Palestine Polytechnic University
Deanship of Graduate Studies and Scientific Research
Master of informatics

Vision-Based Flowering Plant Detection and Recognition

Submitted by:

Ala' Mohammad Ahmad Jabari

Thesis submitted in partial fulfillment of requirements of the degree Master of Science in Informatics

May, 2013

The undersigned hereby certify that they have read, examined and recommended to the Deanship of Graduate Studies and Scientific Research at Palestine Polytechnic University the approval of a thesis entitled: **Vision-Based Flowering Plant Detection and Recognition**, submitted by **Ala' A. Jabari** in partial fulfillment of the requirements for the degree of Master in Informatics.

Graduate Advisory Committee:

Dr. Hashem Tamimi (Supervisor), Palestine Polytechnic University.

Signature: Hashem Tamimi Date: 22.07.2013

Dr. Dia Abu Zenah (Internal committee member), Palestine Polytechnic University.

Signature:________ Date: 21.7.2013

Dr. Abualsaud Hanani (External committee member), Birzeit University.

Signature: abualsoud Hanani Date: 10.7.2013

Thesis Approved

Dr.Sameer Khader

Dean of Graduate Studies and Scientific Research

Palestine Polytechnic University

Signature:________ Date: 31.07.2013

DECLARATION

I declare that the Master Thesis entitled **"Vision-Based Flowering Plant Detection and Recognition"** is my original work, and hereby certify that unless stated, all work contained within this thesis is my own independent research and has not been submitted for the award of any other degree at any institution, except where due acknowledgement is made in the text.

Ala'Jabari

Signature:____________________ Date:______________

STATEMENT OF PERMISSION TO USE

In presenting this thesis in partial fulfillment of the requirements for the master degree in Informatics at Palestine Polytechnic University, I agree that the library shall make it available to borrowers under rules of the library. Brief quotations from this thesis are allowable without special permission, provided that accurate acknowledgement of the source is made. Permission for extensive quotation from, reproduction, or publication of this thesis may be granted by my main supervisor, or in his absence, by the Dean of Graduate Studies and Scientific Research when, in the opinion of either, the proposed use of the material is for scholarly purposes. Any coping or use of the material in this thesis for financial gain shall not be allowed without my written permission.

Ala'Jabari

Signature:______________________ Date:_______________

DEDICATION

Many people have contributed in the success of my thesis, and I like to thank in particular, my parents for always being sensible and supportive. Most importantly, I would like to thank my husband for the continuous encouragements. My final thanks go to my sisters and brothers for their support.

ACKNOWLEDGEMENT

I would like to express my deepest appreciation to all those who provided me the possibility to complete this thesis.A special gratitude I give to my supervisor Dr. Hashem Tamimi as well for his assistance, guidance and encouragement. I would like to thank the thesis's examiner Dr. Abualsaud Hanani and Dr. Dia Abu Zenah for their valuable suggestions and corrections to this work, which greatly helped me to improve in various aspects. Furthermore I would also like to thank my family for their endless love and support.

الملخص

تعتبر مشكلة التعرف على النباتات المزهرة مشكلة صعبة بالنسبة للحاسوب. وقد تم حل هذه المشكلة جزئياً تحت شروط معينة وهي الكشف المسبق عن الأزهار في الصورة، كما ويجب أن تكون الزهرة معزولة بشكل مثالي تقريباً حتى يتم التعرف عليها. لقد تناولنا في هذه الأطروحة مشكلة الكشف عن مكان الزهرة والتعرف عليها، وأظهرنا قدرة الحاسوب على الكشف عن مكان الزهرة في صور تحتوي على مشهد كامل عن طريق التعلم الآلي. استخدمنا العديد من وسائل استخراج السمات والخصائص من الصورة لحل مشكلة الكشف عن مكان الزهرة ، ومن هذه الوسائل: تحويل المويجات المتقطعة والمدرج الإحصائي من التدرجات الموجه وتصفية غابور. اعتبرنا مشكلة الكشف عن وجود الأزهار على أنها مشكلة تصنيف، واستخدمنا مجموعة من أجهزة المتجهات الاعتمادية الخطية في التصنيف. وتم اعتماد منهجية وضعية الإنسان كنهج للكشف عن مكان الزهرة.

لتدريب المصنفات حتى تتعرف على نوع الزهرة، استخدمنا مجموعة من الصور القياسية وتم اختبار المصنفات المدربة على صور طبيعية باستخدام نافذة متعددة الأحجام للعثور على نوافذ تحتوي على الأزهار. للكشف عن مكان الزهرة في صور تحتوي على مشهد كامل، بنينا مجموعة بينات جديدة تحتوي على ١٠ أنواع من النباتات المزهرة. أظهرت التجارب أن سمات تحويل المويجات المتقطعة المستخدمة كمدخل في أجهزة المتجهات الاعتمادية الخطية تفوق السمات الأخرى في الأداء والوقت المستغرق في الكشف عن مكان الزهرة.

لقد حققنا دقة في الكشف عن مكان 10 أنواع من الأزهار بمعدل يصل إلى

حوالي 88% ، وأظهرت النتائج التجريبية باستخدام منحنى استقبال الخصائص التشغيلية أننا حققنا 0.92 من المنطقة تحت المنحنى. أما بالنسبة للتعرف على نوع الزهرة فقد حققنا دقة تصل إلى حوالي 81% و 0.85 من المنطقة تحت المنحنى.

Abstract

The problem of flowering plants recognition is considered as a challenging problem and it has been partially solved under a certain condition that the flowers be pre-detected and nearly optimal segmented. In this thesis, we study the flower detection and recognition approaches and we demonstrate the ability to detect flowers in full scene images by means of machine learning.

State of the art feature extraction methods are considered for the detection problems including: Discrete Wavelet Transform, Histogram of Oriented Gradients and Gabor Filter. We consider the flowering plant detection and recognition as classification problems. A set of Linear Support Vector Machines is used for classification. Poslets as a detection approach is considered for flowering plants detection.

For training the classifiers to recognize a flower, we use a set of benchmark images, the classifiers run over a natural test images using a multi-scale scanning window to find the strong activations. For full scene detection, We have built a new testing dataset of flowering plant species that include 10 flowering plant species. The experiments show that Discrete Wavelet Transform features as input for linear Support Vector Machine is superior in the performance of other features.

We have achieved an accuracy rate in detecting 10 flower species reaches about 88%. Experimental results using Receiver Operating Characteristics

show that we have achieved 0.92 area under curve. For recognition we have achieved an overall accuracy of 81% and Receiver Operating Characteristics of 0.85 area under curve.

Table of Contents

List of Figures

List of Tables

List of Abbreviations

Throughout this thesis we use the following acronyms:

- **AUC**: Area Under Curve.
- **CWT**: Continuous Wavelet Transform.
- **DWT**: Discrete Wavelet Transform.
- **GF**: Gabor Filter.
- **HOG**: Histogram of Oriented Gradients.
- **ROC**: Receiver Operating Characteristic.
- **SIFT**: Scale Invariant Feature Transform.
- **SPC**: Specificity.
- **SVM**: Support Vector Machine.

Chapter 1

Introduction

Object detection is one of the challenging fields in computer vision and image processing [10, 13, 19], which concerns in detecting and identifying the location of samples of objects in certain classes. Examples of classes are humans, cars, animals and plants, which can be detected in either static images or videos. The challenges in detecting objects are that objects in images may appear in different viewpoints, colors, scales and illuminations. The objects must also be detected even if they translated, rotated or partially occluded. The most important applications that object detection plays a major role in them are image retrieval and surveillance systems.

