Solving Decomposition Problems in Computer Vision using Linear Optimization

Thesis submitted in partial fulfillment
of the requirements for the degree of

Master of Science (by Research)
in
Computer Science Engineering

by

Ankit Gandhi
200802004
ankit.gandhiug08@students.iiit.ac.in

Center for Visual Information Technology
International Institute of Information Technology
Hyderabad - 500 032, INDIA
July 2014

International Institute of Information Technology
Hyderabad, India

CERTIFICATE

It is certified that the work contained in this thesis, titled "Solving Decomposition Problems in Computer Vision using Linear Optimization" by Ankit Gandhi, has been carried out under my supervision and is not submitted elsewhere for a degree.

Date

Adviser: Prof. C. V. Jawahar

To my Family...

Acknowledgments

First of all, I owe my deepest thanks and gratitude to my adviser, Dr. C. V. Jawahar, for his unparalleled support, motivation, dedication, time and encouragement. From searching a problem to writing a paper, I have learned countless things from him which is almost impossible to enumerate here. Many thanks to Dr. Karteek Alahari also, without him this work would not have been the same.

I am thankful to my lab mates and CVIT peers for having all the fruitful and useful discussions, and for the sleepless nights we were working together before deadlines – Aditya, Chandrashekar, Jay, Anand, Mihir, Harshit, Mayank, Abhinav, Vinay, Siddhartha, Praveen, Aseem, Natraj, Naveen. I would also like to thank all my friends at IIIT - Varun, Sanidhya, Harshit, Mohit, Vipul, Mayank, Nikhil, Kaustav, Archit, Ishan, Aravindh for supporting me through good and bad times, and making life at IIIT a remarkable one.

A special thanks to Dr. P. J. Narayanan, Dr. Anoop Namboodiri, Dr. Jayanthi Sivaswamy for creating a phenomenal environment in CVIT to do research and CVIT administration – Satya, Phani, Rajan for helping me in numerous occasions.

Last but definitely above all, I thank my parents and family for their constant love, support and belief. Inspite of having a business background, my family allowed me to pursue my dreams. They hardly know what research or writing a thesis means but they always back my decisions up. I am forever indebted to my father and uncle (elder).

Abstract

The images can be considered as a union of many parts or a composition of multiple segments. Some of the examples of such parts/segments can be different objects present in the image, foreground and background regions, or the textual (regions containing text) and non-textual regions in the image. In order to get the inherent semantics and higher level knowledge associated with the images, getting insight of such parts/segments is essential. In this thesis, we introduce the notion of "decomposition" in images. The decomposition refers to the phenomenon of break down of images into its constituents meaningful parts/segments. Those meaningful parts depends on the task we are interested in. In case of foreground and background decomposition, it can be a pixel accurate segmentation of a foreground or the tight rectangular box enclosing the foreground segment. In this work, we have discussed the problem of decomposition in two kinds of images – natural images and document images. We also show how popular computer vision tasks such as object detection, semantic segmentation, document layout analysis, word spotting, etc. can be perceived as a decomposition task.

In this thesis, we solve two decomposition problems in a linear optimization framework. Firstly, decomposing a global histogram of a natural image into histograms of its associated objects and regions and secondly, decomposing a questioned document image into regions containing copied and non-copied/original contents in a recognition free setting.

The decomposition of a global histogram representation of an image into histograms of its associated objects and regions is formulated as an optimization problem, given a set of linear classifiers, which can effectively discriminate the object categories present in the image. This decomposition bypasses harder problems associated with localization and the explicit pixel-level segmentation. Our decomposition framework is also applicable for separating histograms of object and background in an image. Our solution is computationally efficient and we demonstrate its utility in multiple situations. We evaluate our method on a wide variety of composite histograms and also compare it with MRF-based solutions. We decompose histograms at an average accuracy of 86.4% on a Caltech-256 based dataset. In addition to merely measuring the accuracy of decomposition, we also show the utility of the estimated object and background histograms for the task of image classification on PASCAL VOC 2007 dataset.

To solve the problem of decomposition in questioned document images, we detect documents from the database which have exact or similar text to a given query document or region image. Exact duplicate is the direct cut and paste of content from multiple documents in the database whereas near duplicate document segments (similar text) could arise due to various document manipulations like summariza-

tion, copying, rewriting, editing, formatting, cut-and-paste, etc. We refer to the corresponding problems as retrieval of exact and near duplicate document images. We formulate the problem as a document retrieval task, and solve it in a recognition-free setting. We propose two approaches which are capable of detecting regions generated by these operations accurately without depending on a reliable OCR. First approach is based on modelling the solution as finding a mixture of homographies, and designing a linear programming (LP) based solution to compute the same while the second approach is based on learning a discriminative classifier for a questioned document region to retrieve duplicate documents. Using both the approaches, we get encouraging results.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The animals and humans use their eye and brain to see, visualize and understand the world they live in. Computer vision and machine learning are the disciplines which intend to emulate similar behaviour for machines. For a machine, an image is just a set of raw pixel values. In order to associate some high-level and semantic information to images, machines need to process them. Processing is task specific and some common computer vision tasks in the past include object detection, image segmentation, image retrieval, image classification and recognition, optical character recognition (OCR), scene text recognition, action recognition, pose estimation, event detection, modelling of objects and environment, image restoration, scene reconstruction, video tracking and many others. Most of these tasks are quite simple for humans and can be done within a blink of an eye as human brain can perform certain computations (e.g., pattern recognition, perception, etc.) faster than the digital computers in existence today because of the complex interconnections of neurons [8] in human brain. Although machines are nowhere close to human performance but in the past we have achieved significant progress. In this chapter, we acquaint with the phenomenon of "decomposition" in case of images and see how it is central to most of the tasks mentioned above.

1.1 Types of Digital Images

The emergence of new technologies and production of cheap equipments has resulted in the enormous increase in the amount of visual digital information. In order to associate some higher level information to them, great strides have been made in the field of digital imaging. Digital image can be of various forms depending upon their use and method of acquisition. Some of them includes medical images, satellite images, 3D images, natural images, and document images. In this thesis, we have worked upon the following two kinds of images -

1. **Natural Images:** Natural images are real life images which are taken from large community photo collections like Flickr, Google, Facebook, etc. Some other examples of natural images include images captured through robotic navigation and visual surveillance. Working on such

images is very challenging as they have been taken in completely unconstrained environment. Some of the examples of natural images are shown in Figure 1.1.

2. **Document Images:** Document images are the digitally scanned or captured images of printed/handwritten documents and books. Document imaging has received significant attention in the past as it makes the books searchable and digitization will also make the renowned libraries accessible worldwide.

1.2 Spectrum of Computer Vision Tasks

In this section, we discuss the wide spectrum of computer vision tasks which have received significant attention in the past both for the natural and document images.

- **Object Recognition and Classification**: Classification is a classical problem in computer vision which involves determining whether or not the image data contains some specific object, pattern or feature. It refers to the task of classifying an image based on its visual content (for example, Is car present in the image ? Yes or No). In most classification tasks, we have given some predetermined candidate categories (e.g., horse, dog, cat, bus, aeroplane, ship, car, bicycle, sheep, train, chair, etc.) and we have to predict the most appropriate category for the unseen image. There exists many types of visual classification tasks which includes scene classification (e.g., city-scape, bookstore, night-time), object recognition (e.g., bottle, chair, diningtable), fine-grained classification (e.g., kitchen, bedroom, office), etc.

- **Object Detection**: Object detection refers to the task of localizing and detecting semantic objects of a certain class (for example, person, cat, dog, etc.) in digital images. Typically, objects are localized by a rectangular tight bounding box which encloses them completely (Figure 1.1). The sliding window approach is the most widely used technique to detect an object in the image.

- **Semantic Segmentation**: Semantic Segmentation aims at pixel-wise classification of images, i.e., it assigns a category label to each pixel in the image. Generally, segmentation algorithms consists of local appearance and consistencies model combined in a unified probabilistic framework. One can consider object localization as a particular instance of semantic segmentation.

- **Optical Character Recognition** (OCR): OCR refers to the task of converting scanned document images of handwritten, typewritten or printed text into machine-encoded text. It is the common tool for automatically digitizing printed texts so that they can be electronically searched, stored compactly, and used in machine processes such as machine translation, text to speech, etc.

- **Document Layout Analysis**: It is the process of identifying and categorizing the regions of interests in the scanned images of a text document. One such task is the segmentation of text zones from non-textual ones in the documents and it is typically used as a pre-processing step before passing a document image to an OCR. The task can be performed at finer level also by dividing text

zones into different logical roles such as titles, captions, footnotes, etc. and non-textual zones into figures, tables, math symbols, etc. Document Layout Analysis [9] and Document Layout Analysis & Reconstruction [10] are the common tools available for performing such decomposition.

- **Word Spotting** Word spotting is the task of detecting query words in handwritten document image collections without involving a traditional OCR procedure. It can be used for the indexing and retrieval of handwritten documents and has attracted the attention of the research community in the field of document image processing and analysis in the past.

1.3 Applications

In this section, we discuss some of the applications of above mentioned computer vision tasks -

- **Content based Image Retrieval (CBIR):** It refers to finding all the images from the large database which have a specific content. Some of the example queries can be – retrieve images which have cars in them, retrieve images containing animals, etc. Image classification and recognition significantly helps in classifying the images and hence, in efficiently retrieving them.

- **Robotic Vision:** Computer Vision is an important part of robots. It enables them to recognize the environment and take decisions accordingly.

- **Labelling and Indexing:** Because of the advent of internet and social networking sites like Facebook, Google+, Flickr, Youtube, Instagram, etc., amount of digital content has increased exponentially. Manually labelling them is almost impossible, so recognition systems can be used to label the images automatically and then, the labels can be used for indexing. Indexing is necessary in order to perform retrieval from such large databases cheaply.

- **Document Reconstruction:** It is an important application of document layout analysis which is used for converting a document image into HTML and PDF files. Text (characters) is recognized using OCR and pages elements (such as images, separators, figures, tables, etc.) are cropped and included in the document. In this way, we can reconstruct a document from its image while preserving its layout.

- **Text Translation:** OCR can be used to first recognize texts in the documents and then, using language processing methods can be used to convert text into different foreign languages. Text recognition have many other applications also apart from these (automatic number plate recognition, text to speech translation, etc.).

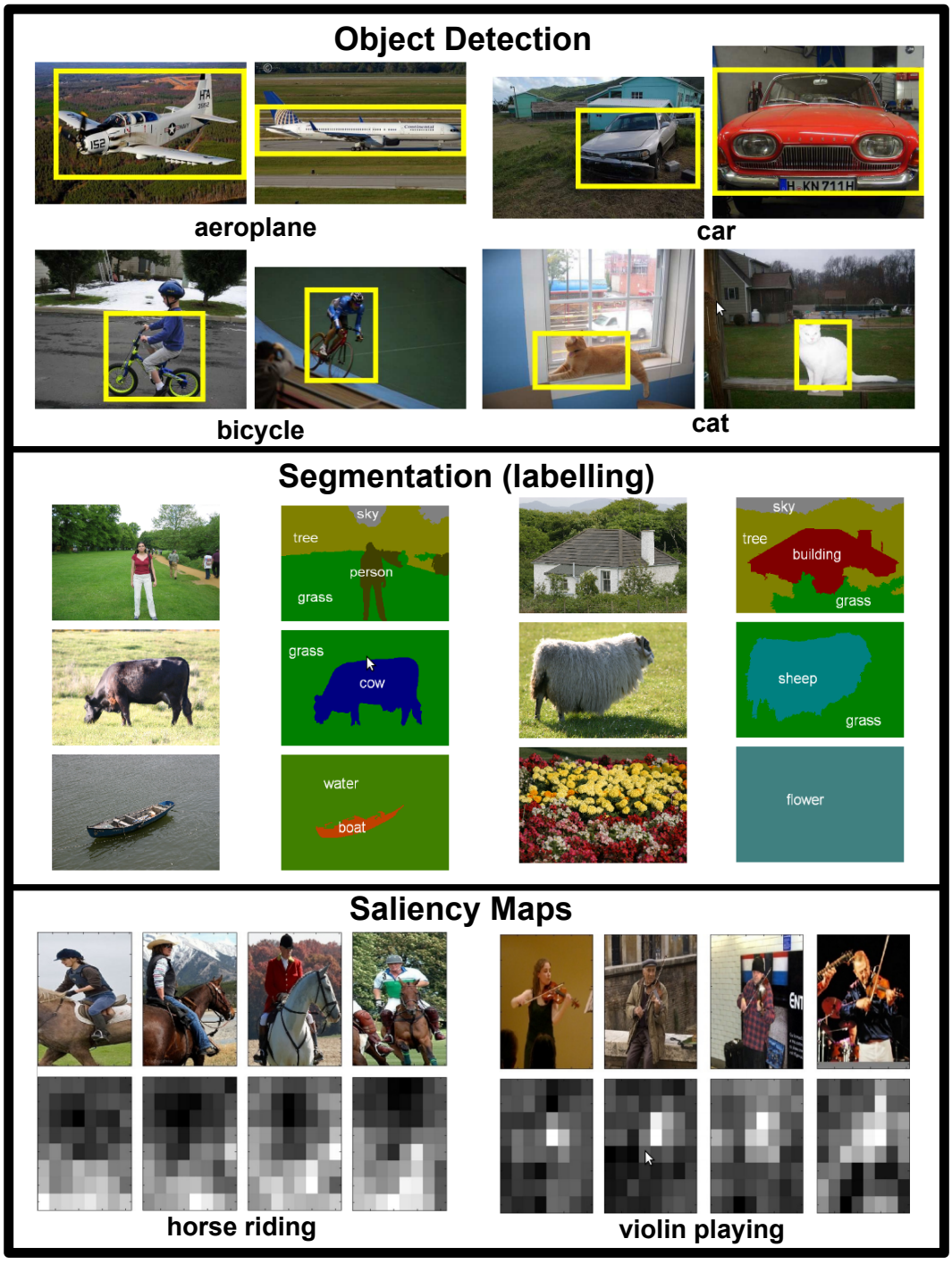


Figure 1.1 *Decomposition of an image into meaningful concepts.* ***Object Detection*** *- decomposition of an image using object detectors,* ***Segmentation (labelling)*** *- decomposition by labelling each pixel of an image to a class.* ***Saliency Maps*** *- decomposition by assigning each region of an image a weight based on its utility for the task of action recognition. Image and the corresponding weights of its cells are shown. Higher intensity of the value represents value closer to 1. Figure courtesy [1, 2, 3].*

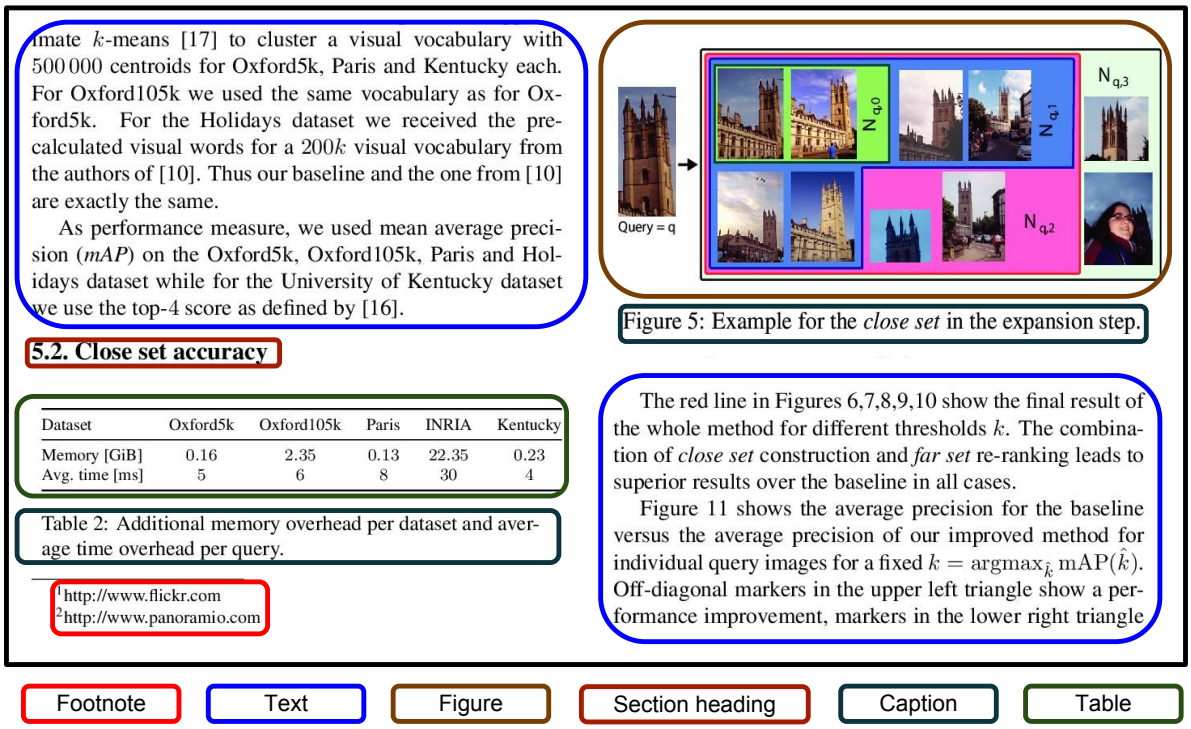


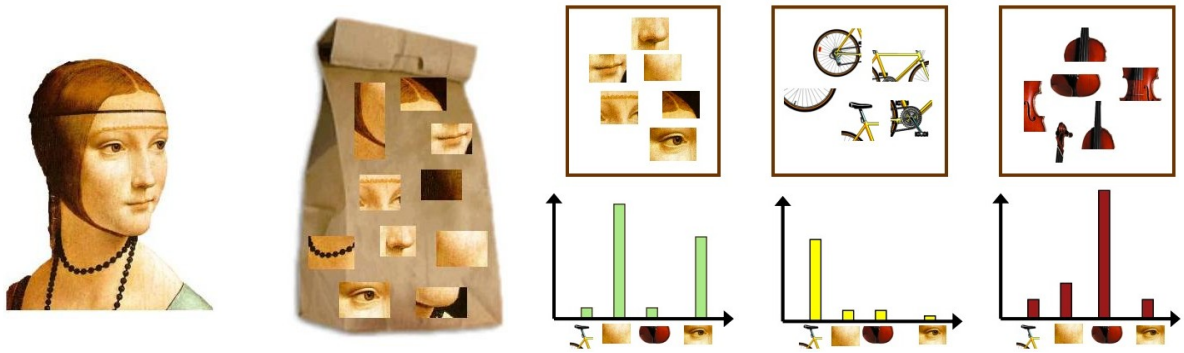
Figure 1.2 *Decomposition of a document image into meaningful concepts. In the figure, a document image is being decomposed into its logical and geometric components. This is also known as Document Layout Analysis.*

1.4 Decomposition in Images

Many of the computer vision tasks discussed in Section 1.2 can be considered as a decomposition task. We can consider image as a union of multiple parts and the aim is to decompose images into constituent meaningful parts. Meaningful parts may depend on the task which we are interested in and they generally gives deeper understanding of the semantic information concealed in the images. Object Detection and Semantic Segmentation (Figure 1.1) can be considered as a decomposition problem where we can consider images as composition of foreground and background. Foreground is tight bounding box around an object in case of object detection and a pixel level partition of an object in case of semantic segmentation.

For the task of classification also, we can conceive images as composition of more discriminative and less discriminative regions. For example, in the task of action recognition, we can decompose images into weighted regions based upon their discriminativeness to the task (in case of violin playing action, regions near the violin and hand will help in better classification of action than the ones near the face of person). Figure 1.1 explains this decomposition scheme pictorially.

In document images also, we can define the similar analogy of decomposition. For document layout analysis problem, images can be considered as a composition of textual and non-textual zones. Further,



(a) (b)

Figure 1.3 *(a) An image and its typical Bag of Words representation (ordering doesn't matter). (b) Histogram representation of the images using visual words obtained from clustering the local feature descriptors of the images in dataset.*

textual zones can be considered as a conjunction of titles, captions, footnotes, etc. and non-textual zones as a conjunction of figures, tables, math symbols, etc. The example of such a decomposition is shown in Figure 1.2. In case of word spotting, we can consider handwritten document images as a composition of regions containing query words and regions which do not contain them. Thus, decomposition can be defined in document images also.

1.5 Focus of the Thesis

The focus of this thesis is on solving decomposition problems in an optimization framework. As discussed, many computer vision problems can be contemplated as a decomposition problem, however, in this thesis, we focus on two decomposition problems – (i) decomposing a global histogram of an image into histograms of its associated objects and regions, (ii) decomposing a questioned document image into regions containing copied and non-copied/original contents in a recognition free setting. We perform decomposition in a linear optimization framework.

Histograms are popular methods for representing images. Histograms are the frequency of occurrence of observations in the given range. In case of images, it can be color histograms, gradient histograms, gray-scale histograms, etc. However, in order to make our representation more robust, we have used Bag of Words representation for images in our work. They are equivalent to Bag of Words (BoW) representation of documents in text domain. BoW representation for images was first introduced in [11, 12]. In text based BoW models, the documents are represented as a histogram of words in the database without storing the order of words. Based on the similar hypothesis, we first extract local feature descriptors (e.g., SIFT, LBP, etc.) in images, generate vocabulary by applying clustering to the above obtained descriptors (clusters thus obtained are known as visual words) and then, represent images as the histograms of visual words. We will explore the details of each step in Chapter 2, when the BoW models are discussed for the task of classification. In many datasets, BoW representation for images

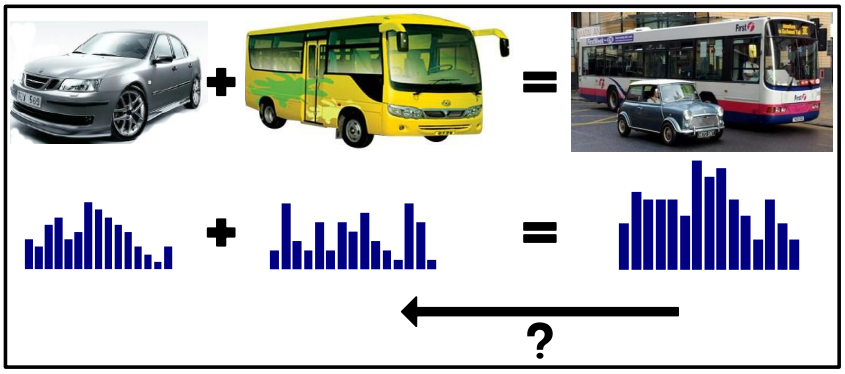


Figure 1.4 ***Decomposing Bag of Words Histograms.*** *We are interested in obtaining the constituent histograms from a composite histogram.*

produces state of the art results for the task of object classification and recognition [7, 13]. Figure 1.3 shows the BoW and histogram representation of an image.

Consider an image which contains two objects: a Bus and a Car (Figure 1.4). With the presence of multiple objects and clutter, often the global image representation (BoW) gets corrupted which results in deterioration of classification performance. A question of interest to us now is the following – Is it possible to filter out the clutter and classify only the signal? In this thesis, we propose a linear optimization framework to decompose a global histogram of an image into multiple histograms corresponding to different categories present in the image (bus, car and background/context in case of Figure 1.4). It must be noted that the detection and class based image segmentation (discussed in Section 1.2) can be adapted to solve the histogram decomposition problem. This involves two steps: (i) Performing object detection or segmentation; and (ii) Computing the individual histograms for the classes using the bounding boxes or segmentation masks obtained. In this thesis, we show that using detection or segmentation for solving the histogram decomposition problem would be an overkill as we are only interested in the histogram of constituent objects not the tight bounding box or the pixel level segmentation of an object. In this work, we present a simpler and computationally efficient alternative for this problem. We evaluate our method on a wide variety of composite histograms.

We solve the problem of decomposing a questioned document image into regions containing copied and non-copied contents in a recognition free setting by detecting documents from the database which contains exact or similar text (near duplicate) to a questioned document image. With the emergence of internet and large document repositories, many documents get created by duplicating contents from existing documents. Exact duplicate is the direct cut and paste of content from multiple documents in the database and near duplicate document segments could arise due to various document manipulations like summarization, copying, rewriting, editing, formatting, cut-and-paste, etc. In this thesis, for detecting duplicate documents, we propose two solutions in an optimization framework. Our first solution is based on finding a mixture of homographies which matches a questioned document image with multiple

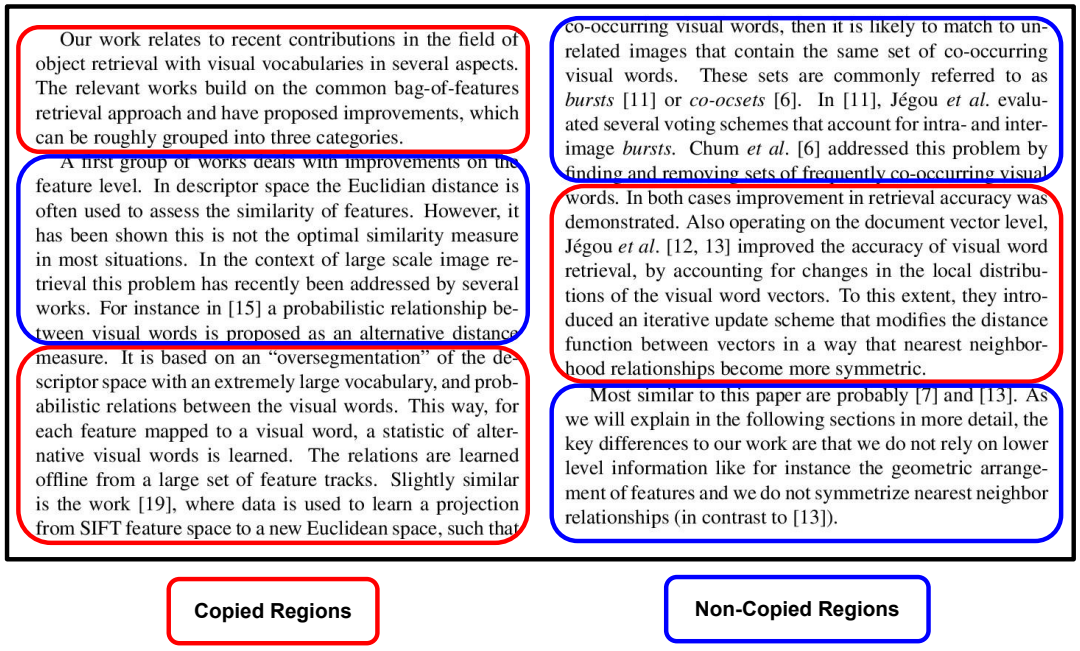


Figure 1.5 ***Decomposing a document image.*** *We are interested in decomposing a document image into copied and non-copied regions.*

documents in the database. Solution to fitting one homography between two images is well known [14]. Simultaneously fitting multiple homographies to match parts in different documents is also a novel contribution of this thesis. Our second solution is based on more flexible BOW settings which represents documents as histograms and learns a discriminative classifier which can effectively separate it from other documents. Using this classifier, we find duplicates in the database. Figure 1.5 illustrates the decomposition we are interested in.

1.6 Challenges

Every vision system suffers from the challenge of describing visual content in images because of various kinds of variations. This includes illumination variations, pose and view point changes, occlusion, truncation, scale and size variations, background clutter, intra and inter class variations and other deformations.

Apart from these challenges, one of the major challenge was to formulate the problems mathematically in a linear optimization framework. We have to make sure it is fast enough and at the same time can be solved accurately. Another challenge was how to efficiently use the background histogram obtained from histogram decomposition for the task of object classification. One approach can be to completely neglect the background histogram. However, completely ignoring the background histogram would be

inappropriate as it also contains context which provides strong cues for classification [15, 16] (e.g., roads for the classification of buses, water for the classification of boats, fields for the classification of animals, etc.). Chapter 3 discusses in detail how the background histogram is incorporated in the representation of object.

Another challenge was the creation of datasets for both the tasks – histogram decomposition and copy detection in document images. For histogram decomposition, we have also evaluated our results on images downloaded from Flickr containing multiple objects. We had to go through the heavy manual filtering process to create such a dataset. Also, there is no publicly available dataset for the task of duplicate detection or plagiarism detection in document images. We had to generate our own synthetic images and evaluate our method. We also had to take help from as many as 20 users for the creation of dataset for copy detection.

