

IMAGE CLASSIFICATION USING GABOR FILTERS AND MACHINE LEARNING

By

SEBASTIAN BERISHA

A Thesis Submitted to the Graduate Faculty of

WAKE FOREST UNIVERSITY

in Partial Fulfillment of the Requirements

for the Degree of

MASTER OF SCIENCE

in the Department of Computer Science

May 2009

Winston-Salem, North Carolina

Approved By:

Victor Paúl Pauca, Ph.D., Advisor

Examining Committee:

David J. John, Ph.D., Chairperson

Robert J. Plemmons, Ph.D.

Acknowledgments

I would like to thank my advisor, Dr. V. Paúl Pauca for his invaluable support, patience, advice, guidance, and friendship. He transformed my research from being Turing recognizable to Turing decidable (which eventually ended in an accept state) by encouraging and challenging me throughout my research and academic work. Words are not enough to express my profound gratitude for his help and motivation. He should be a role model for other professors. I gratefully acknowledge Dr. Robert J. Plemmons for introducing me to image processing, and the necessary math behind it. Thanks for successfully leading the Image Processing Research group and for letting me be a part of it. I would like to thank Mary Jo, Theresa, Sofia, Francesca, and Victor for their kindness and hospitality. I would also like to thank Dr. David J. John for encouraging me to get involved with research from the beginning of my graduate studies and for making sure I meet all the administrative deadlines and requirements.

Thanks to Dr. William H. Turkett for introducing me to probabilistic reasoning and machine learning. I would like to thank Dr. Errin W. Fulp for letting me live in his networking lab (for free!). Thanks to Rita Johnson and Paul Whitener for their administrative, technical support, and friendship. I would also like to thank my friends Harry, Matt, Ranjan, Satria, Saurav, and Shuai. Without their personal support and fun times we have had together my life as a graduate student would not have been the same. More specifically, I would like to thank Ashish Tapdiya for all of his help both in the classes we shared as well as in general knowledge about computer science. Thanks to my friend Astrit for not letting me be bored on the weekends and to his family for their personal support.

A special thank you to my friendship family Anita and Jim Conrad. Anita, thank you for the delicious cakes, and for your continuous kindness and love. Jim, thank you for your efforts in making me be more pet friendly and for encouraging me to be a Deacon fan! Anita and Jim have been my second family and I will always be indebted to the love and friendship they have given me.

I dedicate this thesis to my parents and my brother Albin. Mom and Dad, I thank you for raising me in a well-educated way, for your ceaseless prayers, support, and love you have given me through the years, despite the distance that has separated us for quite some time now. I am mostly thankful to you for teaching me the importance of not giving up, regardless of the sweat and tears. I might have not been your perfect son, but I am improving. Bini thanks for your support and for always being proud of your big brother.

Finally, I want to thank God for his mercy and grace. I have been blessed to have his protection, help, and love throughout my life!

Table of Contents

List of Tables.....	v
List of Figures.....	vi
Abstract	viii
Chapter 1 Introduction.....	1
1.0.1 Contributions of this Thesis	5
Chapter 2 Gabor Filters.....	7
2.1 Time/Space and Frequency Representation	7
2.2 Gabor Filters	8
2.2.1 1-D Gabor Filter	8
2.2.2 2-D Gabor filter	13
2.2.3 Discrete 2-D Gabor Filters	19
2.3 Invariant Object Recognition	23
2.4 Simple Gabor Feature Space	25
Chapter 3 Bayesian Networks and Linear Discriminants.....	27
3.1 Bayesian Networks	27
3.2 Probabilistic Classification	29
3.3 Linear Discriminants	31
3.4 Support Vector Machines (SVM)	33
3.4.1 SVM ^{light} and Kernels	34
Chapter 4 Classification Methodology.....	37
4.1 Distance-based Classification Method	37
4.1.1 Similarity Measures	38
4.1.2 Selection of Gabor Feature Space	40
4.1.3 Selection of Separation Threshold	41
4.2 Probabilistic Classification Method	42
4.3 SVM-based Classification Method	45

Chapter 5	Experimental Results	48
5.1	Observation Experiment	49
5.2	Manjunath and Ma vs. Euclidean distance	53
5.3	Classification Threshold	56
5.4	Classification results using the distance-based classification method	57
5.5	Results for the probabilistic classification method	61
5.6	Classification Results using the SVM-based classification method	65
5.7	Robustness Testing	67
Chapter 6	Conclusions and Future Research	72
6.1	Discussion	72
6.2	Future Work	75
References		78
Vita		81

List of Tables

5.1	Confusion Matrix for classification using the Manjunath and Ma distance	59
5.2	Confusion Matrix for classification results using the Euclidean distance	59
5.3	Confusion Matrix for the PCM method (using 30,614 features, tested on Figure 5.9 (a), (b)).	61
5.4	Confusion Matrix for the PCM distance method (using 18,457 features, tested on Figure 5.9 (a), (b)). Average classification time per image(of size 10 x 10): 156.17ms	62
5.5	Confusion Matrix for the PCM method (using 9,572 features, tested on Figure 5.9 (a), (b)). Average classification per time image(of size 10 x 10): 96.96ms	62
5.6	Confusion Matrix for the PCM method (using 9,572 features, tested on all the images of Figure 5.9). Average classification per time image(of size 10 x 10): 96.96ms	62
5.7	Confusion Matrix for the SVM method. Average classification time per image(of size 10x10): 52.4ms	65
5.8	Confusion Matrix for the PCM method for the new data set before retraining.	69
5.9	Confusion Matrix for the PCM method for the new data set after re-training.	69
5.10	Confusion Matrix for the SVM method for the new data set before retraining.	69
5.11	Confusion Matrix for the SVM method for the new data set after re-training.	70

List of Figures

1.1	Panchromatic image of mountainous areas of Peru	2
1.2	Raw images in the left and the binary images resulting from classification in the right	4
2.1	Visual representation of the Gabor filter	8
2.2	1-D Gabor filter functions in the time and frequency domain with different values for frequency and width. The left side represents functions in the time domain(showing the real part(—) and imaginary part(- - -) of the Gabor functions), and the right side shows functions in the frequency domain.	11
2.3	2-D Gabor filter functions in the space and spatial-frequency domains with different frequency values, $\gamma = \eta = 1$, and $\theta = 0^0$. Shown from left to right: real part, imaginary part, magnitude, and Fourier transform.	16
2.4	Gabor filters constructed with different parameters. From left to right: real part, imaginary part, and frequency space. The γ and η parameters are fixed, $\gamma = \eta = 1$	22
2.5	Lenna image	23
2.6	Gabor filter responses after doing convolution of the Lenna image with different Gabor filters. From left to right: real part, imaginary part, and magnitude of the response	24
3.1	Bayesian networks. (a) Naive Bayes classifier; (b) A more complex Bayesian classifier [26]	30
3.2	Linear discriminant planes	32
3.3	Finding the best plane by maximizing the margin.	33
3.4	Example requiring a quadratic discriminant [3]	35
5.1	One of the images used for experiments	48
5.2	The images used for the observation experiment	49
5.3	Histograms of distance measures between Gabor filter responses of images belonging to two different classes. The figures compare the differences between the magnitudes of Gabor responses(using the MM distance) for tree vs non-tree images. Parameters for Gabor filters are (maximum frequency, number of frequencies, number of orientations, 0 - no lightness normalization and 1 - with lightness normalization).	50
5.4	The images used for tree vs tree classification	51

5.5	Using Gabor Filters to compare tree vs tree areas for images taken from Figure 5.4 (μ - mean , σ - standard deviation, t - computational time in seconds)	52
5.6	Images used in the training part.	53
5.7	Comparing the separation between Gabor feature spaces of images pertaining to different classes. Two methods are used for computing the distance between Gabor feature spaces constructed using different Gabor filter parameters. Blue histogram represents non-tree vs non-tree. Red histogram represents non-tree vs tree. MM = Manjunath and Ma distance, E = Euclidean distance. We apply illumination normalization.	55
5.8	ROC curve	56
5.9	New images used for testing our classification methods.	57
5.10	10x10 tree and non-tree images	58
5.11	Gabor filter responses of the 10×10 sub-images of tree and non-tree areas, for different Gabor filter parameter values . f = frequency value, o = orientation degree, t = tree, nt = non-tree	58
5.12	Original images(taken from Figure 5.1 and their classified versions using the Gabor feature extraction method)	60
5.13	Original images(taken from Figure 5.1) and their classified versions using the PCM classification method	64
5.14	Original images (taken from Figure 5.1) and their classified versions using the SVM classification method.	66
5.15	New data set used for testing. Sub-images (a) through (i) consist of non-tree areas with different features. Sub-images (j) through (r) consist of tree regions with different features.	68
5.16	The histogram of the Euclidean distances for the new data set. The blue part shows the distances non-tree vs non-tree, the red part shows the distance values between non-tree and tree.	70
5.17	The original image on the left and the binary image on the right using the retrained PCM.	71

Abstract

Feature extraction and classification are important areas of research in image processing and computer vision with a myriad of applications in science and industry. The focus of this work is on the robust classification of tree and non-tree areas in aerial imagery of the eastern Andes mountains in Peru. Knowledge of this type of information has strong implications in the study of the effect of climate change on the environment and its conservation. Drawing from recent work on human iris pattern identification, we propose a classification methodology based on Gabor feature space representation of aerial imagery, where the two object classes may be well separated. We evaluate two different distance metrics to discern class separation and use the receiver operating characteristic curve to determine an optimum classification threshold. We then build upon our Gabor representation technique by proposing two additional classification methods based on naive Bayes' and support vector machine classifiers. Mutual information is used for reducing redundant Gabor features not carrying sufficient object information. Extensive experimentation using real aerial imagery of the Peruvian Andes shows that our approach can provide highly accurate classification, even in the presence of variable illumination, different land features and changing topology. The issue of finding an optimal Gabor feature space where object classes are optimally represented is still a challenging problem to be resolved.

Chapter 1: Introduction

Feature extraction and classification are important areas of research in image processing and computer vision. Applications abound in science and engineering: from the classification of information classes (e.g. crops, forest types, tree species, minerals, etc.), material types in man-made space objects, to human facial features for face and iris recognition [18], [20], [6]. The application focus of this thesis work is on the development of robust classification strategies for information classes, such as tree and non-tree regions, in aerial imagery of the eastern Andes mountains in Peru.

The east slope of the Andes harbors Earth's highest biodiversity and is also the area most threatened by climate change. Understanding the historical biodiversity distribution and ecosystem function of this area in relation to climate change is key in the efforts to predict future outcomes and to the design and implementation of effective conservation strategies [5]. Recently significant multi-disciplinary work is being devoted to the analysis of very large datasets, containing current and past multi-modal and multi-sensor imagery of this region [28]. Figure 1.1 shows a panchromatic image of mountainous areas in Peru taken in 1960s. More recent Quickbird panchromatic and multi-spectral satellite imagery of the same regions are also available.

The classification of information classes such as trees, landslides, rivers, roads, etc. pertaining to panchromatic image datasets is a non-trivial problem. Varying observer direction, spatial resolution, illumination, and topology changes make the classification problem hard. Methods based on single-pixel intensity are ineffective for classification purposes of this type of data, and methods that account for spatial relationships in addition to intensity must be considered.

In this thesis we explore the representation of image data in other spaces for improved classification. Past success obtained in the Gabor representation of human iris imagery [6] and the work done at Wake Forest [29, 22] related to iris recognition led to the consideration of Gabor spaces for this thesis. A Gabor filter is a linear filter

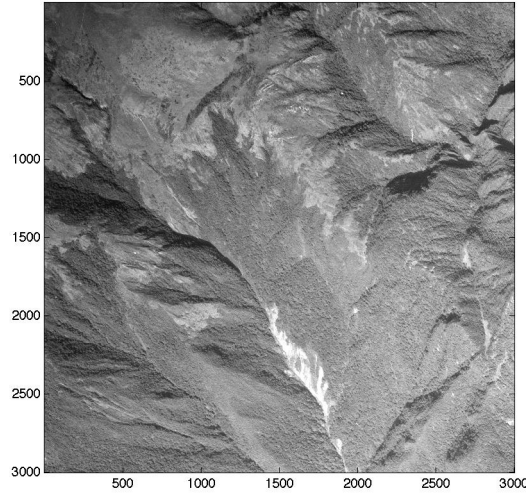


Figure 1.1: Panchromatic image of mountainous areas of Peru

which can be defined as a harmonic/sinusoidal function multiplied by a Gaussian function. These filters can be considered as edge, line, and shape detectors. An important characteristic of Gabor filters is that they can be tuned with different frequencies and orientations.

The seminal work on Gabor elementary functions was conducted in 1946 by Dennis Gabor [10]. Gabor based his work on the mechanical wave theory and in Heisenberg's uncertainty principle. He proposed the representation of signals as a combination of these elementary functions. Subsequent work has analyzed the specific features obtained using Gabor filters, global Gabor features and fundamental frequency Gabor features [16].

One of the main advantages of 2-D Gabor filters is their association with a partic-

ular location in space. Daugman [7] has shown that Gabor feature space representation minimizes the joint 2-D uncertainty principle in space and frequency. Moreover, evidence has been presented that the 2D receptive-field profiles of simple cells in mammalian visual cortex are well described by members of the Gabor 2-D filter family [7]. The advantage of using Gabor filters appears to stand in the ability of such filters to provide some degree of invariance to intensity, translation, and orientation [30]. For these reasons, Gabor filters are finding increasing usage in many applications, such as fingerprint image enhancement [32], texture classification and segmentation [4], image recognition [19], and motion tracking [21]. In this thesis Gabor filters are used to extract spatial features from information classes for subsequent classification purposes.

A naive Bayes classifier is a simple but highly effective probabilistic classifier based on the application of Bayes' theorem. A naive Bayes classifier assumes that the presence of a particular feature of a class is unrelated to the presence of any other feature. As applied to aerial image classification this probabilistic method assumes that features are conditionally independent of each other, given knowledge of their information class. We use a naive Bayes classifier together with mutual information (to reduce the computation cost in the classification process) for probabilistic classification of aerial imagery. Mutual information is a measure of the mutual dependence of two variables. We will see from the results obtained that the Bayes independence assumption is not unreasonable for this problem. Probabilistic learning methods have been widely applied for classification problems, for example in the classification of junk E-mail from good E-mail [26]. Mutual information theory has also been used to select informative Gabor features for face recognition [27]. Using mutual information reduces the computation cost in the classification process.

Like probabilistic methods, support vector machines are a set of supervised learn-

ing techniques used for classification tasks. SVMs are also finding increasing number of applications, a recent interesting application include the problem of on-road vehicle detection from rear views of gray-scale images [30], and network stream classification task [17]. From a geometric point of view an SVM constructs a separating hyperplane between two datasets. The idea is to find the hyperplane which maximizes the margin between the two datasets. SVMs have been shown to perform well in terms of both classification accuracy and computational speed [3] and can be used as a general methodology for many types of classification, regression, and novelty detection problems.

In this thesis we apply linear SVM-based classification, where conditional probabilities from naive Bayes classifier and a mutual information feature vector are used as input. Figure 1.2 shows one of the results obtained from SVM-based classification of Andes aerial imagery.

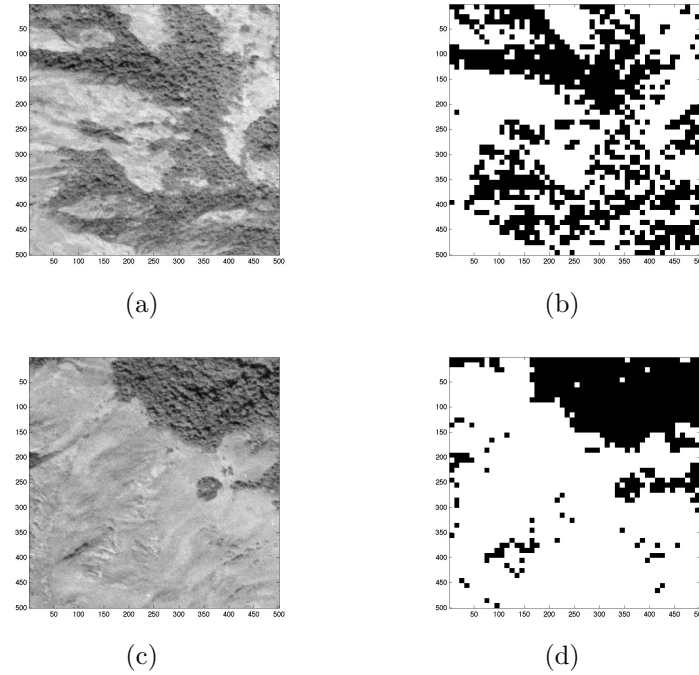


Figure 1.2: Raw images in the left and the binary images resulting from classification in the right

The motivation behind this research is to develop a robust aerial image classification method suitable for use in biological research related to tree area changes over a period of time in the mountainous areas of Peru. To our knowledge this is the first effort to demonstrate the applicability and efficacy of Gabor-based probabilistic and machine learning techniques to studies regarding biodiversity, climate, and conservation of this region.

1.0.1 Contributions of this Thesis

The main contributions of this thesis are:

- A representation of information classes in Gabor feature space for feature extraction.
- Probabilistic and support vector machine classification in Gabor feature space.
- Evaluation of two total similarity measures for Gabor feature space selection.
- The usage of the receiver operating characteristic curve (ROC) for estimation of Gabor feature space separation threshold.
- Validation of the use of magnitude only (no phase) of Gabor feature responses for training a naive Bayes classifier.
- Demonstration that binary classification of aerial imagery can be effectively accomplished in Gabor feature space using only linear classifiers.

The remainder of this thesis is structured as follows. Gabor filters are defined and the mathematical theory behind them is explained in chapter 2. In chapter 3 preliminary background of the mathematical theory behind Bayesian networks, mutual information, and support vector machines is presented. The proposed classification

methods of this thesis are presented in chapter 4. In chapter 5 the experimental results demonstrating the effectiveness of the three proposed classification methods are presented. Conclusions and future work are presented in chapter 6.

Chapter 2: Gabor Filters

2.1 Time/Space and Frequency Representation

Gabor filters were originally proposed for the representation of signals as a function of both time and frequency. For simplicity, we start with a brief overview of the Gabor transform in 1-D time and frequency domains. The frequency domain has an essential role in signal and image processing. Let $g(t)$ denote a complex function. Then the 1-D Fourier transform of $g(t)$ is defined as [12]:

$$G(f) = \mathcal{F}\{g(t)\} = \int g(t)e^{-i2\pi ft} dt, \quad (2.1)$$

where f denotes frequency and $g(t)$ can be recovered using the 1-D inverse Fourier transform:

$$g(t) = \mathcal{F}^{-1}\{G(f)\} = \int G(f)e^{i2\pi ft} df. \quad (2.2)$$

For the 2-D case the Fourier transform is defined by the linear transformation [12]:

$$G(u, v) = \iint g(x, y)e^{-i2\pi(xu+yv)} dx dy, \quad (2.3)$$

and analogously the 2-D inverse Fourier transform is defined as:

$$g(x, y) = \iint G(u, v)e^{i2\pi(xu+yv)} du dv. \quad (2.4)$$

The complex frequencies $e^{-i2\pi ft}$ provide a continuous spectrum for the transform operator $\mathcal{F}$. We will see later that these complex frequencies are also part of the definition of Gabor functions.

2.2 Gabor Filters

2.2.1 1-D Gabor Filter

The Gabor filter consists of the product between an elliptical Gaussian and a sinusoidal. It can be assumed for simplicity that the rotation of the Gaussian and the sinusoidal are the same, i.e. their principal axes correspond to the x and y axes. The

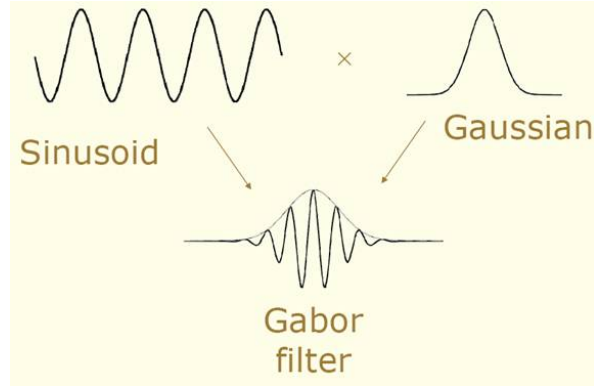


Figure 2.1: Visual representation of the Gabor filter

1-D Gabor elementary function can be defined as the product of a pulse of a form of a probability function with a harmonic oscillation of any frequency:

$$g(t) = e^{-\alpha^2(t-t_0)^2} e^{i2\pi f_0 t + \phi}, \quad (2.5)$$

where α is the time duration and bandwidth of the Gaussian envelope, t_0 denotes its centroid, f_0 is the frequency of the sinusoidal or carrier, and ϕ denotes phase shift.

