University of Alberta

SCALE SPACE FEATURE SELECTION WITH MULTIPLE KERNEL LEARNING AND ITS APPLICATION TO OIL SAND IMAGE ANALYSIS

by

Sharmin Nilufar

A thesis submitted to the Faculty of Graduate Studies and Research
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

Department of Computing Science

Spring 2012
Edmonton, Alberta

To my daughters
Faizah and Rifaya

Abstract

Scale-space representation for an image is a significant way to generate features for object detection/classification. The size of the object we are looking for as well as its texture contents are related to the multi-scale representations. However, any scale-space based features face the inevitable issues of high dimentionality and scale selection. Scale-space analysis of image provides a set of extremely high dimensional features at each scale- the number of pixels in a filtered output image is the feature dimensionality at that scale. Moreover, considering all the output images at various scales, the dimensionality of the feature set is staggeringly high. Selection of features from this high dimensional space is daunting. In addition, the scale selection process is still ad-hoc, while applying scale-space based features for object detection/classification. In this research these two issues are resolved by designing a suitable kernel function on the scale space based features and applying multiple kernel learning (MKL) approach for sparse selection of scales.

A novel shift invariant kernel function for scale space based features is designed here. Also a novel framework for multiple kernel learning is proposed that utilizes a 1-norm support vector machine (SVM) in the MKL optimization problem for sparse selection and weighting of scales from scale-space. The optimized data-dependent kernel accommodates only a few scales that are most discriminatory according to the large margin principle. With a 2-norm SVM this learned kernel is applied to the classification problem.

In this thesis we have applied the proposed classification method for oil sand image analysis. Automatic analysis of oil sand video images is non-trivial due to the presence of dirt and fine materials. In addition, changeable weather and lighting condition make the video quality worse. Two challenging problems in oil sand mining which are detection of large lump and steam from videos are investigated here. Difference of Gaussian (DoG) and wavelet scale space are applied for these two different detection problems, respectively. Our method yields encouraging results on these difficult-to-process video images and compares favourably against other existing methods.

Acknowledgements

First I would like to express my sincere thanks to my advisor Nilanjan Ray. It has been an honor to be his PhD student. I appreciate all his contributions of time, ideas, and funding to make my Ph.D. experience productive and stimulating. The joy and enthusiasm he has for his research was transmissible and motivational for me, even during hard times in the PhD pursuit.

I would like to express my sincere thanks to the members of the CMIS group who have contributed immensely to my personal and professional time at University of Alberta. The group has been a source of friendships as well as good advice and collaboration. I am especially grateful to Hong Zhang for his valuable suggestions, encouragements, and support throughout my PhD research. The past and present CIMS members that I have had the pleasure to work with or alongside of are grad students specially Baidyanath Saha, Zhijie Wang and Kiana Hajebi.

I would like to acknowledge my former teacher and colleague Khademul Islam Molla who was here on a post doc. We worked together on audio signal classification using scale space based MKL methods. I very much appreciated his enthusiasm, intensity, willingness to explore our proposed method on different application domain.

For this dissertation I would like to thank my thesis committee members: Hong Zhang, and Guohui Lin for their time, interest, and helpful comments. I would also like to thank the external members of my oral defense committee, Sriraam Nataranjan and Vicky Zhao, for their time and insightful questions and suggestions.

I gratefully acknowledge the funding sources that made my Ph.D. work possible. I was funded by the Natural Sciences and Engineering Research Council of Canada (NSERC) graduate fellowship and Informatics Circle of Research Excel-

lence (iCore) Scholarship for my first 2 years and was funded by Alberta Ingenuity Scholarship and iCore scholarship for years 3 & 4 of my PhD work. My work was also supported by the University of Alberta teaching and research assistantship programs.

My time at UofA was made enjoyable in large part due to the many friends and groups that became a part of my life. I am grateful for time spent with Bangladeshi Student Association at University of Alberta (BSAUA) and for many other people and memories.

I would like to thank my parents who raised me with enormous love and care and supported me in all my pursuits. Last, but not least, I would like to thank my husband Faruk and daughters Faizah and Rifaya, for their love, patience, and understanding-they allowed me to spend most of the time on this thesis and supported me during all stages of this PhD work.

Table of Contents

List of Tables

List of Figures

List of Symbols

LP	Linear Programming
$\lVert . \rVert_p$	lp-norm
$\lVert . \rVert_F$	Frobenius norm
b	Bias term
C	Trade-of parameter
d	Regularization parameter
D	Dimensionality of the original feature space
$G(.;.)$	Gaussian kernel
$k(.,.)$	Kernel function
$\mathbf{K}$	Kernel matrix
N	Number of training instances
$\mathbb{R}$	Real numbers
$\mathbb{R}_+$	Nonnegative real numbers
$tr(.)x$	Trace
x	Data instance
w	Weight coefficients
y	Output value

AUC	Area Under Curve
CV	Cross Validation
DoG	Difference of Gaussian
DT-CWT	Dual-Tree Complex Wavelet Transform
GMKL	Generalized Multiple Kernel Learning

HMT	Hidden Markov Tree
i.i.d.	independent and identically distributed
FDA	Fisher Discriminant Analysis
LoG	Laplacian of Gaussian
LMKL	Localized Multiple Kernel Learning
LP	Linear Programming
LSMKL	Large Scale Multiple Kernel Learning
MKL	Multiple Kernel Learning
QCQP	Quadratically Constrained Quadratic Programming
QP	Quadratic Programming
RMKL	Regularized Multiple Kernel Learning
ROC	receiver operating characteristic
ROI	Region of Interest
SDP	Semidefinite Programming
SILP	Semi-Infinite Linear Programming
SKM	Support Kernel Machine
SOCP	Second-Order Cone Programming
SVM	Support Vector Machine

Chapter 1

Introduction

This thesis addresses the problem of selecting useful information from multi-scale representation of images of the real physical world. The term “useful” essentially depends on the goal of the image analysis tasks at hand. Some of the most basic questions that still remain to be answered concern what type of information in images is relevant to solve different real world problems, how this information is extracted from the image data and how such features can be related to properties of environment. Scale-space framework is one of the most promising methodologies, where significant structures can be extracted from an image in a solely bottom-up way, without any priori information.

The concept of scale-space was first named and presented to the image analysis community by Witkin in 1983 [6]. The main idea in Witkin’s work is that important signal features would persist through relatively coarse scales even though their location may be distorted by a filtering process. Especially, such regions, which appear to stand out from the surroundings in the original image, seem to be further enhanced within the scale-space. Scale-space analysis typically consists of applying filters at different scaling parameters to an image to obtain a number of output images. The size of the object we are looking for as well as its texture contents are related to these multi-scale (scale-space) representations. However, a high dimensional feature space is generated at each scale by the scale-space analysis of image - the number of pixels in a filtered output image is the feature dimensionality at that scale. Moreover, considering all the output images at various scales, the dimensionality of the feature set is exceedingly high. It is challenging and nontrivial to select features from this high dimensional feature space.

The scale-space theory describes a well-founded framework that provides a rich representation of the behavior of the data across scales. However, it does not address the problem of selecting appropriate scales for further analysis. In many cases it is also important to select a particular scale or a group of appropriate scales. For example, scale selection methods are shown to be crucial for interest point detection [5], object detection [5, 7], tracking [8] and segmentation [9].

A general methodology for feature detection with automatic scale selection has been proposed by Lindeberg [10, 11, 12]. The basic idea proposed by Lindeberg is to apply the feature detector at all scales, and then select scale levels from the scales at which normalized measures of feature strength assume local maxima with respect to scale. Intuitively, this approach selects the scales at which the operator response is the strongest. Several methods for scale selection are found in literature based on the entropy measures or error measures over scales [10, 12]. Actually these scale selection methods are analogous to the method for feature detection. However in this work, we are interested to select scales and weigh them according to their discriminative power to classify images between two classes. Thus in this work, the scale selection problem is similar to the feature weighting problem.

In this thesis, multiple kernel learning technique (MKL) is proposed to deal with the two problems related to the multi-scale representation discussed above. The high dimensionality of the multi-scale feature is handled by designing suitable kernel functions on the multi-scale images viz., dealing with high dimensional scale-space data and relevant scale selection/weighting. A kernel function computes similarity or dissimilarity between two sets of (usually high dimensional) features. For example, if the two sets of features come from the same class, the kernel function assigns high similarity score to the two sets. Otherwise, if the sets of features belong to two different classes, then the kernel function assigns a low similarity score to them. Here we have proposed a method for scale selection via multiple kernel learning. MKL is an interesting approach to address several, challenging real world applications in computer vision involving several possibly heterogeneous data sources to design and integrate kernels. MKL simplifies feature/

feature-group selection to kernel selection from a set of basis kernel functions. If the basis kernel functions are defined on the individual scales of scale-space then their corresponding weights determine the importance of that scale in the classification process. Some traditional feature selection methods for example forward selection or backward elimination methods do not have any well defined objective function [13] [14]. As a result, continuous scale selection and scale weighting is not possible with these kind of approaches. Also these ad-hoc methods suffer from the high computational cost. On the other hand, MKL provides the flexibility of continuous scale selection where weights of the scales are obtained by simply minimizing an objective function. Scale selection using multiple kernel learning is a completely novel idea. In the thesis we have designed a new kernel based on circular convolution function to obtain the shift invariance similarity measure between two scale space based features.

MKL problems can be sometimes tackled via exhaustive cross validation (CV). However, if the number of parameters is large, then CV is not computationally feasible. Recently, several researchers have focused on finding more efficient methods for multiple kernel learning. The MKL framework proposed by Lanckriet *et al.* [15] is the conic combinations of kernel matrices which results in a convex quadratically-constrained quadratic program (QCQP). However, the semidefinite programming based method proposed by Lanckriet *et al.* [15] can solve this problem only for a small number of kernels and a small number of data points. It becomes rapidly intractable as the number of learning examples or kernels increases. To overcome this limitation Bach *et al.* [16] have reformulated the problem so that sequential minimal optimization techniques can be applied to handle medium-scale problems. Their method is known as support kernel machine (SKM). Sonnenburg *et al.* [17] also reformulate the binary classification MKL problem as a semiinfinite linear program, which can be solved by recycling the standard SVM implementations. Sonnenburg's large-scale multiple kernel learning (LSMKL) method makes the MKL approach tractable for large problems, employing existing support vector machine code iteratively. All the aforementioned MKL techniques used linear combination of kernels which may be restrictive in some cases. Varma *et al.* [18]

show how existing MKL formulations can be easily extended to learn general kernel combinations subject to regularizations on the kernel parameters. This generalized multiple kernel learning (GMKL) method provides richer representations of feature spaces by combining kernels in other fashions rather than linear combination. As a competitive alternative to the aforementioned MKL techniques, here the principle of large margin is utilized in 1-norm support vector machine to solve the MKL problem.

As a novel and significant application of the proposed MKL based scale selection method, two important detection problems in oil sand video analysis are solved. We have posed these detection problems as binary image classification problem. Following section discusses the motivation from oil sand image analysis.

1.1 Motivation from Oil Sand Image Analysis

Oil sands were described as "Canada's greatest buried energy treasure" by Time Magazine. Canada has become the largest supplier of oil and refined products to the United States, ahead of Saudi Arabia and Mexico. Most of the oil sands of Canada are located in three major deposits in northern Alberta covering over 140,000 square kilometers that preserve almost 1.75 trillion barrels of bitumen. However, only about two percent of this valuable resource has been harnessed to date [19, 20].

Oil sand is composed of sand, bitumen (heavy black viscous oil), mineral rich clays and water. A thorough processing of oil sand is required to convert it into an upgraded crude oil before it can be used by refineries to produce gasoline and diesel fuels. Oil sands mining occurs around the year, even during the cold winter season in northern Alberta. In surface mining, oil sand is first shoveled, then moved to the processing crusher by conveyor belts and then crushed by a crusher to produce smaller particles that are suitable for bitumen extraction. At several steps of oil sand mining, robust and reliable image processing algorithms are applied to automatically obtain accurate information with which operational decisions are made. Oil sands are observed at many points in the ore preparation pipeline, and each scenario defines a separate problem and presents different challenges. What follows is a description of two important and challenging problem towards automating the screening process of oil sand mining process.

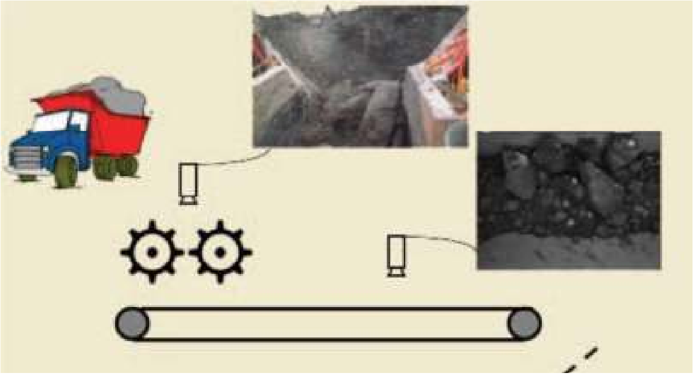


Figure 1.1: Simplified illustration of the ore preparation pipeline [2].

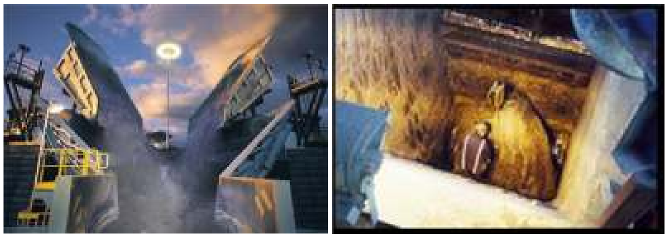


Figure 1.2: (a) Two haul trucks empty their load of oil sand into the top of a double roll crusher [3] and (b) People are employed to remove large lump in crusher caused jam event.

One of the major difficulties in the oil sand processing is the presence of large lumps which can be huge chunks of rock materials or a collection of smaller particles frozen together in cold winter. These large lumps can block the crushers, which are used to crush oil sand materials in the processing steps. Jamming of the crusher causes significant production downtime, and as jamming needs to be cleared, it translates directly into economic losses. According to one mining company data, the amount of downtime of a crusher due to the presence of large lump depends on many factors and varies greatly. It can take from ten minutes to more than two hours to deal with a lump jamming event. Also, during some months, one of the crushers can be down over 100 hours due to lump jams, while in other months it may be zero. On an average the crushers can process 7000 tph (tonnes per hour) of ore and it takes about 2 tonnes of ore to produce 1 barrel of oil. Roughly to give an idea of production loss, if a crusher is down due to a large lump event for an hour the total production loss will be around 7000 tonnes i.e., almost 3500 barrels of oil. The traditional approach of visual inspection by the operator is tedious for large lump detection. Thus, it is very important to design an effective and reliable automatic detection system. The goal of an automated alarm system will be to detect the presence of large lumps in oil sands from the video images captured on the conveyor belt. Once this detection is performed, preventive measures can be taken to reduce the chances of jamming the crusher. An essential feature of such automated technique is real time detection with low false alarm rate.

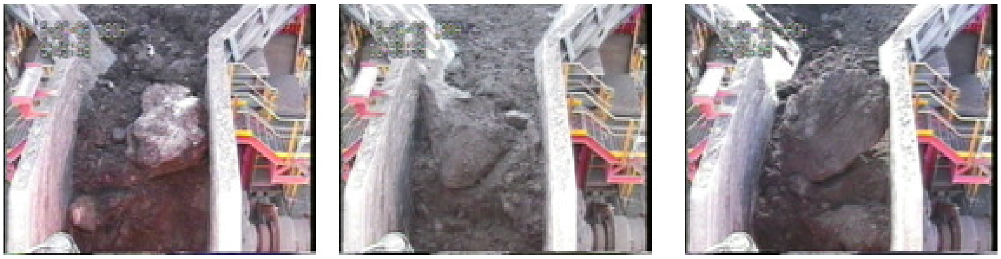


Figure 1.3: Examples of oil sand images containing large lumps.

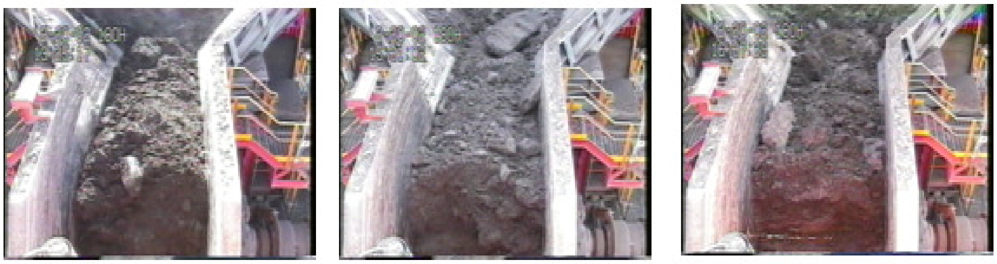


Figure 1.4: Examples of oil sand images with no large lump.

Mining images pose considerable challenge for automated processing, because of severe clutter due to poor lighting, varying weather, and harsh outdoor imaging conditions. Moreover, the shape, size and texture of oil sands show a significant variation. The apparent brightness of the individual sample varies from object to object. Most of the times, large lumps are mixed with dirt and fine materials. The aforementioned factors constitute the principle challenges to automatically detect large lumps from these images. The oil sand images are relatively novel images. To date, very little research toward automated analysis has been performed on these images. Figure 1.3 shows some of typical images of large lumps in oil sand and Figure 1.4 shows images with no significant large lump.

Another challenging problem in the oil sand image processing is the presence of steam. Since the ore is dug from the moist ground and water is used in screening, steam is produced and obscures the view of the camera. Segmentation of ore fragments must exclude images heavily covered by steam. From an image processing point of view, one can treat the problem of steam event detection. Figure 1.5 shows some of typical images covered with steam and Figure 1.6 shows images with no significant amount of steam.

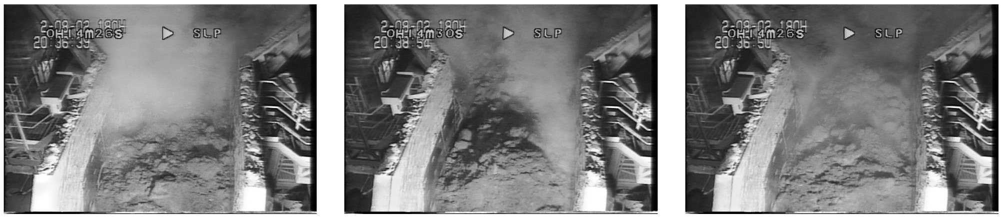


Figure 1.5: Examples of oil sand images with steam.

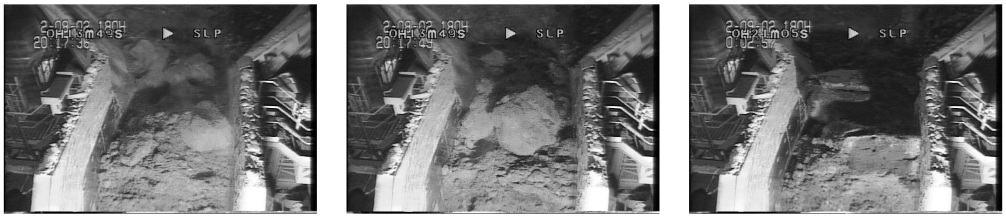


Figure 1.6: Examples of oil sand images with no steam.

1.2 Solution Overview

Automatic analysis oil sand images is non-trivial due to the presence of dirt and fine materials. In addition, changeable weather and lighting condition make the situation worse. Here, we model the problem of large lump detection and steam detection as event detection problems. The goal of the event detection is to identify specified spatiotemporal patterns in videos. This task is similar to object detection in many respects since the pattern can be located anywhere in the scene (in both space and time) and requires reliable detection in the presence of significant background clutter.

Similar to many image object detection systems, we use a sliding window approach in video, as shown in Figure 1.7. First, we specify a model of the event that we are interested in detecting. We scan this model across all locations in the video in space and time. A binary classier is trained to classify the model. When the classifier decides that we have a match, we label the event at that particular location and time, as shown by the arrow sign in Figure 1.7. We propose a novel feature for representing events. However, we use the sliding window framework throughout the entire work.

For any image classification problem we need a set of effective features which are often tackled via segmentation and/or edge-feature detection in practice. Edge detection applied to oil sand images usually results in a tangled web of edges, causing the detection problem more difficult to solve. Also previous researches on oil sand images show that segmentation is an extremely difficult task due to poor quality of these images [21].

To avoid the non-trivial edge detection and segmentation process, the multi-scale features are investigated here. Two important multi-scale representation namely difference of Gaussian (DoG) and wavelet are exploited for the detection of large lump and steam from oil sand videos, respectively. However as discussed before it is neither practical nor profitable to use the entire scale-space. Rather, it is important to find relevant scales for object detection. A novel framework for MKL is proposed for selecting only the useful scales in the image scale-space while dis-

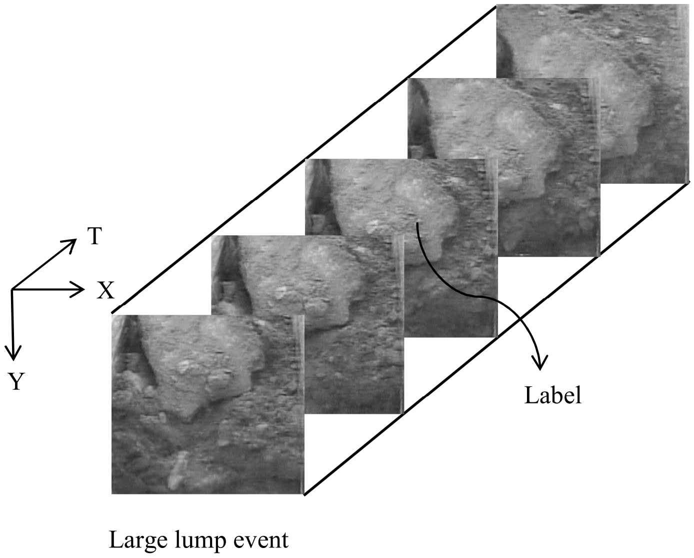


Figure 1.7: A sliding window approach to event detection. The model is scanned at all spatio-temporal locations in the video. We are able to localize the event detection in both space and time.

carding the rest.

1.3 Contribution

The main contributions of the thesis are summarized below:

- A well known issue in scale-space-based classification is the high dimensionality of scale-space data. The issue of high dimensionality is dealt by designing suitable kernel functions on the scale-space image features. Towards this end we have designed a shift invariant circular convolution kernel function for scale-space image features.
- The second issue with scale-space data is scale selection for image classification. To select the appropriate scales via those kernel functions we have proposed a novel multiple kernel learning (MKL) technique for sparse selection and weighting of scales. Thus the scale selection problem is mapped

to MKL problem. The proposed MKL utilizes a 1-norm support vector machine (SVM). The optimized data-dependent kernel accommodates only a few scales that are most discriminatory according to the large margin principle. With a 2-norm SVM this learned kernel is applied to detect/classify objects.

- Finally in this thesis challenging detection problems from novel oil sand videos, namely large-lump detection and steam detection are selected to solve. The proposed method shows promising performance for both of the detection problems.

1.4 Outline of the Thesis

The rest of this thesis is organized in the following way:

- Chapter 2 presents an overview to the scale space feature. Different types of scale-space representations are described. The concept of scale selection and literature review on different scale selection methods are also discussed. Finally a brief literature review on multi-scale feature detection methods are given.

- Chapter 3 mainly focused on SVM based binary classification. An overview of different types of SVM classifiers and kernel methods are given. This chapter also provides an overview and literature review on kernel selection and optimization. Kernel optimization techniques are discussed from two different points of view: first, different kinds of objective/cost functions adopted by different researchers are reviewed and then existing multiple kernel learning techniques are discussed.