1.7 Organization of Thesis

The brief outline of the thesis is as follows: In Chapter 2, we present literature related to decomposition in images (recent works for the task of classification, segmentation, detection, etc.). We also introduce the datasets & evaluation protocol used. We also explain some of the technical details related to feature representation, classifier learning and optimization such as MRF, Linear Programming (LP), etc. In Chapter 3, we introduces our histogram decomposition solution and compare our method with various other techniques on standard datasets. In Chapter 4, decomposition in case of document images is discussed. In Chapter 5, we conclude the final remarks.

Chapter 2

Background

In this chapter, we do the survey of various computer vision tasks in which the decomposition plays a pivotal role. We present background knowledge for the task of classification, detection and segmentation. We also introduce some of the basic tools that we have used throughout the thesis. We also discuss the linear programming (LP) which has been used as a major tool for solving decomposition problems in this thesis and the standard methods for solving LP as it will be helpful in grasping the optimizations proposed in the later chapters.

2.1 Decomposition as Optimization

Optimization problems lie at the core of most machine learning approaches. In the last decade or so, optimization methods have been extended to new learning models and paradigms. Different methods have been employed to create novel models for problems such as uncertain and missing data, and selection of hypothesis. Methods have been developed in order to incorporate domain knowledge into graphical models in the form of constraints and to enforce sparsity in dimensionality reduction methods.

Learning plays an important role in computer vision and in the past, many computer vision tasks have been solved with the help of different optimization techniques. Sharma et al. [3] decompose images into regions based on their discriminativeness by solving an optimization problem in an iterative manner. All the recent works on semantic segmentation and labelling [17, 18, 5, 19, 20, 21, 22, 23, 24] is based on energy minimization which itself is a optimization problem. The state-of-the-art object detection method solves the optimization formulation using stochastic gradient descent method [4].

On the top of above, there have been many works which employs linear programming as optimization for solving the problem. Li [25] solves the problem of multibody motion segmentation using linear programming relaxation. Lempitsky et al. [26] does the image segmentation with a bounding box prior using linear programming. Also, there exists linear programming formulation for the graph-cut based energy minimization for image segmentation. Zhou et al. [27] proposed linear programming based

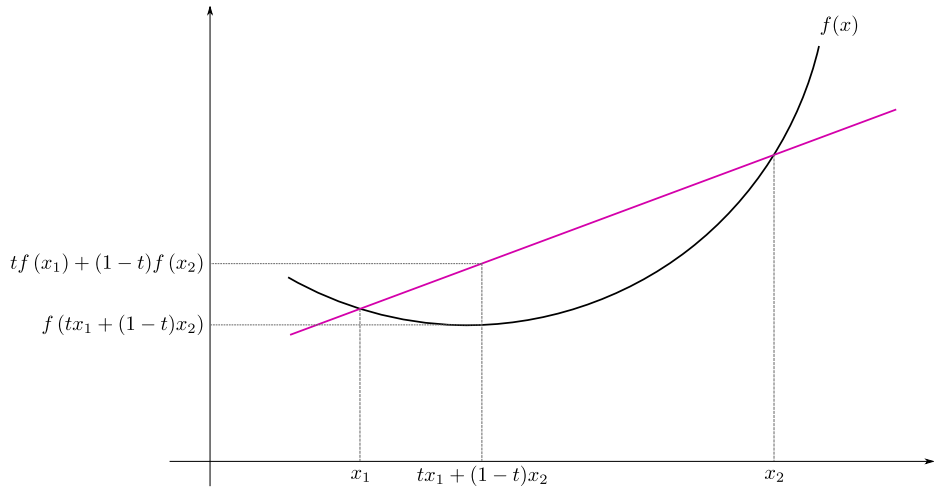


Figure 2.1 *Example of a convex function on an interval. (courtesy - Wikipedia)*

support vector machines. Guo et al. [28] does the simultaneous feature selection and classifier training via linear programming. Tsuda et al. [29] solves the problem of image denoising by linear programming.

In this section, we review some of the common optimization formulations which we have used or discussed in the upcoming chapters and the techniques to solve them.

2.1.1 Convex Optimizations

The convex optimization is a subfield of optimization which minimizes convex functions over convex sets. The convexity property makes the problem easier then the general case as the local minima will also be a global minima in this case. Mathematically, convex optimization is of the form -

$$\begin{aligned} &\text{minimize} \quad f(x) \\ &\text{subject to} \\ &\quad g_i(x) \leq 0, \quad i = 1, 2, \ldots, m \end{aligned}$$

where $f, g_1, g_2, \ldots, g_m : \mathcal{R}^n \rightarrow \mathcal{R}$ are convex. A real valued function $f : X \rightarrow \mathcal{R}$ defined on a convex set X in a vector space is called convex if, for any two points x_1 and $x_2 \in X$ and any $t \in$[0 1] (Figure 2.1) -

$$f(tx_1 + (1-t)x_2) \leq tf(x_1) + (1-t)f(x_2)$$

One of the common techniques to solve convex optimization is by using Lagrange duality. The Lagrange function of the above optimization is given by -

$$\mathcal{L}(x, \alpha) = f(x) + \sum_{i=1}^{m} \alpha_i g_i(x)$$

where x is referred as the primal variables of the Lagrangian and α_i's (≥ 0) are known as dual variables. Intuitively, the Lagrangian can be thought of as a modified version of the objective function to the

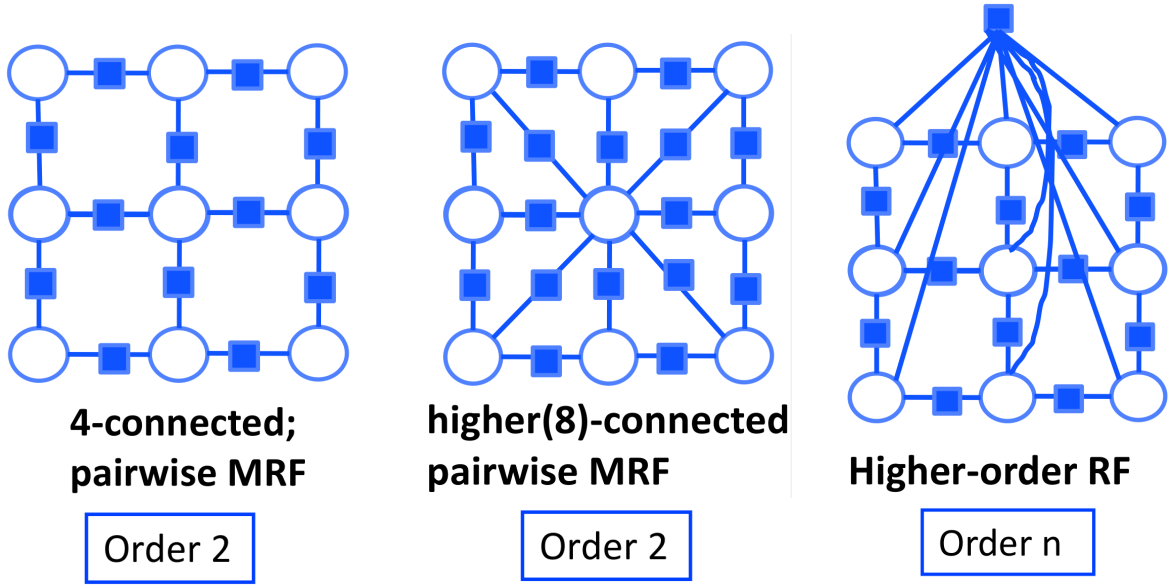


Figure 2.2 *Examples of 2nd and higher-order terms in energy function of* MRF.

original convex optimization problem which accounts for each of the constraints. The Lagrangian theory says that for any convex optimization problem, the unconstrained minimum of the Lagrangian with respect to the primal variables (keeping the dual variables fixed) is equal to the solution of the original constrained minimization problem. Using this property, one can get the solution of original optimization by solving Lagrangian.

2.1.2 Markov Random Fields

In Bayesian statistics, a maximum a posteriori probability (MAP) is used to obtain a point estimate of an unobserved quantity on the basis of empirical data. Mathematically, $P(w|f)$ measures the probability of a estimate/labelling given the observed feature f. Our goal is to find an optimal labelling $\hat{w}$ which maximizes $P(w|f)$. This is known as MAP estimate.

$$\hat{w}^{map} = \text{argmax}_{w \in \Omega} P(w|f)$$

By Bayes Theorem -

$$P(w|f) = \frac{P(f|w)P(w)}{P(f)} \sim P(f|w)P(w), \quad \text{since } P(f)\text{is constant}$$

For defining $P(f|w)$ (likelihood) and $P(w)$ (prior), Markov Random Fields (MRF) are used. MRF is a undirected graphical model in which each node corresponds to a random variable or a collection of random variables, and the edges identify conditional dependencies. Generally, it is used to model prior probabilities in Bayesian statistical inference. Let us assume the graph has n sites (S) and the labelling

field $X = (x_1, x_2, \ldots, x_n)$ can be modelled as a Markov Random Field if it satisfies the following two properties -

$$\forall w \in \Omega : P(X = w) > 0, \text{where } \Omega \text{ denotes the label set of all sites S}$$
$$P(x_i|X) = P(x_i|\{x_j\}), j \in \mathcal{N}_i, \text{where } \mathcal{N}_i \text{ denotes neighbourhood of } i$$

Also, a random field follows a Gibbs distribution. Therefore -

$$P(w) = \frac{1}{Z} exp(-U(w)) = \frac{1}{Z} exp(-\sum_{c \in C} V_c(w))$$

where Z is a normalization constant. A subset C $\subseteq$ S is called a clique if every pair of pixels in this subset are neighbours. The clique containing n pixels is called nth order clique, denoted by C_n. Examples of 2nd and nth order cliques are shown in Figure 2.2. Thus, the total set of cliques is given by $C = C_1 \cup C_2 \cup \ldots C_n$. $V_c(w)$ is called the clique potential of c, where w is called the configuration of labelling field. Thus, energy $U(w)$ of the configuration w is -

$$U(w) = \sum_{c \in C} V_c(w) = \sum_{i \in C_1} V_{C_1}(w_i) + \sum_{(i,j) \in C_2} V_{C_2}(w_i, w_j) + \ldots$$

Using above equation, we can define MRF models via clique potentials and we can get labelling/unobserved quantity by minimizing above energy function.

2.1.3 Linear Programming

Linear Programming or Linear Optimization or LP is a mathematical method to determine a way to achieve the best outcome in a given list of requirements as linear relationships. Linear Optimization is a specific case of convex optimization. In a more formal way, linear programming is a mathematical technique for the optimization of a linear objective function, subject to linear equality and linear inequality constraints. The feasible region of the LP is a convex polyhedron which is the intersection of finitely many half spaces, each of which is determined by a linear inequality. The objective function is a real valued affine function defined on the convex polyhedron. An algorithm for LP will find the optimum point in the convex polyhedron if it exists. Linear optimization is of the form -

$$
\begin{aligned}
&\text{minimize} \quad c_1x_1 + c_2x_2 + c_3x_3 + \ldots + c_nx_n \\
&\text{subject to} \\
&a_{11}x_1 + a_{12}x_2 + a_{13}x_3 + \ldots + a_{1n}x_n \leq b_1 \\
&a_{21}x_1 + a_{22}x_2 + a_{23}x_3 + \ldots + a_{2n}x_n \leq b_2 \\
&a_{31}x_1 + a_{32}x_2 + a_{33}x_3 + \ldots + a_{3n}x_n \leq b_3 \\
&a_{41}x_1 + a_{42}x_2 + a_{43}x_3 + \ldots + a_{4n}x_n \leq b_4 \\
&\ldots\ldots\ldots \\
&a_{m1}x_1 + a_{m2}x_2 + a_{m3}x_3 + \ldots + a_{mn}x_n \leq b_m \\
&x_1 \geq 0, x_2 \geq 0, x_3 \geq 0, \ldots, x_n \geq 0,
\end{aligned}
$$

The above problem has n optimization or decision variables $x_1, x_2, x_3, \ldots, x_n$, and a linear cost or objective function $c_1x_1 + c_2x_2 + c_3x_3 + \ldots + c_nx_n$. The inequalities are called the constraints of the problem. A vector $x = (x_1, x_2, \ldots, x_n)$ is feasible if it satisfies all the constraints. A feasible point is a solution if there exists no feasible point with a lower value of the objective function. If x is a solution, then $c_1x_1 + c_2x_2 + c_3x_3 + \ldots + c_nx_n$ is known as the optimal value of the LP. The feasible minimum problem is said to be unbounded if the objective function can assume arbitrary large negative values at feasible vectors otherwise it is said to be bounded.

Generally, it is convenient to write the problem using matrix-vector notation which is also the standard notation for LP-

$$
\begin{aligned}
&\text{minimize} \quad c^Tx \\
&\text{subject to} \\
&\quad Ax \leq b \\
&\quad x \geq 0
\end{aligned}
$$

where A is a $m \times n$ matrix with entries a_{ij}. The inequalities are interpreted elementwise, i.e., $Ax \leq b$ is equivalent to $(Ax)_i \leq b_i$ for $i = 1, 2, 3, \ldots, m$. Note that all the linear programming problems can be converted to the standard form. Most of the LP solvers are based on the standard notation only, i.e., they accept the input A, b, c and gives x as an output.

Linear programming has wide variety of applications and can be applied to various fields of study. Some of them include business, economics, engineering problems, transportation, energy, telecommunications, manufacturing, planning, routing, scheduling, assignment and design. In this thesis, we have shown the application of linear optimization in the field of computer vision.

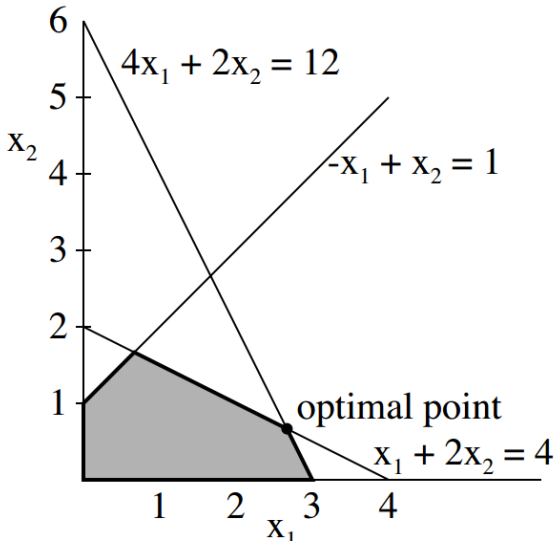


Figure 2.3 *Constraint set or feasible region of a simple linear program. The optimal solution of the* LP *occurs at corner points. In this case, (8/3,2/3) gives the optimal solution.*

Consider a simple example of linear program -

$$\begin{aligned} &\text{max} \quad x_1 + x_2 \\ &\text{subject to} \\ &\quad x_1 + 2x_2 \leq 4 \\ &\quad 4x_1 + 2x_2 \leq 12 \\ &\quad -x_1 + x_2 \leq 1 \end{aligned}$$

Since, there are only two variables, we can solve this problem graphically. Each inequality constraint is satisfied by a half-plane of points, and the constraint set is the intersection of all the half-planes. In the present example, the constraint set is the five-sided figure shaded in Figure 2.3. Optimal solution of a linear program always occurs at a corner point of a constraint set, provided that the constraint set is bounded. Occasionally, the optimum point occurs along an entire edge or face of the constraint set, but then it occurs at a corner point as well. In Figure 2.3, (8/3,2/3) gives the maximum value of the objective function, hence, it is the optimal point.

Not all linear programming problems are so easily solved. There may be many variables and many constraints. The brute force way to solve a linear programming problem is to find all the extreme points of a system and see which one correctly optimizes the problem. The number of extreme points can be very large in case of industrial scale problems and hence, this approach is unrealistic. Instead, the Simplex algorithm is the most common way to solve linear programs and is the basis for most linear program solvers. Simplex algorithm selects an extreme point to start. Then, each iteration of the algorithm takes the system to the adjacent extreme point with the best objective function value. These iterations are repeated until there are no more adjacent extreme points with better objective function

values. That is when the system is at optimality and the algorithm terminates. More detail about Simplex can be found here [30].

Interior point methods are the another class of algorithm that can be used for solving linear programs. Interior point methods move through the interior of the feasible region unlike the simplex algorithm which finds an optimal solution by traversing the edges between vertices on a polyhedral set. Different solvers employ different algorithm to solve linear programs. Some of the common LP solvers are – MATLAB, glpk, CVX, MOSEK, AIMMS, CPLEX, etc. We have used MOSEK in our work which uses interior point method to solve linear programs.

2.1.3.1 Integer Programming

In the linear programming, all the variables have been continuous, in the sense that decision variables are allowed to be fractional. An integer programming formulation is a mathematical optimization in which some or all of the variables are restricted to be integers. Thus, the standard formulation is -

$$
\begin{aligned}
&\text{minimize} \quad c^T x \\
&\text{subject to} \\
&\quad Ax \leq b \\
&\quad x_i \geq 0 \quad (i = 1, 2, \ldots, n) \\
&\quad x_i \in \text{integer} \quad (\text{for all or some } i = 1, 2, \ldots, n)
\end{aligned}
$$

Integer programming problem is said to be mixed integer program when some, but not all, variables are restricted to integers, and is called a pure integer program when all decision variables must be integers. The reasons for using integer variables when modelling problems as a linear program are that the integer variables can represent quantities that can only be integers (for example, one cannot produce 7.6 auto-mobiles), and integer variables can represent decisions and so should only take on the value 0 or 1.

Integer programming is generally hard to solve and is NP-hard. While the simplex method is effective for solving linear programs, there is not single technique for solving linear programs. Over the years, a number of methods have been developed, and the performance of any particular technique appears to be highly problem-dependent. Some of the popular methods for solving integer programs include – branch and bound procedure, cutting plane methods, branch and cut, and heuristic based methods.

The simplest and efficient way to solve an integer linear programming is to solve using LP-relaxation where we simply remove the integral constraints on x_i, and then round the entries of the solution to the LP- relaxation. Simplex method can give integer solutions to the LP- relaxation problem in a special case when constraint matrix A is total unimodular. In this work, we have solved our integer programming formulations using LP- relaxation.

2.2 Classification and Recognition

The typical object recognition pipeline consists of the following steps – (i) Interest point detector and descriptors, (ii) quantization of local descriptors to form a vocabulary, (iii) encoding of local features in an image descriptor, and (iv) learning a classifier. The pipeline is explained diagrammatically in the Figure 2.7. We will discuss each of them in detail in the following subsections. Generally, the baseline performance for recognition is computed using spatial histograms (discussed later) and BoW model [12, 11, 31]. Over the years, people have proposed many techniques to boost the classification performance by improving on above mentioned steps [1, 32, 3, 33, 34].

2.2.1 Interest point detectors and descriptors

In this step, first the interest points are detected and then, the regions around them are represented in the form of local descriptors. The process of finding interest points in images is also known as feature detection. Each of these interest points are invariant to image translation, rotation, scaling, affine transformation, partially invariant to illumination changes and robust to local geometric distortion. Some of the examples of "interest point/ region of interest" detector include (i) Harris Points, (ii) Harris-Laplace regions, (iii) Hessian- Laplace regions, (iv) Harris-Affine, (v) Hessian-Affine and (vi) Maximally Stable Extremal Regions. The typical procedure for finding such interest points involves searching keypoints at multiple scales by obtaining so called Gaussian scale space followed by filtering. The details about them can be found here [35].

Although for the task of classification or recognition, uniform sampling of points at dense grid of pixels is proven to be more advantageous than the detected sparse set of points [7]. Figure 2.4 shows the difference between two strategies of feature detection. We have used uniform sampling of points in our experiments while performing classification task whereas sparse set of points are used while solving the corresponding problem which occurs due to view point variations of camera.

Now, the regions of interest detected above is represented in the form of local descriptors. We have used Scale Invariant Feature Transform (SIFT) [36] as a local descriptor in our work. SIFT is a local histogram computed using gradient magnitudes and direction around the point. The region around the interest point is quantized into 4×4 location grid and gradients are quantized into 8 bins. This results in a 128 dimensional vector representing an interest point. Figure 2.4 summarizes the approach for local features extraction.

2.2.2 Vocabulary Construction

Having computed the feature descriptors of the images, next step in the pipeline involves grouping similar features into clusters. This process is known as vocabulary/codebook/dictionary generation. This is akin to a dictionary of words in text domain. Here, it is dictionary of "visual" words. The purpose of dictionary is to represent images using these visual words. Most popular algorithm for clustering in

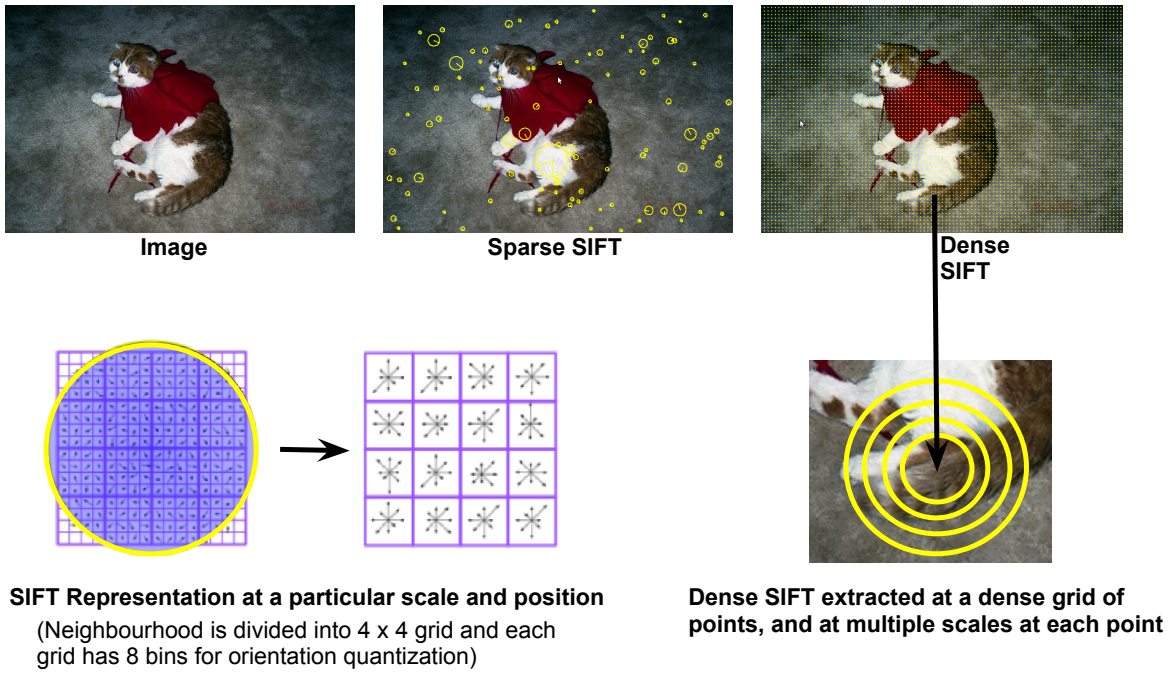


Figure 2.4 *Interest point detector and descriptor using* SIFT. *Example shows the detection and extraction of* SIFT *descriptors at sparse region of interests and at dense grid of points (5 × 5).*

the literature is K-Means clustering algorithm. Number of descriptors in training images set is generally very large, so a subset of samples are selected randomly and clustering is performed on them to create a vocabulary. Number of clusters is determined experimentally. K-means clustering is computationally very expensive and takes a lot of time to converge. To improve the convergence rate, hierarchical K-means (HiK-Means) is a popular technique which recursively applies K-means to compute finer and finer partitions [37]. We have used K-Means to train vocabularies of size less than 4k and HiK-Means to train vocabularies of size greater than 4k (∼25k).

2.2.3 Encoding of local features and Histogram Computation

In this step, all the local feature descriptors computed are mapped to the nearest visual word (using some distance metric) present in the vocabulary. This process is known as local feature encoding. This assigns every feature descriptor present in an image to a visual word. Now, the image is represented by a histogram which stores the distribution of all the visual words. The size of the histogram is equal to the size of vocabulary where each bin corresponds to a visual word. The advantage of the feature encoding is that the images containing different number of local descriptors are finally mapped to the same feature dimension as long as the underlying vocabulary remains the same. Loss of spatial information is one of the major disadvantages of this approach. Spatial information can be incorporated in the image

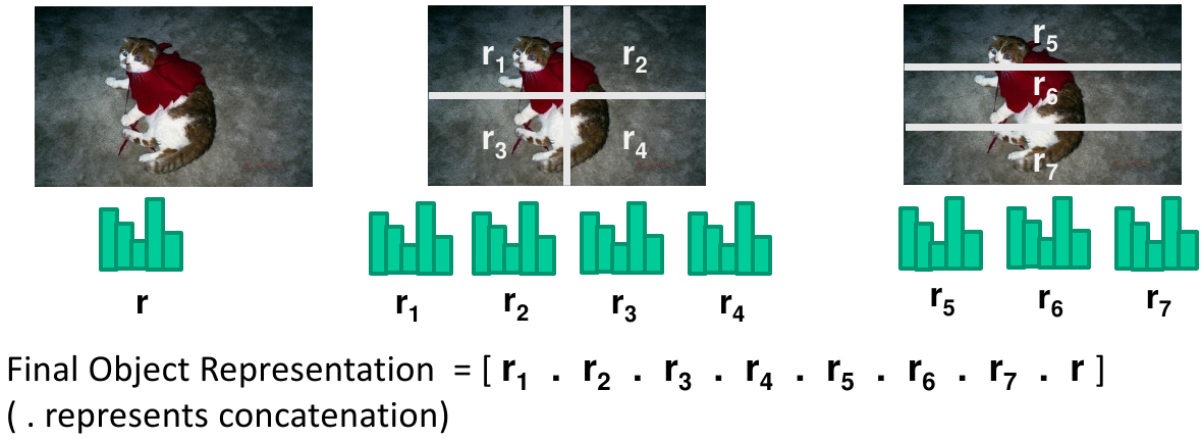


Figure 2.5 *Computation of Spatial Pyramid Histograms. An image is divided into multiple sub-regions at multiple resolutions and a histogram is computed at each spatial sub-region. In the figure, an image is divided into 1 × 1, 2 × 2 and 3 × 1 grids, resulting in a total of 8 regions. If the size of vocabulary is K, then the overall image representation will have size of $8K$.*

representation with the use of spatial pyramid for computing histograms [31] (discussed below).

Recent Improvisations:

- LLC encoding: The above encoding scheme where we associate each local descriptor to a single visual word is known as hard encoding or hard assignment. Whilst this provides reasonable expressive power but a single descriptor belongs to only one closest word in a dictionary. This yields a high quantization error. To some extent, soft assignment mitigates this effect by allowing soft contribution of each descriptor to its closest words in a dictionary. Although recently, sparse and local coding schemes have been shown to be a better choice for classification task. Locality-constrained Linear Coding (LLC) is one of the recently proposed local encoding scheme. We have used hard encoding and LLC in our experiments. LLC optimize a linear combination of few visual words to approximate a local feature and code it with the optimized coefficients [38].

- Spatial Histograms: Spatial binning is the standard way of introducing weak geometry in the histogram representation of an image. For computing spatial pyramid, an image is divided into increasingly fine sub-regions at multiple resolutions and histogram of local features is computed inside each sub-region. The final image representation is a concatenation of histograms obtained for each spatial region. It must be noted that the histogram of each region is first normalized before concatenation. As such there is no fixed formula to decide the spatial splitting of image into sub-regions. Splitting criteria used in our experiments is explained in Figure 2.5.

- Pooling: When computing the encoding of local feature descriptors for each spatial region, image features can be pooled in different ways – sum pooling, max pooling, average pooling, etc. We have used sum pooling for hard encoding and max pooling for LLC in our work. In case of sum

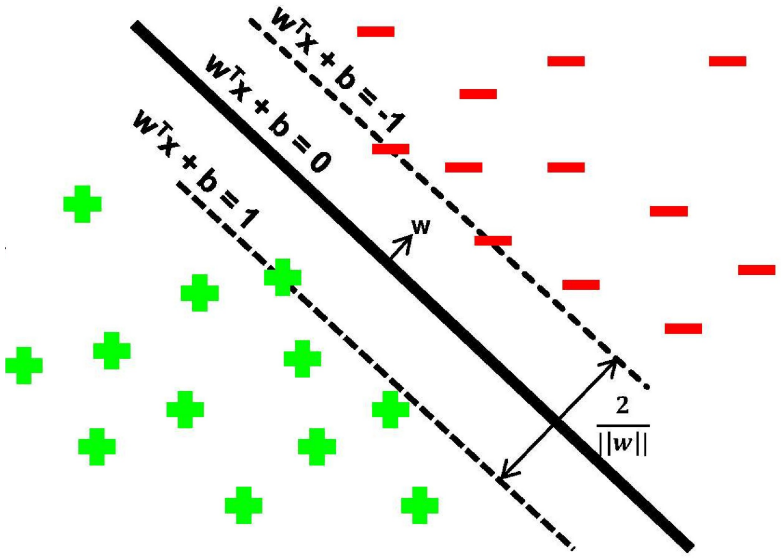


Figure 2.6 *Maximum margin hyperplane and the margins output by* SVM.

pooling, encoding of local feature descriptors in a given region are combined additively whereas for max pooling, each bin in the feature representation is assigned a value equal to the maximum across local descriptors encodings in that region [7].