The 1-D Fourier transform of the above Gabor elementary function is:

$$G(f) = \sqrt{\frac{\pi}{\alpha^2}} e^{-(\frac{\pi}{\alpha})^2(f-f_0)^2} e^{-i2\pi(f-f_0)t + \phi} \quad (2.6)$$

The Gabor elementary function in (2.5) is concentrated near the time instant t_0 and since for convolution an origin-centered filter is preferred, we set $t_0 = 0$ and $\phi = 0$.

Thus we consider Gabor elementary functions of the form:

$$g(t) = e^{-\alpha^2 t^2} e^{i2\pi f_0 t}. \quad (2.7)$$

These functions are most often as complex linear filters. To do so we calculate the 1-D filter response of a function ξ at some location t_1 with convolution as follows:

$$\begin{aligned}
r(t_1) &= g(t_1) \otimes \xi(t_1) \\
&= \int g(t_1 - t) \xi(t) dt \\
&= \int e^{-\alpha^2(t_1-t)^2} e^{i2\pi f_0(t_1-t)} \xi(t) dt \\
&= e^{i2\pi f_0 t_1} \int e^{-\alpha^2(t_1-t)^2} \xi(t) e^{-i2\pi f_0 t} dt
\end{aligned} \tag{2.8}$$

1-D Normalized Form

One problem with the filter equation in (2.7) is the lack of control of α , the time duration and bandwidth of the filter, as this parameter is the same regardless of the central frequency f_0 . We employ an approach from multi-resolution analysis and wavelets suggested in [16] to guarantee that the filters on different frequencies are scaled versions of each other; that is, the time duration of the filter is connected to the central frequency as follows :

$$\alpha = \frac{|f_0|}{\gamma}, \tag{2.9}$$

where f_0 is the central frequency of the filter and γ controls the effective width of the filter [16]. Substituting (2.9) for α into equation (2.7) the following is obtained:

$$g(t) = e^{-(\frac{|f_0|}{\gamma})^2 t^2} e^{i2\pi f_0 t}. \tag{2.10}$$

From equation (2.6) we can see that the peak filter response occurs at $f = f_0$, i.e. $\max_f |G(f)| = \sqrt{\frac{\pi}{\alpha^2}}$. Thus $\sqrt{\frac{\alpha^2}{\pi}}$ can be used as a normalization factor in the definition of Gabor filters. Thus the normalized 1-D Gabor filter function is defined as:

$$g(t) = \sqrt{\frac{\alpha^2}{\pi}} e^{-(\frac{|f_0|}{\gamma})^2 t^2} e^{i2\pi f_0 t} = \frac{|f_0|}{\gamma \sqrt{\pi}} e^{-(\frac{|f_0|}{\gamma})^2 t^2} e^{i2\pi f_0 t}. \tag{2.11}$$

For the Fourier domain $g(f)$ is given by:

$$G(f) = e^{-\frac{\gamma\pi}{f_0}(f-f_0)^2}. \quad (2.12)$$

Figure 2.2 shows the Gabor filter plots in time and frequency domains for different values of frequency f and width γ . The parameter γ can be used to adjust the effective width of the Gabor filter, by increasing γ the filter spreads in the time domain and shrinks in the frequency domain and vice versa (See Figures 2.2 c-d and Figures 2.2 e-f). The center frequency f_0 can also change the width of the filter when γ is fixed.

Gabor Expansion

Gabor filters can be used as basis functions to represent functions under mild conditions [10]. Specifically a signal $s(t)$ can be represented as a sum of Gabor elementary functions $g_{kl}(t)$, multiplied with specific Gabor expansion coefficients a_{kl} , as:

$$s(t) = \sum a_{kl}g_{kl}(t), \quad (2.13)$$

where $g_{kl}(t) = e^{-\alpha^2(t-kt_0)^2} e^{i2\pi lf_0t+\phi}$, and k and l denote time and frequency shifts of Gabor functions, respectively. It is not difficult to compute expansion coefficients for orthogonal wavelets and a biorthogonal function set is usually used in the analysis. However for non-orthogonal Gabor functions such as those in (2.13) the computation of the expansion coefficients is more computationally challenging. Bastiaans [2] has introduced more stable solutions utilizing window functions and the Zak transform for the calculation of expansion coefficients. The Gabor expansion has been successfully applied to time-frequency signal analysis, for various applications, where it has significant advantages compared to other time-frequency methods [25].

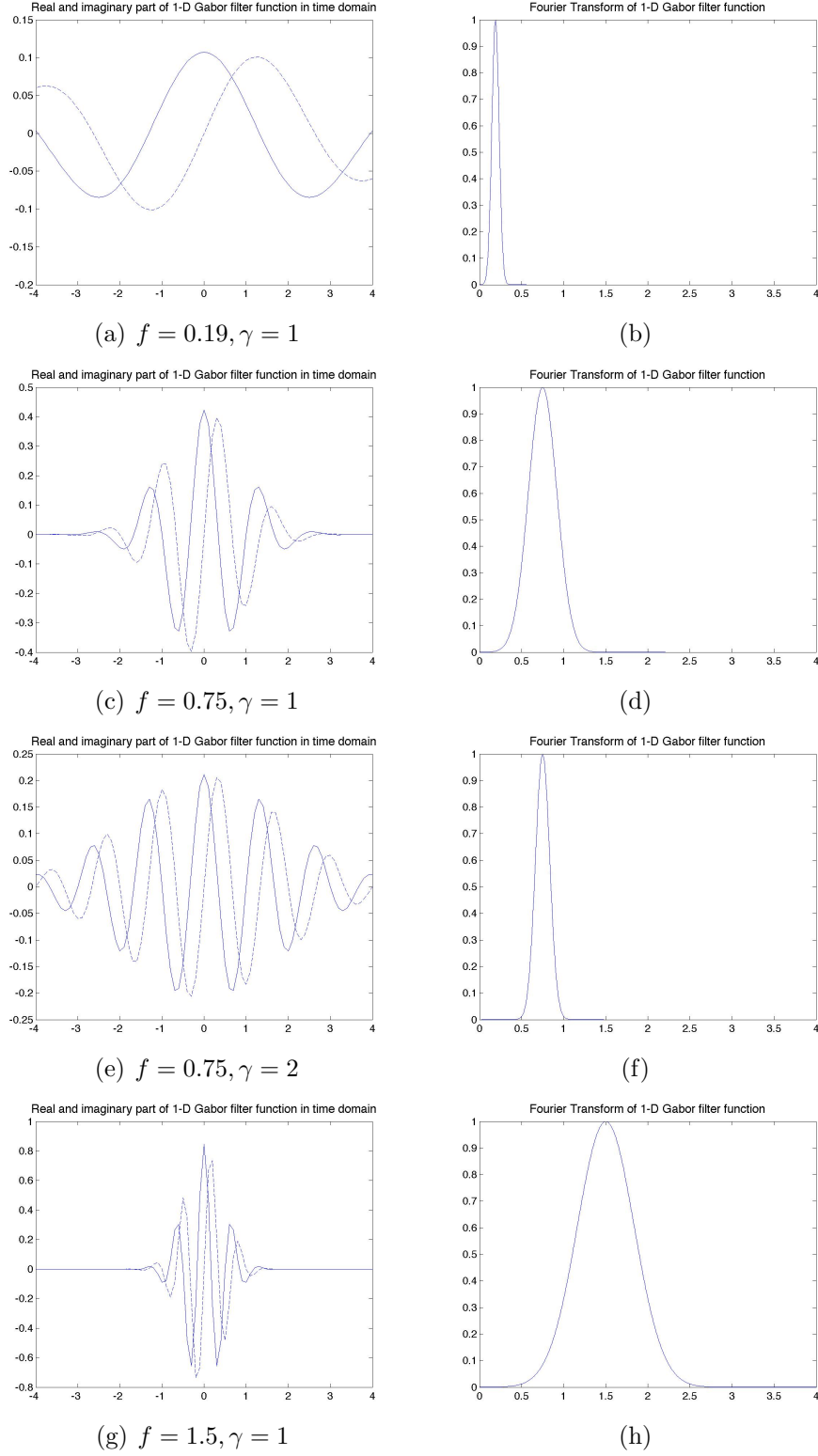


Figure 2.2: 1-D Gabor filter functions in the time and frequency domain with different values for frequency and width. The left side represents functions in the time domain (showing the real part (—) and imaginary part (---) of the Gabor functions), and the right side shows functions in the frequency domain.

Locality in Space-Frequency Domain

There is always a trade-off associated with the resolution of signals or images in the time/spatial and frequency domains. This trade-off is formally specified for general functions in terms of Heisenberg's uncertainty principle. Gabor in his famous paper on theory of communication [10] showed that the family of signals of the form (2.5) achieve the theoretical limit of joint uncertainty in time and frequency, optimizing the trade-off on both time and frequency domains.

Let $f(x)$ be a 1-D complex signal. For simplicity we assume that f is centered at the origin, such that $x = 0$. Then the standard measure of effective width, Δx , is given by:

$$\Delta x = \sqrt{\frac{\int f f^* x^2 dx}{\int f f^* dx}}, \quad (2.14)$$

where $f^*(x)$ is the complex conjugate of $f(x)$ [7]. In the above equation the numerator represents the variance or the second moment of f , and $f f^*$ corresponds to the energy distribution and its integral (the denominator) normalizes the measure of the effective width so it is independent of the amplitude of $f(x)$. If $f(x)$ has Fourier transform then its effective bandwidth in frequency space, Δu , can be defined in an analogous manner as:

$$\Delta u = \sqrt{\frac{\int F F^* u^2 du}{\int F F^* du}}. \quad (2.15)$$

For a general complex signal the uncertainty principle specifies a fundamental lower bound [10] on the product of equations (2.14) and (2.15), i.e.

$$\Delta x \Delta u \geq \frac{1}{4\pi} \quad (2.16)$$

Gabor showed that 1-D Gabor elementary functions of equation (2.5) achieve this lower bound, occupying the minimum area $\Delta x \Delta u = \frac{1}{4\pi}$. This fact provides 1-D

Gabor filters with the very important property of being able to extract information locally.

2.2.2 2-D Gabor filter

The normalized 1-D Gabor filter in equation (2.11) can be generalized to two dimensions. In the 2-D form the time variable t is replaced by spatial coordinates (x, y) in the spatial domain and the frequency variable f is replaced by the frequency variables (u, v) in the frequency domain. 2-D Gabor filter functions were mainly derived for use in image processing, especially for feature extraction and texture analysis. The 2-D Gabor function is typically defined in space domain as:

$$g(x, y) = g(x, y; f_0, \theta) = e^{-(\alpha^2 x_p^2 + \beta^2 y_p^2)} e^{i2\pi f_0 x_p}, \quad (2.17)$$

where $x_p = x \cos \theta + y \sin \theta$, $y_p = -x \sin \theta + y \cos \theta$, and θ is the rotation angle of the Gaussian major axis and the plane wave (sinusoidal). Following [16], to guarantee that filters in different frequencies are scaled versions of each other as in the 1-D case, we substitute $\alpha = \frac{|f_0|}{\gamma}$ and $\beta = \frac{|f_0|}{\eta}$. Now γ and η control the bandwidth of the filter along the x and y axis respectively. The normalized compact closed form of the 2-d Gabor filter function is thus given by:

$$g(x, y) = \frac{f_0^2}{\pi\gamma\eta} e^{-(\frac{f_0^2}{\gamma^2} x_p^2 + \frac{f_0^2}{\eta^2} y_p^2)} e^{i2\pi f_0 x_p}. \quad (2.18)$$

A 2-D Gabor filter is a product of a 2-D Gaussian with oriented sinusoids, as can be seen in Figure 2.1 (page 8).

Locality

Let $f(x, y)$ be an arbitrary 2-D complex function. We assume $f(x, y)$ is centered at (x_0, y_0) . Then as in the 1-D case, the following effective widths in space and frequency

domains can be calculated:

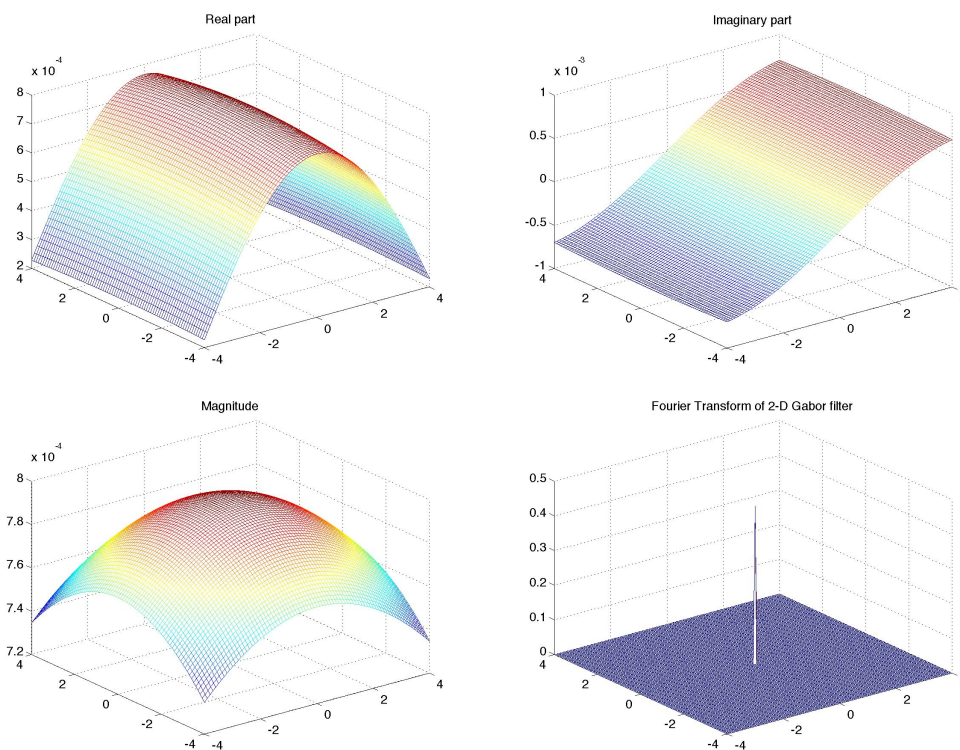
$$\begin{aligned}
\Delta x &= \sqrt{\frac{\iint (x - x_0)^2 f(x, y) f^*(x, y) dx dy}{\iint f(x, y) f^*(x, y) dx dy}}, \\
\Delta y &= \sqrt{\frac{\iint (y - y_0)^2 f(x, y) f^*(x, y) dx dy}{\iint f(x, y) f^*(x, y) dx dy}}, \\
\Delta u &= \sqrt{\frac{\iint (u - u_0)^2 F(u, v) F^*(u, v) du dv}{\iint F(u, v) F^*(u, v) du dv}}, \\
\Delta v &= \sqrt{\frac{\iint (v - v_0)^2 F(u, v) F^*(u, v) du dv}{\iint F(u, v) F^*(u, v) du dv}},
\end{aligned} \tag{2.19}$$

where $F(u, v)$ is the 2-D Fourier Transform (centered at (u_0, v_0) of $f(x, y)$). Applying the 1-D uncertainty principle it can be shown that $\Delta x \Delta u \geq \frac{1}{4\pi}$, and $\Delta y \Delta v \geq \frac{1}{4\pi}$. Then uncertainty principle for the 2-D case says that the joint resolution that can be achieved by any 2-D function in the space and frequency domains is constrained by a fundamental lower limit [7]:

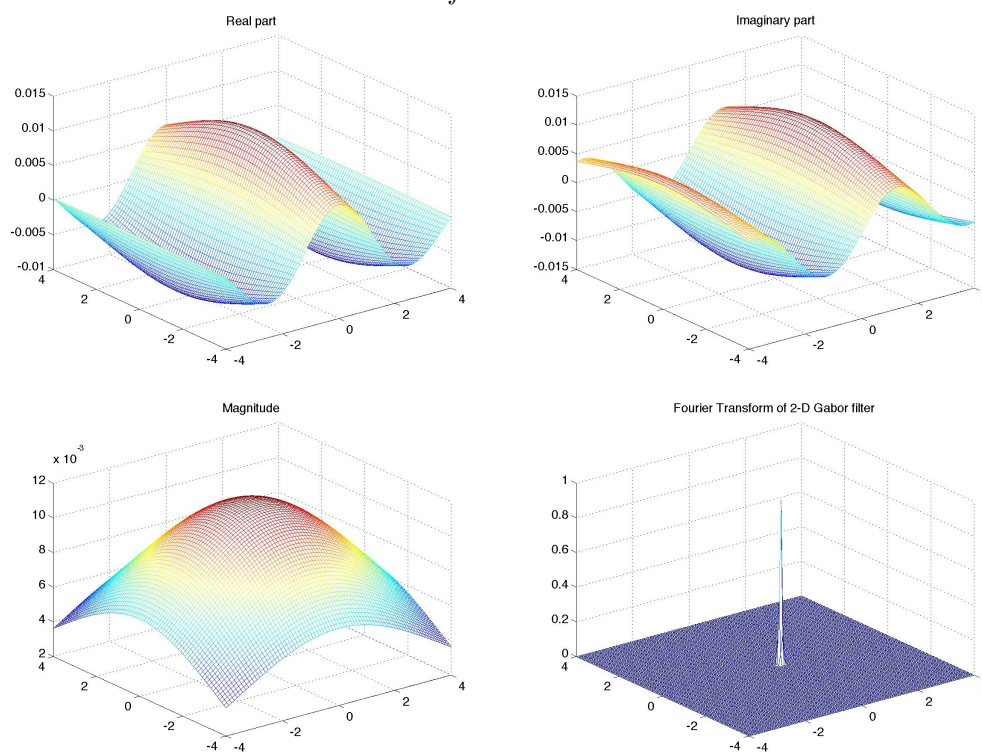
$$\Delta x \Delta y \Delta u \Delta v \geq \frac{1}{16\pi^2}. \tag{2.20}$$

Daugman [7] has shown that the family of 2-D Gabor functions occupy the minimum area in the space-frequency domain, i.e. $\Delta x \Delta y \Delta u \Delta v = \frac{1}{16\pi^2}$. This property of 2-D Gabor filters allows for extraction of local features and information from images. Gabor filters have been widely used in image processing and computer vision, mostly for texture analysis, feature extraction, segmentation and local classification.

There exist other methods which can be used for texture analysis and feature extraction, such as Fourier transform analysis. The Fourier transform can be used for quantification of homogenous periodic structure(e.g. stripes), making it specific to period and orientation.



$$f = 0.05$$



$$f = 0.19$$

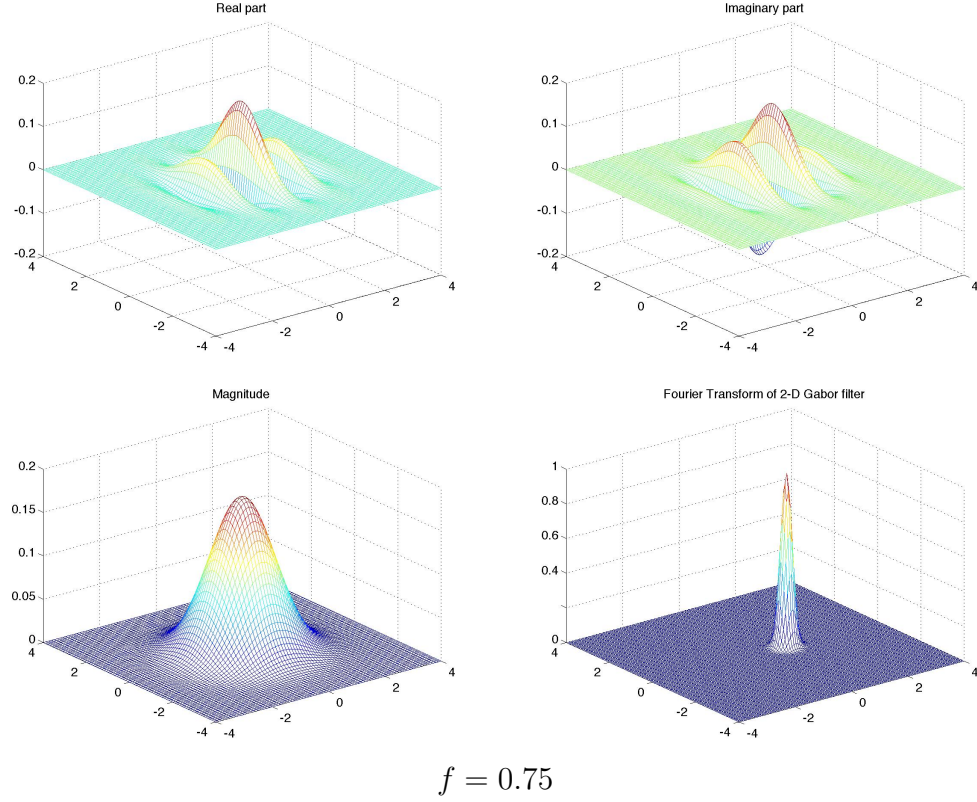


Figure 2.3: 2-D Gabor filter functions in the space and spatial-frequency domains with different frequency values, $\gamma = \eta = 1$, and $\theta = 0^0$. Shown from left to right: real part, imaginary part, magnitude, and Fourier transform.