Recently, most of works in object detection contribute to human detection [24, 5, 37, 26] and plant detection [10, 22, 40]. The most commonly used way to detect such objects in images is based on using a sliding window over the image, then classifying the resulted local window to image that includes either the desired object or a background. The classification is based on the extracted features from images. This approach has been used in most applications such as face and pedestrian detection [24, 23, 14, 20].

In many applications, object detection is considered as a pre-processing phase to isolate the desired object from the background [28]. One interesting application that needs object detection or equivalent techniques to isolate objects from background is object recognition. The main difference between object detection and object recognition is that in object detection, objects are to be isolated from background, which means determine whether it belongs to a class or not (binary classification). But object recognition is more complicated, because object must be identified and classified to which class it belongs to from certain classes based on its features (multi-class classification). In plant detection and recognition systems, plants detection concerns at determining whether the window represents a plant or a background. But in plant recognition, the system must determine the type of the plant and to what class or family it belongs to. The difference between object detection and recognition is shown Figure 1.1.

In our work, we address the problem of detecting and recognizing flowering plant species in static image based on machine learning classifiers and image processing feature extraction approaches. Before expressing how we use computer vision techniques in detecting and recognizing plants, we will briefly discuss how botanist recognize flowering plant species.

Flowering plant species are recognized by botanist based on the characteristics of the flowers of a given species, which are shown in Figure 1.2. The most important characteristics that are used to identify the flowering plant species are:

1. Color of the flower.
2. Number and style of petals.
3. Number and style of sepals.
4. Number and style of stamens.

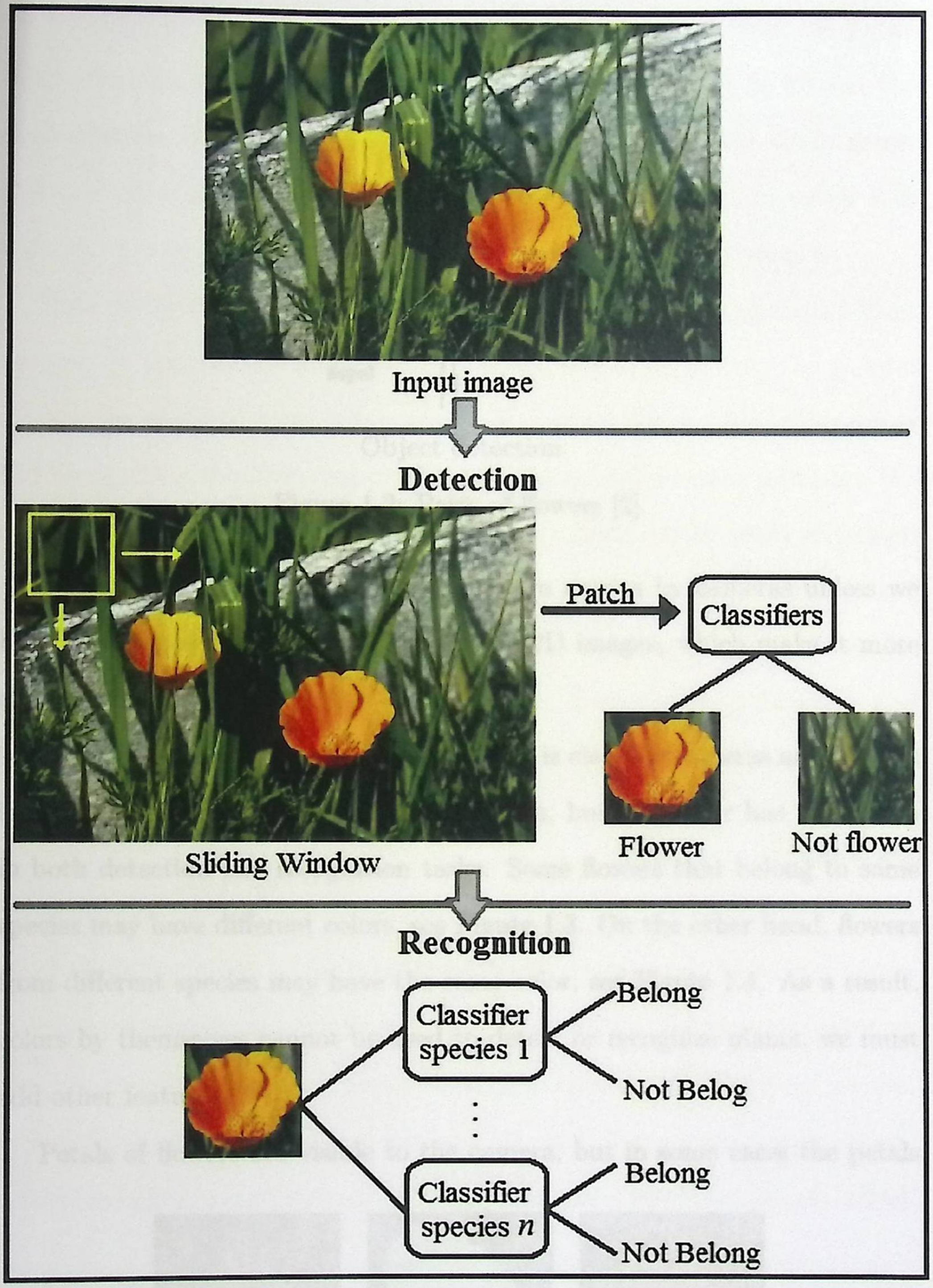
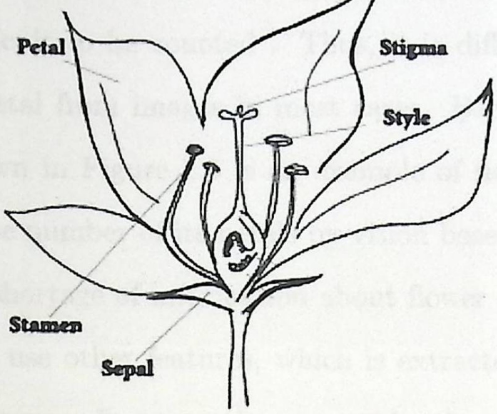


Figure 1.1: Flowering plant detection Vs. recognition.

5. Number and style of pistil.

6. Style of stalk.



Object detection

Figure 1.2: Parts of flowers [2].

Unfortunately, most of these features are unseen by cameras unless we have 3D images. In our work we only use 2D images, which make it more challenging.

Color information is a characteristic that is clear for cameras and it plays important roles in detection and recognition, but still color has a problem in both detection and recognition tasks. Some flowers that belong to same species may have different colors, see Figure 1.3. On the other hand, flowers from different species may have the same color, see Figure 1.4. As a result, colors by themselves cannot be used to detect or recognize plants, we must add other features.

Petals of flowers are visible to the camera, but in some cases the petals

Figure 1.3: Samples of flowers from one species with different colors.

are occluded by other objects in the image. Also in some flowers, the petals themselves are difficult to be counted . Thus, it is difficult to extract the exact number of petal from images in most cases. Barberton daisy flower species that is shown in Figure 1.5 is an example of flowers in which it is difficult to count the number of its petals by vision based techniques.

Because of the shortage of information about flower characteristics from images, we tend to use other features, which is extracted from the general appearance of the flowers. In our work, we consider the problem of detecting and recognizing the flowering plant species as a classification problem. We aim at comparing different feature extraction methods for plant detection and recognition. Also, build a vision based system that aids at detecting and recognizing flowering plant species. The system passes through four stages namely, image preparation, feature extraction, flowering plant detection and finally recognition.

1.1 Contribution

Many flowering plant recognition applications require pre-processing phase to isolate flowers from background. The applications are highly depend on image segmentation, which require sensitive care about capturing images, such as focusing on the plants or having a 3D view. For real time applications, natural images are captured without care of focusing flowers in images, which

Figure 1.4: Samples of flowers from different species with the same color .