2.2.4 Classifier Learning

The last step in the object classification/recognition task is to assign a class/category to the image feature descriptor obtained from above step. This can be done using two type of classifiers – generative and discriminative. Generative classifiers learn a model of the joint probability distribution $p(x, y)$ of the input x and label y, and make their decisions by using Bayes rule to calculate $p(y|x)$, and then, picking the most likely label y. Discriminative classifiers model the posterior $p(y|x)$ directly and give a direct map from inputs x to label y. In the past, discriminative classifiers have significantly outperformed generative classifiers for the task of classification. Support Vector Machine (SVM) is one of the most popular discriminative classifier and we have used it in all our experiments.

SVM is a supervised learning method that learns from a given set of labelled samples (training samples) and predicts label on a set of unlabelled samples (testing samples). It learns the decision boundary with the largest margin to the training samples. SVM basically learns a linear classifier which separate a K dimensional data using $K-1$ dimensional hyperplane. There may exists many hyperplanes which separates the data. The SVM choses the one which best generalizes the given set of training samples by maximizing margin. Maximizing margin is equivalent to maximizing the distance between the nearest

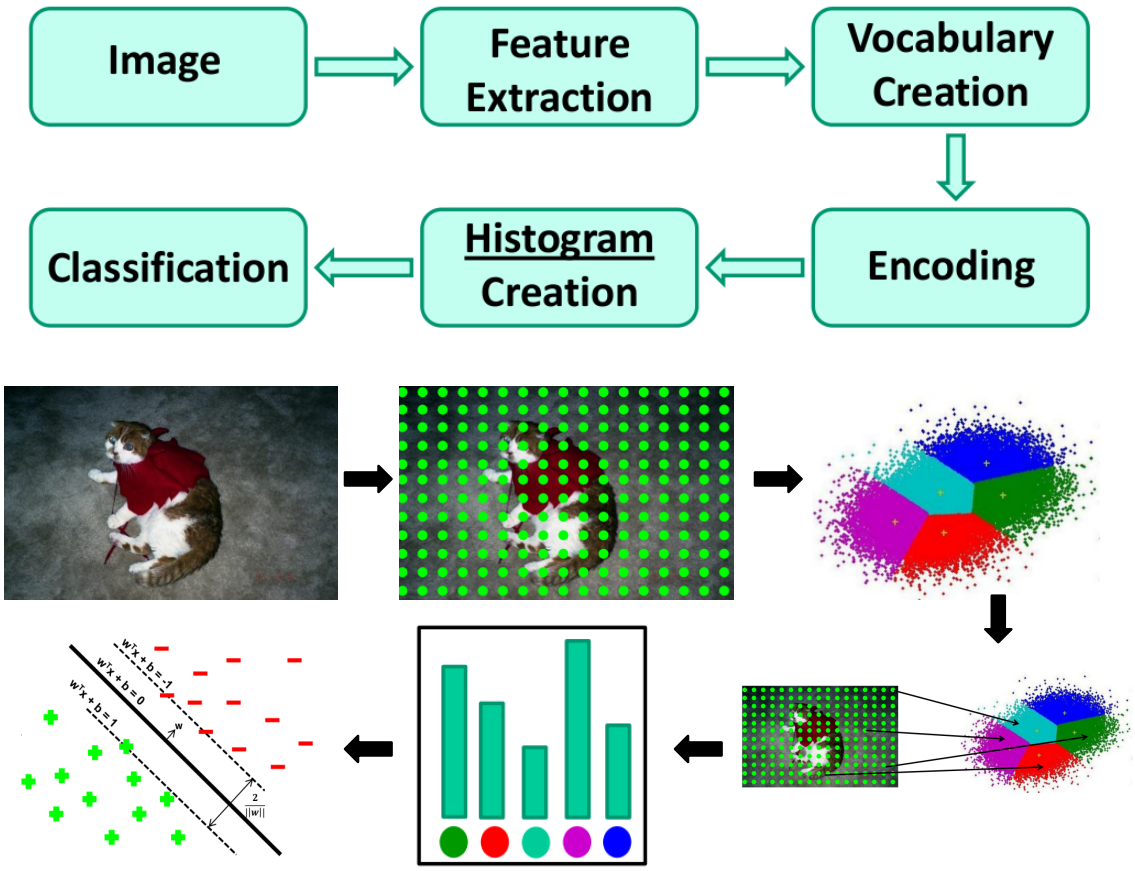


Figure 2.7 *Typical steps involved in performing classification of images which includes* BoW *representation of an image and classifying it to to a class/category.*

points on both sides of the hyperplane. The classifier thus obtained is known as the maximum-margin classifier.

Consider a set of training samples $\mathbf{T} = \{\mathbf{x}_i, y_i\}_{i=1}^{n}$ such that $\mathbf{x}_i \in R^K$ and $y_i \in \{-1, 1\}$, where $\mathbf{x}_i$ represents a K dimensional training sample and y_i its label. Now, the hyperplane which classifies training samples into its class can be written as $\mathbf{w}^T\mathbf{x}_i + b = 0$, where $\mathbf{w}$ is perpendicular to separating hyperplane and b is a bias term (Figure 2.6). $\mathbf{w}$ and b has to be chosen such that it maximizes margin. The two hyperplanes enclosing the margin can be written as, $\mathbf{w}^T\mathbf{x}_i + b = 1$ and $\mathbf{w}^T\mathbf{x}_i + b = -1$. The distance between the two hyperplanes is $\frac{2}{||\mathbf{w}||}$ and hence, the optimization problem will be to minimize $||\mathbf{w}||$. Also, there will be no data points between two hyperplanes which encloses the margin, giving rise to following set of constraints -

$$\mathbf{w}^T\mathbf{x}_i + b \geq 1, \forall i \text{ such that } y_i = 1$$

$$\mathbf{w}^T\mathbf{x}_i + b \leq -1, \forall i \text{ such that } y_i = -1$$

Now, the optimization problem can be written as -

$$\min_{\mathbf{w},b} \frac{1}{2}||\mathbf{w}||^2$$

such that

$$y_i(\mathbf{w}^T\mathbf{x}_i + b) \geq 1, \forall i$$

The above optimization can be solved using standard quadratic programming solvers and the solution we get is -

$$\mathbf{w} = \sum_{i=1}^{n} \alpha_i y_i \mathbf{x}_i$$

And the classification of sample **x** can be performed by -

$$y = sgn(\sum_{i=1}^{n} \alpha_i y_i \mathbf{x}_i.\mathbf{x} + b)$$

The above version of SVM model is known as hard margin SVM. The hard margin classifier will fail in the case of outliers or mislabelled samples (it will give infeasible solution). In the past, soft margin SVM has been proposed to overcome such challenges. Even if there exists no hyperplane splitting the positive and negative samples, the soft margin based classifier will chose a hyperplane that splits the samples as cleanly as possible while at the same time maximizing the margin of the cleanly split samples also.

Using SVM we can also create non-linear classifiers which can learn complex distributions of samples. This is done by applying the kernel trick to maximum margin hyperplane. Kernel-SVM can be formulated similar to above except that the dot product of $\mathbf{x}_i.\mathbf{x}$ is replaced by a kernel function $K(\mathbf{x}_i, \mathbf{x})$. Now, the classification of sample $\mathbf{x}$ is done using -

$$y = sgn(\sum_{i=1}^{n} \alpha_i y_i K(\mathbf{x}_i, \mathbf{x}) + b)$$

It must be noted that the time complexity of Kernel-SVM is much larger than the complexity of Linear-SVM as Kernel-SVM involves the computation of kernel functions (y depends on summation over all "support vectors") during test time whereas in Linear-SVM, **w** can be pre-computed and hence, only a dot product has to be performed during test time. Although Kernel-SVM have much higher accuracy than Linear-SVM in practice. Figure 2.8 compares the computation time and discriminativeness of different kernels. Linear kernel is fastest but its discriminativeness power is low.

We want higher accuracy as well as want to enjoy the high testing speed of Linear-SVMs in the same algorithm. This can be achieved using additive kernels. Additive kernels are way to speed up non-linear SVMs. It avoids support vector expansion and actually computes the feature map. Exact feature map is usually hard to compute and has very high dimension. Hence, the researchers seek for approximations. Additive kernels are powerful representations that enables the derivation of closed form feature maps based on Fourier Analysis. For additive kernels, we can computed the finite, low dimensional, tight

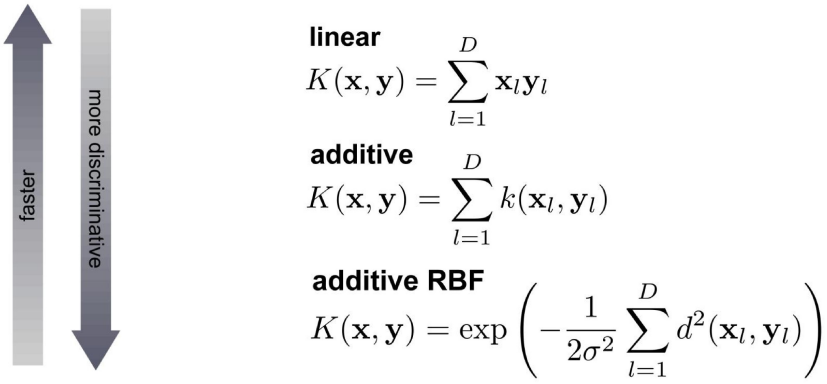


Figure 2.8 *Comparison of time complexity and discriminative power of different types of kernels. For additive kernels, this tradeoff can be resolved more or less with the use of explicit feature maps.*

approximations (homogeneous kernel maps) of the given feature maps. Using this, linear kernel can be applied directly to the obtained homogeneous kernel maps. Some of the examples of additive kernels are Hellinger's kernel, intersection kernel and χ^2 kernel [39]. Homogeneous kernels maps (also known as explicit feature maps) can be computed efficiently for these kernels. We have used χ^2 kernel in our experiments.

2.3 Object Detection: Deformable Parts Model

Object Detection and localizing generic objects in static images is one of the fundamental challenges in computer vision. Object localization is difficult problem because objects can vary significantly in appearance. Variations arise not only due to illumination and viewpoint changes but also due to non-rigid deformations, intraclass variability, shape and sizes of objects.

Sliding window based approaches has emerged as one of the most popular methods in the past for performing object detection. They scan the images at multiple scales and locations and determines whether or not there is an instance of target object at the given scale and position. The Dalal and Triggs detector [40] is the first powerful method introduced which uses sliding window mechanism to detect pedestrians in static images. They employ a single filter on histogram of oriented gradient (HOG) features to represent an object category.

Another way to represent objects is using part based representation where object is decomposed into parts and spatial relations among parts. Pictorial structures [41, 42] and grammar based [43, 44, 45] models are an example of such a representation framework. Deformable part models [4] is an another example of such representation and is currently, the state-of-the-art object detection model available. They have won the PASCAL object detection challenge for the consecutive 5 years. Deformable part models represents objects using a star-structured part-based model defined by a root filter (analogous to the Dalal and Triggs filter) along with a set of part models and associated deformation models. The

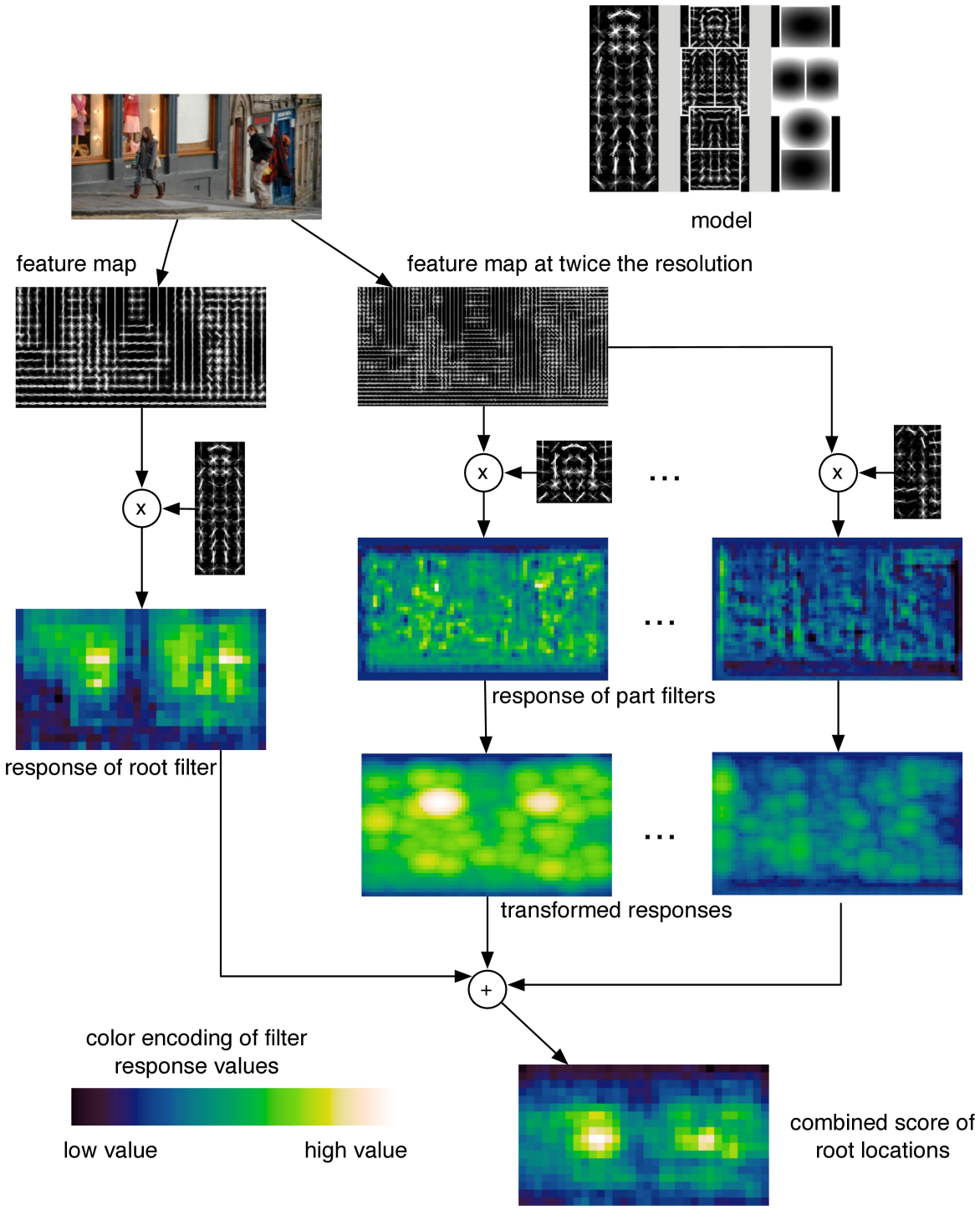


Figure 2.9 *Figure showing the matching at one scale. Root and part filter responses are computed at different resolutions in the feature pyramid and the responses are combined to yield a final score for each root location. Figure courtesy [4].*

score of the star based model at a particular position and scale within an image is the score of the root filter at the given location plus the sum over parts of the maximum, over placements of that part, of the part filter score on its location minus a deformation cost measuring the deviation of the part from its ideal location relative to the root. Both root and part filter scores are defined by the dot product between filter (a set of weights) and a subwindow of a feature pyramid computed from the input image. The part filters capture features at twice the spatial resolution relative to the features captured by the root filter to model visual appearance at multiple scales. Figure 2.9 summarizes the approach.

The training of such a part based model from images is done only with bounding boxes around the objects of interest. Since, the part locations are not available, they are treated as latent variables in the formulation during training. Thus, the part based models find the location of parts automatically in the training images. and it saves us from doing more elaborate labelling (of parts) which is expensive, time consuming and tedious. To train models, latent variable formulation of MI-SVM is used. In MI-SVM, example x is scored by a function -

$$f_\beta(x) = \max_{z \in Z(x)} \beta \,.\, \phi(x, z)$$

where β is a model parameter vector, z are latent variables, and $\phi(x, z)$ is a feature vector. In this case, β is the concatenation of root filters, part filters and deformation costs, z is a specification of object configuration, and $\phi(x, z)$ is the concatenation of features from root and part feature pyramid.

Because of huge variation in the shapes and sizes of object, a single star structured model is generally not enough to model all the variations. Hence, instead of using a single star-structured model, mixture of star models was developed. At a particular position and scale, score of a mixture model is maximum over all the star structured models learned for that category. In this way, deformable part models can model objects of different shapes, aspect ratios and appearance without having to fit a single star structured for all of them. Figure 2.10 shows some of the object detection results using deformable part models.

2.4 Semantic Segmentation: Automated Labelling Environment

Segmentation refers to a task of assigning a category label to each region/pixel. Again, this is another challenging computer vision problem because of great variety in appearance, size, shape and no notion of the inherent semantics. MRF based graph cut energy minimization has received significant attention in the last decade from researchers for solving the problem of segmentation. In this technique, the properties and features of an image is represented in the form of graph. Every pixel in the image represents a node in the graph and is connected to label nodes by means of "data term" and is also connected to other neighbouring nodes by the "pairwise term" or "smoothness term". The graphical representation is illustrated in Figure 2.11. In typical segmentation of an image into foreground and background, label set $\mathcal{L} = \{0,1\}$, label 0 indicates pixel has been assigned to background and label 1 indicates pixel has been assigned to foreground. Generally, for the task of semantic segmentation, the

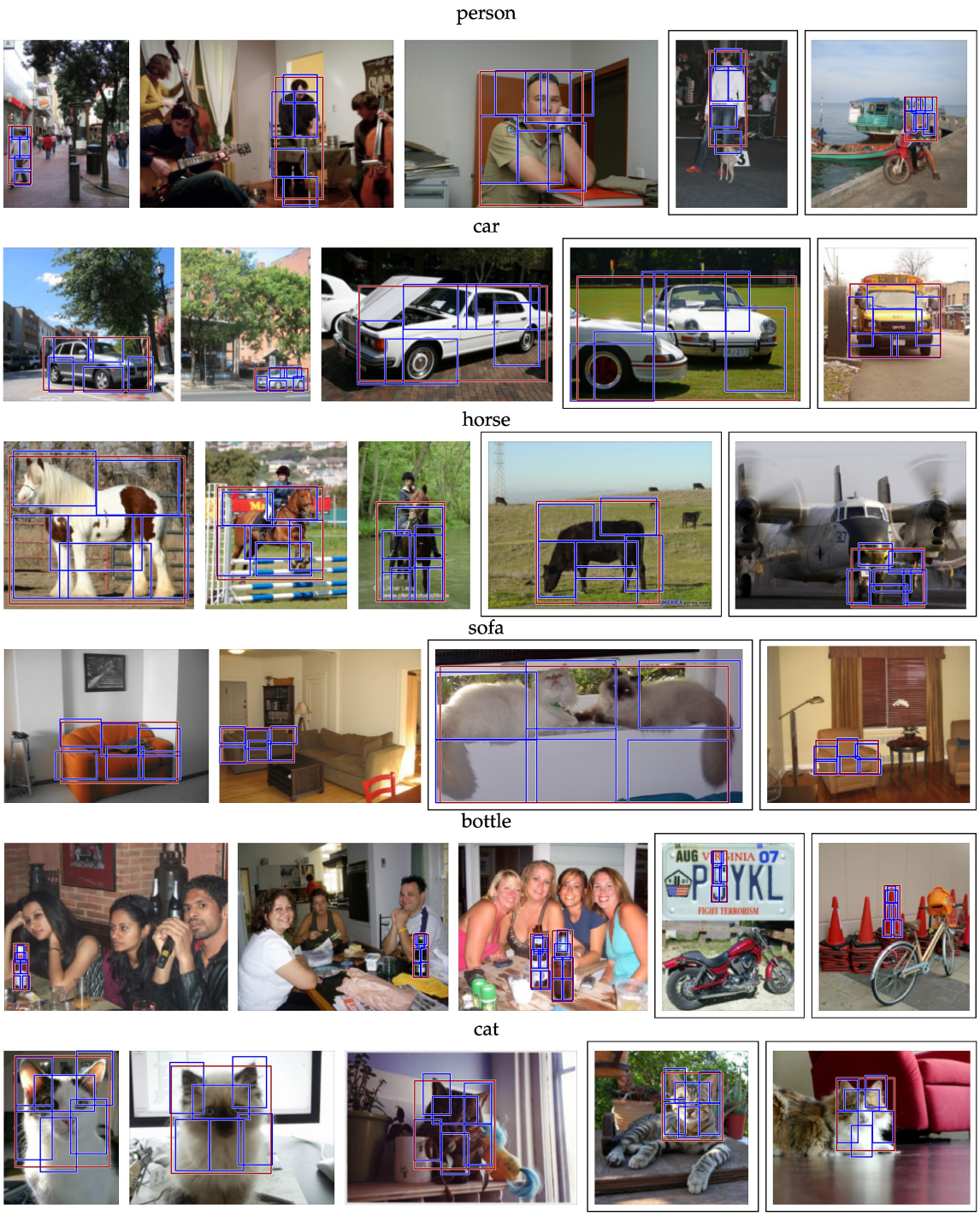


Figure 2.10 *Detection results on few of the images from* PASCAL VOC 2007. *The last two images in each row illustrates the false positives. Figure courtesy [4].*

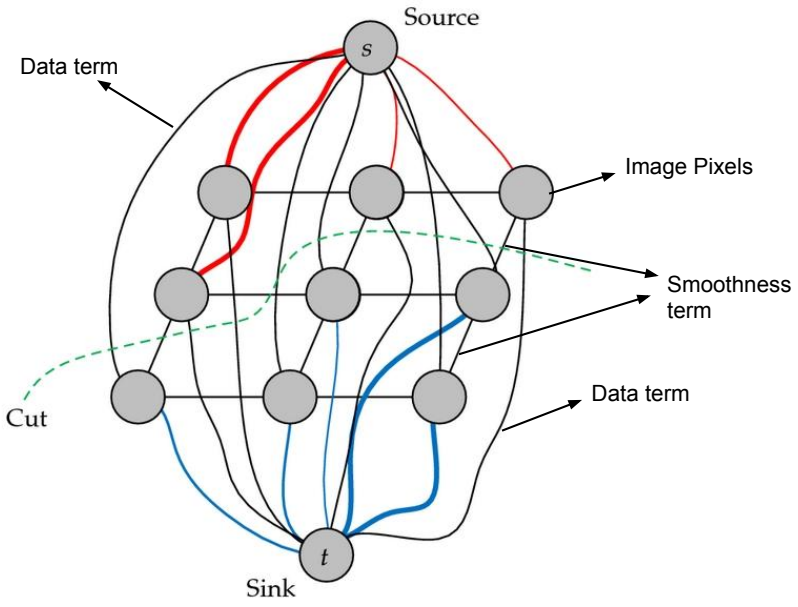


Figure 2.11 *Graphical representation of the properties of an image. Red line shows pixels belonging to source (foreground) and blue line shows the pixels belonging to sink (background). Data term connects pixels to source and sink whereas smoothness term connects pixels to its neighbouring pixels. Graph cut gives the optimum value of the energy function and thus, generates the segmentation of image.*

data term or likelihood (in MRF) is represented by Gaussian distributions and smoothness term or prior (in MRF) is determined by clique potentials.

Mathematically, energy function is written as -

$$E(f) = \sum_{p} D_p(f_p) + \sum_{p} \sum_{q \in \mathcal{N}_p} V_{pq}(f_p, f_q)$$

p denotes a pixel and at a pixel p, f is the label, D is the data term, V is the smoothness term and $\mathcal{N}$ is the neighbourhood. The above energy function for two label case can be optimized using graph cut. For multilabel segmentation, we can create a graph with k terminal nodes and the minimum of the energy corresponds to the multiway cut of the graph. Multiway cut is an NP hard problem with no easy solutions. Move making algorithms such as α-expansion and $\alpha\beta$-swap are used to get approximate solutions. Some other popular techniques are TRW-S message passing and belief propagation. In our work, we have used TRW-S algorithm for solving MRF formulations. CRFs which are notable variants of MRFs are also common for the segmentation. In CRF, each random variable may also be conditioned upon a set of global observations, i.e., in CRF, both data and smoothness term are functions of all the observation data as well as that of labels whereas in MRF, smoothness term is function of labels only. MRF models the joint probability distribution $p(x, y)$ and then, infers $p(x|y)$ while CRF is a discriminative model and directly model $p(x|y)$.

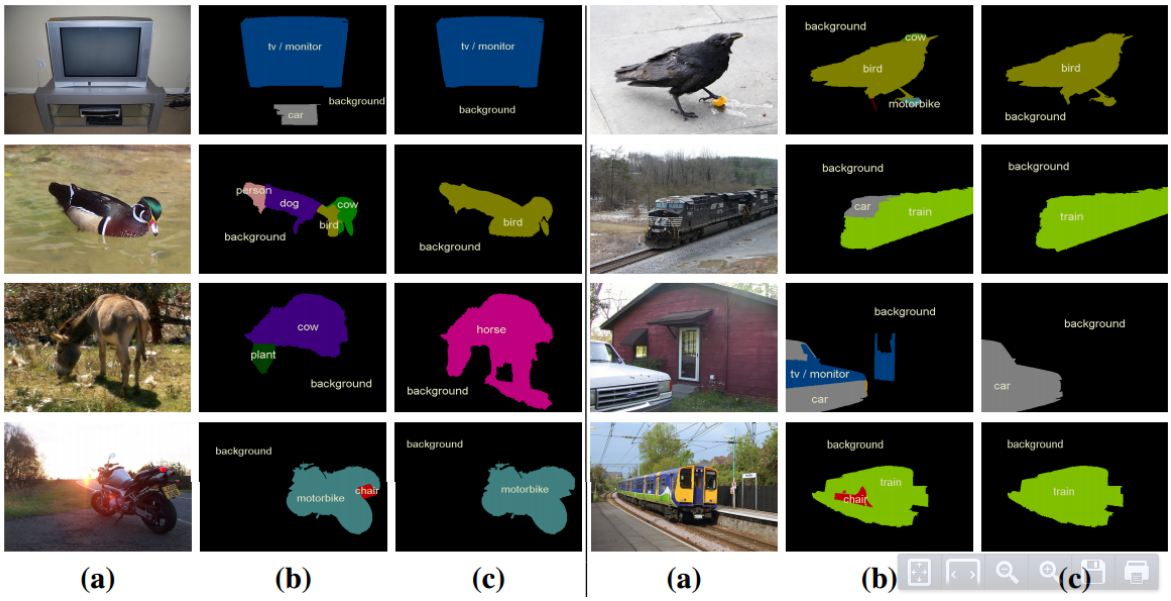


Figure 2.12 *(a) Some image from* PASCAL *dataset, (b) labelling based on pixel based random field model, (c) A labelling of the similar model using co-occurrence statistics and hierarchical random field models.*

We have also used state-of-the art semantic segmentation implementation available in our work. It is known as "Automated Labelling Environment (ALE)" [5, 2]. They improvise on the previous methods for object class segmentation which are mostly formulated as a labelling problem over a single choice of group of quantization of an image space – pixels, segments or group of segments. This quantization has some pros and cons, and the existence of optimal quantisation level suitable for all object categories is highly unlikely. So, they proposed a hierarchical random field model, that allows integration of features at different levels of the quantization hierarchy. They perform inference using graph cut based move making algorithms. Apart from these, they have also included higher order term (or potential) which models the co-occurrence statistics of objects in image (a measure of which classes are likely to occur in the same image altogether) while the MRFs and CRFs are based on local interactions between variables only. Object co-occurrence statistics capture the knowledge of scene semantics that humans often take for granted; for example that the cows and sheep are not kept together and are less likely to appear in the same image; or that motorbikes are unlikely to occur near televisions. The use of such costs can help prevent some of the most glaring failures in object class segmentation, such as the labelling of a cow as half cow and half sheep. Figure 2.12 illustrates segmentation results on some of the images and shows how co-occurrence statistics and hierarchical random field model can improve the performance.