Another advantage of using Fourier Analysis is the fast computation via the FFT algorithm. The main disadvantage however is that Fourier basis provide a completely non-local signal representation. 2-D Gabor filters provide some of the benefits of Fourier analysis, while preserving spatio-frequency locality, allowing the study of the spatial distribution of texture. They enable detection of gradual changes of texture and texture variations [20].

An appropriately designed filter will extract useful information such as spots and edges. For example, based on pixels with large magnitudes in the particular filter response, one can determine the presence of strong edges of certain orientation in different images. Filtering can be performed at different scales to find patterns of different sizes. Gabor filters induce strong responses from images containing local

components that have a particular spatial frequency and orientation. Feature extraction in the space and spatial-frequency domains can be done with 2-D Gabor filters via convolution:

$$r_s(x, y; f, \theta) = g(x, y; f, \theta) \otimes s(x, y) = \int \int g(x - x_\tau, y - y_\tau; f, \theta) s(x_\tau, y_\tau) dx_\tau dy_\tau, \quad (2.21)$$

In the following section we discuss some of the properties of Gabor filters useful for image classification. These properties apply also in the 1-D case.

Translation Invariance

One of the central properties of the Gabor filter is the translation invariance property. Let s be a translated version of some 2-D signal s_1 :

$$s_2 = s_1(x - x_1, y - y_1). \quad (2.22)$$

Then it can be shown that the Gabor response of s_2 is a translation of the Gabor response of s_1 :

$$\begin{aligned} r_{s_2}(x, y; f, \theta) &= \int \int s_2(x - x_\tau, y - y_\tau) g(x_\tau, y_\tau; f, \theta) dx_\tau dy_\tau \\ &= \int \int s_1(x - x_1 - x_\tau, y - y_1 - y_\tau) g(x_\tau, y_\tau; f, \theta) dx_\tau dy_\tau \quad (2.23) \\ &= r_{s_1}(x - x_1, y - y_1; f, \theta). \end{aligned}$$

Scale Invariance

2-D Gabor filters are also scale invariant as we show next. Let the 2-d signal s_3 be a homogeneously scaled version of the signal $s_1(x, y)$, i.e.:

$$s_3(t) = s_1(ax, ay) \quad (2.24)$$

The filter response for s_2 can be found as follows [16]:

$$\begin{aligned}
r_{s_3}(x, y; f, \theta) &= \int \int g(x - x_\tau, y - y_\tau; f, \theta) s_3(x_\tau, y_\tau) dx_\tau dy_\tau \\
&= \int \int s_3(x - x_\tau, y - y_\tau) g(x_\tau, y_\tau; f, \theta) dx_\tau dy_\tau \\
&= \int \int s_1(ax - ax_\tau, ay - ay_\tau) g(x_\tau, y_\tau; f, \theta) dx_\tau dy_\tau
\end{aligned} \tag{2.25}$$

Let $\hat{x}_\tau = ax_\tau$ and $\hat{y}_\tau = ay_\tau \Rightarrow dx_\tau = \frac{d\hat{x}_\tau}{a}$ and $dy_\tau = \frac{d\hat{y}_\tau}{a}$. Then the integrals in (2.25) can be redefined in terms of $\hat{x}_\tau$ and $\hat{y}_\tau$ as:

$$\begin{aligned}
r_{s_3}(x, y; f, \theta) &= \int \int s_1(ax - \hat{x}_\tau, ay - \hat{y}_\tau) g(\hat{x}_\tau, \hat{y}_\tau; \frac{f}{a}, \theta) d\hat{x}_\tau d\hat{y}_\tau \\
&= r_{s_1}(ax, ay; \frac{f}{a}, \theta).
\end{aligned} \tag{2.26}$$

The above equation represents the scale property of Gabor filters, i.e. the filter response of a scaled version of a signal at a particular location is equivalent to a scaled version (determined by the scaling factor of the signal) of the filter response.

Rotation and illumination invariance

Let $s_2(x, y)$ be a rotated version of an image $s_1(x, y)$, where s_1 is rotated anti-clockwise around the spatial location (x_0, y_0) by an angle ϕ , such that:

$$\begin{aligned}
s_2(x, y) &= s_1(\hat{x}, \hat{y}) \\
\hat{x} &= (x - x_0) \cos \phi + (y - y_0) \sin \phi + x_0 \\
\hat{y} &= -(x - x_0) \sin \phi + (y - y_0) \cos \phi + y_0
\end{aligned}$$

The Gabor filter response of the rotated image in terms of the rotated parameters is:

$$r_{s_2}(x_0, y_0; f, \theta) = \int \int g(x_0 - x_\tau, y_0 - y_\tau; f, \theta) s_2(x_\tau, y_\tau) dx_\tau dy_\tau \tag{2.27}$$

By changing the integration axes to (x'_τ, y'_τ) which are correspondingly rotated around the point (x_0, y_0) by the angle $-\phi$ equation (2.27) can be rewritten as:

$$\int \int g(x_0 - \hat{x}_\tau, y_0 - \hat{y}_\tau; f, \theta - \phi) s_1(x'_\tau, y'_\tau) dx'_\tau dy'_\tau = r_{s_1}(x_0, y_0; f, \theta - \phi) \quad (2.28)$$

where

$$\begin{aligned} \hat{x}_\tau &= (x_0 - x'_\tau) \cos(\theta - \phi) + (y_0 - y'_\tau) \sin(\theta - \phi) \\ \hat{y}_\tau &= -(x_0 - x'_\tau) \sin(\theta - \phi) + (y_0 - y'_\tau) \cos(\theta - \phi). \end{aligned} \quad (2.29)$$

The rotation property in equation (2.28) shows that the response of the Gabor filter for a rotated image is equal to the response of the correspondingly rotated filter for the original image without rotation.

Let $s_5(x, y)$ be a differently illuminated version of the image $s(x, y)$. The uniform illumination change can be modeled as a multiplication by a constant, implying that $s_5(x, y) = cs(x, y)$. Based on the linearity of the convolution, it can be written [18]:

$$r_{s_5}(x, y; f, \theta) = cr_s(x, y; f, \theta). \quad (2.30)$$

The above equation shows the illumination invariance property, i.e. the Gabor filter response for an illuminated image (the original image multiplied by a constant) is equal to the correspondingly filter response (multiplied by the same constant) of the original image.

2.2.3 Discrete 2-D Gabor Filters

In all of the above equations Gabor functions are defined in the continuous domain. For a proper application of Gabor filters in image processing there is a need to obtain a proper discrete domain representation. In order to assure a proper transformation from the continuous domain to the discrete domain we need to apply certain restrictions.

Let $f(x, y)$ represent an image function and $\mathcal{F}(\xi_1, \xi_2)$ its Fourier transform. Then function $f(x, y)$ is called bandlimited if its Fourier transform is zero outside a bounded region in the frequency plane, that is [12]:

$$\mathcal{F}(\xi_1, \xi_2) = 0, \quad \text{for } |\xi_1| > \xi_{x0}, \quad |\xi_2| > \xi_{y0}, \quad (2.31)$$

where ξ_{x0} and ξ_{y0} are called the x and y bandwidths or sampling frequencies of the image. The band-limited image function $f(x, y)$ can be reconstructed from a discrete representation if the sampling frequencies are greater than twice the bandwidths [12]:

$$\xi_{xs} > 2\xi_{x0}, \quad \xi_{ys} > 2\xi_{y0}, \quad (2.32)$$

where the lower bounds on the sampling rates, $2\xi_{x0}$ and $2\xi_{y0}$, are called the Nyquist rates or Nyquist frequencies. According to the sampling theorem any bandlimited image $f(x, y)$, sampled uniformly on a rectangular grid with spacing Δx and Δy , can be recovered from the sample values if the sampling rate is greater than the Nyquist rate [12]. Some of the restrictions for discrete Gabor filters are the Nyquist frequency and the need for discrete Gabor filters to have a negligible response above the Nyquist frequency. Kamarainen et al. [15] give an analysis of discrete Gabor filtering. As they explain a proper construction of the filter in the 2-D frequency domain can be accomplished by satisfying the inequality:

$$\int \int_{-0.5}^{0.5} G(u, v) du dv \geq p_f \int \int G(u, v) du dv, \quad (2.33)$$

where p_f is the percentage of the filter envelope that must lie within the Nyquist limit [15]. The constraint in the above equation guarantees a proper sampling of the filter in the frequency domain, but we need to make sure we have a proper filter in the time domain also. The minimum spatial size n_{min} for a proper spatial domain

construction of the filter in equation (2.11) can be approximated from:

$$\int \int_{-\frac{n_{min}}{2}}^{\frac{n_{min}}{2}} |g(x, y)| dx dy \geq p_s \int \int |g(x, y)| dx dy \Rightarrow$$

$$\text{erf}\left(\frac{1}{2} \frac{|f| n_{min}}{\gamma}\right) \text{erf}\left(\frac{1}{2} \frac{|f| n_{min}}{\eta}\right) \geq p_s \quad (2.34)$$

where erf is the error function

$$\text{erf}(x) = \frac{2}{\sqrt{\pi}} \int_0^x e^{-t^2} dt \quad (2.35)$$

and p_s is the percentage of the filter envelope inside a rectangular filter of size $n_{min} \times n_{min}$ where $n_{min} \leq \min(M, N)$ for an image of size $M \times N$ [15]. Figure 2.4 shows that by varying the free parameters in equation (2.18) and by satisfying the above constraints, discrete Gabor filters of arbitrary orientation and bandwidth can be generated. The Gabor filter responses are invalid near the edges of an image where the distance to the edge is below $\frac{n_{min}}{2}$, due to the equivalence of Fourier domain multiplication and circular convolution [15]. A proper construction of discrete Gabor filters with the above constraints will give reliable results in feature extraction applications.

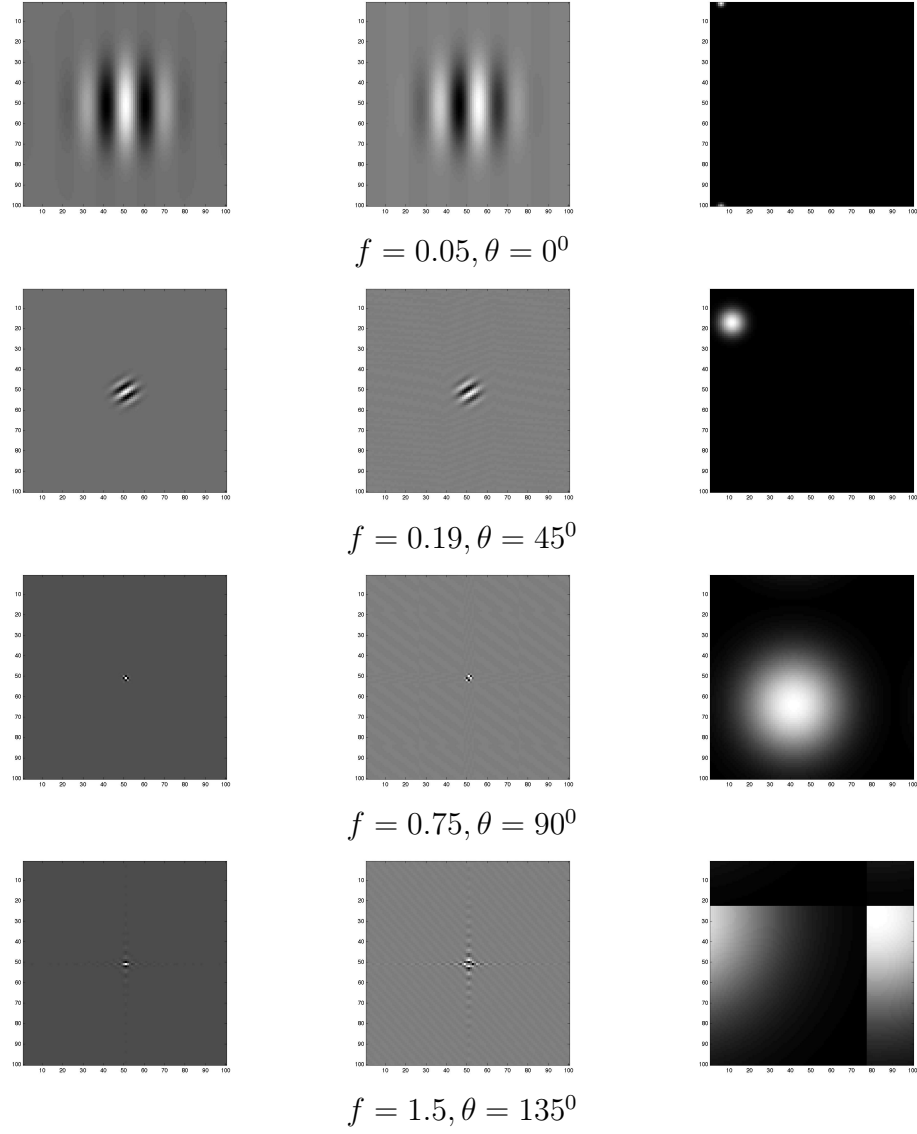


Figure 2.4: Gabor filters constructed with different parameters. From left to right: real part, imaginary part, and frequency space. The γ and η parameters are fixed, $\gamma = \eta = 1$.



Figure 2.5: Lenna image

2.3 Invariant Object Recognition

There have been many promising results in the applications of Gabor filters for object classification. Recently Gabor filters have been used in invariant object recognition, which is one of the most challenging problems in computer vision. Kyrki et al. [18] proposed object representation in a simple Gabor feature space, where illumination, rotation, scale, and translation invariant recognition of sufficiently distinguishable objects can be realized within a reasonable amount of computation. Gabor feature spaces have been successfully applied, for example in invariant face detection to extract facial features in demanding environments.

In the case when the objects are too complex to be distinguished in a simple Gabor feature space, it is possible to distinguish salient sub-parts, which then can be used as input to more general classification systems. This is the approach taken in this thesis for aerial image classification. In the following sections additional fundamental properties of Gabor filters for aerial feature representation, the construction of a Gabor feature space, and invariant search operations in this feature space are discussed in more detail. Figure 2.6 shows an example of how Gabor filters can be applied to the Lenna image in Figure 2.5, whose responses can be used afterwards for feature extraction or pattern recognition. As we can see from the figure, filters with different

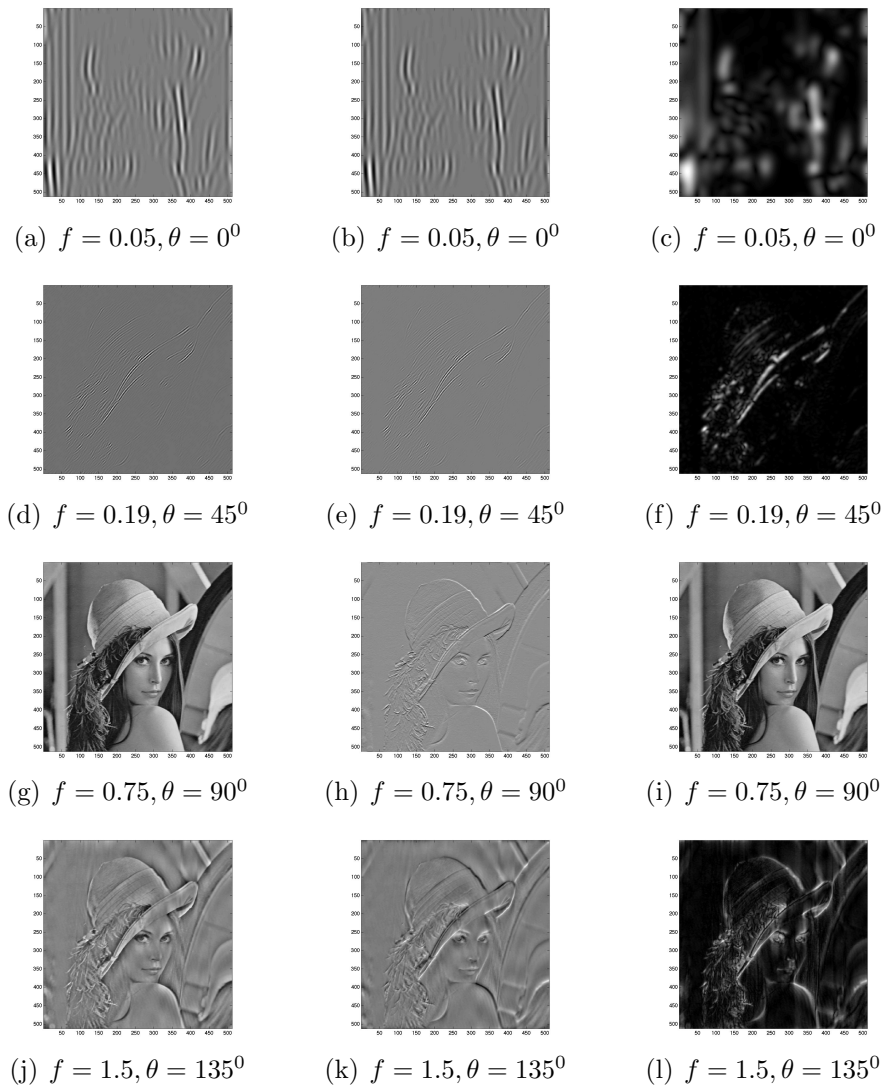


Figure 2.6: Gabor filter responses after doing convolution of the Lenna image with different Gabor filters. From left to right: real part, imaginary part, and magnitude of the response

frequencies and orientations detect different patterns on the image.

2.4 Simple Gabor Feature Space

Different Gabor filter parameters can be used to extract different features. The frequency and orientation of the Gabor filters are some of the most useful parameters. In order to distinguish between objects in most cases it is necessary to apply several Gabor filters to an image, by forming a so called "filter bank", and then for classification purposes we can look at the relationship between Gabor responses of different images. The selection of discrete rotation angles θ_k is very important, and it has been shown the orientations must be spaced uniformly as [18]:

$$\theta_k = \frac{k2\pi}{n} \quad \text{for } k = 0, \dots, n-1 \quad (2.36)$$

where θ_k is the k th orientation and n is the number of orientations to be used. However in this thesis the computation is reduced to half, as for real signals the responses on angles $[\pi, 2\pi)$ are 90° phase shifted from responses on $[0, \pi)$ [16]:

$$\theta_k = \frac{k\pi}{n} \quad \text{for } k = 0, \dots, n-1. \quad (2.37)$$

Equation (2.26) represents the scale invariance property of the Gabor filters but in real applications it is required to establish a spacing between the scales. In order to maintain homogeny spacing between the scales a logarithmic relation between the frequencies f has been established [18], [7]:

$$f_k = a^{-k} f_{max}, \quad \text{for } k = 0, \dots, m-1. \quad (2.38)$$

where f_k is the k th frequency, $f_0 = f_{max}$ is the highest frequency desired, and a is the frequency scaling factor. In this thesis a scaling factor of $\sqrt{2}$ is used which has been used for half-octave spacing, but another useful value for octave spacing is $a = 2$.

Gabor Feature matrix

A Gabor feature matrix is defined as follows:

$$G = \begin{bmatrix} r(x_0, y_0; f_0, \theta_0) & r(x_0, y_0; f_0, \theta_1) & \dots & r(x_0, y_0; f_0, \theta_{n-1}) \\ r(x_0, y_0; f_1, \theta_0) & r(x_0, y_0; f_1, \theta_1) & \dots & r(x_0, y_0; f_1, \theta_{n-1}) \\ \vdots & \vdots & \vdots & \vdots \\ r(x_0, y_0; f_{m-1}, \theta_0) & r(x_0, y_0; f_{m-1}, \theta_1) & \dots & r(x_0, y_0; f_{m-1}, \theta_{n-1}) \end{bmatrix}, \quad (2.39)$$

where each of $r(x_0, y_0; f_i, \theta_j)$ represent the Gabor filter responses (2.21) at location (x_0, y_0) , f_i and θ_j are the sampled parameters from equations (2.37) and (4.10), for $i = 0, \dots, m - 1$ and $j = 0, \dots, n - 1$ [18].

