	4
1.4	Google image search vs. CBIR image search . . . . .	6
2.1	Examples of pose variation and illumination changes . . . . .	12
2.2	Illustration of the Semantic Gap . . . . .	13
2.3	Block based image annotation flowchart . . . . .	15
2.4	Feature extraction system flowchart . . . . .	16
2.5	Example of CRM annotations on the COREL image dataset. . . . .	17
2.6	Benefits of capturing keyword correlation . . . . .	19
2.7	Calculating Mean Average Precision (MAP) . . . . .	29
2.8	Illustration of the beam search algorithm . . . . .	32
3.1	The architecture of the BS-CRM Model . . . . .	34
3.2	Remarkable similarity of images in the COREL dataset . . . . .	37
3.3	Example images from the PASCAL VOC 2007 dataset . . . . .	38
3.4	SIFT detector and descriptors applied to an example PASCAL image . . . . .	40
3.5	Illustration of the use of Gabor texture features on an example image. . . . .	44
3.6	Contents of the main matrices used in the custom CRM model . . . . .	46
3.7	Amending the CRM model to capture keyword correlation in the manner suggested by Wang et al. . . . .	51
3.8	The proposed BS-CRM model using beam search to find a close to optimal set of tags for an image. . . . .	52
3.9	Illustration of the operation of the proposed BS-CRM model on a toy example. . . . .	53
3.10	The BS-CRM algorithm expressed in matrix terminology. . . . .	54
4.1	Optimization of annotation $\beta$ and $\mu$ for No-Beam N-CRM Model, annotation length=5 . . . . .	61



4.2	Performance comparison of the BS-CRM model with annotation length 5 against the literature. . . . .	63
4.3	Chart depicting the mean per word precision for 70 COREL words, N-CRM model, Annotation Length=5 . . . . .	66
4.4	Chart depicting the mean per word recall for 70 COREL words, N-CRM model, annotation length=5 . . . . .	67
4.5	Chart depicting the effect of beam width on F1 measure for the N-CRM model, Annotation Length=4 . . . . .	69
4.6	Ranked retrieval optimization of annotation $\beta$ and $\mu$ for no beam N-CRM Model	74
4.7	Optimization of annotation $\beta$ and $\mu$ for no beam D-CRM Model, Annotation Length=5 . . . . .	77
4.8	Optimization of annotation $\beta$ and $\lambda$ for no beam M-CRM Model, Annotation Length=5 . . . . .	80
4.9	Optimization of annotation $\beta$ and $\lambda$ for no beam B-CRM Model, Annotation Length=5 . . . . .	83
4.10	Optimization of retrieval $\beta$ and $\mu$ on the PASCAL dataset for the no beam N-CRM Model . . . . .	86
4.11	Recall-precision charts for the aeroplane, bicycle, bird, boat and bottle classes in the PASCAL dataset . . . . .	88
4.12	Recall-precision charts for the bus, car, cat, chair and cow classes in the PASCAL dataset . . . . .	89
4.13	Recall-precision charts for the table, dog, horse, motorbike and person classes in the PASCAL dataset . . . . .	90
4.14	Recall-precision charts for the plant, sheep, sofa, train and TV monitor classes in the PASCAL dataset . . . . .	91
4.15	Ranked retrieval results of the N-CRM model on the PASCAL dataset (horse class) . . . . .	92
4.16	Ranked retrieval results of the N-CRM model on the PASCAL dataset (person class) . . . . .	93
4.17	Ranked retrieval results of the N-CRM model on the PASCAL dataset (tv-monitor class) . . . . .	94
4.18	Optimization of annotation $\beta$ and $\mu$ for No-Beam N-CRM Model, Annotation Length=5 . . . . .	96
4.19	Example annotations on the PASCAL dataset . . . . .	98
4.20	Example annotations on the PASCAL dataset . . . . .	99
4.21	Examples of pruning noisy keywords on the PASCAL dataset . . . . .	100

# List of Tables

4.1	N-CRM model performance on the COREL testing dataset (Annotation Length=5) for differing beam widths . . . . .	62
4.2	Table comparing the BS-CRM model developed in this dissertation against the state-of-the-art results from the literature . . . . .	64
4.3	Table demonstrating the actual labels assigned by the BS-CRM model to some of the COREL test set images . . . . .	65
4.4	N-CRM model performance on the COREL testing dataset (Annotation Length=4) for differing beam widths . . . . .	70
4.5	N-CRM model performance on the COREL testing dataset (Annotation Length=3) for differing beam widths . . . . .	72
4.6	Table demonstrating the image retrieval performance of the N-CRM model . . .	75
4.7	D-CRM model performance on the COREL testing dataset (Annotation Length=5) for differing beam widths. . . . .	78
4.8	M-CRM model performance on the COREL testing dataset (Annotation Length=5) for differing beam widths. . . . .	81
4.9	B-CRM model performance on the COREL testing dataset (Annotation Length=5) for differing beam widths. . . . .	84
4.10	Table displaying the Average Precision results obtained by the N-CRM model on the PASCAL dataset . . . . .	87
4.11	N-CRM model performance on the PASCAL testing dataset (Annotation Length=5) for differing beam widths . . . . .	97

# Chapter 1

## Introduction

### 1.1 Why Automatic Image Tagging?

Over the past decade the number of images being captured and shared has grown enormously. There are several factors behind this remarkable trend. In the modern age it is now commonplace for private individuals to own at least one digital camera, either attached to a mobile phone, or as a separate device in its own right<sup>1</sup>. The ease with which digital cameras allow people to capture, edit, store and share high quality images in comparison to the old film cameras. This factor, coupled with the low cost of memory and hard disk drives, has undoubtedly been a key driver behind the growth of personal image archives. Furthermore, the popularity of social networking websites such as Facebook and Myspace, alongside image sharing websites such as Flickr (see Figure 1.1) has given users an extra incentive to capture images to share and distribute amongst friends all over the world<sup>2</sup>.

Substantial still image archives are also being amassed in the commercial domain. Forsyth, in *Computer Vision: A Modern Approach* [15], cites some examples of commercial organizations that have substantial still image archives. One particular example includes dedicated Stock Photo archives, such as Getty and Rex Features which have many thousands if not millions of still images stored within their computer networks. Another example are Newspapers; Markkula and Sormunen [38] studied the image archive of a Finnish Newspaper, and described how archivists annotated pictures with keywords, with Journalists searching the image collection based on those keywords. These companies take the tagging of images very seriously indeed, employing teams of people to manually view each image in turn and assign relevant keywords to describe the contents of the images [45] [38]. Even the search behemoth, Google,

---

<sup>1</sup>The 2008 IDC whitepaper on the scale of the “Digital Universe” put the number of Digital Cameras and Camera Phones in use at over 1 Billion worldwide in 2006 [17]

<sup>2</sup>To give an idea of the current scale of online image libraries, studies in 2007 suggested that the Internet photo sharing website, Flickr.com, has 40 million monthly visitors a month and hosts two billion photos, with millions of new photos being added on a daily basis. Towards the end of 2008 Facebook was reported to have amassed 10 billion photos.

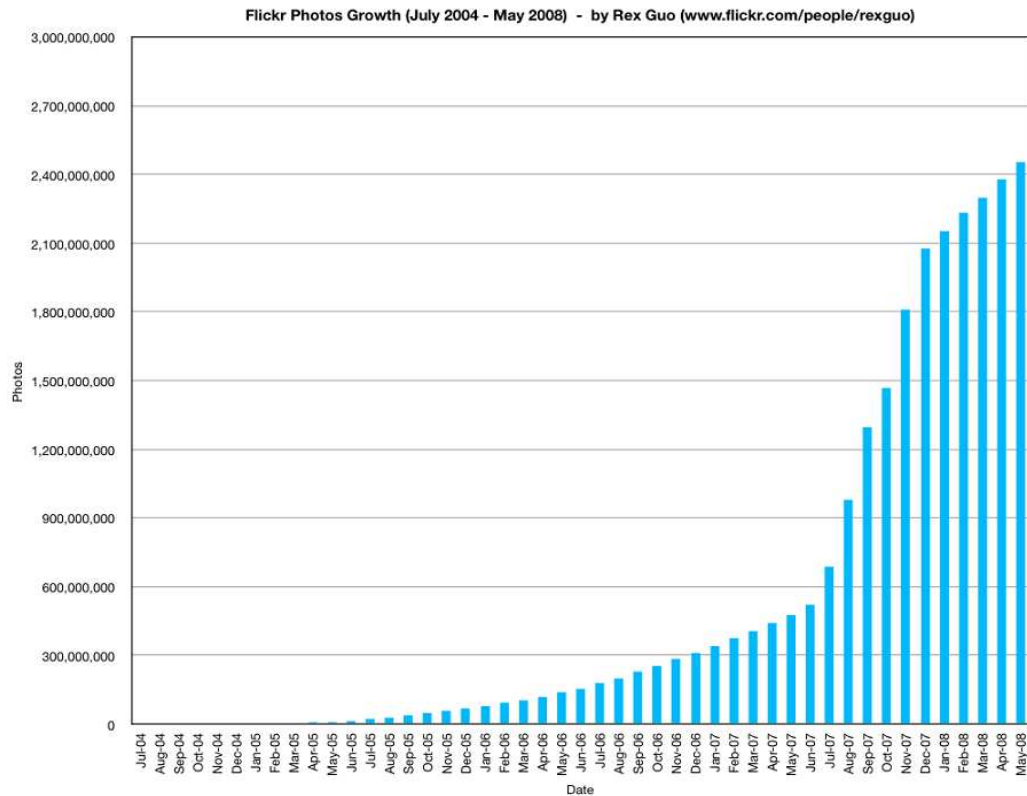


Figure 1.1: Chart depicting the growth in the number of images being stored online on Flickr.com between July 2004 and May 2008. Chart derived from published image count results by Flickr.com and courtesy of: <http://www.flickr.com/photos/rexguo/2467112209/>

has attempted to recruit its own users to tag random images from its index (see Figure 1.2), by re-framing the process as a collaboration between users with those tags matching between users selected as the labels for the images<sup>3</sup>.

For commercial organizations, correct keywording of images has a direct effect on their revenues and efficiency in satisfying the needs of consumers; an incorrectly or insufficiently labelled image is unlikely to be found, particularly within the stringent deadlines commonly experienced within the commercial world, thereby leading to a loss in operational efficiency. The social study conducted by Ames et al. [1] provided some insights into the motivations that drive private individuals to annotate their images. This study revealed a changing opinion of the usefulness of tagging, from it being nearly completely avoided for personal offline collections through to it being heartily embraced for online collections such as those on Flickr.com. The authors revealed a taxonomy of reasons behind this increase in motivation, with one of the most interesting being the social incentives brought about by online libraries, where for example,

<sup>3</sup><http://images.google.com/imagelabeler/>

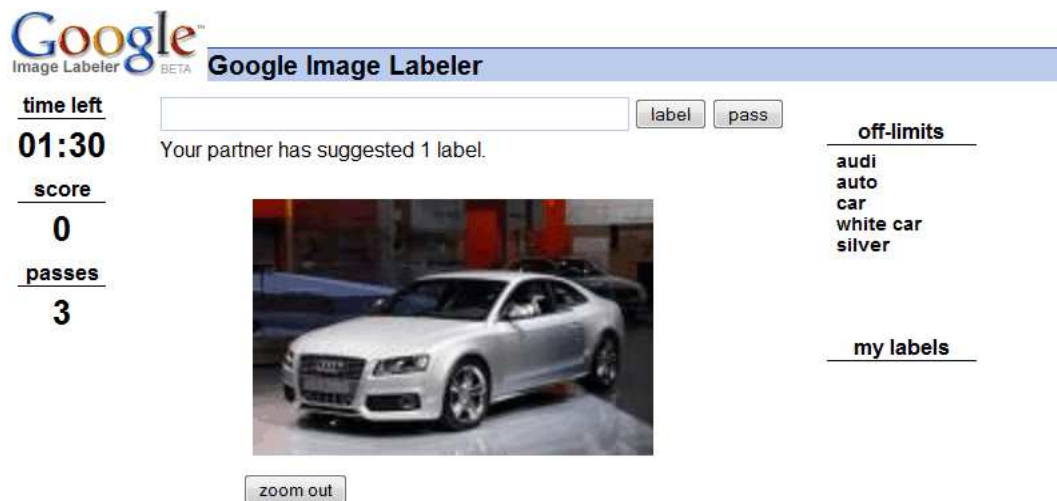


Figure 1.2: This image depicts the Google Image Labelling website where users compete against each other to assign the most relevant labels to randomly selected images from Google's index. Given that the Search Giant is using this manual means of image tagging demonstrates the difficulty inherent in the automated image tagging process particularly with regard to scaling those models suggested in the literature to multi-million scale image libraries. A great deal of work needs to be completed before the models of the research literature can be migrated as robust and scalable technologies to the commercial world.

a photographer may obtain the “satisfaction” of having made available a highly popular (or most viewed) photograph on the website. These social factors have the potential to drive the popularity of automatic image tagging<sup>4</sup> tools amongst the general public in the future.

Nevertheless, the explosion in the amount of images being captured and stored has meant that the vast majority of images, particularly those residing online on the Internet, have no associated keywords to describe their content. Manual labelling clearly suffers from the disadvantages of not only being slow, expensive and highly subjective, but just as importantly given the current explosion in the number of images being captured and stored, this method is clearly not scalable to multi-million image libraries.

Given the immense practical applications of a means of automatic image tagging, along with the deep academic challenges associated with recognising real world objects within images, it is not surprising to find that there has been great interest amongst the computer vision and information retrieval community in the development of robust and efficient automatic image tagging systems. The main purpose of tagging images in this manner is to allow for the retrieval of images based on natural language keywords as opposed to alternative content based image retrieval (CBIR) techniques such as query by sketch or query by example. Query by

<sup>4</sup>“Tagging” is used interchangeably with “annotation” throughout this dissertation.

sketch (shown in Figure 1.3) and query by example have been largely dismissed as less flexible and user friendly<sup>5</sup> means of querying image libraries than the familiar query by text already employed to search document collections, and as such there appears to be a shift of focus in the community towards CBIR by textual query [41] [24] [14] [10] [13] [23] [5]. Automatic image annotation technology will be at the forefront of this revolution in enabling users to use familiar natural language search interfaces to retrieve images of relevance.

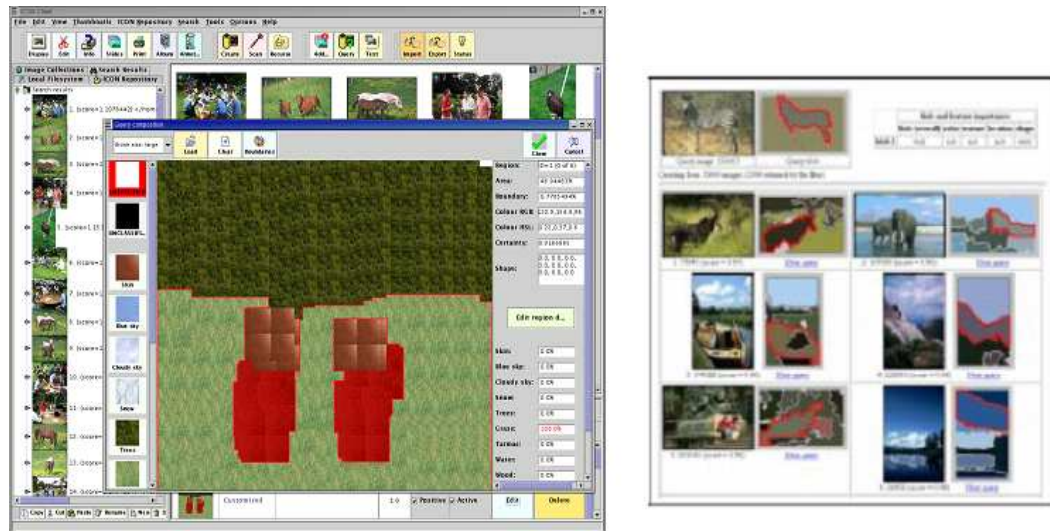


Figure 1.3: *Query by Sketch. On the left: the user makes a rough sketch of the desired image characteristics. On the right: the retrieved images that have a similarity to the sketch. This technique for image retrieval is time consuming and makes it hard to represent abstractions and invariants. Source: Imense Ltd.*

At the present time companies such as Behold<sup>6</sup> and Imense Ltd<sup>7</sup> have already entered the CBIR market with their own specialized CBIR Search Engines. Behold specializes in searching just over 1 million high quality images from the Flickr.com website. In the case of the Imense search service, a user can click on professional images and be brought straight to the copyright owner's website, thereby providing the company with advertising revenue in the spirit of Google's business model. Imense Ltd's key insight is to provide a means for users to search large collections of images by means of a specially designed query language built around a large ontology of visual content such as objects, scene features, and properties<sup>8</sup>. The company also offers a standalone Image Auto Tagging tool to organizations to annotate their image libraries. Given the poor performance of the major search engines with regards to image

<sup>5</sup>Users are known to find it particularly difficult to represent their image needs via abstract image features [24] [64]

<sup>6</sup>[www.behold.cc](http://www.behold.cc)

<sup>7</sup>[www.imense.com](http://www.imense.com)

<sup>8</sup>For further information on the Imense technology please refer to the introductory presentation: <http://www.nesc.ac.uk/talks/ahm2008/1117.pdf>

search<sup>9</sup> (Figure 1.4), it's not difficult to imagine that further start up companies, and indeed the current text based search incumbents, will likely enter this potentially lucrative market in the near future with their own CBIR search services.

## 1.2 Limitations of Automatic Image Tagging

Having as so far advocated the use of automatic image annotation it is worth stepping back for a moment and considering the other side of the technology and some of the inherent limitations of the approach that are performed much better by manual means of annotation. Enser [11] cites two main examples as to why manual annotation is superior to automated image annotation in some cases.

Firstly, the so-called *visibility limitation*, attempts to describe how automated image tagging algorithms typically depend on successfully linking visible image features to words. It is very difficult for automated algorithms to capture content and contextual information from images that do not have any associated image features. Enser provides the CBIR query, “*find a picture of the first public engagement of Prince Charles*” as a prime example of content that would be hard to automatically extract from images.

In addition, the author goes on to mention another significant limitation in the form of *generic object limitation*, which questions the use of very generic tags for the images such as “sun”, “grass” and “tiger”:

“...they have the common property of visual stimuli which require a minimally-interpretive response from the viewer.” Enser et al. [11]

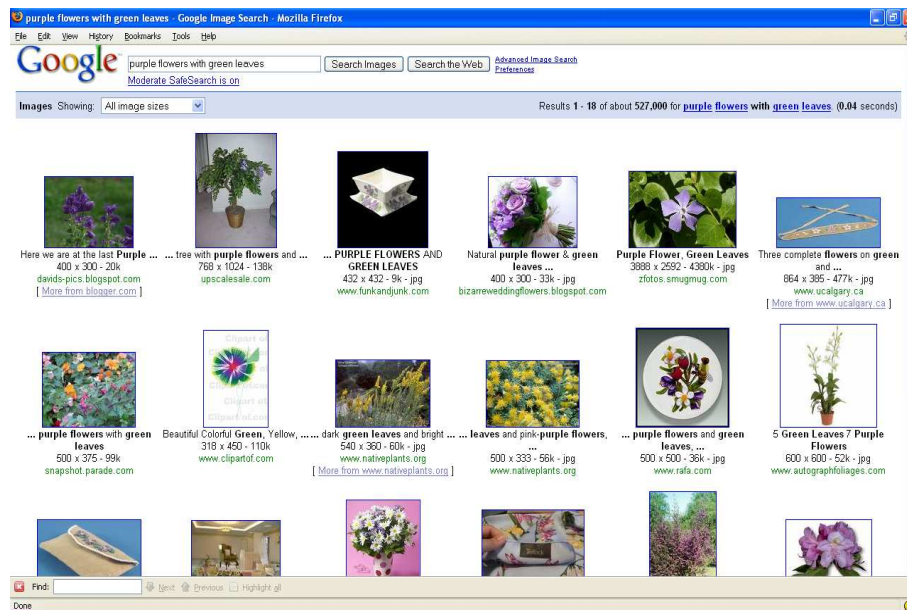
Enser cites numerous studies that demonstrate the fact that most users tend to issue queries that refer to objects by proper name which usually have limited associated visual stimuli in images. Enser concludes his thesis by stating that any defining textual annotations will necessarily always have to be manually assigned to images.

Despite the critique of Enser, many authors [61] still consider the search for robust and accurate image tagging systems to be of paramount importance and benefit given, as has been mentioned, the proliferation of untagged image libraries. It is widely regarded that “*semantic indexing*” of such images using the current breed of automatic image tagging systems, whilst not capturing the conceptual properties, is still infinitely better than having no associated keywords at all.

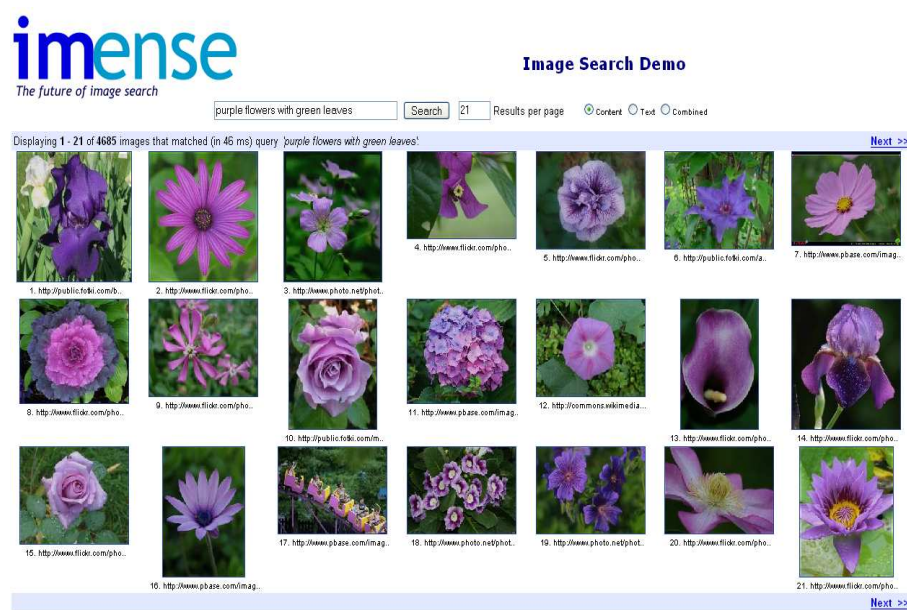
---

<sup>9</sup>Most large search engines today such as Yahoo or Google use surrounding text, such as image filenames or web page content, to index images on the web. Using this text as a cue for image content is not very effective as can be witnessed by their poor performance.





(a)



(b)

Figure 1.4: Typing “Purple Flowers with Green Leaves” into Google (Figure 1.4(a)) yields many results which are completely irrelevant or are of poor quality. In contrast, the Imense CBIR search engine (Figure 1.4(b)) takes into account the actual image content (utilizing an auto-annotation mechanism) thereby providing enhanced retrieval accuracy on the same search query.



### 1.3 Problem Identification

This dissertation is concerned with improving the accuracy of automatic image tagging<sup>10</sup>. Generally speaking image tagging is a form of supervised classification of pictorial data. Each image class contains images, which are semantically similar and thus have at least one annotation in common. Furthermore an image usually can be provided with more than one annotation and hence most images can belong to multiple classes. Typically a model is trained on a small set of manually annotated images which can be used to assign an image to one or more classes. The training set provides a unique mapping between a textual annotation and the described semantic entities within the image. Given a novel image, the annotation model compares the visual words with an unknown image, annotating an image with a textual word in the case where the novel image contains the corresponding visual word.

Many different models have been proposed in the literature to learn the dependencies between the visual content of an image dataset and the associated text captions and we will undertake a detailed review of these approaches in Chapter 2. A commonality between most approaches to automatic image tagging is that they tend to predict each candidate keyword for an image independently of other keywords for that particular image. Any correlations that may exist between keywords are generally not taken into account by the majority of models in the literature.

This project will seek to measure the benefits of modifying one particular probabilistic model of image annotation, the Continuous Relevance Model (CRM) of Lavrenko et al. [34] to take account of the dependencies between annotation keywords. This extension has the possibility of increasing annotation accuracy in the event where the extracted image features are not of adequate quality to distinguish between annotation keywords with sufficiently high probability. If we predict a set of keywords together, rather than each keyword independently there is the possibility that some words in the set will boost the probability of correct, but otherwise low probability keywords whilst suppressing the probability of irrelevant but higher probability keywords. For example, consider the annotation keywords “sky” and “ocean”. As both refer to concepts that are some shade of the colour blue, it is difficult to differentiate between either based on extracted colour features. However, if we consider “airplane” and “bird” as part of the annotation set then we can differentiate more easily between these two concepts given that we expect “airplane” and “bird” to be associated more to “sky” than to “ocean”.

The key issue in predicting sets of tags in this manner is the exponential complexity that arises in finding the best (in terms of highest probability) set of keywords for an image. For modest vocabulary sizes, a simple exhaustive search strategy over sets of tags is impossible. In

---

<sup>10</sup>Also referred to as automatic image annotation or image semantic annotation in the literature.

this dissertation we take the novel approach of using a customized beam search algorithm in combination with the amended CRM model to efficiently search over sets of tags in a “greedy” fashion, only adding those keywords that have the best chance of increasing the probability of the entire set of keywords<sup>11</sup>. This amendment has the effect of reducing the exponential complexity to linear in the depth of the search tree, whilst finding a near-optimal set of keywords. We refer to the novel beam search amalgamated algorithm as the “Beam-Search CRM” or BS-CRM model.

## 1.4 Aims and Objectives

The overall goal of this dissertation is to investigate the extent to which predicting tags as sets increases annotation accuracy over automatic tagging methods that treat the tags independently. To maintain a modular structure in the implementation, this objective was refined down into the following four sub-objectives:

1. Extract a discriminative set of image features from the COREL and PASCAL datasets.
2. Implement an efficient version of the original Continuous Relevance Model (CRM) image tagging algorithm [34].
3. Extend the CRM to capture the correlations between keywords.
4. Design an efficient algorithm using Beam Search for searching over sets of tags.
5. Evaluate image tagging accuracy and image retrieval performance on the standard COREL and PASCAL datasets.

## 1.5 Summary of Main Results

All original objectives of the dissertation were completed successfully. The key results are summarised hereunder:

- Custom implemented CRM model performance closely matches that of the results of the original CRM model published in the literature [34].
  - For the COREL dataset:
    - \* **Normalized CRM Model:** Mean per word recall of 0.184 and a mean per word precision of 0.197 with 97 words with recall greater than zero. Ranked retrieval performance over 1, 2, 3 and 4 word queries closely matches that of the original CRM model.

---

<sup>11</sup>Beam search has been applied with notable results to the decoding problem in the field of statistical machine translation [55].

- For the PASCAL dataset:
  - \* Ranked retrieval performance for each class closely matches that of the literature [59].
  - \* **Normalized CRM Model:** Mean per word recall of 0.427 and a mean per word precision of 0.197 with all 20 words having a recall greater than zero.
- Custom implemented CRM model annotates an entire set of 500 images in 0.45 seconds compared to 660 seconds cited by the authors of the original model [12].
- Successful integration of an efficient beam search algorithm (BS-CRM) to search over sets of tags for those maximizing a keyword correlation objective function:
  - For the COREL dataset:
    - \* Over the original CRM model as published by Lavrenko et al. [34] the BS-CRM model achieves a *6.8% increase in mean per word recall and a 31.0% increase in mean per word precision with an increase of 6.5% in the number of words with recall greater than zero.*
    - \* Compared to the keyword correlation model of Zhou et al. [64] the BS-CRM model achieves a *9.1% increase in mean per word recall and an increase of 6.3% increase in mean per word precision.*
    - \* Performance gain is consistent over annotation lengths of 3, 4 and 5 keywords.
    - \* Performance greatly depends on the beam width selected, with widths between 5-15 performing the best on the COREL dataset, with performance declining for wider beam widths. This suggests there is no advantage in expending additional computation effort search over wider beams than around 15.
    - \* Performance is also highly dependent on the word smoothing function that is chosen with the most significant gains being realised for the N-CRM model, lower gains with the Multinomial and Dirichlet models and no significant gains with the Bernoulli word smoothing model.
  - For the PASCAL dataset:
    - \* BS-CRM model achieves a modest 2.4% increase in F1 measure over the CRM model.

## 1.6 Dissertation Structure

This dissertation provides an overview of the work that has been accomplished with respect to the objectives outlined in Section 1.4. The following itemized list provides a chapter by chapter overview of the content and structure of this dissertation:

- **Chapter 2 - Background:** Before undertaking an analysis of the results, we will provide an overview of the previous work in the automated image tagging literature, covering both the main classes of models that attempt to tag images without taking into consideration keyword correlation along with the recent work that has been carried out in augmenting the models to capture the correlation between keywords. We will also examine the theoretical structure of the Continuous Relevance Model and the concept of Beam Search in detail, given that both algorithms underpin the bulk of the work in this dissertation.
- **Chapter 3 - Methodology:** Having provided a suitable background to the field, this Chapter will provide a detailed description of the methodology adopted in developing and adapting the CRM model to capture keyword correlation, including the conceptual design work and the actual implementation of the BS-CRM automatic tagging system.
- **Chapter 4 - Evaluation:** This Chapter will present the results of the BS-CRM system that has been developed as part of this dissertation. To fully grasp the advantages of the BS-CRM model, a thorough analysis of the original CRM performance on the PASCAL and COREL datasets is firstly provided and subsequently used as a benchmark for comparison. Having done this, the BS-CRM is then evaluated using identical evaluation metrics as per the CRM evaluation.
- **Chapter 5 - Conclusions and Future Work:** We conclude by presenting a summary of the main findings and contributions of the work presented in previous Chapters. The Chapter will also include several pointers to possible future work in this particular research area alongside the author's own personal opinion as to the major challenges that still need to be overcome before a fully robust and scalable automatic image annotation system can become a reality.

# Chapter 2

## Background

This Chapter provides an overview of related work in the field of automated image tagging both with regards to those models that do and do not seek to capture correlations amongst image keywords. A particularly detailed examination is given into the properties of the Continuous Relevance model (CRM) in preparation for the discussion of the implementation details of this model and its augmentation with Beam Search to form the BS-CRM model in Chapter 3.

### 2.1 Automatic Image Tagging Challenges

Despite the popularity of automatic image tagging as a research topic and the compelling commercial opportunities for a robust automatic image annotation technology, the field is still very much an open research problem, mainly due to the fact that the analysis and understanding of images in unrestricted domains is an extremely challenging, if not the most difficult problem in the modern field of Computer Vision [53]. The reason as to why this field is particularly challenging is the balance that has to be maintained by any algorithm between two conflicting goals: firstly the image representation chosen has to be very specific so as to be able to correctly differentiate between difficult objects, such as a tiger and a cat. On the other hand, any representation must be invariant to various confounding factors present in images such as occlusions, deformation, scale, background clutter, illumination and view point variations. These latter factors can make the same object look very different between images (see Figure 2.1).

In addition, researchers in the Computer Vision field have also to contend with problems arising from the well known “semantic gap”. According to Smeulders [52] the semantic gap is:

*“...the lack of coincidence between the information that one can extract from the visual data and the interpretation that the same data have for a user in a given situation.” Smeulders [52]*

The semantic gap highlights the wide difference between human interpretation of scene

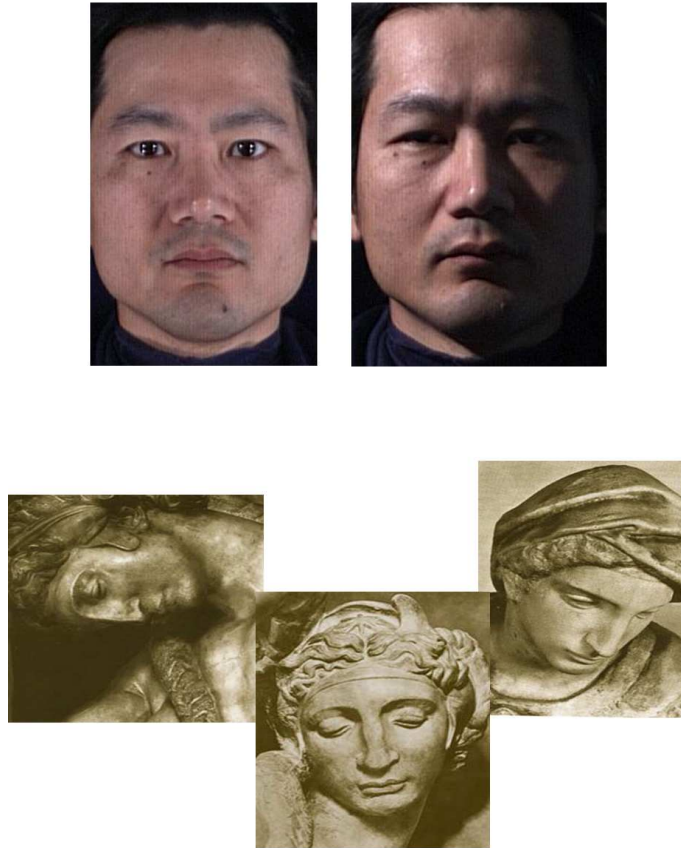


Figure 2.1: *These images illustrate some of the difficulties that are commonly encountered by any algorithm that attempts to understand the content of images. On the left we can see the same statue but at different angles. How is it possible to make an algorithm robust to such wide variations in object appearance? Furthermore, on the right we see an image of the same person but under varying illumination conditions. The person's appearance varies dramatically depending on the lighting conditions. These factors, and many others besides, ensure that image tagging is a non-trivial problem.*

contents and those that are possible through a machine (see Figure 2.2). The raw data obtainable from an image (e.g. pixels, colour histogram) is inherently ambiguous and semantically impoverished. The authors conclude their article by stating:

*“A critical point in the advancement of content-based retrieval is the semantic gap, where the meaning of an image is rarely self-evident. The aim of content-based retrieval systems must be to provide maximum support in bridging the semantic gap between the simplicity of available visual features and the richness of the user semantics.”*

Clearly advancement in the robustness and accuracy of automatic image tagging algorithms has the potential to bridge the semantic gap and therefore improve the performance of content

based image retrieval systems [19]. However, it is worth keeping in mind that the process of moving from this low-level representation to an accurate understanding of the high level concepts present in a scene is at the heart of object recognition in computer vision. Despite nearly five decades of intense research, object recognition is by no means a solved issue and therefore both the variability of object appearance in images and the problem brought about by the semantic gap will continue to ensure that the development of robust and accurate automatic image tagging techniques continue to pose a significant challenge to the research community for the foreseeable future.

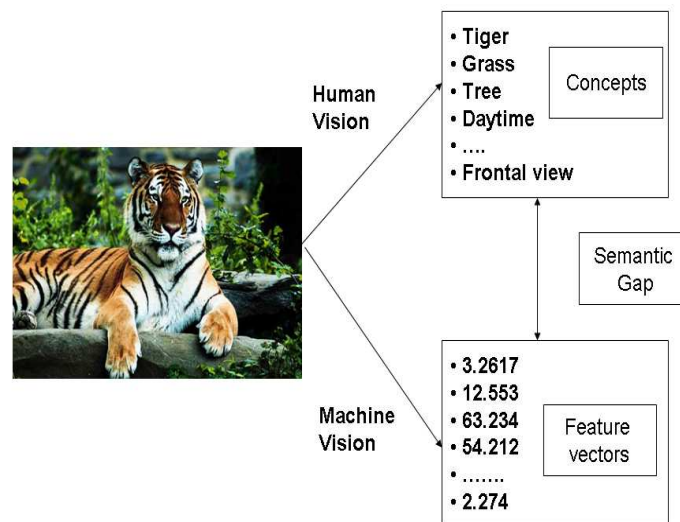


Figure 2.2: An illustration of the semantic gap problem. Here we have an image of a tiger at rest. The concepts likely to be noticed by a human looking at the image are shown in the box at the top right. On the bottom right we have the feature vector representation of the same image, detailing properties such as position, colour, shape and so forth. How does one effectively map this extremely impoverished low level data representation of the image to the high level concepts so easily understood by human beings? This is the crux of the semantic gap issue in Computer Vision.