2.5 Datasets and Evaluation Protocol

In this section, we discuss the standard datasets and evaluation metrics used in our work. Apart from these datasets, we have also evaluated our methods on our own datasets which have been discussed in the subsequent chapters (Chapter 3 and Chapter 4).

2.5.1 Datasets

We have used the following two datasets in our work -

- PASCAL VOC 2007 [46]: The PASCAL Visual Object Classes (VOC) challenge is a benchmark in visual object category recognition and detection that aims to provide the vision and machine learning communities with a standard dataset of images, annotations and their standard evaluation procedures. We have used the 2007 version of the PASCAL VOC challenge. This dataset contains images downloaded from Flickr which are taken in natural settings. It contains a total of 9963 annotated images spanning across 20 different categories. The data is divided into two splits: (i) 5011 images for training and validation, and (ii) 4952 images for testing. Training data is used for training the model and early testing of an algorithm whereas testing data is used for obtaining benchmark on the dataset. Some of the examples from PASCAL-VOC 2007 can be seen in Figure 2.10.

- CALTECH-256 [47]: It is a challenging set of 256 object categories containing a total of 30607 images. It has been created by downloading examples from Google Images followed by manual screening. The minimum number of images in each category is atleast 80. The dataset is extremely popular among vision communities for performing the classification task.

2.5.2 Evaluation Protocol

We have used the mean Accuracy Precision (mAP) as evaluation measure to measure the performance of our method. mAP evaluates the ranked list in an information retrieval task. We first define the precision and recall, precision is the fraction of retrieved instances that are relevant while recall is the fraction of relevant instances that are retrieved. Mathematically, they are defined as follows -

$$\text{Precision} = \frac{\text{True positives}}{\text{True positives} + \text{False positives}}$$

$$\text{Recall} = \frac{\text{True positives}}{\text{True positives} + \text{False negatives}}$$

Now, AP is defined as the area under precision recall curve. A classifier has high AP, if it retrieves the "object of interest" near the top of ranked list. For multi-class classification problem, we measure the performance by evaluating mean AP, i.e., mean over APs of all the individual classifiers.

Chapter 3

Decomposing Bag of Words Histograms

3.1 Introduction

There has been significant success in addressing the visual categorization problem in recent years. This can be attributed to advancements in feature descriptors (e.g. SIFT [48], HOG [49]), representations (e.g. Bag of Words (BOW) [11]), and classifiers (e.g. fast, scalable support vector machines (SVMs) [50, 39]). Often, success is measured in terms of increase in quantitative performance achieved on popular datasets such as Caltech [47] and PASCAL VOC [51]. In this context, the pipeline of BOW representation computed from dense SIFT (DSIFT) descriptors, followed by an SVM classifier has emerged as one of the most successful, as well as popular solutions for a wide spectrum of object and scene categories such as automobiles, rigid man-made objects, natural scenes [7, 52].

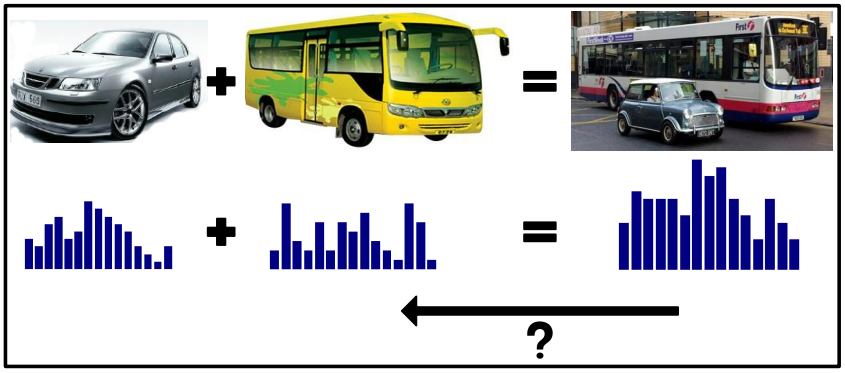


Figure 3.1 Decomposing Bag of Words Histograms. We are interested in obtaining the constituent histograms from a composite histogram.

An SVM classifier is often trained to recognize only a single class category. When multiple objects (or uncorrelated noise) are present in an image, the performance deteriorates. To better understand this issue let us consider a split of the PASCAL VOC 2007 test data into images containing a single class category (PASCAL-S) and multiple class categories (PASCAL-M). In this setting, the average precision (AP) of the BOW-trained SVM classifier for the category "cat" is 0.589 on PASCAL-S, while only 0.189

on PASCAL-M. Also, it has been observed that BOW histograms of single isolated objects are relatively easy to classify. For example, accuracy as high as 77.78% is reported on Caltech 101 dataset [7], while more complex images, which contain multiple objects and natural clutter are harder to work with (e.g. 62.8% is still the best score on PASCAL VOC 2007 [53]). An important reason for this deterioration in performance is the fact that a classifier trained on single objects often fails to recognize the object when the global image representation (BOW) is "corrupted" by additional objects and clutter present in the image. A question of interest to us now is the following. *Is it possible to filter out the clutter and classify only the signal?*

In this work, our aim is to decompose a global BOW histogram into multiple histograms corresponding to different categories present in the image, as shown in Figure 3.1. We assume some prior knowledge about the possible categories present in the image, a superset of the class categories, for instance, and that each of these categories is learnt with an appropriate SVM classifier. We solve the problem by partitioning the image into regular cells and assigning weights that correspond to each of the categories. The weights are computed using a linear optimization scheme, while maintaining their spatial continuity. Thus, the histogram of each of the categories in the image can be computed using a weighted sum of the cell histograms. Our method is specially designed for feature representations, which are additive for an image, i.e. the feature representation of the image can be computed as the sum of the features corresponding to its associated regions.

Histogram decomposition has many applications, and can be used in multiple settings to boost the classification performance as we show in the experiments section — both when single and multiple categories are present in an image. The decomposition can also be used for separating object and background histograms in an image. We evaluate our method on images from various sources: Caltech-256, Flickr and PASCAL VOC 2007.

Related work. We note that existing approaches for object detection and semantic segmentation can be adapted to solve the histogram decomposition problem. This involves two steps: (i) Performing object detection or segmentation; and (ii) Computing the individual histograms for the classes using the bounding boxes or the segmentation masks obtained.

In the case of object detection, a classifier is run over an image at multiple scales and locations. Naturally, this process is computationally expensive. Many approaches have been proposed to overcome this by restricting the number of potential windows [54], segmenting the image [55], searching only the salient regions [56], sharing features across categories [57], speeding up the individual classifiers [58]. However, the more successful methods [4] still depend on an exhaustive search in the image space. On the other hand, segmenting an image, unsupervised [17, 18] as well as category-based [5, 19, 59], is a more complex problem, where the task is to obtain a (super)pixel-level labelling. In essence, using detection or segmentation approaches for solving the histogram decomposition problem would be an overkill. In this work, we present a simpler and computationally efficient alternative for this problem.

The work of [60] uses pLSA statistical modelling to discover objects. They represent an image as a mixture of topics, and compute a histogram from a mixture of histograms corresponding to each topic.

Verbeek et al. [61] build on this by introducing a spatial coherency of labels for region classification. The recent works of [1, 62] are more closely related to our work. Russakovsky et al. [1] use the intuition that performing localization and classification simultaneously can be beneficial. They first infer the location of an object of interest and then pool low-level features separately in the foreground and background regions to form an image-level representation. This can be viewed as decomposing the image into object and background using an object detection method. Although this is an interesting approach, it suffers from high computational costs both for training and testing. Nevertheless, we compare the decomposition obtained by their work with ours in Section 3.4. It must be noted that our goal is not to achieve an exact localization of objects, unlike that in [1]. The work of Sharma et al. [62] on spatial saliency also partitions the image into regular cells and assigns weights to them. As claimed in their paper, it is more suitable for fine-grained and scene classification, and less so for classification of objects – a task we consider in this work. Furthermore, it does not consider the spatial continuity of weights while assigning them to cells as we do. The significance of this continuity term for classification of objects is further discussed in Section 3.4.

The remainder of the chapter is organized as follows. We formulate our task as an optimization problem in Section 3.2. We also contrast our approach with MRF-based methods, and discuss the latter's limitation in Section 3.2. Inspired by the success of fast and scalable classifiers [50, 39, 63], we discuss our work using linear SVM as an example. We show how the formulation can be generalized to spatially-constrained decomposition in Section 3.3. Section 3.4 presents an exhaustive evaluation of our method. We discuss the application of proposed formulation in Section 3.5 and then make concluding remarks in Section 3.6.

3.2 Decomposing Histograms

Consider an image with k object classes of interest. Let $\mathbf{h}$ denote the global unnormalized[1] histogram of the image. The set of linear classifiers (e.g. SVM) trained on the k classes are represented by $\mathbf{w}_1, \ldots, \mathbf{w}_k$. Our objective is then to decompose the histogram $\mathbf{h}$ into k constituent histograms represented by $\mathbf{x}_1, \ldots, \mathbf{x}_k$, corresponding to each of the classes.

To solve the histogram decomposition problem, we begin by partitioning the image into $M \times N$ regular rectangular cells. Let $\mathbf{h}_{ij}$ denote the histogram computed independently for each cell. We introduce a binary variable $b_{ij}^p \in \{0, 1\}$ for each cell to denote whether it is part of an object from the pth category or not. With this, our decomposition problem is formulated as:

$$\max \sum_{p=1}^{k} \mathbf{w}_p^T \mathbf{x}_p, \tag{3.1}$$

[1] Such histograms have been used successfully in the past [54].

such that $\mathbf{x}_p = \sum_{ij} b_{ij}^p \mathbf{h}_{ij}$ and $\sum_p b_{ij}^p = 1$. Note that $\mathbf{w}_p^T \mathbf{h}_{ij}$ can be compared for different classes since the classifiers ($\mathbf{w}_p$'s) are trained on normalized histograms. This problem can be solved in closed form by taking b_{ij}^p to be 1 for the p that maximizes $\mathbf{w}_p^T b_{ij}^p$ and 0 for all other p's.

The optimal solution to the problem (3.1), however, is not always semantically meaningful. For instance, cells from sky or road may be labelled as part of other object categories such as bus or car. Furthermore, object cells of a specific category may be scattered and spatially disconnected. We also need a mechanism to incorporate some of the prior knowledge one may have in many practical situations. Examples of such constraints include: (i) a specific shape or aspect ratio of an object, (ii) spatial continuity and penalizing configurations that result in a set of scattered cells for an object category, (iii) a prior on the scale of the object of interest, or even (iv) favouring objects in certain parts of the image, say in the center. To make the formulation more realistic, we relax the assumption in (3.1) that all the cells are to be assigned to one of the k objects of interest. This implicitly accounts for the fact that not all parts of the image may be recognized with existing classifiers $\mathbf{w}_p$. Thus, the problem is to:

$$\max_{b,\lambda} \sum_{p=1}^{k} \mathbf{w}_p^T \mathbf{x}_p - \gamma \sum_{(i,j)\in\mathcal{N}} |\lambda_{ij}|, \tag{3.2}$$

subject to the constraints:

$$A : \mathbf{x}_p = \sum_{i,j} b_{ij}^p \mathbf{h}_{ij}, \qquad B : \sum_p b_{ij}^p \leq 1,$$

$$C : \sum_p \sum_{ij} b_{ij}^p \geq \mathbf{P}, \qquad D : b_{ij}^p \in \{0, 1\},$$

$$E : 8 \times b_{ij}^p - (b_{i-1j-1}^p + b_{i-1j}^p + b_{i-1j+1}^p +$$
$$b_{ij+1}^p + b_{i+1j+1}^p + b_{i+1j}^p + b_{i+1j-1}^p + b_{ij-1}^p) = \lambda_{ij},$$

where γ is a regularization parameter. The constraint A defines an object class category as a weighted sum of histograms from multiple cells, similar to (3.1). The constraint B allows some of the cells to remain unlabelled. We introduce constraint E in a neighbourhood system $\mathcal{N}$, which enforce neighbouring cells to take a similar label. Figure 3.2 shows the neighbourhood systems which are considered for spatial continuity of the labels. In other words, E define penalties λ_{ij} and provide object smoothness constraints. The parameter γ controls the emphasis on the spatial smoothness of the object. All neighbourhood relations shown in Figure 3.2 can be represented by the constraint E. The SVM classifiers used in this formulation do not include a bias term, but it can be easily incorporated by augmenting every histogram $\mathbf{x}_p$ with 1. Empirically, we found the effect of introducing bias negligible, as we are using unnormalized histograms $\mathbf{h}_{ij}$. The no-bias formulation favours discarding assignments with negative dot product ($\mathbf{w}_p^T \mathbf{x}_p$). However, the constraint C helps overcome this issue by enforcing a minimum of label assignments.

A familiar analogy of the objective function (3.2) with constraints A, B, D, E, are energy functions that can be modelled as a Markov random field (MRF) with unary and pairwise potentials, and solved

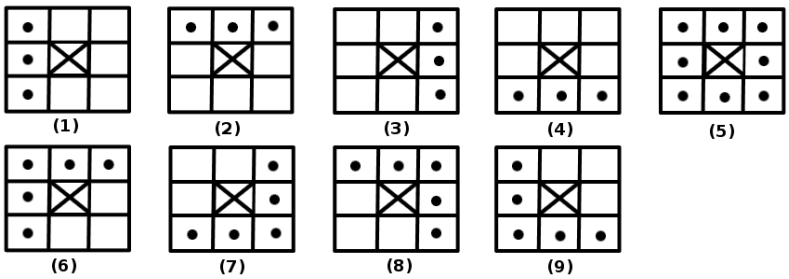


Figure 3.2 Neighbourhood systems considered in the LP formulation.

efficiently [20]. In such a setting, the term $\mathbf{w}_p^T \mathbf{x}_p$ in (3.2) corresponds to the unary potential and the term $|\lambda_{ij}|$ represents the pairwise potential. However, as discussed in Section 3.2.1, the constraint C, which introduces a lower bound P on the number of cells assigned to any of the k classes, cannot be easily added into this framework. This constraint avoids a trivial solution for (3.2), and we will study the importance of this constraint in Section 3.4. Note that the residual histogram, i.e. $\mathbf{r} = \mathbf{h} - \sum_{i=1}^{k} \mathbf{x}_i$, need not be empty by design. If there is a need to use context for enhancing the representation, $\mathbf{r}$ can be added to a specific category. We relax the constraint D as $b_{ij}^p \in [0, 1]$ and solve the resulting linear program (LP) relaxation. The spatial extents of the individual constituent histograms can be obtained by rounding b_{ij}^p's to their nearest integers.

The histograms of different categories in an image can be obtained directly using the solution of the LP relaxation, i.e. taking the weighted sum of the cell histograms (LP-relax) or by first rounding-off the solution to the nearest integer and then adding the corresponding cell histograms (LP-round). We analyze the performance on these two solutions in the experiments section, and observe that LP-relax performs better than LP-round, as the former makes a soft-assignment of cells to categories.

3.2.1 An MRF-based solution

As noted earlier, the decomposition problem can be modelled as an MRF energy minimization problem. Here, each cell in the $M \times N$ grid is represented as a node in a graph. Each node takes a label from the set $\mathcal{L} = \{1, \ldots, k\}$. This is equivalent to introducing binary variables b_{ij}^p for all the classes, and constraining them with B. Popular techniques such as sequential tree-reweighted message passing (TRW-S) [21], belief propagation [64], and alpha expansion [22] can be used to solve this formulation, which is equivalent to problem (3.2) with constraints A, B, D and E. We observed (see Section 3.4) that most of the cells are assigned to the background in this solution. This is not surprising as the classifier used in the formulation is learnt from images with large intra-class variations, and is not expected to fire positively when only some part of the object is considered in a cell. We overcome this issue by introducing the constraint C. We study the effect of its introduction in Section 3.4 by solving the LP and MRF problems in the absence of C. Global constraints such as C cannot be incorporated easily into a standard pairwise MRF [23, 65]. Although MRFs with higher order potentials can be adapted to do so,

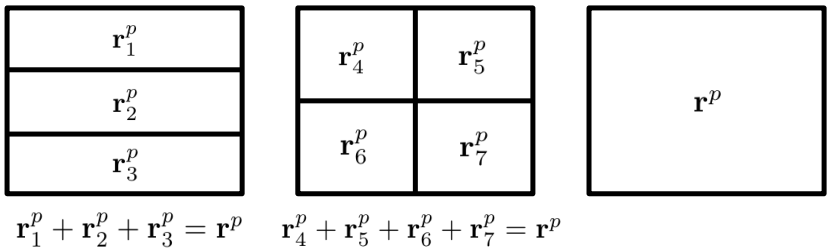


Figure 3.3 Incorporation of spatial pyramid histograms into our LP formulation. We show the sub-regions considered while computing the spatial histogram for an object category p. In addition to the constraints mentioned here, we add constraints that $\mathbf{r}_1^p$, $\mathbf{r}_2^p$ and $\mathbf{r}_3^p$ should contain equal number of visual words, as they occupy the same area in the image, and a similar constraint for $\mathbf{r}_4^p, \mathbf{r}_5^p, \mathbf{r}_6^p$ and $\mathbf{r}_7^p$.

they often lead to approximate and computationally expensive solutions. The work of [66] introduces a count based global prior constraint, similar to C, into an MRF formulation, but resorts to LP-based techniques to solve the problem. Further, it focusses on obtaining integral solutions for segmenting an image, whereas the solution of the LP relaxation suffices for our histogram decomposition task.

3.3 Spatially-constrained Decomposition

Our formulation discussed thus far encodes limited spatial information of the objects or the regions into the histograms. Often, incorporating spatial information, such as in [67], and concatenating histograms from multiple sub-regions, has shown to improve the classification performance in many cases. In this section we present a more general framework to address this issue.

We extend our framework introduced in (3.2) to incorporate spatial information into histograms. We introduce weak geometry constraints into the histograms, without affecting the linearity of the problem, inspired by the work on spatial histograms [67]. The spatial region for an object category *p* is divided into 3×1, 2×2 and 1×1 grids giving rise to a total of eight sub-regions, as shown in Figure 3.3, similar to [7]. The final representation of an object is obtained by concatenating histograms of the eight sub-regions. Let $\mathbf{r}^p$ be the histogram of class p and $\mathbf{r}_1^p, \ldots, \mathbf{r}_7^p$ denote the corresponding sub-region histograms. In this formulation involving Spatial Pyramid Matching (SPM), we simultaneously solve for b_{ij}^p and sub-region histograms $\mathbf{r}_1^p, \ldots, \mathbf{r}_7^p$. We replace the constraint A in problem (3.2) with the following set of constraints:

$$A1 : \mathbf{x}_p = [\mathbf{r}_1^p \ldots \mathbf{r}_4^p \ldots \mathbf{r}_7^p \, \mathbf{r}^p], \quad A2 : \sum_{ij} b_{ij}^p \mathbf{h}_{ij} = \mathbf{r}^p,$$

$$A3 : \sum_{i=1}^{D} \mathbf{r}_{1i}^p = \ldots = \sum_{i=1}^{D} \mathbf{r}_{3i}^p, \quad A4 : \sum_{k=1}^{3} \mathbf{r}_k^p = \mathbf{r}^p,$$

$$A5 : \sum_{i=1}^{D} \mathbf{r}_{4i}^p = \ldots = \sum_{i=1}^{D} \mathbf{r}_{7i}^p, \quad A6 : \sum_{k=4}^{7} \mathbf{r}_k^p = \mathbf{r}^p,$$

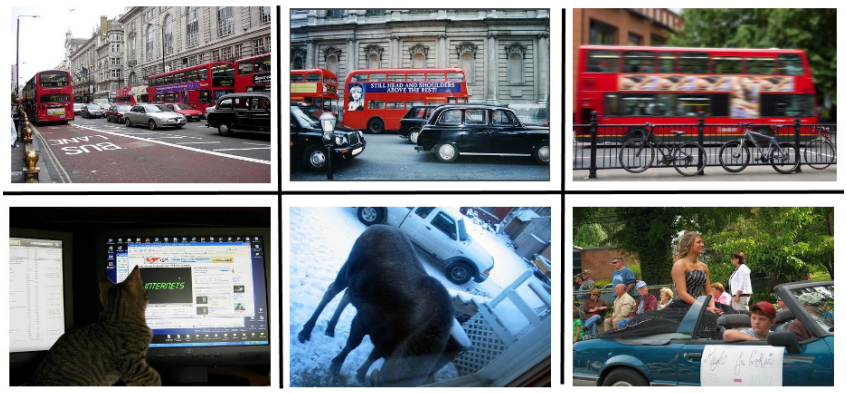


Figure 3.4 Example images from Flickr (first row) and PASCAL VOC 2007 (second row), containing multiple objects.

where D is the dimension of the histogram, $\mathbf{r}_j^p$ is the histogram of the jth sub-region, and $\mathbf{r}_{ji}^p$ denotes the ith visual word count in the histogram $\mathbf{r}_j^p$. Constraint $A1$ defines the histogram of object p as the concatenation of eight sub-region histograms shown in Figure 3.3. The constraint $A2$ represents the histogram in terms of its cell histograms. Constraints $A3$ and $A5$ imply that $\mathbf{r}_1^p$, $\mathbf{r}_2^p$, $\mathbf{r}_3^p$ and $\mathbf{r}_4^p$, $\mathbf{r}_5^p$ $\mathbf{r}_6^p$, $\mathbf{r}_7^p$ contain equal number of visual words, as they occupy the same area in the image. Constraints $A4$ and $A6$ correspond to the conditions mentioned in Figure 3.3, that the sum of the sub-region histograms is equal to the histogram of class p. Note that only weak spatial constraints for sub-region histograms have been considered in the above formulation, so as to keep our problem linear. It does not impose constraints on the positions of sub-regions in the image, and hence, does not compute exact spatial histograms. However, it encodes some spatial information, which results in a better decomposition of the global histogram.

We use SVMs trained on spatial histograms of tight bounding boxes around the object as classifiers ($\mathbf{w}_p$), computed by dividing object bounding box into 3×1, 2×2 and 1×1 grids, as shown in the illustration in Figure 3.3. Hence, when we maximize $\mathbf{w}_p^T\mathbf{x}_p$ in the objective function, $\mathbf{r}_1^p, \ldots, \mathbf{r}_7^p$ approximately correspond to the histograms of sub-regions assumed.

3.4 Experiments and Results

We demonstrate the performance of our method for decomposing the global BoW histogram of an image into its constituent histograms in a variety of settings. In addition to merely measuring the accuracy of the decomposition, we also show its utility for the task of image classification. We use PASCAL VOC 2007, Flickr multiple object, and Caltech-256 based datasets in our experiment.

Composed Caltech dataset The Composed Caltech dataset, we refer to as CALTECH, is a synthetic dataset generated from Caltech-256 images [47]. It consists of images formed by "pasting" scaled, translated, and rotated versions of the original Caltech-256 images onto other images from the dataset.

k	CALTECH-2			CALTECH-3			CALTECH-4		
	LP$_{wc}$	TRW-S	LP$_c$	LP$_{wc}$	TRW-S	LP$_c$	LP$_{wc}$	TRW-S	LP$_c$
DIR	0.19			0.23			0.22		
CV	0.23			0.25			0.26		
2	1.00	1.00	**1.00**	NA	NA	NA	NA	NA	NA
3	0.78	0.80	**0.97**	0.92	0.93	**0.98**	NA	NA	NA
4	0.61	0.63	**0.89**	0.79	0.80	**0.94**	0.84	0.86	**0.97**
5	0.51	0.54	**0.76**	0.69	0.72	**0.81**	0.82	0.82	**0.92**
10	0.29	0.25	**0.41**	0.35	0.37	**0.45**	0.36	0.37	**0.39**
20	0.15	0.16	**0.26**	0.17	0.17	**0.29**	0.19	0.21	**0.31**

Table 3.1 Comparison of LP (without C), TRW-S & LP (with C). Mean classification AP for different k's on CALTECH using TRW-S and our LP-based solution (with and without the constraint C). CV and BOW are two baseline methods. BOW uses the entire composite image histogram and CV uses histograms obtained via cell-based voting. Note that the formulation is not solved (NA) for 2 classes ($k = 2$) when 3 or 4 objects are present in the image.

This provides us with a controlled setting to regulate the complexity of the image histograms. Note that the visual content of some of these images may not be appealing, but we are only interested in the histograms. We divide the images from CALTECH into three multi-category object datasets – CALTECH-2, CALTECH-3 and CALTECH-4 – each consisting of 10,000 images. They have been created by concatenating two, three, and four objects, randomly selected from 20 predefined candidate categories, respectively. The scale of each object in the composite image is measured as the percentage of the composite histogram it contributes to. This varies from 10% to 90%. The purpose of introducing this dataset was to study the sensitivity and robustness of our formulation especially when the object size in the image, and k, the number of categories considered in the objective function (3.2), vary.

PASCAL VOC and Flickr multiple object dataset We also use natural images from PASCAL VOC 2007 [46] and a set of images downloaded from Flickr. The Flickr dataset is composed of images downloaded with "*bus* & *car*" and "*bus* & *bicycle*" text queries, which were then filtered manually. We refer to the Flickr multiple object datasets as Flickr-M1 and Flickr-M2. Flickr-M1 has 196 positive images containing both *bus* & *car*, and Flickr-M2 has 209 positive images with both *bus* & *bicycle* in them. For both these datasets, we harvest negative training examples from the PASCAL VOC 2007 images containing neither of the two object categories. A few samples from these datasets are shown in Figure 3.4.

Experimental setting In all our experiments an image is divided into 16×16 cells, and a vocabulary of size 4K is used, unless otherwise stated. DSIFT features are extracted at a step size of 5 pixels and a hard quantization is used to assign them to visual words. We trained SVM classifiers using liblinear. For the CALTECH dataset experiments, the classifiers are trained using 25 samples from each category that are not in the CALTECH dataset. We set P (from constraint C in (3.2)) to 50% of the total cells in an

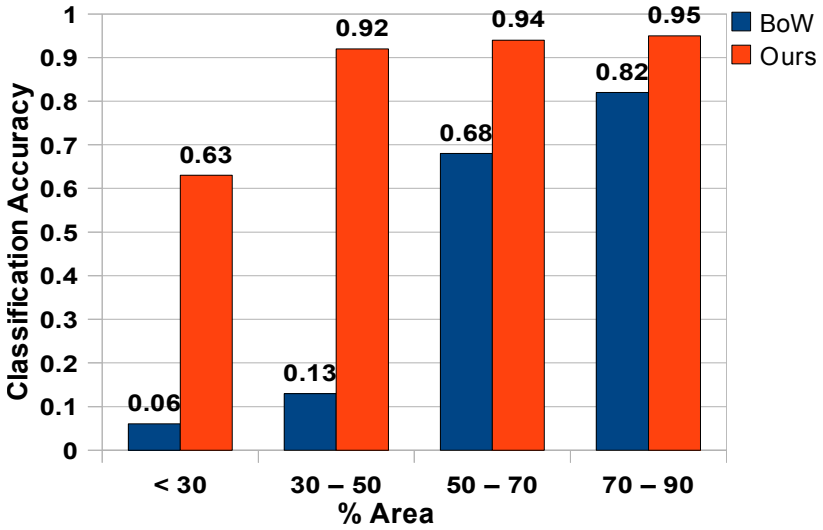


Figure 3.5 Comparison of the classification performance and the scale of objects in the CALTECH dataset. We use k=5 for this experiment. We observe that our method outperforms the naive BOW based classifiers at all scales.

image. The parameter γ is set to 1 for CALTECH and 0.7 for Flickr and PASCAL VOC 2007 datasets by cross validation. We used MOSEK[2] for solving the linear programs.

3.4.1 CALTECH histogram decomposition

Comparison of LP and TRW-S We begin by evaluating the performance of our histogram decomposition method on the CALTECH dataset. Recall that the problem (3.2), albeit without the inclusion of constraint C, can be solved: (i) directly as an LP formulation; or (ii) using an MRF-based solution (Section 3.2.1) such as sequential tree-reweighted message passing (TRW-S) [21]. We follow both these approaches on BOW histograms of the CALTECH dataset. Since we use DSIFT-BOW histograms, the orientation and location of the individual objects do not significantly affect the composite histogram. We evaluate the performance of our decomposition by obtaining the mean AP over all the classes, when the constituent (category-level) histograms are passed to the respective SVM classifiers. Table 3.1 shows the MAP obtained using LP (LP_{wc}) and TRW-S, without constraint C. It can be observed that the performance of these two solution schemes is comparable, with LP_{wc} performing better in some cases, and marginally inferior in some other cases.