- Chapter 4 discusses the proposed the scale-space based feature extraction process. Methods for construction DoG and wavelet scale-space for our problems are shown. Each step of the proposed method are described in detail.

- Chapter 5 describes how to use MKL approach for space space based feature selection. It describes the construction process of the novel kernel function based on circular convolution of two scale-space. The algorithm of the proposed 1-norm SVM based MKL is described in detail.

- Chapter 6 shows the application of our proposed method on oil sand video images. A literature review on the automatic oil sand image analysis is discussed. The experimental set up, datasets, and results for large lump and steam detection problems are reported. Different parts of our proposed method are evaluated with extensive experimental studies.
- Chapter 7 concludes this thesis by summarizing the whole research work and pointing some future plans.

Chapter 2

Scale-Space Feature

This chapter introduces the scale-space representation of image data that replaces an image by a family of smoothed versions of the same image. The scale-space based multi-scale features have proved as powerful for vision task as it allows a visual system to “concentrate” on the appropriate level of detail and to relate “things” across different levels of detail. This chapter is organized as follows. The concepts of multi-scale feature extraction and some examples are given in first section. Next section gives a brief summary of the scale-space methods. Several linear scale-spaces are also discussed. Some useful properties of scale-space are given in section 2.3. Finally an overview of multi-scale feature detection are given in the last section.

2.1 Multi-Scale Feature Extraction

Feature extraction is usually performed as a first step for image classification/object detction problems. Usually, a segmentation algorithm is applied to divide the image into semantically significant regions, or objects, to be recognized by further processing steps. However, it is well known that semantically significant regions are found in an image at different scales of analysis. It is often not trivial to determine the correct scale of analysis in advance, because different kinds of images require different scales of analysis, and furthermore in many cases significant objects appear at different scales of analysis in the same image. One of the remedy of this problem is to use scale-space information as features. There are several different multi-scale image processing techniques such as wavelets [22], scale-space [10], quad-tree and pyramid [23]. In the following subsection we will focus on mainly two important multi-scale approaches which we have utilized for our detection problems in this thesis.

2.2 Scale-Spaces

The structure of images has a close relationship with multi-scale representation [6, 11]. For automatically analyzing and deriving information from real-world measurements, we need some kind of operator to extract meaningful information from the image data. The relationship between the size of the actual structures in the data and the size of the operators plays an important role to determine the type of information to be extracted. If one can address this relationship properly, the task of interpreting the operator response becomes simple and efficient. Scale-space concept was first named and presented to the image analysis community by Witkin in 1983 [6]. The main idea in Witkin's work is that important signal features would persevere through relatively coarse scales even though their location may be distorted by a filtering process. Later, scale-space theory was also investigated by several other researchers to handle the multi-scale nature of image data. An example of multi-scale representation of an image is shown in Figure 2.1.

There are two important advantages of the scale-space approach. First, scale-space representation allows multiple interpretations of the data from a fine degree of detail to a higher level of description of the overall structure of the image content. Second, the scale-space approach provides the flexibility for selecting a scale or a set of scales by looking at how the interpretation of the structure captured in the scale-space changes as the scale is varied [7].

Different principles can be employed to obtain scale-spaces that achieve a description of image structures through scales. According to the application, different scale operators are applied to derive the scale-space stack. A classical approach for choosing a scale-space representation for a particular application is to establish a set of scale-space axioms [11], describing basic properties of the desired scale-

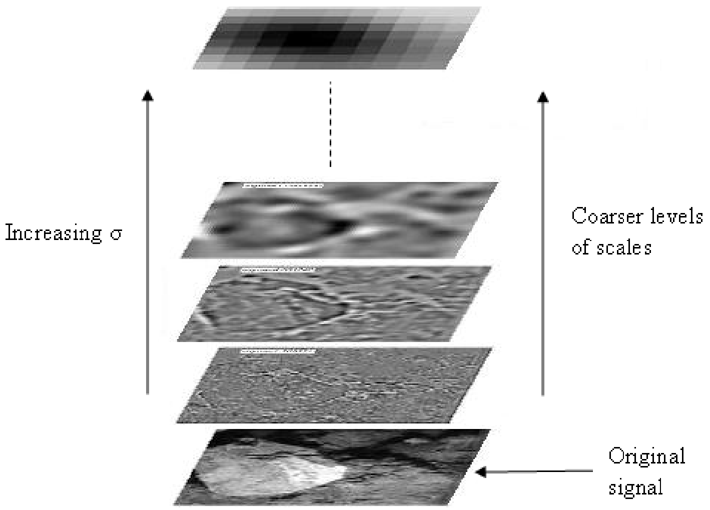


Figure 2.1: Multiscale representation of an image.

space representation. Different operators are investigated in the literature to obtain scale-space. However, the scale-spaces may be classified in two main groups: linear scale-spaces and non-linear scale-spaces.

2.2.1 Linear Scale-Spaces

The fundamental theme of the linear scale-space representation is to obtain successively higher level descriptions of a signal by convolving it with a filter. Given an image $I(x, y)$ the linear scale-space representation is obtained by convoluting $I(x, y)$ with the Gaussian kernel

$$G(x, y; \sigma) = \frac{1}{2\pi\sigma^2} e^{-(x^2+y^2)/2\sigma^2}$$

such that

$$L_I^\sigma(x, y) = G(x, y; \sigma) * I(x, y), \tag{2.1}$$

where σ is the smoothing parameter that controls the scale. A higher value of σ corresponds to coarser scale, describing the overall features of the data, while a smaller value of σ means finer scales containing the details. The Gaussian filter does not introduce any new feature points as the scale increases. Starting from fine to coarse scales, the number of features generated in scale-space either remains the same or decreases. Actually merging of adjacent features at coarser scales causes a decrease in the total number of features. The Gaussian kernel is unique in this respect for use in scale-space operator as discussed in [24, 11, 7].

2.2.2 Examples of Scale-Spaces

This section contains the brief description of different kinds of scale-spaces. However, the technical details are avoided here to make the concepts simple and easy to understand. In Chapter 3, we discuss the two scale-space approaches we have used in our application in detail.

DoG Scale-Space

Construction of Difference of Gaussians (DoG) scale-space involves the subtraction of one smoothed version of an original grayscale image from another, less smoothed version of the original. The smoothed images are obtained by convolving the original grayscale image with Gaussian kernels having differing standard deviations (scales). The difference of Gaussians is a band-pass filter that retains only some frequency components present in the original grayscale image. Smoothing an image using a Gaussian kernel subdues high-frequency spatial information. Subtracting one image from the other preserves spatial information that lies between the range of frequencies that are preserved in the two smoother version of images. Figure 2.3 shows the DoG scale-space of the example image of Figure 2.2.

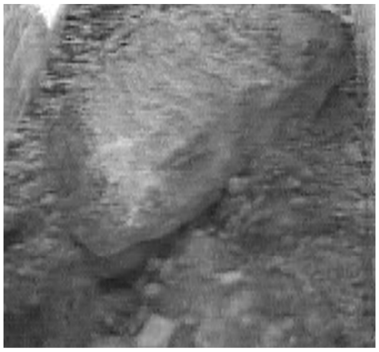


Figure 2.2: Example oil sand image

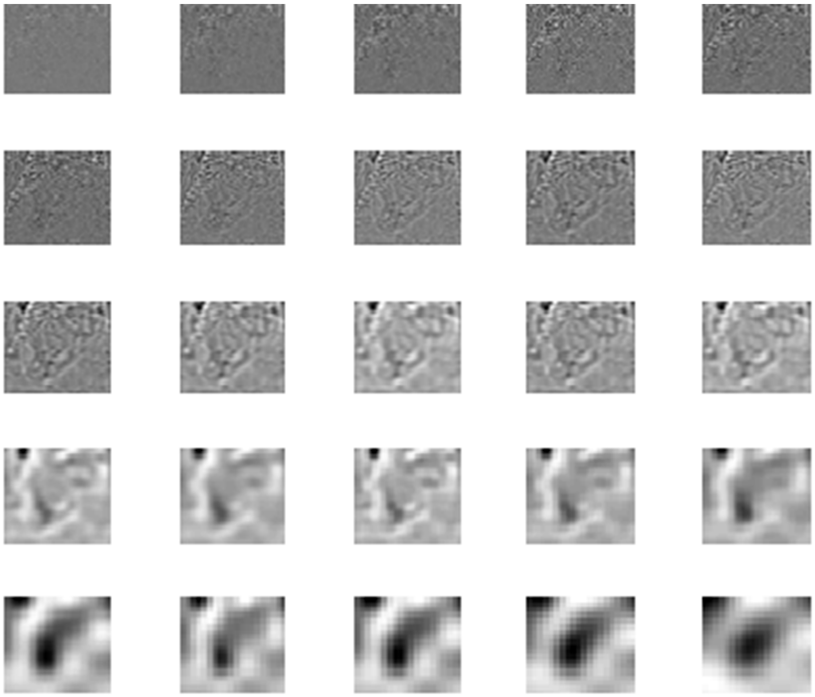


Figure 2.3: DoG filter responses for image in Figure 2.2 at different scales.

Wavelet Scale-Space

Over last fifteen years wavelet scale-space representation developed and investigated by many researchers from different fields [22]. The idea of wavelet transform was first introduced by Alfred Haar in 1990s. The wavelet transform of a signal is defined as the internal product between the signal and the mother wavelet function in a specific scale and shifted by some factor. The most important properties of wavelets are the admissibility and the regularity conditions and these are the properties which gave wavelets their name. Due to admissibility condition the frequency component of the function at frequency zero must be zero. And due to the latter condition the wavelet function must be local in both time and frequency domains. These two conditions together imply that a wavelet function must be a band pass filter. The linear scale-space is a particular case of the wavelet transform when the derivatives of the Gaussian functions are used to extract information. The relationship between the wavelets and the basic linear scale-space that were demonstrated by Mallat [22].

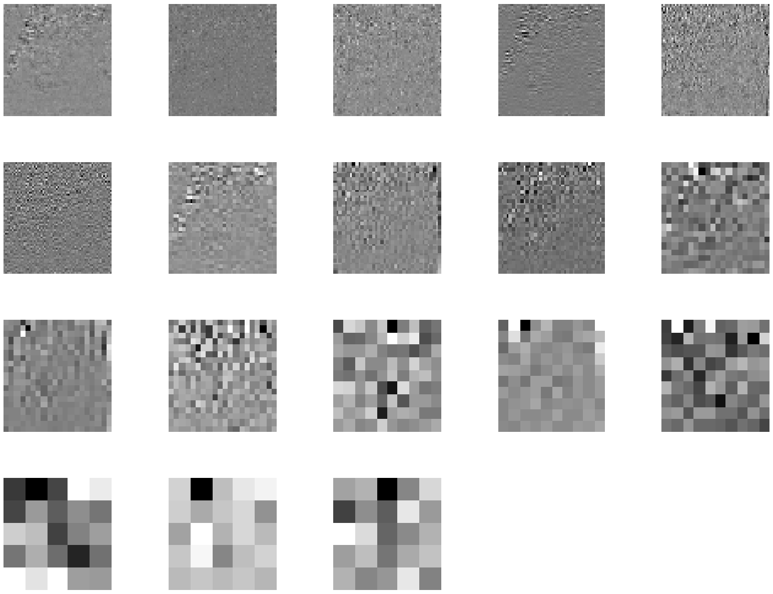


Figure 2.4: Wavelet responses for image in Figure 2.2 at different wavelength and orientation.

Gabor Scale-Space

Gabor filter is a linear filter particularly suitable for texture representation and discrimination. Frequency and orientation representations of Gabor filters are analogous to those of the human visual system. In the spatial domain, a 2D Gabor filter is a Gaussian kernel function modulated by a sinusoidal plane wave. The Gabor filters are self-similar: all filters can be generated from one mother wavelet by dilation and rotation.

Gabor scale-space is created by convolving the signal with Gabor filters of different scales and orientation . This process is closely related to processes in the primary visual cortex [25]. The Gabor scale-space is used widely for image processing applications such as optical character recognition, iris recognition and fingerprint recognition. An example of Gabor scale-space is shown in Figure 2.5.

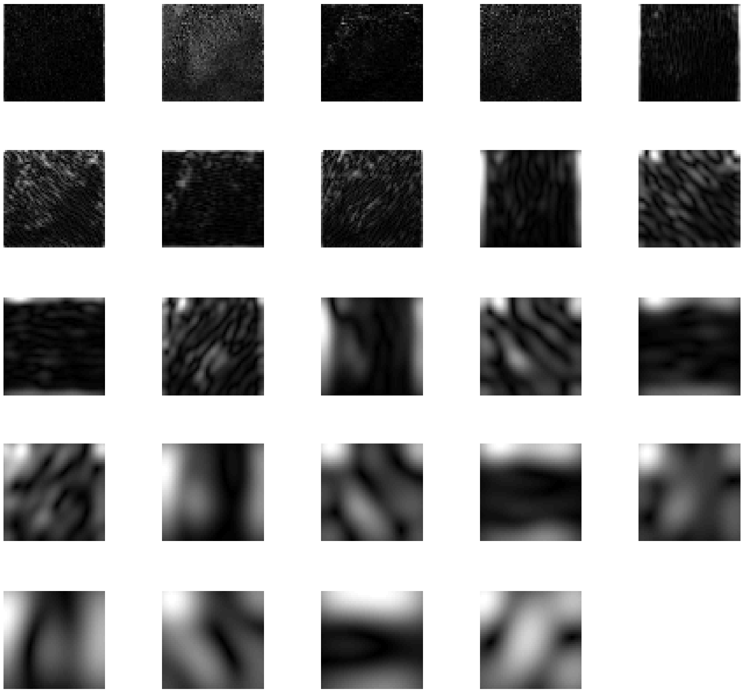


Figure 2.5: Gabor responses for image in Figure 2.2 at different scales and orientation.

Steerable Scale-Space

The Steerable scale-space is a linear multi-scale, multi-orientation image decomposition. In order to overcome the limitations of orthogonal separable wavelet decompositions, this representation was proposed in 1990s [26]. The basis functions of the steerable pyramid are directional derivative operators, that are usually designed in different sizes and orientations.

The steerable pyramid performs a polar-separable decomposition in the frequency domain, thus allowing independent representation of scale and orientation. More importantly, the representation is translation and rotation invariant which can make a big difference in applications that involve representation of position or orientation of image structure. The pyramid can be designed to produce any number of orientation bands, k. However the primary drawback of steerable filter is in computational efficiency: the steerable pyramid is substantially overcomplete. Also, the space-domain implementation of steerable filter does not provide perfect-

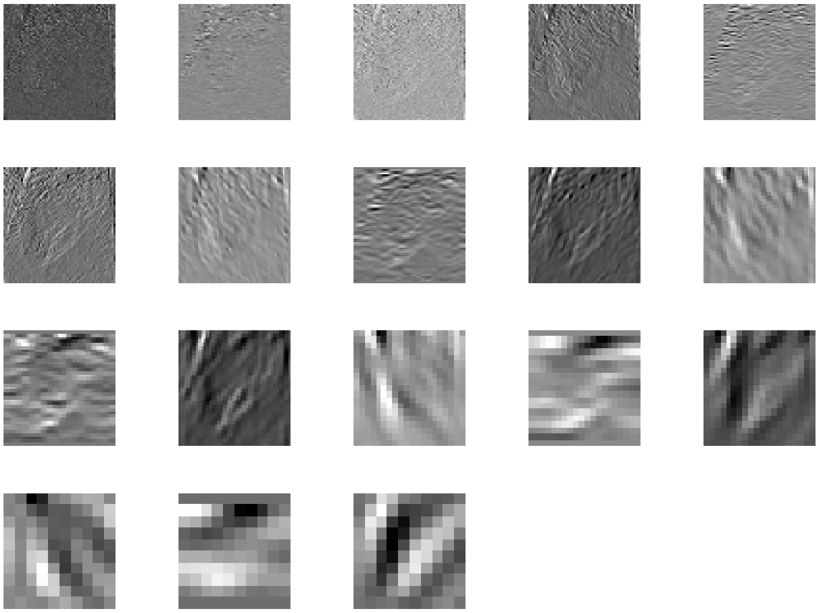


Figure 2.6: Steerable filter responses for image in Figure 2.2 at different scales and orientation.

reconstruction.

2.2.3 Non-Linear Scale-spaces

Non-linear scale-space formulations are the extensions of basic linear scale-space theory committed to specific purposes [27, 28]. Different non-linear scale-spaces are proposed in literature in connection to address different problems. Each of these non-linear scale-space representations has its own properties. To use the multi-scale representation for high-level image processing it is important to understand what kind of information is present in the decomposition and how this high level process can be benefited from this representation. The Gaussian kernel actually blurred the image region uniformly. As a result, some important regions of interest like edges can also become blurred. Furthermore, localization of the structures of interest becomes highly imprecise at larger scales. In many cases it is difficult to trace the object at a very large scale due to excess blurring and the appearance of spurious extrema in two dimensions. Various solutions have been proposed to reduce this problem. One possible solution is to use non-linear scale-spaces. Among some important non-linear scale-spaces, anisotropic diffusion proposed by Perona and Malik [28] are inhibited by high gradient values. In simple words, this method encourages smoothing within homogeneous regions in preference to smoothing across the edges. Blurring is then performed individually in each region, letting region boundaries remain sharp. Later Alvarez *et al.* [29] proposed non-linear image smoothing by mean curvature motion. Further improvement of non-linear scale-space called tensor-dependent diffusion proposed by Osher and Rudin [30]. The principle advantage of this evolution is better edge enhancement, longer edge conservation and intra-region smoothing.

Another significant non-linear scale-space is morphological scale-space [31]. In this case multi-scale system can be obtained by applying different basic morphological elements in a smarter way. The first morphological scale-space approach is associated to a continuous-scale family of openings or closings. Although standard morphological openings also displace the contours, they do not create spurious extrema. Later, an improved morphological filter called the openings and closings by reconstruction was proposed to preserve the contours. Reconstruction filters have been widely applied in the field of image enhancement, segmentation and feature detection [31, 9].

2.3 Useful Properties of Scale-Space

This section describes some properties of the scale-space representation. These properties give an idea of why the scale-space representation could be useful to vision. In other words this properties also answer the question of what abilities a visual system should possess in order to perceive the physical world around it. Here, the ideas are presented concisely rather than with technical details.

2.3.1 Simplification

It is apparent that the details are lost with the increase of the scale value. From the original data at scale σ the slices of scale-space make a transition to constant intensity at infinite scale. This transition essentially corresponds to a gradual simplification of the image content. This gradual simplification is an important property of the scale-space representation because it allows the level of details to be chosen appropriate to the image content for some application. Some of the mathematical notions of simplifications are given below.

Non-Creation of Local Extrema in One Dimension Simplification property of one dimensional scale-space was first formulated by Witkin [6]. His definition of non-creation of local extrema implies that going from small to large scales no new local extrema along space will appear as he observed that the number of zero-crossings in the second derivative of the signal decreased monotonically with scale.

Non-Enhancement of Local Extrema

The simplification property of scale-space must be characterized somewhat differently in higher dimention since here it is possible that new local extrema appear with increasing scale. A view of simplification that applies in any dimension is that all local maxima should decrease with increasing scale and conversely all local minima should increase.

Koenderink [32] first formulated the simplification property for two dimension, called causality which means that new level curves must not be created when the scale parameter is increased. In other words, it should always be possible to trace a grey-level value existing at a certain level of scale to a similar grey-level at any finer level of scale. However, the reverse statement does not need to be true. Combined with spatial homogeneity and isotropy constraints, which essentially mean that all spatial points and all scale levels are a priori equivalent and should be handled in a similar manner. It was shown that these criteria necessarily and sufficiently lead to a formulation in terms of the diffusion equation, both in one and two dimensions.

2.3.2 Translation and Rotation Invariance

Translation and rotation invariance are two indispensable properties to a visual system. For a visual system, it is important that the information content of a description remain unchanged if it is translated or rotated. This concept is expressed in terms of translation and rotation invariance as follows: translation/rotation of an image before computation of scale-space is identical to translation/rotation after construction of scale-space. Technically this can be describes as follows: Let T denotes the coordinate transformation $T(x) = Mx + a$ for some vector $a \in R^N$ and some orthonormal $N \times N$ matrix M and $f \circ T$ denotes the concatenation of T and f, i.e. $(f \circ T)(x) = f(T(x))$. Then the translation and rotation invariance of the scale-space can be verified as follows:

$$(G(:, \sigma) * (f \circ T))(x) = ((G(; , \sigma) * f) \circ T)(x) \tag{2.2}$$

where G is the gaussian kernel defined as $G(x, \sigma) = \frac{1}{\sqrt{(2\pi)\sigma}} e^{-x^2/2\sigma^2}$.

2.3.3 Differentiability

Differentiability is one of the technically useful properties of the scale-space representation. Which means $L(x; \sigma) = (G(:; \sigma) * f)(x)$ can be differentiated up to any order by the relation

$$\partial_1^{n_1} \cdots \partial_N^{n_N} L(x; \sigma) = ((\partial_1^{n_1} \cdots \partial_N^{n_N} G(:, \sigma)) * f)(x) \tag{2.3}$$

This property is widely applied in the first steps of processing the scale-space representation.

2.4 Scale Selection

The scale-space theory describe each object within an image at its appropriate scale. With increasing scale the degree of smoothing increases, as a result objects vanish from the image, small objects first and larger objects later. The degree of smoothing at which an object vanishes basically measures the size of the object. It is important to analyze the image at all scales and then select those that are “particularly informative”.

Unfortunately, scale-space theory, which is designed to describe each object at its appropriate scale, has not been dealt with the problem of scale selection for almost a decade. In fact, scale-space theory has, for a long time, focused on invariance requirements rather than the scale selection problem. These two problems are completely different in the sense that invariance requirements "conserve" information under some specified transformations of the data. On the other hand scale selection "destroys" information in the sense that the particularly informative scales, or positions and scales, of some operator response do not contain the same information as the original data. Of course the purpose of scale-selection is not to destroy information, but rather to distinguish between relevant and irrelevant scales.

A general methodology for feature detection with automatic scale selection has been proposed by Lindeberg [10, 11, 12]. The basic idea proposed by Lindeberg is to apply the feature detector at all scales, and then select scale levels from the scales at which normalized measures of feature strength assume local maxima with respect to scale. Intuitively, this approach selects the scales at which the operator response is the strongest. Several methods for scale selection are found in literature based on the entropy measures or error measures over scales [10, 12].

2.5 Multi-Scale Feature Detection

In computer vision, feature detection usually refers to the computation of local image features as intermediate results of making local decisions about the local structure in the image; Most of the feature detectors extract features at a single scale, which are determined by the internal parameters of the detector. However, the framework of multi-scale differential geometry can be applied to calculate different types of multi-scale feature detectors. This feature detector then used to produce meaningful features for further higher level application in computer vision such as object detection, object tracking, image matching and image reconstruction. Several multi-scale or multi-resolution feature detector are investigated in literature [33]. In [34] interest points are detected at the local maxima of the Harris function applied at several scales. However, multi-scale approaches cannot cope well with the case where a local image structure is present over a range of scales, which results in multiple interest points being detected at each scale within this range. As a result, there are many points, which represent the same structure, but with slightly different localization and scale. The ambiguity and the computational complexity of matching and recognition are increased with the increasing number points. Therefore, efficient methods for selecting accurate correspondences of the features among the several scales are necessary at further steps of the algorithms.

To overcome the problem of redundant detections, scale-invariant methods have been introduced. Both the location and scale of the local features are determined efficiently by this multi-scale approach. The idea for detecting local features in scale-space was first introduced by Crowley *et al.* [35]. Low pass filters are applied to construct representation and a feature point is detected if it is at a local maximum of a surrounding 3D cube (x,y and scale) and if its absolute value is higher than a certain threshold. After then several different approaches have been proposed for selecting points in scale-space. However the approaches actually differ in terms of the differential expression that is used to build the scale-space representation.