## 2.2 Existing Image Tagging Models

Nevertheless, despite the difficulties inherent in the understanding of image content, substantial progress has still been realised in the area of automated image tagging over the past few years. Researchers have simplified the problem somewhat by assuming that users can toler-

ate imperfect retrieval results and that there is a much greater leeway for erroneous inferences in automatic image annotation compared to the field of pure object recognition for individual images [18]. It is also not necessary to detect the exact location of concepts of interest in the image, rather computing a likelihood of the presence or not of the concept is usually sufficient for the purposes of tagging.

Given these assumptions, most if not all of the techniques suggested in the research literature tackle the issue by computing low-level image feature distributions for each concept of interest. This essentially boils down to the derivation of a probability table which links annotation keywords to image features. This probability table of associations between features and words can then be used to retrieve high probability keywords for a new feature set derived from an unknown image. Recent results have shown that this approach is indeed viable in improving retrieval results for a number of real-world image retrieval systems [33] [21].

Despite still being in its relative infancy, the automatic image tagging field is extremely large and there exist many different techniques in the literature designed to tackle the problem. Qi et al. [48] and Yavlinksy et al. [62] suggest a useful split of the field according to the feature representation chosen. The authors cite the following two broad categories, dependent on the scale of image analysis:

- Global Feature Based Image Tagging (also known as the Scene-based approach)
- Block/region-based Image Tagging

*Global Feature Based Image Annotation*, utilizes the properties of global image features such as global colour and texture distributions. There are many examples of this approach in the literature. For example, Yavlinksy et al. [62] prepare a vector of real valued image features and a signature of image features to represent each candidate image. A nonparametric density estimator is then employed to differentiate between the annotation classes by exploiting the irregularity inherent in the distributions of image features. Chapelle et al. [9] utilize SVM's on global HSV colour histograms derived from the images of interest, whilst Huang et al. [20] employ a classification tree to model the spatial correlation of colours in the images.

The key idea in this approach is to somehow find a feature representation that is separable enough to allow us to distinguish between different annotation classes but dense enough to develop a model based on a small selection of training images for each class. These approaches provide good results for classifying images when applied to image classes whose discriminative visual properties are spread equally over the whole image surface. The classification of “city” images for instance provides good classification results since city scenes typically show strong vertical and horizontal edges.

When applied to visual object detection, however, global visual features often insufficiently represent the prominent objects that have been used to annotate the images. Hence, the other



*Block Based Image Annotation* branch utilizes an automatic segmentation step before the actual learning stage to identify real-world objects within the images. The general assumption is that feature computation based on a potentially strong segmentation better describes the visual objects, depicted in the image, than global features. Figure 2.3 depicts the end to end flow from training to annotation for a block based annotation algorithm whilst Figure 2.4 demonstrates the processing pipeline of a block-based feature extraction engine. This methodology depends highly on the performance of the selected segmentation algorithm to extract a good selection of coherent objects. At the present time no general and robust automatic segmentation algorithm has been presented, thereby limiting the accuracy of block based algorithms.

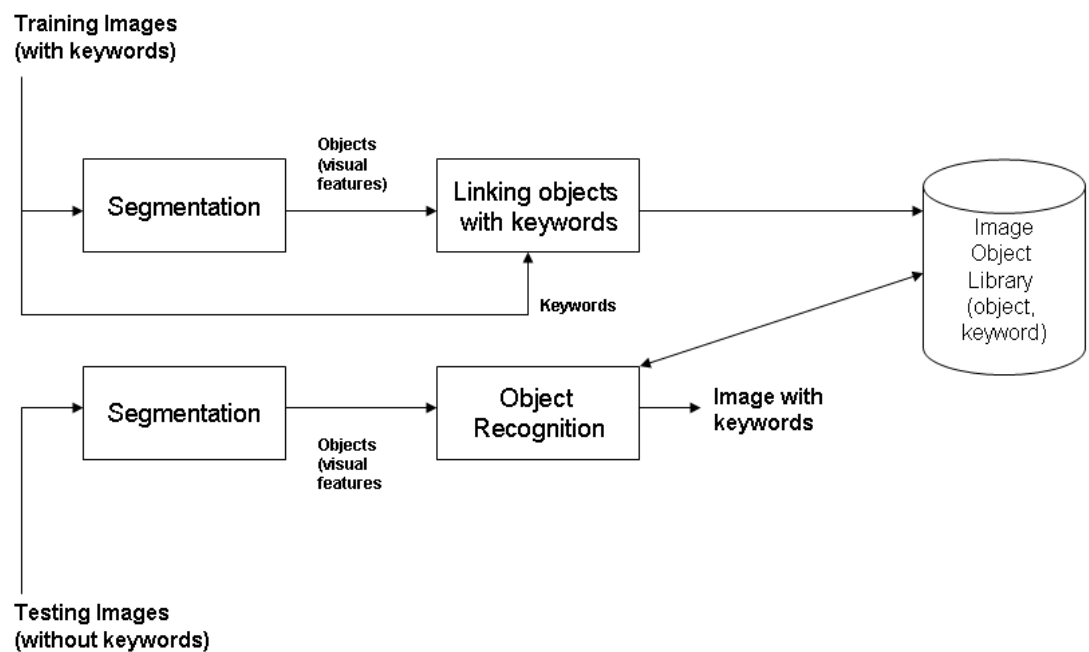


Figure 2.3: A flowchart depicting the stages of model training and image annotation for the block based image annotation approach. The key difference between this approach and the global scene based approach is the segmentation step which divides the images into coherent sub-regions over which feature vectors are computed. Adapted from a similar diagram in the presentation by Lei Wang, Latifur Khan, Bhavani Thuraisingham at the University of Texas at Dallas: [www.utdallas.edu/~yohan/openCVIntro.ppt](http://www.utdallas.edu/~yohan/openCVIntro.ppt)

As for the global feature branch, the literature is also replete with examples of this particular approach to image annotation. The pioneering paper by Mori et al. described how candidate images were divided into a regular grid and a co-occurrence model applied to represent the co-occurrence of words with the image regions [41]. In contrast Duygula et. al. [10], utilize the statistical machine translation model of Brown et al. [6] and apply the EM algorithm to learn a maximum likelihood association of words to image regions using a bi-lingual corpus. A notable feature of this approach is the association of words to actual image regions, in comparison to many other approaches which do not tell us which image structure gave rise to which word. The pre-processed COREL data-set made available by Duygula et al. has become a widely used and popular benchmark of annotation systems in the literature.

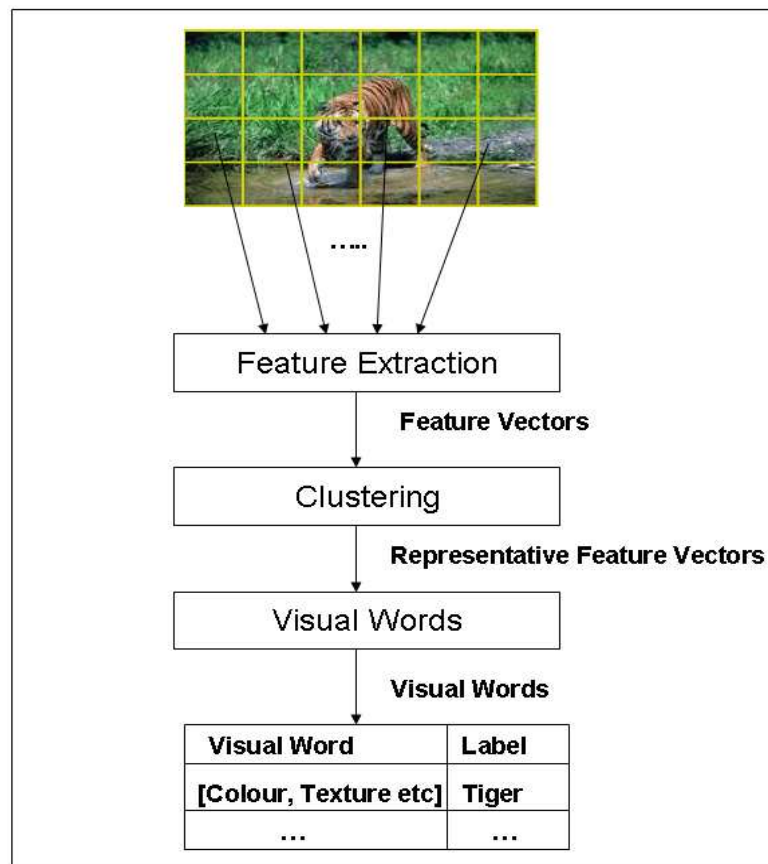


Figure 2.4: Flow chart depicting the major modules and outputs of a typical block based feature extraction pre-processing system. The model extracts features (colour, SIFT descriptors, texture etc) using a rectangular grid and clusters these features into a representative set of visual words to form the codebook.

The most pertinent block-based image tagging model for this dissertation is the Continuous Relevance Model (CRM) of Lavrenko et al. [34]. The CRM has been a highly influential

and often cited automatic tagging model in the literature, mainly due to the excellent results produced by the algorithm on the standard COREL dataset. The model superseded the existing annotation models in the literature by an impressive margin in terms of annotation accuracy. The key feature of this model is that it works with continuous image features directly using non-parametric kernel density estimators therefore avoiding the error prone k-means vector quantization step commonly associated with many of the other block-based algorithms in the literature (see Figure 2.5).





Images				
CMRM Annotation	water sky plane bear	water sky plane jet tree	water sky tree people	people rocks water buildings
CRM Annotation	lizard marine iguana rocks	snow bear polar tundra	train railroad tracks locomotive	cat tiger water forest

Figure 2.5: This diagram illustrates the tags that the CRM has assigned to four example images from the COREL dataset. As can be observed the CRM model, in utilizing continuous features directly, produces more accurate tags compared to the cluster based CMRM. Source: Lavrenko et al., A model for learning the semantics of pictures [34]

As indicated in [34] and in [32], annotation quality is very sensitive to clustering errors and depends on being able to a-priori select the right cluster granularity. Selecting too many clusters results in extreme sparseness of the space, while too few will lead us to confuse different objects in the images<sup>1</sup>. This feature combined with a Dirichlet word smoothing mechanism were the key contributions of this model to the automatic image tagging field. The significant downside of the model however is the inefficiency and high computational load required to calculate the relevant probabilities. We will provide a thorough description of the CRM model in Section 2.4.

Historically the CRM has been the evolution of the earlier image tagging model, the Cross Media Relevance Model (CMRM) also developed by Lavrenko et al. [34]. In contrast to the CRM, this model applied the k-means algorithm to vector quantize or cluster the set of image features to form a visual codebook. Furthermore the model applies a multinomial word smoothing mechanism which has been shown to be inadequate for image tagging and retrieval

<sup>1</sup>On this latter point of too few clusters, Lavrenko in his recent book, “A generative theory of relevance” [32], cites the example of finding images in a database consisting of a few technical diagram images and many animal images. The technical diagram images are likely to be collapsed into the closest animal cluster, completely wiping out the chance of the user finding these rare technical images in the collection.

given that many datasets have widely varying annotation lengths per image. A multinomial smoothing model focuses on the *prominence* of words rather than the *presence* of words in the annotation, effectively splitting the word probability mass of between multiple words in the annotation. Therefore in the application of image retrieval an image annotated with “person, tree” would be given lower preference to an image annotated with “person” as the first image will have a probability of  $\frac{1}{2}$  for person with a probability of 1 for person in the second image. This is clearly undesirable.

To overcome this issue, Lavrenko et al. introduced their best performing relevance model to date which has been shown to surpass the performance of the CRM on the task of image tagging and retrieval [13]. The Multiple-Bernoulli Relevance Model (MBRM) introduced two key advancements into the field. Firstly, the authors replace the multinomial smoothing model of the CMRM with a Bernoulli word smoothing model. In addition, rather than applying the commonly used Normalized Cuts [51] segmentation algorithm to segment the images into coherent sub-regions, the authors partition each image into a regular grid and compute continuous image features over these regions. This latter technique avoids the computational expense of a dedicated image segmentation algorithm and provides the model with a larger set of image regions for learning the association between regions and words. The authors report an increase in performance by 38% over the CRM. In their subsequent paper [33] the authors demonstrate how the CRM model can be amended with a modified Dirichlet word smoothing distribution (the so called Normalized CRM or N-CRM model) to capture most of the advantages of the MBRM model. Chapter 3 we will discuss how these lessons from the MBRM have been taken into account during the development of the custom built CRM model for the purposes of this dissertation.

Other important developments in the block-based branch include the model of Ghoshal et al. [18] who brought the brunt of the elegant mathematical formalism of Hidden Markov Model’s (HMM) [49] to bear on the problem, by positing that image feature vectors describing low level image content can be stochastically generated by a HMM, the states of which represent the keywords of interest. A multitude of other statistical techniques have also been applied to this problem, including Latent Dirichlet Allocation (LDA) [3], Probabilistic Latent Semantic Analysis (p-LSA) models [14] [40] and Maximum Entropy [24].

## 2.3 Capturing keyword correlation

All of the aforementioned approaches predict image keywords independently. Recently, researchers have turned to the question of how best to capture correlations between keywords to enhance the performance of the annotation models. Figure 2.6 illustrates the benefits that can arise in taking into consideration keyword correlation in the process of image tagging. Given

that we are essentially aiming to select the “best” set of keywords that are most correlated with each other for a particular image, the question naturally arises on how one can find this best keyword set out of the word vocabulary. The complexity of adding an optimal (highest probability) sets of tags out of a vocabulary of words grows exponentially with the size of the vocabulary, and therefore for the modest sized vocabularies in the literature an exhaustive search over all possible keyword subsets is not possible<sup>2</sup>.




Images			
Image ID	108019	109012	163068
CMRM Annotations	grass, albatross, wings, cat	water, sky, tree, people	tree, water, sky, grass
CIAR Annotations ( $N=10, m=4$ )	grass, cat, tiger, forest	tree, water, people, snow	birds, tree, grass, water

Figure 2.6: This diagram from the paper by Wang et al. [58] illustrates the increase in annotation accuracy that can be brought about by taking into consideration the correlation between the keywords for an image. For example, consider the image on the far left with the tiger - here we can see that the CMRM annotation contains two noisy keywords “albatross” and “wings”. Using keyword correlation the CIAR model is able to eliminate these keywords and correctly determines that “tiger, forest” best correlates with “cat, grass”. Source: Wang et al., Content based Image Annotation Refinement [58]

Zhou et al. [64] overcome the exponential complexity by proposing a heuristic greedy iterative algorithm to estimate the keyword subset for a particular image which is found to significantly improve the performance of a state of the art image annotation algorithm. The authors essentially amend the CMRM with an objective function to determine the keyword that brings about the maximum gain of probability to the existing keyword subset. Nevertheless their approach has a number of notable flaws. Firstly the objective function itself is deficient in that the probability of a quantity arrived at by the model in two or more different ways may not always be equal. Furthermore the function considers the word co-occurrence and word-image co-occurrence probabilities separately<sup>3</sup> which further reduces the discriminative power of the function.

<sup>2</sup>Take the commonly used COREL image dataset as an example. This dataset has a vocabulary size of 371 words, giving around 60 billion 5 word subsets. To find a particular set from these 60 billion sets would take on the order of around 10 million years, assuming we could check a subset of keywords against our objective function every 1 second.

<sup>3</sup>The authors assume that the current predicted subset of keywords and the image features are conditionally independent given the current word - a blatantly false assumption.

Wang et al. [2] improve on this approach in their “Progressive Image Annotation” model by applying a more powerful objective function in the form of the CRM to capture keyword correlation of words. As for Zhou et al., the authors attempt to overcome the exponential complexity of finding the best subset of keyword in a vocabulary by adding words to an existing set which lead to the greatest increase in the objective function. The suggested method involves, for each testing image, adding successive words to the image annotation based on the joint probability of words already in the annotation. Essentially the authors are attempting to compute the “next best” word to add to the image annotation at each stage. The authors of the paper demonstrate that amending the CRM in this manner can effectively improve the annotation performance.

Extending this approach a step further, Zhu et al. demonstrate that it is possible to eliminate noisy keywords by reformulating the problem as one of graph ranking using the random walk with restarts algorithm [56]. Having calculated an initial set of keywords using the amended keyword correlation CRM model of Wang et al., the authors then build a graph with nodes representing the candidate annotations and weights linking nodes reflecting the similarity between the nodes of the graph. The random walk with restarts algorithm is then applied to this fully connected graph to re-rank the candidate annotations, of which the top few words are chosen as the final annotation for an image.

The content-based image annotation refinement algorithm of Wang et al. [58] is yet another example of an approach that uses a relevance model, in this case the CMRM for image annotation refinement. Here the authors re-frame the annotation refinement problem as a Markov process and define the candidate annotations as the states of a Markov chain. The algorithm takes into consideration both corpus information and the image features during the refinement process leading to notable results on the standard COREL dataset.

Other interesting non-relevance based approaches have also been suggested in the literature. For example, in their paper Kang et al. [30] proposed a correlated label propagation algorithm for multi-label learning that explicitly models interactions between labels efficiently. In this approach the authors use properties of sub-modular functions to develop a model that simultaneously co-propagates multiple labels attached to training data to the test data. This is in contrast to standard label propagation which propagates labels individually from training to test set.

In their paper, Wang et al. [60] present a probabilistic approach using a relevance vector machine (RVM) to refine image annotations by incorporating semantic relations between annotation words. The authors model semantic relationships between words using a conditional random field (CRF) model where each vertex indicates the final decision (true / false) on a candidate annotation word. The refined annotation is obtained by using the model to infer the most likely states of the vertexes. In this approach the confidence scores given by the RVM classifiers are used as local evidence with the Normalized Google distances (NGD's) between

two words taking into consideration their contextual relationship. The authors obtain excellent results on the standard COREL dataset.

In contrast, Jin et al. [25] apply the EM algorithm to a coherent language model (CLM) in order to generate a subset of annotation keywords. Besides capturing keyword correlation, the authors also suggest a means to automatically determine annotation length and apply an active learning technique to reduce the number of labelled training images required for the model. Naphade et al. [43] suggested a graphical modelling approach to multimedia indexing that captured the correlation between different concepts whilst Qi et al. [47] tackled the problem in the case of video annotation by using a correlative multi-label (CML) annotation framework which simultaneously classified concepts and modelled their respective correlations in a single step.

Other authors make use of the large lexical WordNet database to prune noisy keywords [39]. Jin et al. [26] discard an annotated keyword from an image that does not correlate well with other annotated keywords that appear in the image by applying WordNet coupled with multiple evidence fusion based on Dempster-Shafer evidence combination. However, it has been shown that this approach, whilst removing noisy keywords, also has the undesirable effect of removing many relevant keywords as well leading to a decrease in the F1 measure. Srikanth et al. [54] investigated the extent to which a hierarchy created based on the annotation words derived from WordNet could be applied to capture keyword correlations. In a similar vein, Liu et al. [35] proposed a novel automatic image annotation method that utilized WordNet to obtain the word-to-word correlations to prune irrelevant annotations for each image. By conducting experiments on the standard COREL dataset and a web image dataset the authors were able to demonstrate the effectiveness and efficiency of their proposed method for image annotation.

## 2.4 The Continuous Relevance Model (CRM)

### 2.4.1 Overview

Having reviewed the relevant literature in the field of automatic image tagging, a more detailed exposition will now be provided into one particular annotation model, the Continuous Relevance Model (CRM) of Lavrenko et al. [34]. As mentioned in Chapter 1, in this dissertation we will aim to develop the original CRM model from first principles and then augment the model to capture keyword correlation efficiently through application of the Beam Search algorithm. In this description we roughly follow the approach given in the original paper [34] but simplify the explanation of the model without any loss of generality<sup>4</sup>.

In commonality with many authors, Lavrenko et al. take inspiration from the field of

---

<sup>4</sup>An alternative presentation of the model, more biased towards the application of image retrieval, is given in the book, “*A Generative Theory of Relevance*” [32].

text information retrieval, and develop the CRM, a statistical *generative language model* with respect to images which is similar to the relevance language models of text information retrieval [46]. The CRM attempts to estimate the joint probability distribution of a set of words  $\mathbf{w}_I = \{w_1 \dots w_k\}$  together with an image  $I$  represented by a set of image features denoted by  $\mathbf{f}_I = \{f_1 \dots f_m\}$ <sup>5</sup>. The modelling of the joint distribution  $P(\mathbf{w}_I, \mathbf{f}_I)$  of words and image regions in this manner is key to the model and gives it the ability to both annotate images and to perform image retrieval:

- **Image Annotation:** Knowing the image features  $\mathbf{f}$  we can use the joint distribution to arrive at the  $P(\mathbf{w}|\mathbf{f}_I)$ . Ranking the keywords in the vocabulary by their conditional probability given the image features and taking the top 3-5 words will allow us to annotate a novel image based on its contents.
- **Image Retrieval:** Given a query  $\mathbf{w}_{\text{qry}}$  consisting of one or more keywords, we can utilize the joint probability distribution to arrive at  $P(\mathbf{w}_{\text{qry}}|\mathbf{f}_I)$  for every testing image  $I$  of interest. Ranking the images by these probabilities will give a list of images sorted by relevance to the user query.

### 2.4.2 Image representation

Given the high dimensionality of the raw image regions  $\{r_1 \dots r_n\}$ , it is necessary to firstly distil the regions down into a lower-dimensional set of discriminative image feature vectors for use with the CRM. We will assume that we have an algorithm in place for computing feature vectors  $\{f_1 \dots f_{nT}\}$  for every region  $\{r_1 \dots r_n\}$ . The feature vectors typically consist of colour, texture, shape and position information for each extracted region although SIFT [36] features are rising in prominence due to their desirable qualities of invariance to object scale and rotation.

### 2.4.3 Annotation Model

Having pre-processed the images in the aforementioned manner, the question now turns to how we can effectively link both the words and images to perform the automatic tagging task. In the image tagging field we are given a collection  $\mathcal{C}$  of testing images with no associated annotations, alongside a typically larger training collection of images  $\mathcal{C}_{\text{train}}$  with associated manually provided tags per image. The basic goal of the CRM model is to annotate the unseen test images using the joint probability distribution learned from the training image dataset. Section 2.4.3.1 describes how the CRM formulates this joint distribution.

---

<sup>5</sup>Compared to the Translation Model [10], the authors here do not isolate particular image features as being associated to particular keywords, so it is not possible to tell which feature gave rise to a keyword using this methodology.



### 2.4.3.1 Joint distribution of Images and Words

To construct this joint distribution of words and images, the model makes the assumption that the training images  $J$  consisting of features  $\{f_1 \dots f_{nT}\}$  and tags  $\{w_1 \dots w_l\}$  are generated by two underlying probability distributions for the words and image features:

- **Word distribution,  $P_{\mathcal{V}}(\cdot|J)$ :** This is modelled as a multinomial distribution in the original formulation, although there exists many other maximum likelihood smoothing functions for words including Bernoulli and Dirichlet smoothing. We will touch upon these latter representations in Section 2.4.3.4.
- **Image feature distribution,  $P_{\mathcal{F}}(f_i|f_{jT})$ :** This distribution captures the similarity between the testing  $f_i$  and training  $f_{jT}$  image feature vectors respectively.

Having defined these distributions we are now in a position to perform image tagging on the unseen test images. As the CRM is a generative model we can best think of this process as one of *generating* the unseen test image features  $\mathbf{f}_I = f_1, \dots, f_m$  and associated tags  $\mathbf{w}_I = w_1, \dots, w_k$  by a process of sampling using the aforementioned word and image feature vector distributions respectively<sup>6</sup>.

Intuitively, for a candidate test image  $I$ , the relevance model (denoted by  $P(\cdot|I)$ ) assumes that the words and associated features are contained in a hypothetical urn and that we can effectively generate the features  $\mathbf{f}_I = f_1, \dots, f_m$  making up the image by taking random samples from this urn. Therefore to annotate an image with keywords we need to find a method to sample words from this urn.

Unfortunately as the form of  $P(\cdot|I)$  is unknown, this is not directly possible, however an estimate can be made by using the training set of images and computing a joint probability of the words  $\mathbf{w}_I = w_1, \dots, w_k$  and the image features  $\mathbf{f}_I = f_1, \dots, f_m$ . Assuming  $P(w|I) \approx P(w|f_1, \dots, f_m)$ , then, the joint distribution can be estimated as:

$$P(w, f_1, \dots, f_m) = \sum_{j \in T} P(J) P(w, f_1, \dots, f_m | J)$$

Where we are taking the expectation over all the images in the training set,  $T$ . The urn model lets us assume that the two events  $w$  and  $f_1, \dots, f_m$  are mutually independent<sup>7</sup>, which permits the equation to be further decomposed in the following manner:

$$P(w, f_1, \dots, f_m) = \sum_{j \in T} P_T(J) P_{\mathcal{V}}(w|J) \prod_{i=1}^m P_{\mathcal{F}}(f_i|J) \quad (2.1)$$

<sup>6</sup>Note the number of words and image features ( $k$  and  $m$ ) may be different between images.

<sup>7</sup>In fact *exchangeability* or order invariance is a sufficient but weaker criterion for the CRM than mutual independence and allows us to decompose the formula in a similar way. This gives extra power to the use of the CRM model as the objective function in our beam search algorithm described in Section 2.6.

This is the central equation of the CRM model. By normalizing this equation as follows we are able to calculate the required conditional probability of a word given a set of image features  $P(w|f_1, \dots, f_m)$ :

$$P(w|f_1, \dots, f_m) = \frac{\sum_{J \in T} P_{\mathcal{V}}(w|J) \prod_{i=1}^m P_{\mathcal{F}}(f_i|J)}{\sum_{J \in T} \prod_{i=1}^m P_{\mathcal{F}}(f_i|J)} \quad (2.2)$$

The parameters and distributions of Equation 2.1 are now discussed in some detail in Sections 2.4.3.2, 2.4.3.3 and 2.4.3.4.

#### 2.4.3.2 Uniform Prior Distribution

The probability of selecting a training image or  $P_T(J)$  is modelled as a uniform prior over the training image dataset i.e.  $P_T(J) = \frac{1}{N_T}$ , where  $N_T$  is the size of the training set. As the prior distribution  $P_T(J)$  is uniform and constant across all images and appears in the numerator and denominator of Equation 2.2 it therefore cancels out upon division and does not need to be considered any further.

#### 2.4.3.3 Non-parametric Kernel density Estimators

The image feature distribution  $P_{\mathcal{F}}(f_i|J)$  is modelled by non-parametric kernel density estimators of the following form:

$$P_{\mathcal{F}}(f_i|J) = \frac{1}{n} \sum_{j=1}^n \frac{1}{\sqrt{2^k \pi^k |\Sigma|}} \exp \left\{ -\frac{\|f_i - f_{jT}\|^2}{\beta} \right\} \quad (2.3)$$

Essentially we are placing a Gaussian kernel over every feature vector of every training image  $J$ . In this equation  $k$  denotes the dimensionality of the image feature vectors  $f_i$  and  $\beta$  is the kernel bandwidth parameter of the model that is optimized on a held out validation set.

#### 2.4.3.4 Maximum Likelihood Word Smoothing Functions

In the original specification of the model, the word smoothing distribution  $P_{\mathcal{V}}(\cdot|J)$  that describes the annotations of the training images  $J$  is derived under a Bayesian framework. Specifically the authors place a Dirichlet prior over the simplex of multinomial distributions  $\phi^{\mathcal{V}}$  over the vocabulary  $\mathcal{V}$  with parameters  $\mu p_v$  with  $v \in \mathcal{V}$ . As with  $\beta$ ,  $\mu$  is another parameter that is selected based on a held out validation set. Given the observation  $\mathbf{w}_J$ , a Dirichlet posterior results over  $\phi^{\mathcal{V}}$  with parameters  $\mu p_v + N_{v,J}$ , with  $N_{v,J}$  being the number of times the keyword  $v$  appears in the annotation  $w_J$  of training image  $J$ . Taking the expectation of this Dirichlet posterior leads to the following equation for  $P_{\mathcal{V}}(\cdot|J)$ :

$$P_{qv}(\cdot|J) = \frac{\mu p_v + N_{v,J}}{\mu + \sum_{v'} N_{v',J}} \quad (2.4)$$

As with all of the smoothing functions discussed in this Section, the function in Equation 2.4 essentially mixes the observed word frequencies of the training set for a particular image with the global word frequencies across the entire training set. This has the effect of “smoothing over” and zero probabilities, which are generally an issue in the image tagging literature given the relatively small dataset and vocabulary sizes that are made available. In Equation 2.4 the value of  $\mu$  determines the degree of interpolation, with a larger value of  $\mu$  given more precedent to the background probability over the image annotation word frequencies.

In addition to this Bayesian estimator, there are several other maximum likelihood word smoothing functions worth considering, including Multinomial, Bernoulli and the so-called “Normalized CRM” function. In this dissertation we will investigate all four of these distributions given that the type of smoothing applied to the words in the vocabulary is likely to have a significant impact on the performance of the keyword correlation mechanism.

The Normalized CRM (or N-CRM) model was a refinement of the original CRM model introduced by Lavrenko et al. [33] to overcome the limitations of the Dirichlet smoothing function when annotation length varies widely:

$$P_{qv}(\cdot|J) = \frac{N_{v,J} + P_v(\mu - \sum_{v'} N_{v',J})}{\mu} \quad (2.5)$$

Here we essentially eliminate the spread of the probability mass across annotation words by removing the expression  $\sum_{v'} N_{v',J}$  from the denominator of the equation. It is important to note that the parameter  $\mu$  can take on any value greater than or equal to the length of the image annotation. As touched on earlier during the literature review (see Section 2.2), the multinomial Dirichlet formulation spreads the probability between words implying that that annotations focus on prominence rather than presence of objects. So the longer the annotation length, the lower the probability of the word for that image. As discussed in the literature review, using the N-CRM model has been found to reap most of the benefits of the higher performance MBRM model.

Aside from the Bayesian formulations of the Dirichlet and Normalized-CRM smoothing functions, we also have the basic multinomial and Bernoulli smoothing distributions. The multinomial smoothing distribution is defined as:

$$P_{qv}(\cdot|J) = \lambda \frac{N_{v,J}}{N_J} + (1 - \lambda) \frac{N_v}{N} \quad (2.6)$$

In this equation,  $N_{v,J}$  is the number of times  $v$  occurred in the annotation of image  $J$ ,  $N_J$  is the length of the annotation,  $N_v$  is the total number of times  $v$  occurred in the training set and  $N$  is the aggregate length of all training annotations. As per  $\mu$  in the Dirichlet function, here  $\lambda$

is the parameter that controls the degree of smoothing between the global training set and local image annotation word frequencies.

In contrast, the Bernoulli smoothing distribution is given by:

$$P_{\nu}(\cdot|J) = \lambda 1_{v \in J} + (1 - \lambda) \frac{\sum_J 1_{v \in J}}{\sum_J 1} \quad (2.7)$$

The Bernoulli and Multinomial smoothing distributions as presented, both model completely different events. For the multinomial distribution the event space is the set of all words in the vocabulary, with the probability being that of a random word from  $J$  being word  $w$ . For the Bernoulli model on the other hand, the event space is the set  $\{0, 1\}$  and the probability being modelling is the probability of the presence of absence of the word  $w$ .

For all of the aforementioned smoothing functions, the absolute value of the smoothing parameters  $\mu$  and  $\lambda$  generally do not matter too much for one word queries. However, when one comes to consider multiple word queries and for capturing word correlation the setting of these smoothing parameters will be absolutely critical to performance.

#### 2.4.4 Image Annotation and Retrieval

Having formulated this joint distribution in Equation 2.1 we are able to annotate an unknown image,  $I$ , by extracting feature vectors from the image and computing  $P(w|\mathbf{f}_I)$ . Ordering the resultant probabilities in terms of decreasing value, we typically select the 3-5 words with the highest probability as the annotation keywords for the candidate image.

We are also in a position to use the joint distribution to assign probabilities to multi-word queries, allowing us to retrieve unlabelled images given a query  $q_1 \dots q_k$ :

$$P(q_1 \dots q_k | f_1, \dots, f_m) = \prod_{j=1}^k P(q_j | f_1, \dots, f_m) \quad (2.8)$$

The important point to note in Equation 2.8 is that we are assuming that the keywords are *conditionally independent* given the current test image, which is a blatantly false assumption as has been discussed in detail in Chapter 1. This dissertation centres on relaxing this rather dubious assumption and amending the CRM model to take into account the correlation between keywords. The manner in which this is achieved for the model is postponed until Chapter 3.

## 2.5 Evaluating Image Tagging Performance

The question naturally arises as to how one can best evaluate the performance of the plethora of widely varying image tagging models. A common theme to many of the approaches in the literature is the use of evaluation metrics borrowed from the field of text information retrieval

such as recall and precision. Variations on these metrics adapted for specific use in the image annotation field were popularised by Duygulu et al. [10] in their seminal paper on the translation model. Since the publication of this paper many subsequent authors have followed a similar evaluation methodology both using the same dataset and the same metrics. As the use of these metrics ensures that different approaches can be compared in a strictly controlled manner the decision was therefore taken to adopt the evaluation approach of Duygulu et al. in this dissertation.

When evaluating image tagging models we need to take into consideration both *annotation* and *retrieval* performance. Both approaches differ in their unit of evaluation: for annotation, we take the unit of measurement as an image and seek to calculate the proportion of images that have been correctly annotated with a given word for all words in the vocabulary. In contrast, for retrieval we take a word as the basic unit of evaluation, and seek to rank all of the images by their probability of containing this word, and applying evaluation measures that take into account the position of the relevant images in the ranked list. Whilst retrieval performance takes into account all of the images for a given word, annotation performance only considers the top e.g. 5 words for each image. Both of these evaluation approaches will now be discussed in some detail.

### 2.5.1 Annotation Performance

As touched upon in our discussion of the CRM model, we annotate a novel test image  $J$  with automatically generated keywords  $\mathbf{w}_{\text{auto}}$  and compare these annotations to the ground truth annotation  $\mathbf{w}_J$  for the image. In this dissertation we will follow, unchanged, the annotation evaluation methodology of Lavrenko et al. [34] in their original CRM paper. Given pre-segmented image regions  $\mathbf{r}_J$ , Equation 2.4 is used to calculate the  $P(w|\mathbf{r}_J)$ . The top e.g. 5 words are then taken from this distribution and used to annotate the test image  $J$ .

- **Word Recall:** is the number of images correctly annotated with a given word, divided by the number of images that have that word in the human annotation. This metric measures the completeness in annotating images with word  $w$ :

$$p_w = \frac{c_w}{e_w} \quad (2.9)$$

Here  $c_w$  is the number of correct images annotated with word  $w$  and  $e_w$  the number of images annotated with  $w$  in the ground truth.

- **Word Precision:** is the number of correctly annotated images divided by the total number of images annotated with that particular word (correctly or not). This metric measures the accuracy in annotating images with word  $w$ .

$$p_w = \frac{c_w}{r_w} \quad (2.10)$$

Here  $r_w$  is the number of images the system has annotated with word  $w$ .

- **Number of words with recall greater than zero:** This metric seeks to measure the ability of the system to label images with rare keywords which are hard to annotate due to the small number of positive instances in the training set. This metric is also important as it is possible to achieve high precision and recall values by performing very well on a small selection of common words.