Figure 1.5: Barberton daisy flower.

means that segmentation cannot be used to isolate flowers from background in such applications. Flower detection is an alternative to isolate flowers from background, where detection is done regardless of flowers scale and viewpoint or even if the images are not focused.

Our main contribution:

1. Build a system to isolate flowering plants from background in natural images using detection approach, regardless of its size, shape, color and viewpoint.

2. Study different feature extraction approaches and their effect on detection an recognition

3. Use ROC analysis curve to depict the performance of our proposed approach, because ROC curves provide more reliable and thorough information about the performance of the proposed approach than other performance measures.

4. We are interesting in studying the proposed approach performance on flowering plant recognition.

1.2 Formal definition

Given an image I that includes the desired flowering plant x and background. Detecting the flower x in an image can be formally defined as estimating $P(x = i|f(I))$, where $f(I)$ is a feature vector that is extracted from the image and $i \in 0, 1$ represents if the object is flower or not.

Recognizing the flower x in image I can be formally defined as estimating $P(x = j|f(I))$, where $j \in 1, \ldots, M$ is the set of possible flowering plant classes.

1.3 Thesis Organization

The rest of the thesis is organized as follows. Chapter 2 is a theoretical background that describes all the terms, topics, and methods that are needed to understand this thesis. Chapter 3 provides literature review about the previous works related to the plant detection and recognition. Chapter 4 explains the methodology of our work, which includes all the operations, sequences, diagrams that are needed to implement the detection and recognition system. Chapter 5 presents all the experiments that are conducted and the results that are obtained from each experiment. Finally, conclusion and the future work will be given in Chapter 6.

Chapter 2

Theoretical Background

In this chapter, we will provide a brief theoretical explanation of all approaches, terms, topics, and methods that are needed to understand this thesis. We provide explanation about classification, feature extraction and detection approaches.

2.1 Classification

The problem of identifying to which class a new data belongs to, among a set of classes, is known as classification, which is one of the machine learning fields that is considered as a supervised learning technique [38]. In supervised learning, a set of input data and their desired output are provided to the machine. After a period of training, the machine is theoretically capable of generalizing from the provided set of data to novel sets of data.

Theoretically, when training a classifier to represent a binary classification problem, we have N training data points as input, each belongs to one of two possible classes $y_i = -1$ or $+1$. The goal is to decide to which class a new data point belongs to. To do so, each data point x_i is represented by D attributes (features), so that the training data will be in the form [6]:

$\{x_i, y_i\}$ where $x \in R^D, y_i \in \{-1, +1\}, i = 1...N$

One of the classification techniques that shows its efficiency in classification problems is support vector machine.

2.1.1 Support Vector Machine

A support vector machine (SVM) is a new generation learning system in computer science that is based on statistical learning [30]. SVM is among the best of supervised learning algorithms, which are used to analyse data and recognize patterns within data [6]. In addition, SVM can also be used as classification and regression analysis tool [16, 3]. In classification problem, standard SVM generates input-output mapping relations from a set of labelled training data [6, 39], and predicts which of the two possible classes represents the input data.

The input data may be linearly or non-linearly separable. When data is linearly separable, we can draw a line that can separate the two classes in case where $D = 2$, and a hyperplane in $D \geq 2$ case. This is called a linear SVM classifier. In fact, there will be more than one hyperplane that might linearly separate the data. The best is the one with the largest separation, or margin between the two classes. Where the distance between the hyperplane and the nearest data from each class is maximized. Such hyperplane is called maximum margin hyperplane (see Figure 2.1).

The hyperplane can be written as: $w \cdot x + b = 0$, where w is the normal vector to the hyperplane, and $\frac{w}{||v||}$ is the perpendicular distance from the hyperplane to the origin along w. The term $||\cdot||$ denotes the Euclidean norm.

The support vectors (shown in circles in Figure 2.1) are the examples that are closest to the hyperplane in both classes. The goal of SVM is to maximize the distance between the support vectors by orienting the hyperplane so it

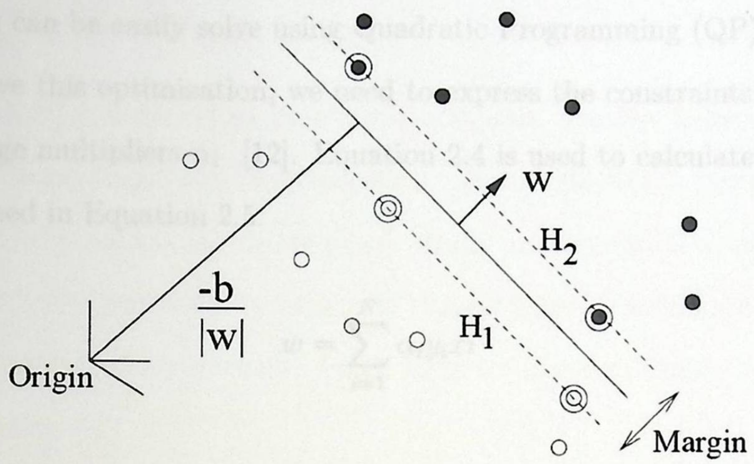


Figure 2.1: Hyperplane through two linearly separable classes [6]

will be as apart as possible from the support vectors in both classes while maintain the separation data. We want to choose w and b that maximize the margin between the parallel hyperplanes that support vectors lie on to avoid overfitting.

For linearly separable data, we can describe the training data by the following equations [12]:

$$w \cdot x_i + b \geq +1, y_i = +1 \tag{2.1}$$

$$w \cdot x_i + b \leq -1, y_i = -1 \tag{2.2}$$

Both equations can be rewritten as in Equation 2.3:

$$y_i(w \cdot x_i + b) - 1 \geq 0, \forall x_i \tag{2.3}$$

SVM margin is equal to $\frac{1}{||w||}$. Maximizing this margin subject to the stated constraints in Equation 2.3. This optimization is difficult because it depends on $||w||$, which involves a square root. Thus, we can use the equivalent term of this optimization, $\frac{1}{2}\,||w||$ subject to the constraints in 2.3.

This term can be easily solve using Quadratic Programming (QP).

To solve this optimization, we need to express the constraints by means of Lagrange multipliers α_i [12]. Equation 2.4 is used to calculate w. Offset b are defined in Equation 2.5.

$$w = \sum_{i=1}^{N} \alpha_i y_i x i \tag{2.4}$$

$$b = w \cdot x_i - y_i \tag{2.5}$$

To get more robust value for b, it is best to average all support vector. See Equation 2.6, where N_{sv} is the number of support vectors:

$$b = \frac{1}{N_{sv}} \sum_{i=1}^{N_{sv}} w \cdot x_i - y_i \tag{2.6}$$

In some cases when the data cannot be lineally separable, some functions can be used to make input data more separable, these functions are called nonlinear kernel functions that are often used to transform input data into a high-dimensional feature space [6].

2.2 Feature Extraction

In any classification problem, input data cannot be used directly, rather features must be extracted from it. Features are used to represent the attributes and characteristics of input data. The process of transforming input data into a set of features is called feature extraction. For reliable applications, features must be invariant to scale, rotation and translation. Following are a brief explanation for some interested features.

2.2.1 Wavelet Transform

Images can be considered as time-domain signals, usually, these signals are plotted as time-amplitude representation. From this signal representation, no further information can be obtained. Many mathematical transformation techniques have been used to transform images from time-domain into other domains like frequency domain, which includes distinguishable information about signals.

In Wavelet Transform, signals are treated using an approach called the multi-resolution analysis (MRA), where signals are analysed at different frequencies with different resolutions. The MRA approach is useful in image analysis, where images are expressed at different scales [36] and thus we have a scale-invariant interpretation of the image. In which scale changes and variations should not affect our interpretation.