A normalized LoG function was applied in [36] and [37] to build a scale-space representation and search for scale-space extrema. Derivatives of Gaussian kernels of increasing size are applied to blur the high resolution image successively. Automatic scale selection is performed by selecting local maxima in scale-space. The LoG operator, which is circularly symmetric and is thus invariant to rotation. It is also useful for detecting bloblike structures. In [38] automatic scale selection are investigated for detecting scale invariance point. A combined framework for corner detection and blob detection with automatic scale selection were also proposed in [39] for feature tracking. Lowe [5] proposed an efficient algorithm called Scale Invarint Feature Transform (SIFT) for object recognition based on local 3D extrema in the scale-space pyramid built with DoG filters. The local 3D extrema in DoG the pyramid representation determine the localization and the scale and interest points.

Chapter 3

Support Vector Machine based Classification

In this chapter, a brief overview to the relevant background information of SVM based classifier including the formulation for standard 2-norm and 1-norm SVM, construction of kernel matrix, kernel selection are given. Based upon our literature review, a number of research groups have dedicated their efforts in developing optimized application dependent kernel. A summary to some of the representative approaches is presented as well in section 3.2. Finally discussion and literature review on different multiple kernel learning are illustrated in section 3.3

3.1 Binary Classification of Image

In binary classification, images are classified into two labeled categories on the basis of whether they have some property or not. Researchers have made great efforts in developing classification methods for improving binary image classification accuracy. The main process of binary image classification has been divided into two basic steps: (a) a preprocessing step to build a feature set and (b) classification method on the feature set to classify images. The classification processes inherently relies on feature extractor. One of the best approaches suitable for learning binary classifiers is support vector machines [40]. Other significant methods include the Bayesian networks, decision trees and neural networks [41].

3.1.1 Support Vector Machine

A comprehensive mathematical formulation of SVM can be found in [4]. We only provide here a brief description of the SVM method . Support vector machines are one of the most effective approaches for supervised learning problems. The algorithm of SVM was originated from the statistical learning theory developed by Vapnik and Chervonenkis [42, 43]. It is a hyperplane classifier, which intends to locate a hyperplane on a feature space that can separate the features belonging to different classes.

To describe SVM, let us consider a binary classification problem. Let us have a training data set $\{x_i, y_i\}$, $i = 1, ..., l$; where $y_i \in \{-1, 1\}$ represents the label of arbitrary example $x_i \in \mathbb{R}^N$; N being the dimension of input space. The equation of a linear decision surface is given by the equation

$$f(x) = w.x + b = 0 \tag{3.1}$$

The goal of learning is to find $w \in \mathbb{R}^N$ and the scalar $b \in \mathbb{R}$ such that the margin between positive and negative examples is maximized. An example of the decision surface and the margin are shown in Figure 3.1.

The parameters of equation 3.1.1 can be determined by solving the following quadratic optimization problem:

$$\begin{gathered} \text{minimize } \tfrac{1}{2} \|w\|_2^2 + C \textstyle\sum_{i=1}^{N} \xi_i \\ \text{with respect to } \boldsymbol{w} \in \mathbb{R}^N, \boldsymbol{\xi} \in \mathbb{R}_+^N, b \in \mathbb{R} \\ \text{subject to } \; y_i(\langle \boldsymbol{w}, \boldsymbol{x_i} \rangle + b) \geq 1 - \xi_i \;\; and \;\; \boldsymbol{\xi} \geq 0 \forall_i \end{gathered} \tag{3.2}$$

Here C is a predefined positive trade-off parameter between model simplicity and classification error, ξ is the vector of slack variable, and b is the bias term of the separating hyperplane.

The Lagrange dual of the primal problem 3.2 can be written as follows:

$$L_D = \frac{1}{2} \|\boldsymbol{w}\|_2^2 + C \sum_{i=1}^{N} \xi_i - \sum_{i=1}^{N} \alpha_i (y_i(\langle \boldsymbol{w}, \boldsymbol{x_i} \rangle + b) - 1 + \xi_i) - \sum_{i=1}^{N} \beta_i \xi_i \tag{3.3}$$

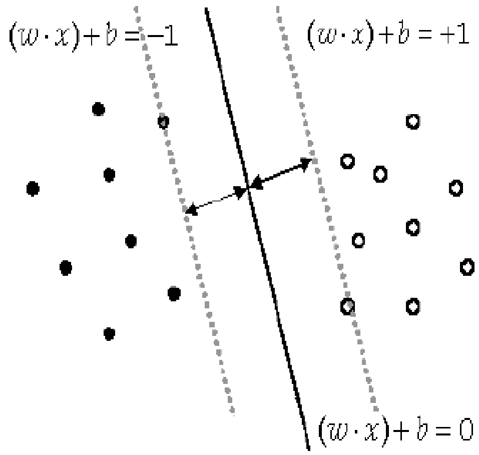


Figure 3.1: An example of decision surface and margin in SVM [4]

Now taking the derivatives of L_D with respect to the primal variables gives

$$\begin{aligned}\frac{\partial L_D}{\partial \boldsymbol{w}} &= 0 \Rightarrow \boldsymbol{w} = \sum_{i=1}^{N} \alpha_i y_i \boldsymbol{x_i} \\ \frac{\partial L_D}{\partial b} &= 0 \Rightarrow \sum_{i=1}^{N} \alpha_i y_i = 0 \\ \frac{\partial L_D}{\partial \xi_i} &= 0 \Rightarrow C = \alpha_i + \beta_i \quad \forall_i \end{aligned} \tag{3.4}$$

From 3.3 and 3.4, the dual formulation is obtained as

$$\begin{gathered}\text{maximize } \textstyle\sum_{i=1}^{N} \alpha_i - \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} \alpha_i\alpha_j y_i y_j \left\langle \boldsymbol{x_i}, \boldsymbol{x_j} \right\rangle \\ \text{with respect to } \boldsymbol{\alpha} \in \mathbb{R}_+^N \\ \text{subject to } \textstyle\sum_{i=1}^{N} \alpha_i y_i = 0 \\ C \geq \alpha_i \geq 0 \quad \forall_i \end{gathered} \tag{3.5}$$

where α is the vector of dual variables corresponding to each separation constraint. The solution of the above equation can be given by $w = \sum_{i=1}^{N_s} \alpha_i y_i x_i$ and the decision function can be written as:

$$f(x) = sgn(\sum_{i=1}^{N_s} \alpha_i y_i \langle x_i, x \rangle + b) \tag{3.6}$$

However, in most of the practical problems, the data items are distributed in a more complicated way which makes the data set non-linearly separable. To solve this problem, an SVM applies a kernel function k to map all the training data from the input space onto a higher-dimensional feature space and derives a decision boundary based upon it. Given a testing datum x, the SVM obtains its mapping onto the feature space and grants its membership to a class according to its geographical location on the feature space. The decision function then can be obtained by replacing $\langle x_i, x \rangle$ by $k(x_i, x)$ as follows:

$$f(x) = sgn(\sum_{i=1}^{N_s} \alpha_i y_i k(x_i, x) + b) \tag{3.7}$$

where k is the kernel function, α_i is the coefficient associated with a support vector x_i and N_s is the number of support vector.

3.1.2 1-norm SVM

The loss+penalty formulation of standard 2-norm SVM can be given by

$$\text{minimize} \sum_{1}^{N} [1 - y_i(< w.x_i > +b)]_+ + \lambda \left\| w \right\|_2^2 \tag{3.8}$$

Here λ is the tuning parameter that controls the tradeoff between the loss and the penalty terms. The loss $(1 - yf)_+$ is called hinge loss and penalty term is called ridge penalty. 1-norm SVM proposed by Zhu et al [44] replaces ridge penalty by lasso penalty as follows:

$$\text{minimize} \sum_{1}^{N} [1 - y_i(< w.x_i > +b)]_+ + \lambda \left\| w \right\|_1 \tag{3.9}$$

which is an equivalent Lagrange version of the optimization problem. One of the important properties of the lasso penalty is to handle sparsity. Because of L1 nature of penalty, making λ sufficiently large will cause some of the coefficients w s to be exactly zero. Thus 1 norm SVM performs a kind of continuous feature selection, while in case of standard 2-norm SVM this is not the case. In the presence of the noisy features, the performance of the 2-norm SVM is affected severely by noise.

3.1.3 Kernel Selection and Optimization

Kernel functions provide a powerful way of detecting nonlinear relationship using some linear algorithms in an appropriate feature space. Given a kernel function k and a training set $S = \{x_1, \cdots, x_l\}$, we can construct the kernel or Gram matrix K as follows:

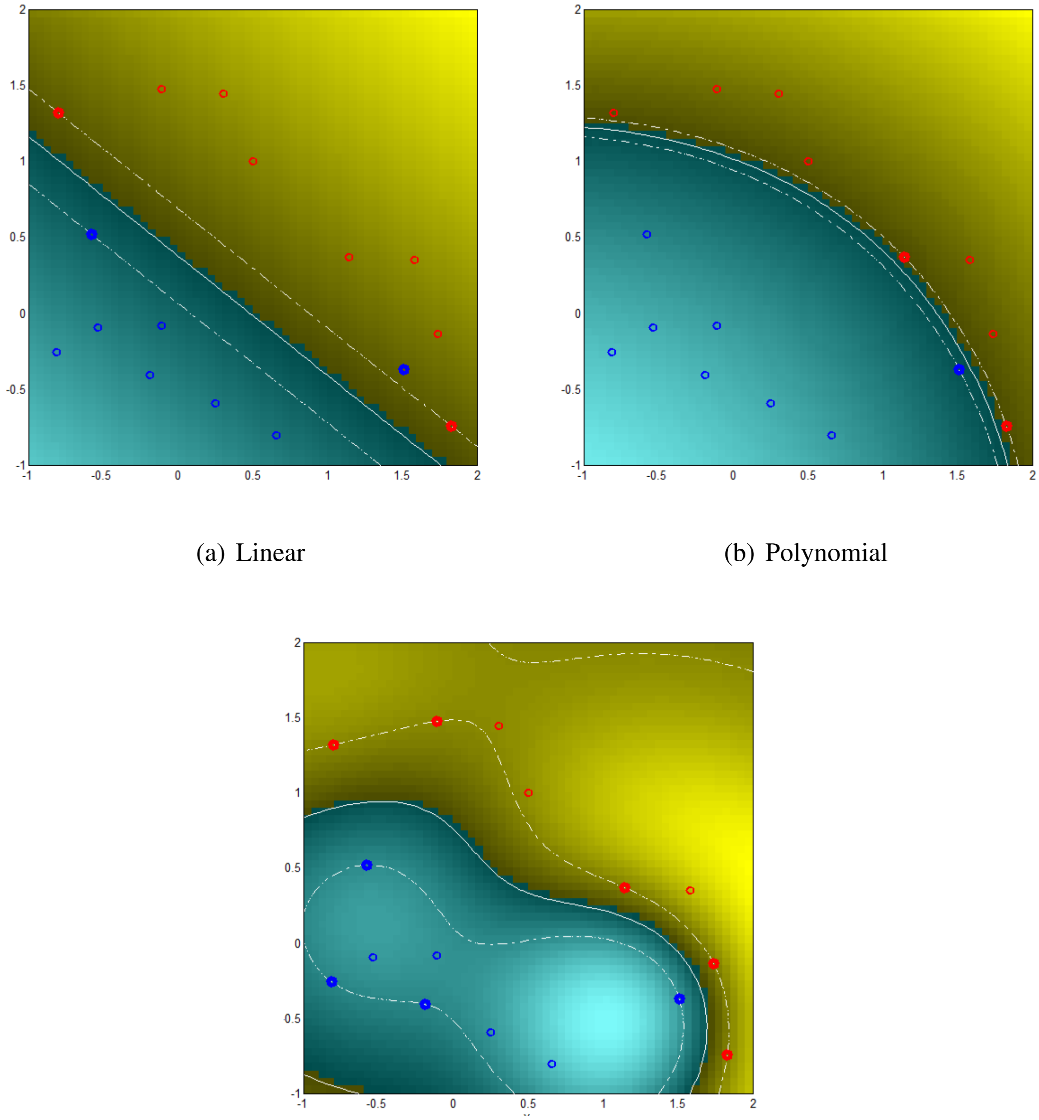
$$K_{ij} = k(x_i, x_j),\ for\ i, j = 1, \cdots, l. \tag{3.10}$$

This kernel matrix is regarded as the point-wise similarity matrix between all pairs of points in the training data set. The kernel matrix is a symmetric, positive semidifinite matrix, which completely determines the relative positions of those points in the feature space. This matrix acts as an information bottleneck as it contains all the information needed to perform the learning step. For example, learning algorithms obtain information about the choice of feature space and also about the training data only through the kernel matrix. Thus, kernel matrix acts as an interface between the data feature and the learning algorithm [40].

The choice of kernel function in kernel based methods plays an important role for solving various problems in machine learning. Data mapped to the high dimensional feature space through kernel function are expected to show better linear separability in the feature space than in the input space. However, improper choice of kernel function can make the situation worse. Therefore, careful selection of kernel function plays a very important role in kernel methods. There are several popular kernels. For example:

- Linear kernel: $k(x, x') = x^T x'$
- Polynomial kernel: $k(x, x') = (x^T x' + 1)^d$
- Gaussian radial basis (RBF) kernel: $k(x, x') = exp\left(-\frac{\|x-x'\|^2}{2\sigma^2}\right)$

Figure 3.2 shows the discriminants and support vectors found by SVM with three kernels on a toy data set using the KerMet toolbox [45]. Actually, the range of valid kernels is extremely large, some are given in closed form, others can only be computed by means of recursion or other algorithms [40]. Thus the most challenging part of applying the kernel based methods is to select the best kernels among this wide-range of possibilities. Apart from the fact that the kernel matrix should be positive definite there is almost no proposition on how to select kernels for an application. The selection of kernel can depend on our prior knowledge about the data and the types of patterns we want to identify.



(a) Linear

(b) Polynomial

(c) Gaussian

Figure 3.2: SVM solutions on a toy data set using three different kernels. The solid lines show the discriminants learned and the dashed lines show the margin boundaries. The bold circle data represent the support vectors stored

When designing a kernel function for a specific application domain, it is crucial that the kernel should incorporate as much domain knowledge as possible. Kernel optimization is a technique to design data dependent kernel by incorporating such domain knowledge. In other words, an optimized kernel function that adapts well to the input data and the learning tasks can increase the performance of the kernel based methods significantly. Several methods have been proposed for kernel optimization in the last decade. Early work on kernel optimization was limited to learning parameters of some popular kernel functions. Recently several researches encouraged optimizing the kernel function instead of optimizing the kernel parameters.

3.2 Literature Review of Kernel Optimization

A brief description of some of the notable optimization techniques from the literature are provided in this section. Kernel optimization techniques from two different points of view have been been discussed. First, different kinds of objective/cost functions adopted by different researchers are reviewed and then multiple kernel learning techniques, which are closely related to proposed MKL method are discussed .

3.2.1 Objective Function for Kernel Optimization

A good objective function should be selected in the kernel optimization technique, to evaluate the quality of chosen kernel functions for a target classification task.

Kernel Alignment

Cristianini *et al.* [46] use a measure called kernel alignment to evaluate the compliance of a kernel to the data. The alignment between two kernels K_1 and K_2 is calculated as follows:

$$A(K_1, K_2) = \frac{\langle K_1, K_2 \rangle_F}{\sqrt{\langle K_1, K_1 \rangle_F \langle K_2, K_2 \rangle_F}} \tag{3.11}$$

where $\langle K_1, K_2 \rangle_F = \sum_{i=1}^{N} \sum_{j=1}^{N} k_1(x_i^1, x_j^1) k_2(x_i^2, x_j^2)$ This similarity can be thought of as the cosine of angle between K_1 and K_2.

In [46] the alignment objective between a kernel and the ideal kernel yy^T is defined as

$$\begin{aligned} A(K, yy^T) &= \frac{\left\langle K, yy^T \right\rangle_F}{\sqrt{\langle K, K \rangle_F \langle yy^T, yy^T \rangle_F}} \\ &= \frac{\left\langle K, yy^T \right\rangle_F}{N\sqrt{\langle K, K \rangle_F}} \end{aligned} \tag{3.12}$$

is maximized where K is the kernel matrix, $y = [y_1, \cdots, y_n]$ are the training data labels, and n is the number of training data points. Lanckriet *et al.* [15] employ semidefinite programming techniques to learn a kernel matrix, which is actually a linear combination of positive semidefinite matrices. The hard margin, 1-norm soft margin and 2-norm soft margin of the support vector machine are used as cost functions. They optimized these cost functions with respect to kernel matrix in a semidefinite programming setting to determine the value of the coefficients of linear combination.

Fisher Discriminant

Xiong *et al.* [47] used Fisher discriminant criterion as an objective function to optimize the kernel function to augment the margin between different classes. Fisher scalar j for measuring the class linear separability, is defined as follows:

$$j = tr(S_b)/tr(S_w)) \tag{3.13}$$

Where S_b is the between-class scatter matrix, S_w is the within-class scatter matrix, and tr denotes the trace of a matrix. However, Fisher criterion is optimal only under the assumption that all the classes are generated from underlying multivariate normal distributions of common covariance matrix but different means and each class is expressed by a single cluster [48]. As a result this discriminant criterion is not an appropriate choice as an objective function of kernel optimization for multimodally distributed data. In order to solve this problem, Chen *et al.* [48] investigated many improved discriminant criteria (DC) such as local Fisher criteria (LFC), subclass Fisher criteria (SFC), marginal Fisher criteria (MFC), localized kernel Fisher criteria and so on. Inspired by the kernel optimization method of [47], Yeung *et al.* [49] proposed a class separability criterion as cost function that is similar to those used by linear discriminant analysis (LDA). In [47] finding optimal linear transformation corresponds to solving a generalized eigenvalue problem, which requires that the pooled within-class scatter matrix be invertible. This condition can be a problem in some application like face recognition and microarray data analysis where the dimensionality of the input space is larger than the sample size. Yeung *et al.* [49] reformulate a different optimization criterion j as: $j = Tr(S_b) - \alpha tr(S_w)$ where $\alpha > 0$ is a parameter that can be determined using some technique, for example cross-validation. Their proposed reformulation of the cost function does not require inversion of the singular within-class scatter matrix.

Other Similarity Measures

He *et al* [50] choose the distance between the combined kernel matrix and ideal kernel as the objective function to optimize instead of optimizing the kernel alignment score as follows

$$\left\|K - yy^T\right\|_F^2 \tag{3.14}$$

Nguyen *et al.* [51] proposed another cost function called feature space-based kernel matrix evaluation measure(FSM) is defined as follows

$$FSM(K, y) = \frac{s_+ + s_-}{\|m_+ - m_-\|_2} \tag{3.15}$$

where s_+, s_- are standard deviations of positive and negative classes and m_+, m_- are class centers in the feature space.

Ying *et al* [52] proposed a novel information-theoretic approach to learn the kernel combinatorial weights. They proposed to quantify the similarity between combined kernel and the optimal kernel through a Kullback-Leibler (KL) divergence or relative entropy term as follows:

$$KL(N(0, K)||N(0, yy^T)) \tag{3.16}$$

The weight of the kernel combinations are calculated by using a projected gradient decent method.

3.3 Multiple Kernel Learning (MKL)

Recent research on the kernel method based classification such as SVM has proved that using a weighted combination of multiple kernel functions can improve the interpretability of the decision function and can improve classifier performance [15]. The reasoning of MKL is very similar to combining different classifier to increase the accuracy. There are two main advantages of MKL as discussed follows:

- Different kernels provide different kinds of similarity of the data. MKL does optimum selection of the kernels instead of choosing the best one or applying cross validation. A better accuracy can be obtained by selecting multiple kernels instead of selection of a specific one.

- Different kernels may be constructed from heterogenous data sources or modalities. In this case combining kernels means combining multiple information sources.

There are several techniques available for combining multiple kernels. One promising technique to combine multiple kernel functions is proposed by Lanckriet *et al.* [15]. This article discussed the optimization over the coefficients in a linear combination of kernel functions. In such cases, the kernel $k(x, x')$ is considered as a convex linear combination of other basis kernels. The proposed weighted kernel function is given by the following equation:

$$k(x, x') = \sum_i w_i k^i(x, x') \tag{3.17}$$

where w_i represents the non-negative weighting factor for the corresponding basis kernel function k^i. By minimizing SVM dual problem with respect to w_i for all i the authors proposed to learn k. This problem can be posed as a semidefinite problem which reduces to second order cone program when w_i parameters are constrained to be nonnegative. Each basis kernel k^i may either use the full set of variables describing x or only a subset of these variables. Alternatively, kernels k^i can simply be traditional kernels (such as Gaussian or polynomial kernel) with different parameters, or may rely on different data sources associated with the same learning problem. Within this framework, the problem of data representation through the kernel is then converted to the choice of weights w_i.

The objective function $\omega(K)$ in the MKL framework proposed in [15] is defined as

$$\begin{aligned}\omega(K) = \text{maximize } & \textstyle\sum_{i=1}^{N} \alpha_i - \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} \alpha_i\alpha_j y_i y_j \langle \boldsymbol{x_i}, \boldsymbol{x_j} \rangle \\ \text{with respect to } & \boldsymbol{\alpha} \in \mathbb{R}_+^N \\ \text{subject to } & \textstyle\sum_{i=1}^{N} \alpha_i y_i = 0 \\ & C \geq \alpha_i \geq 0 \quad \forall_i \end{aligned} \tag{3.18}$$

The combined kernel matrix is selected from the following set:

$$K_L = \left\{ K : K = \sum_{m=1}^{P} \eta_m K_m, K \succeq 0, tr(K) \leq c \right\} \tag{3.19}$$

here the selected kernels should be positive semidifinite.

The resulting optimization problem that minimizes the objective function value corresponding soft margin SVM optimization problem is formulated as

$$\begin{aligned} \text{minimize } & \omega(K_\eta^{tra}) \\ \text{with respect to } & K_\eta \in K_L \\ \text{subject to } & tr(K_\eta) = c \end{aligned}$$

where K_η^{tra} is the kernel matrix calculated over only the training set and the problem can be posed as the following semidifinite programming formulation:

$$\begin{aligned} & \text{minimize } t \\ & \text{with respect to } \eta \in \mathbb{R}^P,\ t \in \mathbb{R},\ \nu \in \mathbb{R}_+^N, \delta \in \mathbb{R}_+^N \quad \text{subject to } tr(K_\eta) = c \\ & \begin{pmatrix} (yy^T)K_\eta^{tra} & e + \nu - \delta + \lambda y \\ (e + \nu - \delta + \lambda y)^T & t - 2C\delta^T e \end{pmatrix} \succeq 0 \qquad K_\eta \succeq 0. \end{aligned}$$

Later Lanckriet *et al.* reformulated the semidefinite programming formulation to the following Quadratically constrained quadratic programming (QCQP) formulation as follows:

$$\begin{gathered}
\text{minimize } \tfrac{1}{2}ct - \textstyle\sum_{i=1}^{N} \alpha_i \\
\text{with respect to } \boldsymbol{\alpha} \in \mathbb{R}_{+}^{N} \; t \in R \\
\text{subject to } tr(K_m)t \geq \alpha^T((yy)^T K_m^{tra})\alpha \;\; \forall m \\
\textstyle\sum_{i=1}^{N} \alpha_i y_i = 0 \\
C \geq \alpha_i \geq 0 \quad \forall_i
\end{gathered}$$

where the weights are restricted to have non negative values by selecting the kernel matrix from

$$K_L = \left\{ K : K = \sum_{m=1}^{P} \eta_m K_m, \eta \geq 0 \; K \succeq 0, tr(K) \leq c \right\} \tag{3.20}$$

Solving this formulation the support vector coefficients and kernel weights can be obtained jointly. The interior point methods are applied to solve this QCQP formulation.