Chapter 3: Bayesian Networks and Linear Discriminants

In this chapter we present preliminary background of the mathematical theory behind Bayesian networks, mutual information and support vector machines. These methods are later used for aerial image classification.

3.1 Bayesian Networks

Bayesian networks are graphical structures used for representing the probabilistic relationships between a large number of variables and for doing probabilistic inference with those variables [23]. One situation where Bayes' theorem can be applied for performing probabilistic inference is when one feature of an entity has a direct influence on another feature of that entity. We can represent a large instance in a Bayesian network using little space by exploiting conditional independencies entailed by influence chains [23]. Probabilistic learning methods have been applied in various areas, such as E-mail classification [26], face recognition [27], feature selection in computer-aided diagnosis [31], fast binary feature selection [9], and handwritten digital recognition [1].

A Bayesian network is a directed acyclic graph, in which each node is assigned a random variable or an uncertain quantity, which in turn can take two or more values [24]. Let X_i be a random variable. In a Bayesian network (otherwise called a causal network), each X_i is denoted by a node [26]. An edge between two nodes represents direct influence or probabilistic influence. This dependency is involved between a parent node and a child node. For example, an edge from node X_1 to node X_2 , $X_1 \rightarrow X_2$, denotes that the first node has a direct influence on the second node.

Creating edges that indicate probabilistic influences is one way to construct a Bayesian network. A second way to construct such a network is by designing a structure of the Bayesian network which assumes that each random variable or node X_i is conditionally independent of its nondescendants given its parents [26]. In this case we are dealing with roots and we call them prior probabilities. Let X_1 and X_2 be events such that $P(X_2) \neq 0$, where $P(X_2)$ is the probability density function of X_2 . Then the conditional probability of X_1 given X_2 is defined as:

$$P(X_1|X_2) = \frac{P(X_1 \cap X_2)}{P(X_2)}. \quad (3.1)$$

Each node X_i in the Bayesian network is associated with a conditional probability table, specifying the probabilistic distribution over X_i given any possible values assigned to its parents [26]. Another way to construct a Bayesian network is by accomplishing probabilistic inference among pairs of features using the Bayesian network, i.e. computing the posterior distribution of variables given some observed evidence. Conditional probabilities of events can be computed from known probabilities using Bayes' theorem. Let again X and Y be two events, such that $P(X) \neq 0$ and $P(Y) \neq 0$. Then the following formula can be established:

$$P(X|Y) = \frac{P(Y|X)P(X)}{P(Y)}. \quad (3.2)$$

Suppose we have n events $X_1, X_2, \dots, X_n$. For any other event Y , the *law of total probability* says that:

$$P(Y) = \sum_{i=1}^n P(Y \cap X_i). \quad (3.3)$$

If $P(X_i) \neq 0$ for all i , then using (3.1) we can derive $P(Y \cap X_i) = P(Y|X_i)P(X_i)$.

Thus equality (3.3) can be written in the following form:

$$P(Y) = \sum_{i=1}^n P(Y|X_i)P(X_i) \quad (3.4)$$

Now if we have n mutually exclusive and exhaustive events X_i ($1 < i < n$), $X_i \cap X_j = \phi$ for $i \neq j$, such that $P(X_i) \neq 0$ for all i , equality (3.2) takes the following form:

$$P(X|Y) = \frac{P(Y|X_i)P(X_i)}{\sum_{i=1}^n P(Y|X_i)P(X_i)}. \quad (3.5)$$

Formulas (3.2) and (3.5) are called *Bayes Theorem* and they were both originally developed by Thomas Bayes. Equality (3.2) enables us to compute $P(X|Y)$ if we know $P(Y|X)$, $P(X)$, and $P(Y)$, while equality (3.5) enables us to compute $P(X_i|Y)$ if we know $P(Y|X_i)$ and $P(X_i)$ for all $i = 1, \dots, n$ [23].

3.2 Probabilistic Classification

Bayesian networks have been widely used for classification tasks. For example, Sahami et al. built a probabilistic classifier to detect junk E-mail. A Bayesian classifier (which is simply a Bayesian network applied to a classification task) contains a node C for the class variable and a node X_i for each of the features [26]. Let x be a specific instance, an assignment of values $x_1, x_2, \dots, x_n$ to the feature variables, and let C be an assignment of values $c_1, c_2, \dots, c_m$ to the class variable. Then the Bayesian network enables us to compute the probability $P(C = c_k|X = x)$ for each possible class c_k , using Bayes theorem [26]:

$$P(C = c_k|X = x) = \frac{P(X = x|C = c_k)P(C = c_k)}{P(X = x)} \quad (3.6)$$

The quantity $P(X = x|C = c_k)$ in the above equation is impractical to compute without independence assumptions [26]. A naive Bayes classifier embodies one of the

most restrictive forms of such independence assumptions. It makes the assumption that each feature X_i is conditionally independent of each other feature, given the class variable C , thus yielding [26]:

$$P(X = x|C = c_k) = \prod_i P(X = x_i|C = c_k). \quad (3.7)$$

Besides the Naive Bayes classifier there exist much more expressive and complex Bayesian classifiers. Figure 3.1 shows two Bayesian networks. The one on the left represents the Naive Bayes classifier with its restrictive independence assumptions, i.e. each of X_i are conditionally independent of each other given class C . The graph on the right shows a more expressive Bayesian classifier allowing for limited dependences between feature variables.

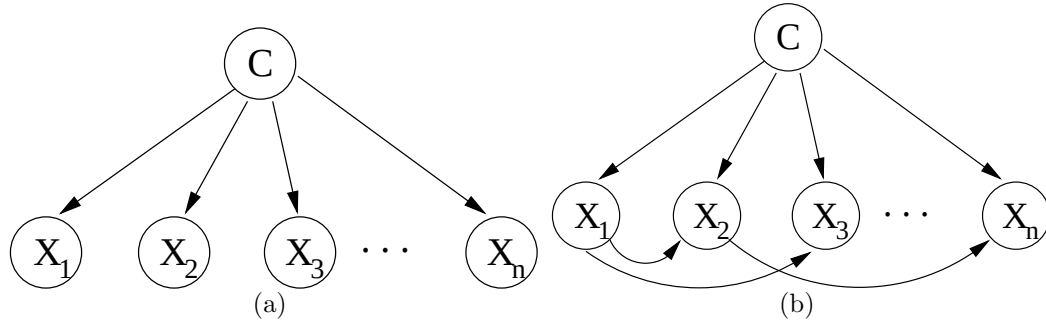


Figure 3.1: Bayesian networks. (a) Naive Bayes classifier; (b) A more complex Bayesian classifier [26]

Feature Selection

This thesis focuses on using the Naive Bayes classifier as one of the methods for aerial image classification. As we will see the feature space tends to be large, and we employ feature selection to reduce the number of redundant features. Mutual information $MI(X_i; C)$ between each feature X_i and the class C is computed by [26]:

$$MI(x \in X; c \in C) = \sum_{x_i, c} P(x_i, c) \log \frac{P(x_i, c)}{P(x_i)P(c)} \quad (3.8)$$

We select a smaller feature set (those containing the largest MI values) using the above equation.

3.3 Linear Discriminants

Because of their applicability to a large number of applications Support Vector Machines (SVMs) have been studied and used widely by the research community. SVMs are a class of algorithms that use the idea of kernel methods, which appear to be well-suited for data mining tasks [3]. In this section we describe the basic concepts behind SVMs from a geometric perspective. Three key ideas which are crucial for understanding the power of SVMs are: *margins*, *duality*, and *kernels*.

Let D represent a set of n datapoints, such that each datapoint is represented in a d dimensional input space as follows:

$$D = \{(x_i, y_i) | x_i \in R^d, y_i \in \{-1, 1\}\}_{i=1}^n, \quad (3.9)$$

where $y_i = \pm 1$ indicating the class to which each point x_i belongs to. We say two disjoint classes $C \subseteq \{x_i\}$ and $D \subseteq \{x_i\}$ are linearly separable if there is a hyperplane that separates C from D . Assuming C and D are linearly separable the problem of interest is to find the hyperplane which is furthest from both classes C and D in some well defined sense. Any hyperplane can be defined as:

$$w \cdot x - b = 0, \quad (3.10)$$

where $\cdot$ denotes the dot product, the vector w determines the orientation of a discriminant hyperplane between two sets of points x_i , and the scalar $\frac{b}{\|w\|}$ determines the offset of the hyperplane from the origin. As it can be seen from Figure 3.2, we can think of many planes that will separate the two sets. The hyperplanes that maximize the margin between two sets can be described as follows:

$$w \cdot x - b = 1 \quad \text{and} \quad w \cdot x - b = -1. \quad (3.11)$$

Using geometry, the distance between the above mentioned hyperplanes can be defined as $\frac{2}{||w||}$. Thus in order to maximize the margin between the hyperplanes we need to minimize $||w||$.

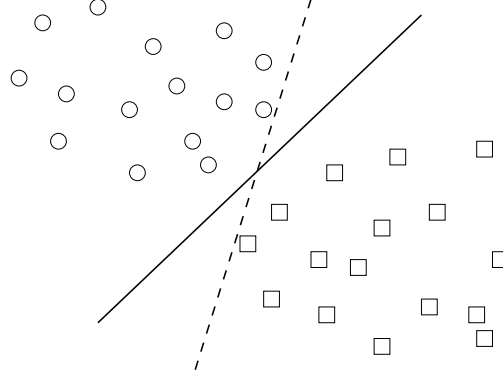


Figure 3.2: Linear discriminant planes

The small number of data points which clash with the support planes in both sides are called *support vectors* (shown in Figure 3.3) [3]. Thus to minimize the number of the support vectors, the following constraint needs to be added for each i :

$$\begin{aligned} w \cdot x_i - b &\geq 1 \text{ for } x_i \in \text{Class1 or} \\ w \cdot x_i - b &\leq -1 \text{ for } x_i \in \text{Class-1} \end{aligned} \quad (3.12)$$

The above constraints can be simplified with:

$$y_i(w \cdot x_i - b) \geq 1, \text{ for all } 1 \leq i \leq n. \quad (3.13)$$

Thus the optimization problem becomes:

$$\min_{w,b} ||w||, \quad (3.14)$$

subject to (3.13). The above optimization problem is difficult to solve because it depends on $||w||$, which involves a square root. To simplify the optimization problem

we can substitute $\|w\|$ with $\frac{1}{2}\|w\|^2$ in equation (3.14). Now the optimization problem can be solved using the following quadratic program:

$$\min_{w,b} \frac{1}{2} \|w\|^2, \quad (3.15)$$

subject to (3.13). The dual of the supporting plane from equation (3.15) yields the

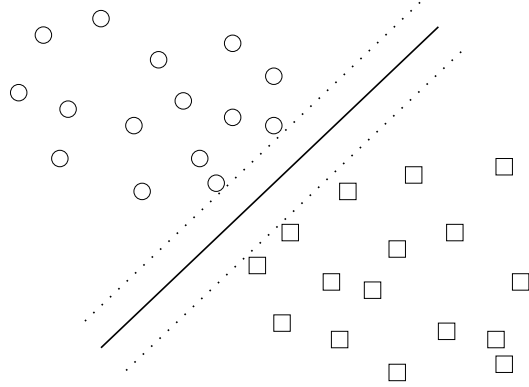


Figure 3.3: Finding the best plane by maximizing the margin.

following optimization problem:

$$\max_{\alpha_i} \sum_{i=1}^n \alpha_i - \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n \alpha_i \alpha_j y_i y_j x_i^T x_j, \quad (3.16)$$

such that $\sum_{i=1}^n y_i \alpha_i = 0$, $\alpha_i \geq 0$, and $i = 1, \dots, n$. Equations (3.15) and (3.16) yield the same normal to the plane $w = \sum_{i=1}^n y_i \alpha_i x_i$ and the threshold b determined by the support vectors (for $\alpha_i \geq 0$) [3].

3.4 Support Vector Machines (SVM)

A method proposed for on-road vehicle detection involves using support vector machines for classification purposes [30]. Given a set of points SVMs are used to find the hyperplane that leaves the largest set of points on one side, and at the same

time maximizes the distance between both sets from the hyperplane. The approach assumes l examples from two classes:

$$(x_1, y_1)(x_2, y_2) \dots (x_l, y_l) \quad (3.17)$$

where $x_i \in \mathbb{R}^N$ and $y_i \in \{-1, +1\}$ [30]. The authors solve a constrained optimization problem using quadratic programming to find the optimal hyper-plane. The width of the margin between the classes is used as an optimization criterion and the hyperplane is defined as:

$$f(x) = \sum_{i=1}^l y_i a_i k(x, x_i) + b \quad (3.18)$$

where $k(x, x_i)$ is a kernel function and the membership of x is determined by the sign of $f(x)$ [30]. A Gaussian kernel is used, as it outperforms other kernels in experiments on vehicle detection. Constructing the optimal hyperplane in equation (3.18) is equivalent to finding all the nonzero a_i and the values corresponding to such a_i form a support vector of the optimal hyperplane [30].

3.4.1 SVM^{light} and Kernels

The implementation of SVM learners for large scale problems is non-trivial, having high memory and computation time requirements. Regularization to compensate for noise in the data is also often needed. Recently a new algorithm, SVM^{light} for training SVMs on large scale problems was presented in the literature [13]. This algorithm involves the numerical solution of a quadratic optimization problem with bound constraints and one linear equality constraint. The implementation of SVMs in SVM^{light} makes large-scale SVM training more efficient.

For some classification problems the simple linear discriminant functions do not work well or sometimes do not work at all. For instance, for the classification problem shown in Figure 3.4 a quadratic function is needed. Bennett and Campbell [3]

use a classic method for converting a linear classification algorithm into a nonlinear classification algorithm by adding additional attributes to the data which represent nonlinear functions of the original data. If we have a two dimensional vector space with attributes $[r, s]$ then we can construct a quadratic discriminant function by simply mapping the original two dimensional input space $[r, s]$ into a five dimensional feature space $[r, s, rs, r^2, s^2]$, and constructing a linear discriminant in that space [3].

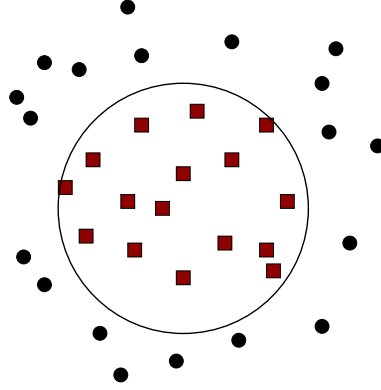


Figure 3.4: Example requiring a quadratic discriminant [3]

The mapping function, $\theta(x) : R^2 \Rightarrow R^5$, is defined as:

$$\theta([r, s]) = [r, s, rs, r^2, s^2]. \quad (3.19)$$

Using w as a weight vector,

$$w \cdot \theta(x) = w_1 r + w_2 s + w_3 rs + w_4 r^2 + w_5 s^2,$$

the classification function from equation (3.18) can be rewritten as:

$$f(x) = \text{sign}(w \cdot \theta(x) - b) = \text{sign}(w_1 + rw_2s + w_3rs + w_4r^2 + w_5s^2 - b) \quad (3.20)$$

The above function is linear in the mapped five dimensional feature space but quadratic in the two dimensional feature space. According to Bennett and Campbell there are

two problems associated with the above nonlinear mapping method and there exist two ways to get around these problems. One of the problems stems from the fact that the dimensionality of the feature space explodes exponentially. A penalty term is added to the minimization function to solve this problem [3].

The second problem is that computing $\theta(x)$ is not practical. The solution to this problem are *kernels*. Bennett and Campbell use Hilbert-Schmidt kernels to simplify this mapping to take advantage that for certain mappings θ and two points u and v , the inner product of the mapped points can be evaluated using the kernel function without ever explicitly knowing the mapping, i.e. $K(u, v) = \theta(u) \cdot \theta(v)$. Other popular kernel functions used in applications of SVMs are:

- the degree d polynomial kernels defined as:

$$K(u, v) = (u \cdot v + 1)^d,$$

- the Radial Basis Function Machine defined as:

$$K(u, v) = \exp\left(-\frac{\|u-v\|^2}{2\sigma}\right), \text{ and}$$

- the two-layer neural network defined as:

$$K(u, v) = \text{sigmoid}(\eta(u \cdot v) + c).$$

Chapter 4: Classification Methodology

We present a new method for image classification that combines Gabor feature space representation and several similarity measures. We consider a similarity measure proposed by Manjunath and Ma [20] as well as a simple Euclidean distance metric. This new method is then applied to the aerial image classification problem introduced in Chapter 1. Our key goal is to distinguish between tree and non-tree areas in aerial images, though several related classification problems can also be posed, e.g. detection of landslides, tree types, rivers, roads, etc. (See Chapter 1 and Chapter 5 for more details). We also propose a new classification approach where histograms of similarity (distance) values and the receiver operating characteristic (ROC) curve are employed for Gabor filter parameter selection.

4.1 Distance-based Classification Method

Let F denote a collection of Gabor filters, or Gabor filter bank,

$$F = [g_{k\ell}], \quad k = 1, \dots, m_1, \quad \ell = 1, \dots, m_2. \quad (4.1)$$

A filter bank F defines a transformation of an image region into a corresponding representation in Gabor feature space, $F : R^{n \times n} \rightarrow C^{nm_1 \times nm_2}$. Thus,

$$F(s(i, j)) = [r_s(i, j, f_k, \theta_\ell)], \quad k = 1, \dots, m_1, \quad \ell = 1, \dots, m_2. \quad (4.2)$$

The matrix $F(s(i, j))$ is also called a Gabor feature matrix and expressed as:

$$F(s(i, j)) = \begin{bmatrix} r_s(i, j; f_1, \theta_1) & r_s(i, j; f_1, \theta_2) & \cdots & r_s(i, j; f_1, \theta_{m_2}) \\ r_s(i, j; f_2, \theta_1) & r_s(i, j; f_2, \theta_2) & \cdots & r_s(i, j; f_2, \theta_{m_2}) \\ \vdots & \vdots & \ddots & \vdots \\ r_s(i, j; f_{m_1}, \theta_1) & r_s(i, j; f_{m_1}, \theta_2) & \cdots & r_s(i, j; f_{m_1}, \theta_{m_2}) \end{bmatrix}. \quad (4.3)$$

Let T_s denote a set of training image regions, consisting of features in two class types tree and non-tree, e.g.

$$T_s = \{(s, c) \mid s \text{ is an } n \times n \text{ image region and } c = 1 \text{ (tree) or } c = -1 \text{ (nontree)}\}. \quad (4.4)$$

Let T_F denote the set of image regions represented in Gabor feature space:

$$T_F = \left\{ (F(s), c) \mid F(s) \text{ is an } nm_1 \times nm_2 \text{ Gabor filter response and } \right. \\ \left. c = 1 \text{ (tree) or } c = -1 \text{ (nontree)} \right\}. \quad (4.5)$$

Let s_r be a reference $n \times n$ image region of a know feature and $S_r = F(s_r)$ its representation in Gabor feature space. We wish to determine the transformation F defining the Gabor feature space where two known classes are best separated. A measure of similarity between elements of T_F and S_r is needed.

4.1.1 Similarity Measures

Similarity measures are very important for comparison and classification purposes, and many have been proposed in the literature. We consider two similarity measures for the selection of appropriate Gabor feature spaces. Manjunath and Ma proposed a novel Gabor wavelet-based texture analysis scheme [20]. This measure assumes that the local texture regions are spatially homogenous, and the mean μ and the standard deviation σ of the magnitude of the Gabor filter response coefficients are used to represent the region for classification and retrieval purposes.

Mean-Variance Distance Measure

The mean μ and the standard deviation σ of the response of an image region to a Gabor filter is used to construct a distance measure. For an image region $s(x, y)$, the mean and the standard deviations of the Gabor filter response of s , $r_{k\ell}$, with respect

to a particular Gabor filter g_{kl} , is given by:

$$\mu_{k\ell} = \iint |r_{k\ell}(x, y; f_k, \theta_\ell)| dx dy \quad \text{and} \quad \sigma_{k\ell} = \sqrt{\iint (|r_{k\ell}(x, y; f_k, \theta_\ell)| - \mu_{j_k})^2 dx dy}. \quad (4.6)$$

Let s_i and s_j represent different image regions. Then the total similarity measure between s_i and s_j in Gabor feature space is given by:

$$d(S_i, S_j) = \sum_{k=1}^{m_1} \sum_{\ell=1}^{m_2} \bar{d}(r_{s_i}(x, y; f_k, \theta_\ell), r_{s_j}(x, y; f_k, \theta_\ell)), \quad (4.7)$$

where $\bar{d}$ is given by the Manjunath and Ma similarity measure:

$$\bar{d} \equiv d_{MM}(r_{s_i}(x, y; f_k, \theta_\ell), r_{s_j}(x, y; f_k, \theta_\ell)) = \left| \frac{\mu_{k\ell}^i - \mu_{k\ell}^j}{\alpha_\mu} \right| + \left| \frac{\sigma_{k\ell}^i - \sigma_{k\ell}^j}{\alpha_\sigma} \right|. \quad (4.8)$$

The values α_μ and α_σ are used to normalize the individual feature components. They represent the standard deviations of the respective features when combining the vectors of means and standard deviations of the reference image and the target image for classification.