Typically the single word recall and precision are averaged over all words that exist in the testing dataset. The mean per word precision is given by:

$$\bar{P}_n = \frac{1}{n} \sum_{w=1}^n p_w \quad (2.11)$$

With the mean per word recall expressed as:

$$\bar{R}_n = \frac{1}{n} \sum_{w=1}^n r_w \quad (2.12)$$

For the COREL dataset, it is also quite popular in the literature [10] to calculate the mean per word precision of the top 49 words (those with precision greater than 0.15) and the mean per word recall of the top 49 words (those with recall greater than 0.4).

The recall and precision values can be combined into one metric of performance referred to as the F1 measure, which is the harmonic mean of precision and recall that penalizes very low values of either quantity:

$$F = \frac{2pr}{p+r} \quad (2.13)$$

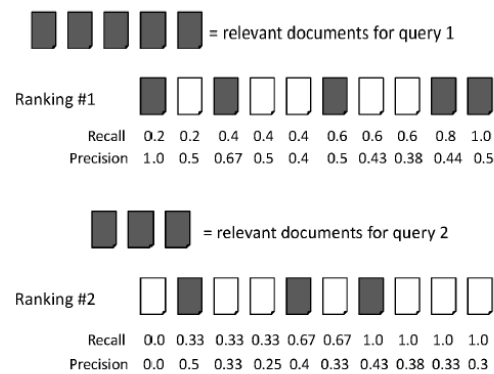
As before, the F1 measure can be calculated on a per word basis or averaged across the entire set of words in the vocabulary. Computing these metrics for the novel approach suggested in this dissertation will be sufficient to compare the algorithm with the state-of-the-art image tagging models in the literature.

### 2.5.2 Retrieval Performance

Unlike the annotation performance, for retrieval performance we seek to specifically take into account the ranking of the images that the system has specified are relevant to both a single query and typically multi-word queries, consisting of 2, 3 and 4 words.

Many authors [34] in the image tagging literature choose to measure the Mean Average Precision (MAP) and Precision at 5 (P@5) for the ranked list of images<sup>8</sup>.

- **Mean Average Precision (MAP):** or *non-interpolated average precision*, is the mean of the average precision (AP) for each query, where the average precision for a query is calculated as the average of the precision values where the relevant images occur in the ranked list (see Figure 2.7). The average precision (AP) gives an indication for the retrieval quality for one topic and the mean average precision (MAP) provides a single-figure measure of quality across recall levels averaged over all queries.
- **Precision at 5 (P@5):** This metric measures the precision of the system at 5 retrieved images.



$$\text{average precision query 1} = (1.0 + 0.67 + 0.5 + 0.44 + 0.5)/5 = 0.62$$

$$\text{average precision query 2} = (0.5 + 0.4 + 0.43)/3 = 0.44$$

$$\text{mean average precision} = (0.62 + 0.44)/2 = 0.53$$

Figure 2.7: This diagram illustrates the calculation of the Mean Average Precision metric for a toy example. Essentially we record the precision values at those points in the ranking where the recall increases and take the average of this value per query. This is the average precision per query. We then average the per query average precision across all queries to derive the value of the Mean Average Precision. . Source: Edinburgh School of Informatics, Text Technologies Lecture Notes 2008/2009: <http://www.inf.ed.ac.uk/teaching/courses/tts/pdf/eval-2x2.pdf>

These metrics essentially measure the needs of two different types of potential users of image retrieval systems: professional users (MAP) who seek to find a large number of relevant items, and casual users (P@5) who only wish to obtain a small number of relevant items without

<sup>8</sup>Note these metrics still make sense in the annotation paradigm, given that we can rank the annotation labels in order of probability for any given image.

viewing too many irrelevant items in between. Both of these metrics produce a single number to measure the performance. In addition to measuring the precision and recall at fixed ranks, it is also very useful to construct a graph (recall-precision plot) detailing how precision and recall vary as we increase the recall.

In this dissertation we will evaluate the original (non beam search) algorithm on the multi-word retrieval performance only so as to determine whether or not the basic CRM algorithm custom implemented for this dissertation matches the results as published in the original paper [34]. This accords with the majority of the existing work in the literature on capturing keyword correlation for image tagging who focus solely on the annotation performance of the algorithm.

## 2.6 Reducing computational complexity through Beam Search

Having so far discussed the current image tagging models, we will now conclude our foray into the background of the image tagging field by taking a brief detour into the arena of combinatorial optimisation, where we will consider the properties of the beam search algorithm. As discussed in Chapter 1 a key research idea of this dissertation is in the application of beam search to efficiently search of sets of tags to find the set that have the (close to) highest mutual correlation for the test image of interest.

A useful and succinct summary of the Beam Search algorithm has been provided by Bisiani [4]. Bisiani states that Beam Search is any search algorithm..

*“...in which a number of [...] alternatives (the beam) are examined in parallel. [It] is a heuristic technique because heuristic rules are used to discard [prune] non-promising alternatives in order to keep the size of the beam as small as possible.”*

The essential reason for wishing to perform beam search is to overcome the excessive memory requirements of best-first search whilst still obtaining a near to optimal solution. In this algorithm only the most promising nodes at each level of the search graph are selected for further branching, and the remaining nodes are pruned off permanently. Since its inception beam search has found many practical applications particularly with regards to problems requiring combinatorial optimisation such as Speech Recognition [44], Job Scheduling [57] and Image Understanding [50].

Figure 2.8 illustrates the operation of Beam Search on an example problem. The standard version of beam search expands nodes in breadth-first order. In each layer of a breadth-first search graph, it expands only the  $B$  most promising nodes, and discards the rest, where the integer  $B$  is called the *beam width*. A heuristic is used to select the most promising nodes. By bounding the width, the complexity of the search becomes linear in the depth of the search instead of exponential; the time and memory complexity of beam search is  $wd$ , where  $d$  is the



depth of the search<sup>9</sup>. Pseudo-code describing the original beam search algorithm is presented in Algorithm 1.

```

input :  $k_{bw}, s_{bsf}$ 
output:  $\arg\max \{|s|s \in B\}$ 

1 let  $B = \{\epsilon\}$ ;
2 while  $B \neq \emptyset$  do
3   let  $C = \text{CHILDREN\_OF}(B)$ ;
4   let  $B = \emptyset$ ;
5   while  $C \neq \emptyset$  do
6     let  $s^t = \text{GET\_PARTIAL\_SOLUTION}(C)$ ;
7     if  $\text{HEURISTIC}(s^t) \leq |s_{bsf}|$  then
8        $B = B \cup \{s^t\}$ 
9     end
10    let  $C = C \setminus \{s^t\}$ 
11  end
12  let  $B = \text{REDUCE}(B, k_{bw})$ 
13 end

```

**Algorithm 1:** *The original Beam Search algorithm. The algorithm maintains a set of  $B$  partial solutions. At the start  $B$  only contains the empty partial solution  $\epsilon$ . The set  $C$  contains all of the children of the partial solutions in  $B$ . Each partial solution is then retrieved from  $C$  and evaluated using a heuristic evaluation function  $\text{HEURISTIC}$ . If the value is lower than a threshold then the partial solution is discarded. If the value is higher the partial solution is appended to  $B$ . After evaluating all of the partial solutions the solutions in  $B$  are reduced by the function  $\text{REDUCE}$  if  $B$  contains more than  $k_{bw}$  (beam width) partial solutions.  $\text{REDUCE}$  could simply sort the values in  $B$  by heuristic value and take the top  $k_{bw}$  solutions to expand at the next iteration.*

This reduction in computation and memory comes at a cost, in this case, the algorithm is not guaranteed to find an optimal solution and cannot recover from wrong decisions. That is to say, if a node leading to the optimal solution is discarded during the search, there is no longer any way to reach that optimal solution<sup>10</sup>. Varying the beam width parameter  $B$  trades off the risk of missing optimal goal states against the computational cost of the search - a wider beam considers more hypotheses concurrently, whilst taking up more memory and processing power,

<sup>9</sup>Relating this to Image Tagging, in terms of the vocabulary size  $v$ , the complexity of the greedy beam search algorithm is  $dvw$ , whereas the non-greedy search is of complexity  $v^d$ . A substantial improvement.

<sup>10</sup>It is worth noting that some authors have since amended the original beam search algorithm with back tracking to essentially allow the algorithm to investigate previously pruned paths. A good example is the Beam Stack Search algorithm of Zhou et al. [63].

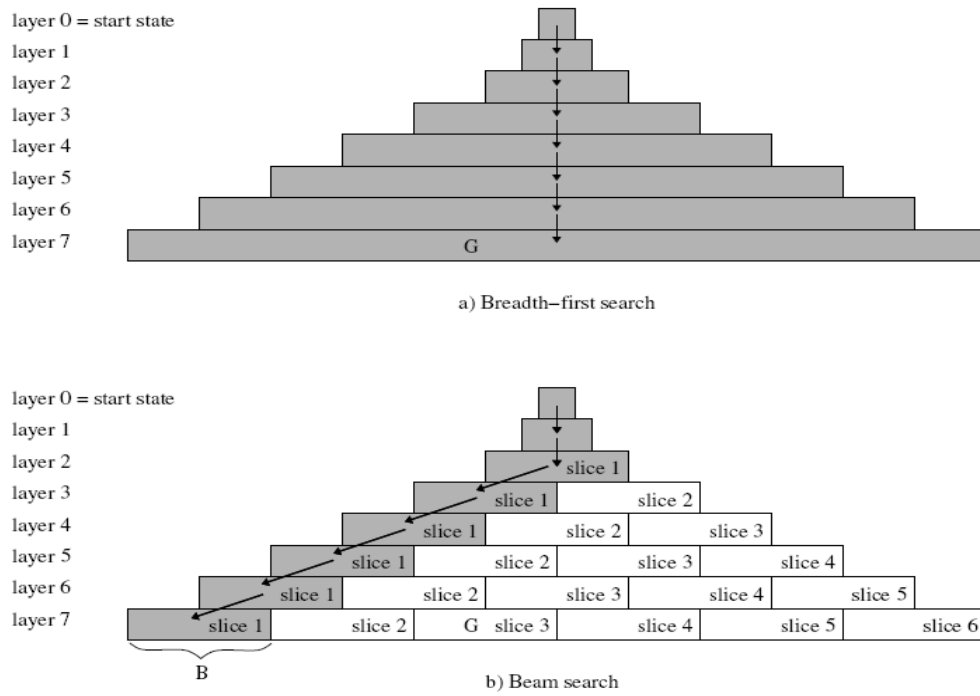


Figure 2.8: This diagram compares Breadth First Search with Beam Search. Essentially Beam Search narrows the width of the breadth first search using a parameter  $B$ , the width of the beam which is the maximum number of states at each level. At each level beam search generates all successors of the states at the current level, sorts them in order of increasing heuristic values, splits them into slices of at most  $B$  states each, and then extends the beam by storing the first slice only. Beam search terminates when it generates a goal state or runs out of memory. Source: Furcy et al., Limited Discrepancy Beam Search [16]

and vice-versa for lower beam widths.

# Chapter 3

## Methodology

This Chapter will describe the implementation of the CRM model and its adaptation to capture keyword correlation efficiently through the use of beam search. The Chapter will be arranged to mirror the typical sequence of processing events in most automatic image tagging systems, namely *image pre-processing*, *image tagging* and *system evaluation*. We begin this Chapter by discussing the adopted system architecture in Section 3.1. This will provide an introduction and high-level view of the major system components the implementation of which will be further described in the remaining sections of the Chapter.

### 3.1 Software & Architecture

It was decided that the CRM and BS-CRM models would be implemented in the MATLAB programming language. The reason for this choice was due to the wide use of MATLAB amongst the research community enabling the code developed as part of this dissertation to be re-used in the future and to enable the author to leverage open source research code libraries where possible. Furthermore, the high-level of abstraction from the underlying machine provided by MATLAB would ensure a respectable level of productivity on the actual model development rather than needless work on memory management and low-level processing that is the hallmark of other languages such as C.

On the other hand we trade-off efficiency and speed in using such a high-level language, which meant that any algorithms produced would need to be highly optimised for the CRM and BS-CRM models to run in reasonable amounts of time and memory. Essentially it was crucial for one to think of the basic unit of implementation as a matrix and to work with matrices and matrix operations as far as possible in order to avoid expensive for-loops. The latter programming constructs are notoriously inefficient and time consuming in MATLAB. Section 3.4 will describe how the CRM and BS-CRM models were engineered to run extremely quickly within MATLAB cutting down on the published runtime of 660 seconds in the literature

to a remarkable 0.45 seconds (amortised) for the custom built solution.

Figure 3.1 illustrates the adopted system architecture detailing the main components of the image tagging model and their interaction. Initially, if the features do not already exist on disk, we run the *image pre-processing module* to take each image and extract a representative set of features. The aim of this step is to replace the high-dimensional images with lower-dimensional features that capture the main properties of the images and enable the model to work on the data with limited memory and computational resources. This module is only ever executed once (unless the nature of the features change) and so pure speed is not a requirement and the pre-computed feature sets can simply be stored in a text file and loaded in by the algorithm at runtime. This module is discussed in Section 3.2.

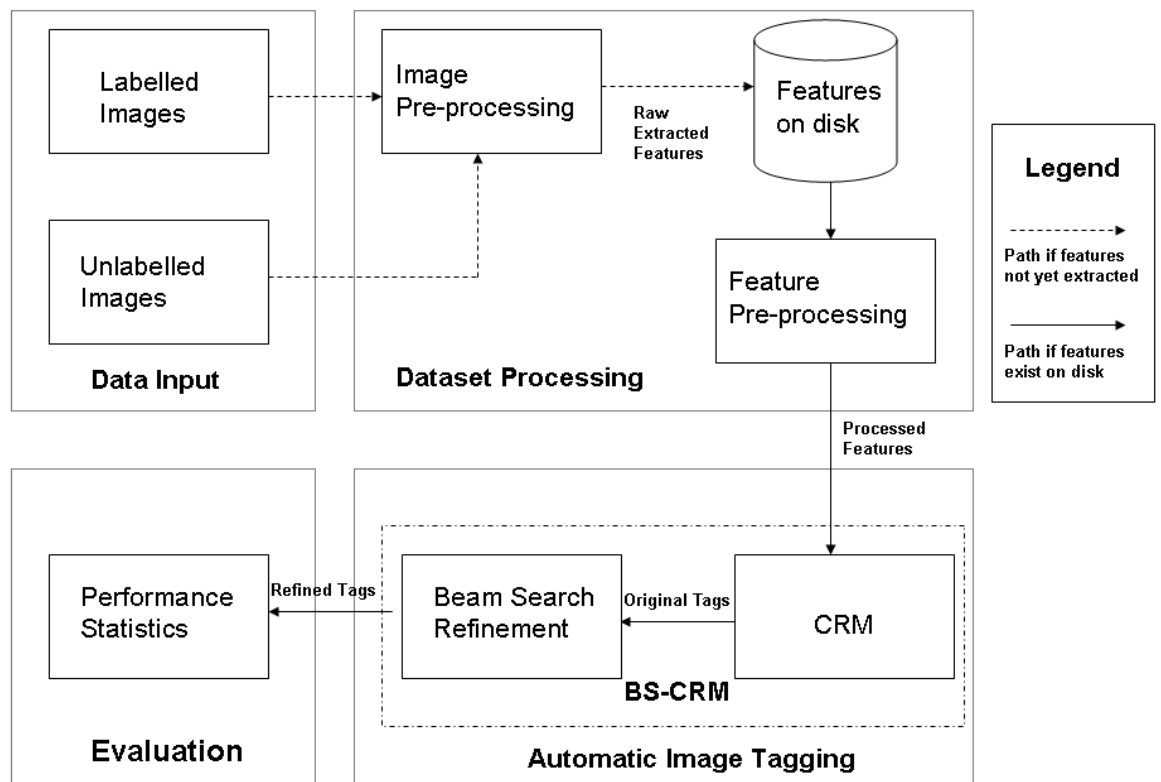


Figure 3.1: This diagram provides an overview of the main components of the automatic image tagging system. The system consists of four main processing components, namely data input, data pre-processing, automatic image tagging and performance evaluation. The dotted arrows are a one off path that is followed initially to extract features from the images. In the future the features are then simply pre-loaded from disk.

The system loads in the pre-computed image features from the text files stored on disk in the

next step of processing which is referred to as the *feature pre-processing module*. Before input into the model these features are further processed to, for example, extract word frequency counts, standardize image features, compute all combinations of 2, 3 and 4 word queries and re-arrange the image features into data structures that allow fast processing within the model. We describe this module in Section 3.3.

The output of the feature pre-processing module is then fed into the *CRM model* itself which constructs a probability distribution to link the provided words and features and allow for the actual automated image tagging and ranked retrieval (described in Section 3.4.1). The initial tags assigned to the images can then be further refined by an optional *beam search tag refinement* module that seeks to find a near to optimal set of tags with high mutual correlation (see Section 3.4.4). Finally, the model outputs the results in a format amendable to processing by the *performance evaluation* component. This final component calculates both the annotation and retrieval performance metrics of the algorithm during the run (refer to Section 3.5).

## 3.2 Image Pre-processing

### 3.2.1 COREL Dataset

#### 3.2.1.1 Overview

The COREL dataset is a very commonly used dataset throughout the image annotation literature with many of the best known image tagging algorithms having been evaluated on this set of images. The dataset contains 5,000 photographs from 50 COREL Stock photograph CD's, with each CD containing 100 images on the same topic. Each image in the dataset contains between 1 and 5 keywords, with a total of 374 keywords in the vocabulary, 371 words of which are present in the training dataset. If we include only those words that exist in the testing set then the vocabulary size is reduced further to 260 words. Most authors appear to split the dataset into 4500 training images, and 500 test images. The training images are also typically further subdivided into 4000 training images and 500 validation images for the purposes of parameter optimisation.

#### 3.2.1.2 Feature Representation

Given the copyright restrictions imposed on the COREL dataset, the author was unable to get hold of the actual images themselves and so the decision was made to utilize the already pre-processed dataset provided by Duygulu et al.<sup>1</sup> which was used in their original translation model paper [10]. The pre-processed data consisted of several text files containing feature vectors for every image in the dataset, the annotations for each image and the image regions

<sup>1</sup>Freely available to download from: [http://kobus.ca/research/data/eccv\\_2002/index.html](http://kobus.ca/research/data/eccv_2002/index.html)

to which the features vectors correspond. The feature representation chosen by Duygulu et al. was designed to capture colour, texture, position and shape information from the Normalized Cuts [51] segmented image regions. Specifically, the authors chose to extract (the number in brackets beside each feature denotes its dimensionality):

- **Position**

- X,Y coordinates of the region normalized by image dimensions (1).

- **Shape**

- Area of the region (1).
- The length of the boundary of the region divided by its area (1).
- Convexity of the region (1).
- Moment-of-inertia or angular mass of the region (1).

- **Colour**

- Average RGB (3).
- Average RBG (duplicated) (3).<sup>2</sup>
- RGB standard deviation (3).
- Average L\*a\*b\* by transforming RGB colour-space (3).
- Average L\*a\*b\* (duplicated) (3).
- L\*a\*b\* standard deviation (3).

- **Texture**

- Mean orientated energy in 30 degree increments (12).

The final feature vector for each region had a dimensionality of 36 with 1-10 regions per image. Whilst there are certainly better feature representations available in the literature<sup>3</sup>, the chosen representation does have the advantages of being relatively compact thereby saving on memory requirements and allowing fast processing with a non-parametric model such as the CRM which needs to store and manipulate the training set at runtime. Furthermore having the exact set of features and the training and testing set splits will bring the further advantage of allowing us to compare the methods developed in this dissertation directly to the work of Duygulu et al. and all subsequent authors that have used their pre-processed dataset.

<sup>2</sup>The authors cite that the RGB and L\*a\*b\* features were duplicated so as to increase their weight for a previous experiment, and that they did not subsequently remove the duplicated columns. There is no good reason to have these duplicated for the purposes of image tagging, nevertheless they were retained in the pre-processed dataset to ensure comparability.

<sup>3</sup>For example, as we will discuss in Section 3.2.2 SIFT [36] and Colour SIFT [7] descriptors coupled with salient region image detectors such as the Kadir & Brady saliency operator [29] which are specifically designed to be invariant to rotation and scale have found to be particularly effective.

### 3.2.2 PASCAL Dataset

#### 3.2.2.1 Overview

There has to be some concern voiced over the validity of the COREL dataset as a challenging and representative testing suite of images for image retrieval. Muller et al. [42] cite that a chance pick of a subset of these images may yield significantly better performance compared to the case where another subset of the images are chosen as the validation set. Furthermore many of the test images in the COREL dataset are very similar to the corresponding training image<sup>4</sup> (see Figure 3.2), which does not give one confidence of how such models will perform on more challenging datasets such as images from the Internet [62]. Indeed, Yavlinksy et al. [62] have demonstrated that taking into account simple features such as the global colour distribution in an image can yield excellent annotation and retrieval performance on this dataset.



Figure 3.2: *These example images from the COREL dataset demonstrate how similar some images are to each other in the dataset. For this reason many authors have criticised this dataset as being relatively easy to annotate.* Source: Jiayu Tang, Automatic Image Annotation and Object Detection, PhD Thesis [27]

Given this, it was decided that in this dissertation the CRM and BS-CRM models would

<sup>4</sup>This can be shown as follows: in the training set there are 2705 images with 4 word tags comprising a vocabulary of 342 different words. These 4 word annotations only make up 1833 different combinations. The probability of getting such an extremely low number of combinations for a sample size of 2705, assuming random selection, is approximately zero.

be further evaluated on the much more challenging PASCAL VOC 2007 image dataset (Figure 3.3 demonstrates some example images from this dataset) that includes objects over wide viewpoint and pose variations. Furthermore the actual images are freely available and therefore it will be possible to demonstrate the system generated annotations against the original images.

The PASCAL Visual Objects Challenge (VOC) is an object recognition competition that has been running annually since 2005<sup>5</sup> with the express objectives of both evaluating performance on object class recognition and compiling a standardised collection of object recognition databases. The dataset consists of 5011 training images and 4052 testing images with standard splits provided that partition the training set into 2501 training and 2510 validation images. In total there are 9,963 images, containing 24,640 annotated objects. Despite having nearly double the number of images of the COREL dataset, the total vocabulary size is significantly smaller with 20 words.

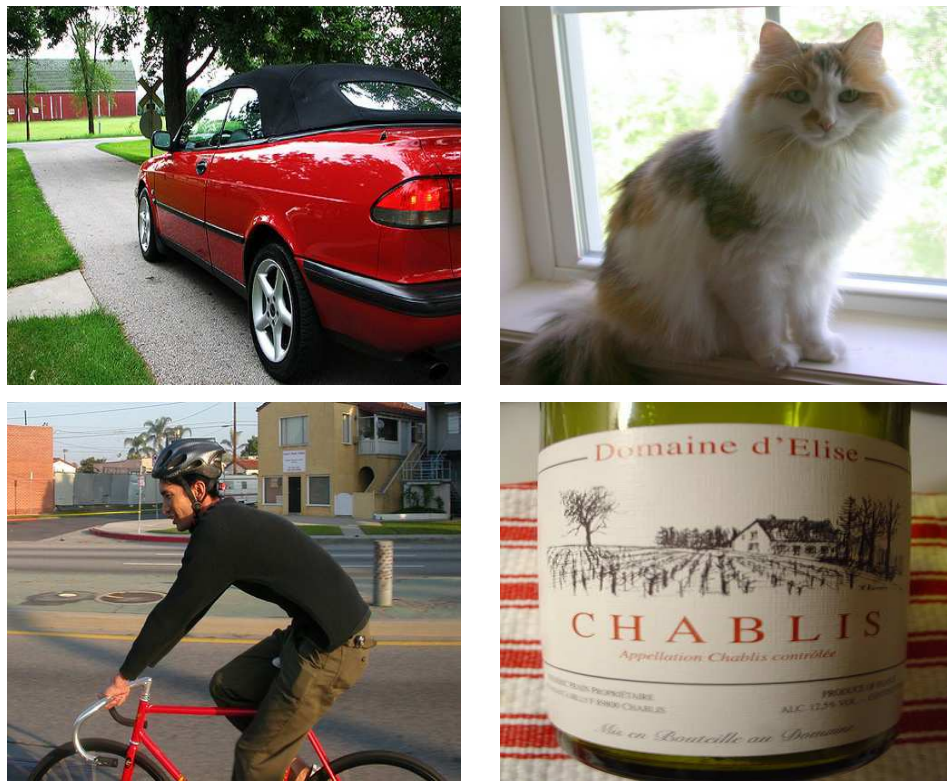


Figure 3.3: *Some example images from the PASCAL VOC 2007 challenge dataset.*

It has to be noted that most of the publications using the PASCAL VOC 2007 dataset are more specifically related to the pure object recognition and detection literature rather than to automated image tagging. The image tagging literature does not include many past publications that use this dataset and in this sense by testing both the CRM model and BS-CRM model on this dataset the research boundaries are being extended in this dissertation by ascertaining how

<sup>5</sup><http://pascallin.ecs.soton.ac.uk/challenges/VOC/>



well the model migrates from a relatively easy dataset to a dataset that is generally considered to be much more challenging.

### 3.2.2.2 Feature Representation

In terms of feature extraction, unlike COREL, there is no widely available pre-processed version of the PASCAL VOC 2007 dataset. Therefore, a custom feature extraction module had to be engineered for this dataset. Given the wide availability of standard code in MATLAB, the use of 128 dimensional SIFT descriptors alongside the SIFT salient region detector were initially investigated<sup>6</sup>.

#### (a) First consideration: SIFT - a local descriptor for saliency

The Scale Invariant Feature Transform or SIFT [36] (also referred to as a Circular Region Detector [14]) is a technique for detecting and describing local image features for the purpose of object recognition in Computer Vision. The feature set extracted by the algorithm has the appealing properties of invariance to scale, translation and rotation, with a partial invariance to illumination changes and affine or 3D projection. To extract these features the SIFT algorithm applies a four stage approach to feature detection and description:

1. Identification of points of interest (or *keypoints*) that are detectable from different view-points of the same object. This is achieved by applying the difference of Gaussians scale space operator.
2. Filtering out those keypoints that have poor contrast and those that are poorly localized on edges.
3. Assignment of consistent orientations to the remaining keypoints using local image properties. This is performed to achieve rotation invariance.
4. Calculating a set of 128 element feature vectors (known as a SIFT feature vectors) from the keypoints. Each SIFT feature consists of 16 histograms, aligned in a 4x4 grid, each with 8 orientation bins.

It is typically the case that around the order of 2000 SIFT features are extracted for a 500x500 pixel image with the sheer number of features ensuring that the algorithm has considerable robustness to occlusions in images. SIFT features have also been demonstrated to yield better performance compared to other methods used in the literature [28] making the algorithm a popular choice for region detection and representation.

---

<sup>6</sup>The open source VLFeat SIFT descriptor and detector MATLAB package were used (<http://www.vlfeat.org/overview/sift.html>).

However this SIFT based approach was eventually discarded as a potential image descriptor for the PASCAL dataset exactly due to the high volume of features the algorithm extracts per image (see Figure 3.4). For the PASCAL dataset SIFT typically produced 1000 or more salient regions per image, and given each of these salient regions are represented in 128 dimensional feature vectors, the final feature datasets for all 9,963 images were over 4GB in size on disk. Loading and manipulating such large matrices in 32-bit MATLAB was simply not possible, so a number of other methods were attempted to handle the sheer size of the SIFT representation before the final rejection of the idea.

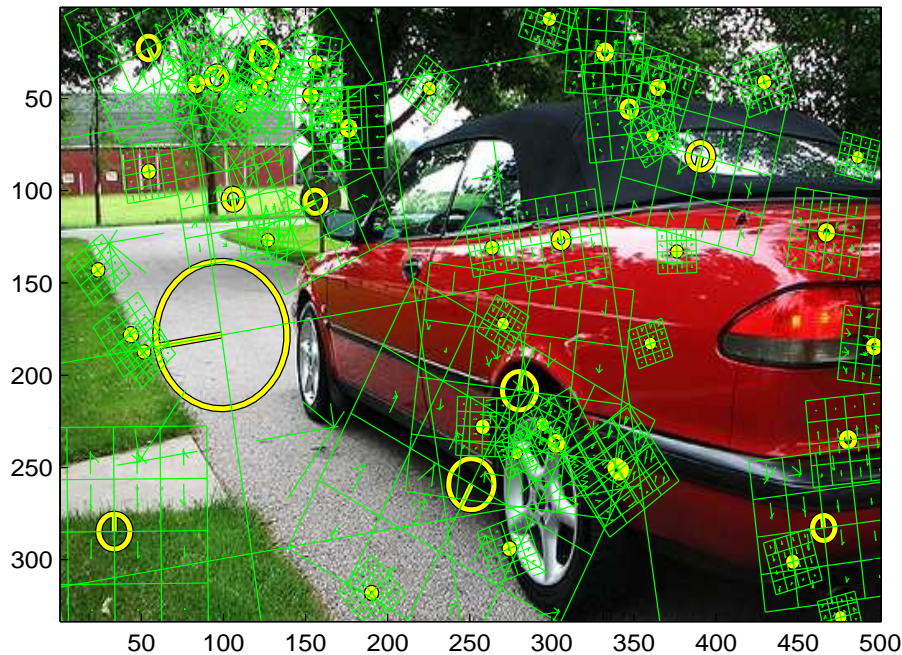


Figure 3.4: *The VLFeat SIFT library used to create a SIFT representation of an image from the PASCAL training dataset. Notice the large number of descriptors (denoted by the yellow circles) extracted for this image (there are 545 in total, although not all can be clearly seen by the human eye).*

Firstly, an attempt was made to take “slices” of the features from file and build up the image feature similarity matrix incrementally (see Section 3.4 for a discussion of the similarity matrix in the context of the CRM model). However continually loading from disk, even from an optimized memory mapped file was substantially slower than main memory, and it was estimated that the calculation of the final image feature similarity matrix for the dataset would take of the order of 40 days, which is clearly infeasible, especially considering that this matrix needs to be re-calculated many times the  $\beta$  parameter for the CRM model changes which is

required in the parameter sweep during cross-validation.

Therefore, as a next attempt the author reduced the sensitivity setting on the SIFT salient region detector so that it would return fewer salient regions per image. Unfortunately, at the setting that would give workable sized feature sets this detector was quite temperamental returning widely varying number of salient regions from 300 to 0 depending on the image of interest. This wide variability, especially with some images returning 0 features is clearly unacceptable if we wish to obtain reasonable image tagging performance on the dataset. As a final attempt to quell the curse of dimensionality with this particular image representation the use of the recently developed PCA-SIFT [31] algorithm was investigated.

PCA-SIFT applies principal components analysis (PCA) to the normalized gradient patch produced by the original SIFT algorithm. The authors demonstrated that the PCA-based local descriptors were more distinctive, more robust to image deformations, and more compact than the standard SIFT representation and are ideal for image retrieval scenarios. The implementation of PCA-SIFT that was investigated was that provided by the original authors and made available to download on their website<sup>7</sup>. This algorithm included a pre-processed PCA subspace matrix that could be used with any dataset to reduce the SIFT feature dimensionality. This pre-processed matrix was of dimensionality 36 therefore limiting the dimensionality reduction of the PASCAL dataset SIFT representation from 128 to 36.

Despite this significant reduction, it was soon discovered that the real issue with the SIFT descriptors was not the dimensionality but the number of features extracted. Over 1000 features of reduced dimensionality of 36 were still too large to manipulate (approximately 1GB on disk). The problems that were faced in using the SIFT features essentially boils down to the nature of the CRM model in that it operates directly on the continuous image features without any dimensionality reduction step such as clustering which many of the alternative models in the literature apply to the features. Therefore the CRM is essentially trading off image feature robustness against increased accuracy from avoiding an error prone vector quantization step.

### **(b) Final Representation: Mixture of simple descriptors**

Having decided against the use of SIFT descriptors, inspiration was taken from the aforementioned COREL dataset representation advocated by Duygulu et al. This 36 dimensional representation has the attractive properties of being very compact and memory efficient. Furthermore by selecting a large mixture of different features in this manner we can lower the bias of any one individual feature and maximize the amount and variety of information extracted from the images.

Therefore the decision was made to replicate as far as possible this feature representation for the PASCAL dataset. Given that no code was provided by Duygulu et al., this necessitated

---

<sup>7</sup><http://www.cs.cmu.edu/~yke/pcasift/>

engineering custom code to produce the following 42 dimensional feature vectors for every extracted image region:

- **Position**

- X,Y coordinates of the region normalized by image dimensions (2).

- **Colour**

- Average RGB (3).
- RGB standard deviation (3).
- RGB skewness (3).
- Average  $L^*a^*b^*$  by transforming RGB colour-space (3).
- $L^*a^*b^*$  standard deviation (3).
- $L^*a^*b^*$  skewness (3).
- Average HSV by transforming RGB colour-space (3).
- HSV standard deviation (3).
- HSV skewness (3).

- **Texture**

- Mean orientated energy in 30 degree increments (12).

Here we add two more simple properties to the Duygulu et al. representation, namely the skewness metric to capture the asymmetry of the colour distributions and the HSV colour space. Derivation of the position and colour features (standard deviation, mean per channel) are mostly self-explanatory, however it is worth making note of why one has chosen to use both RGB,  $L^*a^*b^*$  and HSV features in combination. These three colour space measure different properties of colour all of which are useful in the object recognition process. In comparison to RGB which is the default colour space for image capturing and display, the HSV (Hue, Saturation, and Value) colour space encodes the amount of light illuminating a colour in the Value channel whilst the  $L^*a^*b^*$  colour space captures human perception of brightness in its luminance channel.

In their paper describing the MBRM model [13] Feng et al. reveal that partitioning an image into a regular grid yields superior performance with the relevance model based approach compared to the use of a dedicated segmentation algorithm such as Normalized Cuts [51] or Blobworld [8] which attempt to find a coherent set of salient regions within the images. Segmentation is a very difficult problem<sup>8</sup> and the current set of algorithms we have at our disposal

---

<sup>8</sup>It is worth noting here that perfect segmentation is itself nearly as difficult as the general problem of image understanding.

today are by no means perfect and will fail to segment meaningful regions from time to time ultimately leading to a degradation in annotation performance. Furthermore the segmentation process is inherently computationally expensive and so any model relying on this function as a pre-processing step will necessarily suffer in terms of scalability and performance.

Each PASCAL image is therefore partitioned into a regular non-overlapping grid consisting of approximately 85x85 pixels yielding 20 regions per image. Samples of the aforementioned features would then be taken from each of these regions. The final pre-processed dataset size was approximately 40Mb on disk taking just over 4 hours to create for all 9,963 images (this is a one off cost, unless the features change), a much more reasonable memory requirement compared to the use of SIFT features. Given that we are partitioning over a regular grid rather than irregularly shaped “blob” features as are produced by dedicated segmentation algorithms, the shape information (which requires the silhouettes of objects) captured by Duygulu et al. was therefore not extracted in the chosen representation.

Finally in addition to colour properties, we follow the example of Duygulu et al. and extract texture features from each region. Texture is another important low-level visual cue for image representation, and can be captured through the use of oriented *Gabor filters* which have been shown to be particularly effective in creating sparse yet discriminative image features (see Figure 3.5):

*“Properly tuned Gabor filters, can remove noise, preserve the true ridge and valley structures, and provide information contained in a particular orientation in the image.” Jain et al. [22]*

For the PASCAL dataset, a Gabor filter bank is constructed consisting of 12 different oriented Gabor filters in 30 degree increments of orientation. This bank is applied to every extracted region across the regular grid, with the final feature value computed by calculating the average absolute deviation from the mean of the filter responses:

$$T_{\theta} = \frac{1}{N} \sum_{i=1}^N |F_{i\theta} - \mu_{i\theta}| \quad (3.1)$$

Here  $\theta \in \{0^{\circ}, 30^{\circ}, 60^{\circ}, 90^{\circ}, 120^{\circ}, 150^{\circ}, 180^{\circ}, 210^{\circ}, 240^{\circ}, 270^{\circ}, 300^{\circ}, 330^{\circ}\}$ ,  $N$  is the number of pixels in the extracted region, and  $\mu_{i\theta}$  is the mean of the pixels in region  $F_{i\theta}$ .