Unfortunately, the semidefinite programming based method proposed by Lanckriet *et al.* can solve this problem only for a small number of kernels and a small number of data points and become rapidly intractable as the number of learning examples or kernels become large. To overcome this limitation Bach *et al.* [16] have reformulated the optimization problem as a Second-order cone programming (SOCP) as follows:

$$\begin{gathered}
\text{minimize } \tfrac{1}{2}\gamma^2 - \textstyle\sum_{i=1}^{N} \alpha_i \\
\text{with respect to } \boldsymbol{\alpha} \in \mathbb{R}_{+}^{N} \; \gamma \in \mathbb{R} \\
\text{subject to } \gamma^2 d_m^2 \geq \textstyle\sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_j y_i y_j k_m(x_i^m, x_j^m) \; \forall m \\
\textstyle\sum_{i=1}^{N} \alpha_i y_i = 0 \\
C \geq \alpha_i \geq 0 \quad \forall_i
\end{gathered}$$

In the above formulation, sequential minimal optimization (SMO) like techniques have been be applied to handle medium-scale problems. The optimal kernel weights can be obtained from the optimal dual variables and the weights satisfy

$\sum_{m=1}^{P} d_m^2 \eta_m = 1$. The dual problem equivalent to the Lanckriet's QCQP formulation if you consider $d_m = \sqrt{tr(K_m)/c}$. Sonnenburg *et al.* [17] again reformulate the binary classification MKL problem as a semiinfinite linear program (SILP), which can be efficiently solved using a simple LP solver and a standard SVM implementation.

$$\begin{gathered}
\text{minimize } \gamma \\
\text{with respect to } \gamma \in R, \alpha \in R_+^N \\
\text{subject to } \; \textstyle\sum_{i=1}^{N} \alpha_i y_i = 0 \\
C \geq \alpha_i \geq 0 \;\; \forall i \\
\gamma \geq \tfrac{1}{2} \textstyle\sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_j y_i y_j k_m(x_i^m, x_j^m) - \sum_{i=1}^{N} \alpha_i \;\; \forall m
\end{gathered} \tag{3.21}$$

Sonnenburg's method makes the MKL approach tractable for large-scale problems, employing existing support vector machine code iteratively. However, this method needs several iterations before converging to a reasonable solution. A more efficient approach is proposed in [53], where the authors tried to tackle the MKL problem through an adaptive 2-norm regularization formulation. Weights on each kernel matrix are included in the standard SVM empirical risk minimization problem with a constraint to allow sparsity [53]. All the aforementioned MKL techniques used linear combination of kernels which may be restrictive in some cases. Varma *et al.* [18] show how existing MKL formulations can be easily extended to learn general kernel combinations subject to regularizations on the kernel parameters. This generalized multiple kernel learning (GMKL) method provides richer representations of feature spaces by combining kernels in other fashions rather than linear combination.

Recently Gonen *et al.* [54] introduced a regularized multiple kernel learning framework that use the response surface methodology to search for the best regularization parameter set using validation data. Optimizing such regularization parameters author claimed to obtain more robust decision functions for the classification task. Kernels which are not helpful to increase the classification accuracy are pruned by selecting their regularization parameters accordingly, obtaining smoother discriminants.

Chapter 4

Multi-Scale Feature Extraction

Chapter 1 sketches an outline of the problem we are addressing and how it is intended to be solved. This chapter illustrates the multi scale feature extraction process. DoG and wavelet scale-spaces are described in detail and corresponding examples are shown.

4.1 Multi-Scale Feature Extraction

Multi-scale feature extraction is particularly useful when designing methods for automatically analyzing and deriving information from real-world measurements [10]. As discussed in Section 2.2, scale-space framework has been developed to handle the multi-scale nature of image data. In the multi-scale approach we can get a description of the signal changes at different scales by applying different sizes of scale operators on an image. In general, for a small scale operator, we get fine details of the intensity changes and the operator is more noise sensitive; for a large scale operator, we get coarse intensity change information. Tracking the behavior of some features of the signal across varying scales in multi-scale analysis unveils valuable information about the nature of the underlying physical process. For many applications, no single scale response is emphatically right. In the following two sections, construction of DoG and wavelet scale-space are discussed in detail.

4.1.1 Difference of Gaussian

Difference of Gaussians is actually an edge enhancement algorithm for grayscale images which is obtained by subtracting one Gaussian-smoothed image from another less Gaussian-smoothed image. Smoothing the image using Gaussian kernel has the excellent property of suppressing noise. Most of the edge enhancement algorithms used in digital image processing often produce the undesirable side effect of increasing random noise in the image. While the Difference of Gaussians algorithm removes high frequency detail that often includes random noise. As a result, DoG is most suitable for processing images with a high degree of noise like oil sand images. The Difference of Gaussians is a band-pass filter that retains only handful of spatial frequencies that are present in the original grayscale image. The Difference of Gaussians is very similar to the architecture of the retina's visual receptive field [55]. The retina actually implements DoG bandpass filters at several spatial frequencies. The plot of the cross sectionss of two Gaussian curves with different standard deviations and their difference is shown in Figure 4.1 and corresponding 2D difference of gaussian is drawn in Figure 4.2.

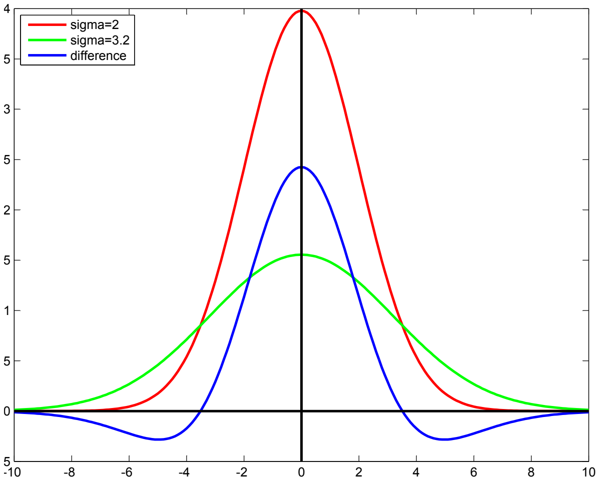


Figure 4.1: Gaussians and Difference of gaussian in one dimension.

An interesting scale-space is produced by Difference of Gaussian (DoG), which

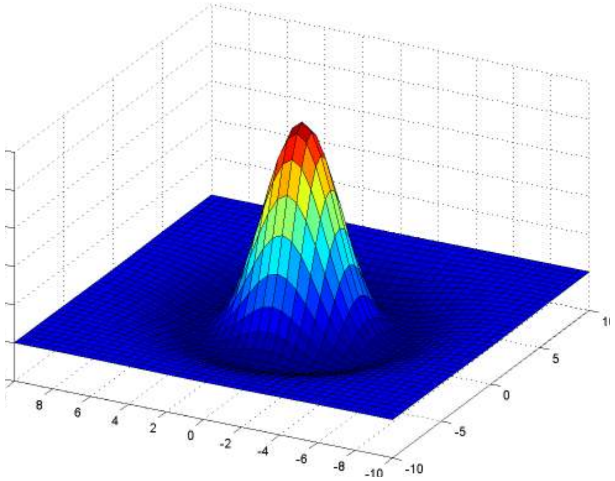


Figure 4.2: Difference of Gaussian in two dimension.

is the difference of two Gaussians with nearby scales separated by a constant multiplicative factor c [5]:

$$D_I^{\sigma}(x,y) = (G(x,y;c\sigma) - G(x,y;\sigma)) * I(x,y), \tag{4.1}$$

where G is the Gaussian kernel and I is the image. DoG filter provides a close approximation to the scale normalized Laplacian of Gaussian (LoG) filter, and it is computationally more efficient than the LoG filter [5].

The efficient approach for construction of $D_I^{\sigma}(x,y)$ as discussed in [5] is shown in Figure 4.3. The input image is successively convolved with Gaussian functions to produce a scale-space stack, where scales of the smoothed images separated by a constant factor c are shown in the left column. Each octave of scale-space are divided into an integer number, s, called sub-level, so that $c = 2^{1/s}$. Adjacent image scales are subtracted to produce the DoG images shown on the right. Once a complete octave has been processed, the Gaussian smoothed images are re-sampled that has twice the initial value of σ by taking every second pixel in each row and column. This technique is able to reduce the computation greatly. [5].

The octave index o and sub-level index s respectively are mapped to the corre-

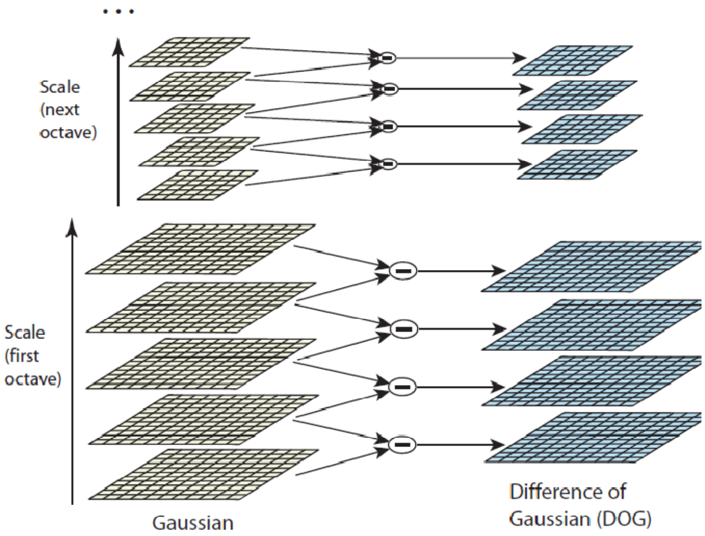


Figure 4.3: Efficient approach for calculating DoG scale-space(taken from [5]).

sponding scale σ using the following equation:

$$\sigma(o, s) = \sigma_0 2^{o+s/S}, \quad o \in O_{min} + [0, \cdots, O-1], \;\; s = [0, \cdots, S-1]. \qquad (4.2)$$

where O is the number of octaves, S is the number of sub-levels and σ_0 is the base scale level.

DoG plays an important role in blob-like object detection [5]. For the application at hand, Figure 4.5 and Figure 4.7 show the DoG scale-spaces with different scales for images in Figure 4.4 and Figure 4.6 respectively. Note that the image containing a large lump and not containing one have different filter responses specially at coarser scales. In the presence of the large lump there is a clear blob like structure which corresponds to the large lump structure in most of these scales.

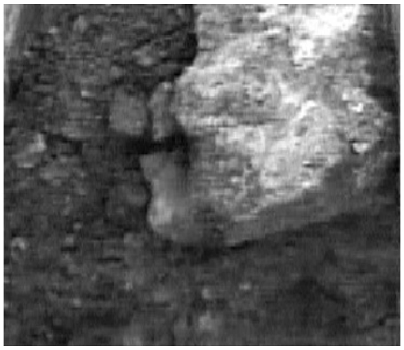


Figure 4.4: Image of large lump.

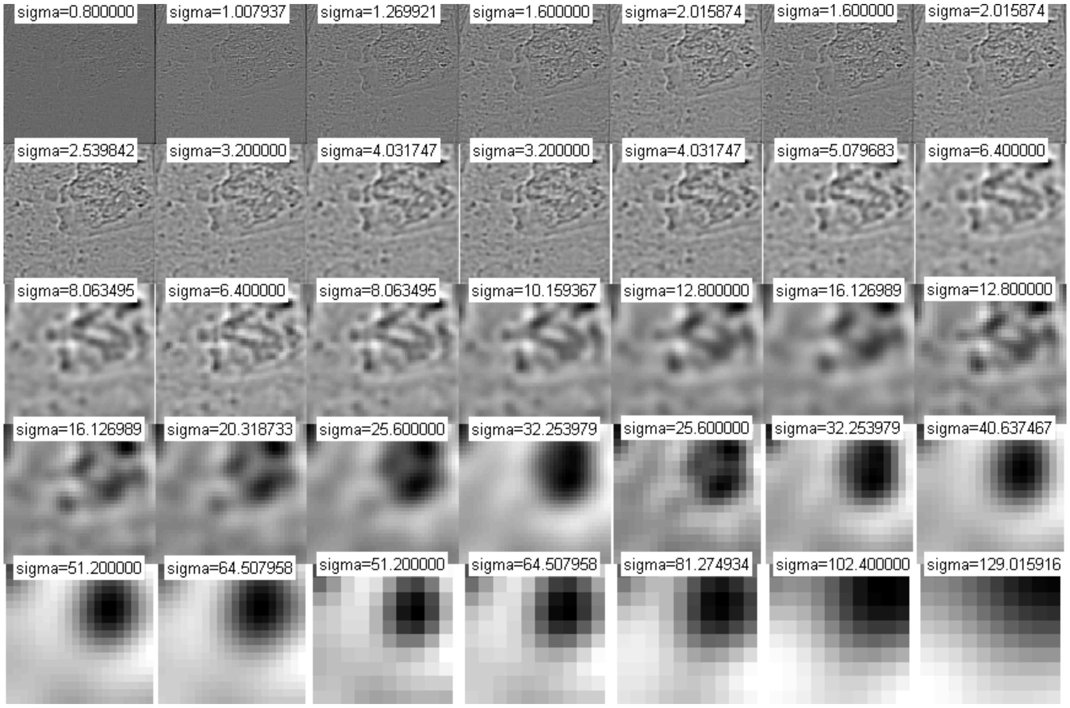


Figure 4.5: DoG responses for image in Figure 4.4.

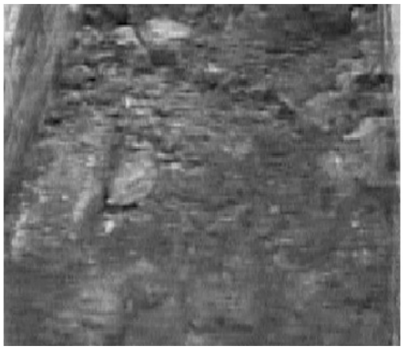


Figure 4.6: Image with no large lump.

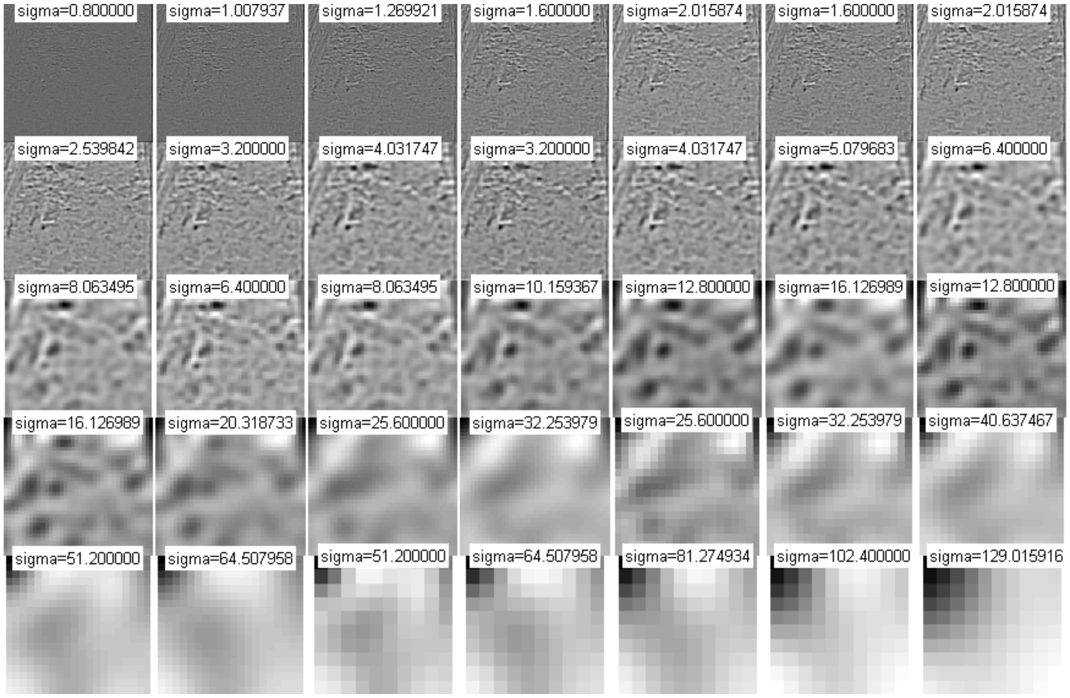


Figure 4.7: DoG responses for image in Figure 4.6.

4.1.2 Wavelet Scale-Space

Multi scale analysis is basic to wavelet analysis. Analogous to human vision system, discrete wavelet transform (DWT) decomposes images into different frequency subbands which make it efficient for the processing and classification of images. The wavelet transform encompasses a variety of unique but related transformations which are related by their expansion functions called “wavelets”, of varying frequency and limited duration. The wavelet kernel can be represented by three separable 2-D wavelets

$$\begin{aligned}\psi^H(x,y) &= \psi(x)\phi(y)\\ \psi^V(x,y) &= \phi(x)\psi(y)\\ \psi^D(x,y) &= \psi(x)\psi(y)\end{aligned} \tag{4.3}$$

where $\psi^H(x,y)$, $\psi^V(x,y)$ and $\psi^D(x,y)$ are called horizontal, vertical and diagonal wavelets, respectively and one separable 2-D scaling function

$$\phi(x,y) = \phi(x)\phi(y) \tag{4.4}$$

Both $\phi(x)$ and $\psi(x)$ can be expressed as linear combinations of double resolution copies of themselves via the series expansions as follows:

$$\begin{aligned}\phi(x) &= \sum_n h_\phi(n)\sqrt{2}\phi(2x-n)\\ \psi(x) &= \sum_n h_\psi(n)\sqrt{2}\phi(2x-n)\end{aligned} \tag{4.5}$$

where h_ϕ and h_ψ - the expansion coefficients of fast wavelet transform (FWT) are called scaling and wavelet vectors respectively. The iterative approach of FWT are shown in Figure 4.8

The $W_\phi(j,m,n)$ and $\{W_\psi^i(j,m,n) \;\; for \; i = H, V, D\}$ are the DWT coefficient at scale j. The input image is downsampled by 2 by extracting every other point from a sequence of points. The $h_\phi(-n)$ and $h_\psi(-m)$ are lowpass and highpass decomposition filters are applied to decompose it into four lower resolution subbands. The W_ϕ coefficients are created via two lowpass filters and are thus called approximation coefficients and $\{W_\psi^i \;\; for \; i = H, V, D\}$ are horizontal, vertical and diagonal detail coefficients.

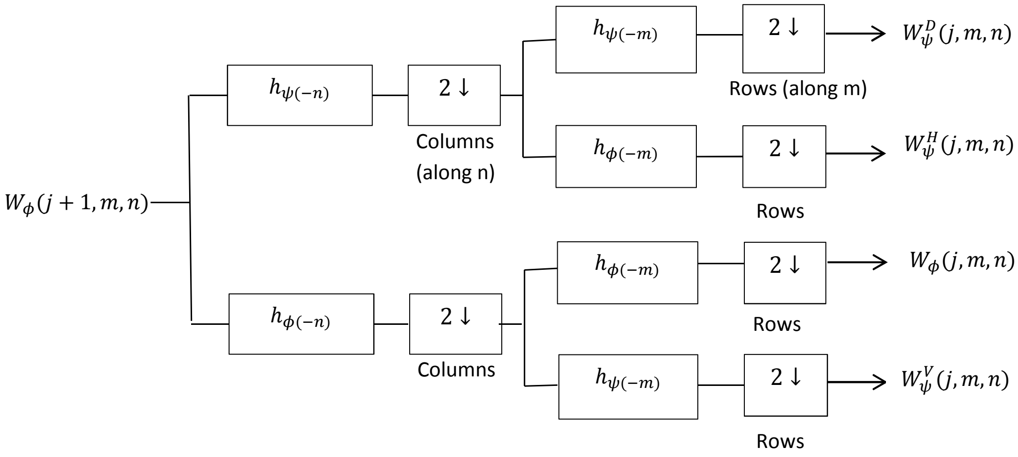


Figure 4.8: A two dimensional four band filter bank.

In this context, the wavelet transform can be interpreted as frequency decomposition, with each set having a spatial orientation. To handle steam detection problem via supervised classification, we have used wavelet subbands at different scales. Figure 4.10 and Figure 4.12 show the wavelet coefficients for image frame without steam and with steam.

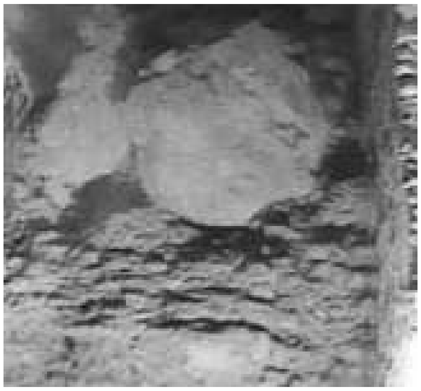


Figure 4.9: Image frame without steam.

With this large number of subbands, appropriate features need to be extracted to obtain a representation that is as discriminative as possible in the transform domain. In several previous researches, especially on texture image classifications such as

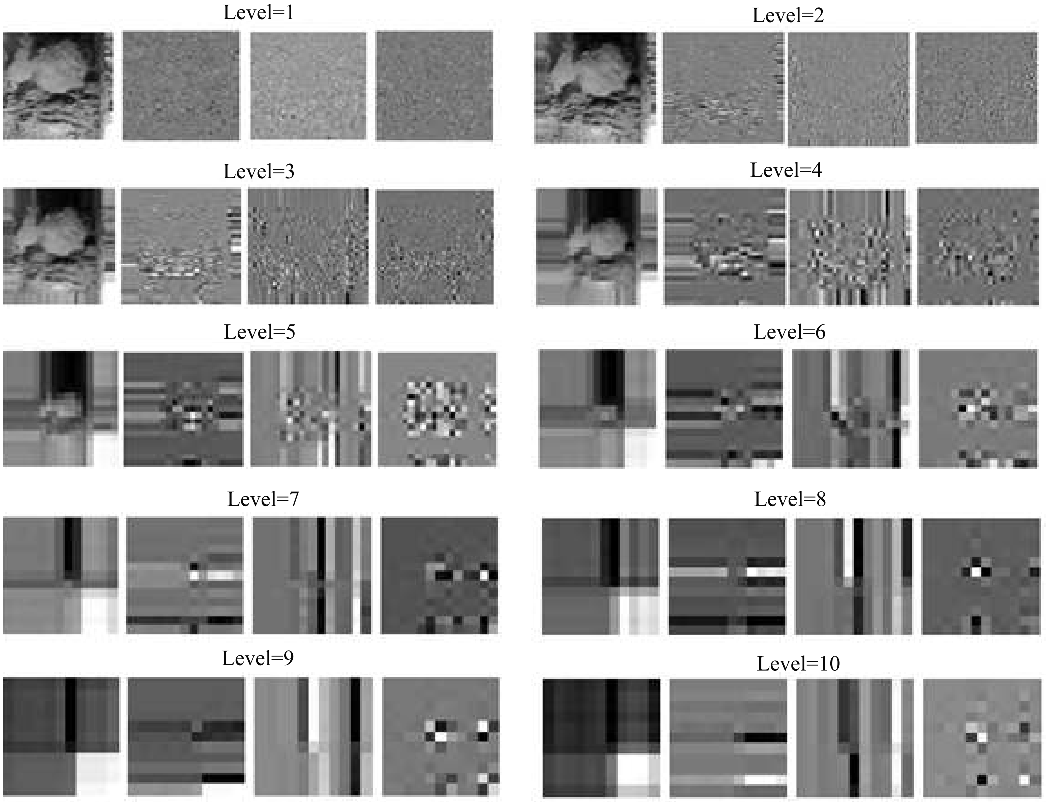


Figure 4.10: Wavelet responses at different decomposition levels for the image containing no steam of Figure 4.9.

in [56] and [57], features from all wavelet subbands are used. However, it is well understood that proper feature selection is likely to improve the classification accuracy with fewer numbers of features [58]. Recently a wavelet feature selection algorithm based on statistical dependence proposed in [59] showed significant improvement in case of image classification. Wavelet feature selection is equivalent to selecting a set of subbands for image decomposition. Therefore, wavelet feature selection and wavelet subband selection are interchangeably used in this thesis.