Euclidean Distance Measure

We use the Euclidean distance as another similarity measure to define (4.7). In particular:

$$\bar{d} \equiv d_E(r_{s_i}(x, y; f_k, \theta_\ell), r_{s_j}(x, y; f_k, \theta_\ell)) = \|r_{s_i}(x, y; f_k, \theta_\ell) - r_{s_j}(x, y; f_k, \theta_\ell)\|_2. \quad (4.9)$$

Illumination Invariance

In most cases of interest illumination can change per observation and hence images of same targets can differ significantly in magnitude. Illumination invariance is desirable

for proper processing of aerial imagery and can be achieved by normalizing the Gabor feature matrix given in (4.3):

$$\bar{F}(s(i, j)) = \frac{F(s(i, j))}{\sqrt{\sum_{k, \ell} |r_s(i, j; f_k, \theta_\ell)|^2}}. \quad (4.10)$$

4.1.2 Selection of Gabor Feature Space

To select a Gabor feature space where two feature classes of interest are best separated, we use the total similarity measure (4.7) to compute

$$d(S, S_r), \quad \text{for all } S \text{ in } T_F. \quad (4.11)$$

Let S_r be a reference image associated with $c = -1$ (nontree). One seeks to maximize (4.11) for $(S, 1) \in T_F$, and minimize (4.11) for $(S, -1) \in T_F$. This is a difficult min-max optimization problem to solve numerically. In this thesis we propose a Monte Carlo type optimization approach as follows:

1. Create a set of candidate frequency and orientation parameters defining a Gabor filter space,
2. Compute (4.11) for all $S \in T_F$ and fixed S_r ,
3. Compute the histogram H of all the distance values obtained,
4. Choose the frequency and orientation parameters pertaining to the histogram which has the minimum area of overlap.

We do this twice, once with Manjunanth and Ma's measure (4.8) and the other with the Euclidean distance (4.9). A good Gabor feature space shows separation of two clusters within H . A separation threshold must now be chosen between these two often overlapping clusters.

4.1.3 Selection of Separation Threshold

There are several ways in which one can choose a separation threshold value between two overlapping clusters. Here we use a method based on the receiver operating characteristic (ROC) curve. More specifically, we propose the following simple but accurate procedure:

1. Find the overlapping interval $[\tau_1, \tau_2]$ of the two clusters in the histogram H ,
2. For a finite number of $\tau \in [\tau_1, \tau_2]$ and all $S \in T_F$ classify s as belonging to class $c = 1$ (tree) if $D(S, S_r) > \tau$ and to class $c = -1$ (non-tree) if $D(S, S_r) \leq \tau$.
3. Construct the ROC curve by computing the false positive rate (FPR) and true positive rate (TPR) for each τ . TPR and FPR are defined as:

$$\text{TPR}(\tau) = \frac{\text{Number of } s \in T_s \text{ correctly classified as "positive"}}{\text{True positives} + \text{False Negatives}} \quad (4.12)$$

$$\text{FPR}(\tau) = \frac{\text{Number of } s \in T_s \text{ incorrectly classified as "positive"}}{\text{True Negatives} + \text{False positives}}. \quad (4.13)$$

4. Choose τ for which the point (FPR,TPR) is closest to the upper left corner (0,1). The point (0,1) represent perfect accuracy, meaning there is a 0% false positive rate and 100% true positive rate.

Classification of New Data

The threshold and Gabor feature space obtained in the preceding procedure can now be used to classify new image regions as belonging to one of two classes, e.g. tree or nontree. In other words, given a new image $\hat{s}$, we compute $d(\hat{S}, S_r)$ and use τ to determine what feature class it belongs.

Remarks

In practice one cannot easily decide the optimal number of frequency and orientation parameters that should be used in Gabor feature extraction. The number of frequency and orientation parameters is dependent on the application. One important constraint to keep in mind when choosing this number is the computation time increase. As we described above, the size of the Gabor feature matrix increases with the increase of the number of frequency and orientation parameters used. One other constraint in this process of classification is the choice of an optimal classification threshold.

Complexity Analysis

Let images be of size $n \times n$ and sub-images be of size $k \times k$. Then, for a sub-image s the cost of computing S is $\mathcal{O}(m_1, m_2, k^2 \log k)$, where m_1 and m_2 are the number of frequencies and orientation parameters, respectively. If we assume that T_F is constructed for a single image of size $n \times n$ then the cost of computing T_F becomes $\mathcal{O}(\text{cost of } S \times |T_F|) = \mathcal{O}(m_1, m_2 k^2 \frac{n^2}{k^2} \log k) = \mathcal{O}(n^2 m_1 m_2 \log k)$. The cost of computing the similarity measure using the E and MM distances is $\mathcal{O}(m_1 m_2 n^2)$. Note that MM has higher constant. The cost of selecting a separation threshold is the same.

4.2 Probabilistic Classification Method

The classification approach previously described is a simplistic scheme that is computationally efficient but generally not very accurate. Its accuracy depends heavily on the features of the reference image selected. In this section we build upon the previous scheme by considering a probabilistic approach which is computationally more expensive but leads to more accurate results.

We develop a system that uses a naive Bayes classifier to determine whether an image represents a tree or non-tree area, replacing the need for a distance metric such as the total similarity measure of equation (4.11). For a given image s , the probability we are interested in is:

$$P(s \in C | Feature1 = A_1, Feature2 = A_2, \dots, FeatureN = A_n), \quad (4.14)$$

where $Feature1$ through $FeatureN$ are Gabor filter response values, and C denotes class 'tree' or 'non-tree'.

Since for a naive Bayes classifier all features are conditionally independent of each other given C , using equation (3.7) the above probability can be rewritten as the product over all features,

$$P(Feature = f | s \in C) = \prod_i P(Feature = A_i | s \in C). \quad (4.15)$$

Our probabilistic classification method consists of a training part and a classifying part. We use mutual information (3.8) for the feature selection part. This allows us to reduce the number of redundant Gabor features.

Training Part

The training part employs T_F , where the magnitude of the Gabor filter response is used as input. The training procedure involves the following steps:

1. Calculate the conditional probabilities of Gabor features in S given that $s \in c$ for $c = 1$ (tree) and $c = -1$ (non-tree).
2. Calculate the prior probabilities of 'tree' ($c=1$) and 'non-tree' ($c=-1$) examples being true,

$$P(c = \pm 1) = \frac{\text{Number of}(s, c = 1) \text{ or } (s, c = -1)}{\text{Total number of}(s, c)}, \quad (4.16)$$

3. Compute the mutual information of each feature using using (3.8) and the two classes to which it can belong, $MI(Feature = A_i, c = \pm 1)$. A threshold must be used to remove the features with a low mutual information value.

The above training procedure outputs two vectors with the probabilities of specific features belonging to each class 'tree' and 'non-tree',

$$\begin{aligned} T_V &= [P(Feature = A_i | c = 1)], \\ NT_V &= [P(Feature = A_i | c = -1)], \end{aligned} \quad (4.17)$$

and a vector with a list of classifiers obtained using mutual information:

$$MI_V = [Feature = A_i] \quad \text{for which} \quad MI(Feature = A_i, c = \pm 1) > \tau, \quad (4.18)$$

where τ is the mutual information threshold.

Classification of new data sets

The classifying procedure takes as input T_F , vectors containing the tree and non-tree feature probabilities from (4.17), and the mutual information vector containing the classifiers (4.18). The following procedure is followed for classification of an image s :

1. If $(Feature = A_i) \in S$ and $(Feature = A_i) \in MI_V$ then create two vectors containing the 'tree' and 'non-tree' probabilities of $(Feature = A_i)$ as in (4.17),
2. Compute the total probability of the image belonging to class 'tree' or 'non-tree':

$$\begin{aligned} P_T(s \in c = 1) &= \prod_i P(Feature = A_i | c = 1), \\ P_{NT}(s \in c = -1) &= \prod_i P(Feature = A_i | c = -1), \end{aligned} \quad (4.19)$$

3. If $P_T > P_{NT}$ then $s \in (c = 1)$, else $s \in (c = -1)$.

Remarks

We represent and compute the conditional probabilities for each feature (4.15) in *log* format, to avoid underflow and also to speed up computation. Since the feature probabilities are less than 1, the more probabilities we multiply together, the smaller the product becomes. We avoid this by using *log* probabilities instead of *raw* probabilities, so we get numbers that are not as small. The *log* probabilities are combined by adding them, since adding in *log* space is equivalent to multiplying in linear space. The key benefit of this method is that a reference image is not needed for classification of new data sets.

Complexity Analysis

If we have t training examples and f features, then the time required to learn a naive Bayesian classifier is $\mathcal{O}(tf)$ [8]. In our specific application this would be equivalent to $\mathcal{O}(m_1 m_2 k^2 |T_F|) = \mathcal{O}(m_1 m_2 n^2)$.

4.3 SVM-based Classification Method

We propose a third classification method which uses support vector machines. This method builds upon the two previous methods. Our SVM classification method use a linear kernel and consists of training and the classification parts.

Training Part

The training procedure involves the following steps:

1. Obtain the vectors (for the training examples) with probabilities of specific features belonging to each class 'tree' and 'non-tree' using (4.17) the mutual

information vector (3.8) using the naive Bayes classifier and mutual information approach.

2. Select the Gabor feature values of the training image which are also present in the mutual information vector. Compute a feature vector SVM_V containing the tree or non-tree probability values as follows:

$$\text{If } P_T > P_{NT} \text{ then } SVM_V(i) = P_T, \text{ else } SVM_V(i) = P_{NT}, \quad (4.20)$$

where i is the index of the SVM_V vector.

3. For a tuple (s, c) , output the feature vector SVM_V in the following format:
 $\pm 1 : \textit{Feature number} : \textit{Feature value}$, where *feature number* is the number of all features from 1 - $|SVM_V|$, and *feature value* is the value of the corresponding feature in SVM_V ,
4. Use the output from the previous step as input to the SVM learner. The SVM learner will output a model file containing necessary information (support vectors) for the classification part.

Classification of new data sets

For preparing the right input format for the SVM classifier we use the tree/non-tree probabilities, and the mutual information vector as explained in the above steps. The only difference here is that since the class to which the input from the new dataset pertains is not known then we output 0 as class value (so the correct format for the SVM classifier would be: 0 *feature number* : *feature value* *feature number* : *feature value* ...). The SVM classifier takes as input the file containing the feature values for the new image being tested and the model file created in the training part. The

classification for an image (s, c) is done as follows:

$$\begin{aligned} \text{If } \text{Output}(SVM_{classifier}) > 0 \quad \text{then } s \in (c = 1), \\ \text{If } \text{Output}(SVM_{classifier}) < 0 \quad \text{then } s \in (c = -1). \end{aligned}$$

Note that an implementation developed by Joachims (SVM^{light}) has been used for applying some of the above steps.

Complexity Analysis

The time complexity for linear-SVM training is $\mathcal{O}(ft)$, where f is the maximum number of non-zero features, and t is the number of training data [14]. Thus the equivalent time complexity for our SVM application is $\mathcal{O}(mt)$, where $m \leq m_1 m_2 n^2$. For classification using a linear SVM the computation time can be reduced to $\mathcal{O}(f)$ in doing the dot product between the weight vector $\|w\|$ and the vector of features, where f is the number of non-zero features in the image being classified.

Chapter 5: Experimental Results

We present experimental results that demonstrate the effectiveness of the models presented in sections 4.1, 4.2 and 4.3, when applied in aerial image classification. This section explains the results obtained and the detailed analysis of various experiments. All imaging data pertains to the mountainous areas of Peru (one such image is shown in Figure 5.1). This data was gathered from the U.S. Air Force in 1960s.

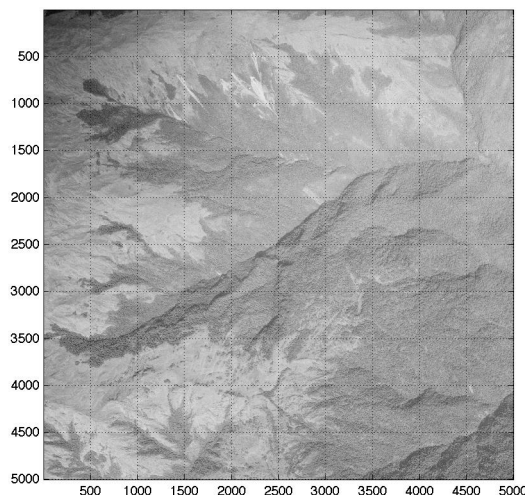


Figure 5.1: One of the images used for experiments

The original images were of size $12,000 \times 12,000$. For the experiments we cropped the original images to size 5000×5000 and then sub-images of size 500×500 were used. The intent is to classify between the tree and non-tree areas of the images. The fact that these images are taken from airplanes in a high distance from the ground, makes the classification problem difficult as there is a lack of clearly distinguishable features which can be used for classification and the illuminance changes across the image.

In each experiment, the overall data pool of images and the specific selection

of Gabor filter parameters are described. Relevant figures such as the ROC curve, histograms, and tables with classification accuracy values are presented for effectively compare the results using different methods. The visual comparison of the histograms should help adequately in choosing the best Gabor filter parameters to be used for classification of new data sets.

Although most of the experiments aim to test the robustness of the classification models, specific goals guide the composition of the various experiments. A key goal is to show the ability of the new models to classify with high accuracy. Another primary goal involves showing that Gabor filters' performance used in combination with proper classification methods is robust even in classification problems where the number of distinguishable features in an image is limited. A third goal is to demonstrate the ability of our methods to classify accurately for a large set of images under different illuminance conditions.

5.1 Observation Experiment

The first experiment shows the difficulty associated with selecting the proper Gabor feature space which maximizes the total similarity measure, (4.11), between $(S_r, 1)$ and $(S, -1)$ and minimizes the total similarity measure between $(S_r, 1)$ and $(S, 1)$.

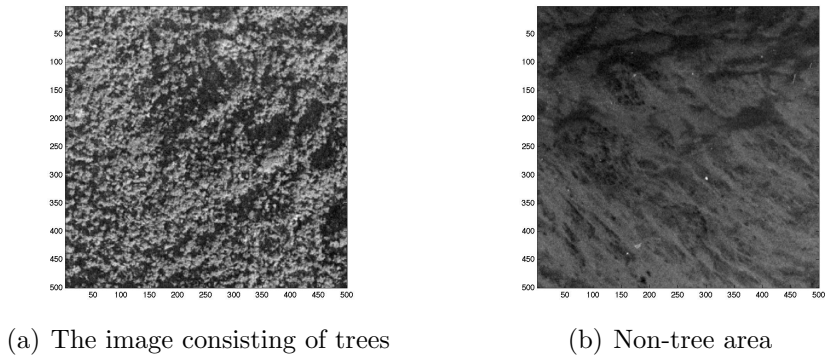


Figure 5.2: The images used for the observation experiment

As we explain in this section this min-max optimization problem is very difficult to solve when illumination invariance is not applied to the Gabor feature matrix in (4.3). In this experiment, we choose two sub-images from Figure 5.1. One represents a tree area and the other a non-tree area (shown in Figure 5.2). The 500×500 images consisting of the non-tree and tree areas are segmented into 10×10 patches. By varying the Gabor filter parameters a Gabor filter bank consisting of different orientations and frequencies is generated. The Gabor filter responses of each of the 10×10 sub-images are computed using (4.3) for different frequency and orientation parameters. We fix the η and γ parameters to be 1, so our Gabor filters are of

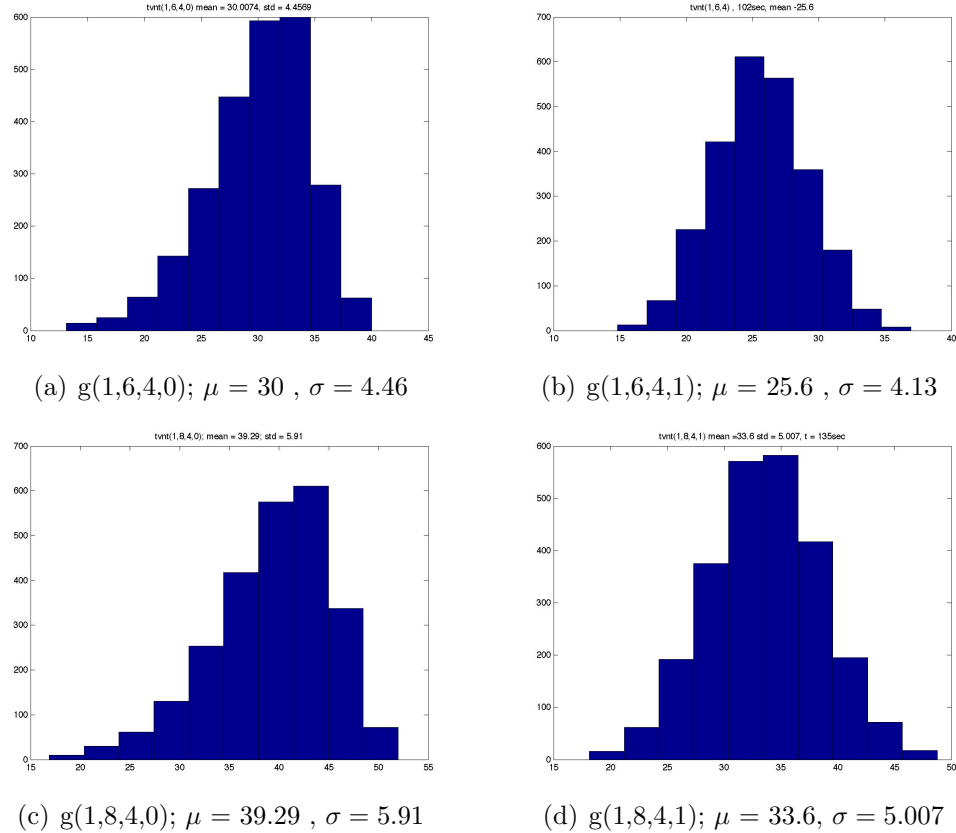


Figure 5.3: Histograms of distance measures between Gabor filter responses of images belonging to two different classes. The figures compare the differences between the magnitudes of Gabor responses (using the MM distance) for tree vs non-tree images. Parameters for Gabor filters are (maximum frequency, number of frequencies, number of orientations, 0 - no lightness normalization and 1 - with lightness normalization).

circular shape. A 10×10 image of the upper left side of image a) from Figure 5.2 is compared to all the 10×10 sub-images of image b) using MM distance. Recall that we are comparing a tree area with non-tree areas, therefore a large difference between them is desired. Figure 5.3 shows the histograms of the distance measures computed using (4.7) and (4.8). As we can see from the graphs the mean of the differences is higher when we do not apply lightness normalization. We repeated the

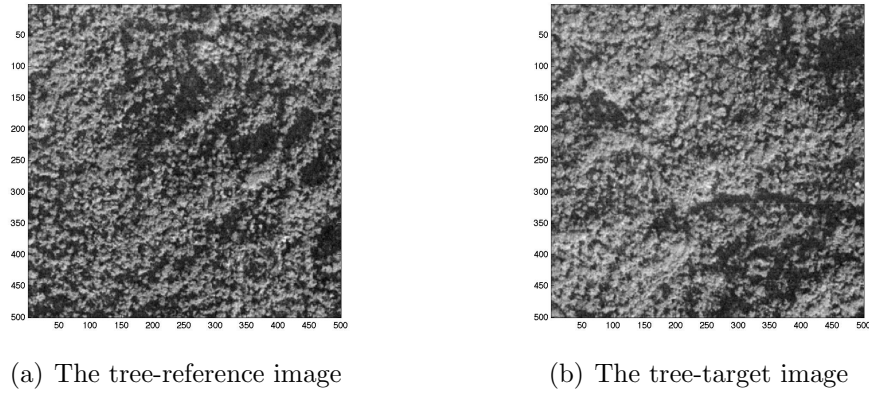


Figure 5.4: The images used for tree vs tree classification

experiment for other frequency parameters and found that if we use more frequency parameters the mean of the distance values increases, which in this case is good since we need to have a higher difference between trees and non-tree areas. The drawback is that the computational time increases with the increase in the number of frequencies (for 8, 10, and 11 frequencies it takes 130 sec, 166 sec, and 192 sec respectively). Another problem associated with classification of these type of images is deciding a threshold for classification. Figure 5.5 shows the results obtained when comparing tree vs tree areas with Gabor filters of different parameters. The goal is to minimize (4.11) between $(S_r, 1)$ and $(S, 1)$, where S_r is the reference image in Gabor feature space, and S is the target image in Gabor feature space.