The Continuous Wavelet Transform (CWT) is one type of the wavelet transform. The signal in CWT is divided into small stationary signals using a window function, where the width of the window is changed as the transformation is computed for every spectral components. CWT is defined as follows:

$$\psi_{a,b} = \frac{1}{\sqrt{a}}\psi\left(\frac{t-b}{a}\right) \tag{2.7}$$

From Equation 2.7, the transformation is a function that depends in two variable, a and b, the translation and scale of the window function, respectively. Scale is used either to dilate or compress the signal. ψ is the transformation funct ion, which is known as the wavelet mother function [1]. For more information see Appendix C

The CWT should be done at every position and every scale, which is considered a costly process. Fortunately, images are stored discretely in computers (band-limited), thus we do not need to compute the continuous version of wavelet transform. So we have to use the Discrete Wavelet Transform (DWT) in image analyses, in which the transformation can be scaled and translated in discrete steps. DWT depends on two types of functions, scaling functions and wavelet functions [1]. These functions are associated with low pass and high pass filters, respectively. The discrete wavelet function is expressed as follows:

$$\psi_{m,n} = \frac{1}{\sqrt{a_0}^m}\psi\left(\frac{t - nb_0{a_0}^m}{{a_0}^m}\right) \tag{2.8}$$

The variables n and m are used to control the expansion and translation of the wavelet. While variables a_0 and b_0 are usually set to 2 and 1, respectively.

Haar Wavelet

The Haar wavelet is the simplest possible wavelet. Its scaling function can be described as:

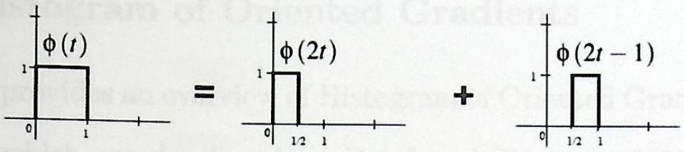
$$\phi(t) = \phi(2t) + \phi(2t-1) \tag{2.9}$$

where

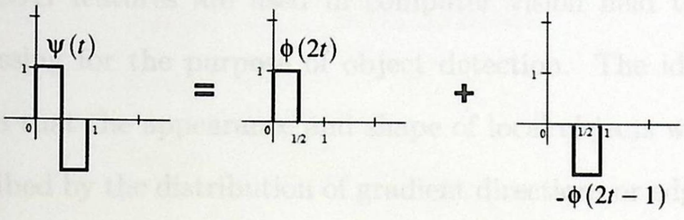
$$\phi(t) = \begin{cases} 1 & 0 \le t < 1 \\ 0 & otherwise \end{cases} \tag{2.10}$$

The Haar wavelet mother function can be defined as:

$$\psi(t) = \phi(2t) + \phi(2t-1) \tag{2.11}$$



(a) Haar scaling function



(b) Haar wavelet function

Figure 2.2: Haar wavelet function and Haar scaling function [36]

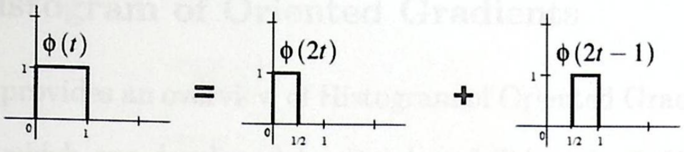
where

$$\psi(t) = \begin{cases} 1 & 0 \leq t < \frac{1}{2} \\ -1 & \frac{1}{2} \leq t < 1 \\ 0 & otherwise \end{cases} \tag{2.12}$$

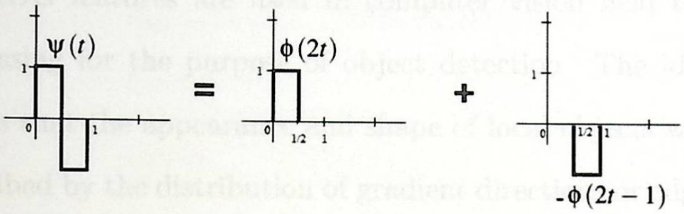
The Haar wavelet function and its associated scaling function are shown in Figure 2.2.

Daubechies Wavelet

The daubechies wavelet transforms are similar to Haar wavelets in the way of calculating the running averages and differences via production of scalar with scaling and wavelet function. The differences between them is in how the scaling and wavelets functions are computed [42]. In Daubechies the frequency responses are balanced, but phase responses are non-linear. Unlike Haar wavelets, Daubechies wavelets use overlapping windows to reflect all high frequency changes through the high frequency coefficient spectrum [27].



(a) Haar scaling function



(b) Haar wavelet function

Figure 2.2: Haar wavelet function and Haar scaling function [36]

where

$$\psi(t) = \begin{cases} 1 & 0 \leq t < \frac{1}{2} \\ -1 & \frac{1}{2} \leq t < 1 \\ 0 & otherwise \end{cases} \tag{2.12}$$

The Haar wavelet function and its associated scaling function are shown in Figure 2.2.

Daubechies Wavelet

The daubechies wavelet transforms are similar to Haar wavelets in the way of calculating the running averages and differences via production of scalar with scaling and wavelet function. The differences between them is in how the scaling and wavelets functions are computed [42]. In Daubechies the frequency responses are balanced, but phase responses are non-linear. Unlike Haar wavelets, Daubechies wavelets use overlapping windows to reflect all high frequency changes through the high frequency coefficient spectrum [27].

2.2.2 Histogram of Oriented Gradients

This section provides an overview of Histogram of Oriented Gradients (HOG) descriptors, which are developed by Dalal and Triggs in 2005 (see Figure 2.3) [31]. HOG features are used in computer vision field together with image processing for the purpose of object detection. The idea behind l2 descriptors is that the appearance and shape of local objects with an image can be described by the distribution of gradient directions or edge directions. This is done depends by evaluating histograms of image gradient directions or edge directions in a localized portions of an image. The local histograms are well-normalized and they are built on a dense grid.

Figure 2.3 shows how HOG features are extracted, where the input image is divided into small spatially connected regions called ("cells"). An accumulated local 1-D histogram of gradient directions or edge orientations is calculated for each cell. Then, a larger spatially connected regions called ("blocks") is used to normalize the contrast of the local histograms. Each block consists of a set of adjacent cells, with a possible overlaps between block. The normalization process is done by measuring the "energy" or intensity of the histograms for cells within one block and using the result to normalize each cell's histogram within that block. The normalization process is done to provide better performance in the detection process; as it becomes more invariant to illumination and shadowing changes. Finally, the combination of these normalized histograms represent the descriptor, which are called Histogram of Oriented Gradient (HOG) descriptors.

HOG features have 4 steps that are provided below:

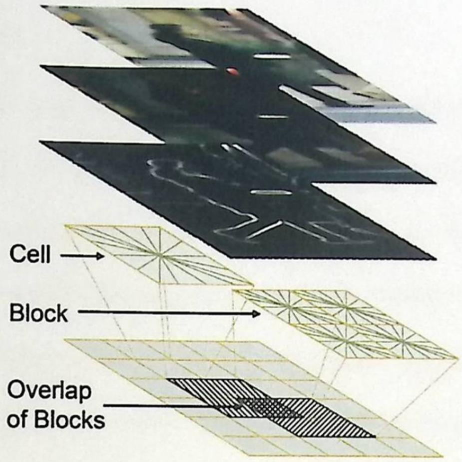


Figure 2.3: Histogram of Oriented Gradient[31]

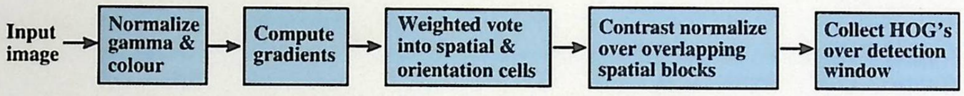


Figure 2.4: HOG feature extraction chain [31]

Step 1:Gamma/Colour Normalization:

In this step, images can be represented using different colour spaces, this include RGB, gray-scale, and LAB colour spaces [15]. From [5] experiments, it is better to use colour information when available when calculating HOG features. So using RGB or LAB colour spaces provide better performance than gray-scale.