### 3.3 Feature Pre-processing

The system loads in the pre-computed image features from the text files in the next step of processing. Before input into the model these features are further processed in the following manner:

- Image Features:

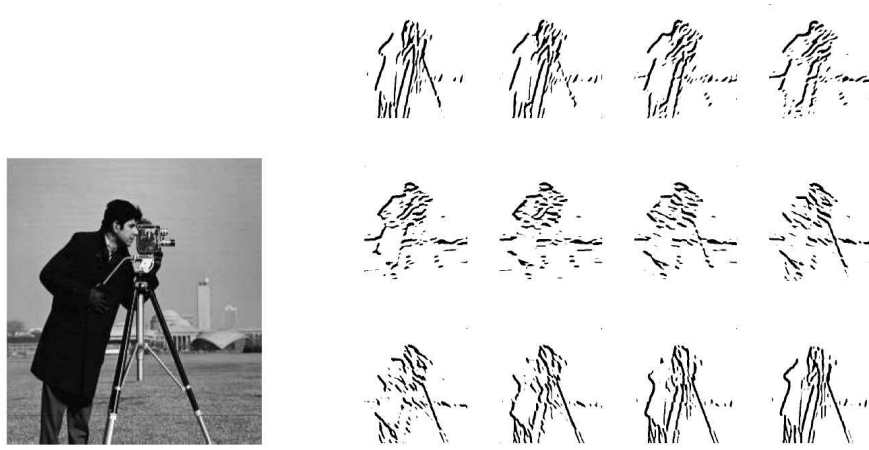


Figure 3.5: This diagram illustrates the application of a Gabor filter bank to an example image. The original image is shown on the left. On the right we can see the different responses generated by convolution of the image with the oriented Gabor filters. Only those edges in close proximity to the preferred orientation of the given filter produce a significant output response.

- Normalization of the training features by subtracting the mean of the training feature set and dividing by the standard deviation.
  - Similar normalization for the testing dataset (mean and standard deviation of the *training dataset* is used).
  - Re-shaping of the image features into 3-dimensional matrices amenable to fast processing by the CRM module (see Section 3.4).
- Word Features:
    - Calculation of those words that occur in the training dataset.
    - Calculation of the frequencies of the training dataset words.
    - Calculation of those words that occur in the testing dataset.
    - Calculation of those words that occur at least twice in the testing dataset.
    - Calculation of the relative frequency of the training set words.
    - Calculation of all possible 2,3,4 word queries. After calculation these queries can be loaded from disk in the future to save processing time.

### 3.4 CRM Model Implementation

#### 3.4.1 Original CRM Model

Equation 2.2 presented in Chapter 2 is the defining equation of the CRM model. As discussed, the goal of the CRM model implementation is to calculate, for a word  $w$  and image features

$f_1, \dots, f_m$  the conditional probability of the word given those features or  $P(w|f_1, \dots, f_m)$ :

$$P(w|f_1, \dots, f_m) = \frac{\sum_{J \in T} P_{\mathcal{V}}(w|J) \prod_{i=1}^m P_{\mathcal{F}}(f_i|J)}{\sum_{J \in T} \prod_{i=1}^m P_{\mathcal{F}}(f_i|J)} \quad (3.2)$$

From a cursory glance at this equation, it is apparent that any implementation of the model needs to take into consideration two basic components:

- The image feature probability:  $P(I|J) = \prod_{i=1}^m P_{\mathcal{F}}(f_i|J)$  where  $J$  is the training image with features  $f_{1T}, \dots, f_{nT}$  and  $I$  is the testing image with features  $f_1, \dots, f_m$ . Incorporating the Gaussian non-parametric kernel density estimator into this equation yields:

$$P(I|J) = \prod_{i=1}^m \frac{1}{n} \sum_{j=1}^n \frac{1}{\sqrt{2^k \pi^k |\Sigma|}} \exp \left\{ -\frac{\|f_i - f_{jT}\|^2}{\beta} \right\} \quad (3.3)$$

We will refer to the complete set of the logarithm of these probabilities  $P(I|J)$  across all training and test images as the *image similarity matrix*  $\mathbf{S}$ .

- The word smoothing distribution:  $P_{\mathcal{V}}(w|J)$ . The matrix holding these probabilities for every training image and word will hereby be referred to as the *word probability matrix*  $\mathbf{W}$ .

Our goal is to use these matrices to obtain  $\mathbf{P}$ , a matrix which has as elements  $P(w|I)$  for each word  $w$  in the vocabulary and each image  $I$  in the test set (Figure 3.6 illustrates the structure of these three important matrices). The construction of the word probability matrix  $\mathbf{W}$  is relatively straightforward and simply requires the application of the appropriate smoothing equation to build a matrix of dimension  $N_{train} \times N_{word}$ <sup>9</sup>, where  $N_{train}$  is the size of the training dataset and  $N_{word}$  is the size of the vocabulary. The creation of the image similarity matrix requires more thought however, mainly due to the sheer size of the features extracted for each image. It is to the efficient creation of this matrix that our attention will now turn.

#### 3.4.1.1 Image similarity matrix

The CRM model relies on having a similarity value for every testing image  $J$  against every training image  $I$  as given in Equation 3.3. Given that we take the product of generative probabilities in the CRM equation we are therefore very likely to encounter numerical underflow during annotation. Therefore it is necessary to convert the image feature probabilities to the

<sup>9</sup>Here we use the MATLAB matrix dimension notation:  $R \times C$  means we have  $R$  rows and  $C$  columns in our matrix.

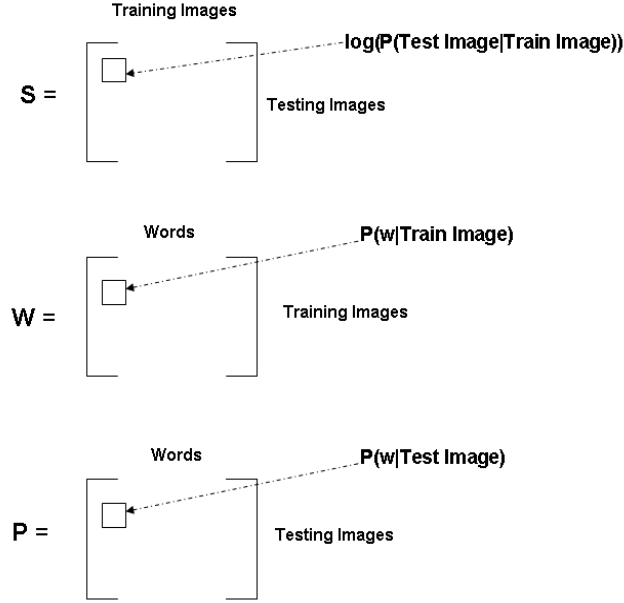


Figure 3.6: Illustration of the structure of the three main matrices used in the CRM model for the purposes of image tagging.  $\mathbf{W}$  contains  $P(w|J)$  in every entry,  $\mathbf{S}$  contains  $\log\{P(I|J)\}$  as elements and  $\mathbf{P}$  contains  $P(w|I)$  which is the final probabilities that we require for image tagging.

logarithmic domain which effectively replaces multiplication by addition. We only come back out into the probability domain after we divide by the denominator in Equation 2.2. To convert the  $P(I|J)$  we take the logarithm to obtain:

$$\log\{P(I|J)\} = -\sum_{i=1}^m \left\{ \log(n) + \log \sum_{j=1}^n \exp \left( \log \left\{ \frac{1}{\sqrt{2^k \pi^k |\Sigma|}} \exp \left\{ \frac{\|f_i - f_{jT}\|}{\beta} \right\} \right\} \right) \right\} \quad (3.4)$$

This leads to a neat implementation in MATLAB as a matrix  $\mathbf{S}$  with the testing image number  $j$  denoting the row and the training image number  $i$  denoting the column and every element of the matrix giving the log similarity  $S_{ij}$  between image  $j$  and  $i$ . The implementation question that needs to be solved is how one processes the raw image features to obtain this matrix of dimensions 500x4500 for COREL and 4950x5011 for PASCAL.

From Equation 3.2, we can see that in order to construct  $\mathbf{S}$  we need to compute the pairwise distances between the features  $f_i$  and  $f_{jT}$  of both images within the non-parametric kernel:  $\|f_i - f_{jT}\|$ . A possible starting point therefore would be to compute a feature similarity matrix  $\mathbf{F}$  with each element  $\mathbf{F}_{ij}$  giving the Euclidean distance between  $f_i$  and  $f_{jT}$ . From this matrix



we could then derive the required log probabilities  $\log \{P(I|J)\}$  by taking the appropriate sums as given in Equation 3.4.

For the COREL dataset, computing the pairwise distance outright between every feature vector of the testing and training dataset would yield a matrix of size approximately 5000x45000 (of the order of 2GB on disk) given that there are roughly 10 regions per image. For PASCAL we would have an even larger matrix of size 99000x100220 (80GB on disk) given there are 20 regions per image. Both matrices are far too large to load and manipulate in main memory within MATLAB.

The decision was taken therefore to avoid these matrices altogether and incrementally build up the required image similarity matrices  $\mathbf{S}$  by taking the test image features in discrete “blocks” of size  $N_{block\_size} \ll N_{test}$ . The training matrix contains all of the features vectors for every training image in three-dimensional matrix of size  $N_{dim} \times N_{blobs} \times N_{train}$  where  $N_{dim}$  is the dimensionality of the features (36 for COREL, 42 for PASCAL),  $N_{blobs}$  is the number of blobs per image (1-10 for COREL, 20 for PASCAL) and  $N_{train}$  is the number of training images (4500 for COREL, 5011 for PASCAL). As we take the testing images in blocks<sup>10</sup> the testing matrix will also be three-dimensional of shape  $N_{dim} \times N_{blobs} \times N_{block\_size}$ .

The squared Euclidean distance function *sqdist* from the open source Lightspeed library<sup>11</sup> is then used to efficiently calculate the distance between these two matrices returning a four-dimensional result of shape  $N_{blobs} \times N_{blobs} \times N_{train} \times N_{test}$  with each element giving the pairwise image feature distances.

The key point is, now that we have this four dimensional distance matrix for a subset of the test images against all of the training images, we proceed by effectively reducing its dimensionality by deriving the  $P(I|J)$  using Equation 3.4 for test image  $I$  and training image  $J$  yielding a final matrix of size  $N_{block\_size} \times N_{train}$ .

As we require a final  $\mathbf{S}$  matrix of size  $N_{test} \times N_{train}$  we continually append each block in an incremental fashion. This methodology is extremely efficient calculating (in memory) the required similarity matrix in 30 seconds for COREL and approximately 600 seconds for PASCAL. This similarity matrix can then be stored and loaded straight from disk in future (unless the kernel bandwidth  $\beta$  changes, in which case the matrix will obviously need to be re-calculated.).

### 3.4.2 Annotating Images

Having built up the required image  $\mathbf{S}$  and word  $\mathbf{W}$  matrices we are now in a position to tag unseen test images. Our aim here is to calculate a matrix  $\mathbf{P}$  of dimensions  $N_{test} \times N_{word}$  where

<sup>10</sup>The block size is ultimately limited by machine memory. On a 4GB 32-bit machine this was found to be 25 images. The larger the block the faster the processing, enabling the algorithm to easily scale to 64-bit machines in the future.

<sup>11</sup><http://research.microsoft.com/en-us/um/people/minka/software/lightspeed/>

$N_{test}$  is the number of test images and  $N_{word}$  is the number of words in the vocabulary (for PASCAL this is 4952x20 and COREL 500x374). Each element of the matrix  $\mathbf{P}_{ij}$  gives the required conditional probability  $P(w_i|I)$  of a word belonging to a particular test image.

It is possible to compute  $\mathbf{P}$  in one matrix multiplication. To see this we need to re-arrange the CRM joint probability:

$$P(w, f_1, \dots, f_m) = \sum_{J \in T} P_T(J) P_{\mathcal{V}}(w|J) \prod_{i=1}^m P_{\mathcal{F}}(f_i|J)$$

Aggregating from the feature to the image level, this equation can also be expressed as:

$$P(w, I) = \sum_{J \in T} P_T(J) P_{\mathcal{V}}(w|J) P(I|J)$$

Given this, we calculate  $P(w|I)$  by dividing by  $P(I)$ :

$$P(w|I) = \sum_{J \in T} P_{\mathcal{V}}(w|J) \frac{P(I|J)}{\sum_{J \in T} P(I|J)} \quad (3.5)$$

The prior  $P_T(J)$  vanishes. Here we divide  $P(I|J)$  which is a small value by the sum of small values  $\sum_{J \in T} P(I|J)$ , which will result in a much larger value that does not result in numerical underflow.

Now, by Bayes Theorem we can re-arrange this equation to obtain the posterior probability  $P(J|I)$ :

$$P(w|I) = \sum_{J \in T} P_{\mathcal{V}}(w|J) P(J|I) \quad (3.6)$$

As discussed,  $P(w|I)$  are simply elements of  $\mathbf{P}$  and  $P_{\mathcal{V}}(w|J)$  are elements of  $\mathbf{W}$ . Therefore, this equation is another way of writing the matrix multiplication of the word probability matrix  $\mathbf{W}$  with the *posterior* image similarity matrix  $\mathbf{S}_{ap}$  which will calculate all of the relevant  $P(w|I)$  in  $\mathbf{P}$  simultaneously:

$$\mathbf{P} = \mathbf{S}_{ap} \times \mathbf{W} \quad (3.7)$$

If we can obtain the matrix  $\mathbf{S}_{ap}$  we will be able to annotate the entire set of testing images in one matrix operation which will be extremely fast within MATLAB.

For Equation 3.7 to work correctly both  $\mathbf{W}$  and  $\mathbf{S}_{ap}$  need to be in the probability and not logarithmic space. This is not a problem for  $\mathbf{W}$  as this matrix is already in the probability space and does not suffer from numerical underflow given we are not taking the product of probabilities as we are in the kernel density estimation part of the CRM equation.

However, working with the original image similarity matrix probabilities  $\mathbf{S}$  is not possible due to their extremely small magnitude. We therefore use the posterior probability matrix  $\mathbf{S}_{ap}$

which contains  $P(J|I)$  in every element. As the mathematical manipulation to obtain Equation 3.6 demonstrated, this will still correctly give us  $P(w|I)$  which is the quantity that we desire.

To calculate  $\mathbf{S}_{ap}$ , we firstly derive the normalization factor  $\sum_{J \in T} P(I|J)$  but expressed in the logarithmic domain:

$$\mathbf{Z} = \log \left\{ \sum_{J \in T} \exp \{ \log \{ P(I|J) \} \} \right\} \quad (3.8)$$

Now given this, the matrix  $\mathbf{S}_{ap}$  can be calculated in the following manner:

$$\mathbf{S}_{ap} = \exp \{ \mathbf{S} - \{ \mathbf{Z} \times \mathbf{1}_{1 \times N_{train}} \} \} \quad (3.9)$$

$\mathbf{Z}$ , being a vector of dimension  $1 \times N_{test}$  is replicated by the cross-product  $\mathbf{Z} \times \mathbf{1}_{1 \times N_{train}}$  column wise by the number of training images  $N_{train}$  so that it may be subtracted from  $\mathbf{S}$  in Equation 3.9. We now have all of the required machinery to annotate an image.

As mentioned in Section 3.1 the fact that we can now annotate all 500 test images (for COREL) and 4052 test images (for PASCAL) in one line of code brings a significant speed benefit. In implementation of the CRM model in [12] the authors cite a runtime of 660 seconds to annotate all 500 COREL images. With the custom built solution for this dissertation we can do so in 0.45 seconds<sup>12</sup>. As we are attempting to push the research boundaries in this dissertation the quick runtime has the benefit of allowing the algorithm to be run many times enabling one to try out a wider range of parameter values on the validation set thus ensuring our algorithm is better tuned.

### 3.4.3 Adding Keyword Correlation

Mathematically, it is relatively straightforward to amend the CRM model to capture keyword correlation. We do so as follows:

$$P(w_1 \dots w_k | I) = \sum_{J \in T} \prod_{j=1}^k P_{q'}(w_j | J) P(J | I) \quad (3.10)$$

Here we are pushing the product over the word probabilities into the sum over the training images  $J$ , rather than, as in the original CRM equation, calculating  $P(w|I)$  separately for each word then multiplying each probability together, thereby assuming each word is conditionally independent given the image  $I$ :

$$P(w_1 \dots w_k | I) = \prod_{j=1}^k P(w_j | I)$$

---

<sup>12</sup>This time assumes we have built the image similarity matrix already, if not then this adds approximately 30 seconds for the COREL dataset to be pre-processed before annotation, still significantly less than the literature. Furthermore unless the  $\beta$  parameter changes we do not need to re-calculate the image similarity matrix again, and so we are justified in quoting 0.45 seconds as the amortized runtime of the algorithm.

### 3.4.4 The BS-CRM Model

As discussed in the literature review, previous authors amend the CRM [2] and CMRM [64] to add keywords to an existing set  $S_k$  of  $k$  words using a formula similar to Equation 3.10. As the complexity of finding an optimal (highest probability) sets of tags out of a vocabulary of words grows exponentially with the size of the vocabulary<sup>13</sup>, they therefore do so in a “greedy” manner only adding the keyword that leads to the maximum probability  $P(S_k|I)$  of having all of the keywords together in the set:

$$S_k^* = \operatorname{argmax}_{S_k \subset V} P(S_k|I) \quad (3.11)$$

In the automatic image tagging field we can live with a good sub-optimal solution if it allows us to isolate irrelevant words that would have otherwise been selected. This dissertation investigates to what extent augmenting this approach of Zhou et al. [64] using *beam search* can be used to select a set of high probability words to describe novel images. The authors have demonstrated that considering keyword correlations is a viable approach to increasing the performance of automated image annotation algorithms, with for example, Zhou et al. reporting a 15.5% improval in mean per word recall, and a 3.7% improval in mean per word precision.

However for the existing approaches in the literature, the width of the beam search is effectively 1 as they only keep one hypothesis (set of keywords) at every step in the search tree (please refer to Section 2.6 for a detailed review of beam search). Their approach of finding the set of keywords  $\{w_1 \dots w_k\}$  works in the manner shown in Figure 3.7.

In this approach, we are not guaranteed that the next word chosen, even if it does contribute the maximum gain to the selected subset, does not cause the probability mass of future words to be skewed such that one or more relevant keywords down the line are therefore missed. To overcome this issue, in this dissertation, we use beam search with multiple beams to keep several hypotheses at level in the search tree and actively evaluate each in parallel. The proposed BS-CRM algorithm operates as shown in Figure 3.8. Figure 3.9 illustrates the operation of this algorithm on a toy example.

We can also further express the Beam Search algorithm using matrix terminology by considering the  $\mathbf{P}$  annotation matrix defined earlier. In this case we run the original CRM model and calculate  $\mathbf{P}$  using Equation 3.7 as before. Now rather than sorting the probabilities in  $\mathbf{P}$  and simply taking the top, for example, five words as the annotation for each image, we instead input  $\mathbf{P}$  into the beam search keyword refinement algorithm for further processing as shown in Figure 3.10.

---

<sup>13</sup>To see this, consider the number of ways of selecting a subset of size  $N$  from a vocabulary of size  $V$ , where  $V \gg N$ , this is approximately  $N!$ , a huge number for the relatively modest vocabulary sizes (typically 20-400) used in the automated annotation literature.

- **Step 1:** Suppose  $w_1$  is the keyword with largest probability:

$$\begin{aligned}\{w_1\} &= \operatorname{argmax}_{w \in V} P(w|I) \\ &= \operatorname{argmax}_{w \in V} \sum_{J \in T} P_{q'}(w|J)P(J|I)\end{aligned}$$

- **Step 2:** We now add a second word  $w_2$  to the set that maximizes the objective function:

$$\begin{aligned}\{w_1, w_2\} &= \operatorname{argmax}_{w_1, w_2 \in V} P(w_1, w_2|I) \\ &= \operatorname{argmax}_{w_1, w_2 \in V} \sum_{J \in T} P_{q'}(w_1|J)P_{q'}(w_2|J)P(J|I)\end{aligned}$$

- **Step 3:** We repeat this procedure at each step adding a new word to the existing set of words that achieves the greatest probability of all words occurring together in the set:

$$\begin{aligned}\{w_1 \dots w_k\} &= \operatorname{argmax}_{w_1 \dots w_k \in V} P(w_1 \dots w_k|I) \\ &= \operatorname{argmax}_{w_1 \dots w_k \in V} \sum_{J \in T} P_{q'}(w_1|J) * \dots * P_{q'}(w_k|J)P(J|I)\end{aligned}$$

- **Step 4:** The algorithm terminates when the number of words in our set is equal to the desired length of the caption (usually 5 words).

Figure 3.7: Amending the CRM model to capture keyword correlation in the manner suggested by Wang et al. [58].

- **Step 1:** Spawn B candidate sets, where B is the length of the beam. Add the first word  $w_1$  with the highest probability to all B sets. At this stage all sets contain the same word. For the  $j$ th set we have:

$$\begin{aligned}\{w_1\}_j &= \operatorname{argmax}_{w \in V} P(w|I) \\ &= \operatorname{argmax}_{w \in V} \sum_{J \in T} P_{\mathcal{V}}(w|J)P(J|I)\end{aligned}$$

- **Step 2:** For the  $j$ th beam we add the  $j^{\text{th}}$  largest probability word to the current word in the set, therefore each set will have a different second word added  $w_2^j$ , with the superscript  $j$  indicating that the second word might be different for each set  $j$ :

$$\begin{aligned}\{w_1, w_2^j\}_j &= \operatorname{argjth\_max}_{w_1, w_2^j \in V} P(w_1, w_2^j|I) \\ &= \operatorname{argjth\_max}_{w_1, w_2^j \in V} \sum_{J \in T} P_{\mathcal{V}}(w_1|J)P_{\mathcal{V}}(w_2^j|J)P(J|I)\end{aligned}$$

- **Step 3:** We then continue as per Step 1 and add the *maximum probability* word to each set, which again might be different between sets:

$$\begin{aligned}\{w_1 \dots w_k^j\}_j &= \operatorname{argmax}_{w_1 \dots w_k^j \in V} P(w_1 \dots w_k^j|I) \\ &= \operatorname{argmax}_{w_1 \dots w_k^j \in V} \sum_{J \in T} P_{\mathcal{V}}(w_1|J) * \dots * P_{\mathcal{V}}(w_k^j|J)P(J|I)\end{aligned}$$

- **Step 4:** The algorithm terminates when the number of words in each set is equal to the desired length of the caption (usually 5 words). Out of the B sets we then select the one set with the highest probability as the final annotation of the image.

Figure 3.8: The proposed BS-CRM model using beam search to find a close to optimal set of tags for an image.

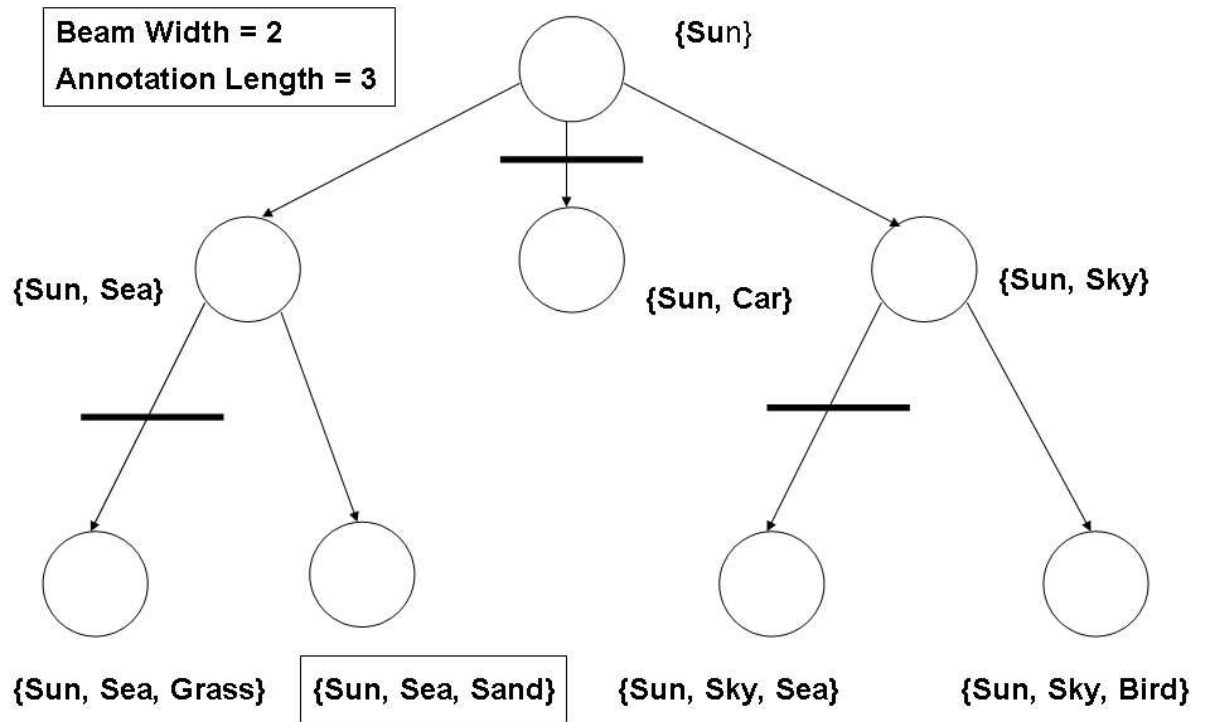


Figure 3.9: This diagram illustrates the operation of beam search on an example tree in the context of automated image tagging. In this case the beam width is set to  $B=2$ . The algorithm proceeds in a breadth-first manner and branches the  $w$  (2 in this case) most promising nodes (as measured by  $P(S_k|I)$  where  $k$  is the tree level) at each level. So we can see that the algorithm first picks Sun as the root. From this we choose Sea and Sky giving the two hypotheses (Sun,Sea) and (Sun, Sky) with (Sun, Car) pruned. The algorithm then expands the (Sun,Sea) and (Sun, Sky) nodes and picks two more best words in this case Sand and Bird giving the hypotheses (Sun,Sea,Sand) and (Sun, Sky, Bird). (Sun,Sky,Sea) and (Sun,Sea,Grass) are pruned. We proceed in this manner until the desired annotation length has been reached. In this case the annotation length is 3 so we determine which set (Sun,Sea,Sand) or (Sun,Sky, Bird) has the highest probability. In this case the set (Sun,Sea,Sand) has the highest probability and is therefore selected as the final annotation of the image.

- **Step 1:** Set beam width  $B$ . Use the original CRM model to calculate  $\mathbf{P}^t$ , where the superscript  $t$  denotes the value of  $\mathbf{P}$  at iteration  $t$ :

$$\mathbf{P}^t = \mathbf{S}_{\text{ap}} \times \mathbf{W}$$

- **Step 2:** For the  $j$ th beam, find the  $j$ th highest probability words  $\mathbf{w}$  from  $\mathbf{P}^t$ :

$$\mathbf{w} = \arg\text{-}j\text{thmax}_{\text{rowwise}} \{\mathbf{P}^t\}$$

- **Step 3:** Define a function `extract` that obtains the probabilities  $\mathbf{h}$  of the  $j$ th highest probability words in  $\mathbf{P}^t$  from  $\mathbf{W}$ :

$$\mathbf{h} = \text{extract}(\mathbf{w}, \mathbf{W})$$

- **Step 3:** Add the  $j$ th highest probability words to a bookkeeping matrix  $\mathbf{C}_j$  of dimension  $N_{\text{test}} \times N_{\text{annotation\_length}}$ , where  $N_{\text{test}}$  is the number of test images and  $N_{\text{annotation\_length}}$  is the desired annotation length:

$$\mathbf{C}_j = \text{append}(\mathbf{h}, \mathbf{C}_j)$$

- **Step 4:** Replicate  $h$  column-wise to obtain the matrix  $\mathbf{H}$ :

$$\mathbf{H} = \mathbf{h} \times \mathbf{1}_{1 \times N_{\text{train}}}$$

- **Step 5:** Obtain the probabilities  $\mathbf{H}^{t+1}$  of the new highest word occurring with the previous highest words  $\mathbf{H}^t$ . Here we take the matrix element pairwise product (denoted by  $\cdot$ \*) and not the matrix product:

$$\mathbf{H}^{t+1} = \mathbf{H} \cdot \mathbf{H}^t \text{ and } \mathbf{H}^t = \mathbf{H}^{t+1}$$

- **Step 6:** Now we can obtain  $\mathbf{P}^{t+1}$  taking into consideration word correlation:

$$\mathbf{P}^{t+1} = \{\mathbf{H}^{t+1} \cdot \mathbf{S}_{\text{ap}}\} \times \mathbf{W}$$

- **Step 7:** Set the probabilities of the selected words in this iteration within  $\mathbf{P}^{t+1}$  to zero. This ensures that the algorithm simply does not pull out the same word again as a word is always best correlated with itself. Prepare for the next iteration:  $\mathbf{P}^t = \mathbf{P}^{t+1}$

- **Step 8:** Repeat Steps 2-7 until the desired annotation length in  $\mathbf{C}_j$  has been achieved. Execution is now complete for beam  $j$ .

- **Step 9:** Repeat Steps 2-8 for the remaining beams.

- **Step 10:** Find the best wordset in  $\{\mathbf{C}_1 \dots \mathbf{C}_B\}$  for each image. The algorithm has now terminated.

Figure 3.10: The BS-CRM algorithm expressed in matrix terminology.



## 3.5 Evaluation Framework

### 3.5.1 Cross-Validation

There are two main parameters that need to be set for the CRM model; the  $\beta$  parameter giving the bandwidth of the kernel density estimator and either  $\mu$  if we are using the N-CRM or Dirichlet word smoothing functions or  $\lambda$  if we are using the Bernoulli or Multinomial functions. These parameters have a great effect on the performance of the model and therefore a dedicated cross-validation framework was constructed for this dissertation in order to optimize the model before application on the testing dataset.

The cross-validation algorithm essentially performs an exhaustive search over the parameter space of  $\beta$ - $\mu$  or  $\beta$ - $\lambda$  for those combinations of the two parameters that maximize an objective function (for example this could be the Mean Average Precision obtained on the validation set). This cross-validation module is fully automated and integrated with the third-party `trec_eval` (see Section 3.5.2) information retrieval evaluation function so that no manual intervention is required during the parameter optimization process.

The 4500 images of the COREL training dataset is typically divided into 4000 training images and 500 validation images. As no splits are provided, the cross-validation framework therefore randomly splits the 4500 training images into these portions. The splits can then be saved on file to ensure experimental repeatability. Standard splits are provided for the PASCAL dataset which roughly divides the training dataset into 50% training and 50% validation images. These standard splits are therefore used by the algorithm.

### 3.5.2 Integration with Trec Eval

`Trec_eval`<sup>14</sup> is an executable provided for participants of the TREC conference to evaluate their information retrieval algorithms. The executable is extremely comprehensive automatically calculating many popular information retrieval evaluation metrics, including Recall, Precision, Average Precision, Mean Average Precision, P@5 and the data required to precision-recall chart amongst many other useful metrics.

The function requires as input two files: a so-called `.qrel` file detailing the actual documents that are relevant to a particular query and a `.top` file with the system generated results with confidence values (between 0 and 1) indicating how well the system thought a particular document was relevant to a query.

It is reasonably straightforward to adapt these inputs for the purposes of automated image tagging evaluation. In terms of image annotation performance we output a `.qrel` and `.top` file with the image numbers as the query identifiers and the word numbers as the document identifiers. In contrast, for retrieval, we want a ranked list of the images for a particular query or

---

<sup>14</sup>[http://trec.nist.gov/trec\\_eval/](http://trec.nist.gov/trec_eval/)

word. Here we therefore set the words to be the TREC query identifiers and the image numbers to be the corresponding document identifiers, sorted in decreasing order of probability per query.

As mentioned in Section 3.5.1, the `trec_eval` function has been integrated into the CRM framework. The CRM evaluation framework outputs the required `.qrel` and `.top` files and calls the `trec_eval` executable. The output of `trec_eval` is saved to a text file which is then parsed by the CRM evaluation framework with the parameter (e.g. MAP) being used as the optimization criterion for cross-validation being retrieved and used to drive the optimization of the model.

### 3.5.3 Custom Evaluation Functions

`Trec_eval` is perfect for evaluating the image retrieval side of image tagging generating those metrics that are used in most research papers on the topic. Unfortunately the most popular method of image annotation evaluation, namely mean per word recall, mean per word precision and the number of words with recall greater than zero are not calculated by `trec_eval`. The author therefore created a custom evaluation model within the CRM evaluation framework to calculate these metrics.

# Chapter 4

## Evaluation

In this Chapter we will seek to thoroughly evaluate the performance of the CRM and BS-CRM models as custom implemented for this dissertation. The high-level objectives of the evaluation are as follows:

- Verify that the custom implementation of the CRM model operates as expected given the results published in [34]. This includes both annotation and ranked retrieval performance.
- Measure the performance of the BS-CRM model with different word smoothing functions (Bernoulli, N-CRM, Multinomial, Dirichlet) and annotation lengths (3,4,5 words). This will only encompass annotation performance.
- Test both models on the COREL and PASCAL datasets. This will ensure that the feature pre-processing of the datasets (particularly PASCAL) have been performed to the required standard.

The entire suite of tests in this Chapter have been designed to verify these core objectives. Having performed these tests we will then be able to determine whether or not the original objectives of the dissertation as stated in Chapter 1 have been successfully accomplished. We will now briefly discuss the adopted experimental methodology before conducting a detailed analysis of the results.

### 4.1 Experimental Methodology

The CRM and BS-CRM models are evaluated on the standard COREL and PASCAL datasets both of which have been pre-processed into a feature representation as explained in Section 3.2. For the PASCAL dataset, as the same image can have the same annotation many times (such as a photograph with multiple people), we also binarize the ground truth image annotations to 0 (representing absence) or 1 (represent the presence of the object in the image).

### 4.1.1 COREL Dataset

For the COREL dataset we will present the experimental results of applying Normalized CRM (N-CRM), Bernoulli (B-CRM), Multinomial (M-CRM) and Dirichlet (D-CRM) word smoothing to the CRM and BS-CRM models. Given that our overriding goal is to capture keyword correlation efficiently, we hypothesise that the nature of the word smoothing function used will have a significant impact on the results. It is therefore interesting to investigate as wide a range of functions as possible to ascertain their effect on the model performance. An annotation length of 5 keywords will be investigated for all smoothing models. Furthermore in the case of the N-CRM model we will also examine the effect of 3 and 4 keyword annotations.

Before applying the models to the testing dataset we will firstly optimize the models on the validation set. The COREL training dataset of 4500 images is randomly split into 4000 training images and 500 validation images. For the no-beam variant we perform a joint optimization over the  $\beta$  and  $\mu$  or  $\lambda$  parameters as appropriate for the particular smoothing function in question. The best  $\beta$  parameter is then kept constant and we vary the smoothing parameter ( $\mu$  or  $\lambda$ ) for beam lengths of 1, 5, 10, 15 and 30 respectively. The best smoothing parameter value for each beam is then recorded. For the N-CRM model only we also parameter optimize the image retrieval performance with 1 word queries, optimizing the  $\beta$  and  $\lambda$  parameters in a separate cross validation stage.