Importance of the constraint C The advantage of using an LP-based solution is evident with the inclusion of the constraint C. Table 3.1 also shows the MAP over all the 20 Caltech classes when the problem is solved using the LP formulation with the constraint C, i.e. (3.2), referred to as LP_c in the table. It shows a significant boost in performance over TRW-S and LP_{wc} – an average improvement of 30.35%. Note that including this global constraint in the TRW-S solver is not trivial, as discussed in

[2]http://www.mosek.com

Section 3.2.1. We also compare the LP method with two other baseline approaches: BoW and cell-based voting (CV). We use the entire composite histogram of an image for the BoW method, while for the CV approach we assign each cell independently to atmost one object, and build the object histogram from histograms of cells that belong to the object. Our decomposition based methods outperform the baselines, as shown in Table 3.1. For the remainder of the experiments, we used the LP formulation with the constraint C.

Figure 3.5 analyzes the classification accuracy when the scale of the object (percentage of the area occupied) in the image varies. As expected, the problem is relatively hard when the object is very small. When the scale is 30% or more, decomposition helps classify the image more than 92.4% of the time. Even when the scale of the object is small (10-30%), our method correctly discriminates the object histograms more than 63% of the time. Note that the improvement of our method over BoW is more pronounced when the scale is small. This is significant since, it is these small objects that we often find hard to recognize in composite images, in the presence of clutter.

3.4.2 Multiple object classification

Figure 3.6 Histogram decomposition on Flickr-M2 dataset. LP is solved for two classes *bus* and *bicycle* simultaneously. The images and the corresponding weights obtained for their cells using the LP solution are shown. The cells shown in red are weights of *bus*, while those in green are of *bicycle*. Higher intensity of the colour represents a value closer to 1. (**Best viewed in colour.**)

In this experiment we investigate how the presence of one object in an image can negatively affect the classification of others. We consider the AP obtained when the entire histogram of an image is given to an SVM classifier (BoW) as the baseline. Using the LP formulation proposed in Section 3.2, we split the image histogram into histograms of constituent objects, and the background/context. Figure 3.6 shows the decomposition of the global histograms in a few examples containing bus and $bicycle$ categories. One approach to evaluate this decomposition is by using the constituent histograms directly in an object classifier. However, ignoring the background/context histogram would be an unwise move, given

Figure 3.7 An illustration of the decomposition results on sample images from PASCAL VOC 2007 dataset. We show an image and the corresponding weights of its cells obtained from our LP solution. Higher intensity of the colour green represents a value closer to 1. (**Best viewed in colour.**)

previous work [68] suggesting that background/context can provide strong cues useful for classifying objects. Thus, we use the object-background feature representation of [1], where a histogram is represented by a concatenation of object and background histograms. The background histogram is obtained by subtracting histograms of objects from the global image histogram.

Method	Flickr-M1		Flickr-M2		Pascal-M
	bus	*car*	*bus*	*bic*	
BoW	0.522	0.516	0.302	0.108	0.287
LP (round)	0.561	0.574	0.373	0.235	0.312
LP (relax)	0.582	0.590	0.397	0.246	0.331
LP-SPM	**0.598**	**0.612**	**0.408**	**0.278**	**0.348**

Table 3.2 Classification AP on Flickr-M1, Flickr-M2, and PASCAL-M datasets. In LP-relax, we use soft assignment of cells whereas in LP-round, hard assignment of cells is used.

We compute AP on the Flickr dataset with classifiers trained on features extracted from object bounding boxes, concatenated with the features extracted from the remainder of the image. The classifiers used in our LP formulation are trained on features extracted from the training+validation sets of PASCAL VOC 2007. LP is solved for all images in the dataset to get the constituent histograms (for Flickr-M1, it is solved using classifiers for *bus* and *car*, and for Flickr-M2, using classifiers for *bus* and *bicycle*). Table 3.2 compares the classification AP obtained using BoW, LP and LP-SPM (Section 3.3). Using LP, we see an improvement of 12.9% on Flickr-M1, and 56.8% on Flickr-M2. The relaxed LP solution performs better than the rounded-off solution, as it does not make hard assignments for cells. Table 3.2 also compares the AP on PASCAL-M, with LP showing the best results.

Method	Pascal		Comments
	Simple	SPM+Ex.F.M.	
BoW	0.379	0.528	Baseline
TestBB	0.659	0.834	Golden baseline
DPM	0.386	0.543	Decomp. using existing methods
Sem. Seg.	0.434	0.561	
LP (round)	0.418	0.536	Decomp. using LP-based methods
LP (relax)	0.435	0.558	
LP-SPM	**0.447**	**0.567**	

Table 3.3 Mean classification AP on PASCAL VOC 2007 over all the 20 classes. AP for each of the 20 classes can be found on the project website. TestBB shows the AP when the decomposition is done using ground truth bounding boxes, DPM when using [4], Sem. Seg. is with ALE [5], and LP is the proposed formulation. See text for details.

3.4.3 Decomposition into object and background

We now discuss the decomposition results on the challenging PASCAL VOC 2007 dataset. We use the proposed formulation to decompose an image histogram into histograms of an object (category) of interest and background. Figure 3.7 shows the assignment of object *vs* background labels to the image cells on a few sample images from the dataset. This decomposition is evaluated in the context of the image classification problem. The final representation of an image for evaluating an AP of category 'X' is represented by concatenation of individually normalized (L2 norm = 1) foreground ('X') and background histograms (L2 norm = 0.5). The final representation is renormalized to have L2 norm of 1 (similar to [24]). The APs for DPM, Sem. Seg. and LP have been obtained using above normalized histogram representation of an image whereas for BoW, image is represented as a normalized histogram of an entire image. The representation is then used by an SVM classifier.

Table 3.3 shows a comparison of our LP decomposition scheme with baseline methods. The image representation is a concatenation of individually normalized object and background histograms for the methods TestBB, DPM, Sem. Seg. and LP, while a normalized histogram of the entire image is used in the case of BoW. The representation is then used by the SVM classifier. The image classification AP is computed for all the methods mentioned in Table 3.3, with (SPM + Ex.F.M.) and without (*Simple*) the use of spatial pyramids and explicit feature maps [7, 39]. Spatial features are computed as in [7], both for the object and the background regions. In the table, BoW refers to a direct baseline when the entire image histogram is used. TestBB is the "golden" baseline, where the object histograms are extracted from ground truth bounding boxes, and used in combination with histograms from the remainder of the image. This provides us with an upper bound on this dataset for an object-background representation model. We also compared our approach with methods using regions from sliding window detectors [4] (DPM) and semantic segmentation [5] (Sem. Seg.). For DPM, we use the publicly available detectors to get the bounding boxes, and for semantic segmentation, we use the automated labelling environment

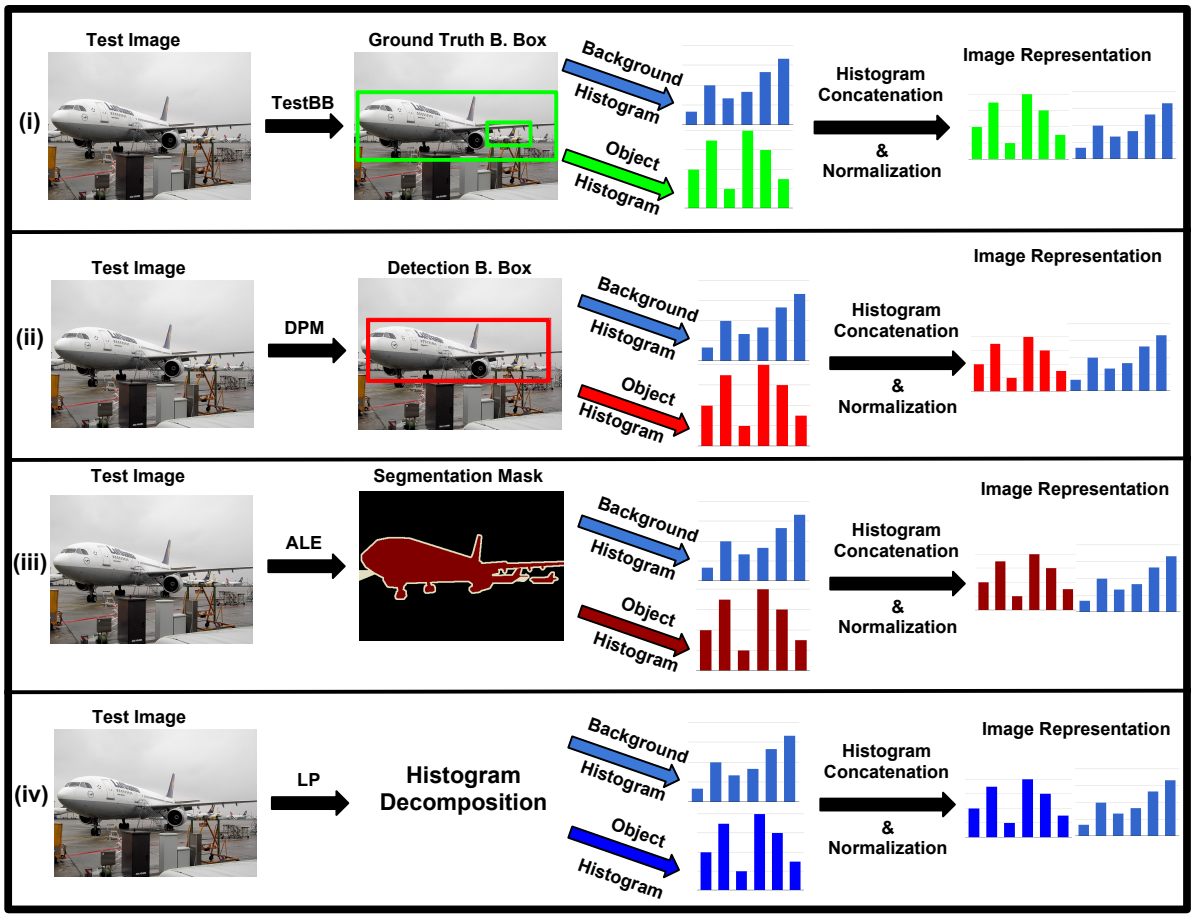


Figure 3.8 Image representation for the classification task with various methods, namely, (i) TestBB, (ii) DPM [4], (iii) Sem. Seg. [5], and (iv) LP, discussed in this chapter. **(Best viewed in colour.)**

(ALE) framework to find the segmentation mask [5].[3] The features from object regions are concatenated with features from the remainder of the image, as done for other methods, for both these approaches. We observe that our LP decomposition scheme outperforms DPM and and is comparable to Sem. Seg., with a much lower computation cost (less than 1s), as shown in Table 3.4. Although the running time of DPM and Sem. Seg. approach can be improved significantly by applying them at a coarse resolution, this results in an inferior AP. For example, down sampling PASCAL VOC 2007 images by a fourth reduces the performance of (re-trained) ALE from 0.567 in Table 3.3 to 0.288.

3.4.4 Decomposition in a weakly supervised setting

In the histogram decomposition formulation (3.2), we require a linear SVM classifier, which can discriminate the categories present in the image. We learn them using bounding boxes annotated in the

[3]We trained ALE on ($\sim$ 2223) images from PASCAL VOC 2011 and not on the ($\sim$ 422) images from PASCAL VOC 2007.

Method	Testing time (per image)
DPM	5-6 seconds
Sem. Seg.	4-5 minutes
LP	0.5-1 seconds
LP-SPM	3-4 seconds

Table 3.4 A comparison of testing times of different methods. The reported times are with publicly available implementations on a standard 3.3GHz, 4GB machine. They also include the time for feature extraction.

training data. We now extend our approach to learn classifiers for a more general weakly supervised setting, where bounding box annotation is not available for most of the images. Recent approaches, such as [69, 1], can be adapted to learn classifiers in a weakly supervised setting, but being based on object localization, they are computationally expensive.

We use an iterative procedure to learn the classifier in this setting. Given an initial classifier for an object, we decompose histograms of training images with our LP formulation. Next, we compute the new histograms of objects as a weighted sum of the cell histograms and re-train the classifiers with them. We repeat this procedure 10 times. In order to initialize, we use 10 ground truth bounding box annotations per object and train an initial classifier. We compare this decomposition scheme for the image classification task on PASCAL VOC 2007 to object-centric spatial pooling [1]. We use DSIFT at a step size of 3 and sum pooling based LLC encoding (to ensure additivity of features) while solving LP. Following [1], we use max pooling based LLC encoding for the object-background feature representation when classifying images. We achieve a mean AP of 0.560 with LP-relax and 0.571 with LP-SPM, using a vocabulary of size 4k compared to 0.572 in [1]. In the same setting, when we increase the vocabulary size to 25k, we obtain an AP of 0.589 for LP and 0.594 for LP-SPM. The training time on a single machine for the method in [1] is 3 days, compared to under 7 hours for our LP method. It must be noted that the AP scores mentioned in Table 3.3 are not directly comparable to those given in this section, as the method for pooling is based on [1], and different to those used previously. The features used and the vocabulary size are different as well. We also analyzed the importance of the spatial smoothness constraint by setting $\gamma = 0$, which results in an AP of 0.458 compared to 0.560 for LP-relax. Table 3.5 summarizes the results including comparison with Chatfield et al. [7].

3.5 Discussion

The use of our formulation for improving object localization can be a possible extension. Presently, the proposed formulation only approximately localizes the object. With the presence of better heuristics and prior information, we believe the proposed formulation can be extended to get tight bounding box around an object. In this section, we show an application of our method for object localization in CALTECH dataset. For this task, 100 random images from CALTECH-2 are selected and bounding boxes are annotated manually. We show object localization can achieve significant speed-up using proposed

Method	Vocab.	AP
OCP [1]	-	0.572
Chatfield et al. [7]	4k	0.538
	25k	0.573
LP	4k	0.560
	25k	0.589
LP-SPM	4k	0.571
	25k	**0.594**

Table 3.5 Mean classification AP on PASCAL VOC 2007 over all the 20 classes in a weakly-supervised settings. APs are also shown and compared with Russakovsky et al. [1] and Chatfield et al. [7].

Method	Ex. SW	Cell SW	LP-det
Avg. Time	2.24s	1.07s	**0.10s**

Table 3.6 Time comparison of object detection results on CALTECH for exhaustive sliding window, cell sliding window and LP-det (for k=5). LP-det shows significant speed-up for objective detection without compromising on the AP. Note that we achieve $\gamma_{\mathbf{ex}}$=0.92 and $\gamma_{\mathbf{cell}}$=0.96.

formulation without compromising on the accuracy much. We have the labels associated with each of the cells from LP (round) and now, our task is to get bounding boxes from the cell labels.

Using breadth first search, we find all the connected components present in the image. Two neighbouring cells are said to be connected if they share the same label. For each connected component, a bounding box is drawn which covers all its cells. We refer to this method as LP-det. We have compared localization performance on CALTECH dataset with the standard sliding window methods based on the following two measures :

$$\gamma_{\mathbf{ex}} = \frac{\mathbf{AP\ obtained\ using\ Exhaustive\ Sliding\ Window}}{\mathbf{AP\ obtained\ using\ LP\text{-}det}}$$

$$\gamma_{\mathbf{cell}} = \frac{\mathbf{AP\ obtained\ using\ Cell\ Sliding\ Window}}{\mathbf{AP\ obtained\ using\ LP\text{-}det}}$$

A detection is said to be positive if the overlap between ground truth bounding box and the bounding box returned by LP-det is greater than 0.5. Exhaustive sliding window (Ex-SW) involves rigorous testing of sub-windows at very fine level whereas cell sliding window (Cell-SW) involves testing of sub-windows at cell level. We have achieved around 20 times speed-up over Ex-SW and 10 times speed-up over Cell-SW while achieving almost the similar performance. These results are summarized in Table 3.6.

Figure 3.9 shows the localization performance on PASCAL VOC 2007. In this case, bounding box is not drawn and the object location is taken as the region returned by LP/LP-SPM. In this case, we analyse our detection performance by measuring the percentage of images for which overlap ratio of ground

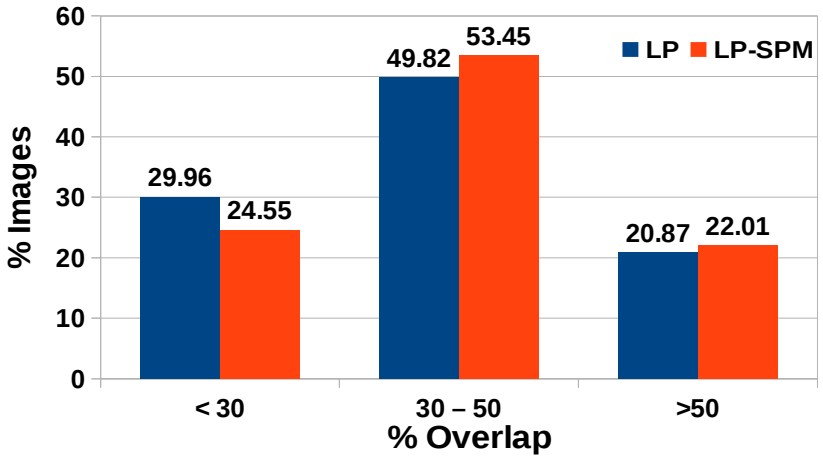


Figure 3.9 Localization results on PASCAL VOC 2007 using LP and LP-SPM. The curve is plotted between number of images v/s overlap ratio between the ground truth and the region predicted by LP/LP-SPM. More than 70% of the samples have an overlap ratio greater than 0.30.

truth bounding box and the region returned by LP/LP-SPM is in the specified range. More than 70% of the images have overlap ratio greater than 0.30 which shows proposed formulation can be used for approximate localization of objects.

3.6 Summary

We proposed an effective method to decompose a global histogram of an image into histograms of its associated objects and regions. Our approach solves the problem using an LP formulation, by taking an intermediate path between two harder problems, namely bounding box accurate object detection and pixel-accurate object segmentation. We showed that a wide variety of composite histograms can be decomposed into their constituent histograms with our LP method. We also demonstrated the application of histogram decomposition for improving the classification performance on multiple object and PASCAL VOC 2007 datasets using an object-background representation of an image. The use of our formulation for further improving object localization is a possible avenue for future research.

Chapter 4

Retrieval of Exact and Near Duplicate Document Images

4.1 Introduction

In recent years, the amount of digital content has increased exponentially because of the emergence of large document repositories. Many applications have been built in the past to retrieve related documents/research papers from such large databases given a query. A query can be a simple text query (e.g., Google Scholar) or a document image [70, 71, 72, 73]. In this work, we focus on the latter, i.e., when the query is a document or region image. The domain of document retrieval can be divided into three halves depending upon their difficulty – retrieval of exact, near and far duplicate document images. Exact duplicate is the direct cut and paste of content from multiple documents (Fig. 4.1), near duplicate document segments could arise due to various document manipulations like summarization, copying, rewriting, editing, formatting, cut-and-paste, etc. (Fig. 4.2) and far duplicate documents are those which contains semantically similar topic to a given query document.

In the past, lot of research has been done for the retrieval of semantically similar documents (far duplicate) but very few have focused on the retrieval of exact and near duplicate documents. For the retrieval of far duplicate documents, firstly words are recognized in the document (using OCR, holistic word recognition [74], etc.) and then, the document is represented as a histogram of words, meaning that the words are assumed to occur independently. Vector representation of the documents are organized in an inverted file [75] to felicitate efficient retrieval. Often, to capture the relationship between words, many approaches has been proposed to group words into topics (Latent Semantic Indexing [76], probabilistic Latent Semantic Indexing [77], Latent Dirichlet Allocation [78, 79], etc.). However, in this work, unlike histogram of words, we represent documents as histogram of visual words for retrieving their exact and near duplicates.

In this work, we also show an important application of our recognition free duplicate retrieval framework to the problem of copy detection in document images. Because of the availability of vast amount of resources in web, many new documents get created by copying existing documents. Often these could also lead to unethical copying or counterfeiting of documents. Different methods have been employed in the literature to detect such cases, with the focus on plagiarism detection. There can be two direc-

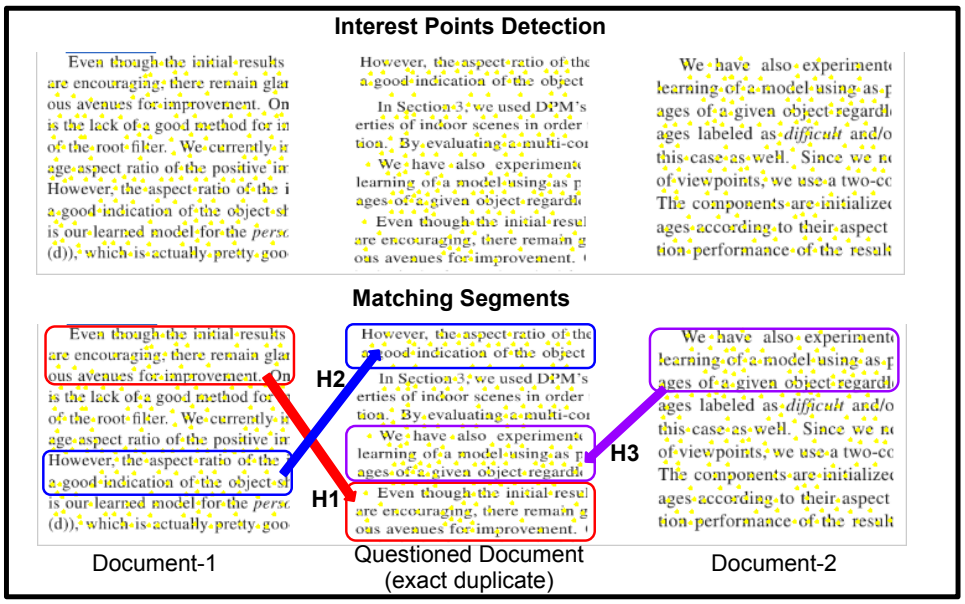


Figure 4.1 *Example of a document containing exact duplicate: We are interested in matching parts of a questioned document with the documents in database using mixture of homographies model. In the figure, a questioned document is matched with Document-1 & Document-2 using 3 homographies.*

tions to detect such cases of document forgeries – recognition based and recognition free. Text level comparison of documents is fairly advanced and there exists many software tools for this [80, 81]. They rely on the OCRs (for text creation) and some level of language processing in the text domain to find similarity between documents. However, text is not always available. In many practical situations (e.g., poor quality documents, historical documents, etc.), OCR based textual solutions are not accurate. To the best of our knowledge, this is the first attempt on recognition free detection of copy and plagiarism.

Our problem, in a way, is a retrieval problem where the query is a part of a document (say a paragraph or few lines, which has been copied) and therefore, falls in between the two categories of work, which retrieves results based on query as entire document [72, 73] or query as words (eg., Google, [82]). However, there is a significance difference for our problem. Our query may be the entire document but we are interested in retrieving multiple documents from which parts are possibly copied. Our "queries" are part of the documents, but we do not know which parts need to be used as queries.

Our first method is based on modelling the solution as finding a mixture of homographies, and designing a linear programming (LP) based solution to compute the same. Our method is independent of the imaging process and simultaneously support the camera based (perspective) imaging and traditional scanner based (orthographic) imaging. Without loss of generality (i.e. without assuming the number of source documents), we can correctly detect and match the duplicate content in a questioned document image by simultaneously comparing with large number of images in the database. Fig. 4.1 shows the matching of parts of questioned document with parts of Document-1 and Document-2 using multiple homographies. Experiments have been performed by changing the imaging process, varying

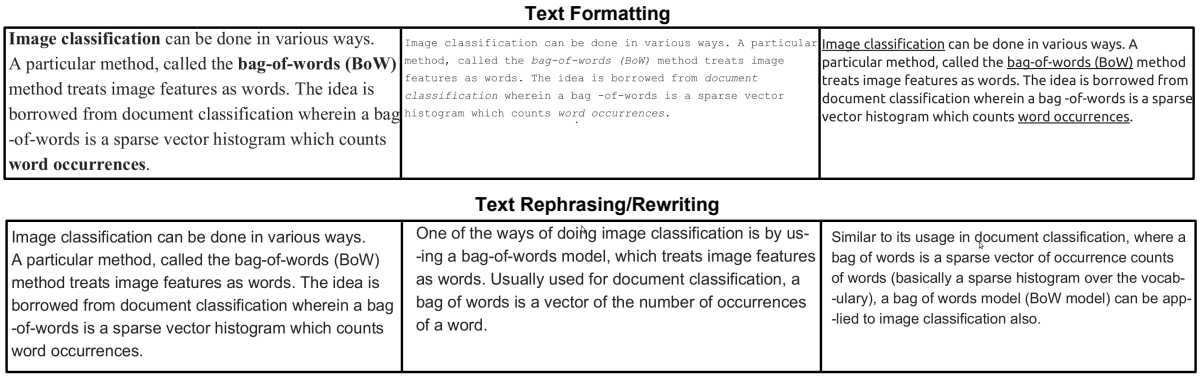


Figure 4.2 *Some examples of near duplicates: A paragraph on* BoW *taken from Wikipedia and its different modified versions. First row shows the different format of the text in which the content is not changed whereas in the second row, content of the original paragraph has been changed, however, its semantic meaning has been remained intact. We are interested in retrieving all the documents from database which contains such similar text/paragraph to a given query document image.* ***(best viewed in 2× zoom)***

the percentage of cut and paste content in document image and changing the number of sources from which content has been copied. In all the above scenarios, we are successfully able to detect duplicate segments with a considerable high accuracy.

To overcome the problems associated with above method, we propose another method for duplicate detection in a flexible Bag of Words (BoW) setting. We aim to retrieve documents from the database which have similar text to a given query document or region image. Unlike previous method, we use a different measure to analyse our performance for this method. Fig. 4.2 shows some of the examples of near duplicates/similar text we have considered in this work. Given a query image, we are interested in retrieving all the documents from the database which contain such modified instances of the query text. Note that retrieving such instances is a challenging task as the entire shape/size and the relative position of the words have been changed as well as the content has been rephrased (addition of some newer and removal of older words). We show the robustness and correctness of our approach for retrieving such similar instances in multiple settings especially when the text has been formatted (different lines breaks, font type/size, etc.) and has been translated by humans and machine keeping the semantic meaning intact.

The rest of this chapter is organized as follows: Section 4.2 briefly reviews some related works on recognition free document retrieval. Section 4.3, 4.4 and 4.5 describes the proposed methods for duplicate detection. Section 4.6 discusses the scalability issues. Experimental results are presented in Section 4.7 and conclusions are given in Section 4.8.

4.2 Related works

In the past, near-duplicate detection has been explored extensively for natural images. Some of them includes method based on matching raw PCA-SIFT features [83] and indexing using min-Hash technique [84].

Recent works on the recognition-free near-duplicate document detection are based on extracting the spatial layout information from document images and on interest point matching. Hu et al. [85] segments the document image into paragraphs and columns and extract coarse shape information , Doermann et al. [86] extract shape features of words in the image without using OCR to retrieve near duplicates, Kesidis et al. [87] uses synthetic images of words to match image-windows centered on words in document images for indexing and retrieving near-duplicates whereas Jain and Doermann [88] present a text retrieval approach using SURF features. Vitaladevuni et al. [6] attempts to retrieve near duplicates by matching SIFT interest points stored in kd-tree.

Since our method is recognition free, our closest work is that of retrieving similar documents given a query document, which is popularly known as recognition free document retrieval. The problem of recognition free retrieval has gained significant attention in recent years from the document analysis community. Many successful attempts have been made in the past for recognition free retrieval from large databases (e.g., word spotting, camera-based document retrieval, etc.). Significant progress in document retrieval credited mostly to the development of better representation of document images (e.g. configuration of interest points [72], BoW [89, 90], profile features [82]) and better indexing schemes (e.g. LLAH [72], LSH [91], inverted-index [89]). There are two directions of work in this category. The first category of attempts consider the query as a complete (or at least a large part of) document [72], and the second category focus on query by specific word examples [91]. Because of the advancement of the portable devices and commodity hardwares, research in this direction also needed to support camera based queries [73, 70], as also is Google-Goggles [92].