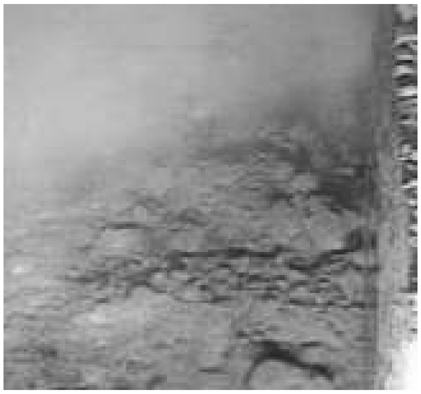


Figure 4.11: Image frame with steam.

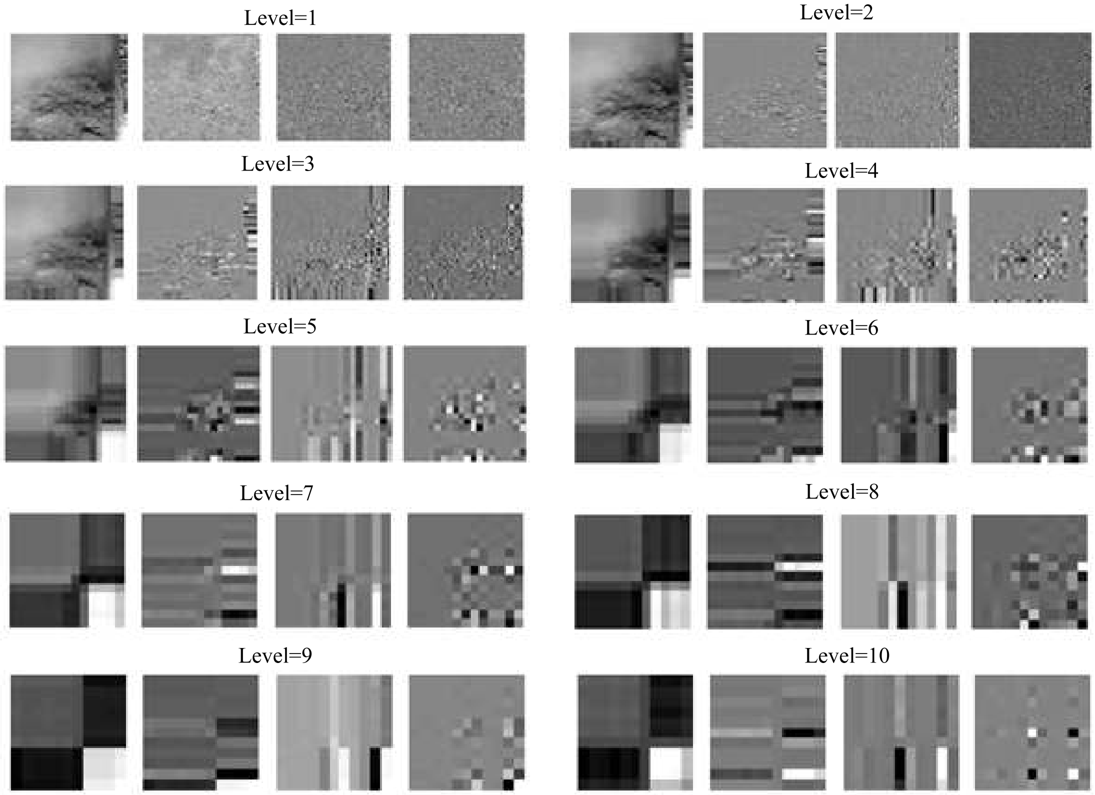


Figure 4.12: Wavelet responses at different decomposition levels for the image containing steam of Figure 4.11.

Chapter 5

Multiple Kernel Learning for Scale Selection

Various methods are proposed in the literature to combine information from multiple scales to solve different types of problems [18]. However multi-scale systems have some limitations: they provide no information on how to relate the descriptions of different scales, or which scales to select under what condition. The ambiguity introduced by multi-scale system is inherent and unavoidable. Thus the goal of scale dependent representation is to manage and combine the multi scale responses while reducing the ambiguity where possible. The multi-scale system involves two aspects:

- how to handle the large dimensionality of multi scale feature
- how to select the scales and combine them effectively

In this thesis we solve the above problems via multiple kernel learning approach.

5.1 Proposed Basis Kernel Function Construction

After constructing the scale-space images from the input images, the next step is to consider a basis kernel function defined on an individual scale. The choice of kernel function in kernel-based methods plays an important role. In considering such a kernel function, inclusion of prior knowledge about possible variations of the patterns can play a significant role to design a robust support vector machine (SVM) classifiers. Shift invariance is important for our application, because the object of interest that is large lumps/steam can appear anywhere in the image frame and yet the kernel function needs to find out similarity among all these cases. Applying the standard kernels of SVM, such as linear, Gaussian or polynomial kernels, are not always appropriate to be used with scale-space features. One principal reason is that these kernels are not shift invariant. In [60], we propose a base kernel function defined with circular convolution, which is shift invariant. The circular convolution between two scale-space images D_I^σ and D_J^σ of size $M \times N$ obtained from images I and J can be computed as follows:

$$(D_I^\sigma \otimes_c D_J^\sigma)(i,j) = \sum_{u=1}^{M}\sum_{v=1}^{N} D_I^\sigma(i-u, j-v) D_J^\sigma(u,v) \tag{5.1}$$

Thus we can define the base kernel function for scale σ as

$$k^\sigma(I,J) = \sum_{i=1}^{M}\sum_{j=1}^{N} \left(\left(D_I^\sigma \otimes_c D_J^\sigma\right)(i,j)\right)^2 \tag{5.2}$$

and call this kernel function as circular convolution kernel. Note that this is equivalent to point-wise multiplication of the Fourier transform coefficients of D_I^σ and D_J^σ and sum of their squares. Thus, this kernel not only takes into account the shift invariance, it also compares frequency bands produced in the scale-space features. It can be shown that circular convolution kernel function is indeed a kernel function i.e., symmetric and positive semi-definite.

Similarly, if $U = (I_1, I_2, \ldots, I_T)$ and $V = (J_1, J_2, \ldots, J_T)$ are two video sequences of the same length, then the circular convolution kernel function can be extended between them as

$$k^{t,\sigma}(U,V) = k^\sigma(I_t, J_t) \tag{5.3}$$

for $t = 1, 2, \ldots, T$.

5.1.1 Proof of Symmetric and Positive Semi-Definiteness

In this section we prove that the function given by (5.2) is a kernel function i.e. circular convolution kernel function is symmetric and positive semi-definite. To prove this, we need to define a kernel matrix $K_{p,q}$, which is defined for a set of images $I_1, I_2, ...$, and a kernel function k as:

$$K_{p,q} = k(I_p, J_q),\ \forall\ i, j = 1, 2, \cdots, . \tag{5.4}$$

To prove k is a kernel function, we need to show that the associated kernel matrix K is symmetric and positive semi-definitive. Toward this goal, we prove the following Lemma.

Lemma 1: If A and B are two doubly block circulant matrices, then $A^T B = B A^T$.

Proof: We first prove the result for circulant matrices. Let X and Y be two circulant matrices:

$$X = \begin{bmatrix} x_0 & x_{N-1} & \cdots & x_1 \\ x_1 & x_0 & \cdots & x_2 \\ \vdots & \vdots & \vdots & \vdots \\ x_{N-1} & x_{N-2} & \cdots & x_0 \end{bmatrix}$$

and

$$Y = \begin{bmatrix} y_0 & y_{N-1} & \cdots & y_1 \\ y_1 & y_0 & \cdots & y_2 \\ \vdots & \vdots & \vdots & \vdots \\ y_{N-1} & y_{N-2} & \cdots & y_0 \end{bmatrix}$$

Then, the $(k, l)^{th}$ element of matrix $X^T Y$ is

$$\begin{aligned} (X^T Y)_{k,m} &= \sum_{l=0}^{N-1} (X^T)_{k,l} (Y)_{l,m} = \sum_{l=0}^{N-1} x_{l-k} y_{l-m} \\ &= \sum_{l=0}^{N-1} x_{l-m+m-k} y_{l-m} \\ &= \sum_{p=0}^{N-1} x_{p+m-k} y_p \end{aligned}$$

On the other hand

$$\begin{aligned} (Y X^T)_{k,m} &= \sum_{l=0}^{N-1} (Y)_{k,l} (X)^T_{l,m} = \sum_{l=0}^{N-1} y_{k-l} x_{m-l} \\ &= \sum_{l=0}^{N-1} y_{k-l} x_{m-k+k-l} \\ &= \sum_{p=0}^{N-1} y_p x_{p+m-k} \end{aligned}$$

So, $X^TY = YX^T$.

We will now extend the result to the doubly block circulant matrices. Note that a doubly block circulant matrix has a block structure, which is circulant; moreover each block is a circulant matrix. For two such matrices A and B, let us consider $(k, m)^{th}$ block of A^TB:

$$\begin{aligned}
[A^TB]_{k,m} &= \sum_l [A^T]_{k,l}[B]_{l,m} = \sum_l [A]^T_{l-k}[B]_{l-m} \\
&= \sum_p [A]^T_{p+m-k}[B]_p
\end{aligned}$$

On the other hand,

$$\begin{aligned}
[BA^T]_{k,m} &= \sum_l [B]_{k,l}[A^T]_{l,m} = \sum_l [B]_{k-l}[A]^T_{m-l} \\
&= \sum_l [A]^T_{m-l}[B]_{k-l} = \sum_p [A]^T_{p+m-k}[B]_p :
\end{aligned}$$

Therefore, $A^TB = BA^T$

Poposition 1: The function given in (5.2) is a kenel function.

Proof: In order for a function to be a kernel function, it needs to be (a) symmetric and (b) positive semidefinite. Symmetry is satisfied for function (5.2) because circular convolution is commutative. It remains to show if function (5.2) is positive semidefinite. To show this let F and G be two matrices of size $N \times M$. Thus circular convolution between them is defined as:

$$\begin{aligned}
(F \otimes_c G)_{i,j} = \textstyle\sum_{u=0}^{N-1} \sum_{v=0}^{M-1} F_{i-u,j-v} G_{u,v} \\
&= [F_{i,j}\, F_{i,j-1} \cdots F_{i,j-M+1} \cdots F_{i-N+1,j} \cdots F_{i-N+1,j-M+1}] \\
&\quad [G_{0,0}\, G_{0,1} \cdots G_{0,M-1} \cdots G_{N-1,0} \cdots G_{N-1,M-1}]^T
\end{aligned}$$

In a similar way,

$$
\begin{aligned}
(F \otimes_c G)_{i,j} = \textstyle\sum_{u=0}^{N-1} \sum_{v=0}^{M-1} G_{i-u,j-v} F_{u,v} \\
&= [G_{i,j}\, G_{i,j-1} \cdots G_{i,j-M+1} \cdots G_{i-N+1,j} \cdots G_{i-N+1,j-M+1}] \\
&\quad [F_{0,0}\, F_{0,1} \cdots F_{0,M-1} \cdots F_{N-1,0} \cdots F_{N-1,M-1}]^T
\end{aligned}
$$

Let $F_1, F_2, \cdots$ are 2D matices for which function (5.2) is computed pairwise. Let K denote the kernel matix. $(p,q)^{th}$ element of K is given by
where the notation $\bar{F}$ stands fo the following column vector of size $MN \times 1$
$\bar{F} = [F_{0,0}\, F_{0,1} \cdots F_{0,M-1} \cdots F_{N-1,0}\, F_{N-1,1} \cdots F_{N-1,M-1}]^T$ $\tilde{F}$ is a doubly block circulant matrix of size $MN \times MN$

$$
\tilde{F} = \begin{bmatrix}
F_{0,0} & F_{0,M-1} & \cdots & F_{0,1} & \cdots & F_{1,0} & F_{1,M-1} & \cdots & F_{1,1} \\
F_{0,1} & F_{0,0} & \cdots & F_{0,2} & \cdots & F_{1,1} & F_{1,0} & \cdots & F_{1,2} \\
\vdots & \vdots & \vdots & \vdots & & & & & \\
F_{N-1,M-1} & F_{N-1,M-2} & \cdots & F_{N-1,0} & \cdots & F_{0,M-1} & F_{0,M-2} & \cdots & F_{0,0}
\end{bmatrix}
$$

Thus using *lemma 1*

$$
\begin{aligned}
K_{p,q} &= \bar{F_q}^T \tilde{F_p}^T \tilde{F_q} \bar{F_p} \\
&= \bar{F_q}^T \tilde{F_q} \tilde{F_p}^T \bar{F_p} = (\tilde{F_q}^T \bar{F_q})^T (\tilde{F_p}^T \bar{F_p})
\end{aligned}
$$

Thus K can be written as $K = LL^T$ for some matrix L. Hence K is a positive semidefinite symmetric matrix.

5.1.2 Proposed Multiple Kernel Learning

Instead of choosing any particular single scale-based kernel, we propose to use a data dependent kernel function, which is a weighted linear combination of base kernel functions $(k^{t,\sigma})$ defined on individual video frames and scales. For the large lump detection problem, the reason for considering multiple kernel functions can be justified from a frequency domain perspective that an object can span over more than a single scale in the scale space. We can consider the convex combination of base kernel functions:

$$
k(U,V) = w_{0,0} + \sum_t \sum_\sigma w_{t,\sigma} k^{t,\sigma}(U,V), \tag{5.5}
$$

Each kernel $k^{t,\sigma}$ is the base kernel function for video frame t and scale σ and w is the weight for the same frame and scale. The goal of MKL is to obtain an optimized combination of these coefficients. There is a straightforward interpretation of Equation (5.5) as follows. If the base kernel functions $k^{t,\sigma}$ are considered features, then k in (5.5) is a prediction function with linear combination of non-negative weights that need to be learned from the data. The standard machinery of large margin hyper plane classifier, viz., SVM can be applied on (5.5) with slight modifications to yield non-negative weights. If a linear program is applied for the 1-norm SVM then the only extra constraints are non-negativity. Alternatively, one can use Mangasarian's unconstrained minimization [61]. The 1-norm SVM has the advantage of a sparse solution [44]. In our case, the sparse solution ensures that only a handful of the coefficients $w_{t,\sigma}$'s will be non-zero and a few convolutions corresponding to these scales need to be evaluated at the runtime.

Once the base kernel weights $(w_{t,\sigma})$ are learned, the next step is to classify a test video. We have applied a non-linear 2-norm SVM to classify an video sequence V:

$$f(V) = \alpha_0 + \sum_i \alpha_i y_i k(U_i, V), \tag{5.6}$$

where $\{U_1, U_2, \cdots\}$ are support vectors (training video clips) with weights $\{\alpha_1, \alpha_2, \cdots\}$ and corresponding labels $\{y_1, y_2, \cdots\}$. We assume $y_i = 1$ implies a large lump, and $y_i = -1$ implies non-large lump. Here, α_0 is the bias. The prediction label for an unknown video sequence V is obtained as the sign of $f(V)$. In our experiments, each training video clip consists five image frames. For a test video stream $(J_1, J_2, \dots)$, if the current time is denoted by t, then the test video clip V for the current time t would consist of five frames: $V = (J_{t-2}, J_{t-1}, J_t, J_{t+1}, J_{t+2})$. We learn the two classifiers (5.5) and (5.6) in a cascaded way. Notice that because $\sum_i \alpha_i y_i = 0$ in a standard 2-norm SVM, the bias term $w_{0,0}$ in (5.5) is not required in (5.6). Below we provide the proposed MKL algorithm.

Proposed MKL Learning Algorithm

Inputs:

Training video set $\{U_1, U_2, U_3, \cdots, U_l\}$ and hold-out test video set $\{V_1, V_2, V_3, \cdots, V_p\}$

Outputs:

1. kernel weights $w_{t,\sigma}$ for $t = 1, 2, \cdots, T$ and $\sigma = \sigma_1, \sigma_2, \cdots, \sigma_N$
2. SVM weights $\alpha_0, \alpha_1, \alpha_2, \cdots, \alpha_l$

Perform the following three steps (a) through (c) in sequence:

(a) From the training set, obtain all possible pairing of video sequences of the form (U_i, U_j). With these paired videos compute the non-negative weights $(w_{t,\sigma})$ for (5.5) using linear 1-norm SVM.

(b) Using the training set, compute the support vector weights $\alpha_0, \alpha_1, \alpha_2, \cdots, \alpha_l$ in (5.6) with a standard 2-norm nonlinear support vector machine with mixture kernel function k in (5.5).

(c) Obtain classification accuracy on the hold-out test set using (5.5).

1-norm SVM and 2-norm SVM each has one tuning parameter. Thus, the proposed MKL learning algorithm is run on all possible combinations of these two tuning parameters. The combination that yields the highest classification accuracy is considered and the associated learned parameters $\alpha_0, \alpha_1, \alpha_2, \cdots, \alpha_l$ and $w_{t,\sigma}$'s are retained. Our algorithm requires only a linear equation solver or alternatively the method proposed in [61] making our MKL implementation simple, fast, and easily accessible.

In the proposed algorithm we have done pairing of the training samples. As a result, training space is artificially inflated from n to n^2. Also the study of the generalization properties of this case is a challenging task, since the pairs of samples violate the central i.i.d. assumption of binary classification. Usunier *et al.* [62] show that classifiers trained on interdependent data will "inherit" the generalization bound of the same classifier trained on i.i.d. In [62], a new framework is proposed to study the generalization properties of classifiers over data which can exhibit a suitable dependency structure. Bipartite ranking problem is considered as special case of the classification problem where pairing of the training set are done. The generalization bounds for classifiers trained over interdependent examples are inferred using generalization results known for binary classification. This property is illustrated by proving a new margin-based, data-dependent bound for SVM-like algorithms optimizing the area under the ROC Curve (AUC).

It is natural to ask if our proposed margin function $yy'k(\mathbf{x}, \mathbf{x}')$ has a concentration around an empirical estimation from paired sample points belonging to an independent and identically distributed (i.i.d.) set of patterns and their responses $S = \{(x_1, y_1), \cdots, (x_n, y_n)\}$. To derive a concentration bound, note that

$$yy'k(\mathbf{x}, \mathbf{x}') = yy'\phi(\mathbf{x})^T\phi(\mathbf{x}') = \rho(\mathbf{z})^T\rho(\mathbf{z}'), \tag{5.7}$$

where the kernel function k can be expressed as a inner product of some feature map ϕ and we denoted $\mathbf{z} = (\mathbf{x}, y)$ as a pattern $(\mathbf{x})$ and corresponding response (y) together, further, we note that

$$\rho(\mathbf{z}) = y\phi(\mathbf{x}) \quad and \quad \rho(\mathbf{z}') = y'\phi(\mathbf{x}'). \tag{5.8}$$

With this setup, the expectation of our proposed margin function is as follows:

$$E[yy'k(\mathbf{x}\mathbf{x}')] = E[\rho(\mathbf{z})^T\rho(\mathbf{z}')] = E[\rho(\mathbf{z})]^T E[\rho(\mathbf{z}')] = E[\rho(\mathbf{z})]^T E[\rho(\mathbf{z})] = \|E[\rho(\mathbf{z})]\|^2$$

The second equality holds because $\mathbf{z}$ and $\mathbf{z}'$ are independent. The third equality holds because $\mathbf{z}$ and $\mathbf{z}'$ are identically distributed.

The empirical expectation estimated from the set $S = \{\mathbf{z_1}, \cdots, \mathbf{z_n}\}$ with paired sample points $(\mathbf{z_i}, \mathbf{z_j})$ is as follows:

$$\widehat{E}[yy'k(\mathbf{x}, \mathbf{x}')] = \frac{2}{n(n-1)}\sum_{i>j} y_i y_j k(\mathbf{x_i}, \mathbf{x_j}) = \frac{2}{n(n-1)}\sum_{i>j} \rho(\mathbf{z}_i)^T\rho(\mathbf{z}_j) \tag{5.9}$$

Let us now define a function $g(S)$ on the set of i.i.d. sample S as follows:

$$g(S) = \frac{2}{n(n-1)}\sum_{i>j} \rho(\mathbf{z}_i)^T\rho(\mathbf{z}_j) - \|E[\rho(\mathbf{z})]\|^2 \tag{5.10}$$

Thus, it is our aim now to find a bound on $|g(S)|$. To do so, let us consider another i.i.d. set of sample points $\tilde{S} = \{\mathbf{z}_1, \mathbf{z}_2, \cdots, \mathbf{z}_{i-1}, \mathbf{z}'_i, \mathbf{z}_{i-1}, \cdots, \mathbf{z}_n\}$, where only the

i^{th} sample point is different between the sets S and $\tilde{S}$. Now, we find a bound on the following:

$$\left|g(S) - g(\tilde{S})\right| = \frac{2}{n(n-1)}\left|(\rho(\mathbf{z}_i) - \rho(\mathbf{z'}_i))^T \sum_{k \neq i} \rho(\mathbf{z}_k)\right| \leq \frac{4R}{n}, \tag{5.11}$$

assuming the kernel function has a bound $|k(\mathbf{x}, \mathbf{x}')| \leq R$, a standard assumption (see for example the treatments in [40]). Because S is an i.i.d. sample and the function $g(S)$ has a bounded variation for each sample point $\mathbf{z_i}$, we can apply McDiarmid's inequality [40] and obtain:

$$P\{g(S) - E[g(S)] \geq \epsilon\} \leq exp(-\frac{n\epsilon^2}{8R^2}), \tag{5.12}$$

where ϵ is any given positive number. Further, we note that

$$\begin{aligned} E[g(S)] &= E[\frac{2}{n(n-1)} \sum_{i>j} \rho(\mathbf{z}_i)^T \rho(\mathbf{z}_j)] - \|E\rho(\mathbf{z})\|^2 \\ &= \frac{2}{n(n-1)} \sum_{i>j} E[\rho(\mathbf{z}_i)^T] E[\rho(\mathbf{z}_j)] - \|E\rho(\mathbf{z})\|^2 \\ &= 0. \end{aligned} \tag{5.13}$$

The significance of $E[g(S)]$ being zero is that our empirical estimate using pairwise sample points is unbiased. Thus, we obtain:

$$P\{g(S) \geq \epsilon\} \leq exp(-\frac{n\epsilon^2}{8R^2}). \tag{5.14}$$

However, reversing the sign of g(S) and following practically the similar derivation as above, we also obtain:

$$P\{-g(S) \geq \epsilon\} \leq exp(-\frac{n\epsilon^2}{8R^2}). \tag{5.15}$$

Adding these two inequalities, we obtain:

$$P\{|g(S)| \geq \epsilon\} \leq 2exp(-\frac{n\epsilon^2}{8R^2}). \tag{5.16}$$

Setting the right hand side to δ and solving for $\epsilon = \sqrt{\frac{8R^2}{n} ln(\frac{2}{\delta})}$, we are able to assert that with probability $1 - \delta$, the following bound holds:

$$|g(s)| \leq \sqrt{\frac{8R^2}{n} ln(\frac{2}{\delta})}, \tag{5.17}$$

$$i.e, \quad \left| E\left[yy'k(x,x')\right] - \hat{E}\left[yy'k(x,x')\right] \right| \leq \sqrt{\frac{8R^2}{n} ln(\frac{2}{\delta})} \tag{5.18}$$

This is the intended concentration bound, which does not depend on the dimension of the patterns $\mathbf{x}$. The concentration bound proves that the empirical estimate $\hat{E}[yy'k(x,x')]$ of our proposed margin function, embedded in a loss (e.g., hinge loss of SVM), is a reasonable target for optimization that a binary classifier can achieve with paired sample points from an i.i.d set.