We performed experiments with a greater number of sub-images from Figure 5.1. To reduce the computation time we used Gabor filter banks with 2 frequencies and 4

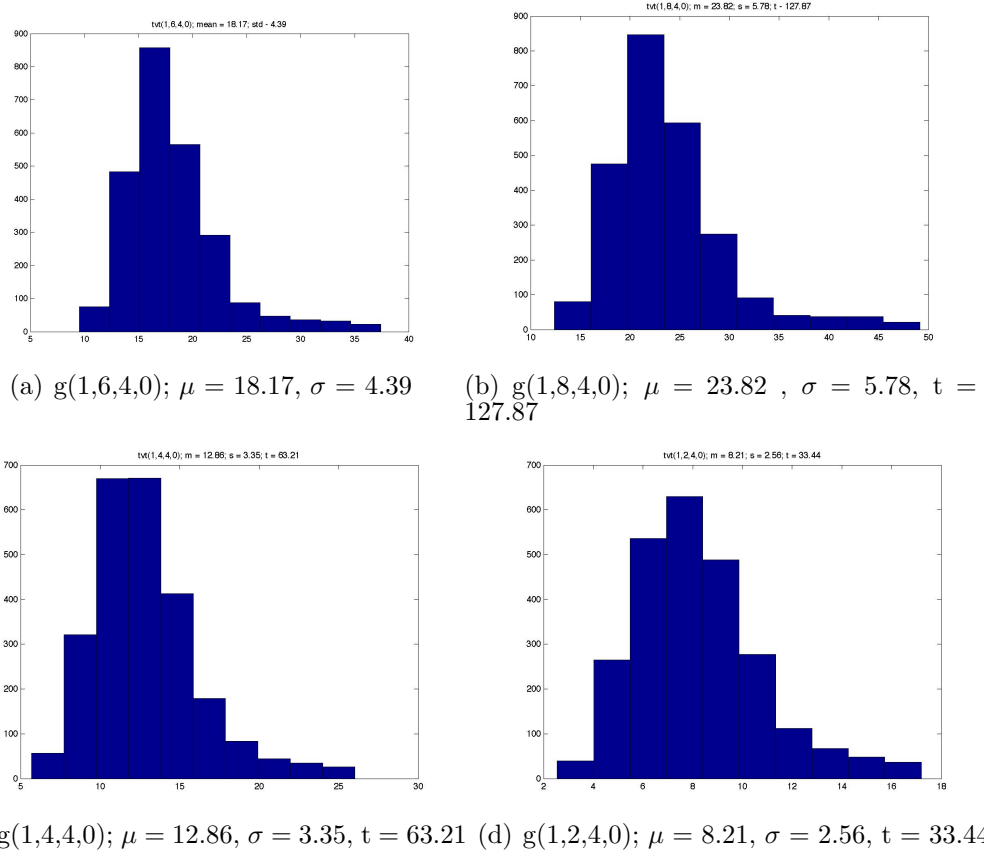


Figure 5.5: Using Gabor Filters to compare tree vs tree areas for images taken from Figure 5.4 (μ - mean , σ - standard deviation, t - computational time in seconds)

orientations (the maximum frequency was 1). A tree sub image was compared versus 25,000 non-tree subimages (each of size 10 x 10) and the mean for total similarity measures between Gabor responses came out to be 13 with a standard deviation of 2. We did the same for 25,000 tree subimages and the mean of the differences for tree vs tree was 10 with a standard deviation of 2. This implies a big overlap between the means of Gabor responses making it hard to decide a threshold for classification using the total similarity measure (4.7) with no illumination normalization.

5.2 Manjunath and Ma vs. Euclidean distance

This experiment demonstrates the performance of the similarity measure methods defined in section 4.1.1, applied with illumination normalization defined in (4.10). We compare the Manjunath and Ma (MM), and Euclidean (E) distances in classifying images. Several banks of Gabor filters constructed using (4.1) with different frequency and orientation parameters are used for feature extraction. The selected images used

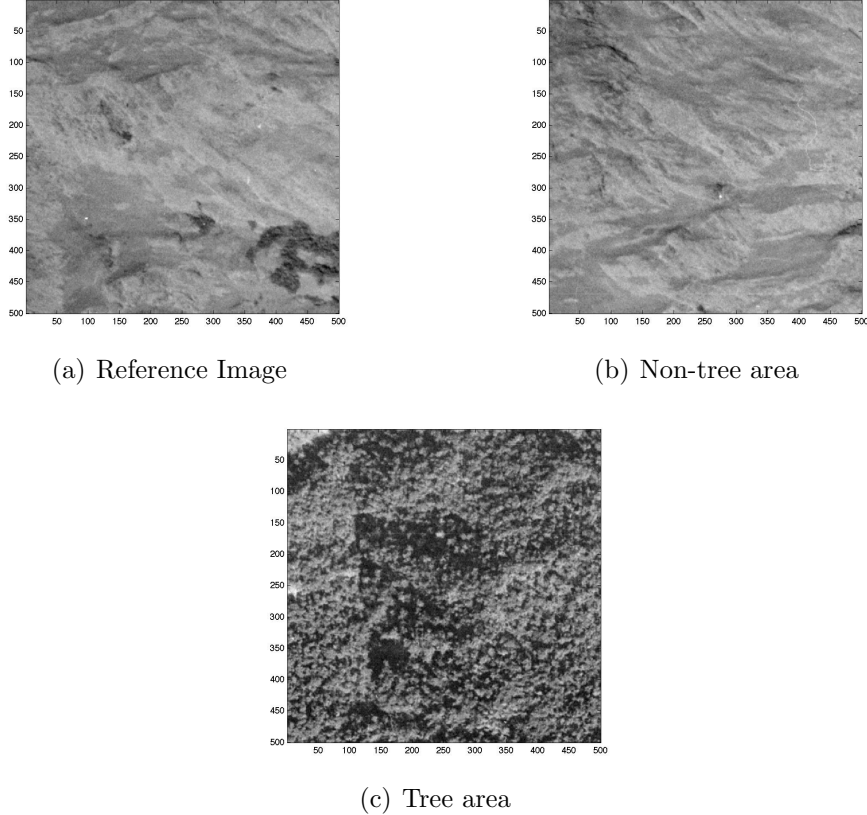


Figure 5.6: Images used in the training part.

for this experiment are shown in Figure 5.6. The top 10×10 top sub-image of (a), representing a non-tree area, was compared against (b) and (c). The histograms (Figure 5.7) show the similarity measure values for tree vs. tree and tree vs. non-tree areas for all the Gabor filter banks. The number of the 10×10 images used for the

comparison is 2500 for each of the tree and non-tree areas.

Another interesting characteristic of this experiment is that for low frequencies the overlap area is bigger than for the high frequencies. It is known that the low frequencies detect the shape of objects, while the high frequencies are useful in detecting the sharp edges. In our testing data set the lightness conditions change throughout the image. In order to make the Gabor feature extraction be independent of lightness conditions we apply illumination normalization to the feature matrices.

We are interested in finding the histogram with the minimum overlapping area between two clusters. The histogram that minimizes this overlap corresponds to the Gabor feature space, where the features are best represented. The histogram will also help us choose the similarity measure which should be used in this particular classification problem. As it can be seen from Figure 5.7 we get the minimum overlapping area using the Euclidean distance. So for the classification of this set of images we decide to use the Euclidean distance with Gabor filter parameters: maximum frequency = 2, number of frequencies = 4, number of orientations = 8 and with illumination invariance.

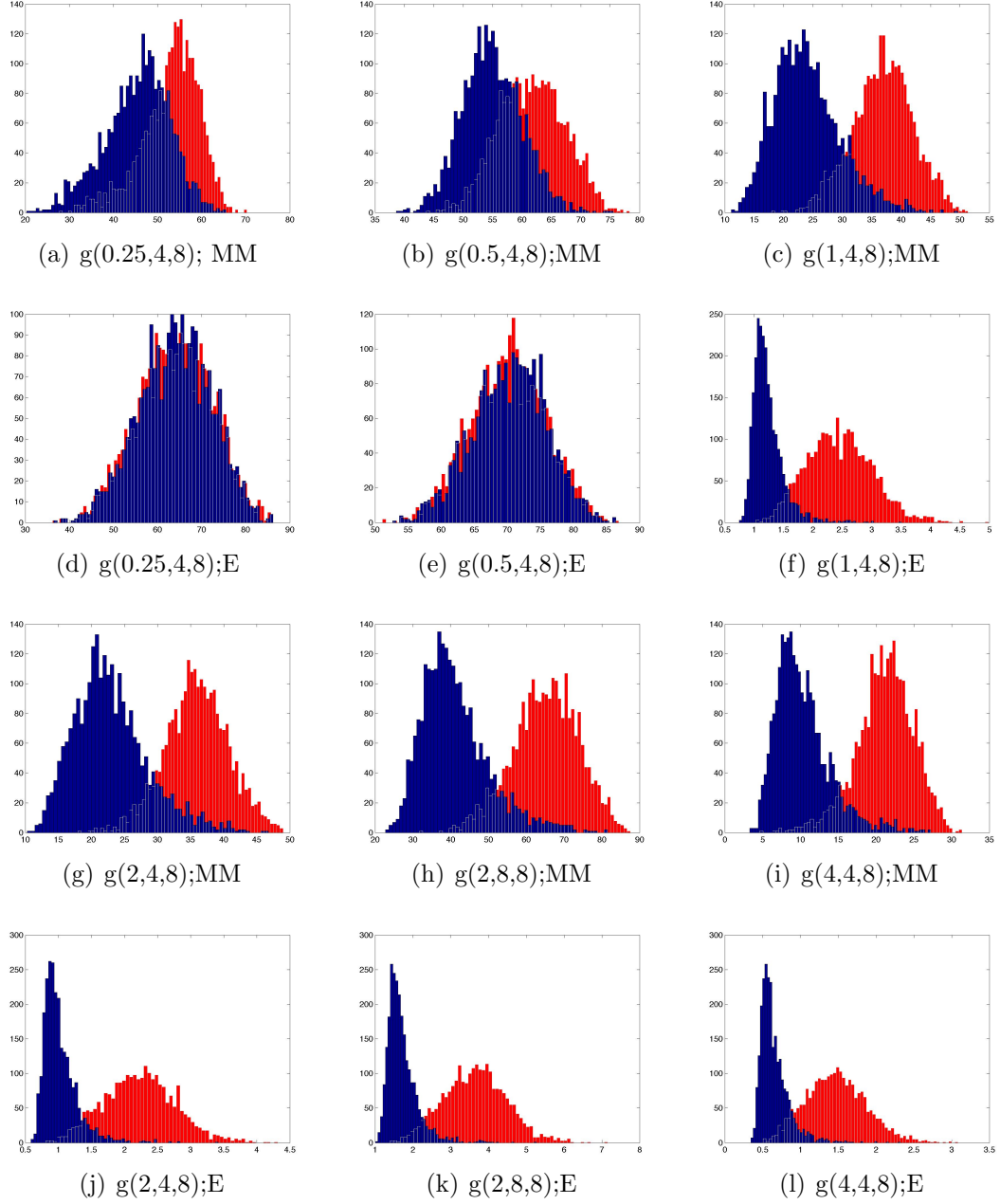


Figure 5.7: Comparing the separation between Gabor feature spaces of images pertaining to different classes. Two methods are used for computing the distance between Gabor feature spaces constructed using different Gabor filter parameters. Blue histogram represents non-tree vs non-tree. Red histogram represents non-tree vs tree. MM = Manjunath and Ma distance, E = Euclidean distance. We apply illumination normalization.

5.3 Classification Threshold

This section describes the method used to find a threshold for classification. For feature extraction we use the Gabor filter parameters chosen in the previous section. The reference image, s_r is the 10×10 top left non-tree area of image (a) from Figure 5.6. We compute the ROC curve using different threshold values τ in the intervals of the overlapping areas of the plots (g) and (j) from Figure 5.7.

For each τ , to compute the true positive rate, (4.12), we compare the reference image with all 10×10 sub-patches of image (b) from Figure 5.6. The percentage of the correctly classified non-tree areas will be the TPR value. To compute the false positive rate, (4.13), we compare the reference image with all 10×10 sub-patches of image (c) from Figure 5.6. The percentage of the incorrectly classified instances as non-tree will be the FPR value. We do this for both similarity measures used for distance calculation, i.e. the MM and E distances. As it is shown in Figure 5.8 the

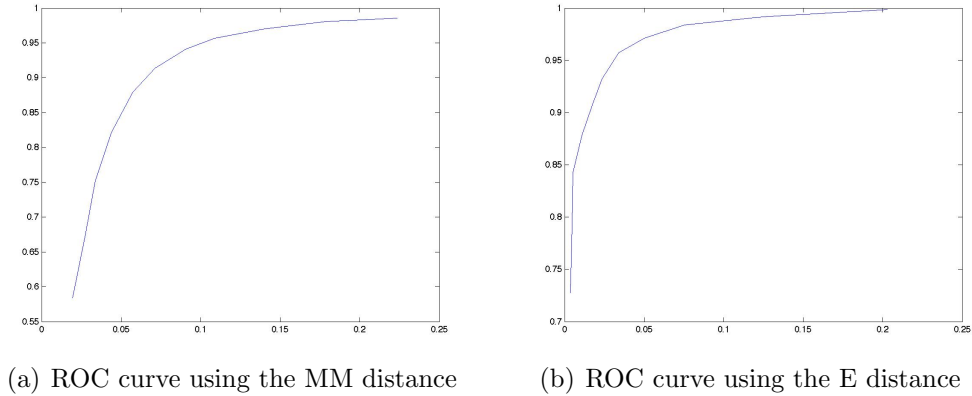


Figure 5.8: ROC curve

ROC curve using the E distance is more L shaped. This implies that the E distance is giving more accuracy in the classification results. The classification threshold which gives the coordinates closest to the point (0,1) in Figure 5.8 (a) is 24, while from graph (b) in Figure 5.8 the corresponding classification threshold is 1.7 . These two

thresholds are used for classifying new data sets using the Manjunath and Ma distance and the Euclidean distance, as explained in Section step 2 of the procedure in Section 4.11 of Chapter 4.

5.4 Classification results using the distance-based classification method

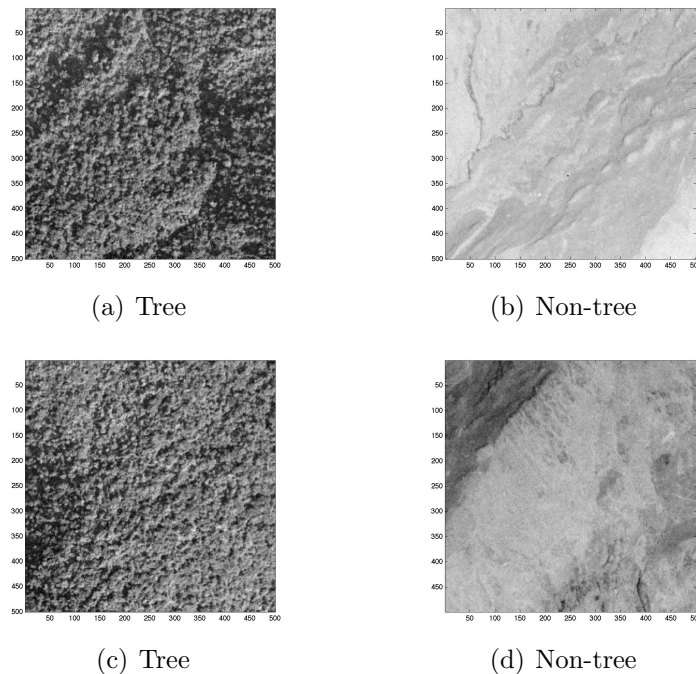


Figure 5.9: New images used for testing our classification methods.

New images were chosen to apply our classification method. The images (a), (c), and images (b), (d) from Figure 5.9 represent tree and non-tree areas respectively. In this section we experiment with different images and show their classified versions using the simple classification method described in the previous section. Figure 5.10 shows how the 10x10 sub-images of tree and non-tree area look like. Figure 5.11 shows the Gabor filter responses for these areas in Gabor feature space. Notice the visual difference in the responses between tree and non-tree areas.

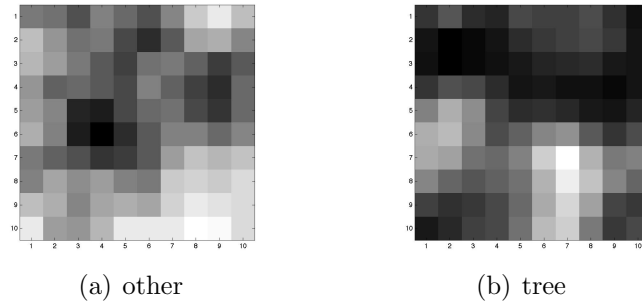


Figure 5.10: 10x10 tree and non-tree images

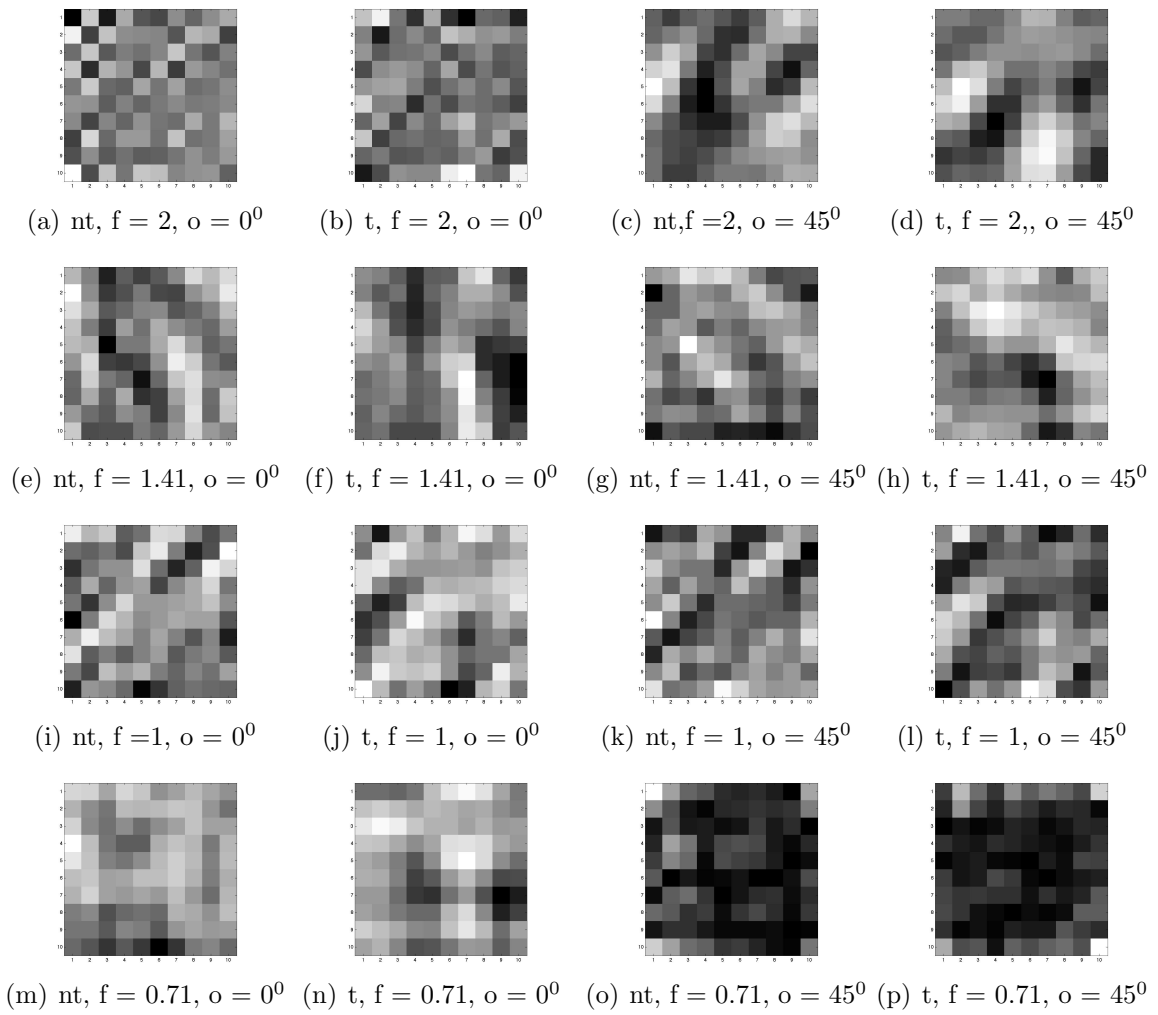


Figure 5.11: Gabor filter responses of the 10×10 sub-images of tree and non-tree areas, for different Gabor filter parameter values. f = frequency value, o = orientation degree, t = tree, nt = non-tree

Tables 5.1 and 5.2 show confusion matrices obtained by classifying the images in Figure 5.9 using the MM and E distances. 5000 images of tree areas and 5000 images of non-tree areas were classified (each of size 10×10). As it can be noticed from the tables the MM distance shows better accuracy when classifying tree areas but the E distance shows better overall accuracy.