Having tuned the model on the validation set we merge the training and testing images to create a training set of 4500 images. Using this training set we then apply the model (using the best parameters found during cross validation) to the testing 500 image dataset and record the annotation performance using the research standard annotation evaluation metrics as discussed in Chapter 2.5.

For the N-CRM model we also present the image retrieval results on 1, 2, 3 and 4 word queries on the test set, which in combination with the CRM annotation performance, will allow us to verify that the CRM model performs at the level expected given the results in the literature. Finally examples will also be given of both the original CRM and BS-CRM annotated keywords against the ground truth keywords so that we may easily visualize the effects of the model with and without beam search.

### 4.1.2 PASCAL Dataset

The performance of the N-CRM model with and without beam search will be investigated on the PASCAL dataset for an annotation length of 5 words. The testing methodology adopted for this dataset will closely follow that of the COREL dataset. A first stage of parameter optimization will be conducted. In this case we will use the PASCAL standard splits of the training set into 2501 training images and 2510 validation images. Having found the optimal parameters

the training and validation sets will then be merged into a training dataset consisting of 5011 images with which we use with the model to annotate the 4952 testing images. Furthermore as the actual images themselves are not subject to copyright restrictions we will also display a selection of images against their system generated annotations.

## 4.2 COREL: N-CRM Model

In this section we report the parameter optimization and test results for the Normalized CRM (N-CRM) model both for annotation (Section 4.2.1) and retrieval (Section 4.2.2) performance.

### 4.2.1 Image Annotation Performance

#### 4.2.1.1 Parameter Optimization

In this Section we optimize the parameters of the CRM model based on an annotation length of 5 as this is the most common annotation length in the literature. Furthermore, as is also standard in the literature, the vocabulary of the COREL dataset has been filtered to contain only those 260 words that occur in the testing dataset [23] [34]. Figure 4.1 presents the results of parameter optimization on this dataset.

For the annotation performance, we optimize on the annotation MAP performing an exhaustive search jointly over the  $\beta$  and  $\mu$  parameters for the original CRM model without beam search. As the dataset is standardized to have a zero mean and unit standard deviation we can expect that the  $\beta$  parameter will be in the region of 1.0. Given the log-linear structure of the CRM model we therefore chose a log scale of  $\beta$  values to sweep, namely: 0.01, 0.03, 0.1, 0.3, 1, 3, 10 and 30. For each of these  $\beta$  values, we traverse through the following range of  $\mu$  values: 5, 10, 15, 30. As we are using the normalized CRM smoothing function we must have a minimum  $\mu$  value of 5 (the maximum annotation length of any image in the original collection). The maximum  $\mu$  value is theoretically unconstrained, but for computational reasons we limit our search to a ceiling of 30 for this particular parameter.

As the results in Figure 4.1(a) demonstrate, the CRM model has a peak MAP at 0.30780 for  $\beta = 1$  and  $\mu = 5$ . Holding the  $\beta$  constant at 1.0 we then search through the  $\mu$  parameters for beam widths of 1, 5, 10, 15 and 30. Here we make the, not unreasonable assumption, that the  $\beta$  parameter, which has the most direct effect on the distribution of image features, will not have as significant an effect on the word smoothing distribution of the CRM model. Thus we will only optimize on  $\mu$  for each beam width whilst holding  $\beta$  constant. Figure 4.1(b) illustrates that the best  $\mu$  for all beam widths is 5.

We now fix the CRM model parameters at a  $\beta$  of 1.0 and a  $\mu$  of 5 and examine the performance of the model on the held out test set for annotation lengths of 3, 4 and 5.

#### 4.2.1.2 Annotation Length 5

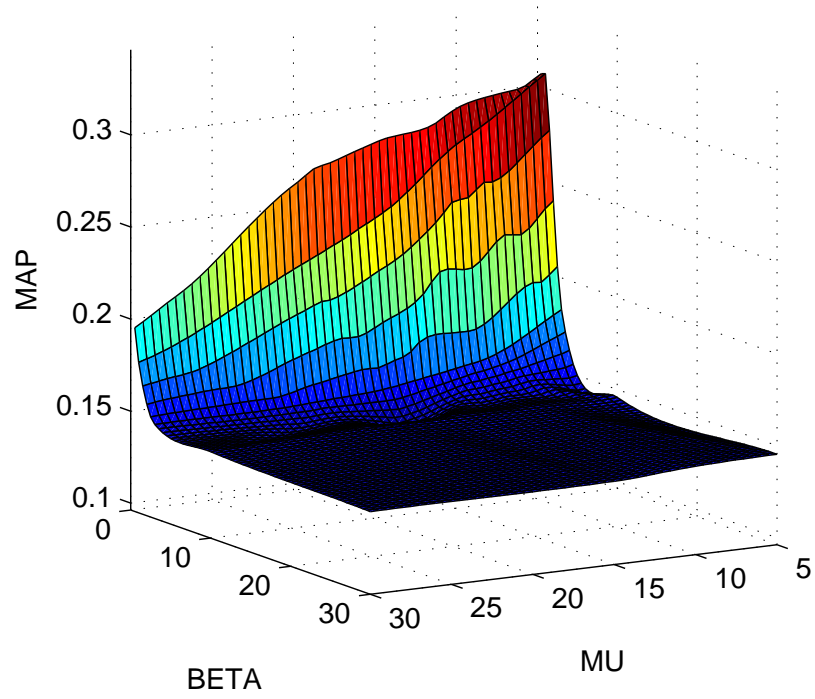
Figure 4.2 and Table 4.1 hold the results of applying the N-CRM model to the test set. The first point of note is that the CRM model (without beam search) designed for this dissertation performs as expected given that the mean per word recall and precision are in close proximity to the original model. The mean per word recall is 0.184 and precision is 0.197 for the custom solution compared to 0.190 and 0.16 for the original CRM model. Having demonstrated the annotation capability of the custom developed CRM model all that remains to be done to prove its correctness is to test ranked retrieval performance. We will perform the ranked retrieval test in Section 4.2.2.

Examining the BS-CRM results in Figure 4.2 and Table 4.2 we observe that the model outperforms a selection of the state-of-the-art models in the literature. The performance on the model appears to depend on the beam width selected and peaks at a value of 15 with an F1 measure of 0.20612. Over the original CRM model as published by Lavrenko et al.[34] the BS-CRM model achieves a 6.8% increase in mean per word recall and a 31.0% increase in mean per word precision. For the top words the recall increases by 3.4% and the precision by 28.0%. The rise in the number of words with a recall greater than zero is also significant (97 to 114 a rise of 6.5%), which demonstrates that the BS-CRM model is able to “promote” rarer words to the keyword set through using keyword correlation.

Compared to the keyword correlation model of Zhou et al. [64] the BS-CRM model achieves a 9.1% increase in mean per word recall and an increase of 6.3% increase in mean per word precision. This result demonstrates that our methodology of capturing keyword correlation is more effective on the COREL dataset than that proposed by Zhou et al. Table 4.3 illustrates the actual labels assigned by the BS-CRM and CRM models to an example subset of the COREL images against the manually assigned ground truth. In this table we can easily visualize the operation of the BS-CRM model and how it is able to increase annotation accuracy by eliminating a selection of noisy keywords.

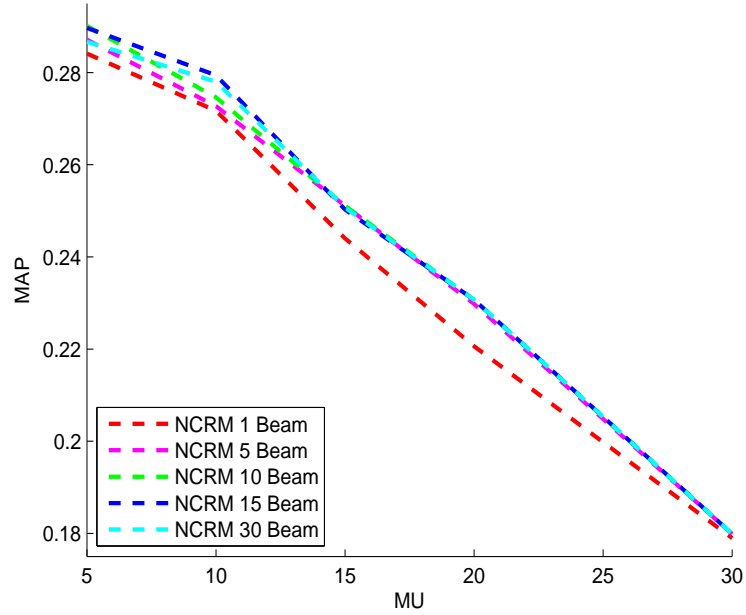
Following the authors of the CMRM [23] we also present the mean per word recall and precision for 70 selected COREL words in the bar charts of Figure 4.3 and Figure 4.4. Here we can obtain a detailed overview of how the BS-CRM model compared to the original CRM model on individual words. The per word precision in bar chart 4.3 for many of the BS-CRM words exceeds that of the CRM model (mean precision is 0.44 for the BS-CRM and 0.41 for the CRM). However for recall performance (Figure 4.4), due to the subset of keywords selected, the CRM model outperforms the BS-CRM model with an average recall of 0.46 over these 70 words compared to 0.42 for the BS-CRM model. This latter observation can be explained due to the fact that, by selecting these 70 words, we effectively eliminate the “rarer” words that were annotated by the BS-CRM and not the CRM model thus reducing the observed performance.

N-CRM (No Beam) annotation MAP for varying MU and BETA



(a)

Plot of annotation MAP vs. MU for BS-NCRM (BETA=1)



(b)

Figure 4.1: Figure 4.1(a) illustrates the joint optimization of the kernel bandwidth  $\beta$  and  $\mu$  value for annotation MAP performance on the validation COREL dataset for an annotation length of 5. The surface reaches a maximum of 0.308 at  $\beta = 1$  and  $\mu = 5$ . In Figure 4.1(b) we hold the  $\beta$  parameter constant at 1.0 and optimize  $\mu$  for varying lengths of Beams 1,5,10,15,30 for an annotation length of 5. The best value of  $\mu$  for all beam widths is found to be 5.

<b>N-CRM (Annotation Length=5)</b>	<b>No Beam</b>	<b>Beam 1</b>	<b>Beam 5</b>	<b>Beam 10</b>	<b>Beam 15</b>	<b>Beam 30</b>
Mean Per Word Recall	0.184	0.183	0.196	0.200	0.203	0.202
Mean Per Word Precision	0.197	0.208	0.216	0.211	0.209	0.198
Words with Recall > 0	97	108	113	114	114	112
F1-Measure	0.190	0.195	0.206	0.206	0.206	0.200
Mean Per Word Recall (top words)	0.693	N/A	0.710	0.725	0.724	0.723
Mean Per Word Precision (top words)	0.739	0.767	0.805	0.771	0.755	0.720

Table 4.1: N-CRM model performance on the COREL testing dataset for differing beam widths. The BS-CRM model dominates the original CRM over all beam widths, with the peak performance occurring at a beam width of 15 which yields an F1 measure of 0.20612 which is 8.5% above the 0.190 value obtained by the CRM model.



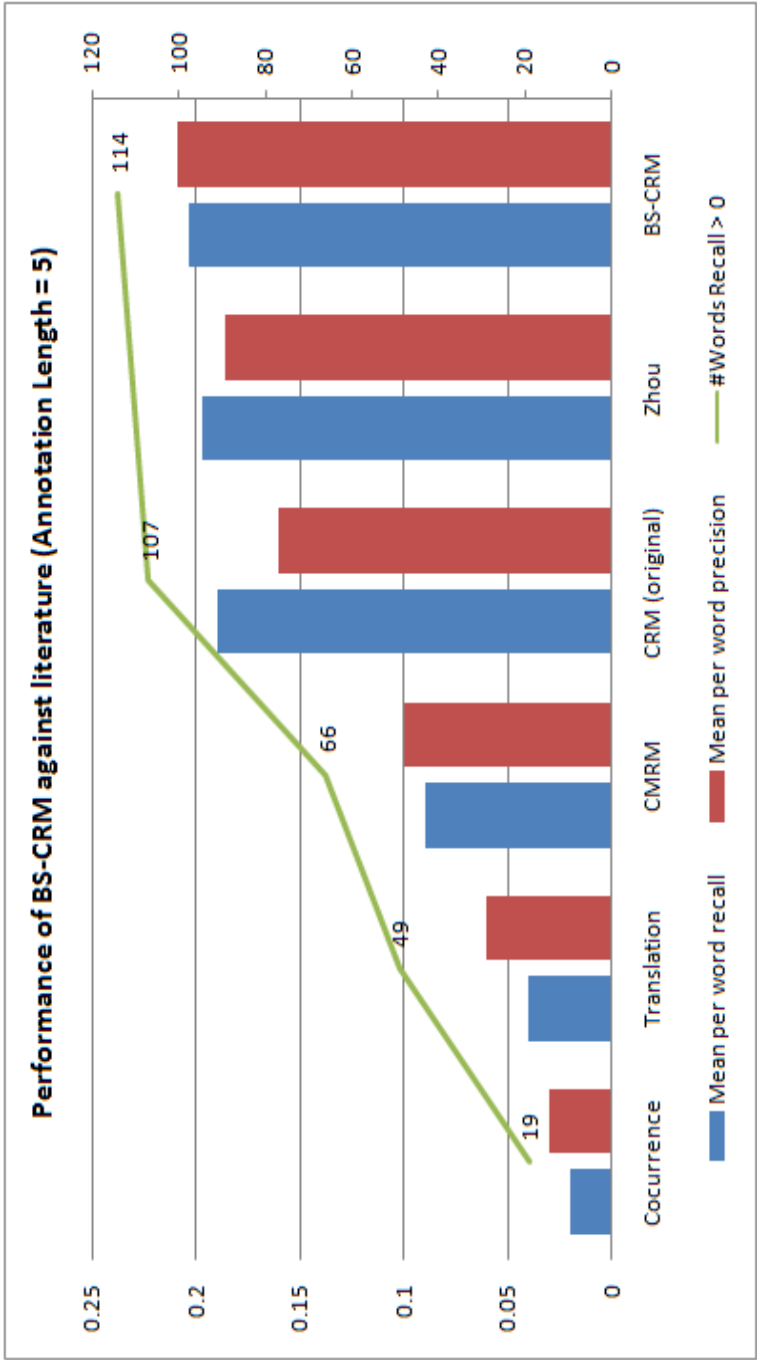


Figure 4.2: This chart depicts the performance of the BS-CRM model against the state-of-the-art image tagging models in the literature. The BS-CRM model clearly exceeds all models in performance on the COREL dataset.

<b>N-CRM (Annotation Length=5)</b>	<b>Co-occurrence</b>	<b>Translation</b>	<b>CMRM</b>	<b>Original CRM</b>	<b>Zhou [64]</b>	<b>BS-CRM</b>
Mean Per Word Recall	0.020	0.040	0.090	0.190	0.186	0.203
Mean Per Word Precision	0.030	0.060	0.100	0.160	0.197	0.209
Words with Recall > 0	19	49	66	107	N/A	114
F1-Measure	0.020	0.050	0.090	0.170	0.190	0.206
Mean Per Word Recall (top words)	N/A	0.340	0.480	0.700	N/A	0.724
Mean Per Word Precision (top words)	N/A	0.200	0.400	0.590	N/A	0.755

Table 4.2: Comparing the BS-CRM model developed in this dissertation against the state-of-the-art results from the literature. Over the original CRM model the BS-CRM model achieves a 6.8% increase in mean per word recall and a 31.0% increase in mean per word precision with an increase of 6.5% in the number of words with recall greater than zero. For the top words the recall increases by 3.4% and the precision by 28.0%. Compared to the keyword correlation model of Zhou et al. [64] the BS-CRM model achieves a 9.1% increase in mean per word recall and an increase of 6.3% in mean per word precision.

Image ID: 100067	BS-CRM	CRM	Ground Truth
	grass water bear grizzly meadow	grass tree water bear field	grass bear meadow grizzly
Image ID: 104082	BS-CRM	CRM	Ground Truth
	grass water antlers caribou bulls	grass water tree deer white-tailed	grass tundra caribou
Image ID: 13092	BS-CRM	CRM	Ground Truth
	flowers water branch petals sky	flowers water branch petals people	sky flowers stems
Image ID: 130062	BS-CRM	CRM	Ground Truth
	grass tree water plane zebra	grass tree cars tracks prototype	grass birds plane zebra

Table 4.3: This table demonstrates the actual labels assigned to some of the COREL test set images (actual COREL image id's are shown). It is clear that the BS-CRM model is able to eliminate some of the noisy keywords produced by the CRM model to increase annotation accuracy. For example, consider the first image. Here we see that the BS-CRM model selects “grizzly and meadow” as more correlated to the existing labels of “bear, water, grass” than are “tree and field”. Eliminating the noisy keywords of “tree and field” enables the BS-CRM model to perfectly label this particular image in comparison to the CRM.

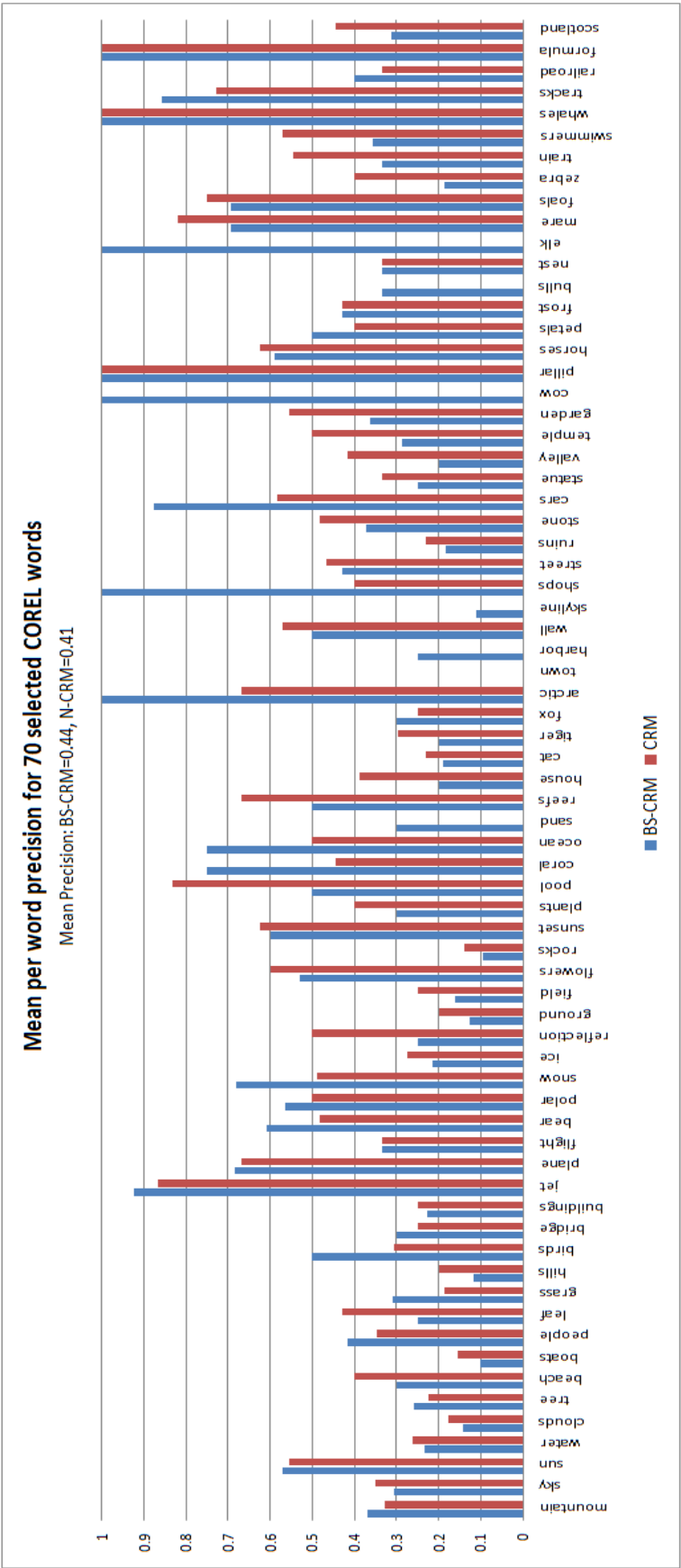


Figure 4.3: This chart depicts the mean per word precision for 70 COREL words as obtained by applying the CRM and BS-CRM models to the testing dataset with an annotation length of 5.

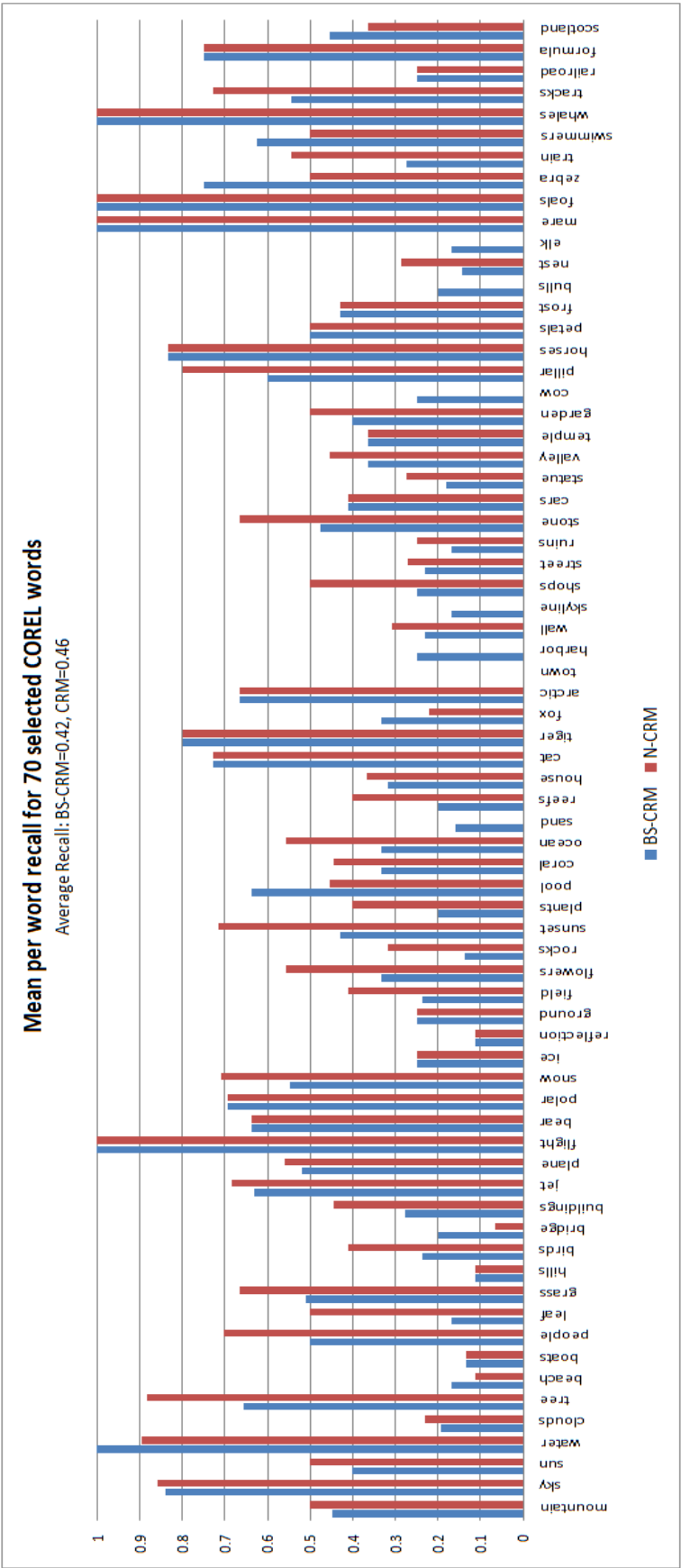


Figure 4.4: This chart depicts the mean per word recall for 70 COREL words as obtained by applying the CRM and BS-CRM models to the testing dataset with an annotation length of 5.

#### 4.2.1.3 Annotation Length 4

In this Section we reduce the number of words that the N-CRM tags a particular image with from 5 to 4 and evaluate the model performance with and without beam search. As for an annotation length of 5, the parameters of the model for this particular test were set to  $\beta = 1.0$  and  $\mu = 5$ .

The results on the testing set are presented in Table 4.4. Again we can see a definite increase in all three of the key annotation performance measures through the use of the BS-CRM model. In this particular case a beam width of 5 results in the best performance giving a 8.7% increase in precision and a 20.3% increase in recall with a 23.1% increase in the number of words with recall greater than zero compared to the custom implemented CRM algorithm.

The general recall and precision results obtained for the shorter annotation length concord well with our expectations. Theoretically we can expect the number of words in the annotation to have a direct influence on the recall and precision of the system, with shorter annotations leading to higher precision and lower recall, since fewer images will be annotated with any given word. In our results we find that, for the non beam CRM model with an annotation length of 5, the recall drops to 0.158 from a high of 0.184 and the precision rises from 0.197 to 0.200.

We can also observe a rather interesting result in Figure 4.5 which measures precision and recall performance (F1 measure) against beam width. Here we can see that the BS-CRM model performance is particularly sensitive to the beam width with an optimum width existing (in this case 5). Furthermore, it would appear that excessively long beam widths, for example 15 and 30, do not necessarily increase performance, suggesting that it may not be worth the extra computational effort of investigating beam widths of above 15.

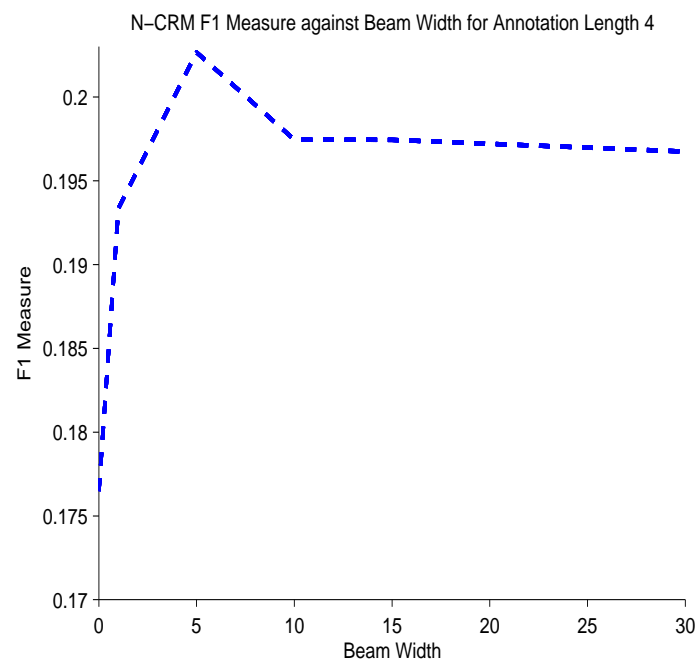


Figure 4.5: Chart depicting the effect of beam width on F1 measure for the N-CRM model with annotation length of 4.

<b>N-CRM (Annotation Length=4)</b>	<b>No Beam</b>	<b>Beam 1</b>	<b>Beam 5</b>	<b>Beam 10</b>	<b>Beam 15</b>	<b>Beam 30</b>
Mean Per Word Recall	0.158	0.180	0.190	0.188	0.190	0.190
Mean Per Word Precision	0.200	0.210	0.220	0.210	0.206	0.205
Words with Recall > 0	91	107	112	110	111	111
F1-Measure	0.176	0.193	0.203	0.197	0.197	0.197
Mean Per Word Recall (top words)	N/A	N/A	0.689	0.685	0.687	0.687
Mean Per Word Precision (top words)	0.763	0.768	0.803	0.776	0.764	0.763

Table 4.4: N-CRM model performance on the COREL testing dataset (annotation length=4) for differing beam widths. Again we reap a clear performance gain by using the BS-CRM model with an increase in F1 measure of 14.8% (at a beam width of 5) over the CRM model.



#### 4.2.1.4 Annotation Length 3

Finally we evaluate the performance of the N-CRM model with an annotation length of 3. Again the model parameters were set at  $\beta = 1.0$  and  $\mu = 5$  for this experiment. The results on the test set are displayed in Table 4.5. Here we again find that the beam width has a significant impact on performance with a beam width of 1 being the best performing in this particular test with a 21.1% gain in mean per word recall, 5.6% in mean word precision and a 15.6% increase in the number of words greater than zero over the CRM model.

Even though the F1 measure remains above that of the CRM model over all beam lengths, we do recognise a drop in mean per word precision below that of the CRM model for beams of 5, 10, 15 and 30. Furthermore at this particular annotation length any beam of 10 and over yield the same performance statistics suggesting that no further significant gains can be made on higher beam widths. This observation concords with that made in Section 4.2.1.3 for the N-CRM model with an annotation length of 4. Here we also found that there was an optimum beam width with decreasing returns realised for widths over 15.

Given the results presented in this Section and in Sections 4.2.1.3 and 4.2.1.2 for annotation lengths of 4 and 5 respectively, we see an increase in the performance (in terms of F1 measure) of the BS-CRM model over the CRM as the annotation length decreases from 5 (8.6%) to 4 (14.8%) and 3 (14.6%). We posit that this performance increase directly relates to the nature of the COREL dataset where it is more common for images to have 3-4 closely related salient objects in this dataset.

In addition we also have to keep in mind that word-to-word correlation generally has little impact on the very top-ranked words that have been determined by the image features with high confidence. The correlation measure is much more effective in retrieving those words not ranked at the very top by the image features. In this case the BS-CRM model effectively “promotes” the words that are more consistent with the very top-ranked words and hence the observed performance increase from 3 to 4 annotation keywords.

Furthermore we can observe the average precision reaches its maximum value (0.220) when the annotation length is set as 4. We suggest that this is because even though a longer annotation length results in more matched words, the number of unmatched words is also increased. Since precision is a ratio of the number of matched words to the total number of generated words, an annotation length of 4 appears to give the best trade-off between these two conflicting factors.

In summary, after examining annotation lengths of 3, 4 and 5 keywords, we have found that the proposed BS-CRM model for automatic image tagging performs consistently better than as the original CRM in all three annotation performance metrics, with the most significant gains being produced at an annotation length of 4 keywords.

<b>N-CRM (Annotation Length=3)</b>	<b>No Beam</b>	<b>Beam 1</b>	<b>Beam 5</b>	<b>Beam 10</b>	<b>Beam 15</b>	<b>Beam 30</b>
Mean Per Word Recall	0.114	0.139	0.142	0.142	0.142	0.142
Mean Per Word Precision	0.182	0.192	0.178	0.179	0.179	0.179
Words with Recall > 0	77	89	92	92	92	92
F1-Measure	0.140	0.161	0.158	0.158	0.158	0.158
Mean Per Word Recall (top words)	N/A	N/A	N/A	N/A	N/A	N/A
Mean Per Word Precision (top words)	0.773	0.772	0.718	0.719	0.719	0.719

Table 4.5: N-CRM model performance on the COREL testing dataset (annotation length=3) for differing beam widths. Performance in this case peaks at a beam width of 1 in comparison to the annotation lengths of 4 and 5 which showed a preference for a wider beam width of 5. Here we recognise an increase in the F1 measure of 14.6% over the CRM model.

### 4.2.2 Ranked Retrieval Performance

In this Section we will evaluate the performance of the N-CRM model on the task of image retrieval. As discussed in Section 2.5, for image retrieval performance we seek to issue queries consisting of 1 or more words and measure the performance on a ranked list of un-annotated images returned by the system. For a given query the relevant images are those that happen to contain all query words in the manual annotation.

For this particular evaluation, we follow the methodology of Lavrenko et al. in their original paper on the CRM model [34]. Since the number of all 3 and 4 word combinations is prohibitively large we filter the COREL vocabulary to have only those words that occur at least twice in the testing dataset. This yields a vocabulary size of 179 words.

As per our evaluation of the annotation performance, a cross-validation step is performed first to determine the appropriate values of the  $\beta$  and  $\mu$  parameters for the model. Here we follow the same methodology as for the annotation performance performing an exhaustive search (guided by MAP) over  $\beta$  values 0.01, 0.03, 0.1, 0.3, 1.0, 10, 30 and  $\mu$  values 5, 10, 15, 30. Figure 4.6(a) charts the variation in MAP as these parameters are adjusted. The best  $\mu$  is found to be 5 and the best  $\beta$  is 1.0 giving a MAP of 0.212 on the validation set.

The retrieval results for the custom N-CRM model on queries of 1, 2, 3 and 4 words are presented in Table 4.6 alongside established baselines and state-of-the-art models. The results we have obtained are very encouraging indeed, being in very close proximity (or slightly better than) original CRM published retrieval results. This result in combination with the CRM annotation performance on an annotation length of 5 words serves to prove that the custom built CRM model operates as expected.

Furthermore, from Table 4.6 it is interesting to observe that the precision at 5 metric is between 0.2 and 0.3 suggesting that there is at least one relevant keyword within the top 5 images returned by the system. This is a particularly good result given that the results are averaged over many queries. Finally from the recall-precision chart in 4.6(b) we can observe that the longer queries are higher performing dominating the shorter queries at all recall levels which accords with our expectations [23].

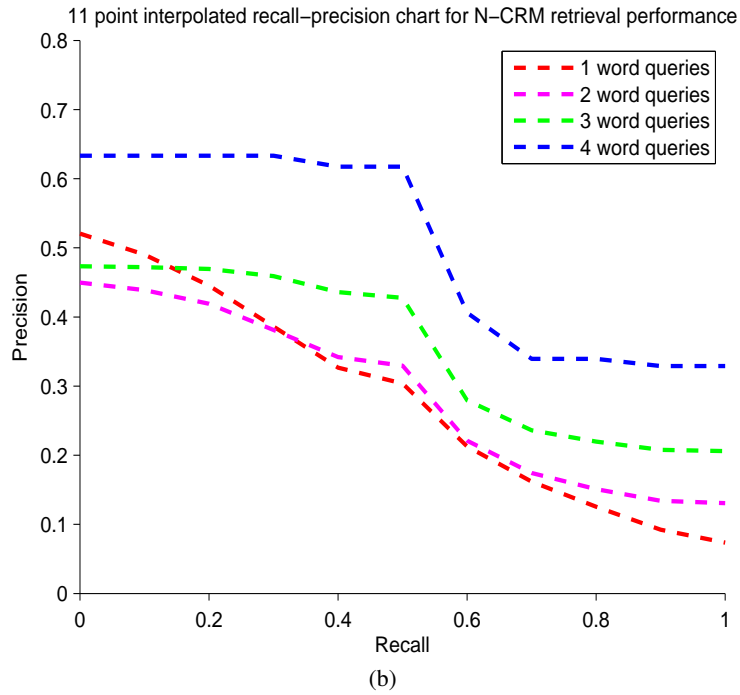
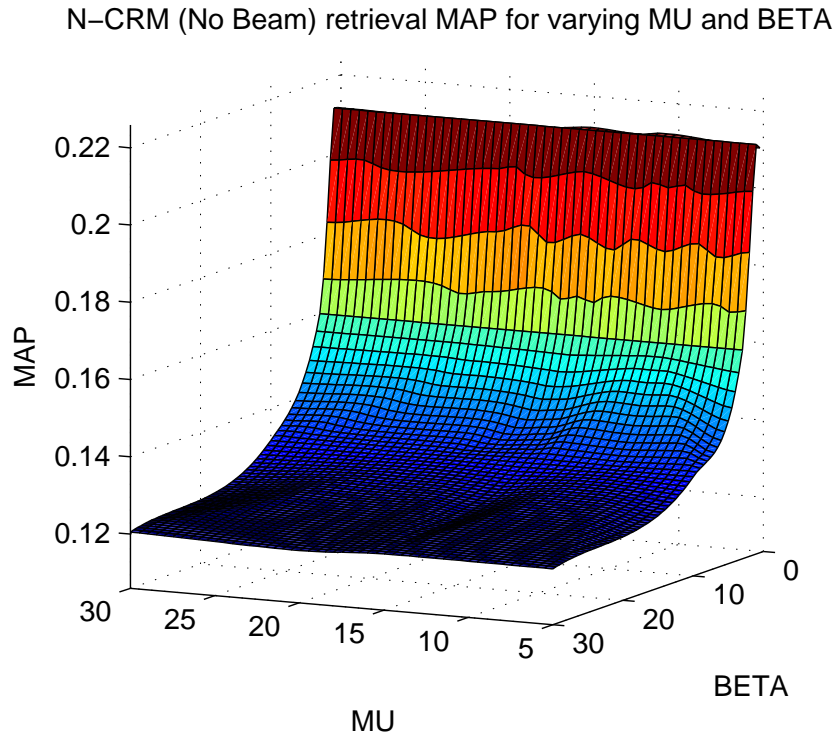


Figure 4.6: Figure 4.6(a) illustrates the optimization of the  $\beta$  and  $\mu$  parameters of the N-CRM model for ranked image retrieval. The surface reaches a maximum at a MAP of 0.212 for  $\mu$  of 5 and  $\beta$  of 1.0. Setting the parameters to these optimum values and applying the model to the test set results in the recall-precision chart depicted in Figure 4.6(b) for 1, 2, 3 and 4 word queries. Comparing this recall-precision chart to that in [23] for the CMRM, we can see that the 1 word query line in our chart dominates that in the paper by a considerable margin for all recall levels.