Chapter 6

Applications to Oil Sands Image Classification

In this chapter we describe the experiments and results of our proposed multi-scale based method on the oil sand video image analysis. The chapter is mainly divided into three main sections. In the first section a brief literature review of oil sand image analysis is given. Second section describes the experimental setup, data set and comparative results of DoG scale-space based MKL method for large lump detection problem. The final section highlights the experimental setup and results for wavelet based steam detection problem. For both detection problems we have performed vigourous testing and comparison of different parts of the algorithm to some of the state of art methods.

6.1 Related Work on Oil Sands Image Analysis

Oil sand images are relatively novel images. To date, very little research toward automated analysis have been done on these images. Zhang [2] investigated the application of different image processing algorithms to improve efficiency, reduce cost, and minimize the environmental impact in various stages of oil sands mining process. Majority of work on oil sands image analysis concentrated on image segmentation for size analysis [21, 63, 64, 65]. These methods apply various machine learning techniques combined with active contour or snake [21], watershed [63, 64], and grayscale image threshold method [65]. Ray *et al.* show that connected operator-based pre-filtering helps snake methods to a significant degree in oil sands mining image segmentation [66].

6.1.1 Large lump Detection

Most of the research works have been done for oil sand image segmentation, only few works have been proposed in large lump detection problem. Recently Wang *et al.* [67] describe a particle filter based solution for detecting large lump from oil sand images. They proposed application specific observation model for the Bayesian tracker for joint detection and tracking of large lump in an image sequence. To define the observation model, a feature detector is proposed. First, a local Otsu thresholding is done on the original image to get the thresholded image. Then, foreground pixel density within a local window for each location is calculated to get a density image. This step is due to the observation that in the thresholded image, foreground pixels in the real lump region tend to be more compact and smooth than the foreground pixels in the clutter regions, and consequently real lump pixels have higher densities in the density image than clutter pixels. Finally, a global thresholding on the density image is done to get the final observation. However, this method is heavily dependent on thresholding parameters. In addition it does not perform well on images with very low contrast and images captured in more difficult outdoor condition for example nightlight and snow.

6.1.2 Steam Detection

Automatic detection of steam from oil sand images is a challenging problem as the background of the images are not static. Steam detection is a relatively novel field. Ferrari *et al.*[1] proposed a real-time steam detection technique in oil sand video images. They treated the detection problem as a supervised pattern recognition problem which uses the dual-tree complex wavelet transform (DT-CWT) and the statistical HMT model computed for small 48×48 local regions of the image frames in order to characterize the steam texture pattern. A image processing technique is employed to automatically decide whether the frame can be used for further analysis by detecting the total area covered by steam in a video frame.

Steam detection problem is the similar to the problem of smoke detection, which is a well researched area for industrial applications. In [68] the background of the scene is estimated and decrease of high frequency energy of the scene is monitored using the spatial wavelet transforms of the current and the background images to detect smoke. In addition, chrominance values of pixels, periodic behavior in smoke boundaries and convexity of smoke regions are also analyzed and combined to obtain a final detection result.

A three step method is proposed in by Kim *et al.* [69]. The first step of the method decides whether the camera is moving or not. If the camera is moving, subsequent steps are skipped. Otherwise, the areas of change in the current input frame against the background image are detected and the regions of interest (ROIs) are located by connected component analysis in the second step. The block-based approach is applied in both the first and second steps. The final step is to determine whether each blob of the current input frame is smoke by using temporal information of color and shape in the detected blobs. The aforementioned methods only work if the background is stationary since the method used background information detect the features for smoke detection. This method will not perform well if the background is not static or the camera is moving. Also, changes in lighting (from clouds, time of day, etc.) is another factor that can easily disturb these methods as previous methods also depend on color information.

Kenji *et al* [70] applied fractal encoding concepts to extract smoke regions from an image. The self-similarity property of smoke are produced by fractal encoding of an image. Several papers showed that the smoke detection problem can be treated as a dynamic texture classification problem. Many authors proposed to model the spatio-temporal dynamics of image regions by using Gauss-Markov models, and infer the model parameters as well as the boundary of the regions in a variational optimization framework using the level-set technique.

Inspired by the work of Treyin *et al.* [68] and Ferrati *et al.* [1], which used wavelet-based technique for smoke and steam detection respectively, in this thesis we have proposed new wavelet based approach for steam detection from oil sand image. Local extrema in the wavelet domain correspond to the edges in an image. Steam gradually decreases the values of these extrema which eventually reduces the texture detail in an image. Because of the semi-transparent nature of steam these exterma values do not vanishes, they just loose some of their energy that results in degradation of sharpness of edges in image. To capture this information we have proposed to use wavelet subbands as feature to differentiate steam image from non steam image.

6.2 Results for Large Lump Detection

This section is divided into four subsections. First we describe the data sets used in our experiments. Next, we illustrate the efficacy of the proposed kernel function compared to other standard kernel functions, such as Gaussian, polynomial etc. Next section shows the performance of some standard object detection features, such as HoG, bag of visual words etc. on the large lump dataset. Finally, we compare our proposed MKL with three other popular MKL techniques from recent literature.

6.2.1 Description of Data

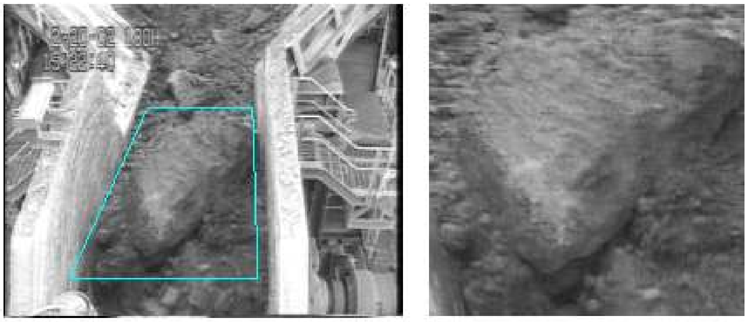
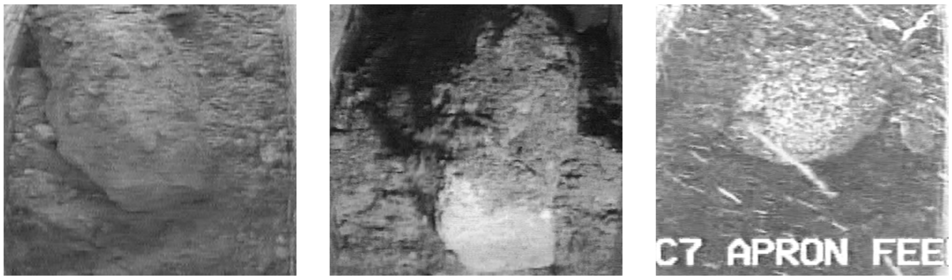


Figure 6.1: Region of interest for largelump detection.

Experiments have been performed on three oil sand data sets captured in three different outdoor conditions: normal daylight, night light and snowy day. For all experiments, a region of interest is defined as shown in Figure 6.17. The image sequence is cut according to the region of interest and perspective corrected to make it rectangular.

In daylight video set there are 768 image frames. Within those, 183 frames contain large lump and 585 contain no large lump. The main challenge with daylight dataset is the changing of the lighting condition. Night light video set has 585 frames with 316 large lump frames and 269 no large lump images. Although the lighting is fixed in this dataset, the main problem is the presence of shadow. Video on the snowy day contains 378 frames with 144 positive cases and 234 negative cases. Most of the image frames of this dataset contains snow flakes captured in a heavy snowy day. Some of the example large lump images from three data sets are shown in Figure 6.2.



(a) (b) (c)

Figure 6.2: Example large-lump images from (a) Daylight (b) Nightlight and (c) Snow datasets.

6.2.2 Experimental Setup

Any large lump is usually comprised of several image frames which correspond to multiple stages of a large lump evolution event. For example, Figure 6.3 shows five stages of a large lump event. Given such a multistage structures, it makes sense to extend the frame based classification to video clip based classification as we discussed above.

For each data set, we first construct training set with 50 video clips. Where 25 clips came from large lump events and 25 clips came from non- large lump events. Each such training video clip contains $T = 5$ consecutive frames. The labeling of each clip is done based on the middle frame information. If the middle frame (which is 3rd frame in our case) of a training clip contain a large lump we label this video clip as a positive sequence otherwise negative sequence. The number of DoG scales $N = 40$. Thus, the MKL algorithms will be sparsely choosing weights from $5 \times 40 = 200$ base kernel functions.

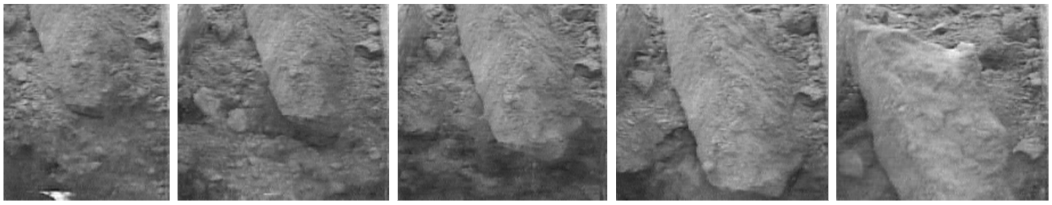


Figure 6.3: Large lump event sequence from Daylight dataset.

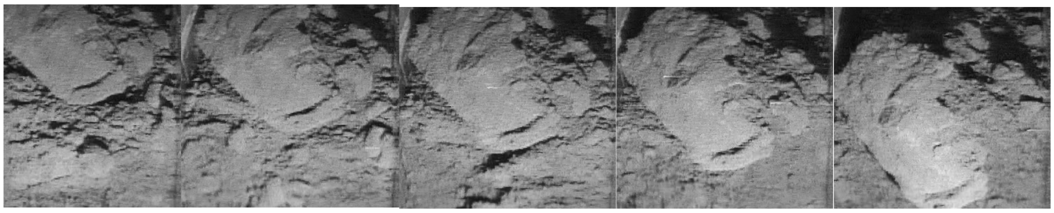


Figure 6.4: Large lump event sequence from Night dataset.

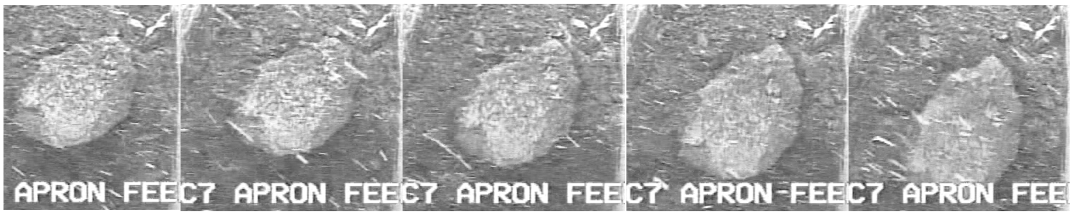


Figure 6.5: Large lump event sequence from Snow dataset.

For testing we have used sliding window approach discussed in section 1.2. Basically, we are testing each frame of the test video sequence. Each frame of the input video are classified based on the information of five consecutive frames.

The performance of our classification method is evaluated with precision, recall, accuracy and MCC (Matthew Correlation Coefficient) defined as follows:

$$\begin{aligned}
Precision &= \frac{tp}{tp + fp}, \\
Recall &= \frac{tp}{tp + fn}, \\
Accuracy &= \frac{tp + tn}{tp + tn + fp + fn}, \\
MCC &= \frac{tp * tn - fp * fn}{\sqrt{(tp + fn) * (tp + fp) * (tn + fp) * (tn + fn)}}
\end{aligned}$$

where tp, fp, tn and fn are true positive, false positive, true negative and false negative. MCC is generally regarded as a balanced measure that is very effective if the classes are of quite different sizes.

6.2.3 Convolution Kernel Function vs. Others

For each data set, we have generated its DoG scale-spaces, which contains 40 DoG responses corresponding to 40 consecutive scales. For each scale, we have constructed the base kernel k^{σ}. To compare our base convolution kernel function (5.2) with traditional kernel functions we have created kernel matrix from our training data using linear, polynomial, Gaussian, sigmoid kernel functions and calculate the kernel alignment score [46] for each scale to evaluate the compliance of a kernel to the data. The range of the alignment score is $[0, 1]$. The larger its value is, the better the kernel function. The graph in Figure 6.6 shows the alignment score on daylight training set for different scales using different kernel functions. The proposed circular convolution kernel function obtains a better alignment score than the other traditional kernel functions, especially in the coarser scales. This score plot illustrates that circular convolution kernel is appropriate for our application.

6.2.4 DoG vs. Other Features

In this section we have compared DoG with different popular features for object detection namely, Histogram of oriented gradient(HoG), bag of visual words and

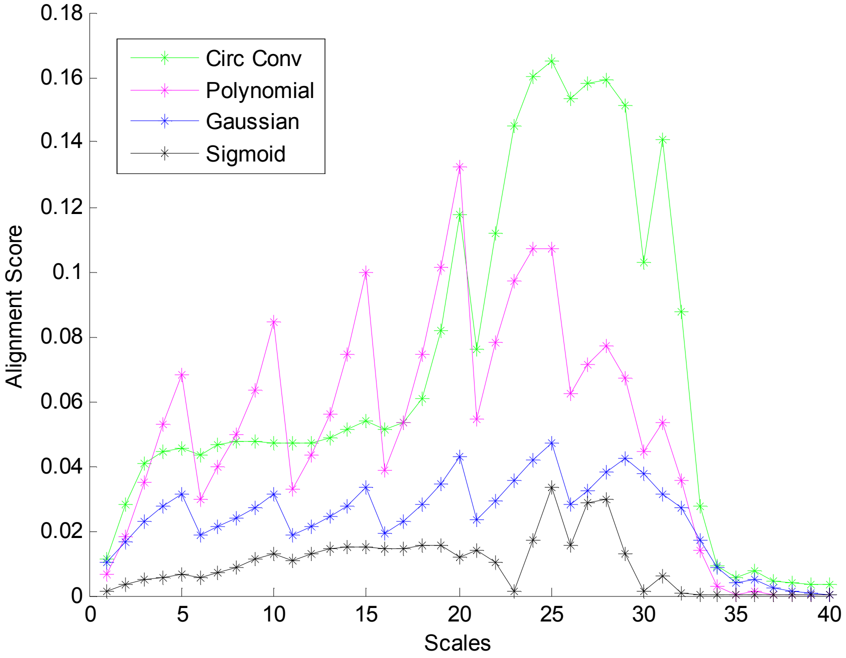


Figure 6.6: Alignment scores for different kernels at different scales.

dense word.

Histogram of Oriented Gradients

Histogram of Oriented Gradients (HOG) proposed in [71] is one of the popular feature descriptors used widely for object detection problem. HoG calculates occurrences of gradient orientation in localized portions of an image. The key concept behind the HoG descriptors is that local object appearance and shape of the image objects can be represented by the distribution of intensity gradients or edge directions. To calculate the descriptors, images are divided into cells which are small connected regions, and for each cell a histogram of gradient directions or edge orientations for the pixels is calculated. The combination of these histograms then represents the descriptor. For improved accuracy, the local histograms can be contrast-normalized by calculating a measure of the intensity across a larger region of the image, called a block, and then using this value to normalize all cells within the block. This normalization results in better invariance to changes in illumination or shadowing. The HOG descriptor has several important advantages over other descriptor methods. HoG descriptor is invariant to geometric and photometric transformations as it operates on localized cells. The HOG descriptor is thus particularly suited for human detection in images. Table 6.1 shows the performance for different block sizes on daylight dataset.

Table 6.1: Prediction performance of HoG features for different block sizes on Daylight Dataset

Block Size	Recall	Precision	MCC	Accuracy
5×5	0.6230	0.7451	0.5936	0.8597
9×9	0.7532	0.8655	0.6147	0.8034
13×13	0.6230	0.7451	0.5936	0.8597
17×17	0.6011	0.7639	0.5929	0.8610
19×19	0.6736	0.8291	0.6178	0.8228

Table 6.2: Prediction performance of different features for Daylight dataset

Feature	Recall	Precision	MCC	Accuracy
HoG	0.6011	0.7639	0.5929	0.8610
Bag of visual words	0.6612	0.7035	0.5869	0.8532
Dense word	0.6503	0.7391	0.6058	0.8623
DoG	0.7760	0.8023	0.7247	0.9013

Bag of Visual Words

This feature is proposed in [72]. Bag of Visual Words is one of the most popular techniques in object categorization because of their simplicity and relatively good performance over other prevalent methods. In this method, the SIFT descriptors are extracted at Hessian-Laplace points and quantized in a vocabulary of 3000 words, trained on features from several object instances. By using agglomerative information bottleneck (AIB) introduced in [72] vocabulary is then discriminatively compressed down to 64 words for each class. In our experiment, the same training images as before from each class are used.

Table 6.3: Prediction performance of different features for Nightlight dataset

Feature	Recall	Precision	MCC	Accuracy
HoG	0.7532	0.8655	0.6147	0.8034
Bag of visual words	0.8418	0.8693	0.6916	0.8462
Dense word	0.8323	0.8709	0.6854	0.8427
DoG	0.9076	0.8333	0.7043	0.8525

Dense Word

Dense word features applied in [73] for object detection are used here for large lump detection problem. This feature is also based on the SIFT descriptor. Rotationally invariant SIFT descriptors are extracted on a regular grid of five pixels at four multiple scales with raddi $r = 10$, 15, 20 and 25 pixels, zeroing the low contrast ones. Finally descriptors are quantized into 300 visual words.

Tables 6.2, 6.3 and 6.4 show the performances of the different features for three different data sets. Notice that the performance of the proposed MKL-based DoG feature appears as the last row in these tables and compares very well with its competitors.

Table 6.4: Prediction performance of different features for Snow dataset

Feature	Recall	Precision	MCC	Accuracy
HoG	0.6736	0.8291	0.6178	0.8228
Bag of visual words	0.6528	0.9400	0.6904	0.8519
Dense word	0.8333	0.7453	0.6463	0.8280
DoG	0.8056	0.9431	0.8039	0.9074

6.2.5 Proposed MKL vs. Others

After constructing basis kernels on each DoG scales, MKL algorithms are applied for sparse selection and weighting of kernels. For performance comparisons, three popular existing MKL techniques i.e. SKM [16], LSMKL [17] and GMKL [18] are chosen. For each MKL method a five-fold cross-validation has been performed to determine the value of the tuning parameters.

First, a comparison of the computational time by proposed method, SKM, LSMKL and GMKL on the three data sets are presented. All experiments were run on Intel core (R) TM(2) Dual processor with 2.43 GHz 64 bit machine with 3 GB RAM. The proposed 1-norm based MKL is implemented in Matlab 2009. The Matlab code of SKM and LSMKL is downloaded from the website http://homes.esat.kuleuven.be / sistawww/bioi/syu/l2lssvm.html and the Matlab code of GMKL are downloaded from http://research.microsoft.com/en-us/um/people/manik/code/GMKL/download.html. For LSMKL, MOSEK Optimization Software which combines the convenience of MATLAB with the speed of C code is used to solve optimization problems. Table 6.5 shows the CPU time needed for each multiple kernel learning method. It can be seen that the computational efficiency of the proposed method is comparable to the other popular MKL techniques.

Table 6.5: The CPU time (in seconds) needed for each method

Dataset	Proposed Method	SKM	LS-MKL	GMKL
Daylight	0.6473	0.6138	0.3105	2.4056
Nightlight	0.6510	0.6867	0.3495	2.3910
Snow	0.6033	0.3411	0.3126	3.7630

Table 6.6 shows the number of base kernels selected by each method. Although the average number of selected scales of the proposed algorithm with respect to other MKL methods is relatively high, the number is still sparse represents only 12% of the total kernels on average. As a result our method is reasonable for real time application.

Figure 6.7 shows the selected weights from different frames for different datasets using our method. From these plots we can see, most of weights are selected in middle frames comparing to beginning frames and trailing frames. It is because we have designed the training set in such a way so that for the positive video clip the middle frame will always contain large lump and for the negative video clip the middle frame will have no large lump. The beginning and trailing frames may or may not contain large lump based on the time point of the middle frame. For example if the middle frame is the staring point of large lump event then the beginning frames will not have any large lump while if the middle frame is end point of large lump event then begging frames will certainly contain the image of that large lump. So middle is always more important than other frames for classification process which we actually want to classify using our classifier.

Table 6.6: Number of kernels needed for each method

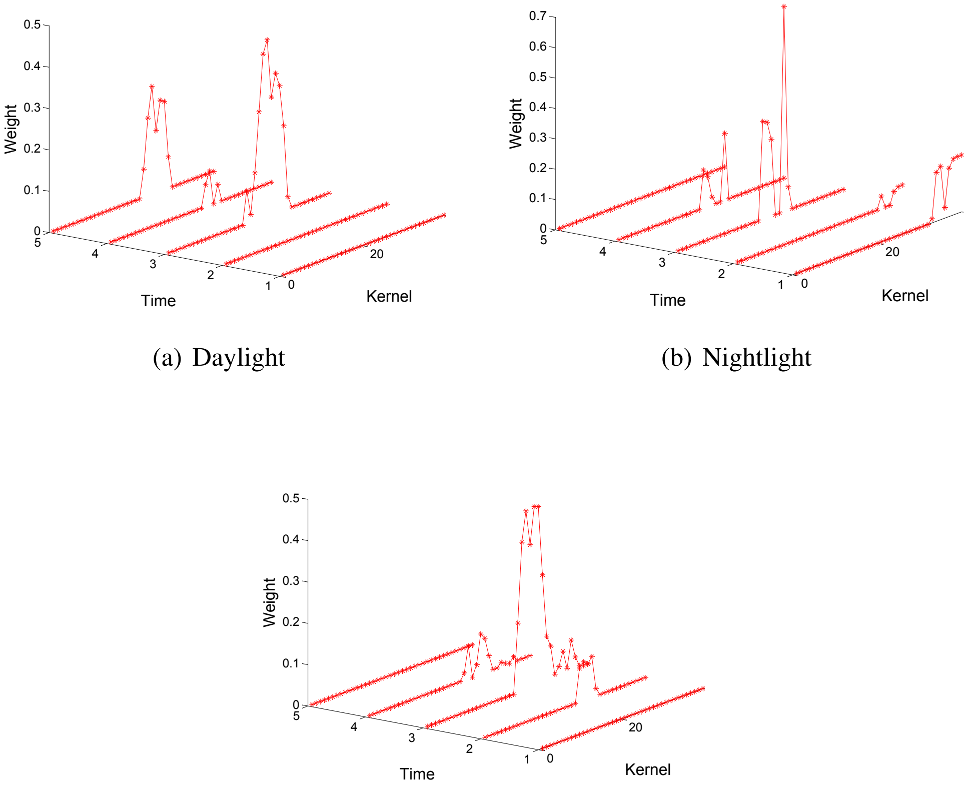
Dataset	Proposed Method	GMKL	SKM	LS-MKL
Daylight	21	15	9	9
Nightlight	21	40	11	12
Snow	32	18	7	10

Table 6.7: Prediction Results of different MKL methods for Daylight dataset respectively.