Step 2:Gradient Computation

Several methods exist for the computation of the gradient values. The most common and the simplest one is to apply Gaussian smoothing followed by a discrete derivative mask, which can be applied on one or both of the vertical and horizontal directions of an image. The mask can be done by simply filter the color or intensity data of the image using one of the following 1-D filter kernels:

$$\begin{bmatrix} -1 & 0 & 1 \end{bmatrix} \text{ and } \begin{bmatrix} -1 & 0 & 1 \end{bmatrix}^T$$

Other filter kernels were tested by Dalal and Triggs, this include 3x3 sobel kernels as well as centred, uncentred and diagonal kernels. From their experiments these kernels achieved poorer performance in human detection. For Gaussian smoothing, omitting any smoothing exhibited better performance. In case where color images were used, a separate gradients were calculated for each channel, and the one with largest norm with be used as a gradient vector.

Step 3: Spatial / Orientation Binning

The next step is to create the local histograms. Each cell accumulating a weighted vote for a local 1-D histogram of gradient or edge directions over the pixels of the cell. This local histogram is based on the values found in the gradient computation. The shape of the cells can either be rectangular or radial, and the orientation bins or the histogram bins can either be evenly spread over $0°$ to $180°$ or $0°$ to $360°$, this depends on whether to have "unsigned" gradient or "signed" one. In Dalal and Triggs experiments, they found that unsigned gradients in conjunction with 9 histogram bins exhibited better performance in their human detection experiments. For the vote weight, each pixel can contribute either by the gradient magnitude itself, or some function of the magnitude, such as the square root of the gradient magnitude or its square, or some clipped version of the magnitude, which is used as a representation the presence or absence of an edge at the pixel. Best results were obtained by using the gradient magnitude itself.

Step 4: Normalization and Descriptor Blocks

In this step, the problem of illumination and contrast changes is considered, where the local variations in illumination and contrast cause gradient strength to be vary over a wide range. Thus, the gradient strength must be locally normalized, where the normalization process is based on grouping cells together into larger spatial regions called blocks, then normalize each block separately. The vector of the normalized cell histograms from all of the blocks represents the HOG descriptor. A typical overlap exists between block, which means that each cell contributes several times to the final descriptor vector. Block geometry may be square or rectangular R-HOG blocks or circular C-HOG blocks (log-polar). In R-HOG blocks, images are partitioned into square or rectangular grids, which are described by the number of cells in each block, number of pixels in each cell and number of bins per cell histogram. In C-HOG blocks, the blocks can be described by the number of radial and angular bins, the center bin radius, and the expansion factor for the radius of additional radial bins [31].

Dalal and Triggs evaluate four different schemes for the normalization process. The schemes are in the following:

- L2-norm based normalization:

$$f = \frac{v}{\sqrt{||v||_2^2 + \epsilon^2}} \tag{2.13}$$

- L2-hys based normalization: L2-norm followed by limiting the maximum values of v to 0.2.

- L1-norm based normalization:

$$f = \frac{v}{(||v||_1 + \epsilon)} \tag{2.14}$$

- L1-sqrt based normalization:

$$f = \sqrt{\frac{v}{(||v||_1 + \epsilon)}} \tag{2.15}$$

Where v is the unnormalized vector, $||v||_k$ is it k-norm for $k = 1, 2$ and ϵ is a small constant.

Using HOG features have shown very good results for person detection in images [31], even in occlusion cases, images with complex background, and images with different scales. We will test HOG features efficiency in plant detection, also experiment different type of image processing feature extractors.

2.2.3 Gabor Filter

Gabor Filter (GF) is considered as band-pass filters that is used to represent uni-dimensional signals such as speech signals, it also can be used to represent 2-D signals such as images, in this case GF is called Spatial 2-D Gabor Filter. In images, GF is used for edge detection and texture representation and discrimination, where images are represented in frequency and orientation representation.

GF is computed by implementing one or multiple convolutions of an input image with Gabor function, which is defined as the product of a Gaussian kernel times a complex sinusoidal plane wave. The complex sinusoid is called the **carrier**, and the 2-D Gaussian function is called **envelope**. GF is defined in Equation 2.16 [29]:

$$g(x,y) = \exp\left(-\frac{x'^2 + \gamma^2 y'^2}{2\sigma^2}\right) \exp\left(i\left(2\pi\frac{x'}{\lambda} + \varphi\right)\right) \tag{2.16}$$

GF function has real and imaginary components, the real component is defined as [29]:

$$g(x,y) = \exp\left(-\frac{x'^2 + \gamma^2 y'^2}{2\sigma^2}\right) \cos\left(2\pi\frac{x'}{\lambda} + \varphi\right) \tag{2.17}$$

The imaginary component is [29]:

$$g(x,y) = \exp\left(-\frac{x'^2 + \gamma^2 y'^2}{2\sigma^2}\right) \sin\left(2\pi\frac{x'}{\lambda} + \varphi\right) \tag{2.18}$$

where λ is the wavelength of the sinusoid, θ is the orientation, φ is the phase offset, σ represents the size of the Gaussian envelope and γ represents the spatial aspect ratio.

Parameters details are explained in the following:

Wavelength (λ)

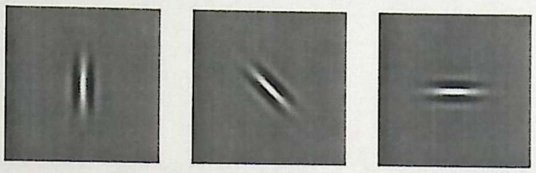
Here λ is used to specify the wavelength of the cosine factor, which is specified in pixels. Valid values of λ are real numbers equal to or greater than 2. Figure 2.5 shows the effect of λ on the filter.

$\lambda = 5$ $\lambda = 10$ $\lambda = 15$

Figure 2.5: The effect of wavelength parameter on GF kernel [29]

Orientation (θ)

The value of θ is used to specify the orientation of the normal to the Gabor function's parallel stripes. Its values are specified in degrees and must be within interval $[0, 360]$. GF kernels with different orientations are show in Figure 2.6.

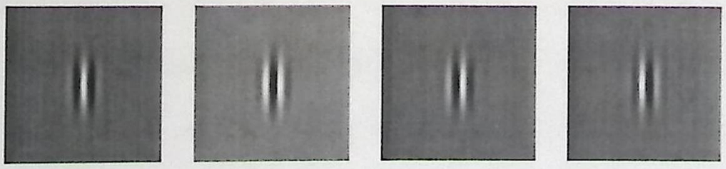


$\theta = 0$ $\theta = 45$ $\theta = 90$

Figure 2.6: The effect of orientation parameter on GF kernel [29]

Phase offset(s) (φ)

The phase offset parameter is used to represent the argument of the cosine factor, which is specified in degrees. φ must be within interval $[-180, 180]$. The function can be center-symmetric, anti- symmetric or asymmetric function, where values 0 and 180 is used to form center-symmetric functions. The values 90 and -90 form anti-symmetric function. All other values form asymmetric functions. See Figure 2.7.

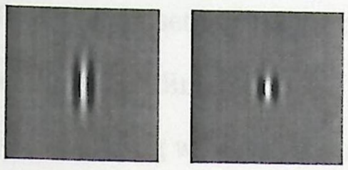


$\varphi = 0$ $\varphi = 180$ $\varphi = -90$ $\varphi = 9$

Figure 2.7: The effect of phase offset parameter on GF kernel [29]

Aspect ratio (γ)

The aspect ratio is used to specify how the support of Gabor function is elliptic. When $\gamma = 1$, the support is circular. Figure 2.8 shows GF kernel with aspect ratio value of 0.5 and 1, respectively.