In the past, word spotting has been emerged as an encouraging method to perform text recognition free based retrieval. In word spotting, given a query word image, one is interested in retrieving the regions of the document images where a query word may be present. It must be noted that the query is limited to words only in this case. Generally, for word spotting, a query image is represented as a sequence of vectors and its similarity is computed with candidate regions of the documents in database using some kind of similarity measure (eg., DTW, HMM-based, etc.) [82, 93, 94]. Many segmentation-free approaches had also been proposed in the recent years [95, 90, 96]. Rusinol et al. [95] proposed a patch-based framework in a bag-of-words setting and uses latent semantic indexing technique whereas Almazán et al. [96] uses a dense grid of HOG descriptors in a sliding window fashion to solve the problem.

As mentioned above, camera-based document retrieval is also a text recognition free based retrieval. Because of the increasing availability of low-priced digital cameras, many applications have been built over the recent years to work on images captured using camera. The images captured from cameras are of often low quality and suffers from perspective distortion. In most applications, perspective distortion

is removed with the help of homography [70, 73] or fundamental matrix [25] so that the same scanner-based document analysis techniques can be applied to camera images also. In camera-based retrieval, aim is to retrieve a document image from the database corresponding to the query image captured. For this task, it is required to solve the problem of "perspective distortion" of images, as well as to establish a way of matching document images efficiently. To solve these problems, an efficient hashing technique called Locally Likely Arrangement Hashing (LLAH) has been proposed [70]. LLAH removes the perspective distortion of the images by extracting features which are geometrically invariant. Over the years, many improvisations of LLAH has been proposed [71, 72, 73]. We also use homography to remove perspective distortion while detecting duplicates. Note that the method is intended to solve the problem when the query document is exactly identical to one of the document images in the database and hence, they do not aim at retrieval of exact duplicate documents and also when there are changes/manipulations in the documents like text formatting, rephrasing of text, etc. However, our focus is on retrieving documents from the database which are exact or near duplicates to the query document.

One of our methods is based on BoW model for duplication detection as remarked above. In the past, BoW representation [7, 11] has been used successfully for the task of visual category recognition which involves the following three steps: (i) Local image features extraction (eg., SIFT [36], HOG [40], etc.), (ii) Histogram creation by pooling local features, (iii) Classifier learning (eg., SVM). We follow the similar pipeline for finding documents containing near duplicate regions to a given document. We begin with extraction of local features and histogram creation, followed by learning a classifier which can effectively discriminate the query document image from other texts. Then, we use scanning window based method [40, 4] to classify and assign score to each of the sub-windows in the candidate documents using the classifier learned from query image. BoW representation has also been used for many document image applications (word spotting [90, 96], word retrieval in large document collections [89], logo spotting [97], etc.) in the past.

Our framework can be used for detecting forgery and plagiarism in documents. In the past, document forensics and security has emerged as a prominent research area with immediate applications. A major direction of research in this domain is to validate the authenticity of the handwritten documents. Some of the recent works in the field of forensic science includes signature verification [98, 99, 100], handwriting analysis [101], fingerprint analysis [102], etc. The work has also been done on automatic detection of forged documents by detecting the geometric distortions introduced during the forgery process [103].

4.3 Optimization and Duplicates Retrieval

In this section, we discuss the two models which can be adapted to solve the problem of duplicates retrieval. Our first model is the geometric-based model which involves matching of documents using homography. Consider the document images in Figure 4.3. Contents of both the images are same, however, the images have been taken at different angles and also geometric transformations have been applied to paragraphs, figures, lines and the other contents in one image. Single homography would

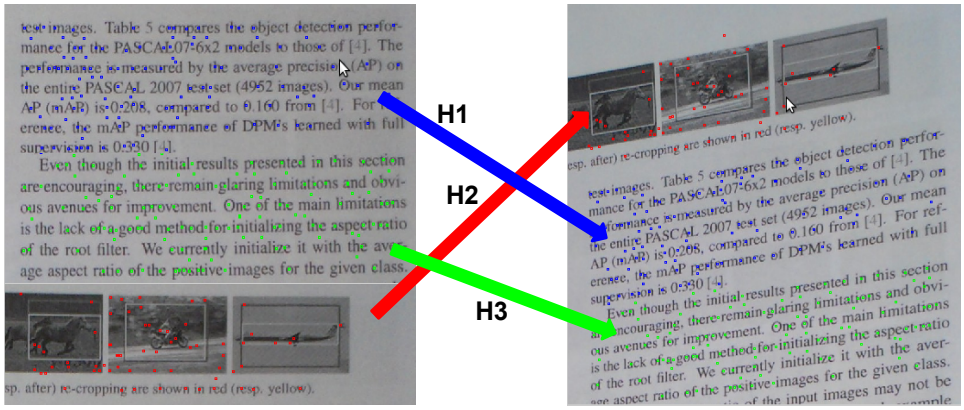


Figure 4.3 *Invariance to view point variations. Same colour shows points belonging to same homography. The two documents are matched using three homographies H1, H2 and H3.*

suffice if we match two views of the same documents [14] and no cut and paste have been applied to their contents. In our case, we need multiple homographies to match the two documents (single homography for each part of the image). Note that the problem is very challenging mainly due to its well known chicken and egg causality dilemma. In order to estimate the multiple homographies between two documents, one should better find the membership of each matching point-pair first and reversely, to find the membership, the information of each homography is much helpful. To practically solve the problem of fitting multiple homographies between two documents, a simple LP formulation is used. One of the major disadvantage of the above model is that it doesn't handle variations in text as it makes the use of geometry very hard.

Another model is the histogram-based model which represents document images as a histogram of visual words. The model is based on training a linear SVM classifier for the query document image (say paragraph). Then, the sliding window approach is applied at multiple positions and scales of a document image with which we are matching. The classifier determines whether there is a near duplicate of the query segment at given position and scale of the document. Naturally, this process is computationally expensive. Hence, we perform the cell-level sliding window to speed up the process. This model is practically more useful than the previous as training a classifier well generalizes the query document histogram which enables us to detect near duplicates also in documents that will have slightly different histograms.

In the next two sections, we give a practical formulation of the above models for recognition free copy detection in questioned document images.

4.4 Duplicate detection as fitting mixture of homographies (Method 1)

The problem is first solved by generating a candidate homographies set and it is assumed that all the true homographies exist in the candidate set. We relax this assumption in the next section. Binary vari-

ables are declared for every pair of matching point-pair correspondence between two document images and the homographies in candidate set. Then, the LP is solved to know which homography matches which point-pair correspondence. At last, homographies are re-estimated from the correspondences that have been assigned to them. Unlike the local RANSAC algorithm introduced by Iwamura et al. [104], we use LP to fit the multiple homographies as it seeks to achieve a better optimum value than the former [25].

Given two documents, and a set of interest points (say extracted with SIFT detectors), we can match them and fit a homography, since the point correspondences across these documents will satisfy the relationship $\mathbf{x_i} = H\mathbf{x'_i}$. Given enough matches, one can compute a robust estimate of this homography by minimizing an error function of the form

$$\sum_i d(\mathbf{x_i}, H^{-1}\mathbf{x'_i})^2 + d(\mathbf{x'_i}, H\mathbf{x_i})^2 \tag{4.1}$$

where $d(\mathbf{x}, \mathbf{y})$ denotes the Euclidean distance between points represented by $\mathbf{x}$ and $\mathbf{y}$.

However, in our problem this direct solution does not work. We want the matches across multiple parts of document. This naturally imply that we need to find multiple homographies simultaneously. For this purpose, we introduce a binary variable z_{ik} to denote whether ith correspondence is part of the kth homography or not. Assume we have a list of M candidate homographies, denoted by $\phi = \{H_1, H_2, \ldots, H_M\}$. We assume that all the true homographies are present in the candidate homographies set. Let $\mathbf{x_1}, \mathbf{x_2}, \ldots, \mathbf{x_n}$ be the matching point correspondences between two documents. We now define the error function as :

$$\sum_{i=1}^{n}\sum_{k=1}^{M} z_{ik}(d(\mathbf{x_i}, H_k^{-1}\mathbf{x'_i})^2 + d(\mathbf{x'_i}, H_k\mathbf{x_i})^2) = \sum_{i=1}^{n}\sum_{k=1}^{M} z_{ik}d'_{ik},$$

where $d'_{ik} = (d(\mathbf{x_i}, H_k^{-1}\mathbf{x'_i})^2 + d(\mathbf{x'_i}, H_k\mathbf{x_i})^2)$.
Let y_k $(k = 1, \ldots, M)$ denotes the binary variables indicating whether H_k is one of the true homographies. Similar to the objective function defined in [25], we also have the data term and complexity term in our formulation explaining how well the homography model fits the data and how complex our models are respectively. Minimization of the error function can be modelled as an integer linear program :

$$\min_{\mathbf{z},\mathbf{y}} \sum_{i=1}^{n}\sum_{k=1}^{M} z_{ik}d'_{ik} + \beta\sum_{k=1}^{M} y_k \tag{4.2}$$

subject to the constraints:

$$A : \sum_{k=1}^{M} z_{ik} = 1 \quad B : \max_{i=1}^{n}\{z_{ik}\} = y_k \quad C : z_{ik}, y_k \in \{0, 1\}$$

Constraint A assures that every matching point-pair correspondence can belong to only one homography of the candidate set and constraint B allows that correspondence i can be assigned to homography k only if H_k exists in the true homography set. The above proposed formulation is binary integer linear

programming which is hard to solve in practice. LP-relaxation is one of the known and effective methods to solve such problems in which binary variables are replaced with real continuous variables. We have used LP-relaxation to solve the above formulation. This provides a soft assignment solution.

Once the membership of all matching point-pair correspondences is known i.e. the homography to which each correspondence belongs, the homographies are re-estimated with all the correspondences that belongs to it using RANSAC algorithm (Equation 4.1).

4.4.1 Handling outliers

The above formulation has limitation that it assigns every matching point-pair correspondences in documents to a homography. But in practical scenarios, we have lot of false positives in point-pair correspondences, e.g., in document images, because of presence of stop words such as *the, who, is* and repetition of characters there will be lot of false point-pair correspondences. In general, we want our model to ignore all such correspondences and learn homography from true point correspondences. Such false point-pair correspondences are known as outliers and they do not correspond to any homography. In order to make our formulation robust to outliers, we introduced another set of binary variables w_i $(i = 1, \ldots, n)$, for each point-pair correspondence to denote whether it is an outlier or not. The variable w_i=1 indicates ith correspondence is an outlier and w_i=0 otherwise. Following is the formulation of mixture of homographies incorporating outliers -

$$\min_{\mathbf{z},\mathbf{y},\mathbf{w}} \sum_{i=1}^{n}\sum_{k=1}^{M} z_{ik} d'_{ik} + \beta \sum_{k=1}^{M} y_k \tag{4.3}$$

subject to the constraints:

$$A : \sum_{k=1}^{M} z_{ik} + w_i = 1 \qquad B : \max_{i=1}^{n}\{z_{ik}\} = y_k$$

$$C : \sum_{i=1}^{n} w_i \leq P \qquad D : z_{ik}, y_k, w_i \in \{0, 1\}$$

Constraint C puts upper bound on the number of outliers in order to avoid trivial solutions. We used the above robust formulation in all our experiments. Note that while matching interest points in document images, number of outliers are very high as explained above, however, our formulation can robustly ignore all such false matchings.

4.4.2 Candidate Set Generation

In Section 4.4 and 4.4.1, we have made strong and crucial assumption that the set of candidate homographies are known to us beforehand. In this section, we relax that assumption. We propose an efficient technique to generate such a candidate set. Note that four matching point-pair correspondences are enough to estimate a homography. One simple naive method to generate such a candidate set is

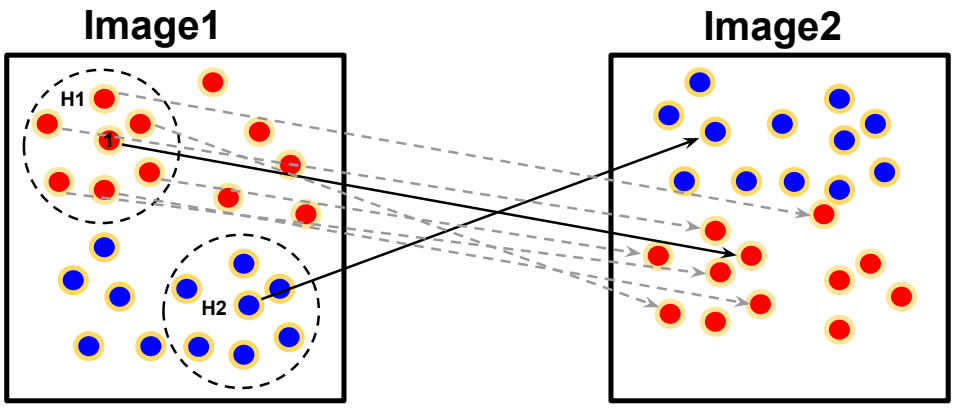


Figure 4.4 *Candidate homographies set generation. Interest points shown in red belong to one homography and blue to another. We have to generate candidate set such that it contains the two true homographies. In this image, we have chosen P=2, and the neighbourhood of corresponding point-pair chosen is given by dotted circle. Candidate homographies H1 and H2 are estimated from neighbourhood correspondences using* RANSAC *algorithm. For the experiments, we have chosen P=100.*

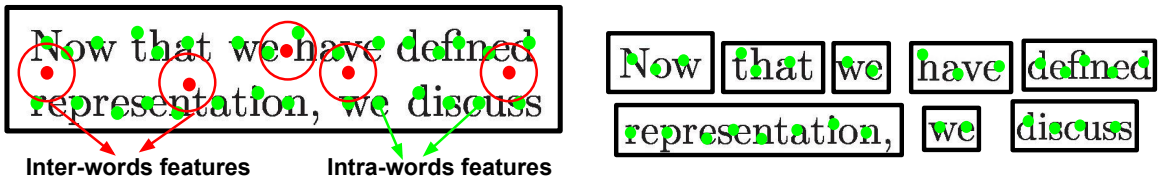
to take different sets of four random matching point-pair correspondences and estimate homographies from them. Although this technique works well but is computationally very expensive as the size of candidate set will be very large if we want to ensure that it contains all the true homographies that exist in the image.

We use a variant of the above method which also takes into account spatial information of the coordinates of matching point-pairs while estimating homographies as there is a high probability that the neighbouring correspondences will also belong to the same homography. Once the correspondence between images are known, M random correspondences are picked which are spanned uniformly across the image. Let us denote this set by ϕ. $\forall(p_{i1}, p_{i2}) \in \phi$, where p_{i1} denotes the coordinates of ith correspondence in Image1 and p_{i2} in Image2, we compute the n_e nearest-neighbours of p_{i1} in Image1. Let $N(p_{i1})$ be the set containing nearest-neighbours of p_{i1} and now, define $Q(p_i) = \{(q_{k1}, q_{k2}) \mid q_{k1} \in N(p_{i1})\}$, where (q_{k1}, q_{k2}) denotes the correspondences between Image1 and Image2. We apply RANSAC algorithm to estimate a candidate homography H_i from correspondences in $Q(p_i)$. In this way, candidate homography set has M homographies in total.

For the results proposed in this chapter, we set P=100 and n_e=20. To speed-up the nearest neighbour search, we have used kd-trees [105]. Figure 4.4 explains the method diagrammatically for candidate set generation.

4.5 Duplicate Detection using BoW model (Method 2)

Consider a query document image p, our goal is to retrieve documents which contain exact or near duplicates to p. Our method is based on the assumption that near duplicates/similar text documents have lot of words (language) in common (we show in experiments section, indeed this is true in most of



(a) (b)

Figure 4.5 *Feature Extraction of the query image: (a) Features are extracted over the whole document image. (b) Features are extracted for each word by first segmenting them from the document. Former also results in the extraction of inter-words features which encodes some geometry among the words whereas in the latter there is no dependence among words.*

the real case scenarios) and hence, their BoW representation model have a lot in common which can be exploited to retrieve such documents. Our method involves the following three steps:

Feature extraction and histogram creation. For a query image, we densely compute the SIFT descriptors with a stepsize of 5 pixels using vlfeat library [106]. We extract features at different scales using squared regions of 10, 15, 20 and 25 pixels. The size of the squared regions have been chosen as above to ensure that the character and its surrounding neighbourhood have been covered appropriately. We use k-means clustering to quantize each of the local descriptors to visual words. Hard encoding has been used to create histogram of the quantized descriptors. The other variants such as LLC [38], Fisher encoding [34], etc. can also be used here.

Note that the above method of feature extraction also results in the extraction of inter-words features along with intra-word features. Inter-words features also capture the relative geometry of the words and hence, are more biased towards the query image. These features do not turn out to be very useful when we search for near duplicates in the database as the same geometry is not guaranteed in them. To tackle this, we first segment words from the query image which is a much simpler task than recognizing them. First, Sauvola's algorithm has been used for adaptive binarization of image [107] and then, X-Y tree based segmentation approach [108] to segment words. For each of the segmented words, we densely compute the SIFT descriptors, create histograms and add them to get the query image representation. The two methods of features extraction are summarized in Figure 4.5.

Finding duplicate images in the large database has been addressed in the past by Chum et al. [84]. The images are considered to be near duplicate or similar when they are of the same scene, but with possibly different viewpoints and illumination. The problem can be seen as a task of instance retrieval from the large database. They use min-Hash method which makes their query very cheap to evaluate. Similar to this, in this work, we have defined near duplicates in case of document images (Section 4.1). However, retrieval of near duplicate document images possess following challenges – (i) not all words will be present, (ii) relative position of the words will change, (iii) distracters will be present, and (iv) location of near duplicate region in the document image is unknown. Because of all these challenges, indexing based fast BoW solution [84] is not directly applicable to retrieve near duplicates in case of

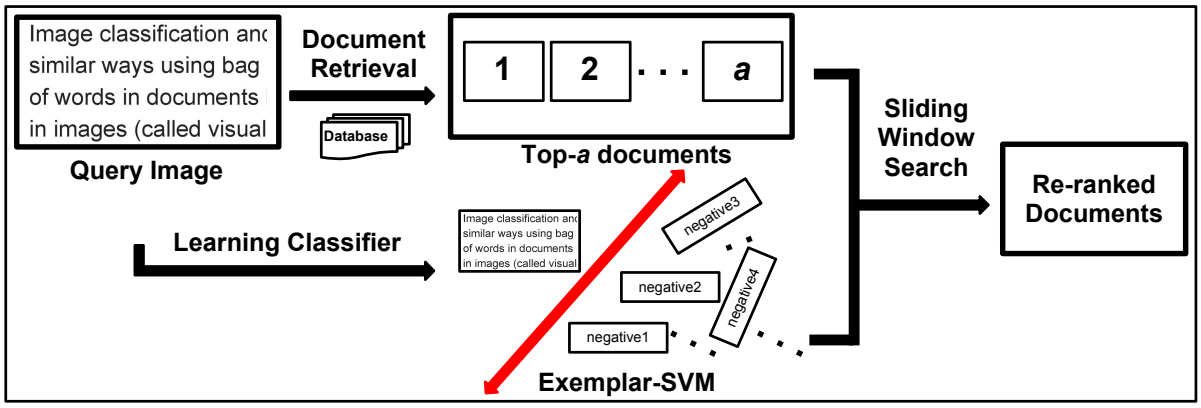


Figure 4.6 *Pipeline for the retrieval of documents containing near duplicates to a query image from database. We begin with selecting candidate documents (top-a) using* BoW *retrieval and then, re-rank them using scanning window search.*

document images. Therefore, we use a discriminative approach for recognizing such near duplicates by learning a classifier.

Learning classifier. Let $\mathbf{x}_p$ denotes the histogram of a query image which is available from previous step. Now, we learn a classifier which can separate $\mathbf{x}_p$ from histograms of other texts or regions which are not near duplicates. We use Exemplar-SVM (E-SVM) for learning such a classifier as our scenario is exactly similar to the one proposed in [109] where we have a single positive sample (p) and lots of negative samples. Let $\mathcal{N}_p$ denotes the set of negative samples which are not near duplicates (semantically different) of query document image. We show in experiments section how negative samples are selected. For learning the hyperplane, following convex objective function is minimized -

$$\begin{aligned}\phi(\mathbf{w}, b) = ||\mathbf{w}||^2 + C_1 \max(0, 1 - (\mathbf{w}^T \mathbf{x}_p + b)) \\ + C_2 \sum_{\mathbf{x} \in \mathcal{N}_p} \max(0, 1 + (\mathbf{w}^T \mathbf{x} + b))\end{aligned} \tag{4.4}$$

where C_1 and C_2 denotes the regularization parameter for positive and negative samples respectively. The above function is similar to standard SVM objective function except regularization parameter being different for positive and negative samples. Unlike the nearest-neighbour algorithm, we use discriminatively trained models for our method which has proven to perform better in the past [109].

Scanning window based search. In this step, we use scanning window based method to assign a score to a document from the database which tells us how closely it is similar to the query image. In a sliding window fashion, we scan the document image at multiple locations and scales, in each position we compute the histogram of the sub-window considered as described above and run a query image classifier. For ith document, score is given by the following equation where k denotes all the

sub-windows over which we evaluate the query image classifier score -

$$s_i = \max_k \mathbf{w}^T \mathbf{x}_k + b \tag{4.5}$$

Naturally, performing exhaustive (pixel-level) sliding window involves rigorous testing of the sub-windows and is computationally very expensive. Thus, we divide document images into $M \times N$ cells and perform sliding window at cell-level. This significantly reduces our computational cost without any loss in the performance.

4.6 Scalability

The naive approach for extending the above methods to large number of documents would be to apply the proposed LP solution to every document in the corpus and match the duplicate contents or do the BoW based scanning window search for all of them and sort them based upon their score (equation (4.5)). But this will be computationally very expensive and the standard method in literature to avoid this or to prune the candidate list is to use BoW based model retrieval [11, 110]. The BoW model quickly filter out related images in the corpus when implemented efficiently using reverse index table. After this, solving of LP or scanning window search can be done for the top-a documents only for finding duplicate documents.

Firstly, SIFT [36] descriptors are computed at interest points for each document image in the corpus. A sample of the descriptors are clustered using approximate k-means algorithm to form a vocabulary of visual words. Every descriptor of the images in corpus is now quantized/mapped to the visual word nearest to it and then, used to index the images. Inverted file index is created which has an entry for each visual word storing a list of all the document images containing that word. Given the duplicate questioned image, all the visual words present in it are obtained. With the help of reverse index table, all the documents containing common visual words with the questioned document are obtained. Every document image is represented using a M (vocabulary size) dimensional vector $< t_1, t_2, \ldots, t_M >$ where each term is weighted using tf-idf -

$$t_i = \frac{n_{id}}{n_d} \log \frac{N}{n_i} \tag{4.6}$$

where n_{id} is the number of occurrences of visual word i in document image d, n_d is the total number of visual words in document image, N is the total number of document images in corpus and n_i is the number of occurrences of visual words i in the whole corpus. The questioned document image is also represented using M dimensional vector and its similarity with other document images is calculated using cosine similarity and then the images are ranked using calculated score. Top-a are then selected for solving LP or scanning window search. This pruning can result in missing of original document for matching. However, as we validate in next section, this rarely happens.

4.7 Experiments and Results

We demonstrate the performance of proposed methods in wide variety of settings for retrieving documents containing exact or near duplicates to a given query document. We begin with detection of exact duplicates in questioned document images and then, show the retrieval of near duplicates when text has been edited and formatted with different line breaks, text widths, font/size variation or consider a scenario where same text have been written in different paper format such as ACM, IEEE, etc. Then, we show the application of our framework for copy detection in real case scenarios.

4.7.1 Datasets used

We use the following datasets in our experiments - ENGLISH **dataset**. This dataset contains 9,936 scanned images, taken from multiple English novels, each of size 2088×3524.

HINDI **dataset**. This dataset contains 9,958 scanned images, taken from multiple Hindi books, each of size 3516×4969. The quality of documents in the dataset is very poor, as a result recognition performance of OCR is low.

CV-PAPERS **dataset**. This dataset has been created by downloading previous years CVPR-10, CVPR-11 and CVPR-12 accepted research papers from the internet[1]. Each paper constitutes 8 images in the database. The dataset contains approximately 5000 images. The size of each image is 1700×2200.

Cut-And-Paste (CAP) **dataset**. CAP dataset contains many questioned images generated by cut and pasting random content/lines from multiple documents in the ENGLISH dataset. Some random text may also be present in the questioned image which does not match with any of the document images in the ENGLISH dataset. We divided different scenarios of cut and paste in document images into four categories depending upon the number of documents from where content has been copied and how much has been copied. They are pictorially explained in Figure 4.7. (A) *Rearrangement from one* - In this case, CAP document has been created by cutting and pasting another document but its contents have been reshuffled, (B) *Parts from one* - In this scenario, some of the contents of CAP document are original and some of them have been cut and pasted from another document. (C) *Rearrangement from many* - In this case, CAP document has been created by cutting and pasting contents from multiple documents. CAP documents doesn't have any original content in this case. (D) *Parts from many* - This case is a generalization of above case. CAP documents are created by cutting and pasting some parts from multiple document sources although they have some original content also. In all these scenarios, the question/candidate document and database documents may be captured in two different imaging scheme. A CAP document may contain contents from as many as 12 different documents (this has been varied from 2 to 12). We generate 100 such CAP documents for each of the settings discussed above.