Method	Recall	Precision	MCC	Accuracy
Proposed Method	0.7760	0.8023	0.7247	0.9013
GMKL	0.7049	0.7914	0.6742	0.8857
SKM	0.6995	0.7901	0.6700	0.8844
LS-MKL	0.6831	0.6720	0.5759	0.8455

Table 6.8: Prediction Results of different MKL methods for Nightlight dataset.

Method	Recall	Precision	MCC	Accuracy
Proposed Method	0.9076	0.8333	0.7043	0.8525
GMKL	0.8766	0.8318	0.6727	0.8376
SKM	0.6234	0.8955	0.5534	0.7573
LS-MKL	0.7500	0.8525	0.5964	0.7949



(a) Daylight

(b) Nightlight

(c) Snow

Figure 6.7: Selected kernel weights for different datasets

Table 6.9: Prediction Results of different MKL methods for Snow dataset.

Method	Recall	Precision	MCC	Accuracy
Proposed Method	0.8056	0.9431	0.8039	0.9074
GMKL	0.8125	0.9213	0.7914	0.9021
SKM	0.6875	0.9612	0.7312	0.8704
LS-MKL	0.7986	0.9350	0.7923	0.9021

From tables 6.7, 6.8 and 6.9 we can see that our sparse MKL method outperforms other three popular multiple kernel learning techniques with respect to most of the performance metrics for daylight and night light data sets. For the snow data set our method achieves the same performance as LSMKL and they perform better than the other two other methods.

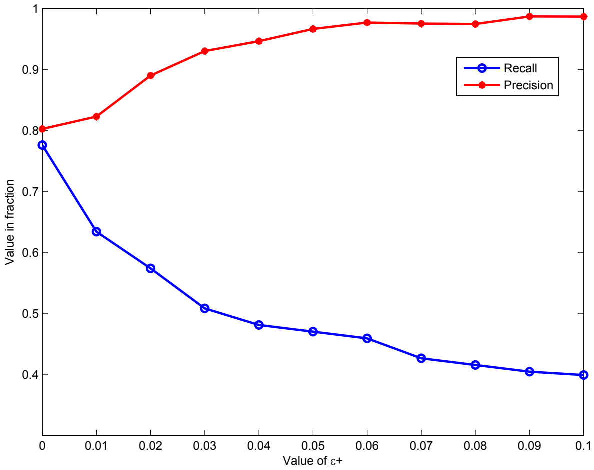


Figure 6.8: Daylight Data: the value of recall and precision versus $\epsilon+$ (x-axis) for a SVM training using weighted circular convolution kernel. As $\epsilon+$ (values are in fraction) increases, the precision increases while the recall drops.

Our system is also able to tune precision or recall based on user requirement. Here we have combined the scheme provided in [74] in our proposed system to adjust precision and recall in support vector machine and provide the performance using recall-precision curve. The balance between recall and precision can be controlled using the following technique: the diagonal elements of the optimized kernel matrix are supplemented by fixed positive contributor $\epsilon+$ and $\epsilon-$. Controlling these two parameters one can vary precision and recall. This method actually corresponds to an asymmetric margin; i.e., the class with smaller ϵ will be kept further away from the decision boundary. Figure 6.8 to 6.13 show the recall precision graph of the proposed classifier with ($\epsilon-$=0) and varying ($\epsilon+$) and with ($\epsilon+$=0) and varying ($\epsilon-$) on three different dataset respectively.

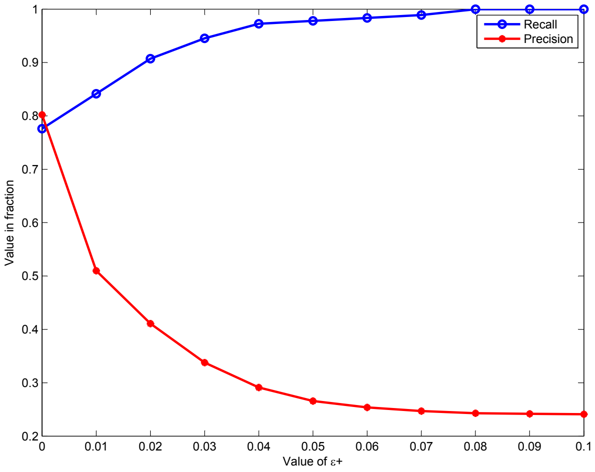


Figure 6.9: Daylight Data: the value of recall and precision versus $\epsilon-$ (x-axis) for a SVM training using weighted circular convolution kernel. As $\epsilon-$ (values are in fraction) increases, the recall increases while the precision drops significantly.

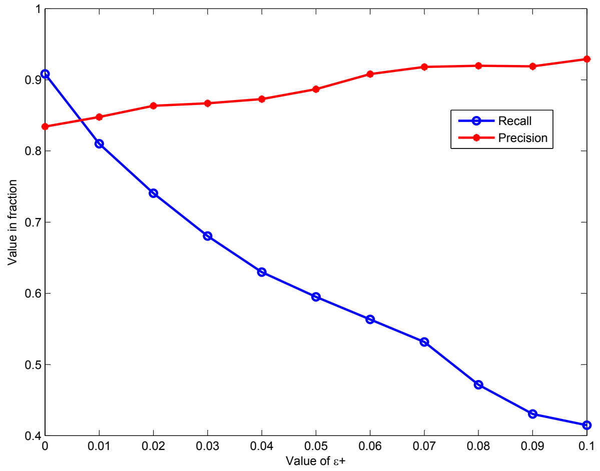


Figure 6.10: Nightlight Data: the value of recall and precision versus $\epsilon+$ (x-axis) for a SVM training using weighted circular convolution kernel. As $\epsilon+$ (values are in fraction) increases, the precision increases while the recall drops.

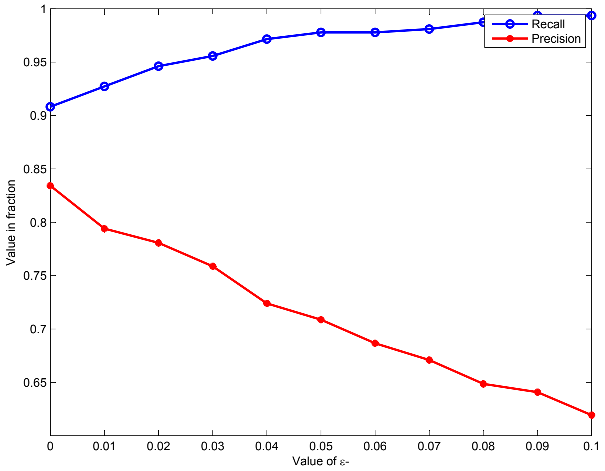


Figure 6.11: Nightlight Data: the value of recall and precision versus $\epsilon-$ (x-axis) for a SVM training using weighted circular convolution kernel. As $\epsilon-$ (values are in fraction) increases, the recall increases while the precision drops significantly.

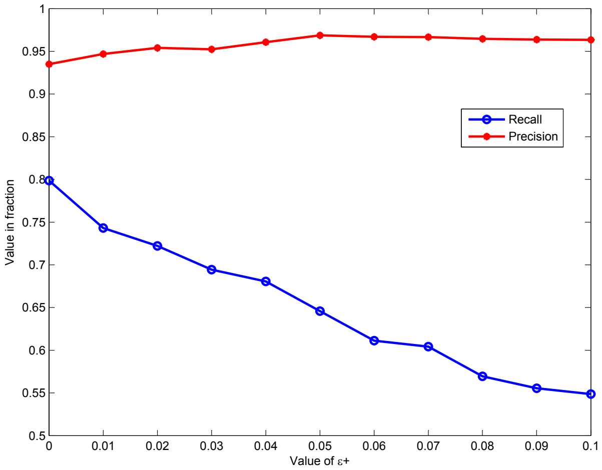


Figure 6.12: Snow Data: the value of recall and precision versus $\epsilon+$ (x-axis) for a SVM training using weighted circular convolution kernel. As $\epsilon+$ (values are in fraction) increases, the precision increases while the recall drops.

To compare with other MKL methods more concisely, we have interpolated the above recall-precision curves to obtain recall values of our method at different precision levels and the compared with other methods at the same precision levels obtained by those methods. Following bar diagrams show the comparison of the recall values of our method on different dataset with GMKL, SKM and LS-MKL respectively.

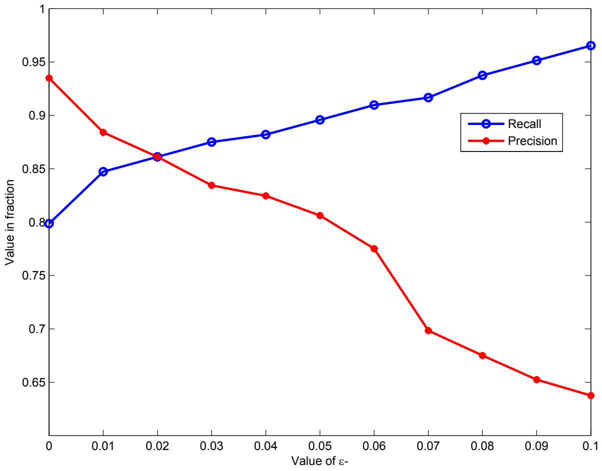


Figure 6.13: Snow Data: the value of recall and precision versus $\epsilon-$ (x-axis) for a SVM training using weighted circular convolution kernel. As $\epsilon-$ (values are in fraction) increases, the recall increases while the precision drops significantly.

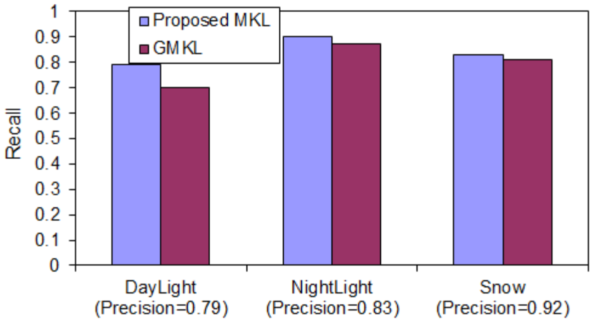


Figure 6.14: Comparison with GMKL for different dataset

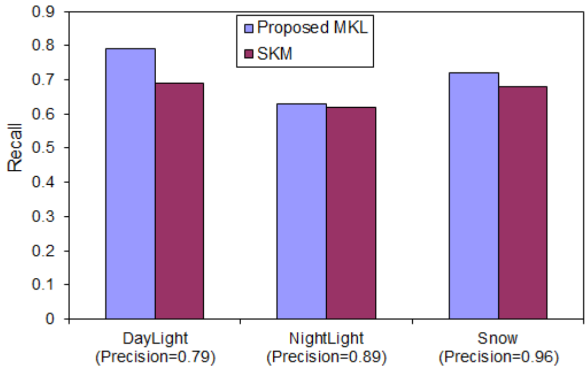


Figure 6.15: Comparison with SKM for different dataset

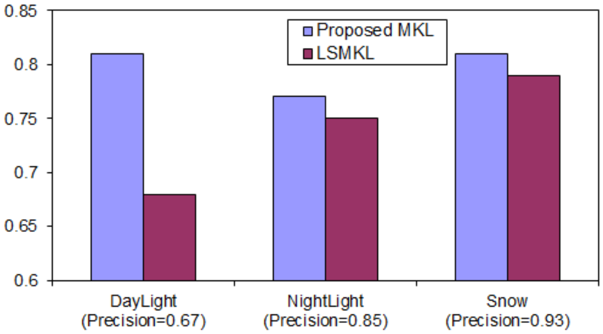


Figure 6.16: Comparison with LSMKL for different dataset

6.2.6 Statistical Significance Test

McNemars test [75] is used to decide whether any apparent difference in error-rates between the proposed algorithm and existing algorithms. McNemar's test is a non-parametric method, used widely to test the significance of the results of binary classifier. This test is performed by summarizing the classification results of the two algorithms tested on the same dataset. McNemar's test is given by the following equation:

$$M = \frac{|b - c| - 1)^2}{b + c} > \chi^2_{1,\alpha} \tag{6.1}$$

where b is the number of examples correctly classified by classifier 1 but misclassified by classifier 2 and c is the number of examples misclassified by classifier 1 but correctly classified by classifier 2

M is distributed approximately as χ^2 with 1 degree of freedom. For a 95% confidence test, $\chi^2_{1,095} = 3.84$. So if $M > 3.84$ then with 95% confidence, we can reject the null hypothesis H_0 that the two classifies have the same error rate.

McNemar's test has been applied to the classification results obtained using DoG features and other types of features. Table 6.10 shows the M values. McNemar's test has also been applied to test the significance of our MKL method results with the results obtained by other MKL algorithms. Table 6.11 shows the M values between our MKL approach and other MKL methods.

Table 6.10: McNemar's test result (value of M) for different types of features

Dataset	HoG	Bag of visual words	Dense word
Daylight	7.2000	10.5366	7.1271
Nightlight	5.5603	0.2049	0.0804
Snow	19.1489	11.7015	6.6852
Overall	**25.8786**	**16.1571**	**9.8908**

Table 6.11: McNemar's test result (value of M) for different MKL algorithms

Dataset	GMKL	SKM	LSMKL
Daylight	1.8333	2.4853	24.1644
Nightlight	0.9552	24.0079	16.0147
Snow	0	5.3333	0
Overall	2.9400	**30.4253**	**34.7222**

From the above two tables, if we consider the overall results on the three datasets, we can say that DoG worked significantly better than other features on the oil sands images. Similarly, proposed MKL performed significantly better than LSMKL and SKM. However the performance difference between proposed method and GMKL is not significant. This is with 95% confidence.

6.3 Results on Steam Detection

This section is divided into four subsections. Data sets and experimental setup used are discussed first. Next, a comparative study of using different wavelet families are shown. The comparison of the proposed circular convolution kernels with some standard kernel functions are discussed in section 6.3.2. A performance evaluation the proposed MKL with three other popular MKL techniques are also studied on steam dataset in section 6.3.3. Final section shows the comparison with existing steam detection method proposed in [1].

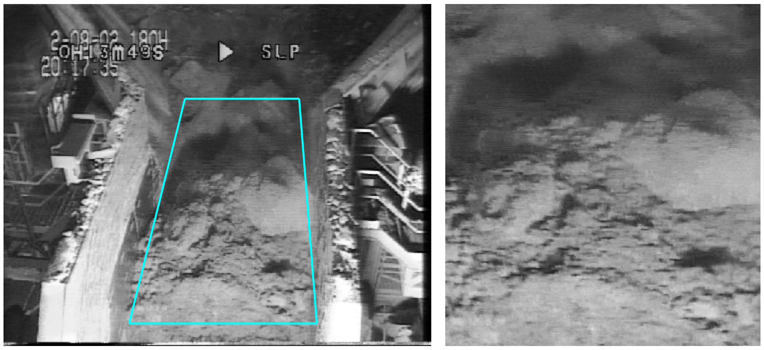


Figure 6.17: Region of interest for steam detection.

The steam dataset contains 648 image frames among which 563 are positive i.e. large lump frames and 85 are negative i.e. no large lump sequences. We have used the same sliding window approach as we used for large lump detection problem. The system is trained with 20 video clips from which 10 video clips came from steam images and 10 video clips came from no steam images. Each such training video clip contains $T = 5$ consecutive frames and labeled according to the information of middle frame of the clip. Each image in the clip are decomposed in 10 level into 40 wavelet subbands. Thus, the MKL algorithms will be sparsely choosing weights from $5 \times 40 = 200$ base kernel functions.

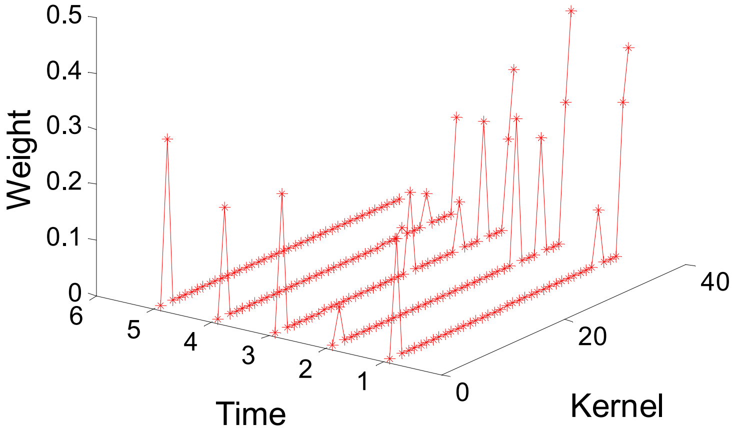


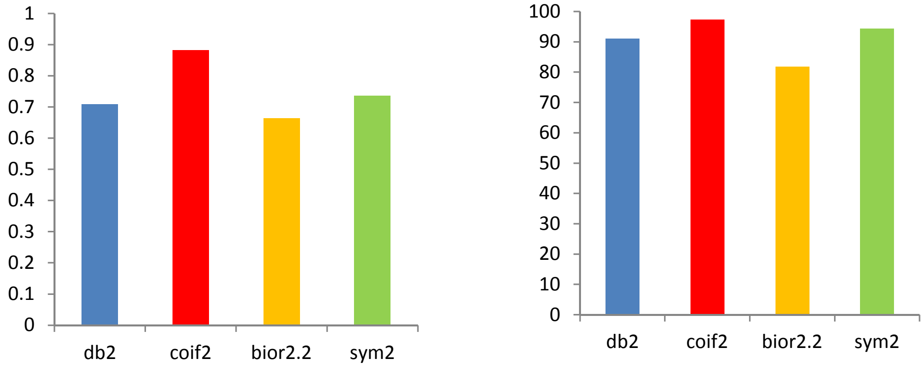
Figure 6.18: Selected kernel weights for steam datasets

6.3.1 Wavelet Families

A large choice of wavelet families exists depending on the choice of wavelet function. In our work, we have compared Haar, Daubechies, Symlets, Coiflets and Biorthogonal on our test dataset, assuming in their capability to localize different information in time and frequency. Following graphs show the performance of these wavelet families for steam detection problem:

6.3.2 Comparison with Different Kernel Functions

To compare our basis cross correlation kernel with traditional kernel functions we have created kernel matrix from our training data using linear, polynomial, Gaussian, sigmoid kernel functions and calculate the kernel alignment score proposed in [46] for each scale to evaluate the compliance of a kernel to the data.



(a) MCC

(b) Accuracy

Figure 6.19: Performance of different wavelet families in terms of MCC and accuracy

Table 6.12: Results for Steam dataset using different kernel.

Kernel	Precision	Recall	MCC	Accuracy
Polynomial	0.8353	0.8987	0.8472	0.9660
Linear	0.6852	0.8706	0.7339	0.9306
Gaussian	0.7209	0.9394	0.6644	0.8182
Circular Convolution	0.9359	0.8588	0.8818	0.9738

6.3.3 Proposed MKL vs. Others

After constructing basis kernels on each wavelet subbands, the proposed MKL algorithms are applied for sparse selection and weighting of kernels. Three popular existing MKL techniques discussed before i.e. SKM [16], LSMKL [17] and GMKL [18] are also applied to generate optimized kernel weights. From tables 6.13 we can see that our sparse MKL method outperforms other three popular multiple kernel learning techniques with respect to most of the performance metrics.

Table 6.13: Results for steam dataset using different multiple kernel learning algorithm.

Method	Precision	Recall	MCC	Accuracy
Proposed Method	0.9359	0.8588	0.8818	0.9738
SKM	0.7642	0.9529	0.8292	0.9552
GMKL	0.7524	0.9294	0.8092	0.9506
LS-MKL	0.7570	0.9529	0.8245	0.9537

6.3.4 Statistical Significance

In this section the statistical significance of the proposed method are given over other existing MKL approaches. Table 6.14, shows the McNemar's test values on steam dataset. From the M values we can conclude that the proposed method perform significantly better than all the other MKL approaches.

Table 6.14: McNemar's test result(value of M) for different MKL algorithms on Steam dataset

Dataset	GMKL	SKM	LSMKL
Steam	7.2593	4.3214	4.9655

6.3.5 Comparison with Previous Steam Detection Method

The proposed steam detection method is compared to the Ferrari *et al*'s steam detection method. The small 48×48 local regions of the image frames are classified by the steam texture pattern which are computed using dual-tree complex wavelet transform (DT-CWT)[76] and the statistical HMT model. After classifying each small regions the whole image frames are classified according to a voting. If $1/4$th of the image region are covered with steam we classify the image frame as steam images. Following table shows the performance of the method proposed in [1]

Table 6.15: Results for Steam dataset obtained using [1].

Kernel	Precision	Recall	MCC	Accuracy
Linear	0.4848	0.5647	0.4449	0.8642
Polynomial	0.6184	0.5529	0.5261	0.8966
Gaussian	0.5625	0.4235	0.4230	0.8812

The steam detection method does not give good result in case of this database. The reason of poor performance is due to the nature of the dataset. The oil sand dataset we used here is really challenging. Following example shows some of misclassified images using the steam detection method in [1].

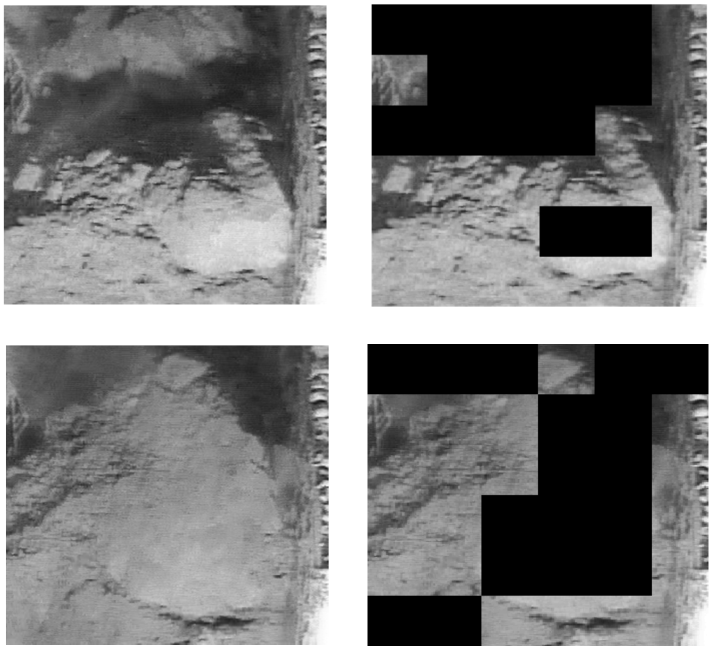


Figure 6.20: Example of false positive obtained with steam detection method in [1].

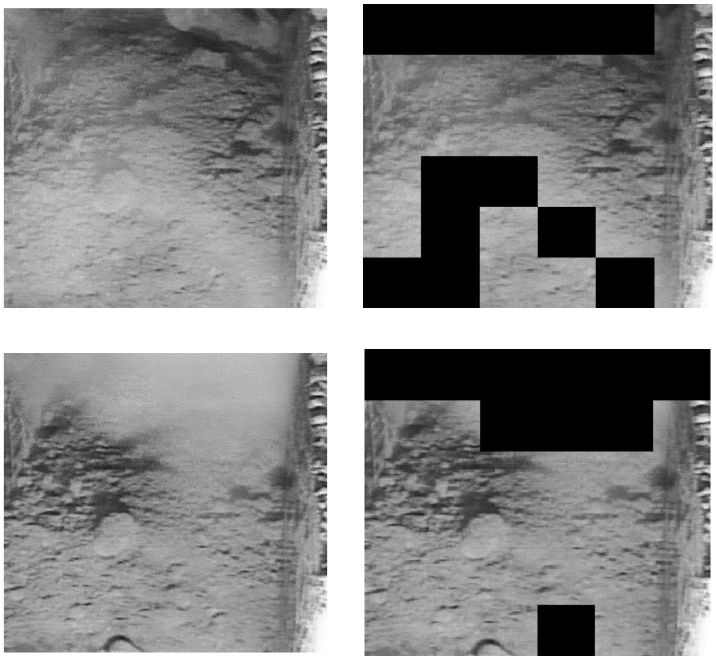


Figure 6.21: Example of false negative obtained with steam detection method in [1].