MM distance	Tree area	Non-tree area
Classified as tree	4974	1692
Classified as non-tree	26	3308

Overall accuracy	Tree accuracy	Non-tree accuracy
82.82 %	99.48 %	66.16 %

Table 5.1: Confusion Matrix for classification using the Manjunath and Ma distance

E distance	Tree area	Non-tree area
Classified as tree	4348	38
Classified as non-tree	359	4962

Overall accuracy	Tree accuracy	Non-tree accuracy
93.1 %	86.96 %	99.24 %

Table 5.2: Confusion Matrix for classification results using the Euclidean distance

The average classification time per image (of size 10×10), using the MM method was 64 ms (classification method was implemented in Matlab). For the E distance the average classification time was 26 ms.

In Figure 5.12 we show final results of classification using the classification method described in the previous section. Black areas represent image regions classified as trees. White areas represent non-tree regions. Original images start from the left, and on their right are their classified versions. Notice that even in images where the tree features are less distinguishable (such as image (o) from Figure 5.12) the classification results are good. At the same time in some cases where the non-tree features are

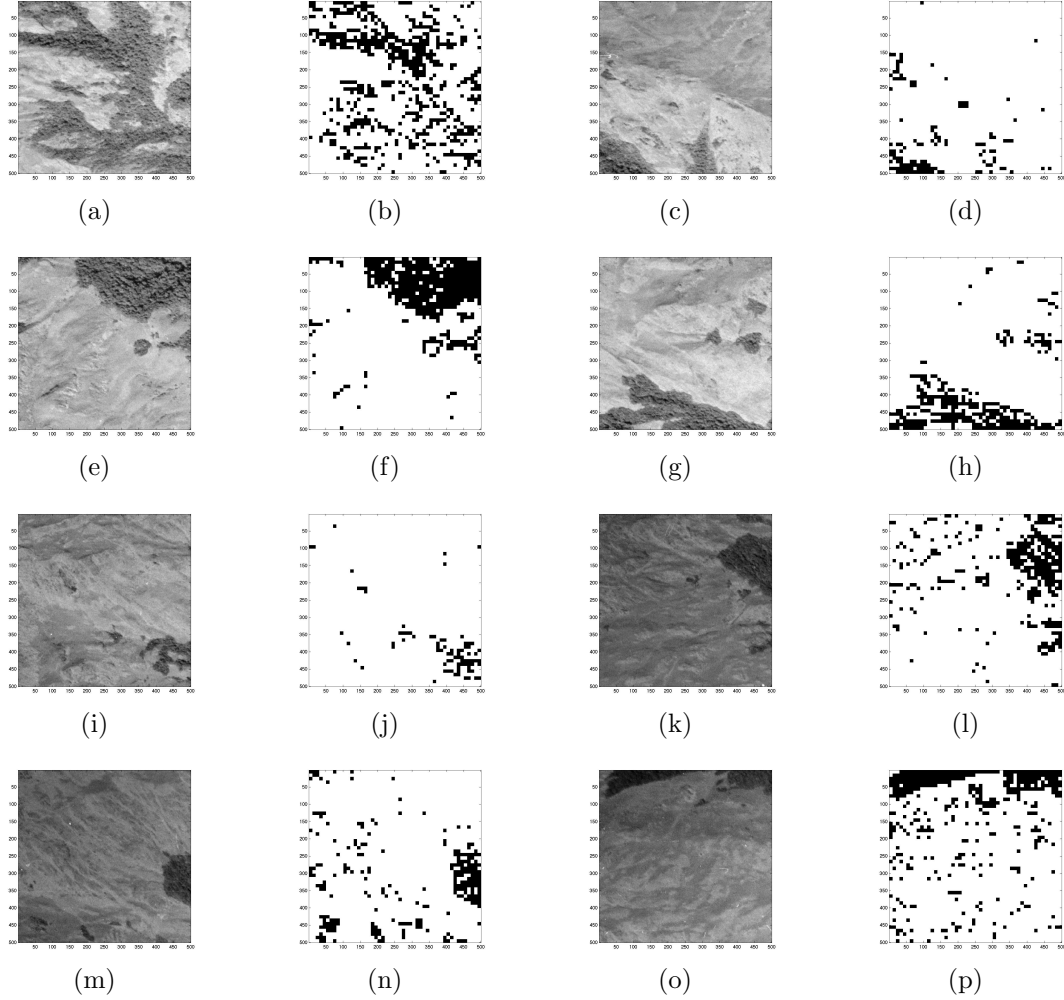


Figure 5.12: Original images(taken from Figure 5.1 and their classified versions using the Gabor feature extraction method)

very similar to the tree features the classifier fails to distinguish some of the images as non-tree areas. The most noticeable feature of the tree areas is the round shape of the top of the trees (since these are aerial images). In some images (such as parts of images (k) and (o) from Figure 5.12) this classification method fails to distinguish some non-tree areas which have round shapes as trees.

To refine this problem, in the later sections we will increase the number of images in the training part, so that non-tree and tree areas of different features are present.

In cases where the resolution of the image is poor, and there are no distinguishable features for the tree areas, the classifier fails to classify them as trees.

5.5 Results for the probabilistic classification method

This section describes the experiments performed using the probabilistic classification method (referred to as PCM method throughout the rest of this thesis) explained in Section 4.2. Like in the testing for the distance-based classification method (referred to as DCM throughout the rest of this thesis) we select the tree and non-tree images shown in Figure 5.9 for testing our second classification method.

PCM	Tree area	Non-tree area
Classified as tree	2420	75
Classified as non-tree	80	2425

Overall accuracy	Tree accuracy	Non-tree accuracy
96.9 %	96.8 %	97 %

Table 5.3: Confusion Matrix for the PCM method (using 30,614 features, tested on Figure 5.9 (a), (b)).

Table 5.3 shows the classification results obtained using the PCM method. Notice that this method performs better than the DCM using the E distance, in classifying tree areas and it is much more accurate than the DCM using the MM distance, in classifying non-tree areas. The overall accuracy of the PCM method is better than DCM method for both similarity measures. For the confusion matrix shown in Table 5.3 we apply mutual information and the total number of features used for classification is 30,614.

The average classification processing time for a 10×10 image was 249.16 ms when using 30,614 features (using a C++ implementation). To test how this approach performs using less number of features, we conducted a number of experiments. Our

goal here is both to measure the performance of this classification approach and to find the minimum number of features to use which would not compromise so much the performance of the classifier. Experimental results are shown in the confusion matrices below.

PCM	Tree area	Non-tree area
Classified as tree	2414	75
Classified as non-tree	86	2425

Overall accuracy	Tree accuracy	Non-tree accuracy
96.78 %	96.44 %	97 %

Table 5.4: Confusion Matrix for the PCM distance method (using 18,457 features, tested on Figure 5.9 (a), (b)). Average classification time per image(of size 10 x 10): 156.17ms

PCM	Tree area	Non-tree area
Classified as tree	2405	75
Classified as non-tree	95	2425

Overall accuracy	Tree accuracy	Non-tree accuracy
96.6 %	96.2 %	97 %

Table 5.5: Confusion Matrix for the PCM method (using 9,572 features, tested on Figure 5.9 (a), (b)). Average classification per time image(of size 10 x 10): 96.96ms

PCM	Tree area	Non-tree area
Classified as tree	4830	109
Classified as non-tree	170	4891

Overall accuracy	Tree accuracy	Non-tree accuracy
97.21 %	96.6 %	97.82 %

Table 5.6: Confusion Matrix for the PCM method (using 9,572 features, tested on all the images of Figure 5.9). Average classification per time image(of size 10 x 10): 96.96ms

As it can be inferred from the above tables we clearly find that reducing the number of features used (from 30,614 to 9,572) does not significantly decrease the accuracy of the classifier. At the same time reducing the number of features to 9,572 speeds up the average classification time by almost a factor of 3 (compare Tables 5.5 and 5.3). Based on the above results we decided to use 9,572 features for the rest of the aerial image classification using the PCM method. One of the major advantages of the PCM method over the DCM method is that we do not need to choose a reference image for classification of new data sets.

The problem associated with choosing a reference image is that it can have a positive or negative effect in the classification results. If a 'good' reference image is chosen (by 'good' we mean an image which best represents in general the class which it pertains to) then the classification results will be better and if a 'bad' reference image is chosen, the accuracy of the classifier will decrease. There is always a dose of ambiguity associated with choosing a reference image, which adds one more constraint to the classification process.

Another advantage of the PCM method is that a classification threshold is not needed. This saves time in the classification process. The PCM classification method uses conditional probabilities and the features selected by mutual information (see section 4.2) to classify between images. The two classification methods, DCM and PCM, do have one part of the process in common, they share the Gabor feature space. In terms of average classification time, DCM is much faster using E distance (26 ms), compared to the DCM using MM distance (64 ms) and the PCM (96.96 ms). In Figure 5.13 we show a visual approximation of the image areas using the classification method described in this section. Black areas represent image regions classified as trees. White areas represent image regions classified as non-trees. More experimental results, comparing the accuracy of our classification methods, will be shown in later

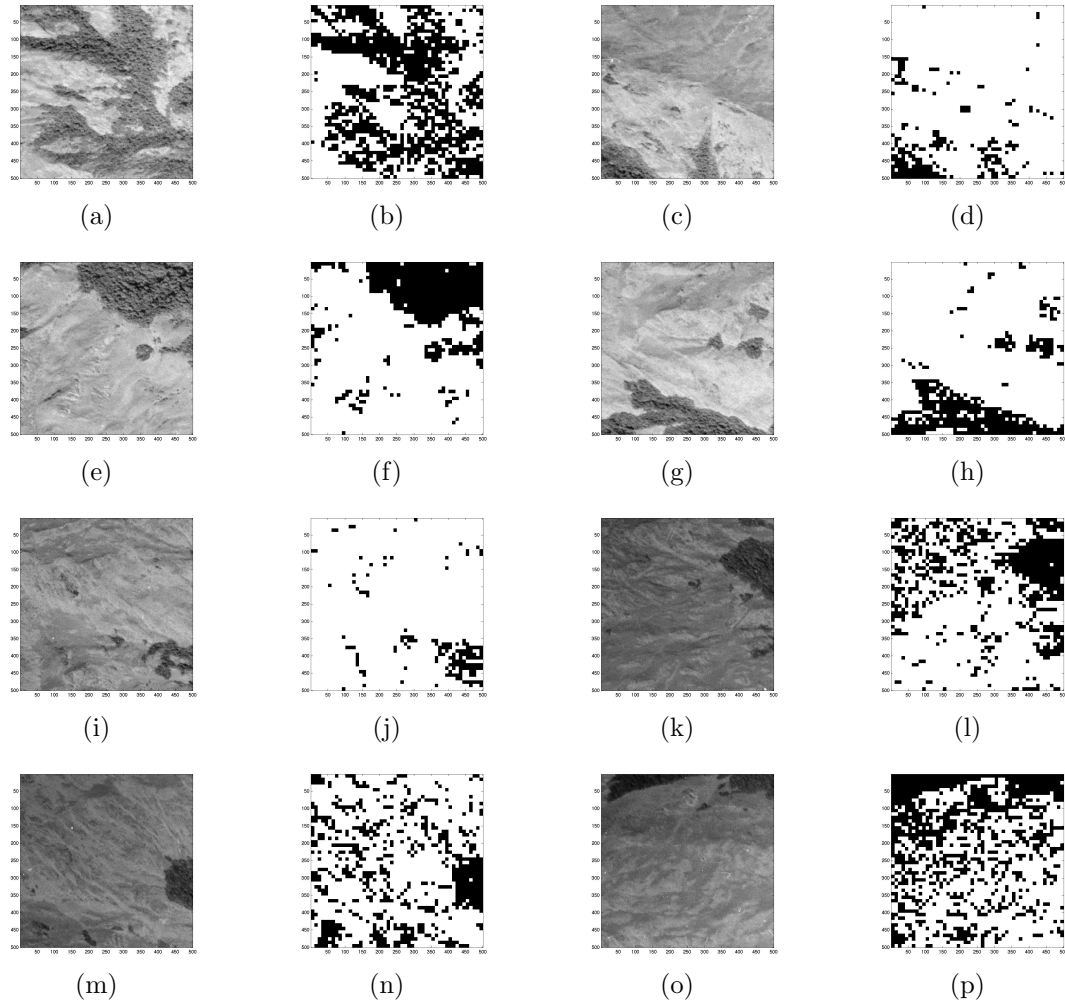


Figure 5.13: Original images(taken from Figure 5.1) and their classified versions using the PCM classification method

sections. Note that the PCM method misclassifies in some visual approximations (for image (k), image (m), and image (o) from Figure 5.13). This implies that we need to increase the number of training image examples which contain different features. We explore this approach in later sections.

5.6 Classification Results using the SVM-based classification method

In this section we present the classification results using our third method of classification. The subject consists of the images from Figure 5.9. The input to the SVM-based classification are the conditional probabilities of tree and non-tree Gabor features as explained in section 4.3. Table 5.7 shows the confusion matrix and classification results obtained.

SVM method	Tree area	Non-tree area
Classified as tree	4882	92
Classified as non-tree	118	4908

Overall accuracy	Tree accuracy	Non-tree accuracy
97.9 %	97.64 %	98.16 %

Table 5.7: Confusion Matrix for the SVM method. Average classification time per image(of size 10x10): 52.4ms

Notice that this method performs better than the PCM method in classifying non-tree areas and tree areas. It performs better than the DCM method using the E distance, in classifying the tree areas.

The overall accuracy of the SVM method is better than the DCM and PCM methods. Recall that the features used in this classification method are the conditional probabilities obtained using the naive Bayes approach (see Section 4.2). The average classification time was 52.4 ms, from which it takes 48.8 ms to output the features in

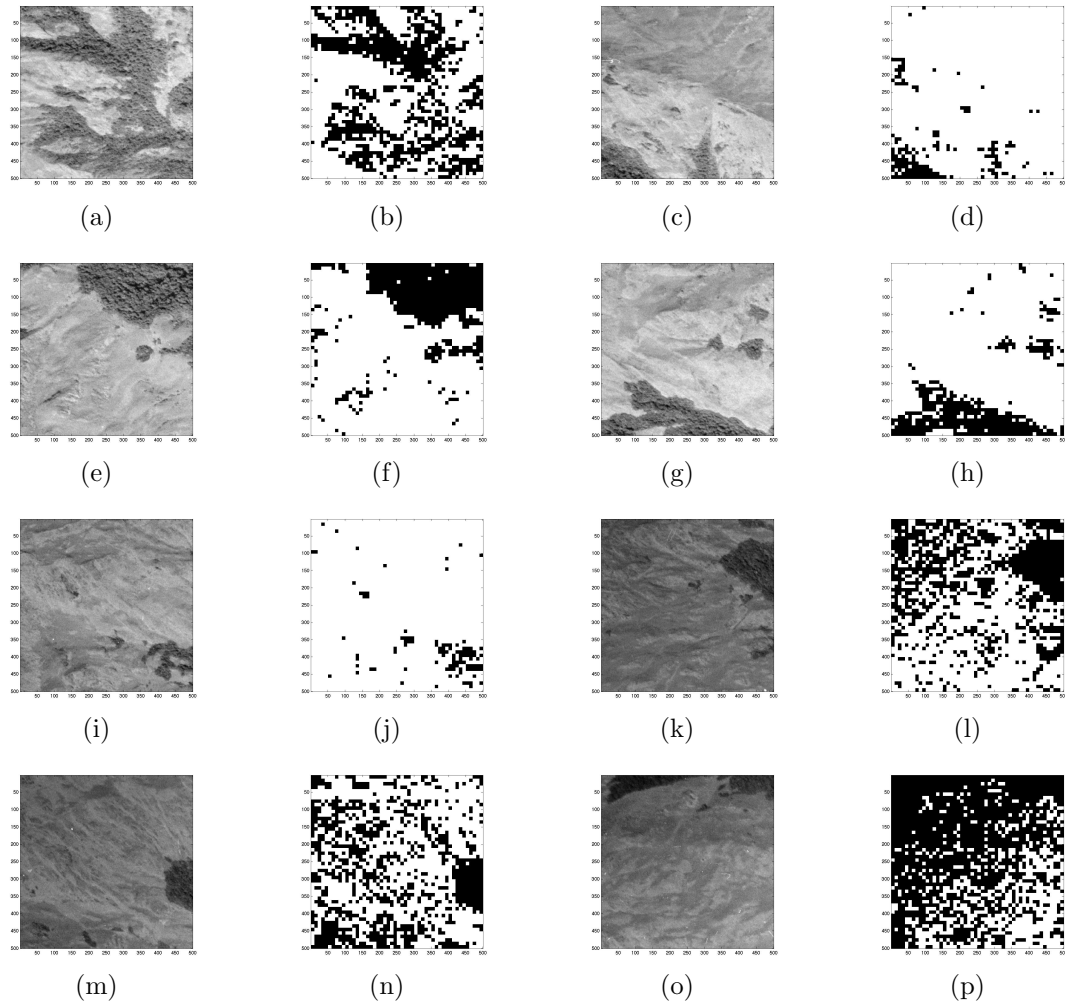


Figure 5.14: Original images (taken from Figure 5.1) and their classified versions using the SVM classification method.

correct format, and it takes only 3.6 ms for the SVM classifier to classify a 10×10 image. In Figure 5.14 we show a visual approximation of the image areas obtained using the SVM classification method.

As it can be seen from Figure 5.14 the SVM classification method misclassifies some areas of the images (for images (k), (m), and (o)). To fix this problem requires to increase the number of training examples having different features.

5.7 Robustness Testing

Finally, we want to test the robustness of the proposed classification methods when classifying images having different features. The new testing image data consists of 41,000 sub-images from the images shown in Figure 5.15. This new dataset contains non-tree areas having many different spatial features and tree areas having different illumination (see Figure 5.15). The goal of the experiment was to test the accuracy of the PCM and SVM when classifying images having different features, especially for the non-tree areas.

We first performed classification using the already trained classifiers presented in Section 5.2. Tables 5.8 and 5.10 show the classification results obtained using PCM and SVM, respectively. before retraining our classifiers. We then retrained (the same Gabor feature space is used) 5000 new tree sub-images and 6500 new non-tree sub-images. As it can be observed from Tables 5.8 and 5.9, the overall classification accuracy using PCM is almost the same as it was before retraining. Nevertheless the non-tree accuracy increases and the tree accuracy decreases after retraining. This implies that there is a trade-off associated with trying to achieve higher accuracy results for non-tree areas, as in our image data the non-tree areas have more varying features than the tree areas. Tables 5.9 and 5.11 show the classification results obtained using the retrained PCM and SVM, respectively. The overall accuracy of the

SVM decreases after retraining by 1% (see Figures 5.10 and 5.11). The tree accuracy using SVM decreases more compared to that of PCM (see Tables 5.9 and 5.11).