$\gamma = 0.5$ $\gamma = 1$

Figure 2.8: The effect of aspect ratio parameter on GF kernel [29]

2.3 Detection Approach (Poselets)

One of the human detection approaches that are considered as a classification problem is Poselets, which are notions that have been developed by Bourdev and Malik for detecting humans in images. The term poselet is used to "describe a particular part of human pose under a given viewpoint" [5]. This includes frontal and profile faces, pedestrians, head and shoulder views . . . etc.

To construct a poselet, a set of image patches that are tightly clustered both in image appearance space as well as configuration space are used as positive examples to train one SVM. The configuration space is determined by annotating each image with a set of keypoints to facilitate the process of finding tightly clustered patches that have similar configuration of keypoints. The system have 19 keypoints annotations, which include joints, eyes, ears . . . etc. The location of each keypoint is expressed as Gaussian distribution, which is calculated from the location of one keypoint in all images that are used to extract patches form them to train one SVM. The Gaussian distribution of keypoints will be used for detection purpose [24]. To ensure scale invariant detection; different sizes of the training patches are used.

HOG descriptors are computed for each image patch. The HOG features of tightly clustered patches are used as positive examples to train a linear

SVM classifier, which will be used to detect the trained part of human. The system includes sets of SVMs, each set include a number of SVMs that are trained with patches have the same dimensions.

At test phase, a multi-scale sliding window of same dimensions as training patches is used to find strong activations for different SVMs in the test image. The size of the sliding window will vary according to the tested set of SVMs. The location of the strongly activated patches and the Gaussian distribution of the keypoints are used to predict the location of a human in the test image. The process of predicting humans in images is started by predicting the keypoints for each strong activated patch. The prediction of keypoints proceeds by the following steps:

1. Projecting the activated SVM keypoints on the test image.
2. Scaling the projected keypoints using the minimum dimension of the activated patch in the test image.
3. Translating the projected keypoints using the center of the activated patch in the test image.

The scale and translation for each keypoint k in the activated patch i is done using Equation 2.19 and 2.20 [24, 5], where mu_k and sig_k are the mean and the standard deviation of the $k-th$ keypoint respectively. While sc_i and tr_i are the scale and translation of the $i-th$ activated patch respectively.

$$mu_k = mu_k \times sc_i + tr_i \tag{2.19}$$

$$sig_k = sig_k \times {sc_i}^2 \tag{2.20}$$

Each strongly activated linear SVM will vote for a possible location of a human, which means more than one SVM will predict the location of the keypoints for each human in the image. To determine which keypoints belong to one hypothetically human, all the predicted keypoints are clustered. Keypoints within one cluster are used to predict the location of one hypothetically human [5].

It will be interesting to investigate the poselets approach to flower detection.

Chapter 3

Literature Review

Most recognition applications that are related to plants require them to be isolated from the background [22, 4]. In addition, they are highly depend on segmentation [22, 40, 4, 7, 32], which requires the images to be focused on the plants. These methods are out of our consideration. We are interested on other cases where the plants are part of a whole image. In this cases, more complicated approaches are used for isolating plants from background. We are interesting at detecting plants in images rather than segmenting them.

Plant recognition is either based on features produced from plant leaves [22, 4] or features extracted to represent the general appearance of plants [10, 40, 7, 21, 19, 20]. The related works of plant detection and recognition is briefly expressed in the following sections, where detection is used to find the location of the flower in the image, but segmentation is used to isolate flowers from background.

3.1 Plant detection

Plants are considered as a desired objects in some general object detection systems, some of these systems is the one that is built by Felzenszwalb et al.[11]. They built a general object detection model, and plants are one of the objects that are considered in their work. The model is based on the use of pictorial structures framework. The pictorial structures are defined according to [11] as "represent objects by a collection of parts arranged in a deformable configurationt". Each part represents a local appearance of the object, while the deformable configuration is used to provide connection information between parts. They have a feature map with a feature vectors at each entry in the map. HOG features are extracted from patches located in a dense grid images. A linear rectangular filter is used to filter each patch. They also built a feature pyramid for each image to evaluate it at different positions and scales. In the deformable part models, a root filter is used to define the entire object. While smaller parts are represented by other high resolution filters. In detection stage, the high scored root filters are used to define detection type, while part filters are used to define a full object location (hypotheses).

The problem of the previous system is the time. It takes a relatively long time to detect objects. Felzenszwalb et al. in their works in [10] accelerated the time that is required to detect the proposed objects in the images. Their system depends on building a cascade classifier for the deformable part models, which is a star-structured model. A partial hypothesis pruning is used to prune hypotheses with low scores. The pruning is done using their introduced probably approximately admissible thresholds. Their main contribution is to speed up detection process, while guarantees the detection performance.

3.2 Plant recognition

One of the systems that are based on leaves features is the one that is introduced by Zhan et al. [22]. They developed a system for identifying plant species based on GF and statistical moments features that are extracted from the leaves of plants. The geometry features for the leaves are also considered, such as the ratio of the leaf length to width and 2-D moment invariants. GF and statistical moments are used as texture features, where gabor are used in multi-resolutions. Next a self organizing map neural network is trained with the feature vectors. Query leaf samples will be identified using the trained classifier.

Belhumeur et al. [4] also depend on leaves features to identify plant species. Firstly, the system must starts with segmentation phase to isolate leaves from background. Images then are mapped into hue saturation value (HSV) color space. The system depend on the shape of leaves to identify their species. They use inner distance shape context (IDSC) as leaf features that are used to match the shape of a query leaf with features of leaves in the dataset. The IDSC is represented as 2-D histogram descriptors, which is constructed for samples of points over the boundary of leaf shapes. Each entry in the histogram represents the angle and the distance between the reference point and all other points along the boundary. To identify the query leaf, IDSC is constructed for it. Then it is compared with all other IDSCs by comparing each sample point in the query leaf by all points in all other IDSCs. The final decision is determined by calculating a distance matrix between the query leaf shape and other shapes.

Using leaf features have a problem of having to pre-process the images to segment leaves from background. As a result, researchers tend to use texture features that are extracted from the general appearance of plants rather than leaves features in identifying plants. Some of plant recognition systems that depend on the general appearance texture features are briefly explained in the following.

The flower recognition system that is introduced by Nilsback et al. [33] depends on the use of a bag-of-words (BoW) approach for the proposed features. Namely, internal dense scale-invariant feature transform (SIFT) [25], boundary SIFT and histograms of gradients. These features are extracted from images in HSV color space. Then a non-linear support vector machine is trained by the feature vectors. Later, they improved their system by adding another geometric layout features.

One use of support-vector-machine for flower recognition can be found in the publication by Chai et al. [7]. In their work, flowers are firstly segmented from the background using their developed superpixel-based flower foreground segmentation algorithm. Then, flower species are recognized using a combination of descriptors, which are extracted from the segmented images in Lab color space. The features are multi-scale dense SIFT, interest point SIFT and multi-scale boundary SIFT. The multi-scale dense SIFT descriptors are used for shape representation and they are the most powerful descriptors from the used features. All of the used features are extracted in multiple points in each, which result in more than one feature vector for each image. They used the BoW approach in order to transform the feature vectors into one vector. A linear support vector machine is trained to classify one flower species from all other species. They tested their system on the Oxford 17 and 102 dataset [1]. In the comparison between their performance and recent publications on Oxford Flower 102 dataset, their system outperform the others [21, 19, 20].