Chapter 7

Conclusion and Future Work

This chapter gives the summary and outlook of the proposed method. The future research works are also described in the last section.

7.1 Summary

Over the last few decades the scale-space representation of images has obtain significant attention in computer vision field. The representation produces a family of blurred versions of the input image where the details of the original image are lost with increasing smoothness. The degree of smoothing at which an object vanishes basically measures the size of an object and provides information about the appropriate level of detail that allows the visual system to concentrate on the object under consideration. In this research work, scale-space based multi-scale image representation are proposed as potential features for image classification. To overcome two basic limitations of using scale-space images as features we have employed multiple kernel learning. The first limitation i.e the high dimensionality of the feature space are managed by designing appropriate kernel functions from theses features. The second problem of scale selection are solved by applying MKL approach on the kernels designed from scale-space image features.

Towards designing such efficient scale-space based classification system shift invariant, convolution kernel function is proposed which is suitable for the proposed features. Further, a novel framework of multiple kernel learning based on 1-norm SVM is proposed which requires only a linear equation solver or alternatively the method proposed in [31] to make the MKL implementation simple, fast, and easily accessible.

The proposed scale-space based MKL method is applied to solve novel image application problems on oil sand mining videos. Oil sand images are completely novel image and the research works performed on this images are very insignificant. Two important detection problems mainely large lump detection and steam detection are cracked here by using the proposed classification framework. For large lump detection problem DoG scale-space is investigated. At larger scale DoG response provides clear and big blob like structure which corresponds to largelump. However in the small scale there is only small blobs. This responses provides clue of using DoG scale-space for large lump detection. We have applied some other popular scale-space i.e. wavelet, steerable filter and gabor on the large lump images to show the response at different scales. From the response we can see that the responses are really not useful to discriminate large lump from no large lump images. For steam detection problem we have applied wavelet scale-space. Since the steam has quite different texture than the background oil sand images, multi-scale analysis of discrete wavelet transform is an effective feature to differentiate images with steam and no steam.

The large dimensionality of scale-space features are managed by designing appropriate kernel and applying kernel based method. We have proposed a novel circular convolution kernel to provide shift invariance through kernel calculation. Shift invariant kernel is effective both for DoG and wavelet scale-space as large lump or steam can appear any part of the image frame and still we need to detect those.

Any scale-space based features face the inevitable issue of scale selection. The principal novelty of our proposed image analysis is that we turn this scale selection into a multiple kernel learning (MKL) problem by designing a new kernel function involving scale-space. Our proposed MKL selects only a few relevant scales from the complete scale space and classify images using a support vector machine (SVM) classifier.

Vigorous Experimental tests are performed for different types of oil sand videos.

Also the different parts of the proposed method are compared with some existing approaches. For example the performance of circular convolution kernel are compared with some standard and popular kernel functions like polynomial, gaussian etc. Results shows the clear improvement of the detection accuracy using circular convolution kernel. The proposed MKL approach are compared with another three standard MKL techniques and result showed the better performance of the proposed 1-norm SVM based MKL. Most of the cases the improvements are statically significant. In addition, the proposed method is very simple and efficient.

7.2 Future Works

7.2.1 Best Basis Selection

Discrete wavelet packet transform, which is the generalization of the DWT, provides richer subband analysis without the constraint of a dyadic decomposition. It generates a huge number of features, as a result an extremely high dimensional feature space is generated most of which are redundant for accurate and efficient classification process. Selection of effective features from this high dimensional space is really challenging as we see in this thesis. In the case of wavelet packet feature selection method, an image is only needed be decomposed into the wavelet subbands selected by the best basis selection method during the training stage. To determine subbands suited well for classification, it is important to identify an appropriate selection criterion and a search strategy to optimize this criterion. The choice over the entire collection of possible subband combinations is called best basis selection [77], since the selection of different subbands corresponds to the selection of different basis functions for representing the image.

Best basis selection is a well-studied field. A cost function based on L_1-norm is proposed by Chang and Kuo [78] for wavelet tree pruning in a top down manner. Acharyya and Kundu [79] proposed a similar idea of adaptive decomposition algorithm by employing an energy based cost function to identify most significant subbands and then decide whether further decomposition of the particular channel would require for better result or not. Although these top down approaches are computationally efficient, the top-down search method employed in a best basis selection algorithm cannot guarantee an optimal solution. Saito and Coifman [80] proposed to maximize the differences in time-frequency energy distributions of each class. To measure the dissimilarity between energy distributions of a particular subband for all classes they applied symmetric Kullback-Leibler (KL) distance. Recently a wavelet subband selection algorithm based on statistical dependence of different subbands is proposed by Huang *et al* [81]. The use of MKL framework to map the best basis selection problem are left for our future work.

7.2.2 Non-Linear Scale-Space

Different non-linear scale-spaces are proposed in the literature in connection to address different problems. Each of this non-linear scale-space representation has its own properties. To use the multi-scale representation for high-level image processing it is important to understand what kind of information is present in the decomposition and how this high level process can be benefited from this representation. From the study it seems that the proposed classification system takes advantage of DoG scale-space based decomposition as the most important components that the structures of large lumps are preserved through scales as blob like structure. However, linear scale-space like DoG scale-space poses several disadvantages. The Gaussian kernel blurs the image region uniformly. As a result some important region of interest like edges can also become blurred. Furthermore, localization of the structures of interest becomes highly imprecise at larger scales. In many cases it is difficult to trace the object at very large scales due to excessive blurring and the appearance of spurious extrema in two dimensions. Various solutions have been proposed to reduce this problem. One possible solution is to use non-linear scale-spaces.

7.2.3 Multiple Object Localization

Object localization is one of the challenging task for the automatic understanding of images. It is also important to separate an object from the background, or to analyze the spatial relations of different objects in an image to each other. However most of the object detection techniques including many state-of the-art methods only solve a binary classification problem. Also our research work on object detection can decide whether an object is present in an image or not, but not where exactly in the image the object is located. As our future work, we propose to apply our MKL framework to perform object localization. However the main challenge of applying our method for object localization is the computational efficiency.

To add the object localization functionality to generic object categorization systems, the common approach is to apply sliding window which has been established as a state of-the-art. Most successful localization techniques at the recent PASCAL VOC 2007 challenge on object category localization relied on this technique [82]. The sliding window principle treats localization as localized detection. A classifier function is applied subsequently to subimages within an image and the maximum of the classification score indicates the presence of an object in this region. However, as the number of subimages grows as n^4 for images of size nxn, the sliding window approach becomes computationally too expensive to evaluate the quality function exhaustively for all of these.

We can tackle this problem by using integral image. In the context of realtime face detection, Viola and Jones have proposed to use integral images [83], which allow for very fast computation of any box-type convolution filter.

7.2.4 Explore Different Application Domains

We have shown the usefulness of using scale-space based MKL method for detection problem in oil sand image analysis. However, it is only one piece of the puzzle in solving the automatic detection problem. There are several broad areas that may be useful for further investigation. For example some very popular application domain like fingerprint verification, speaker identification and verification our scale-space based MKL can be very promising.

Bibliography

[1] R. Ferrari, H. Zhang, and C. Kube, "Real-time detection of steam in video images," *Pattern Recognition*, vol. 40, no. 3, pp. 1148–1159, 2007.

[2] H. Zhang, "Image processing for the oil sands mining industry," *IEEE Signal Processing Magazine*, vol. 25, no. 6, pp. 198–200, 2008.

[3] "Gold in the sands," in *Green Car Congress*, 2006.

[4] C. J. Burges, "A tutorial on support vector machines for pattern recognition," *Data Mining and Knowledge Discovery*, vol. 2, pp. 121–167, 1998.

[5] D. G. Lowe, "Object recognition from local scale-invariant features," in *Computer Vision, 1999. The Proceedings of the Seventh IEEE International Conference on*, vol. 2, 1999, pp. 1150–1157 vol.2.

[6] A. Witkin, "Scale-space filtering: A new approach to multi-scale description," in *in Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing (ICASSP)*, vol. 9, 1984, pp. 150–153.

[7] T. M. Sezgin and R. Davis, "Scale-space based feature point detection for digital ink," in *SIGGRAPH '06: ACM SIGGRAPH 2006 Courses*. New York, NY, USA: ACM, 2006, p. 29.

[8] D. P. Mukherjee and S. T. Acton, "Cloud tracking by scale space classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 40, pp. 405–415, 2002.

[9] A. Petrovic, O. D. Escoda, and P. Vandergheynst, "Multiresolution segmentation of natural images: From linear to non-linear scale-space representations," *IEEE Trans. Image Process*, vol. 13, pp. 1104–1114, 2003.

[10] T. , "Scale-space theory: A basic tool for analysing structures at different scales," *J. of Applied Statistics*, vol. 21(2), pp. 224–270, 1994.

[11] T. Lindeberg, "Scale-space a framework for handling image structures at multiple scales," 1996.

[12] T. Lindeberg and L. Bretzner, "Real-time scale selection in hybrid multi-scale representations," pp. 148–163, 2003.

[13] I. Guyon and A. Elisseeff, "An introduction to variable and feature selection," *J. Mach. Learn. Res.*, vol. 3, pp. 1157–1182, Mar. 2003.

[14] W. Li, K. Mao, H. Zhang, and T. Chai, "Designing compact gabor filter banks for efficient texture feature extraction," in *11th International Conference on Control, Automation, Robotics and Vision, ICARCV 2010, Singapore, 7-10 December 2010, Proceedings*. IEEE, 2010, pp. 1193–1197.

[15] G. Lanckriet, N. Cristianini, P. Bartlett, and L. E. Ghaoui, "Learning the kernel matrix with semi-definite programming," *Journal of Machine Learning Research*, vol. 5, 2002.

[16] F. R. Bach, G. R. G. Lanckriet, and M. I. Jordan, "Multiple kernel learning, conic duality, and the smo algorithm," in *ICML*, ser. ACM International Conference Proceeding Series, C. E. Brodley and C. E. Brodley, Eds., vol. 69. ACM, 2004.

[17] S. Sonnenburg, G. Rätsch, C. Schäfer, and B. Schölkopf, "Large scale multiple kernel learning," *Journal of Machine Learning Research*, vol. 7, pp. 1531–1565, 2006.

[18] B. Varma, M.and Babu, "More generality in efficient multiple kernel learning," *In Proceedings of the International Conference on Machine Learning*, June 2009.

[19] S. McCarthy, "Oil sands bitumen to flow to west coast by 2015: Enbridge," in *The Globe and Mail*, 2010.

[20] A. Burrowes, R. Marsh, N. Ramdin, and C. Evans, "Alberta's energy reserves 2006 and supply/demand outlook 2007-2016," in *Alberta Energy and Utilities Board*, 2007.

[21] B. Saha, N. Ray, and H. Zhang, "Computing oil sand particle size distribution by snake-pca algorithm," in *Acoustics, Speech and Signal Processing, 2008. ICASSP 2008. IEEE International Conference on*, 31 2008-April 4 2008, pp. 977–980.

[22] S. G. Mallat, "A theory for multiresolution signal decomposition: the wavelet representation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 11, pp. 674–693, 1989.

[23] E. P. Simoncelli and W. T. Freeman, "The steerable pyramid: A flexible architecture for multi-scale derivative computation," *IEEE Second Int'l Conf on Image Processing*, 1995.

[24] J. Babaud, A. P. Witkin, M. Baudin, and R. O. Duda, "Uniqueness of the gaussian kernel for scale-space filtering," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 8, no. 1, pp. 26–33, 1986.

[25] H. G. Feichtinger and T. Strohmer, Eds., *Gabor Analysis and Algorithms: Theory and Applications*, 1st ed. Birkhauser Boston, 1997.

[26] P. Perona, "Steerable-scalable kernels for edge detection and junction analysis," in *Image and Vision Computing*, 1992, pp. 3–18.

[27] H. P. Benson, "Fractional programming with convex quadratic forms and functions," *European Journal of Operational Research*, vol. 173, no. 2, pp. 351 – 369, 2006.

[28] P. Perona and J. Malik, "Scale-space and edge detection using anisotropic diffusion," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 12, no. 7, pp. 629–639, 1990.

[29] L. Alvarez, P.-L. Lions, and J.-M. Morel, "Image selective smoothing and edge detection by nonlinear diffusion. ii," *SIAM J. Numer. Anal.*, vol. 29, no. 3, pp. 845–866, 1992.

[30] L. I. Rudin, S. Osher, and E. Fatemi, "Nonlinear total variation based noise removal algorithms," *Phys. D*, vol. 60, no. 1-4, pp. 259–268, 1992.

[31] F. Meyer and P. Maragos, "Nonlinear scale-space representation with morphological levelings," *Journal of Visual Communication and Image Representation*, vol. 11, no. 2, pp. 245 – 265, 2000.

[32] J. J. Koenderink, "The structure of images," *Biol. Cybern.*, vol. 50, p. 363370, 1984.

[33] J. L. Crowley and A. C. Sanderson, "Multiple resolution representation and probabilistic matching of 2d grayscale shape," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 9, no. 1, p. 113121, 1987.

[34] Y. Dufournaud, C. Schmid, and R. P. Horaud, "Matching images with different resolutions," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Hilton Head Island, South Carolina, USA*. IEEE Computer Society Press, Jun 2000, pp. 612–618.

[35] J. L. Crowley and A. C. Parker, "Representation for shape based on peaks and ridges in the difference of low-pass transform," Robotics Institute, Pittsburgh, PA, Tech. Rep. CMU-RI-TR-83-04, May 1983.

[36] T. Lindeberg, "Detecting salient blob-like image structures and their scales with a scale-space primal sketch: A method for focus-of-attention," *International Journal of Computer Vision*, vol. 11, pp. 283–318, 1993.

[37] ——, "Feature detection with automatic scale selection," *International Journal of Computer Vision*, vol. 30, pp. 79–116, 1998.

[38] L. Bretzner and T. Lindeberg, "Feature tracking with automatic selection of spatial scales," *Computer Vision and Image Understanding*, vol. 71, pp. 385–392, 1996.

[39] K. Mikolajczyk and C. Schmid, "Scale and affine invariant interest point detectors," *International Journal of Computer Vision*, vol. 60, no. 1, pp. 63–86, 2004.

[40] J. Shawe-Taylor and N. Cristianini, *Kernel Methods for Pattern Analysis*. Cambridge University Press, June 2004.

[41] "http://en.wikipedia.org/wiki/binary_classification."

[42] V. N. Vapnik, *The Nature of Statistical Learning Theory*. Springer, November 1999. [Online]. Available: http://www.amazon.ca/exec/obidos/redirect?tag=citeulike09-20\&amp;path=ASIN/0387987800

[43] C. Cortes and V. Vapnik, "Support-vector networks," in *Machine Learning*, 1995, pp. 273–297.

[44] Z. J. Rosset, S. Hastie, and T. R., "1-norm support vector machines," *Neural Information Processing Systems*, 2003.

[45] B. Haasdonk and D. Keysers, "Tangent distance kernels for support vector machines," in *Pattern Recognition, 2002. Proceedings. 16th International Conference on*, vol. 2, 2002, pp. 864 – 868 vol.2.

[46] N. Cristianini, J. Shawe-taylor, A. Elisseeff, and J. Kandola, "On kernel-target alignment," in *Advances in Neural Information Processing Systems 14*. MIT Press, 2001, pp. 367–373.

[47] H. Xiong, M. Swamy, and M. Ahmad, "Optimizing the kernel in the empirical feature space," *Neural Networks, IEEE Transactions on*, vol. 16, no. 2, pp. 460–474, March 2005.

[48] B. Chen, H. Liu, and Z. Bao, "A kernel optimization method based on the localized kernel fisher criterion," *Pattern Recogn.*, vol. 41, no. 3, pp. 1098–1109, 2008.

[49] D.-Y. Yeung, H. Chang, and G. Dai, "Learning the kernel matrix by maximizing a kfd-based class separability criterion," *Pattern Recognition*, vol. 40, no. 7, pp. 2021–2028, July 2007. [Online]. Available: http://dx.doi.org/10.1016/j.patcog.2006.12.031

[50] J. He, S. fu Chang, and L. Xie, "Fast kernel learning for spatial pyramid matching," 2008.

[51] C. H. Nguyen and T. B. Ho, "An efficient kernel matrix evaluation measure," *Pattern Recogn.*, vol. 41, pp. 3366–3372, November 2008.

[52] Y. Ying, K. Huang, and C. Campbell, "Enhanced protein fold recognition through a novel data integration approach," in *BMC Bioinformatics*, vol. 10, no. 1, 2009, p. 267.

[53] A. Rakotomamonjy, F. Bach, S. Canu, and Y. Grandvalet, "More efficiency in multiple kernel learning," in *ICML '07: Proceedings of the 24th international conference on Machine learning*. New York, NY, USA: ACM, 2007, pp. 775–782. [Online]. Available: http://dx.doi.org/10.1145/1273496.1273594

[54] G. Mehmet and A. Ethem, "Localized multiple kernel learning," in *Proceedings of the 25th international conference on Machine learning*, ser. ICML '08, 2008, pp. 352–359.

[55] M. J. McMahon, O. S. Packer, and D. M. Dacey, "The classical receptive field surround of primate parasol ganglion cells is mediated primarily by a non-gabaergic pathway," *J Neurosci*, vol. 24, no. 15, pp. 3736–45, 2004.

[56] A. Laine and J. Fan, "Texture classification by wavelet packet signatures," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 15, no. 11, pp. 1186 –1191, nov 1993.

[57] M. Unser, "Texture classification and segmentation using wavelet frames," *Image Processing, IEEE Transactions on*, vol. 4, no. 11, pp. 1549 –1560, nov 1995.

[58] R. Duda, P. Hart, and D. G. Stork, *Pattern Classification*. John Wiley & Sons, 2001.

[59] K. Huang and S. Aviyente, "Wavelet feature selection for image classification," *Image Processing, IEEE Transactions on*, vol. 17, no. 9, pp. 1709 – 1720, sept. 2008.

[60] S. Nilufar, N. Ray, and H. Zhang, "Optimum kernel function design from scale space features for object detection," *IEEE ICIP*, 2009.

[61] O. L. Mangasarian, "Exact 1-norm support vector machines via unconstrained convex differentiable minimization," *J. Mach. Learn. Res*, pp. 1517–1530, 2006.

[62] N. Usunier, M. reza Amini, and P. Gallinari, "Generalization error bounds for classifiers trained with interdependent data," in *Advances in Neural Information Processing Systems 18 (NIPS 2005*. MIT Press, 2006, pp. 1369–1376.

[63] I. Levner and H. Zhang, "Classification-driven watershed segmentation," *Image Processing, IEEE Transactions on*, vol. 16, no. 5, pp. 1437–1445, 2007.

[64] D. P. Mukherjee, Y. Potapovich, I. Levner, and H. Zhang, "Ore image segmentation by learning image and shape features," *Pattern Recognition Letters*, vol. 30, no. 6, pp. 615–622, 2009.

[65] Z. H. Shi, J. and N. Ray, "Solidity based local threshold for oil sand image segmentation," *IEEE International Conference on Image Processing (ICIP)*, 2009.

[66] N. Ray, B. Saha, and S. Acton, "Oil sand image segmentation using the inclusion filter," *IEEE ICIP*, 2008.

[67] Z. Wang and H. Zhang, "Object detection with multiple motion models," *Asian Conference on Computer Vision*, 2009.

[68] B. U. Toreyin, Y. Dedeoglu, and A. E. Cetin, "Contour based smoke detection in video using wavelets," in *EUSIPCO*, 2006.

[69] D. Kim and Y.-F. Wang, "Smoke detection in video," in *Proceedings of the 2009 WRI World Congress on Computer Science and Information Engineering*. Washington, DC, USA: IEEE Computer Society, 2009, pp. 759–763.

[70] K. Terada and N. Fujiwara, "Extraction of fire smoke region using fractal cording image," *IEEE Transactions on Industry Applications*, vol. 125, pp. 808–814, 2005.

[71] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *In CVPR*, 2005, pp. 886–893.

[72] B.Fulkerson, A. Vedaldi, and S. Soatto, "Localizing objects with smart dictionaries," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2008.

[73] A. Vedaldi, V. Gulshan, M. Varma, and A. Zisserman, "Multiple kernels for object detection," in *Proceedings of the International Conference on Computer Vision*, September 2009.

[74] C. C. Veropoulos K. and C. N., "Controlling the sensitivity of support vector machines," *Proceedings of the International Joint Conference on AI*, pp. 55–60, 1999.

[75] V. L. Durkalski, Y. Y. Palesch, S. R. Lipsitz, and P. F. Rust, "Analysis of clustered matched-pair data," *Statistics in Medicine*, vol. 22, p. 24172428, 2003.

[76] I. Selesnick, R. Baraniuk, and N. Kingsbury, "The dual-tree complex wavelet transform," *Signal Processing Magazine, IEEE*, vol. 22, no. 6, pp. 123 – 151, nov. 2005.

[77] R. Coifman and M. Wickerhauser, "Entropy-based algorithms for best basis selection," *Information Theory, IEEE Transactions on*, vol. 38, no. 2, pp. 713 –718, mar 1992.

[78] T. Chang and C.-C. Kuo, "Texture analysis and classification with tree-structured wavelet transform," *Image Processing, IEEE Transactions on*, vol. 2, no. 4, pp. 429 –441, oct 1993.

[79] M. Acharyya and M. Kundu, "Adaptive basis selection for multi texture segmentation by m-band wavelet packet frames," in *Image Processing, 2001. Proceedings. 2001 International Conference on*, vol. 2, oct 2001, pp. 622 –625 vol.2.

[80] N. Saito and R. R. Coifman, "Local discriminant bases," in *Wavelet Applications in Signal and Image Processing II, volume 2303 of SPIE Proceedings*, 1994, pp. 2–14.

[81] N. Rajpoot, "Local discriminant wavelet packet basis for texture classification," in *In: SPIE Wavelets X*, 2003.

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

[83] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in *CVPR 2001*, 2001, pp. 511–518.

Appendix A

Appendix

A.1 Modification of 1-norm SVM for non-negative Weights

Following [61] 1-norm linear SVM for the binary classification problem can be written as:

$$\underset{z,w,w_0}{min} \mathrm{C} \left\| Z \right\|_1 + \left\| w \right\|_1 \;\; s.t., \;\; D(Aw - ew_0) + z \geq e, \; z \geq 0. \qquad \text{(A.1)}$$

where $m \times n$ matrix A represent m points in R^n to be separated by maximal margin with a separating plane: $x^T w = w_0$. D is a $m \times m$ diagonal matrix with elements $D_{ii} = +1, or - 1$ according to the class of each row of A. e is a vector all ones in equation A.1. The objective term $\left\| w \right\|_1$ minimizes the classification error weighted with the positive user tuning parameter C. The term $\left\| w \right\|_1$ called lasso penalty maximizes the 1-norm margin between the positive and negative samples. In our case the vector of weights $w = [w_{\sigma_1}, \cdots, w_{\sigma_N}]^T$ are non-negative, additionally. So, one can rewrite equation A.1 as:

$$\underset{z,w,w_0}{min} \mathrm{C} e^T z + e^T \;\; s.t., \;\;\; D(Aw - ew_0) + z \geq e, \; z, w \geq 0. \qquad \text{(A.2)}$$

The above linear program in A.2 is solvable because it is feasible and its objective function is bounded below by zero. For a fairly large-scale problem, a standard package, such as CPLEX, is able to solve the linear program A.2. Alternatively, one can also apply the unconstrained Newton optimization method [61] defined for a large-scale linear programming of the form such as A.2.