Query Length	1	2	3	4
Number of queries	179	386	178	24
Relevant Images	1675	1647	542	67
	<b>Precision at 5</b>			
CMRM	0.199	0.131	0.149	0.208
CRM (original)	0.248	0.190	0.189	0.233
N-CRM	0.289	0.214	0.231	0.283
	<b>Mean average precision</b>			
CMRM	0.170	0.164	0.203	0.277
CRM (original)	0.235	0.253	0.315	0.447
N-CRM	0.273	0.275	0.340	0.489

Table 4.6: Table with the image retrieval performance of the N-CRM model engineered for this dissertation against the performance of the original CRM model as published in [34] and the original CMRM model [23]. The custom built N-CRM model outperforms the CMRM model and matches or exceeds the performance of the original CRM model on the task of image retrieval.

### 4.3 COREL: Dirichlet Model

This Section reports on the results of applying the Dirichlet CRM (D-CRM) model with annotation length 5 to the COREL dataset.

#### 4.3.1 Image Annotation Performance

##### 4.3.1.1 Parameter Optimization

The optimization results are displayed in Figure 4.7. The MAP metric peaks at a value of 0.31150 for a  $\beta$  of 1.0 and a  $\mu$  of 1, with a  $\mu$  value of 1 preferred across all beam widths.

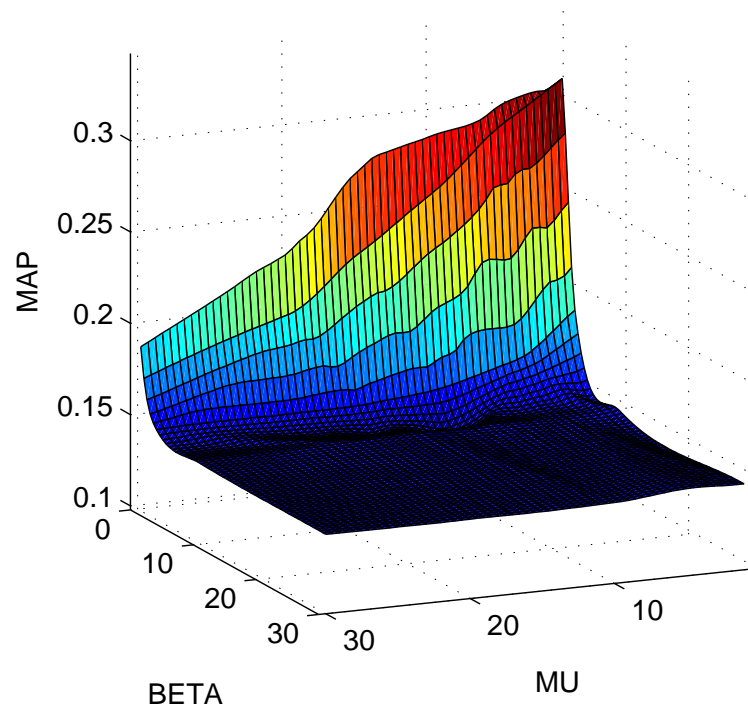
##### 4.3.1.2 Annotation Length 5

The test results are presented in Table 4.7. It is immediately apparent that the performance of beam search with this particular smoothing function is not as pronounced as that of the N-CRM model. Indeed for beam widths of 1,10,15, and 30, the performance actually decreases, with, for example, a beam width of 1 causing a drop in mean per word recall by 11.3% and mean per word precision by 6.7% over the no beam variant of the D-CRM model. Nevertheless as the beam is set 5 the BS-CRM model now outperforms both the normal CRM model and the BS-CRM model with Beam of 1. Here we realise a modest gain in mean per word recall of 1.2% and a gain in precision of 5.9% over the normal CRM model.

The effect of increasing the beam width is particularly interesting for this experiment given that we actually suffer a decline in performance in going from a beam width of 0 to a beam width of 1. This clearly demonstrates the merit underlying the idea of using beam search in the first place: with beam search we consider many different hypotheses for the next best word in parallel only evaluating which is the best set of words at the termination of the algorithm.

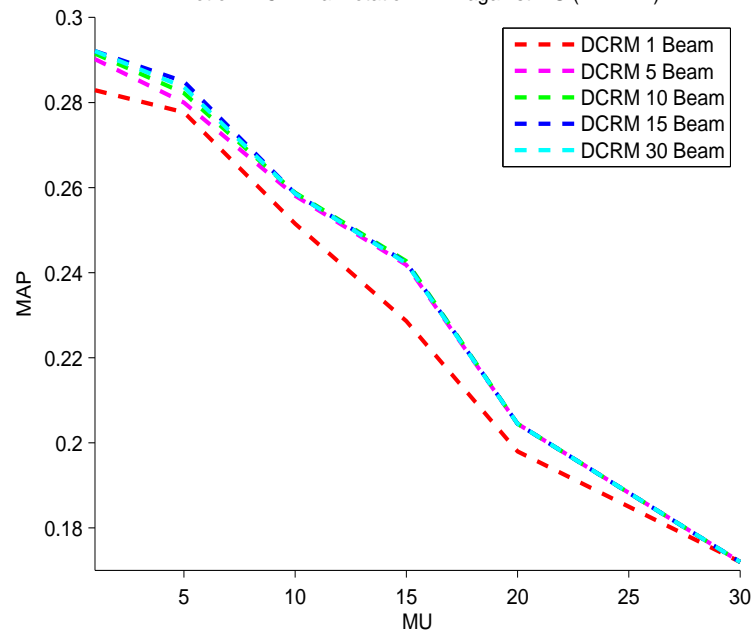
We can therefore explain the performance of the D-CRM model in this case by positing that, for a beam width of 1, the highest probability word added to the set at the second iteration had the effect of causing more noisy keywords to be added later on during execution. For wider beam widths, we not only consider adding the highest probability keyword at the second iteration but also more lower probability words. This clearly has a beneficial effect on performance given the performance increase at a beam width of 5.

D-CRM (No Beam) annotation MAP for varying MU and BETA



(a)

Plot of D-CRM annotation MAP against MU (BETA=1)



(b)

Figure 4.7: Figure 4.7(a) illustrates the optimization of the  $\beta$  and  $\mu$  parameters of the D-CRM model for image annotation. The surface reaches a maximum at a MAP of 0.312 for  $\mu$  of 1 and  $\beta$  of 1.0 Figure 4.7(b) illustrates that a  $\mu$  of 1 is the optimal value for the Dirichlet smoothing function for this dataset across all beam widths.

<b>D-CRM (Annotation Length=5)</b>	<b>No Beam</b>	<b>Beam 1</b>	<b>Beam 5</b>	<b>Beam 10</b>	<b>Beam 15</b>	<b>Beam 30</b>
Mean Per Word Recall	0.191	0.171	0.193	0.185	0.189	0.196
Mean Per Word Precision	0.200	0.187	0.211	0.198	0.198	0.193
Words with Recall > 0	101	103	112	110	110	110
F1-Measure	0.195	0.179	0.201	0.191	0.194	0.194
Mean Per Word Recall (top words)	0.700	N/A	0.701	0.677	0.686	0.696
Mean Per Word Precision (top words)	0.737	0.707	0.787	0.743	0.734	0.704

Table 4.7: Dirichlet CRM (D-CRM) model performance on the COREL testing dataset (annotation length=5) for differing beam widths. Here we find a performance gain over the no-beam model at a width of 5, with a decrease in performance at the remaining beam widths of 1, 10, 15 and 30.



## 4.4 COREL: Multinomial Model

In this section we report on the test results for the Multinomial CRM (M-CRM) model for annotation length 5 on the COREL dataset.

### 4.4.1 Image Annotation Performance

#### 4.4.1.1 Parameter Optimization

The optimization results are displayed in Figure 4.8. The MAP metric peaks at a value of 0.31230 for a  $\beta$  of 1.0 and a  $\lambda$  of 0.9. The optimum  $\lambda$  for beam widths of 1, 5 and 10 is 0.9 and 0.7 for the higher widths of 15 and 30.

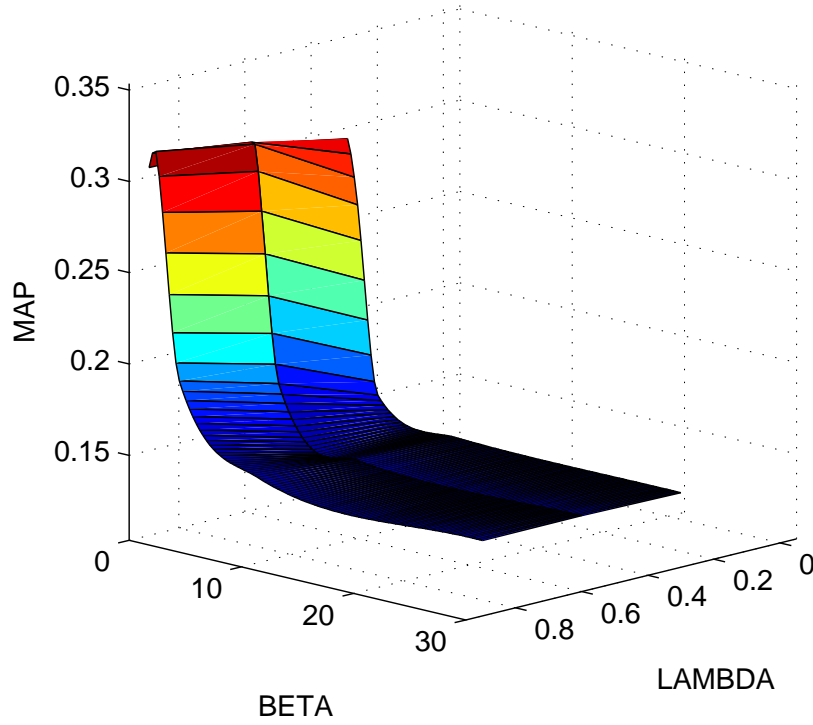
#### 4.4.1.2 Annotation Length 5

The results of applying the M-CRM model to the test set are presented in Table 4.8. As for the D-CRM model we notice a decrease in performance in going from the no beam M-CRM model to a beam width of 1. As for the D-CRM model, we suggest that this might also be a result of constraining the search for correlated keywords to just the top probability keywords which might well introduce more noisy keywords into the set further down the line.

Introducing a wider beam of width 5 can be seen to increase the performance of the CRM model, though again not as much as was observed for the N-CRM model. The peak performance occurs for a beam width of 10 with an F1 measure 5.7% above the CRM model. After a width of 10 the performance clearly falls below that of the no beam model suggesting the greater predominance of noisy keywords being added to the sets for wider beam widths.

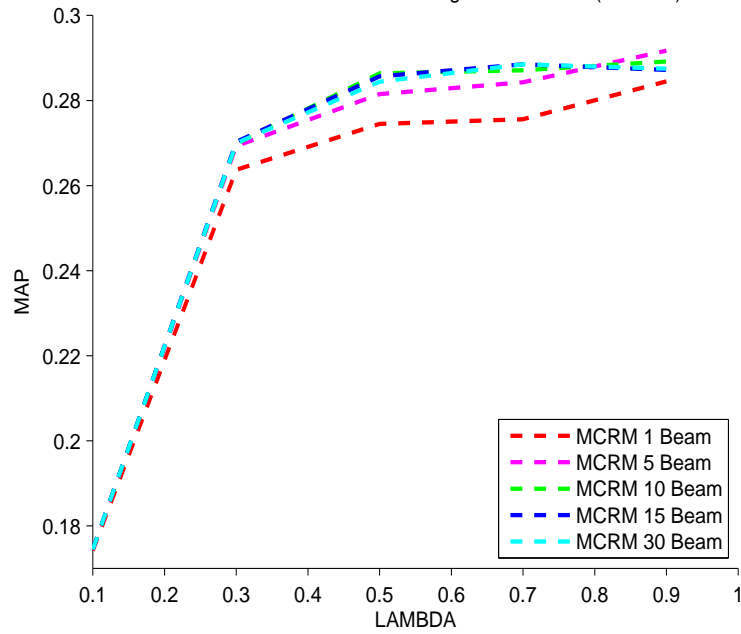
The fact that the N-CRM, M-CRM and D-CRM models all seem to reach peak performance for beam widths of 1-15 is quite suggestive and further re-enforces the emerging observation that higher beam widths are less beneficial to annotation accuracy. A possible reason for this observation is that, in adding lower and lower probability words at very high beam widths, we are effectively increasing the possibility of adding some noisy/unreliable keywords to the set that “attract” (through correlation) more noisy keywords at each iteration thereby swamping the set with irrelevant words.

M-CRM (No Beam) annotation MAP for varying LAMBDA and BETA



(a)

Plot of M-CRM annotation MAP against LAMBDA (BETA=1)



(b)

Figure 4.8: Figure 4.8(a) illustrates the optimization of the  $\beta$  and  $\lambda$  parameters of the M-CRM model for image annotation. The surface reaches a maximum at a MAP of 0.312 for  $\mu$  of 1.0 and  $\lambda$  of 0.9. Optimizing the  $\lambda$  parameter for the BS-CRM model ( $\beta$  held constant at 1.0) yields a value of 0.9 for beam widths 1, 5 and 10, with a width of 0.7 preferred by beams of width 15 and 30 (Figure 4.8(b)).

<b>M-CRM (Annotation Length=5)</b>	<b>No Beam</b>	<b>Beam 1</b>	<b>Beam 5</b>	<b>Beam 10</b>	<b>Beam 15</b>	<b>Beam 30</b>
Mean Per Word Recall	0.196	0.176	0.197	0.218	0.191	0.197
Mean Per Word Precision	0.193	0.186	0.210	0.194	0.187	0.185
Words with Recall > 0	103	104	111	115	106	107
F1-Measure	0.194	0.181	0.204	0.205	0.189	0.191
Mean Per Word Recall (top words)	0.705	N/A	0.710	0.762	0.708	0.709
Mean Per Word Precision (top words)	0.703	0.697	0.776	0.719	0.705	0.680

Table 4.8: Multinomial CRM (M-CRM) model performance on the COREL testing dataset (annotation length=5) for differing beam widths. The F1 measure reaches a maximum at a beam width of 10, with a performance decrease over the CRM model realised at beam widths of 1, 15 and 30.

## 4.5 COREL: Bernoulli Model

We conclude our evaluation of the COREL dataset by testing the Bernoulli word smoothing function with the CRM model (B-CRM) for an annotation length 5.

### 4.5.1 Image Annotation Performance

#### 4.5.1.1 Parameter Optimization

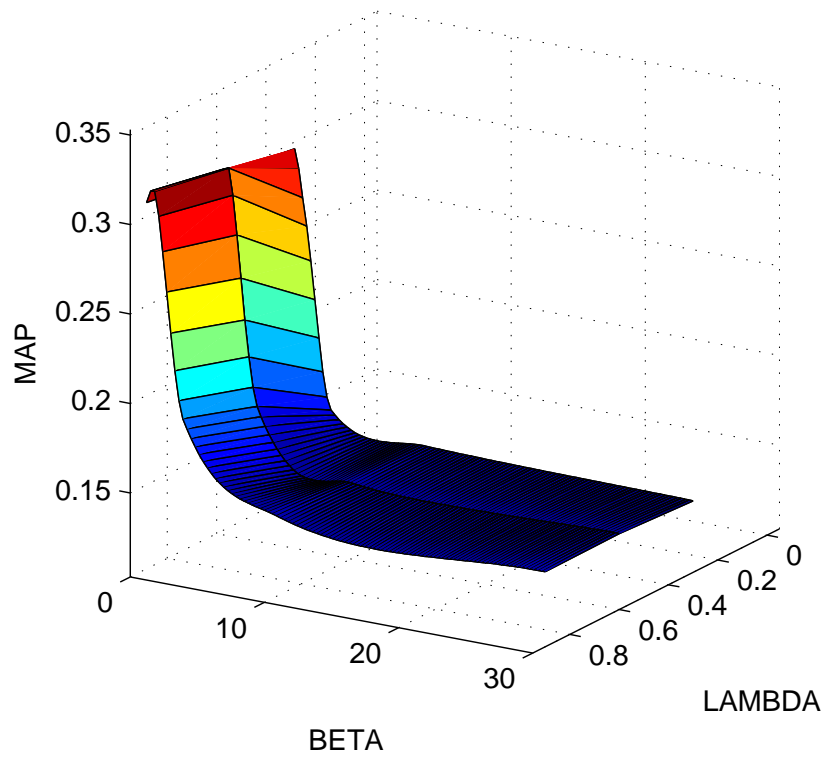
The parameter optimization for the B-CRM model (Figure 4.9(a)) finds a peak MAP at a value of 0.31310 for a  $\beta$  of 1.0 and a  $\lambda$  of 0.9 for widths of 1 and 5, with a value of 0.5 for widths of 10, 15 and 30. In accordance with past experiments we therefore set the model parameters to these values before applying the algorithm to the testing dataset.

#### 4.5.1.2 Annotation Length 5

Comparing the B-CRM model with no beam to that of the M-CRM model with no beam it is clear that we do not particularly see a significant increase in the F1 measure realising only an increase of 2.6% from 0.194 to 0.199. This is rather surprising in a sense as from the arguments in the literature [13] one would come to expect a much larger increase in accuracy given that we are no longer spreading the probability over the length of the annotation as per the multinomial model. One reason for this difference might be due to the rectangular features used by the authors in [13] which they also cite as a contributing factor (in combination with the Bernoulli model) to the observed performance gain. In contrast, the pre-processed COREL dataset of Duygulu et al. used in this dissertation was segmented using the Normalized Cuts algorithm.

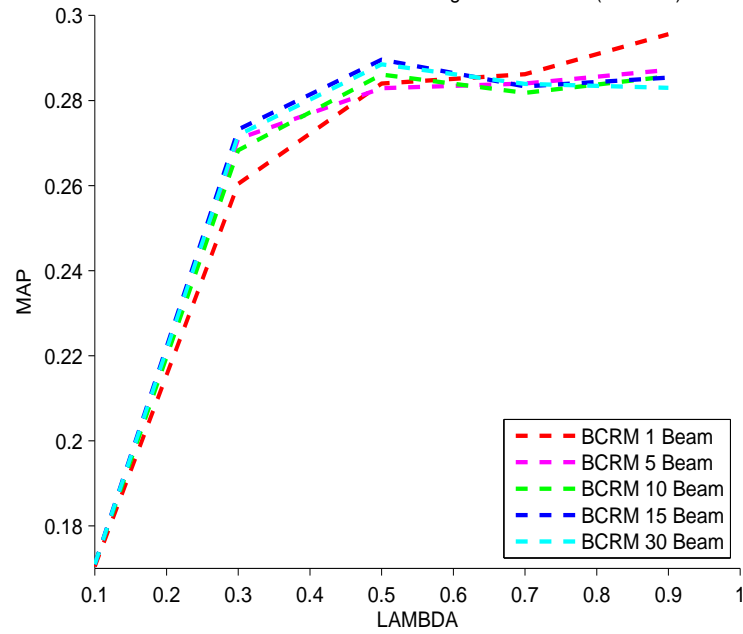
Considering the addition of beam search to this model we observe, once again as for the M-CRM model, a decrease in performance for a beam width of 1, with increasing performance for widths of 5 and 10, the latter width just managing to surpass the no beam model by a minor, if not insignificant, amount in F1 measure (0.5%). This lower performance with a Bernoulli word smoothing function suggests that the benefits of the BS-CRM algorithm highly depends on the function that is ultimately used with the model, a result that is not unexpected given the dependence of the word correlation calculation of the nature and degree of smoothing. Given the results obtained on the N-CRM, M-CRM and D-CRM it would appear that the BS-CRM is at its most effective in combination with the N-CRM smoothing function.

B-CRM (No Beam) annotation MAP for varying LAMBDA and BETA



(a)

Plot of B-CRM annotation MAP against LAMBDA (BETA=1)



(b)

Figure 4.9: Figure 4.9(a) illustrates the optimization of the  $\beta$  and  $\lambda$  parameters of the B-CRM model for image annotation. The surface reaches a maximum at a MAP of 0.313 for  $\beta$  of 1.0 and  $\lambda$  of 0.9. Holding  $\beta$  constant at 1.0, we find an optimal value of  $\lambda$  of 0.9 for beam widths of 1 and 5, with a value of 0.5 for widths of 10, 15 and 30.

<b>B-CRM (Annotation Length=5)</b>	<b>No Beam</b>	<b>Beam 1</b>	<b>Beam 5</b>	<b>Beam 10</b>	<b>Beam 15</b>	<b>Beam 30</b>
Mean Per Word Recall	0.200	0.187	0.195	0.192	0.191	0.191
Mean Per Word Precision	0.198	0.198	0.204	0.209	0.190	0.189
Words with Recall > 0	106	108	111	110	106	105
F1-Measure	0.199	0.192	0.199	0.200	0.190	0.190
Mean Per Word Recall (top words)	0.706	0.675	0.700	0.707	0.707	0.706
Mean Per Word Precision (top words)	0.720	0.740	0.754	0.781	0.708	0.711

Table 4.9: Bernoulli CRM (B-CRM) model performance on the COREL testing dataset (annotation length=5) for differing beam widths. Here we can observe very little performance benefit from using the BS-CRM model with Bernoulli word smoothing.

## 4.6 PASCAL: N-CRM Model

In this section we report the parameter optimization and test results for the Normalized CRM (N-CRM) model on the PASCAL dataset both ranked retrieval (Section 4.6.1) and for annotation (Section 4.6.2).

### 4.6.1 Ranked Retrieval Performance

We will begin our evaluation on the PASCAL dataset by examining the retrieval performance of the algorithm. This mirrors the evaluation performed by all of the research papers that use this dataset and so it will help us to firstly determine whether or not the feature based representation we have chosen and indeed the performance of the CRM model itself is at the standard of the research literature.

#### 4.6.1.1 Parameter Optimization

Figure 4.10 illustrates the result of the parameter optimization step for ranked retrieval. The MAP metric peaks at a value of 0.179 for a  $\mu$  of 9 and a  $\beta$  of 0.3.

#### 4.6.1.2 Ranked Retrieval

PASCAL VOC 2007 participants are evaluated based on the precision/recall curve with the quantitative measure used being the average precision (AP). We therefore follow the evaluation procedure recommended by the competition organizers and sort the images by their probability of containing a particular object. This results in 20 recall-precision charts (Figures 4.11, 4.12, 4.13, 4.14) and 20 average precision values (Table 4.10), one for each word in the vocabulary.

The parameters of the CRM model are set to a  $\mu$  of 9 and a  $\beta$  of 0.3 for this test. From the results in Table 4.10 we can observe that the N-CRM with the custom computed feature representation performs at the level of a model using a similar feature based representation in the literature. Here we compare our average precision results to that of Wang et al. [59] who also use a predominantly colour based representation along with an SVM classifier. Our results are within close proximity to that of Wang et al. which gives credence to the chosen feature representation.

Having shown that the ranked retrieval performance is at the expected level based on the standard evaluation metrics it is now interesting to visualize the top 5 images that are retrieved by the system for some of the classes. In Figures 4.15, 4.16 and 4.17 we display the retrieved images for the horse, person and TV monitor classes respectively. As can be observed from these images, the N-CRM model is able to retrieve at least 2 relevant images for each class peaking at 4 out of 5 correct images for the person class.

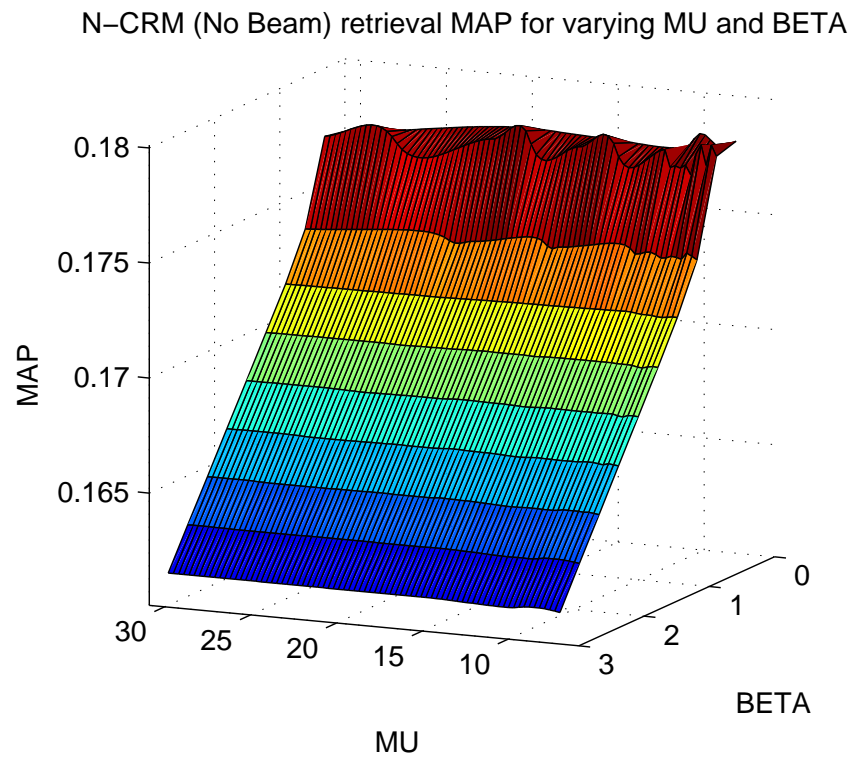


Figure 4.10: This figure illustrates the optimization of the  $\beta$  and  $\mu$  parameters of the N-CRM model for ranked image retrieval. The surface reaches a maximum at a MAP of 0.179 for  $\mu$  of 7 and  $\beta$  of 0.3.



<b>Average Precision</b>	<b>Aeroplane</b>	<b>Bicycle</b>	<b>Bird</b>	<b>Boat</b>	<b>bottle</b>	<b>Bus</b>	<b>Car</b>	<b>Cat</b>	<b>Chair</b>	<b>cow</b>
<b>Wang</b>	0.367	0.124	0.220	0.215	0.112	0.085	0.323	0.134	0.242	0.075
<b>N-CRM</b>	0.425	0.138	0.131	0.280	0.079	0.168	0.323	0.161	0.231	0.087
<b>Average Precision</b>	<b>Table</b>	<b>Dog</b>	<b>Horse</b>	<b>Motorbike</b>	<b>Person</b>	<b>Plant</b>	<b>Sheep</b>	<b>Sofa</b>	<b>Train</b>	<b>monitor</b>
<b>Wang</b>	0.128	0.186	0.442	0.182	0.594	0.146	0.162	0.083	0.243	0.122
<b>N-CRM</b>	0.144	0.184	0.447	0.238	0.582	0.087	0.087	0.089	0.293	0.165

Table 4.10: This table presents the average precision results that have been achieved by the N-CRM model on the PASCAL dataset using the custom computed feature set. As can be observed, the average precision results match that of the results presented by Wang et al. [59] who also predominately use colour features on the same dataset.

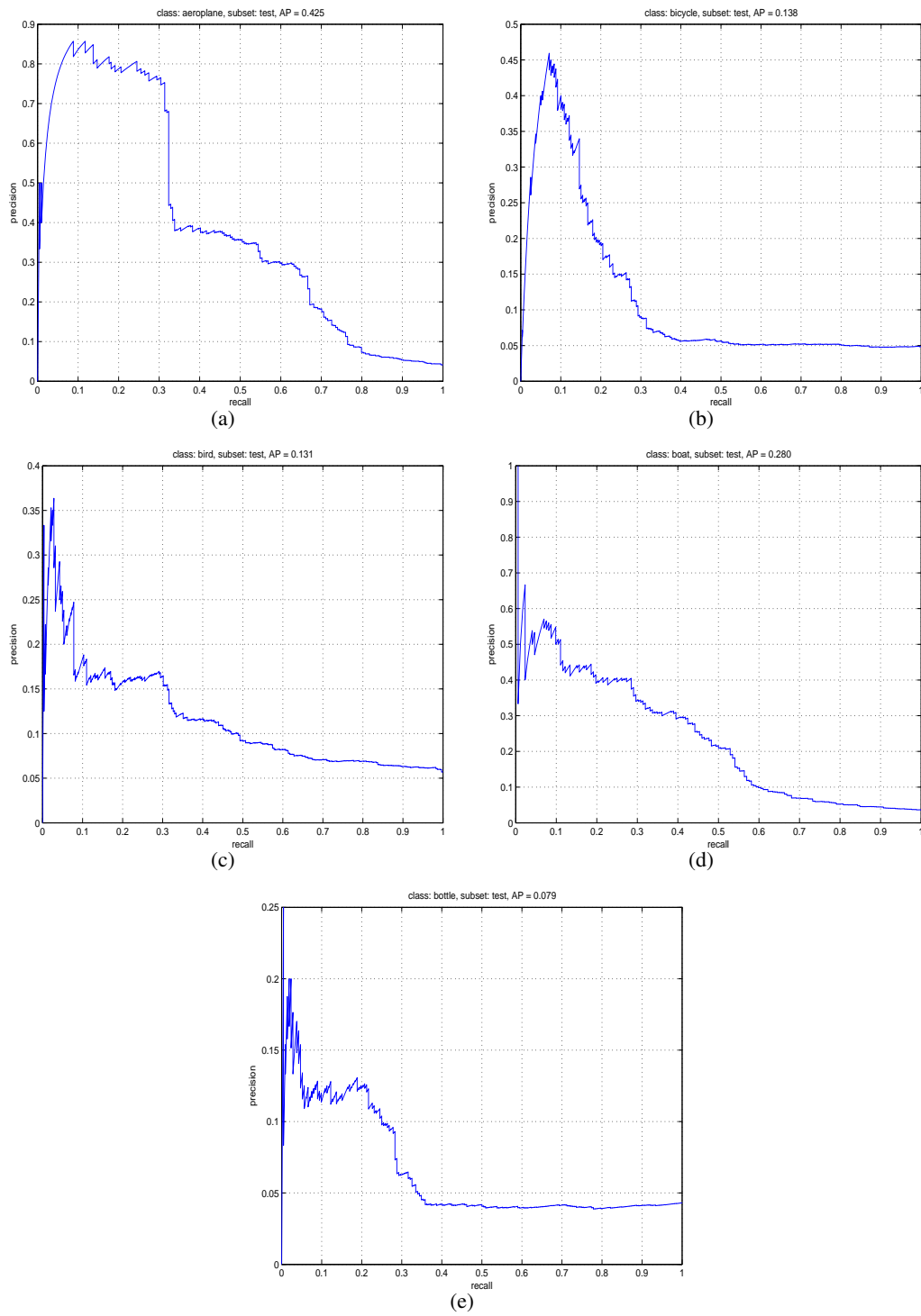


Figure 4.11: Recall-precision charts for the aeroplane, bicycle, bird, boat and bottle classes in the PASCAL VOC 2007 dataset.

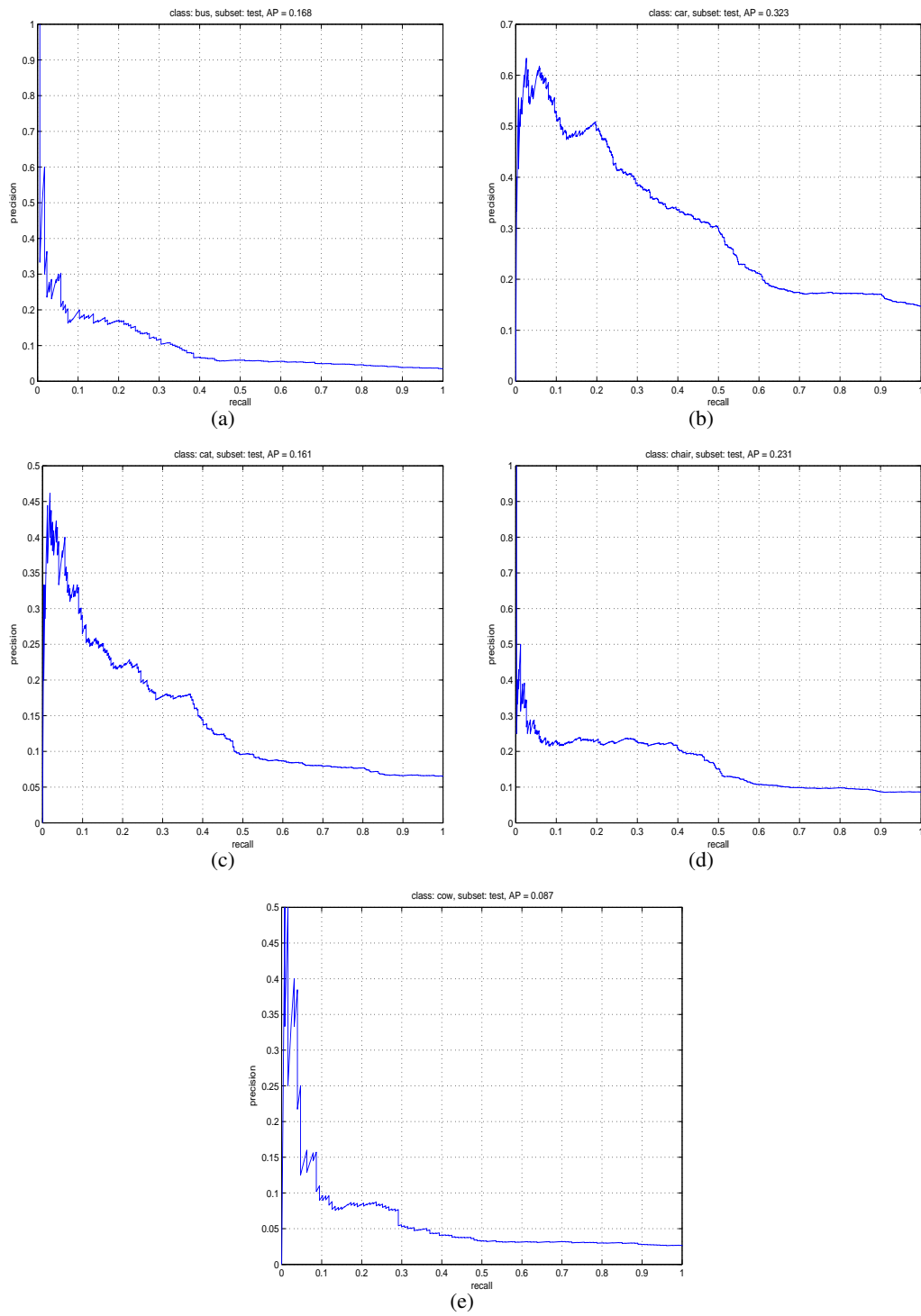


Figure 4.12: Recall-precision charts for the bus, car, cat, chair and cow classes in the PASCAL VOC 2007 dataset.