[1] http://www.robots.ox.ac.uk/ vgg/data/flowers/

In [40], Siraj et al. developed a system for classifying Malaysian blooming flowers. Firstly, images are segmented using thresholding and region filling techniques. Images are then transformed into HSV color space to extract texture features from them. The extracted features are based on the use of a gray-level co-occurrence matrix. From the matrix, four features are calculated and used as features to represent each image, which are contrast, correlation, energy and homogeneity. Finally, these features are used to train neural networks to classify flower species [9].

All of the above reviewed systems are based on using learning techniques and classification tools for plant recognition. Other systems may depend on extracting features for a query image, then use the distance between its features and the features of the images in the database to decide the plant type and recognize it. Next, some systems that depend on the later methodology are briefly explained.

A flower image recognition system that is developed by Hsu et al. [18] starts by providing the user the ability to draw a rectangular window around the flower in the image. Then a sobel edge detector is used for flower edge detection and stamen region estimation. They have used two types of shape features; the whole flower region features and the pistil/stamen area regions. Features that represent the whole flower region are based on the center point, which is used to find the distance between it and every pixel in flower boundary to represent the sharpness of each petal in the flower. Also a normalized distances are averaged to ensure scale invariant characteristic. Finally, the roundness measure is considered to indicate how much the shape of petals is close to the center. The pistil/stamen area features are the mean of the normalized distance values between the centroid of the stamen and every pixel in the stamen area, the standard deviation of the normalized distance values and the third central moment of the normalized distance values. To recognize a query image, the distance between the query image and all images in the dataset is calculated. The image with minimum distance is considered.

Vuarnoz developed a flower recognition system that starts by image segmentation [41]. A watershed algorithm is used to segment flowers from images. The segmented images is then used to construct a binary mask images. The original image and the mask are used for feature extraction. He used three types of features namely; color features, contour-based features and texture features. Color features are color histogram and color moment in different color spaces. The contour-based features that are used to represent the shape of the flowers are the distances and angles between the center of the flower and every pixels in the contour, the ratio of the minimum and maximum distances, number of petals and a histogram of the distances, which is used to represent the distances is distributed. Finally, the texture features are gabor wavelet and gist features. The features are extracted for query image and compared to those in the database.

Some of the reviewed recognition approaches require a pre-processing segmentation phase for flower recognition. While others require a special attention of the input image such as focusing it on the flowering plant. In our proposed approach, we replace the segmentation phase by detection phase that does not need special attention to isolate flowering plant from background. Rather images are in full scene and flowers are part of them.

Chapter 4

Methodology

We considered both the detection and recognition problems as classification problems. Like poselet that is described in Section 2.3, we divide the images into patches and train a set of SVMs with these patches to detect the flowering plant species. But unlike poselets, we tried to use different features. For recognition, we have built new classifiers to recognize the type of the species.

Having a colored image, the system proceeds through 4 steps; preparing images, extracting features, detecting flowers and finally recognizing the detected flowers. Since the detection is considered as classifier using machine learning, we need a learning phase for detection. For this learning phase, a set of classifiers are trained to detect flower species. Then, the trained classifiers are used to detect the flower species in full scene images. A sliding window slides over the image, extract the features form it, then use the trained classifier to determine whether it is a flower or a background.

In the recognition phase another set of classifiers are trained to recognize flower species. The trained classifiers are used to determine the type of the windows that are classified as flowers in the detection phase. Each step in the flower detection and recognition system is described in details later in

Chapter 4

Methodology

We considered both the detection and recognition problems as classification problems. Like poselet that is described in Section 2.3, we divide the images into patches and train a set of SVMs with these patches to detect the flowering plant species. But unlike poselets, we tried to use different features. For recognition, we have built new classifiers to recognize the type of the species.

Having a colored image, the system proceeds through 4 steps; preparing images, extracting features, detecting flowers and finally recognizing the detected flowers. Since the detection is considered as classifier using machine learning, we need a learning phase for detection. For this learning phase, a set of classifiers are trained to detect flower species. Then, the trained classifiers are used to detect the flower species in full scene images. A sliding window slides over the image, extract the features form it, then use the trained classifier to determine whether it is a flower or a background.

In the recognition phase another set of classifiers are trained to recognize flower species. The trained classifiers are used to determine the type of the windows that are classified as flowers in the detection phase. Each step in the flower detection and recognition system is described in details later in

this chapter.

4.1 Dataset

We have built a new dataset contains images from different species that are shown in Figure 4.1. We are constrained to build the dataset for flower detection because no benchmark is available. For flower recognition, Oxford 17 [35] and Oxford 102 [34] datasets are available. Oxford 17 dataset consist of 17 flower categories with 80 images for each category. Oxford 102 dataset contain 102 flowering plant categories. Each category have between 40 and 258 images.

Unfortunately, oxford17 and oxford 102 dataset cannot be used for detection phase, because all images in the dataset contain only focused flowers. Our problem requires images to be in the full scene in order to detect flowers in them.

The dataset that we have built includes 10 kinds of flower species, which are shown in Figure 4.1. Images in the figure represent segmented part of the dataset images. The full dataset images are shown in Appendix A. The dataset includes around 80 full scene images in RGB color space for each species. Images are inclusive, where each species is represented with flowers has possibly different color, growth stage, size or different postures.

4.2 Image Preparation

Image preparation phase is for detection and recognition. In this phase, full scene images in the dataset are used to prepare examples for training phase. The images contain flower(s) and background representation. We need to divide the image into patches, each patch must represent either part(s) of a

this chapter.

4.1 Dataset

We have built a new dataset contains images from different species that are shown in Figure 4.1. We are constrained to build the dataset for flower detection because no benchmark is available. For flower recognition, Oxford 17 [35] and Oxford 102 [34] datasets are available. Oxford 17 dataset consist of 17 flower categories with 80 images for each category. Oxford 102 dataset contain 102 flowering plant categories. Each category have between 40 and 258 images.

Unfortunately, oxford17 and oxford 102 dataset cannot be used for detection phase, because all images in the dataset contain only focused flowers. Our problem requires images to be in the full scene in order to detect flowers in them.

The dataset that we have built includes 10 kinds of flower species, which are shown in Figure 4.1. Images in the figure represent segmented part of the dataset images. The full dataset images are shown in Appendix A. The dataset includes around 80 full scene images in RGB color space for each species. Images are inclusive, where each species is represented with flowers has possibly different color, growth stage, size or different postures.

4.2 Image Preparation

Image preparation phase is for detection and recognition. In this phase, full scene images in the dataset are used to prepare examples for training phase. The images contain flower(s) and background representation. We need to divide the image into patches, each patch must represent either part(s) of a

this chapter.

4.1 Dataset

We have built a new dataset contains images from different species that are shown in Figure 4.1. We are constrained to build the dataset for flower detection because no benchmark is available. For flower recognition, Oxford 17 [35] and Oxford 102 [34] datasets are available. Oxford 17 dataset consist of 17 flower categories with 80 images for each category. Oxford 102 dataset contain 102 flowering plant categories. Each category have between 40 and 258 images.

Unfortunately, oxford17 and oxford 102 dataset cannot be used for detection phase, because all images in the dataset contain only focused flowers. Our problem requires images to be in the full scene in order to detect flowers in them.

The dataset that we have built includes 10 kinds of flower species, which are shown in Figure 4.1. Images in the figure represent segmented part of the dataset images. The full dataset images are shown in Appendix A. The dataset includes around 80 full scene images in RGB color space for each species. Images are inclusive, where each species is represented with flowers has possibly different color, growth stage, size or different postures.

4.2 Image Preparation

Image preparation phase is for detection and recognition. In this phase, full scene images in the dataset are used to prepare examples for training phase. The images contain flower(s) and background representation. We need to divide the image into patches, each patch must represent either part(s) of a

flower or full flower.