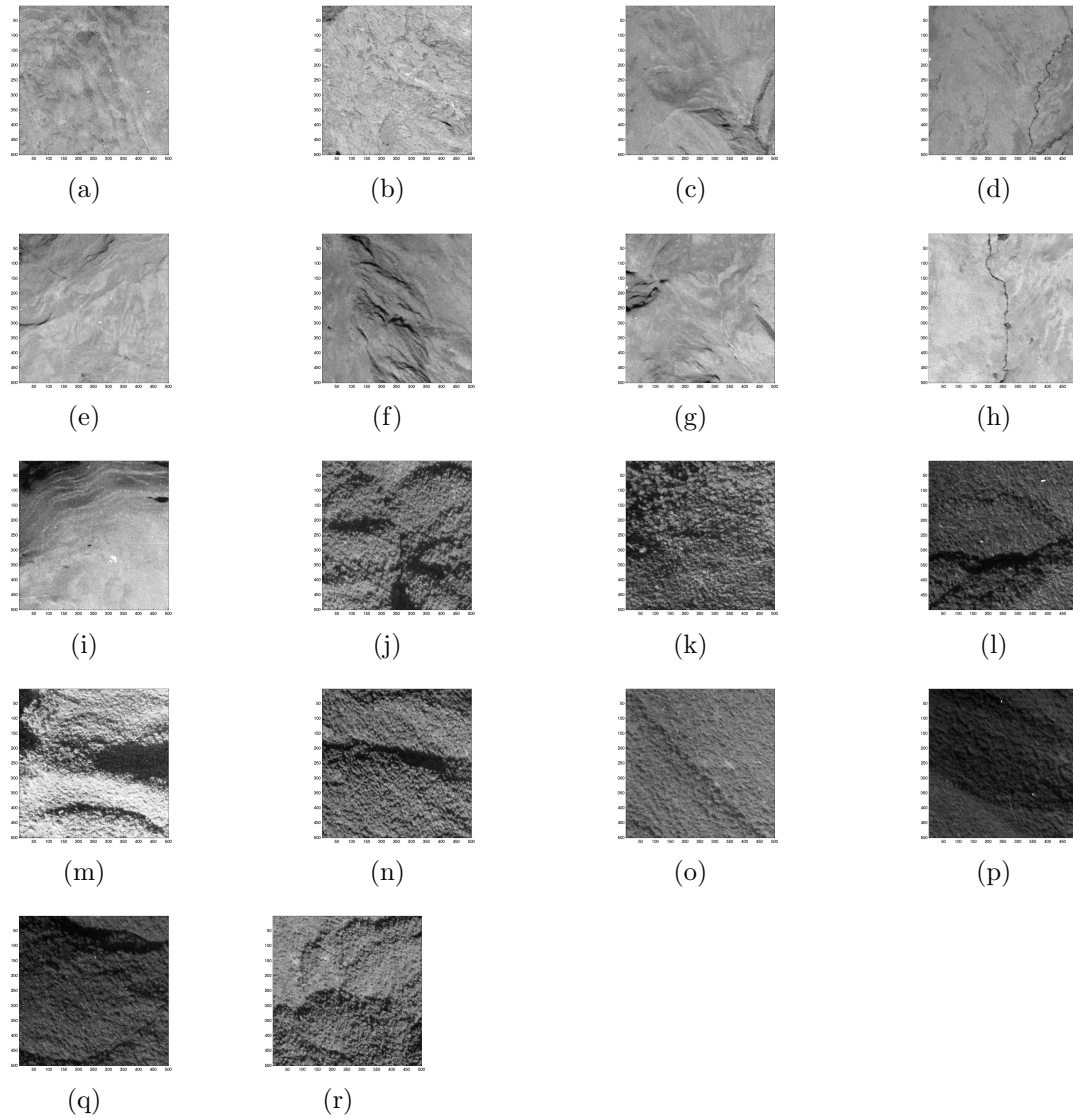


Figure 5.15: New data set used for testing. Sub-images (a) through (i) consist of non-tree areas with different features. Sub-images (j) through (r) consist of tree regions with different features.

PCM method	Tree area	Non-tree area
Classified as tree	19,981	1,774
Classified as non-tree	519	18,726

Overall accuracy	Tree accuracy	Non-tree accuracy
94.41 %	97.47 %	91.35 %

Table 5.8: Confusion Matrix for the PCM method for the new data set before re-training.

PCM method	Tree area	Non-tree area
Classified as tree	19,565	1,323
Classified as non-tree	935	19,177

Overall accuracy	Tree accuracy	Non-tree accuracy
94.49 %	95.45 %	93.55 %

Table 5.9: Confusion Matrix for the PCM method for the new data set after retraining.

SVM method	Tree area	Non-tree area
Classified as tree	20,070	2,100
Classified as non-tree	430	18,400

Overall accuracy	Tree accuracy	Non-tree accuracy
93.83 %	97.9 %	89.76 %

Table 5.10: Confusion Matrix for the SVM method for the new data set before re-training.

It is interesting to note that for DCM when training with images containing different features the histograms of distance values overlap more. Figure 5.16 shows the histogram of the distance values using the Euclidean distance for the DCM. As it can be seen the overlapping area for the training data makes it hard to choose an optimal classification threshold. This implies that a new Gabor feature space would have to be selected using procedures in Section 5.2.

SVM method	Tree area	Non-tree area
Classified as tree	18,819	1,452
Classified as non-tree	1,681	19,048

Overall accuracy	Tree accuracy	Non-tree accuracy
92.36 %	91.8 %	92.9 %

Table 5.11: Confusion Matrix for the SVM method for the new data set after retraining.

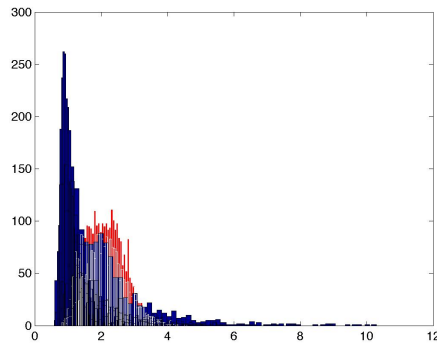
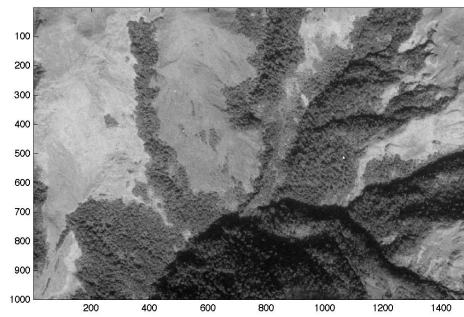
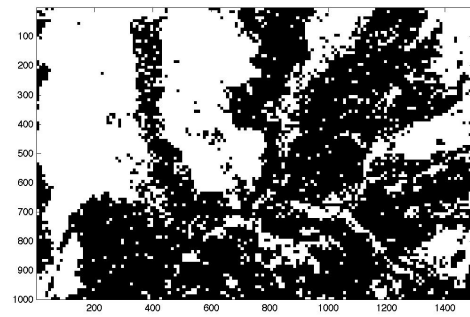


Figure 5.16: The histogram of the Euclidean distances for the new data set. The blue part shows the distances non-tree vs non-tree, the red part shows the distance values between non-tree and tree.

The results obtained in this experiment show that PCM is more robust and gives better accuracy results using the same Gabor feature space. Figure 5.17 shows the classification results obtained after applying the retrained PCM. Notice that the classification results on the binary image show good accuracy even in the areas where the illumination changes (especially the non-tree areas). Another important conclusion from this experiment is that PCM classifies incorrectly when shadows are present as it can be noticed in some tree areas of Figure 5.17. The reason for this is the lack of features in the presence of shadows.



(a) original image



(b) binary image

Figure 5.17: The original image on the left and the binary image on the right using the retrained PCM.

Chapter 6: Conclusions and Future Research

6.1 Discussion

This thesis presents three different Gabor filter based classification methods applied to the classification of information classes in aerial imagery. We use a combination of spatial-domain and frequency-domain analysis to better extract the spatial features of the image data. Representation of images in Gabor feature space has been utilized in the three different classification methods. Based on the illumination variance property of the image data and the relationship of the pixel to its neighbors we show that a simple threshold algorithm is not sufficient for accurate classification. A comprehensive performance evaluation of the methods is given using a large number of image datasets.

The mathematical theory and concepts behind Gabor filters were reviewed and shown to excel for the representation of certain information classes. Gabor filters exhibit fundamental invariance properties that make them very applicable for invariant object representation. We exploit the ability of Gabor filters to represent signals/images in time/space and frequency representation, thus providing joint resolution in the time/spatial frequency domains.

Bayesian networks, mutual information, and support vector machines are finding increasing application for image classification purposes. One of the main reasons is that Bayesian networks have tremendous ability to represent the probabilistic relationships between a large number of variables and for doing probabilistic inference. Similarly the ability of mutual information to measure the dependency between two variables enables the reduction of confounding variables.

In this thesis we have proposed the representation of image data in Gabor feature space for feature extraction. We also propose the use of three different methods for image classification in this space. Two total similarity measures are used for selecting an appropriate Gabor feature space, where two information classes of interest can be separated. We have shown that normalization of the Gabor feature space for the purpose of achieving illumination invariant responses can produce improved feature representation. For the first classification method we used the ROC curve to select a separation threshold. In the probabilistic classification method we showed that the magnitudes of image responses in Gabor feature space can be effectively used to train a naive Bayes classifier. Mutual information was applied for selecting informative Gabor features for image classification, however one must be careful in the selection of mutual information threshold to avoid misclassification. A third classification method based on support vector machines has been proposed. One major advantage of SVM-based classification is that there are few model parameters to pick for the learner and the classifier. From the results obtained, it appears that linear classification in Gabor feature space is appropriate.

A large number of experimental results have demonstrated the effectiveness of the three classification methods proposed in Chapter 4. Extensive experimentation and comparisons using real data show the accuracy, the advantages, and disadvantages of the three classification methods when applied to aerial imagery. In the observation experiment we showed the difficulty associated with selecting the proper Gabor feature space which maximizes the total similarity measure proposed by Manjunath and Ma. It is shown that without applying illumination normalization it becomes very hard to solve the min-max optimization problem presented in Chapter 4.

The Manjunath and Ma versus Euclidean distance experiments demonstrated the performance of these two similarity measures. This experiment also proved the im-

portance of applying illumination normalization to the Gabor feature space for images which are characterized by changes in the intensity magnitudes. In this experiment we chose the number of frequency and orientation parameters defining the Gabor feature space to be used in all of the three classification methods. The ROC curve was demonstrated to be an efficient tool for choosing the classification threshold which produces the combination of the lowest false positive rate and the highest true positive rate.

The Euclidean distance proved to be a better similarity measure than the Manjunath and Ma distance, giving better classification accuracy results. The classification results for the probabilistic classification method showed better accuracy. The experiments revealed this method's ability to classify better using a lower number of Gabor features. Another advantage of this method is that a reference image is not needed for classification of new data sets. In Section 5.11 the confusion matrix for the SVM-based classification method shows better overall accuracy results compared to the other two methods. We used conditional probabilities and the mutual information feature vector as input to the SVM learner and SVM classifier.

In the final experiments, we test the robustness of the proposed classification methods. The probabilistic classification method proves to be more robust than the other two classification methods when classifying images with different features. The classification results for the distance-based classification method using Euclidean distance showed the necessity of choosing a different Gabor feature space for finding the optimal classification threshold. The results of this experiment showed that probabilistic and SVM-based classification methods classify with high accuracy even when using the same Gabor feature space.

6.2 Future Work

The new methods proposed in this thesis open many areas to be explored. Future research could be done in the following topics:

Feature Representation

This thesis uses Gabor filters to extract features which are then used as input to the classification methods. There are other feature spaces, such as the short-time Fourier transform, spectrogram, Wigner-Wille distribution, time-frequency kernel methods, and the wavelet transform [11], which could be explored or combined in addition to Gabor features.

Gabor filter parameters

We have proposed a simple procedure for selecting Gabor filter parameters. The fact is that choosing the Gabor filter parameters which optimally separate images in the Gabor feature space is a difficult optimization problem. Choosing the frequency and orientation parameters is application dependent and a general algorithm for choosing these parameters would reduce the classification time and increase the accuracy.

SVMs

For future experiments using the SVM-based classification method, cross validation could be used for the training part. Also the ROC curve could be used to optimize the error parameter involved in the SVM method. In this thesis we use a linear kernel. Can other types of nonlinear kernels be used to improve performance compared to the linear kernel? As mentioned in the previous section we used conditional probabilities as input to the SVM learner and SVM classifier. One other feature input could be

obtained by counting the number of times a Gabor feature value appears in the Gabor feature space of the training images.

Reference Image Selection

In the distance-based classification method we use a reference image to train our learner and to classify new data sets. The question that arises is: What would be the best reference image to choose? There are other methods which could be used, such as choosing two reference images and then classifying based on the value of the similarity measure (the smaller the distance between the reference image and the target image in the Gabor feature space, the better the chance that the target image belongs to the class of the reference image). One other way would be to have a database of reference images, where each one of them contain different features. However these other methods would increase the computation time. It would be very beneficial to develop a scheme (such as an unsupervised classification method) which would find a reference image that in general has most of the necessary patterns/features for classification. This would improve the classification results.

Redundancy

In this thesis we use mutual information to reduce the number of redundant Gabor features. Other methods can be explored for reducing the redundancy involved in the classification process, such as other information theoretic approaches.

Similarity measures

Two similarity measures have been used in this thesis to compute the distance between images in the Gabor feature space. There are other similarity measures which could be explored.

Noise

In the experiments done with different images we noticed that when there is noise present in the images the accuracy of our classification methods can be affected. More work can be done to integrate knowledge of noise removal in our methods.

Size of the images being classified

In this thesis we chose to classify on 10×10 sub-images. After analyzing our aerial image data we came to the conclusion that if a bigger region is chosen to be classified the misclassification level can be increased. For example if 60% of the image area has tree features then our classifier would classify it as tree, and thus misclassify the rest 40% of land. We decided that if a smaller region is chosen then it will be harder for the classifier to distinguish between tree and non-tree areas because small regions would contain very similar features. In general choosing the sub-image size must be done with respect to the size of the desired feature to be classified and the physical size of the image pixel.

References

- [1] Ali El Akadi, Abdeljalil El Ouardighi, and Driss Aboutajdine. A powerful feature selection approach based on mutual information. *IJCSNS International Journal of Computer Science and Network Security*, 8, 2008.
- [2] M. J. Bastiaans. Gabor’s signal expansion and the zak transform. *Applied Optics*, 33:5241–5255, 1994.
- [3] Kristin P. Bennet and Colin Campbell. Support vector machines: Hype or hal-lelujah? *ACM SIGKDD Explorations*, 2, 2000.
- [4] A.C. Bovic, M. Clark, and W.S. Geisler. Multichannel texture analysis using localized spatial filters. *IEEE Trans. Patter Analysis and Machine Intelligence*, 12, 1990.
- [5] M.B. Bush, M.R. Silman, C. McMichael, and S. Saatchi. Fire, climate change and biodiversity in amazonia: a late-holocene perspective. *Philosophical Transactions of the Royal Society of London*, 2008.
- [6] J. Daugman. Probing the uniqueness and randomness of iriscodes: Results from 200 billion iris pair comparisons. *Proceedings of the IEEE*, 94, 2006.
- [7] J. G. Daugman. Uncertainty relation for resolution in space, spatial frequency, and orientation optimized by two-dimensional visual cortical filters. *Journal of the Optical Society of America*, 2:1160–1169, 1985.
- [8] Charles Elkan. Naive bayesian learning. Technical report, Harvard University, Department of Computer science.
- [9] Francois Fleuret. Fast binary feature selection with conditional mutual information. *Journal of Machine Learning Research*, 2004.
- [10] D. Gabor. Theory of communications. *Journal of Institution of Electrical Engineers*, 93:429–457, 1946.
- [11] F. Hlawatsch and G. Boudreaux-Bartels. Linear and quadratic time-frequency signal representations. *IEEE Signal Processing Magazine*, 9:21–67, 1992.
- [12] Anil K. Jain. *Fundamentals of Digital Image Processing*. Prentice Hall Information And System Sciences Series, 1989.

- [13] T. Joachims. Making large-scale svm learning practical. *Advances in Kernel Methods -Support Vector Learning*, B. Scholkopf and C. Burges and A. Smola(ed.), MIT-Press, 1999.
- [14] Thorsten Joachims. Training linear svms in linear time. *Department of Computer Science, Cornell University*, 2006.
- [15] J. K. Kamarainen, V. Kyrki, and H. Kalviainen. Robustness of gabor feature paramter selection. *IAPR Workshop on Machine Vision Applications*, 2002.
- [16] Joni-Kristian Kamarainen. *Feature Extraction Using Gabor Filters*. PhD thesis, Lappeenranta University of Technology, 2003.
- [17] Andrew Karode. Support vector machine classification of network streams using a spectrum kernel encoding. Master's thesis, Wake Forest University, 2008.
- [18] V. Kyrki, J.-K. Kamarainen, and H. Klviinen. Simple gabor feature space for invariant object recognition. *Pattern Recognition Letters*, 25, 2004.
- [19] B.S. Manjunath and R. Chellappa. A feature based approach to face recognition. *Proc. IEEE Conf. CVPR*, 1992.
- [20] B.S. Manjunath and W.Y. Ma. Texture features for browsing and retrieval of image data. *IEE Transcations on Pattern Analysis and Machine Intelligence*, 18, 1996.
- [21] B.S. Manjunath, C. Shekhar, and R. Chellappa. A new approach to image feature detection with applications. *Pattern Recognition*, 1996.
- [22] R. Narayanswamy, P. Silveira, H. Setty, V. Pauca, and J. van der Gracht. Extended depth-of-field iris recognition system for a workstation environment. *Proc. SPIE Biometric Technology for Human Identification II*, 5779, 2005.
- [23] Richard E. Neapolitan. *Learning Bayesian Networks*. Prentice Hall Series In Artificial Intelligence, 2004.
- [24] Judea Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible inference*. Morgan-Kaufmann, 1988.
- [25] S. Qian and Chen. Joint time-frequency analysis. *IEEE Signal Processing Magazine*, 2:52–67, 1999.
- [26] Mehran Sahami, Susan Dumais, David Heckerman, and Eric Horvitz. A bayesian approach to filtering junk e-mail. *Computer Science Department Stanford University, Microsoft Research*, 1998.

- [27] Linlin Shen and Li Bai. Information theory for gabor feature selection for face recognition. *EURASIP Journal on Applied Signal Processing*, 2006, 2005.
- [28] M.R. Silman. Plant species diversity in amazonian forests. *ch. 11 in Bush M. and Flenly J. eds. Tropical Rain Forest Responses to Climate Change*, 2006.
- [29] N. Smith, V.P. Pauca, A. Ross, T. Torgersen, and M.C. King. Extended evaluation of simulated wavefront coding technology in iris recognition. In *IEEE Conference on Biometrics: Theory, Applications and Systems*, 2007.
- [30] Zehan Sun, George Bebis, and Ronald Miller. On-road vehicle detection using gabor filters and support vector machines. *Computer Vision Laboratory, University of Nevada. Reno e-Technology Department, Ford Motor Company*, 2002.
- [31] G. D. Tourassi, E. D. Frederick, M. K. Markey, and C. E. Floyd Jr. Application of the mutual information criterion for feature selection in computer-aided diagnosis. *Medical Physics*, 28, 2001.
- [32] J. Yang, L. Liu, T. Jiang, and Y. Fan. A modified gabor filter design method for fingerprint image enhancement. *Pattern Recognition Letters*, 24:1805–1817, 2003.

Vita

Sebastian Berisha

EDUCATION

Master of Science Computer Science

Wake Forest University, May 2009

Thesis: Image Classification using Gabor Filters and Machine Learning

Bachelor of Science Computer Science, Mathematics (double major)

Summa Cum Laude

Averett University, Danville, VA, April 2006

HONORS AND ACTIVITIES

Averett University: President's List, Alpha Chi Award, Outstanding Computer Science and Mathematics Award, Mary C. Fugate Award, President's Scholarship, Member of Alpha Chi Honor Society.

Wake Forest University: Wake Forest Graduate School Scholarship, Member of Advanced Imaging Group, Member of Upsilon Pi Epsilon.

Other: Collegiate All-American Scholar Award, The National Dean's List, Elected to Who's Who Among Students in American Universities and Colleges, Madeleine Albright Scholarship.

EXPERIENCE

- *Teaching Assistant*

Department of Computer Science, WFU, Winston-Salem, NC

Lead lab sessions in Fundamentals of Computer Science course. Responsible for preparing and grading homework, lab assignments, quizzes and exams.

- *Research Assistant*

Advanced Imaging Group, WFU, Winston-Salem, NC

Researched and studied scientific papers to analyze, adapt, optimize, and test techniques or algorithms fitting project goals. Utilized sophisticated estimation and detection algorithms for estimating the model parameters for dehazing images.

- *Part-time Summer Intern*

Great Wall Systems, Winston-Salem, NC

Installed and configured *TILE64TM* Processor, a 64 core processor interconnected with Tiler's *iMeshTM* on-chip network, used for implementing distributed firewall architecture.