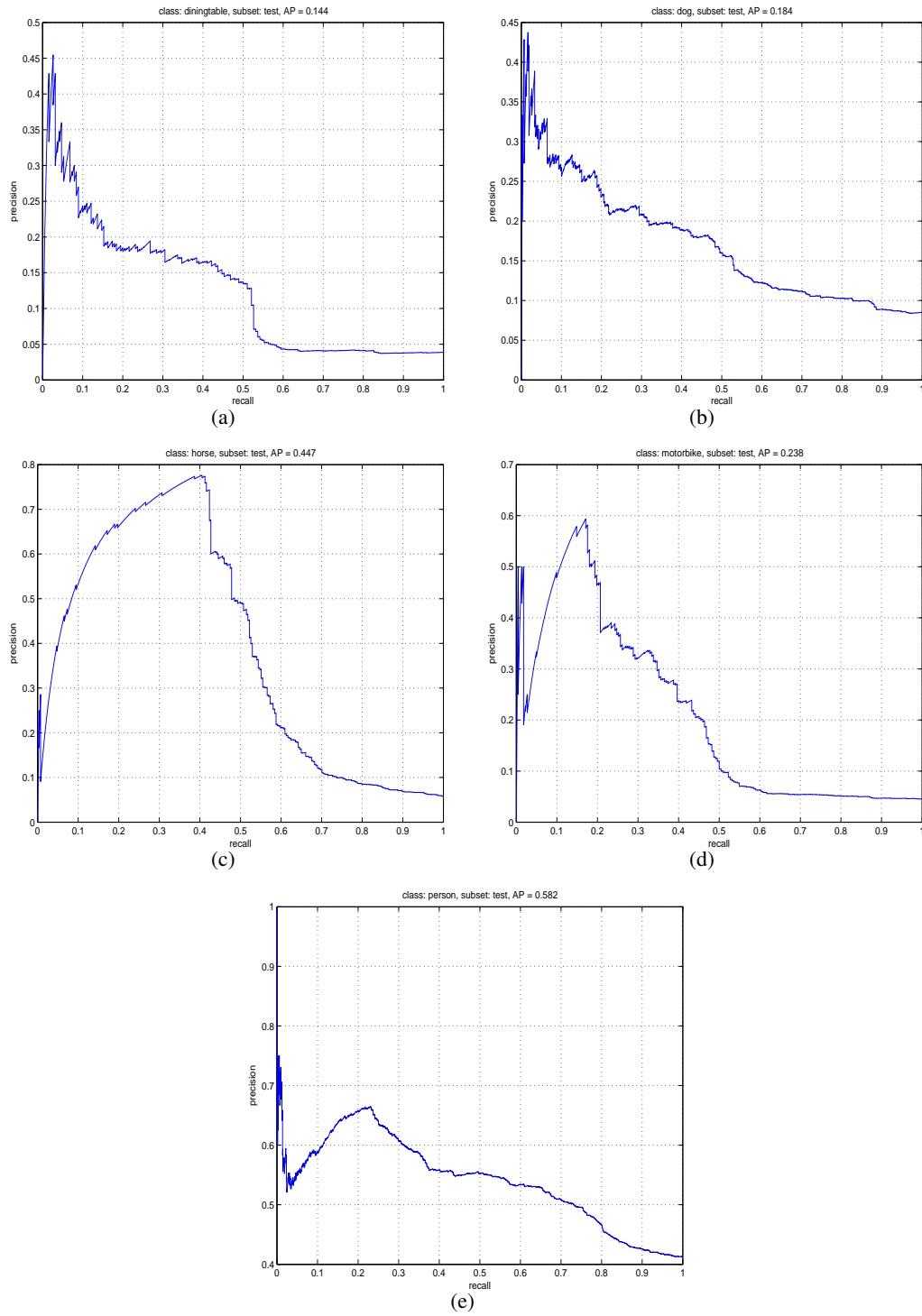


Figure 4.13: Recall-precision charts for the table, dog, horse, motorbike and person classes in the PASCAL VOC 2007 dataset.

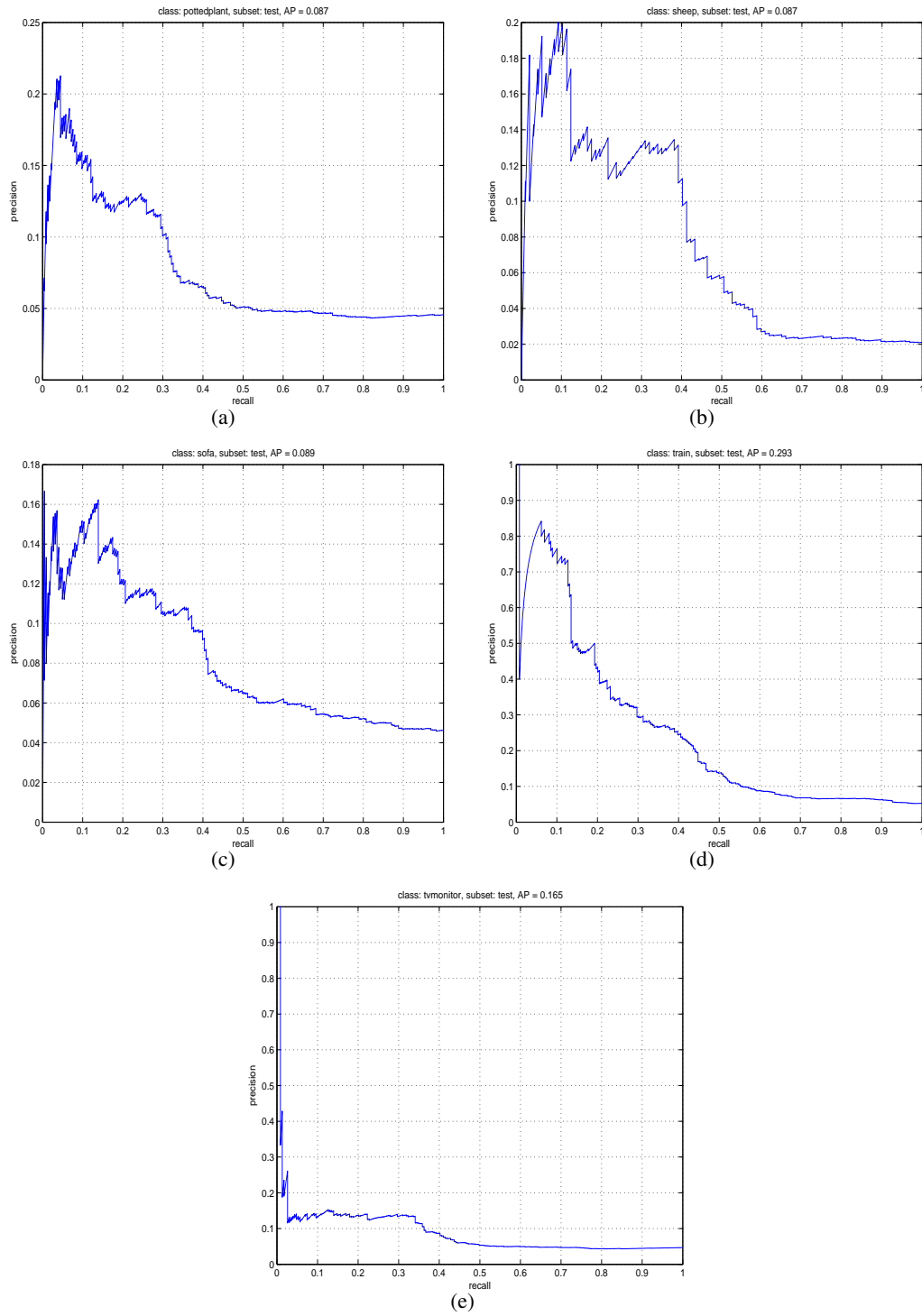


Figure 4.14: Recall-precision charts for the plant, sheep, sofa, train and TV monitor classes in the PASCAL VOC 2007 dataset.

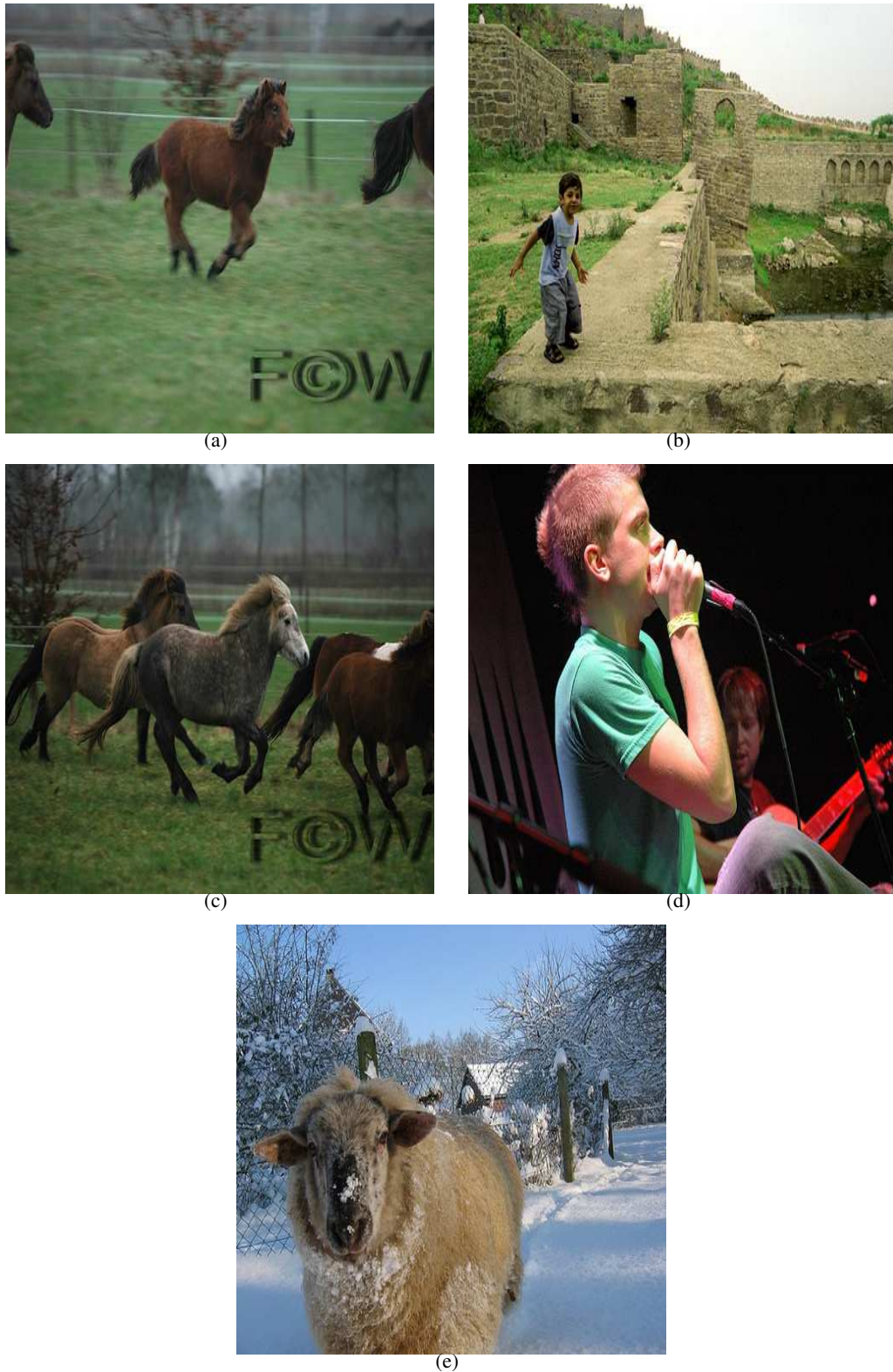


Figure 4.15: *Ranked retrieval results of the N-CRM model on the PASCAL dataset. Here we rank the images according to the probability of a horse occurring in the image and take the top 5. Two of the images returned by the system are relevant to the query which is not unreasonable performance given the basic feature representation that has been used with the model (here two of the images, the image of the sheep and the ruins are of a colour similar to that of a horse which may have confused the predominantly colour based representation we have chosen).*

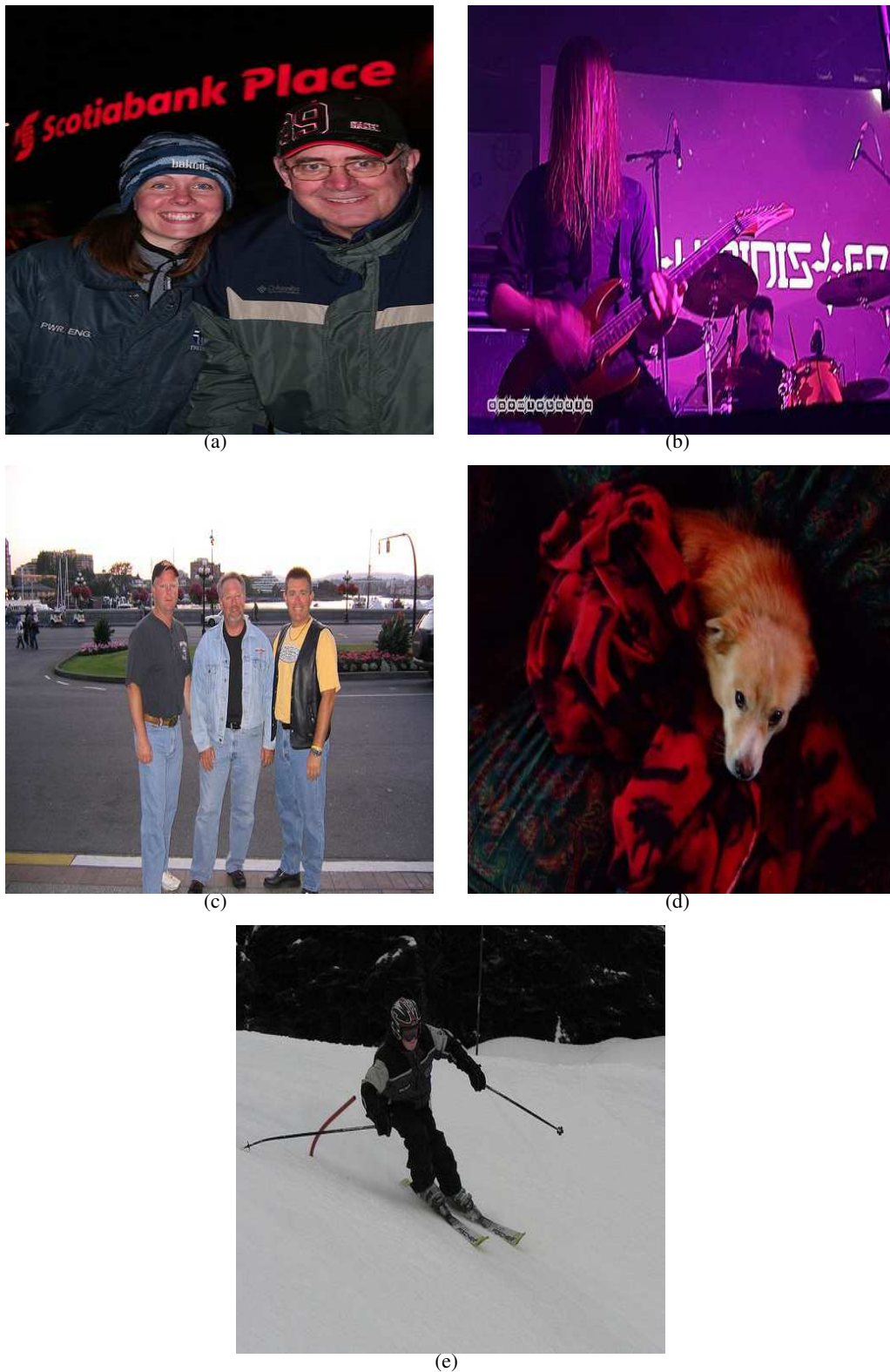


Figure 4.16: *Ranked retrieval results of the N-CRM model on the PASCAL dataset. Here we rank the images according to the probability of a person occurring in the image and take the top 5 (images presented in rank order). The system has returned 4 out of 5 relevant images. This good performance is partly due to the over-representation of the person class in the PASCAL training dataset thereby giving the model many exemplars on which to train.*





Figure 4.17: *Ranked retrieval results of the N-CRM model on the PASCAL dataset. Here we rank the images according to the probability of a TV monitor occurring in the image and take the top 5 (the images are presented here in rank order). This is a particularly valuable test of the system performance given that the TV monitor class is not a majority class (as is person) in the PASCAL dataset. Here we can see that the system has returned 3 out of 5 relevant images.*



## 4.6.2 Image Annotation Performance

### 4.6.2.1 Parameter Optimization

The parameters of the CRM model are optimized as for the COREL dataset with a search over the  $\beta - \mu$  parameter space for the combination that maximize the MAP on the validation set. The optimization results are shown in Figure 4.18(a). Here we find that the surface peaks at a MAP of 0.409 for  $\beta$  of 0.3 and  $\mu$  of 7. Holding the  $\beta$  constant as before we then seek the best value for the word smoothing parameter  $\mu$  for beam widths of 1, 3, 5, 10 and 15. The results of the  $\mu$  parameter optimization for the BS-CRM model are depicted in Figure 4.18(b). Here we can see that a  $\mu$  of 5 is clearly the best performing parameter value with a steep decline in performance realised for higher values. Furthermore the graphs for each  $\mu$  at differing beam widths track each other very closely suggesting that, for this particular dataset, the degree of word smoothing is invariant to different settings of the beam width.

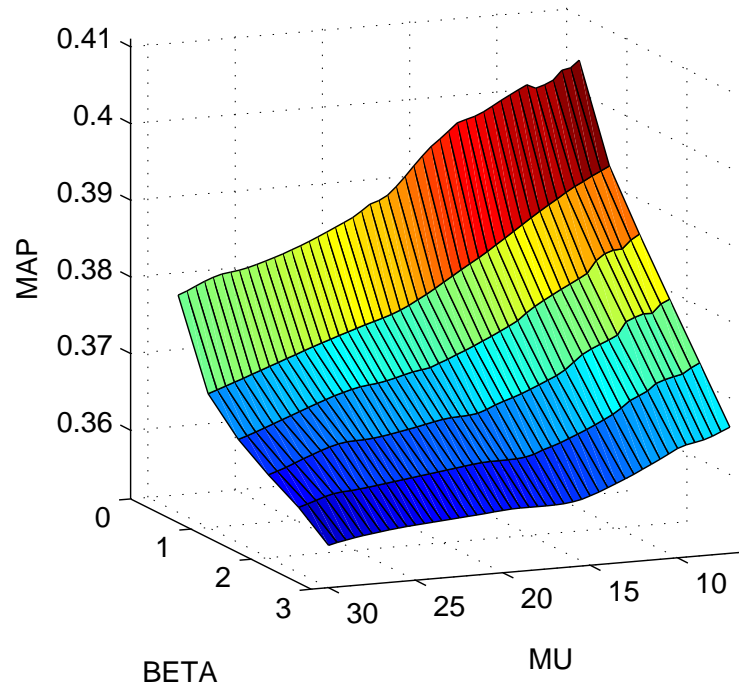
### 4.6.2.2 Annotation Length 5

Table 4.11 presents the annotation performance results for an annotation length of 5 words. Here we can observe a modest increase of 2.4% in the F1 measure over the CRM model. The mean per word precision increases and the mean per word recall falls slightly for the BS-CRM model, but the overall result of the moves in these measure result in the F1 measure increasing. Furthermore we can also observe that for beam widths of 3 and above there is no change in the performance suggesting that there is no reason to search over beams larger than 1-3 for this particular dataset.

There are many possible reasons for this observed performance. Firstly as discussed in Chapter 3 the PASCAL dataset is recognised as being much more challenging than the COREL dataset with objects found in many different poses and scales. Our chosen feature representation is certainly not the best possible to represent salient image regions. These factors have both surely contributed to the more modest performance gain.

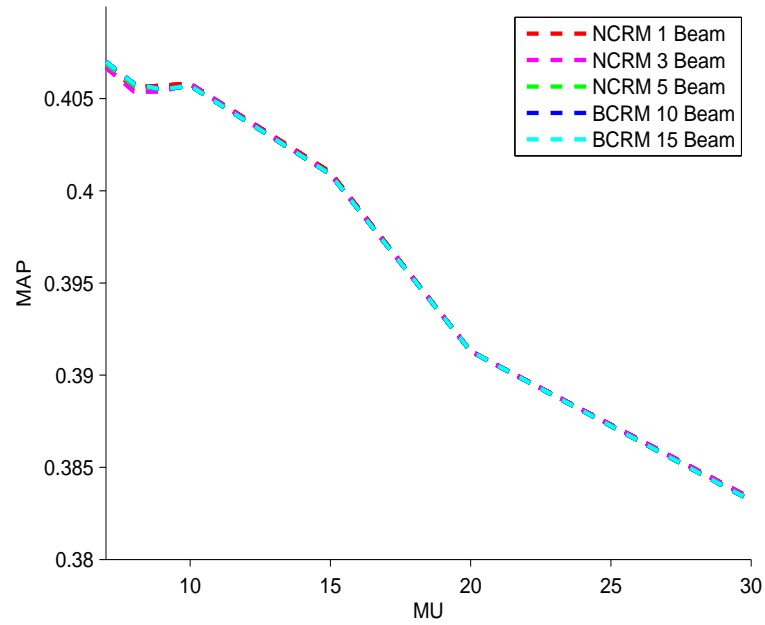
Furthermore we only have 20 keywords for this dataset which is much less than that of COREL, with typically only 1-2 salient objects (such as “person and dog”) appearing in each image (4-5 keyword annotations are quite rare). Thus there is not as great an opportunity compared to the COREL dataset for us to leverage a keyword correlation measure in order to eliminate noisy keywords.

N-CRM (No Beam) annotation MAP for varying MU and BETA



(a)

Plot of N-CRM annotation MAP against MU (BETA=1)



(b)

Figure 4.18: Figure 4.18(a) illustrates the optimization of the  $\beta$  and  $\mu$  parameters of the B-CRM model for image annotation. The surface reaches a maximum at a MAP of 0.409 for  $\beta$  of 0.3 and  $\mu$  of 7. The optimal value of  $\mu$  for the PASCAL dataset is constant at 7 across all beam widths as shown by Figure 4.18(b).

<b>N-CRM (Annotation Length=5)</b>	<b>No Beam</b>	<b>Beam 1</b>	<b>Beam 3</b>	<b>Beam 5</b>	<b>Beam 10</b>	<b>Beam 15</b>
Mean Per Word Recall	0.427	0.416	0.417	0.418	0.418	0.418
Mean Per Word Precision	0.197	0.206	0.206	0.206	0.206	0.206
Words with Recall > 0	20	20	20	20	20	20
F1-Measure	0.270	0.275	0.276	0.276	0.276	0.276
Mean Per Word Recall (top words)	N/A	N/A	N/A	N/A	N/A	N/A
Mean Per Word Precision (top words)	N/A	N/A	N/A	N/A	N/A	N/A

Table 4.11: N-CRM model performance on the PASCAL testing dataset (annotation length=5) for differing beam widths.

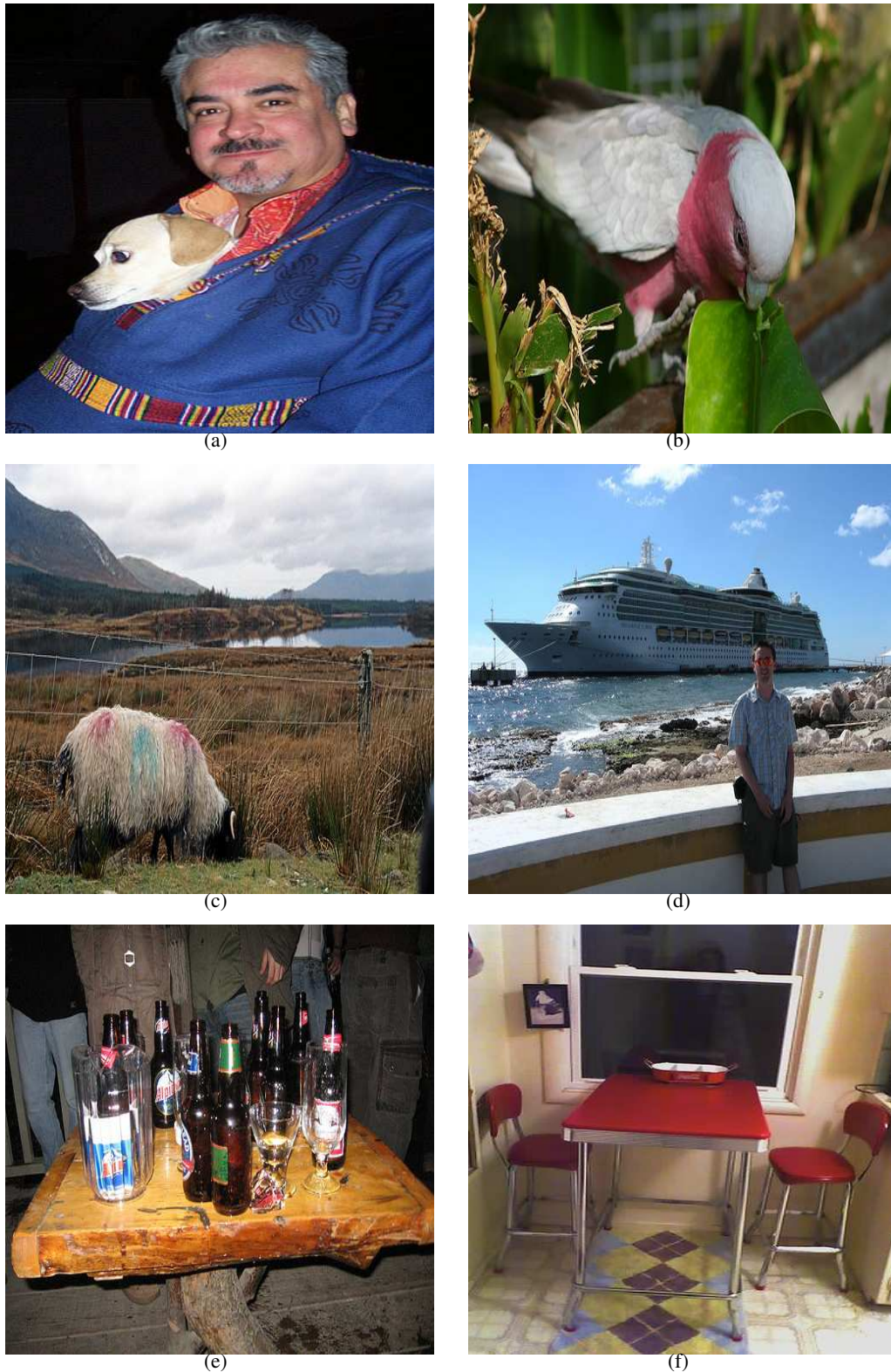


Figure 4.19: Figure 4.19(a) was annotated with the keywords {person, dog, car, chair, sofa}. Figure 4.19(b) was annotated with {person, bird, car, chair, dog}. Figure 4.19(c) was assigned the tags of {person, horse, sheep, car, chair}. Figure 4.19(d) was given the tags {person, boat, car, chair, dog}. Figure 4.19(e) was annotated with {person, chair, pottedplant, bottle, diningtable}. Figure 4.19(f) was tagged with {chair, sofa, car, person, dog}. Here we can see that, in the top 5 system assigned annotations, the N-CRM model was able to find the correct 1-2 words representing the objects in the images.





(a)



(b)



(c)



(d)



(e)



(f)

Figure 4.20: Here we have further successful annotation examples from the PASCAL dataset. Figure 4.20(a) has been assigned the tags of {person, horse, car, chair, dog}. The image in Figure 4.20(b) was given the tags {person, car, diningtable, bottle, chair}. Figure 4.20(c) was annotated with {person, aeroplane, car, chair, dog}. Figure 4.20(d) was tagged with {cow, person, car, chair, dog}. Figure 4.20(e) was annotated with {person, tvmonitor, car, chair, dog}. Figure 4.20(f) was annotated with the keywords {motorbike, person, car, chair, dog}. In this example the model has once again managed to determine the correct object in the image within the set of 5 top probability keywords.



(a)



(b)

Figure 4.21: These example images demonstrate the increase in accuracy obtained using the BS-CRM model on the PASCAL dataset. Figure 4.21(a) was tagged with the keywords {person, dog, car, chair, sofa} by the BS-CRM model. The CRM model tagged the image with {person, dog, car, chair, sheep}. The keyword correlation measure has determined that “sofa” best goes with the existing set {person, dog, car, chair} than does “sheep”. Another example is provided in Figure 4.21(b). Here the BS-CRM model tagged the image with {person, chair, pottedplant, diningtable, bottle}, whilst the CRM model tagged the image with {person, chair, pottedplant, diningtable, bicycle}. Here we observe that the BS-CRM determines that “bottle” best goes with the existing word set than does “bicycle”.

## Chapter 5

# Conclusions and Future Work

### 5.1 Overview

In this dissertation we have investigated the effect of using beam search with the Continuous Relevance Model (CRM) to retrieve a near-optimal set of correlated keywords for the purposes of image annotation refinement. The novel model that was developed as part of this dissertation was christened the “Beam Search CRM” or BS-CRM model. In this Chapter we summarise the main findings of the research and provide some pointers to possible future research questions in the field.

### 5.2 Summary of dissertation achievements

The original Continuous Relevance Model (CRM) and the augmented model with beam search and keyword correlation (the BS-CRM model) were both successfully implemented. A significant amount of thought went into the design of the model so as to ensure that they both ran in reasonable time and memory. Through expressing the central CRM equation as a sequence of matrix operations the amortized runtime to annotate all 500 COREL images was found to be 0.45 seconds compared to 660 seconds as reported in the literature. The memory requirements of the algorithm were of a sufficiently modest level so as to allow the nearly 10,000 image PASCAL dataset to be processed in memory. Furthermore through a comprehensive evaluation we have uncovered several interesting insights into the situations in which the beam search keyword correlation algorithm is most effective on the standard research image datasets.

The results suggest that, in some cases, the BS-CRM model can be an effective means of increasing annotation accuracy as measured by mean per word recall and precision. The best results obtained demonstrated that, over the original CRM model as published by Lavrenko et al. [34], the BS-CRM model achieves a 6.8% increase in mean per word recall and a 31.0% increase in mean per word precision with an increase of 6.5% in the number of words with re-

call greater than zero. Clearly keyword correlation has the ability to eliminate noisy keywords from the image annotation and thereby increase accuracy. Furthermore over varying annotation lengths of 3, 4 and 5 keywords we found that the proposed BS-CRM model performed consistently better than the original CRM model in all three annotation performance metrics, with the most gains being produced at an annotation length of 4 keywords.

We also found that the performance of the BS-CRM model greatly depends on the beam width selected, with widths between 1-15 performing the best on the COREL dataset, with performance declining for wider beam widths. This suggested that there was no advantage in expending additional computation effort search over wider beams than around 15. The hypothesized reason for this observation was thought to be the low probability words that are considered for higher and higher beam widths which might serve to “attract” more noisy keywords to the set through keyword correlation.

In addition, another notable result suggests that the performance of the BS-CRM is also highly dependent on the word smoothing function that is chosen with the most significant gains being realised for the N-CRM model, lower gains with the Multinomial and Dirichlet models and no significant gains at all with the Bernoulli word smoothing model. In this respect it would appear that, through the nature and degree of smoothing provided, the N-CRM model is better able to allow the correlation measure to capture the relationships between the keywords.

The CRM and BS-CRM models were also evaluated on the PASCAL dataset which has been cited as being more challenging compared to the COREL dataset. Indeed the modest increase in F1 measure of 2.4% realised by the BS-CRM is suggestive of the difficulty of this particular dataset. We also have to take into account the relatively simple set of features selected to model the salient regions in the PASCAL images which, along with the small vocabulary size, may have contributed the most to the lower performance of the BS-CRM model on this dataset.

### 5.3 Limitations

There are two areas that in hindsight could be improved upon in this dissertation should more time have been made available both of which relate to the pre-processing of images. Firstly and foremostly, the image feature representation chosen for the PASCAL dataset (mixture of simple colour, texture and position descriptors) was certainly not the most optimal feature representation and may well have lead to the modest performance that was experienced on this dataset. It was certainly very challenging to trade-off the computational and memory limitations of a 4GB, 32-bit machine running MATLAB with the depth of representation given by the extracted features.

Whilst it was an achievement in itself to actually successfully test the CRM model on this



dataset and obtain literature quality results it is nevertheless the author's opinion that to fully ascertain the benefits of the new BS-CRM algorithm a better feature representation would need to be examined. This would result in more high quality words being retrieved initially into the word sets through the strong image features, which the BS-CRM model could subsequently leverage to attract correlated keywords to the set.

In addition some questions have to be raised as to the validity of the PASCAL dataset in itself as a good test bed for the keyword correlation mechanism given the small vocabulary size of 20 keywords and the fact that usually only 1-2 salient objects appear in each image (there is an average of only 1.7 keywords per image in the training dataset). Given this it is likely to be very difficult to make full use of keyword correlation to prune noisy keywords. In hindsight, for this dissertation, it might have been better to examine standard datasets such as those advocated by Makadia et al [37]. Here the authors use the IAPR TC-12 and the ESP Game datasets both of which are of a more challenging nature and of a larger size than the COREL dataset with the further advantage of having an average of 4.5 words per image.

## 5.4 Future Work

There are a significant number of avenues for future work both within the narrow domain of image tagging considered in this dissertation (keyword correlation) and in the field as a whole. In terms of capturing keyword correlation, the following would be interesting avenues for further research:

1. **Different annotation lengths:** We were able to investigate only a small selection of different keyword lengths on the COREL dataset of 3, 4 and 5 words for only one particular word smoothing function (the Normalized CRM model). This experiment could be extended further both to consider datasets that have longer annotations for each image (above 5) and to determine the effect that the word smoothing function used with the BS-CRM has on the annotation refinement capability.

Extending this issue further, given that we observed a peak in performance of the BS-CRM algorithm at a particular annotation length (4 in this case), one could investigate how automatic annotation length determination techniques like those suggested by Jin et al. [25] could be used, if at all, to further improve on performance with the BS-CRM model.

2. **Investigation of much larger keyword vocabularies:** In this dissertation we have followed the bulk of the work in the literature and limited ourselves to relatively small vocabulary sizes (260 words for the COREL dataset). It would certainly be very interesting to ascertain how the BS-CRM algorithm is able to leverage a much larger vocabulary

size to eliminate noisy annotation keywords.

3. **Utilization of outside sources of keyword knowledge:** In a similar spirit, another additional avenue for future research would be to investigate how the World Wide Web and the large WorldNet lexical database could be mined in combination with the image dataset vocabulary itself in order to gain further information on word correlation that may be used by the BS-CRM model to increase keyword refinement accuracy.
4. **Active learning:** In this dissertation we experienced significant difficulty (due to memory limitations) in processing large datasets such as the PASCAL VOC 2007 dataset with nearly 10,000 images. To overcome the memory limitations with large datasets one possibility would be to apply an active learning technique to significantly reduce the number of training examples by selectively sampling annotated images. The basic premise of active learning is to selectively sample data samples so that the uncertainty in determining the right model is reduced by the largest possible margin. Such an approach has been applied to image annotation in [25] with notable success.