[1] http://www.cvpapers.com/

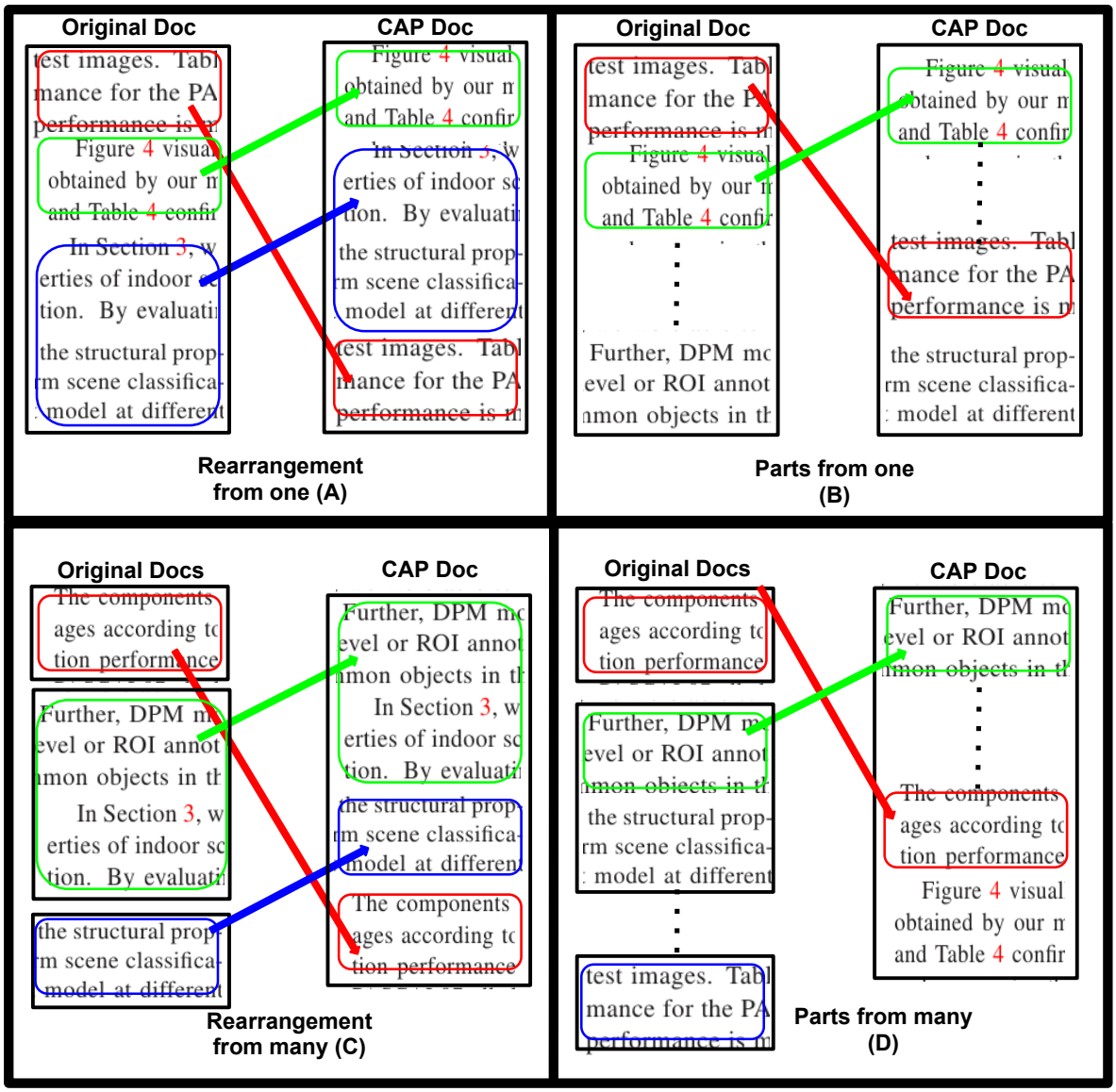


Figure 4.7 *Different scenarios explaining the creation of* CAP *documents. (A)* CAP *document is same as source document with contents being rotated, (B) Only some part of the source document is copied, (C)* CAP *document is the rearrangement of many multiple source documents, (D)* CAP *document is created by copying some parts from multiple documents.*

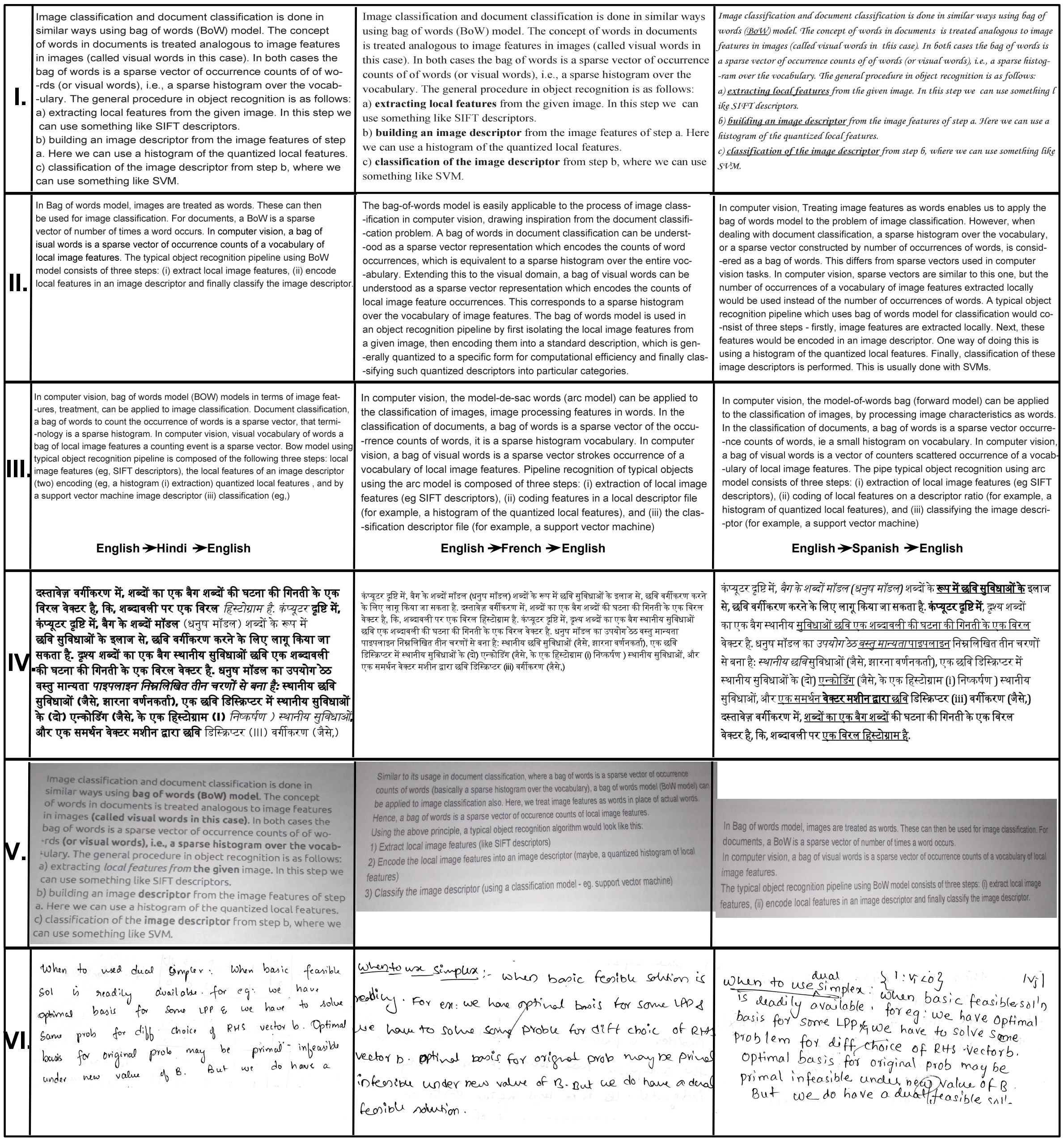


Figure 4.8 *Few examples of (I) text formatting, (II) human translated text, (III) machine translated text, (IV) Hindi text formatting, (V) camera-based documents, and (VI) handwritten assignments.* **(Best viewed in 2x zoom.)**

Thus, the CAP dataset has total of 400 documents.

TEXT-FORMATTED **dataset**. This dataset has been created by taking a 4-5 line text and generating 20 different versions of it by varying the font type/size and line breaks. Figure 4.8 shows few of the examples. The dataset has 20 images.

HUMAN-FORMATTED **dataset**. For creating this dataset, we took a paragraph on BOW from Wikipedia [111] and asked 20 users to rephrase the paragraph without changing its meaning semantically. They were allowed to shuffle sentences and use other words also that were not in the paragraph. In this way, we generated 20 different versions of the same paragraph. Some of the examples can be seen in Figure 4.8 and Figure 4.17.

MACHINE-TRANSLATED **dataset**. We use Google translator[2] to translate the same BOW paragraph taken from Wikipedia [111] into 20 different languages and again, translate it back to English to create this dataset. In this way, this dataset also contains 20 versions of the same paragraph.

HINDI-TEXT-FORMATTED **dataset**. This dataset contains 20 different text-formatted versions of the Hindi text. Some of the examples can be seen in Figure 4.8.

HAND-WRITTEN **dataset**. This dataset consists of 30 handwritten assignments comprising of 195 images. Many assignments are duplicates of each other.

CAMERA-CAPTURED **dataset**. This dataset contains the same documents used in TEXT-FORMATTED dataset but all the 20 versions of the text have been captured with the help of mobile camera.

4.7.2 Experimental Settings

Method 1. Matching point-pair correspondences between two images is found using FLANN library [112]. Firstly, possible matching point-pair correspondences is generated by finding 5 nearest neighbour of all descriptors of the questioned document in the document with which it is matched. Then, the set is trimmed such that it contains only good matches, all the matches whose distance is greater than threshold are ignored. MOSEK[3] library is used for solving linear programs.

Method 2. This method involves learning an SVM classifier which requires a positive sample and a set of negative samples (Eq. 4.4). A positive sample is available from query document while for negative samples, we pick random samples from the dataset which is guaranteed to not contain any duplicate

[2] http://translate.google.com/
[3] http://www.mosek.com/

CAP Setting	A	B	C	D	Mean
Det. Accu. (%)	94.4	93.6	85.2	83.4	89.2

Table 4.1 CAP *detection accuracy on a large corpus using Method 1.*

segments of the query document. So, while querying CV-PAPERS dataset, negative samples are picked from ENGLISH dataset for training classifier of a query document image; for ENGLISH dataset, it has been picked from CV-PAPERS dataset; for HAND-WRITTEN dataset it has been taken from scanned images of altogether different set of assignments; and for HINDI-TEXT-FORMATTED dataset, it has been taken from random scanned images of Hindi books. Hard-negative mining approach has been used to handle the large number of negative windows [4]. We use normal SIFT [36] descriptors for creation of reverse index table and for retrieving the top-a candidate documents whereas densely sampled descriptors are used for the purpose of classifier learning and window scanning as explained in the Section 4.5. As discussed in Section 4.5, we perform scanning window search at cell-level. We divide images into 50×50 cells of equal sizes and consider sub-windows of different sizes at a gap of 2 cell with smallest being 5×5. For retrieval purpose, vocabulary of size M=20,000 is taken whereas for document image representation it is 4000. We use the same regularization parameter C_1=1 and C_2=0.005 (in equation 4.4) for all the experiments and have observed that the formulation is not very sensitive to C_1 and C_2. LIBLINEAR [113] has been used to solve the SVM formulation.

4.7.3 Retrieval of CAP document images

We begin with measuring the detection and retrieval performance in CAP dataset whose content has been directly cut-and-pasted from document images in the ENGLISH dataset. Given a CAP questioned document, we are interested in obtaining all the documents from the dataset from where contents have been taken. In case of Method 1, we also spatially map the duplicate content between a query document image and the documents from database. Since, Method 1 (Section 4.4) is based on the geometry based solution, we measure the detection performance for its evaluation whereas for Method 2 (Section 4.5), we measure the retrieval performance as the method only loosely localizes the duplicate content (bounding box which gives the maximum score). Using BOW retrieval scheme as described in Section 4.6, top-500 candidate document images are obtained from the ENGLISH dataset for a given CAP questioned document. Vocabulary size is taken as $M = 20,000$.

For Method 1, LP is solved for each of the probable (top-500) document images and using mixture of homographies model, duplicate contents are spatially mapped between a questioned document and the documents from corpus. P (upper bound on the number of outliers) in (4.3) is fixed as 25% of the n (total number of matching point-pair correspondences). We have measured the detection performance of our method for every CAP setting A, B, C, D discussed in Section 4.7.1. We evaluate our method using a measure typically used for object detection task – overlap ratio of ground truth and predicted CAP. It is equal to the ratio of two areas – intersection and union of ground truth and predicted CAP. If

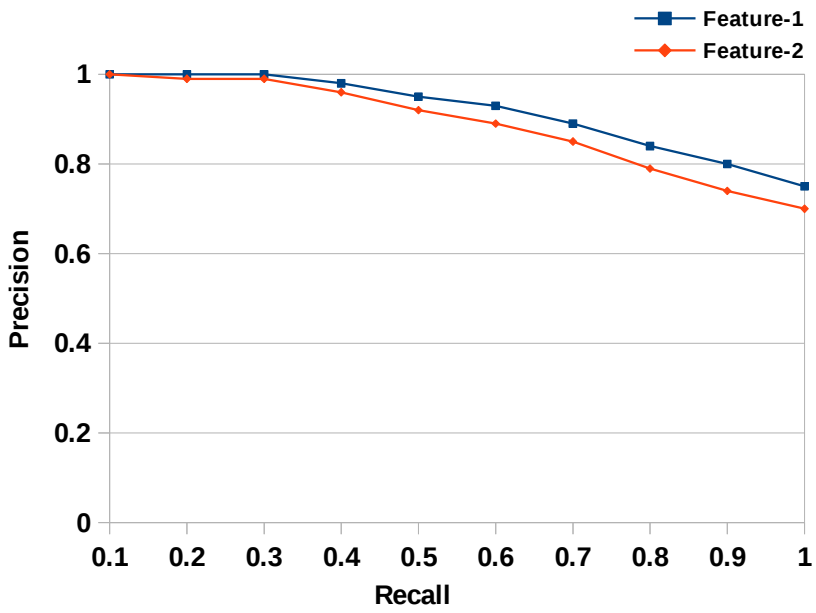


Figure 4.9 *Precision-recall curve for the retrieval of* CAP *document images. Precision and recall is computed for 400* CAP *images and average is shown here. Feature-1 denotes the case when feature is computed over whole document image whereas Feature-2 is when feature of a query document is computed using features of segmented words.*

this overlap ratio exceeds threshold (0.5), we say that the CAP has been identified and detected correctly. Table 4.1 shows the average detection accuracy on 100 CAP questioned document images over all the different settings using Method 1. It must be noted that all the detection accuracies have been obtained at a fixed false positive rate of 0.005%. On an average, we are able to detect 89.2% of the CAP in questioned documents.

For Method 2, we apply the sliding window search in the top documents to re-rank them. In Method 2 as discussed in Section 4.5, we provide few lines of text/paragraph as a query, extract features and learn a classifier, and search for it in the database. In this experiment, for each CAP document image, query is not known to us. So, we consider multiple sub-windows from the CAP documents and query each of them. We divide the CAP document image into equal-sized cells (number of cells depends on size of an image) such that each cell contains block of 34×44 pixels. We consider the region inside cell sub-windows of size 5×5, 10×10, 20×20 as query and sample such sub-windows at a gap of every 2 cells. The score of a document is given by the maximum score it receives from any of the sub-windows. Figure 4.9 shows the precision-recall (average over 400 CAP document images) curve. As expected, BOW features (Feature-1) over whole document performs slightly better than the word segmented features (Feature-2) as we are looking for "identical" region as that of query in the database, hence, geometry of words proves to be useful.

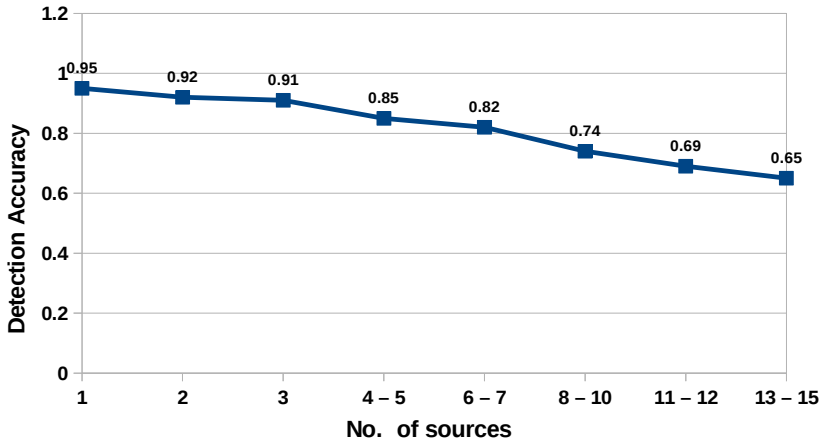


Figure 4.10 *Variation of detection accuracy using Method 1 with number of sources from which* CAP *document has been created.*

4.7.3.1 Variation with number of sources and view point

In this section, we have analysed the detection accuracy using Method 1 when the number of sources from which the content has been copied varies in a CAP questioned image. When the number of source document increases, the number of homographies using which a CAP questioned document has to be matched also increases. Figure 4.10 shows the variation of detection accuracy with the number of sources from which content has been copied. We are able to achieve the accuracy of as high as 74% even if the CAP questioned document has been copied from 10 different sources which further demonstrates the effectiveness of the approach proposed. The proposed method for detecting CAP is robust enough to handle the view point variations between CAP questioned document and the document with which it is matched. Using of SIFT-BOW pruning and mixture of homographies model makes our approach invariant to view point changes. Figure 4.3 shows an example of document matching and the corresponding homographies obtained when the view point is changed.

4.7.3.2 Variation with size of CAP content

We now analyse the variation of detection accuracy of Method 1 with the amount/size of CAP content. Size of CAP content is measured as the percentage of the area of document it occupies. For example, if only one line or word is copied from a document, then it is hard to detect such scenarios and if one paragraph or more has been copied, then it is easy to detect. Figure 4.11 shows the accuracy measure with variation of size of CAP content. As expected the problem is relatively hard when size of CAP content is very small. However, when the size of CAP content is 15% or more than that we are able to correctly detect such scenarios with an accuracy of 91.2%.

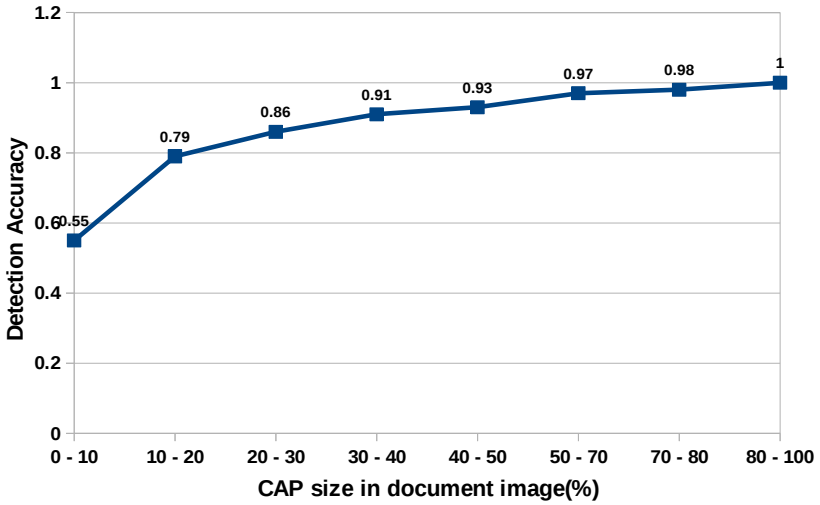


Figure 4.11 *Variation of detection accuracy using Method 1 with size of* CAP *content in questioned document.*

4.7.4 Formatting of text

In this section, we discuss the results on more challenging scenarios and perform experiments on the TEXT-FORMATTED and HINDI-TEXT-FORMATTED dataset when the text we are looking for in the database, has been formatted. Note that the experiments are performed using Method 2. The purpose of performing this experiment was to study the sensitivity of our method when the font type and size changes. We put all 20 versions of the text from the TEXT-FORMATTED dataset in random document images of the CV-PAPERS dataset. We query each of the version and measure the performance (γ_1 and γ_2) as percentage of how many of the correct document images (documents which contains modified version of the query text) are found on average in the top-20 (Table 4.2). We evaluate our method using both kind of features discussed in Section 4.5. γ_1 is when features are computed over whole document image whereas γ_2 is using segmented word features. Using BOW retrieval framework, 500 images are filtered out for applying scanning window search. On an average, our method finds 17.8 (89.0%) correct documents in top-20 which shows the robustness and correctness of our method in case of text formatting. We have also compared our method with SIFT-based matching using kd-tree [6] and BOW approach, and we are significantly outperforming them. Results are summarized in Table 4.2. Akin to above, different HINDI-TEXT-FORMATTED versions were put in random document images of HINDI dataset and retrieval is performed for each of the 20 versions. In case of Hindi documents also, our method retrieves 17.5 (87.8%) correct documents in top-20 which shows invariance of our method to different languages.

In this experiment, we have also evaluated our method when the query is provided as a image captured using mobile phone. We capture all the 20 images in TEXT-FORMATTED dataset with the help of mobile camera and query each of them (see CAMERA-CAPTURED dataset). This results in significant

Query type	# of queries	# of documents in database	Performance (%)			
			γ_1	γ_2	[6]	BoW
I. Formatted text	20	5000	80.5	**89.0**	10.2	08.5
II. Human translated text	20	5000	71.2	**81.6**	14.6	12.2
III. Machine translated text	20	5000	84.0	**94.0**	20.1	16.6
IV. Hindi Text	20	4550	70.1	**87.8**	9.4	08.7
V. Camera-based queries	20	5000	69.3	**82.2**	9.8	08.3
I + II	20	5000	65.5	**78.5**	06.2	04.9
I + III	20	5000	78.0	**90.0**	07.8	05.4

Table 4.2 *Retrieval performance of the different formatting and editing scenarios discussed in Section 4.7.4 and 4.7.5. We measure performance (γ_1 and γ_2) as the percentage of correctly retrieved documents in the top-20 documents. I + II denotes the case when text formatting is also applied to the human translated versions and I + III is when text formatting is done for machine translated versions. γ_1 measures the retrieval performance when features are computed over whole document whereas γ_2 measures the performance when feature of a query document image is computed using segmented word features.*

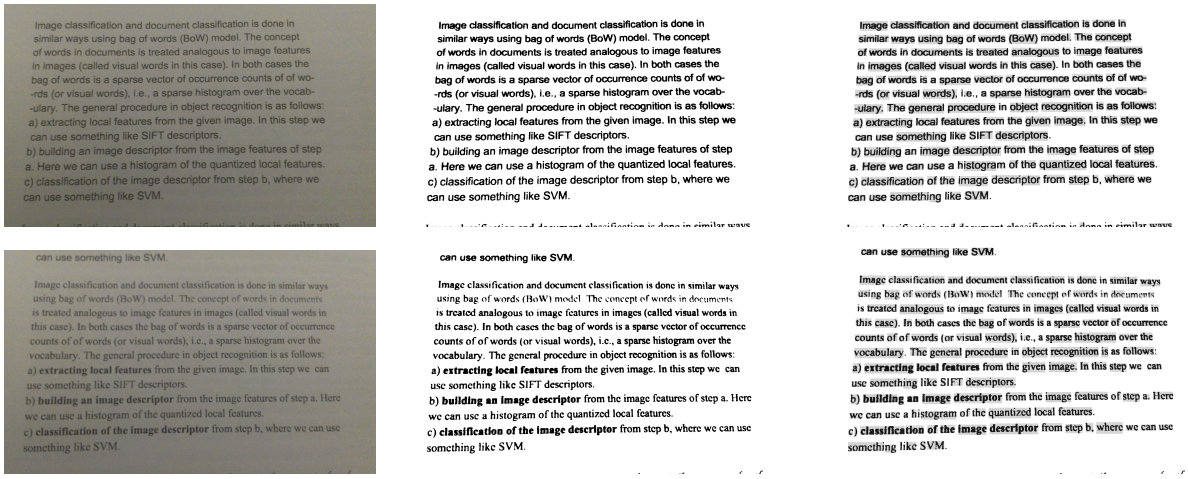


Figure 4.12 *Demonstration of typical steps followed to segment words in a camera-based query document image.*

degradation of documents which makes the use of recognition based methods for retrieving duplicates very difficult. In such degraded documents also, our method is able to retrieve near-duplicate documents with considerable high accuracy. Figure 4.12 summarizes the word-segmentation for CAMERA-CAPTURED dataset.

4.7.5 Human and Machine Translated Queries

In the previous experiment, only format of the text had been modified. In this, we assess our method when text is also changed/rephrased. Again, here also, experiments have been performed using Method 2. HUMAN-TRANSLATED and MACHINE-TRANSLATED dataset have been used for experiments. In this experiment also, 20 versions of the text from HUMAN-TRANSLATED dataset were put in random document images of the CV-PAPERS dataset and we evaluate the performance of these human translated query documents in the same manner as discussed in 4.7.4. We also evaluate our method on machine translated text queries. Each of the 20 machine translated versions from MACHINE-TRANSLATED dataset were provided as query and again, the performance is evaluated in the same manner. Our method retrieves 14.2 (71.2%) and 18.8 (94.0%) documents in top-20 in case of human and machine translated queries respectively, which shows the robustness of our technique in retrieving semantically similar documents. Table 4.2 summarizes and compares the retrieval performance with [6] and BOW. Figure 4.17 shows the qualitative result for one of the human translated query.

4.7.6 Copy Detection in Handwritten Assignments

In this section, we extend our approach to detection of duplicates in handwritten assignments. This experiment presents further more challenges for duplicate detection because of the non-uniformity of fonts in the image. 10 different regions are selected from HAND-WRITTEN DATASET and are provided as a query to HAND-WRITTEN dataset which contains nearly 200 documents. Duplicate images of all the 10 query regions are manually marked to obtain the ground truth. In this experiment, no preprocessing (BOW retrieval) is done to select candidate documents. We have also compared our approach to direct SIFT-based matching [6] and the performance is analysed in Figure 4.14. Our discriminative approach is completely outperforming the SIFT based matching. Figure 4.13 shows the qualitative results of one of the duplicate hand-written assignment. We observe that the most of failure cases is due to the poor performance of word segmentation algorithm.

4.7.7 Copy Detection in Research Papers

In this section, we show a practical application of the proposed methods for detecting duplicates in two highly similar research papers[4]. We added the first paper[5] to our CV-PAPERS dataset and a part from

[4]http://academicsfreedom.blogspot.in/2012/07/plagiarized.html

[5]http://www.computer.org/csdl/trans/tm/2012/01/ttm2012010086-abs.html

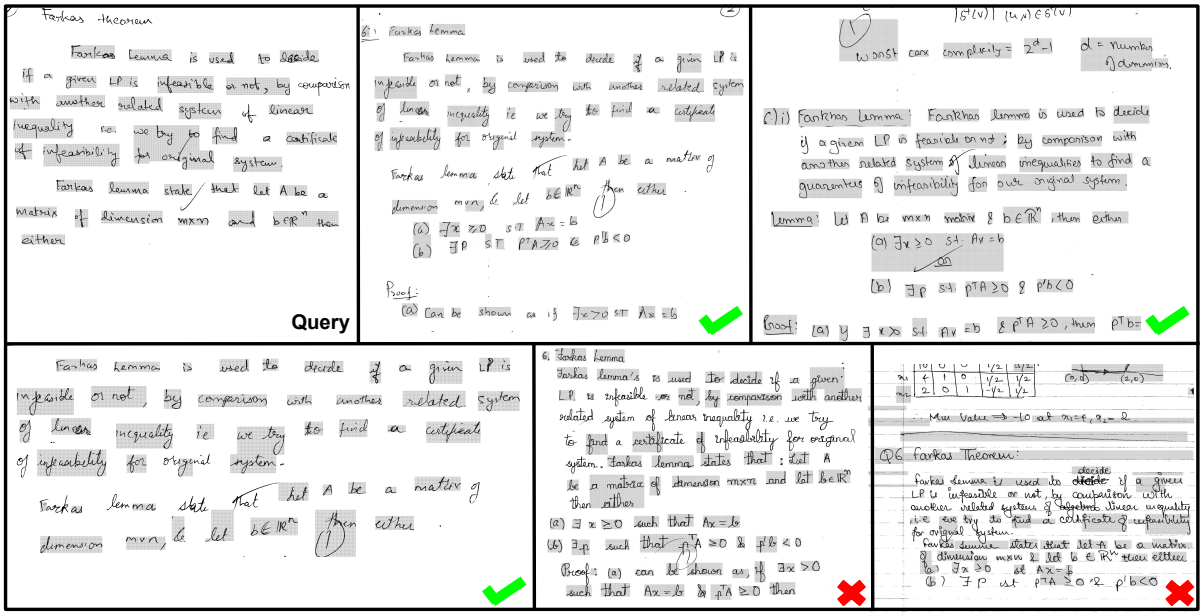


Figure 4.13 *Few hand-written assignments and their performance for one of the given query document. Figure only shows the matching segment of the retrieved documents. Green mark represents that the document containing that near duplicate has been retrieved in the top-10 whereas red mark represents that the document containing that near duplicate has not been retrieved in the top-10. Grey patches shows the word-segmentation results.*

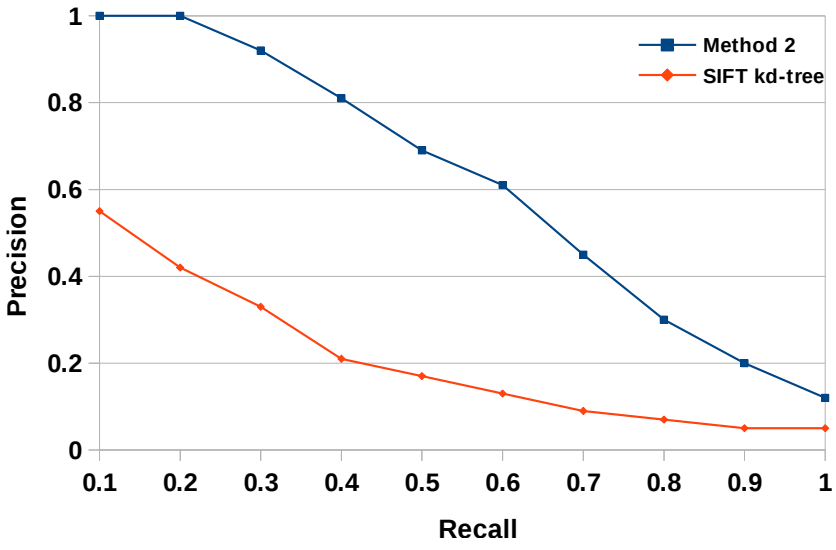


Figure 4.14 *Precision-recall curve for the retrieval of handwritten assignment images. Precision and recall is computed for all the queries and average is shown here. Method 2 denotes the case when the duplicate detection is done using proposed method (Section 4.5) whereas* SIFT *kd-tree is when method proposed in [6] is used to detect duplicates.*

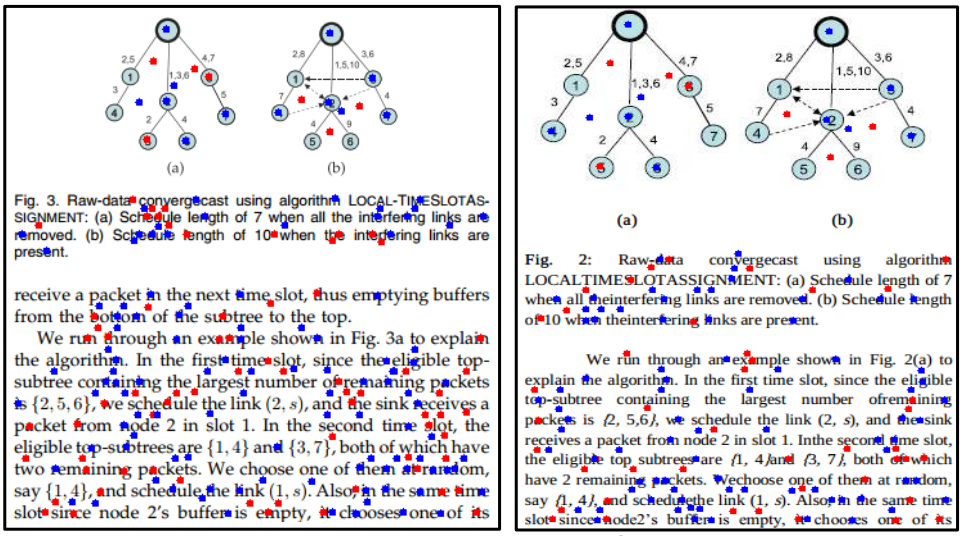


Figure 4.15 CAP *detection in two highly similar research papers. Papers are matched using two homographies. Figure only shows the snippet of the paper from the database, which gets matched with the query.*

the second paper[6] is provided as a query document image (shown in Figure 4.15). We run our geometric-based duplicate detection algorithm (Method 1) for the query document image and our method is able to correctly match it with the other document in the database. Figure 4.15 shows the CAP content matched between the query segment and the document from database using Method 1. Although the formats of the two paper are different but still our method is able to detect CAP. However, Method 1 is generally suited for the scenarios of exact duplicate detection. Also, using Method 1, we can find percentage of the cut and pasted content in the journal and the conference version of the same paper.

We also show that the histogram-based Method 2 can also be used for detecting similarity in two highly similar research papers. Experiments have been performed on the same research papers as mentioned above. We provide multiple lines/paragraphs from the first document as query and in all the cases, the top document retrieved using our method was the one which contains similar paragraph to a query document. The method has been tested for 25 different query regions of the research paper document. Figure 4.16 shows one of the corresponding query paragraph and the maximum scored region of the top document obtained using our method.

4.8 Summary

In summary, we proposed a recognition free solution to the problem of duplicate detection in document images. We propose two models for duplicate detection - geometric and histogram based and their practical challenges. Firstly, a simple LP formulation is used to detect exact Cut-And-Paste in document

[6]http://www.ijmra.us/project%20doc/IJMIE_JULY2012/IJMRA-MIE1412.pdf

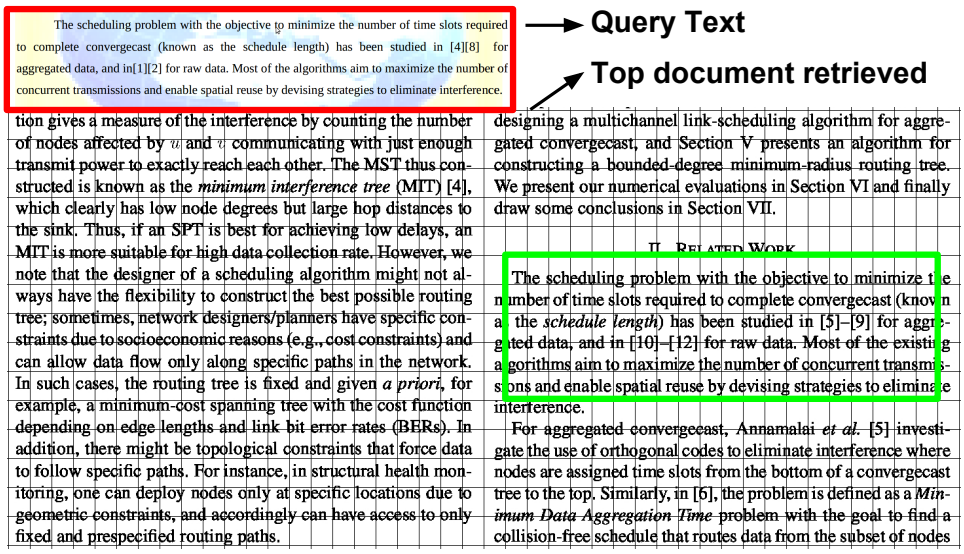


Figure 4.16 *Detecting similarity in two highly related research papers. Query text (red box) shows the paragraph from one of the papers which is used as a query. We retrieve the top-500 documents using* BoW *retrieval and re-rank them using scanning window search. Green box shows the sub-window of the document which gives maximum score using cell-level scanning window search.*

images. Filtering of documents in a large database is done using BoW retrieval approach and then, the duplicate content is identified by matching the top documents with the questioned document using mixture of homographies model. Our experiments confirm the effectiveness of the proposed approach, we are able to achieve considerable high accuracy when number of sources from which CAP document has been created is large and even when the percentage of CAP content in a document is less. The proposed formulation is also robust to handle the outliers and view point variations.

One of the main challenges with the above model is when the text is formatted and edited (for example, the text is formatted in a text editor with different line breaks, text widths and font/size variation.). This makes the use of geometry very hard. In order to overcome that, we propose a simple histogram based sliding window solution in a more flexible Bag of Words setting. Although theoretically previous method is more sound but practically, histogram based method is more effective. In this method also, we filter the database to get candidate documents using BoW retrieval approach and then use scanning window search to re-rank the documents. Score is obtained for each of the filtered document using the discriminative classifier learned using Exemplar-SVM. Our experiments confirm the superiority of our method in multiple settings especially when there is change in pattern, shape, size of the text and also when rephrasing or rewording of the text is done.

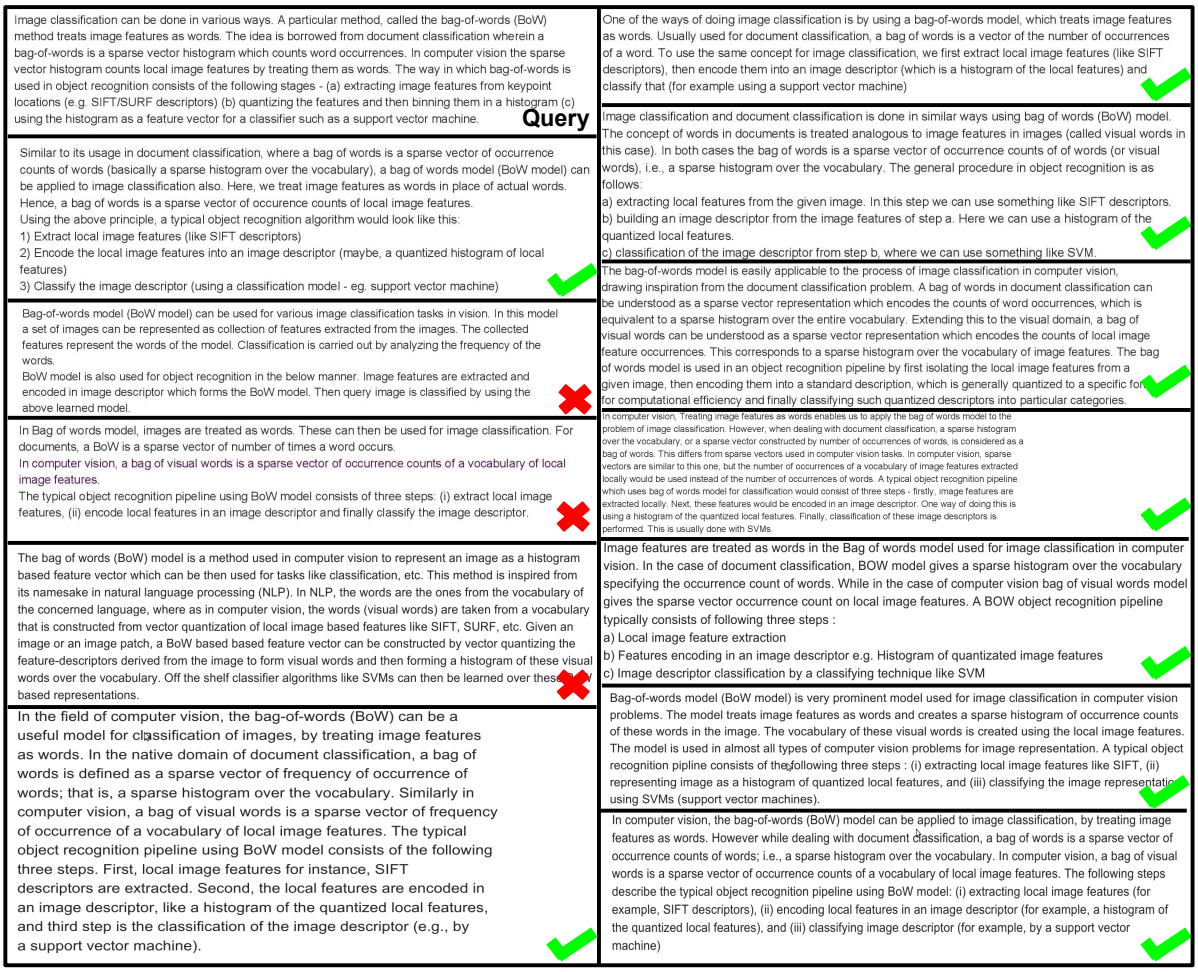


Figure 4.17 *Few human translated near duplicates and their performance for one of the given query document. Green mark represents that the document containing that near duplicate has been retrieved in the top-20 whereas red mark represents that the document containing that near duplicate has not been retrieved in the top-20.* **(Best viewed in 2x zoom.)**

Chapter 5

Conclusions and Future Work

In this thesis, we introduce the concept of decomposition in natural and document images. Decomposition refers to the process of separating logical parts in an image. Logical parts depends upon the task and type of image for which decomposition is performed (for example, it can be wide variety of objects in image or foreground and background regions or textual and non-textual regions). We also show how previous computer vision problems of object detection, semantic segmentation, action classification, document layout analysis, word spotting, etc. can be contemplated as a decomposition task. In our work, we introduced two new decomposition problems and proposed efficient solutions for them in an optimization framework.

Firstly, we decompose a global histogram of a natural image into histograms of its associated objects and regions. We solve it using a LP formulation, by taking an intermediate path between two harder problems, namely bounding box accurate object detection and pixel-accurate object segmentation. We have shown the decomposition results on wide variety of images using our LP formulation and also its superiority over other methods (object detection, semantic segmentation, etc.) for the task of classification. We also exhibit the importance of performing such decomposition for improving the classification performance on multiple object and PASCAL VOC 2007 datasets using object-background concatenated histogram representation of an image.

Secondly, we decompose a questioned document image into regions containing copied and non-copied contents. We solve the problem by detecting exact and near duplicate document images in the database to a query questioned document. We proposed two methods for detecting duplicates, one based on homography and a more general other method based on learning a discriminative classifier which can separate the query document region from other regions. The other method can handle various text editing, manipulations and formatting also. We presented results in wide range of document images including handwritten and camera captured document images.

5.1 Future Work

In this section, we list down some of the possible extensions of the two decomposition problems discussed and presented in the thesis.

5.1.1 Decomposing Bag of Words Histograms

- The use of our formulation for speeding up object detection task can be an important and immediate extension of our work. This will significantly speed up the detection process as our method is not based on sliding windows approach which consumes most of the time. We have shown some preliminary results on object detection in Section 3.5. We believe by including some shape prior (say, rectangular region), we can get tight bounding box around an object using LP formulation.

- For decomposing global histograms, we have used additive features (BoW histograms of dense SIFT). One possible extension is to modify the formulation and solve it for non-additive features also such as HOG and LBP.

- We have used the object-background representation of an image. There is a further possibility here to improve the image representation. One can have better binding of object and background histograms rather than just concatenation using some learning algorithm.

5.1.2 Retrieval of Exact and Near Duplicate Document Images

- A drawback of the near and exact duplicate retrieval framework is its two step nature to solve the problem in large data sets. In the first step, we use an indexing scheme for building a set of candidates and in the second step, we use geometry or sliding window approach to finalize the duplicate. In future, we wish to do both in the same step, with the help of appropriate indexing schemes to speed up the process.

- Our future work also includes semantic linking of the document images to further improve the performance and testing the method on large real datasets ($\sim$ millions images).

Related Publications

- Ankit Gandhi, Karteek Alahari, and C. V. Jawahar
  "**Decomposing Bag of Words Histograms**"
  in *IEEE International Conference on Computer Vision (ICCV)*, Sydney, Australia, 2013

- Ankit Gandhi and C. V. Jawahar
  "**Detection of Cut-And-Paste in Document Images**"
  in *Proceedings of the IEEE Conf. on International Conference on Document Analysis and Recognition (ICDAR)*, Washington DC, USA, 2013

Bibliography

[1] O. Russakovsky, Y. Lin, K. Yu, and L. Fei-Fei, "Object-centric spatial pooling for image classification," in *ECCV*, 2012.

[2] L. Ladicky, C. Russell, P. Kohli, and P. H. S. Torr, "Graph cut based inference with co-occurrence statistics," in *ECCV*, 2010.

[3] G. Sharma, F. Jurie, and C. Schmid, "Discriminative spatial saliency for image classification," in *CVPR*, 2012.

[4] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan, "Object detection with discriminatively trained part based models," *PAMI*, 2010.

[5] L. Ladicky, C. Russell, P. Kohli, and P. H. S. Torr, "Associative hierarchical crfs for object class image segmentation," in *ICCV*, 2009.

[6] S. Vitaladevuni, F. Choi, R. Prasad, and P. Natarajan, "Detecting near-duplicate document images using interest point matching," in *ICPR*, 2012.

[7] K. Chatfield, V. Lempitsky, A. Vedaldi, and A. Zisserman, "The devil is in the details: an evaluation of recent feature encoding methods," in *BMVC*, 2011.

[8] S. Haykin, *Neural Networks: A Comprehensive Foundation*, 2nd ed. Prentice Hall PTR, 1998.

[9] "Historical document layout analysis, http://olena.lrde.epita.fr/demos/historical_document_layout_analysis.php."

[10] "Document layout analysis and reconstruction, http://olena.lrde.epita.fr/demos/document_segmentation.php."

[11] J. Sivic and A. Zisserman, "Video Google: A text retrieval approach to object matching in videos," in *ICCV*, 2003.

[12] G. Csurka, C. R. Dance, L. Fan, J. Willamowski, and C. Bray, "Visual categorization with bags of keypoints," in *In Workshop on Statistical Learning in Computer Vision, ECCV*, 2004.

[13] B. Fernando, E. Fromont, and T. Tuytelaars, "Effective use of frequent itemset mining for image classification," in *ECCV*, 2012.

[14] R. I. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, 2nd ed. Cambridge University Press, ISBN: 0521540518, 2004.

[15] A. Torralba, "Contextual priming for object detection," *IJCV*, 2003.

[16] A. Oliva and A. Torralba, "The role of context in object recognition," *Trends in Cognitive Sciences*, 2007.

[17] Y. Chai, V. Lempitsky, and A. Zisserman, "Bicos: A bi-level co-segmentation method for image classification," in *ICCV*, 2011.

[18] J. Shi and J. Malik, "Normalized cuts and image segmentation," *PAMI*, 1997.

[19] D. Singaraju and R. Vidal, "Using global bag of features models in random fields for joint categorization and segmentation of objects," in *CVPR*, 2011.

[20] V. Kolmogorov and R. Zabih, "What energy functions can be minimized via graph cuts," *PAMI*, 2004.

[21] V. Kolmogorov, "Convergent tree-reweighted message passing for energy minimization," *PAMI*, 2006.

[22] Y. Boykov, O. Veksler, and R. Zabih, "Fast approximate energy minimization via graph cuts," *PAMI*, 2001.

[23] A. Delong, A. Osokin, H. N. Isack, and Y. Boykov, "Fast Approximate Energy Minimization with Label Costs," in *CVPR*, 2010.

[24] N. Komodakis and G. Tziritas, "Approximate labeling via graph cuts based on linear programming," *PAMI*, 2007.

[25] H. Li, "Two-view motion segmentation from linear programming relaxation," in *CVPR*, 2007.

[26] V. S. Lempitsky, P. Kohli, C. Rother, and T. Sharp, "Image segmentation with a bounding box prior," in *ICCV*, 2009.

[27] W. Zhou, L. Zhang, and L. Jiao, "Linear programming support vector machines," *Pattern Recognition*, 2002.

[28] G. Guo and C. R. Dyer, "Simultaneous feature selection and classifier training via linear programming: a case study for face expression recognition," in *CVPR*, 2003.

[29] K. Tsuda and G. Ratsch, "Image reconstruction by linear programming," *Trans. Img. Proc.*, 2005.

[30] D. Bertsimas and J. Tsitsiklis, *Introduction to Linear Optimization*, 1st ed. Athena Scientific, 1997.

[31] S. Lazebnik, C. Schmid, and J. Ponce, "Beyond bags of features: spatial pyramid matching for recognizing natural scene categories," in *CVPR*, 2006.

[32] X. Zhou, K. Yu, T. Zhang, and T. S. Huang, "Image classification using super-vector coding of local image descriptors," in *ECCV*, 2010.

[33] J. Yang, K. Yu, Y. Gong, and T. Huang, "Linear spatial pyramid matching using sparse coding for image classification," in *CVPR*, 2009.

[34] F. Perronnin, J. Sánchez, and T. Mensink, "Improving the fisher kernel for large-scale image classification," in *ECCV*, 2010.

[35] K. Mikolajczyk, T. Tuytelaars, C. Schmid, A. Zisserman, J. Matas, F. Schaffalitzky, T. Kadir, and L. V. Gool, "A comparison of affine region detectors," *IJCV*, 2005.

[36] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *IJCV*, 2004.

[37] D. Nistér and H. Stewénius, "Scalable recognition with a vocabulary tree," in *CVPR*, 2006.

[38] J. Wang, J. Yang, K. Yu, F. Lv, T. Huang, and Y. Gong, "Locality-constrained linear coding for image classification," in *CVPR*, 2010.

[39] A. Vedaldi and A. Zisserman, "Efficient additive kernels via explicit feature maps," *PAMI*, 2011.

[40] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *CVPR*, 2005.

[41] M. A. Fischler and R. A. Elschlager, "The representation and matching of pictorial structures," *IEEE Trans. Comput.*, 1973.

[42] P. F. Felzenszwalb and D. P. Huttenlocher, "Pictorial structures for object recognition," *IJCV*, 2005.

[43] Y. Jin and S. Geman, "Context and hierarchy in a probabilistic image model," in *CVPR*, 2006.

[44] S.-C. Zhu and D. Mumford, "A stochastic grammar of images," *Found. Trends. Comput. Graph. Vis.*, 2006.

[45] P. F. Felzenszwalb and D. Mcallester, "The generalized a* architecture," *Journal of Artificial Intelligence Research*, 2007.

[46] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The PASCAL Visual Object Classes Challenge 2007 (VOC2007) Results."

[47] G. Griffin, A. Holub, and P. Perona, "Caltech-256 object category dataset," California Institute of Technology, Tech. Rep. 7694, 2007. [Online]. Available: http://authors.library.caltech.edu/7694

[48] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *IJCV*, 2004.

[49] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *CVPR*, 2005.

[50] S. Maji, A. C. Berg, and J. Malik, "Classification using intersection kernel support vector machines is efficient," in *CVPR*, 2008.

[51] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes (voc) challenge," *IJCV*, 2010.

[52] A. Vedaldi and B. Fulkerson, "VLFeat: An open and portable library of computer vision algorithms," 2008.

[53] B. Fernando, E. Fromont, and T. Tuytelaars, "Effective use of frequent itemset mining for image classification," in *ECCV*, 2012.

[54] C. H. Lampert, M. B. Blaschko, and T. Hofmann, "Beyond sliding windows: Object localization by efficient subwindow search," in *CVPR*, 2008.

[55] O. M. Parkhi, A. Vedaldi, C. V. Jawahar, and A. Zisserman, "The truth about cats and dogs," in *ICCV*, 2011.

[56] B. Alexe, T. Deselaers, and V. Ferrari, "What is an object?" in *CVPR*, 2010.

[57] A. Torralba, K. P. Murphy, and W. T. Freeman, "Sharing visual features for multiclass and multiview object detection," *PAMI*, 2007.

[58] H. O. Song, S. Zickler, T. Althoff, R. Girshick, M. Fritz, C. Geyer, P. Felzenszwalb, and T. Darrell, "Sparselet models for efficient multiclass object detection," in *ECCV*, 2012.

[59] J. Winn and J. Shotton, "The layout consistent random field for recognizing and segmenting partially occluded objects," in *CVPR*, 2006.

[60] J. Sivic, B. Russell, A. Efros, A. Zisserman, and W. Freeman, "Discovering objects and their location in images," in *ICCV*, 2005.

[61] J. Verbeek and B. Triggs, "Region classification with markov field aspect models," in *CVPR*, 2007.

[62] G. Sharma, F. Jurie, and C. Schmid, "Discriminative spatial saliency for image classification," in *CVPR*, 2012.

[63] J. Wu, "Power mean svm for large scale visual classification," in *CVPR*, 2012.

[64] J. Pearl, *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann, 1988.

[65] O. J. Woodford, C. Rother, and V. Kolmogorov, "A global perspective on map inference for low-level vision," in *ICCV*, 2009.

[66] T. Schoenemann, "Minimizing count-based high order terms in markov random fields," in *EMMCVPR*, 2011.

[67] S. Lazebnik, C. Schmid, and J. Ponce, "Beyond bags of features: spatial pyramid matching for recognizing natural scene categories," in *CVPR*, 2006.

[68] A. Oliva and A. Torralba, "The role of context in object recognition," *Trends in Cognitive Sciences*, 2007.

[69] M. Pandey and S. Lazebnik, "Scene recognition and weakly supervised object localization with deformable part-based models," in *ICCV*, 2011.

[70] T. Nakai, K. Kise, and M. Iwamura, "Use of affine invariants in locally likely arrangement hashing for camera-based document image retrieval," in *DAS*, 2006.

[71] T. Nakai, K. Kise, and M. Iwamura, "Camera based document image retrieval with more time and memory efficient llah," in *CBDAR*, 2007.

[72] K. Takeda, K. Kise, and M. Iwamura, "Real-time document image retrieval for a 10 million pages database with a memory efficient and stability improved llah," in *ICDAR*, 2011.

[73] K. Takeda, K. Kise, and M. Iwamura, "Real-time document image retrieval on a smartphone," in *DAS*, 2012.

[74] S. Madhvanath and V. Govindaraju, "The role of holistic paradigms in handwritten word recognition," *PAMI*, 2001.

[75] I. Witten, A. Moffat, and T. Bell, *Managing Gigabytes: Compressing and Indexing Documents and Images, Second Edition*, ser. Morgan Kaufmann. Morgan Kaufmann Publishers, 1999.

[76] S. Deerwester, S. T. Dumais, G. W. Furnas, T. K. Landauer, and R. Harshman, "Indexing by latent semantic analysis," *JOURNAL OF THE AMERICAN SOCIETY FOR INFORMATION SCIENCE*, vol. 41, 1990.

[77] T. Hofmann, "Probabilistic latent semantic indexing," in *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, 1999.

[78] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent dirichlet allocation," *JMLR*, vol. 3, 2003.

[79] X. Wei and W. B. Croft, "Lda-based document models for ad-hoc retrieval," in *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, 2006.

[80] "Text matching, http://textmatching.com/."

[81] "Compare suite, http://comparesuite.com/."

[82] T. M. Rath and R. Manmatha, "Word spotting for historical documents," *IJDAR*, 2007.

[83] Y. Ke, R. Sukthankar, and L. Huston, "Efficient near-duplicate detection and sub-image retrieval," in *In ACM Multimedia*, 2004.

[84] O. Chum, J. Philbin, and A. Zisserman, "Near duplicate image detection: min-hash and tf-idf weighting," in *BMVC*, 2008.

[85] J. Hu, R. Kashi, and G. Wilfong, "Comparison and classification of documents based on layout similarity," *Information Retrieval*, 2000.

[86] D. Doermann, H. Li, O. Kia, and K. Kilic, "The detection of duplicates in document image databases," in *in Document Image Databases, ICDAR*, 1997.

[87] A. L. Kesidis, E. Galiotou, B. Gatos, and I. Pratikakis, "A word spotting framework for historical machine-printed documents," *IJDAR*, 2011.

[88] R. Jain and D. Doermann, "Logo retrieval in document images," *DAS*, 2012.

[89] R. Shekhar and C. V. Jawahar, "Word image retrieval using bag of visual words," in *DAS*, 2012.

[90] M. Rusinol, D. Aldavert, R. Toledo, and J. Llados, "Browsing heterogeneous document collections by a segmentation-free word spotting method," in *ICDAR*, 2011.

[91] A. Kumar, C. V. Jawahar, and R. Manmatha, "Efficient search in document image collections," in *ACCV*, 2007.

[92] "Google goggles, http://www.google.co.in/mobile/goggles/."

[93] A. Fischer, A. Keller, V. Frinken, and H. Bunke, "Hmm-based word spotting in handwritten documents using subword models," in *ICPR*, 2010.

[94] V. Frinken, A. Fischer, and H. Bunke, "A novel word spotting algorithm using bidirectional long short-term memory neural networks," in *ANNPR*, 2010.

[95] B. Gatos and I. Pratikakis, "Segmentation-free word spotting in historical printed documents," in *ICDAR*, 2009.

[96] J. Almazán, A. Gordo, A. Fornés, and E. Valveny, "Efficient exemplar word spotting," in *BMVC*, 2012.

[97] M. Rusinol and J. Llados, "Logo spotting by a bag-of-words approach for document categorization," in *ICDAR*, 2009.

[98] D. Pu and S. Srihari, "Probabilistic measure for signature verification based on bayesian learning," in *ICPR*, 2010.

[99] V. Blankers, C. Heuvel, K. Franke, and L. Vuurpijl, "ICDAR 2009 signature verification competition," in *ICDAR*, 2009.

[100] M. Malik, S. Ahmed, A. Dengel, and M. Liwicki, "A signature verification framework for digital pen applications," in *DAS*, 2012.

[101] S. Srihari, "Evaluating the rarity of handwriting formations," in *ICDAR*, 2011.

[102] C. Su and S. N. Srihari, "Evaluation of rarity of fingerprints in forensics," in *NIPS*, 2010.

[103] J. van Beusekom and F. Shafait, "Distortion measurement for automatic document verification," in *ICDAR*, 2011.

[104] M. Iwamura, T. Kobayashi, and K. Kise, "Recognition of multiple characters in a scene image using arrangement of local features," in *ICDAR*, 2011.

[105] S. Arya, D. M. Mount, N. S. Netanyahu, R. Silverman, and A. Y. Wu, "An optimal algorithm for approximate nearest neighbor searching fixed dimensions," *J. ACM*, 1998.

[106] A. Vedaldi and B. Fulkerson, "VLFeat: An open and portable library of computer vision algorithms," 2008.

[107] S. J. . P. M., "Adaptive document image binarization." 2000, pattern Recognition 33:225 - 236.

[108] G. Nagy, S. C. Seth, and M. Viswanathan, "A prototype document image analysis system for technical journals," *IEEE Computer*, vol. 25, 1992.

[109] T. Malisiewicz, A. Gupta, and A. A. Efros, "Ensemble of exemplar-svms for object detection and beyond," in *ICCV*, 2011.

[110] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman, "Object retrieval with large vocabularies and fast spatial matching," in *CVPR*, 2007.

[111] "Wikipedia, http://www.wikipedia.org/."

[112] M. Muja and D. G. Lowe, "Fast approximate nearest neighbors with automatic algorithm configuration," in *VISSAPP*, 2009.

[113] R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, and C.-J. Lin, “LIBLINEAR: A library for large linear classification,” *Journal of Machine Learning Research*, 2008.