With regards to the automatic tagging field as a whole, after conducting a review of the literature in Chapter 2, it is the author's own personal opinion that future research avenues in the field can be usefully summarised along the following four distinct dimensions:

1. **Precision/Recall Accuracy:** There already seems to be a flurry of effort in the field within this area. Many of the papers one encounters are interested in improving the precision/recall accuracy of the annotation algorithms on fixed datasets as their major concern. This is both a good and a bad objective, given that in maximizing the precision/recall accuracy other important but perhaps less attractive areas such as feature representation and extraction and unsupervised learning will suffer as a consequence.

It will certainly only be a matter of time before aspects such as feature extraction and representation, and unsupervised learning present themselves on the critical path barring further significant progress in the field. This author would like to encourage more effort in these related areas, but with also a continuing, but perhaps less intense, focus on introducing and refining algorithms which improve the annotation accuracy.

2. **Feature Extraction & Representation:** This area seems to be neglected to an extent at the moment, which simply could be a reflection of the fact that the field is still in its infancy and other areas such as model development are being given precedence before attention then turns to feature engineering. Firstly a review is badly needed into the best set of feature representations for annotation, given that many authors seem to be simply using feature sets from existing authors (e.g. the COREL dataset of Duygulu et al.).

Furthermore, the dependence on segmentation and region based interest operators needs to be reduced or eliminated given the problems these algorithms have in finding coherent objects in images and returning a suitably representative number of regions from images. A method of extending the popular “bag of features” approach with spatial information is also another useful extension which may have the possibility of extending the robustness of the annotation algorithms.

3. **Unsupervised Learning:** There is a dependence on high quality and relatively small manually labelled image libraries in the literature. This is an unhealthy fixation for a number of reasons, the most prominent being the fact that the algorithms are not entirely being tested to the maximum possible extent in environments that would be common outside of a purely research domain, for example in the commercial or private domains. This limits applicability of the algorithms to real world applications.

In the short term, work therefore needs to be conducted in extending the standard set of testing libraries to include images from diverse sources that have images varying in quality, resolution and colour depth. This would then entice new research to tackle these variations by producing more robust algorithms.

In the medium to long term, researchers would do well to take inspiration from the work of Fergus et al. [14] on unsupervised learning. Here the authors effectively bootstrap their model using images downloaded from Google image search. Advancements in this area would reduce the reliance on high quality manually labelled image datasets completely, thereby further opening up the transition of the annotation technology to commercial mainstream applications.

4. **Scalability & Performance:** Finally if the annotation systems in the literature are ever to reach the mainstream it is crucial that future research is concentrated on ensuring that mechanisms are in place to improve the scalability of the algorithms, with regards to the number of images they can handle in terms of training and indeed in the time it takes to learn new annotation concepts.

Depending on the time taken to surmount the aforementioned challenges, the development of a robust and reliable commercially available automated image tagging system may only take only a decade or perhaps may well follow that of Roman character recognition which was a problem that took 50 years to solve but now manages to achieve a remarkable accuracy approaching 99.5%. Nevertheless, no matter how long it may take for researchers to solve these remaining challenges in the field, we can be sure that when they are finally surmounted the advent of robust automated image annotation systems coupled with efficient CBIR solutions will certainly revolutionize our daily lives by placing a truly huge collection of images at our fingertips.

# Appendix A

## A.1 Example Source Code Listing

This Appendix contains a sample of the main functions used for the CRM and BS-CRM models. The following source code is listed which together cover the majority of the main components of the image tagging system:

- Image annotation algorithm
- Beam search algorithm
- Non-parametric kernel density estimation
- Image feature extraction
- Cross validation framework

### A.1.1 Image annotation algorithm

```
1  % CRM_ANNOTATE  Annotates a set of images with keywords from the
2  % vocabulary
3  %
4  %   Authors: Sean Moran
5  function [annotations,crm_prob] = crm_annotate(train_image_ind, test_image_ind,
        test_word_ind)
6
7  global MAT_PATH;
8  global DATASET;
9  global WORD_SMOOTHING_FUNCTION;
10 global BETA;
11 global MU;
12 global LAMBDA;
13 global WORD_SMOOTHING_FUNCTION_NAME;
```

```

14 global DO_BEAM_SEARCH;
15
16 disp('CRM INFO: Beginning Image Annotation...')
17 disp(' ')
18
19 matfile = [MAT_PATH,'\','DATASET','_similarity_matrices.mat'];
20
21 if (exist(matfile, 'file') > 0)
22
23     BETA_CUR = BETA;
24     MU_CUR=MU;
25     LAMBDA_CUR=LAMBDA;
26     WORD_SMOOTHING_FUNCTION_NAME_CUR=WORD_SMOOTHING_FUNCTION_NAME;
27
28     load(matfile, 'log_kde_prob', 'log_word_prob', 'log_kde_norm_prob', 'MU', '
    LAMBDA','BETA', 'WORD_SMOOTHING_FUNCTION_NAME');
29
30     % Check if parameters have changed since last execution - if so
    we need to re-compute...
31     if (BETA~=BETA_CUR)
32
33         disp(' ')
34         disp('CRM INFO: BETA parameter has changed...re-calculating KDE
    Similarity Matrix...')
35         disp(' ')
36
37         BETA=BETA_CUR;
38
39         log_kde_prob = compute_kdesim_matrix(train_image_ind, test_image_ind);
40         log_kde_norm_prob = repmat(logsumexp(log_kde_prob,2),[1 size(
    train_image_ind,1)]);
41
42         save(matfile,'log_kde_prob', 'log_word_prob', 'log_kde_norm_prob', 'MU',
    'LAMBDA','BETA','WORD_SMOOTHING_FUNCTION_NAME');
43     end
44
45     % We re-compute the word smoothing matrix if the paramters MU,
    LAMBDA
46     % have changed, or if the vocab size has changed or if we change
    the
47     % smoothing function.

```

```

48     if ((MU_CUR~=MU) || (LAMBDA_CUR~=LAMBDA) || (    size(test_word_ind,1)~=    size(
log_word_prob,2)) || (    strcmp(WORD_SMOOTHING_FUNCTION_NAME,
WORD_SMOOTHING_FUNCTION_NAME_CUR)~=1))
49
50     disp(' ')
51     disp('CRM INFO: Word smoothing parameter/vocab size has changed...re-
calculating Word similarity matrix...')
52     disp(' ')
53
54     WORD_SMOOTHING_FUNCTION_NAME=WORD_SMOOTHING_FUNCTION_NAME_CUR;
55     WORD_SMOOTHING_FUNCTION=str2func(WORD_SMOOTHING_FUNCTION_NAME);
56
57     LAMBDA=LAMBDA_CUR;
58     MU=MU_CUR;
59
60     log_word_prob = WORD_SMOOTHING_FUNCTION(test_word_ind, train_image_ind);
61
62     save(matfile,'log_kde_prob', 'log_word_prob', 'log_kde_norm_prob', 'MU',
'LAMBDA','BETA','WORD_SMOOTHING_FUNCTION_NAME');
63 end
64
65 else
66     disp(' ')
67     disp('CRM INFO: Calculating Word and KDE similarity matrices...Please wait
....')
68     disp(' ')
69
70     log_kde_prob = compute_kdesim_matrix(train_image_ind, test_image_ind);
71     log_kde_norm_prob = repmat(logsumexp(log_kde_prob,2),[1    size(train_image_ind
,1)]);
72     log_word_prob = WORD_SMOOTHING_FUNCTION(test_word_ind, train_image_ind);
73
74     save(matfile,'log_kde_prob', 'log_word_prob', 'log_kde_norm_prob', 'MU', '
LAMBDA','BETA','WORD_SMOOTHING_FUNCTION_NAME');
75 end
76
77 word_prob =    exp(log_word_prob);
78 image_posterior_prob =    exp(log_kde_prob-log_kde_norm_prob);
79 crm_prob = image_posterior_prob*word_prob;
80
81 beam_word_sets=[]; beam_probs=[];
82 if (DO_BEAM_SEARCH)

```

```

83     % Refine the annotation results via beam search
84     [beam_word_sets,beam_probs] = crm_beam_search(crm_prob, image_posterior_prob
      , word_prob, test_word_ind);
85 end
86
87 annotations = store_result_in_struct(crm_prob, test_word_ind, beam_word_sets,
      beam_probs);
88
89 disp(' ')
90 disp('CRM INFO: Image Annotation complete...')
91 disp(' ')

```

### A.1.2 Beam search algorithm

```

1  % CRM_BEAM_SEARCH Effficently searches for a set of tags maximizing
      the CRM
2  % keyword correlation objective function.
3  %
4  %   Authors:: Sean Moran
5  function [top_word_sets,word_probs] = crm_beam_search(crm_prob,
      image_posterior_prob, word_prob, test_word_ind)
6
7  global BEAM_WIDTH;
8  global TOP_ANNOTATION_WORDS_NO;
9
10 disp('CRM INFO: Refining Annotations using Beam Search...');
11
12 crm_prob_orig=crm_prob;      % Take a copy of the original CRM results to
      re-use for each beam
13
14 word_sets_total=[];
15 real_word_sets_total=[];
16 set_probs_total=[];
17 no_test_images= size(image_posterior_prob,1);
18
19 for i=1:BEAM_WIDTH
20
21     disp(['CRM INFO: On Beam number...',    int2str(i)]);
22
23     count = 1;
24     crm_prob=crm_prob_orig;
25

```

```

26     set_probs=[];
27     top_probs_prev=[];
28     word_sets=[];
29     real_word_sets=[];
30
31     while count <= TOP_ANNOTATION_WORDS_NO
32
33         [top_probs, top_words] = sort (cm_prob,2,'descend');
34
35         if (count == 1)
36             top_probs = top_probs(:,1);
37             top_words = top_words(:,1);
38         elseif (count == 2)
39             top_probs = top_probs(:,i);
40             top_words = top_words(:,i);
41         else
42             top_probs = top_probs(:,1);
43             top_words = top_words(:,1);
44         end
45
46         word_sets = [word_sets,top_words];
47
48         % The word indices in the following set correspond to the
49         actual
50         % word indices in the original vocabulary
51         real_word_sets = [real_word_sets,test_word_ind(top_words)];
52
53         set_probs = top_probs;
54
55         %top_probs_rep=repmat (top_probs,[1 size(image_posterior_prob
56         ,2) ] );
57
58         top_probs_rep=word_prob(:,top_words)';
59
60         if (~isempty(top_probs_prev))
61             top_probs_rep=top_probs_rep.*top_probs_prev;
62         end
63
64         top_probs_prev = top_probs_rep;
65
66         % Re-calculate the annotation probabilities
67         cm_prob = (top_probs_rep.*image_posterior_prob)*word_prob;

```



```

66
67     % Set the top words probability to zero so we don't take out
    the same word again
68     indy = reshape(word_sets', [ size(word_sets,1)* size(word_sets,2),1]);
69     indx = reshape(repmat((1:1:no_test_images)',[1, size(word_sets,2)]),[
    size(word_sets,1)* size(word_sets,2),1]);
70     ind = sub2ind( size(crm_prob), indx, indy);
71     crm_prob(ind)=0;
72
73     count = count + 1;
74 end
75
76 word_sets = reshape(word_sets',[TOP_ANNOTATION_WORDS_NO,1, no_test_images])
    ;
77 real_word_sets= reshape(real_word_sets',[TOP_ANNOTATION_WORDS_NO,1,
    no_test_images]);
78
79 word_sets_total = [word_sets_total,word_sets];
80 real_word_sets_total = [real_word_sets_total,real_word_sets];
81
82 set_probs_total=[set_probs_total,set_probs];
83
84 end
85
86 % Pull out the top wordsets for each image across all beams
87 [sorted_probs, sorted_prob_ind] = sort(set_probs_total,2,'descend');
88 top_set_probs=sorted_probs(:,1);
89
90 real_word_sets_total= reshape(real_word_sets_total(:),[TOP_ANNOTATION_WORDS_NO
    BEAM_WIDTH*no_test_images]);
91 word_sets_total= reshape(word_sets_total(:),[TOP_ANNOTATION_WORDS_NO BEAM_WIDTH*
    no_test_images]);
92
93 top_beams = sorted_prob_ind(:,1);
94 beam_offset_ind = (0:BEAM_WIDTH:(BEAM_WIDTH*no_test_images-1))+top_beams';
95 top_word_sets=real_word_sets_total(:,beam_offset_ind);
96
97 % Now we want to pull out P(w/I) for each word selected for the
    purposes of TREC eval
98 word_sets_total=word_sets_total(:,beam_offset_ind);
99 word_sets_total = [word_sets_total(:), reshape(repmat((1:1:no_test_images)',[1
    TOP_ANNOTATION_WORDS_NO]))',[no_test_images*TOP_ANNOTATION_WORDS_NO,1]]];

```

```

100
101 % Get the words in the sets in the correct order
102 ind=sub2ind( size(crm_prob_orig),word_sets_total(:,2),word_sets_total(:,1));
103 [word_probs,pos] = sort(reshape(crm_prob_orig(ind),[TOP_ANNOTATION_WORDS_NO,
    no_test_images]),1,'descend');
104 pos = [pos(:), reshape(repmat((1:1:no_test_images)',[1 TOP_ANNOTATION_WORDS_NO])
    ',[no_test_images*TOP_ANNOTATION_WORDS_NO,1])];
105 ind=sub2ind( size(top_word_sets),pos(:,1),pos(:,2));
106 top_word_sets= reshape(top_word_sets(ind),[TOP_ANNOTATION_WORDS_NO
    no_test_images]);

```

### A.1.3 Non-parametric kernel density estimation

```

1 % COMPUTE_GAUSSKDE_LOGPROB Computes the similarity between images I
  and J
2 %
3 % Authors: Sean Moran
4 function log_prob = compute_gausskde_logprob(test_image_ind, train_image_ind)
5
6 global BETA;
7 global DIM;
8 global LOG_INV_TRAIN_FEATURE_SET_SIZES;
9 global TEST_DOC_BLOBS_MAT;
10 global DOC_BLOBS_MAT;
11
12 kde_coeff = -1*(( log(2*pi*BETA))*(DIM/2));
13
14 test_blobs= TEST_DOC_BLOBS_MAT(:, :, test_image_ind);
15 features = DOC_BLOBS_MAT(:, :, train_image_ind);
16
17 train_feature_dims = size(features);
18 test_feature_dims = size(permute(test_blobs,[2 1 3]));
19
20 if (size(test_feature_dims,2)==2) % In the case where the block size is
    1 test image
21     test_feature_dims(3)=1;
22 end
23
24 dist = reshape(sqrt(sqdist(test_blobs(:, :), features(:, :))),[test_feature_dims
    (1) test_feature_dims(3) train_feature_dims(2: end)]);
25 dist = permute(dist,[3 1 4 2]);
26

```

```

27 kde_exp_val = (-1*(dist./BETA));
28
29 log_probs = kde_coeff+kde_exp_val;
30
31 log_probs = logsumexp(log_probs);
32
33 log_inv_train_feature_set_sizes= repmat (LOG_INV_TRAIN_FEATURE_SET_SIZES,[1 1 1
    size(log_probs,4)]);
34
35 log_prob = nansum(log_probs+log_inv_train_feature_set_sizes(:,:,train_image_ind
    :,2),2);
36
37 log_prob = squeeze(permute(log_prob, [1 3 2 4]))';

```

#### A.1.4 Image feature extraction

```

1  % COMPUTE_GAUSSKDE_LOGPROB  Extracts image features using a mixture
    of
2  % simple colour, position and texture descriptors.
3  %
4  %   Authors: Sean Moran
5  function [] = compute_image_features(image_dir, results_dir, annotation_path,
    vocab_path, image_type)
6
7  image_files= dir([ image_dir '/*.jpg']);
8  image_file = { image_files.name };
9
10 no_images= length(image_file);
11
12 blobs_fname = [results_dir,'\ ',image_type,'_blobs.txt'];
13 doc_blobs_fname = [results_dir,'\ ',image_type,'_document_blobs.txt'];
14 words_fname = [results_dir,'\ ',image_type,'_document_words.txt'];
15
16 indices_fname = [results_dir,'\ ',image_type,'_image_indices.txt'];
17
18 fp = fopen(indices_fname, 'wt');
19
20 fid = fopen(vocab_path,'r');
21 vocab = textscan(fid,'%s','delimiter','\n');
22 fclose(fid);
23 vocab = vocab{:};
24

```

```

25 blob_id = 1;
26 feature_count =1;
27 M = 85;           % spacing of points in the grid
28
29 for i = 1:no_images
30
31     disp(['Pre-processing image: ',image_file{i}]);
32     image = imread([ image_dir '/' image_file{i} ]);
33
34     image = im2double(imresize( image, [375 500]));
35
36     xyz_im=vl_rgb2xyz( image);
37     hsv_im= rgb2hsv(image);
38     lab_im=vl_xyz2lab(xyz_im);
39
40     [ nx ny dummy ] = size(image);
41
42     x = M:M:(nx);
43     y = M:M:(ny);
44
45     no_features = size(x,2)* size(y,2);
46     feature_vec=[];
47     for xx = 1: length(x)
48         for yy = 1: length(y);
49             feature_vec = [feature_vec;extract_image_features( image, lab_im,
50                 hsv_im, (x(xx)-M+1):x(xx), (y(yy)-M+1):y(yy), (x(xx)/2)/nx, (y(yy)/2)/ny)];
51         end
52     end
53     blob_id_vec = [feature_count:1:(feature_count+no_features)-1]);
54
55     rec=PASreadrecord([annotation_path,'\',image_file{i}(1: end-4),'.xml']);
56
57     % Find the indices of the words from the annotations
58     words=[];
59     for j=1: size(rec.objects,2)
60         word = rec.objects(j).class;
61         for k=1: size(vocab,1)
62             if (~isempty(strfind(word,vocab{k})))
63                 break;
64             end
65         end

```

```

66         if isempty(intersect([k],words))
67             words=[words,k];
68         end
69     end
70
71     no_words = size(words,2);
72     words = [words, repmat(-99,[1 10-no_words])];           % Again we pad the
                                                                words to a length of 10
73
74     dlmwrite(words_fname,words,'delimiter','\t','-append');
75     dlmwrite(blobs_fname,feature_vec,'delimiter','\t','-append');
76     dlmwrite(doc_blobs_fname,blob_id_vec,'delimiter','\t','-append');
77
78     fprintf(fp, '%d\t%s\n', blob_id, image_file{i}(1:      end-4));
79
80     disp([int2str(blob_id),'/', int2str(no_images),' images processed...']);
81
82     blob_id = blob_id +1;
83     feature_count = feature_count + no_features;
84 end
85
86 fclose(fp);

```

### A.1.5 Cross validation framework

```

1  % OPTIMIZE_MU_AND_BETA Cross validation framework for the mu and
   beta
2  % parameters of the CRM model
3  %
4  % Authors: Sean Moran
5  function [] = optimize_mu_and_beta(train_set_ind,test_set_ind,root_dir_name)
6
7  global BETA;
8  global MU;
9  global VOCAB_TEST_SET;
10 global TREC_ANN_CV_REL_FILE;
11 global TREC_RET_CV_REL_FILE;
12 global ANN_CV_QREL_DATA;
13
14 % We need to create the trec .qrel files for this particular split
   of data

```

```

15 ANN_CV_QREL_DATA = create_ann_trec_qrel_file(VOCAB_TEST_SET, test_set_ind,
      TREC_ANN_CV_REL_FILE);
16 create_ret_trec_qrel_file(VOCAB_TEST_SET, test_set_ind, TREC_RET_CV_REL_FILE);
17
18 disp(' ')
19 disp('CRM INFO: Finding optimum MU and BETA parameters for the Model ...');
20
21 beta_values=[0.01,0.03,0.1,0.3,1,3,10,30];
22 mu_values=[5,10,15,20,30];
23
24 max_ann_fm = - Inf;
25 max_ret_map = - Inf;
26 max_ann_mu = 0;
27 max_ann_beta = 0;
28 max_ret_mu = 0;
29 max_ret_beta = 0;
30
31 beta_count = 1;
32
33 ann_results_fname = [root_dir_name,'\cv_ann_results.txt'];
34 fp1 = fopen(ann_results_fname, 'wt');
35 ret_results_fname = [root_dir_name,'\cv_ret_results.txt'];
36 fp2 = fopen(ret_results_fname, 'wt');
37
38 while (beta_count <= size(beta_values,2))
39
40     BETA=beta_values(:,beta_count);
41     mu_count= 1;
42
43     while (mu_count <= size(mu_values,2))
44
45         MU= mu_values(:,mu_count);
46
47         disp(' ')
48         disp(['CRM INFO: MU =', sprintf('%.3f',MU),' BETA =', sprintf('%.3f',
49 BETA)]);
50
51         [ann_fm, ret_map] = compute_cv_stats(train_set_ind, test_set_ind,
52 root_dir_name);
53
54         if (ann_fm > max_ann_fm)
55             max_ann_fm = ann_fm;

```

```

54         max_ann_mu = MU;
55         max_ann_beta = BETA;
56     end
57
58     if (ret_map > max_ret_map)
59         max_ret_map = ret_map;
60         max_ret_mu = MU;
61         max_ret_beta = BETA;
62     end
63
64     fprintf(fp1, '%.5f\t%.5f\t%.5f\n', MU, BETA, ann_fm);
65     fprintf(fp2, '%.5f\t%.5f\t%.5f\n', MU, BETA, ret_map);
66
67     disp(['CRM INFO: Current Annotation MAP ... ', sprintf('%.5f', ann_fm)
68 ]);
69     disp(['CRM INFO: Best Annotation MAP ... ', sprintf('%.5f', max_ann_fm
70 )]);
71
72     mu_count = mu_count + 1;
73
74     end
75
76     beta_count = beta_count + 1;
77 end
78
79 fclose(fp1);
80 fclose(fp2);
81
82 disp(' ');
83
84 best_results_fname = [root_dir_name, '\cv_best_results.txt'];
85 fp3 = fopen(best_results_fname, 'wt');
86
87 fprintf(fp3, '%s\t%.5f\t%.5f\n', 'Best ANN MAP', max_ann_fm);
88 fprintf(fp3, '%s\t%.5f\t%.5f\n', 'Best ANN MU', max_ann_mu);
89 fprintf(fp3, '%s\t%.5f\t%.5f\n', 'Best ANN BETA', max_ann_beta);
90 fprintf(fp3, '%s\t%.5f\t%.5f\n', 'Best RET MAP', max_ret_map);
91 fprintf(fp3, '%s\t%.5f\t%.5f\n', 'Best RET MU', max_ret_mu);
92 fprintf(fp3, '%s\t%.5f\t%.5f\n', 'Best RET BETA', max_ret_beta);
93

```

```
94 fclose(fp3);
95
96 disp('CRM INFO: Annotation Results ...');
97 disp(['CRM INFO: Best MAP: ', sprintf('%0.5f',max_ann_fm)]);
98 disp(['CRM INFO: Best MU: ', sprintf('%0.5f',max_ann_mu)]);
99 disp(['CRM INFO: Best BETA: ', sprintf('%0.5f',max_ann_beta)]);
100 disp(' ')
101 disp('CRM INFO: Retrieval Results ...');
102 disp(['CRM INFO: Best MAP: ', sprintf('%0.5f',max_ret_map)]);
103 disp(['CRM INFO: Best MU: ', sprintf('%0.5f',max_ret_mu)]);
104 disp(['CRM INFO: Best BETA: ', sprintf('%0.5f',max_ret_beta)]);
```



# Bibliography

- [1] M. Ames and M. Naaman. Why we tag: motivations for annotation in mobile and on-line media. In *CHI '07: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 971–980, New York, NY, USA, 2007. ACM Press.
- [2] Z. W. Li and B. Wang. Image annotation in a progressive way. In *In proc. ICME*, pages 811–814, 2007.
- [3] K. Barnard, P. Duygulu, N. de Freitas, D. Forsyth, D.M. Blei, and M.I. Jordan. Matching words and pictures. *Journal of Machine Learning Research*, 3:1107–1135, 2003.
- [4] R. Bisiani. *Encyclopedia of Artificial Intelligence*. Wiley Sons, 1987.
- [5] D.M. Blei and M.I. Jordan. Modeling annotated data. In *SIGIR '03: Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*, pages 127–134, New York, NY, USA, 2003. ACM Press.
- [6] P. Brown, S. A. Della Pietra, V. J. Della Pietra, and R. L. Mercer. The mathematics of statistical machine translation: parameter estimation. *Computational Linguistics*, 19:263–311, 1993.
- [7] G.J. Burghouts and J.M. Geusebroek. Performance evaluation of local colour invariants. *Comput. Vis. Image Underst.*, 113(1):48–62, 2009.
- [8] C. Carson, M. Thomas, S. Belongie, J. Hellerstein, and J. Malik. Blobworld: a system for region-based image indexing and retrieval. Technical report, Berkeley, CA, USA, 1999.
- [9] O. Chapelle, P. Haffner, and V. N. Vapnik. Support vector machines for histogram-based image classification. *Neural Networks, IEEE Transactions on*, 10(5):1055–1064, 1999.
- [10] P. Duygulu, K. Barnard, N. de Freitas, and D. Forsyth. Object recognition as machine translation: Learning a lexicon for a fixed image vocabulary. In *ECCV '02: Proceedings of the 7th European Conference on Computer Vision-Part IV*, pages 97–112, London, UK, 2002. Springer-Verlag.

- [11] G.P. Enser, C.J. Sandom, and P.H. Lewis. *Automatic Annotation of Images from the Practitioner Perspective*, volume 3568. 2005.
- [12] S. Feng and R. Manmatha. A discrete direct retrieval model for image and video retrieval. In *CIVR '08: Proceedings of the 2008 international conference on Content-based image and video retrieval*, pages 427–436, New York, NY, USA, 2008. ACM.
- [13] S.L. Feng, R. Manmatha, and V. Lavrenko. Multiple bernoulli relevance models for image and video annotation. *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*, 2:II–1002–II–1009 Vol.2, June-2 July 2004.
- [14] R. Fergus, L. Fei-Fei, P. Perona, and A. Zisserman. Learning object categories from google’s image search. *Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on*, 2:1816–1823 Vol. 2, Oct. 2005.
- [15] D.A. Forsyth and J. Ponce. *Computer Vision: A Modern Approach*. Prentice Hall, August 2002.
- [16] D. Furcy and S. Kleniv. Limited discrepancy beam search. In *Proceedings of the International Joint Conference on Artificial Intelligence, IJCAI*, 2005.
- [17] J. Gantz. The diverse and exploding digital universe: An updated forecast of worldwide information. *IDC. White Paper*, 2008.
- [18] A. Ghoshal. Hidden markov models for automatic annotation and content-based retrieval of images and video. In *Proceedings of the ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 544–551, 2005.
- [19] J. S. Hare, P. A. S. Sinclair, P. H. Lewis, K. Martinez, P. G. B. Enser, and C. J. Sandom. Bridging the semantic gap in multimedia information retrieval: Top-down and bottom-up approaches. In *3rd European Semantic Web Conference (ESWC-06)*, 2006.
- [20] J. Huang, S. Ravi Kumar, and R. Zabih. An automatic hierarchical image classification scheme. In *ACM Multimedia*, pages 219–228, 1998.
- [21] G. Iyengar, P. Duygulu, S. Feng, P. Ircing, S. Khudanpur, D. Klakow, M. Krause, R. Manmatha, H. Nock, D. Petkova, B. Pytlik, and P. Virga. Joint visual-text modeling for automatic retrieval of multimedia documents. In *MULTIMEDIA '05: Proceedings of the 13th annual ACM international conference on Multimedia*, pages 21–30, New York, NY, USA, 2005. ACM Press.
- [22] A.K. Jain, S. Prabhakar, and L. Hong. A multichannel approach to fingerprint classification. *IEEE Trans. Pattern Anal. Mach. Intell.*, 21(4):348–359, 1999.

- [23] J. Jeon, V. Lavrenko, and R. Manmatha. Automatic image annotation and retrieval using cross-media relevance models. In *SIGIR '03: Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*, pages 119–126, New York, NY, USA, 2003. ACM Press.
- [24] J. Jeon and R. Manmatha. Using maximum entropy for automatic image annotation. In *In Proc. CIVR*, pages 24–32, 2004.
- [25] R. Jin, J.Y. Chai, and L. Si. Effective automatic image annotation via a coherent language model and active learning. In *MULTIMEDIA '04: Proceedings of the 12th annual ACM international conference on Multimedia*, pages 892–899, New York, NY, USA, 2004. ACM.
- [26] Y. Jin, L. Khan, L. Wang, and M. Awad. Image annotations by combining multiple evidence & wordnet. In *MULTIMEDIA '05: Proceedings of the 13th annual ACM international conference on Multimedia*, pages 706–715, New York, NY, USA, 2005. ACM.
- [27] J.Tang. *Automatic Image Annotation and Object Detection*. PhD thesis, 2008.
- [28] k. Mikolajczyk and C. Schmid. A performance evaluation of local descriptors. *IEEE Trans. Pattern Anal. Mach. Intell.*, 27(10):1615–1630, October 2005.
- [29] T. Kadir and M. Brady. Saliency, scale and image description. *International Journal of Computer Vision*, V45(2):83–105, November 2001.
- [30] F. Kang, R. Jin, and R. Sukthankar. Correlated label propagation with application to multi-label learning. In *CVPR '06: Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 1719–1726, New York City, NY, USA, 2006. IEEE Computer Society.
- [31] Y. Ke and R. Sukthankar. Pca-sift: a more distinctive representation for local image descriptors. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, volume 2, pages 506–513, 2004.
- [32] V. Lavrenko. *A Generative Theory of Relevance*. Springer Publishing Company, Incorporated, 2008.
- [33] V. Lavrenko, S.L. Feng, and R. Manmatha. Statistical models for automatic video annotation and retrieval. volume 3 of *Proceedings - IEEE International Conference on Acoustics, Speech, and Signal Processing*, 2004.
- [34] V. Lavrenko, R. Manmatha, and J. Jeon. A model for learning the semantics of pictures. In *in NIPS*. MIT Press, 2003.

- [35] J. Liu, M. Li, W. Ma, Q. Liu, and H. Lu. An adaptive graph model for automatic image annotation. In *MIR '06: Proceedings of the 8th ACM international workshop on Multimedia information retrieval*, pages 61–70, New York, NY, USA, 2006. ACM.
- [36] D.G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60:91–110, 2004.
- [37] A. Makadia, V. Pavlovic, and S. Kumar. A new baseline for image annotation. In *ECCV '08: Proceedings of the 10th European Conference on Computer Vision*, pages 316–329, Berlin, Heidelberg, 2008. Springer-Verlag.
- [38] M. Markkula and E. Sormunen. End-user searching challenges indexing practices in the digital newspaper photo archive. *Information retrieval*, 1:259–285, 2000.
- [39] G. A. Miller, R. Beckwith, C. Fellbaum, D. Gross, and K.J. Miller. Introduction to wordnet: an on-line lexical database\*. *Int J Lexicography*, 3(4):235–244, January 1990.
- [40] F. Monay and D. Gatica-Perez. Plsa-based image auto-annotation: constraining the latent space. In *MULTIMEDIA '04: Proceedings of the 12th annual ACM international conference on Multimedia*, pages 348–351, New York, NY, USA, 2004. ACM Press.
- [41] Y. Mori, H. Takahashi, and R. Oka. Image-to-word transformation based on dividing and vector quantizing images with words. In *MISRM'99: First International Workshop on Multimedia Intelligent Storage and Retrieval Management*, 1999.
- [42] H. Müller, S. Marchand-Maillet, and T. Pun. The truth about corel - evaluation in image retrieval. In *CIVR '02: Proceedings of the International Conference on Image and Video Retrieval*, pages 38–49, London, UK, 2002. Springer-Verlag.
- [43] M.R. Naphade, I. Kozintsev, T.S. Huang, and K. Ramchandran. A factor graph framework for semantic indexing and retrieval in video. In *CBAIVL '00: Proceedings of the IEEE Workshop on Content-based Access of Image and Video Libraries (CBAIVL'00)*, page 35, Washington, DC, USA, 2000. IEEE Computer Society.
- [44] H. Ney, R. Haeb-Umbach, B.H. Tran, and M. Oerder. Improvements in beam search for 10000-word continuous speech recognition. In *Acoustics, Speech, and Signal Processing, 1992. ICASSP-92., 1992 IEEE International Conference on*, volume 1, pages 9–12 vol.1, 1992.
- [45] S. Ornager. View a picture, theoretical image analysis and empirical user studies on indexing and retrieval. *Swedis Library Research*, 2-3:31–41, 1996.
- [46] J.M. Ponte and B.W. Croft. A language modeling approach to information retrieval. In *Proceedings of SIGIR*, pages 275–281, 1998.

- [47] G. Qi, X. Hua, Y. Rui, J. Tang, T. Mei, and H. Zhang. Correlative multi-label video annotation. In *MULTIMEDIA '07: Proceedings of the 15th international conference on Multimedia*, pages 17–26, New York, NY, USA, 2007. ACM.
- [48] X. Qi and Y. Han. Incorporating multiple svms for automatic image annotation. *Pattern Recognition*, 40(2):728–741, February 2007.
- [49] L.R. Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. In *Proceedings of the IEEE*, pages 257–286, 1989.
- [50] S.M. Rubin. *The argos image understanding system*. PhD thesis, Pittsburgh, PA, USA, 1978.
- [51] J. Shi and J. Malik. Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):888–905, 2000.
- [52] A.W.M. Smeulders, M. Worring, S. Santini, A. Gupta, and R. Jain. Content-based image retrieval at the end of the early years. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 22(12):1349–1380, 2000.
- [53] M. Sonka, V. Hlavac, and R. Boyle. *Image Processing, Analysis, and Machine Vision*. Cengage-Engineering, March 2007.
- [54] M. Srikanth, J. Varner, M. Bowden, and D. Moldovan. Exploiting ontologies for automatic image annotation. In *SIGIR '05: Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 552–558, New York, NY, USA, 2005. ACM.
- [55] C. Tillmann and H. Ney. Word reordering and a dynamic programming beam search algorithm for statistical machine translation. *Comput. Linguist.*, 29(1):97–133, 2003.
- [56] H. Tong, C. Faloutsos, and J.Y. Pan. Fast random walk with restart and its applications. In *ICDM '06: Proceedings of the Sixth International Conference on Data Mining*, pages 613–622, Washington, DC, USA, 2006. IEEE Computer Society.
- [57] J. Valente and R. Alves. Filtered and recovering beam search algorithms for the early/-tardy scheduling problem with no idle time. *Comput. Ind. Eng.*, 48(2):363–375, 2005.
- [58] C. Wang, F. Jing, L. Zhang, and H. Zhang. Content based image annotation refinement. In *Proceedings of the 2007 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 1–8, Minneapolis, Minnesota, 2007. IEEE Computer Society.
- [59] G. Wang, D. Forsyth, and D. Hoiem. Building text features for object image classifications. In *CVPR*, 2009.

- [60] Y. Wang and S. Gong. Refining image annotation using contextual relations between words. In *CIVR '07: Proceedings of the 6th ACM international conference on Image and video retrieval*, pages 425–432, New York, NY, USA, 2007. ACM.
- [61] A. Yavlinsky. *Image indexing and retrieval using automated annotation* Alexei Yavlinsky. PhD thesis, 2007.
- [62] A. Yavlinsky, E. Schofield, and S. Rger. Automated image annotation using global features and robust nonparametric density estimation. In *International Conference on Image and Video Retrieval*, pages 507–517. Springer, 2005.
- [63] R. Zhou and E. Hansen. Beam-stack search: Integrating backtracking with beam search. In *Proceedings of the 15th International Conference on Automated Planning and Scheduling (ICAPS-05)*, pages 90–98, Monterey, CA, 2005.
- [64] X. Zhou, M. Wang, Q. Zhang, J. Zhang, and B. Shi. Automatic image annotation by an iterative approach: incorporating keyword correlations and region matching. In *CIVR '07: Proceedings of the 6th ACM international conference on Image and video retrieval*, pages 25–32, New York, NY, USA, 2007. ACM.