To partition images into patches with foreground representation, we must annotate images in the dataset with 2D keypoints. The purpose of annotating images in our work is to:

1. Discriminate flowers from background. Figure 4.2 show how keypoints discriminate flower parts from background. Keypoints are represented by red asterisk. Red rectangles represent patches with flower part(s) representation, while blue rectangles represent patches background representation.

2. Determine which patches with the same configuration space (keypoint

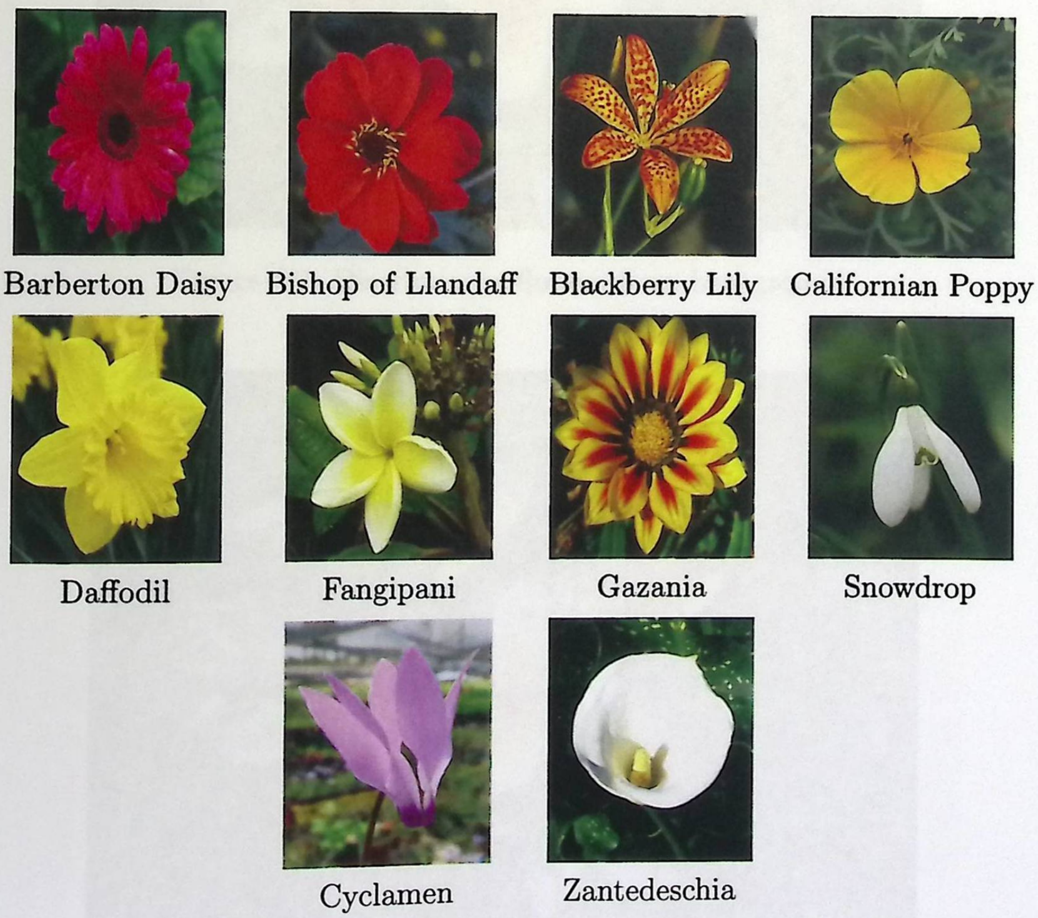


Figure 4.1: Segmented parts from our dataset.

configuration) can be used to train one classifier. In Figure 4.3 rectangles with same color represent patches with the same configuration space.

3. Predict the bounding box for each flower during detection phase. The prediction based on Equation 2.19 and 2.20

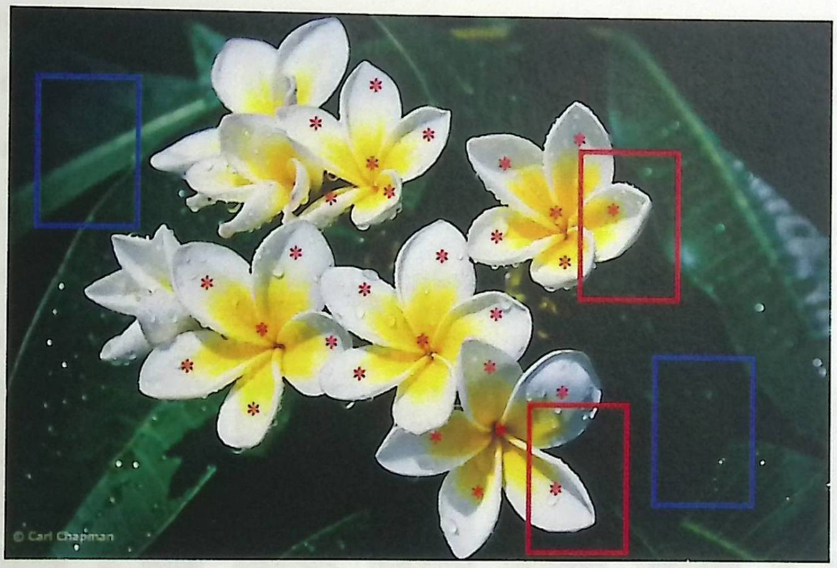


Figure 4.2: Discriminate flowers from background

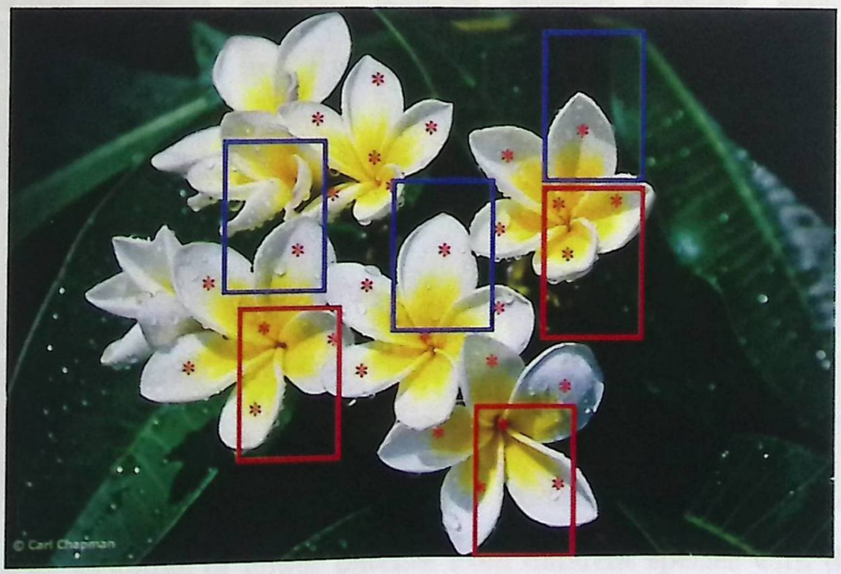


Figure 4.3: Keypoint configuration space

Keypoints are allocated over flower parts in each image. Then, a multi-scale sliding window slides over the full scene images.The sliding window slides each time 8 pixel across the rows and 8 pixels across the columns. The multi-scale sliding window will produces patches with different dimensions to ensure scale invariably detection and recognition task. Some patches with one or more keypoints are considered as foreground patches. While other patches are considered as background patches. Due to the large variation between flower species, the number of keypoints and the configuration of them are varies from one species to another.

Figure 4.4 shows samples of prepared patches, each row in the Figure includes samples of patches belong to the same species that are tightly clustered in both image appearance space and keypoint configuration space.

Figure 4.4: Example of flower patches from different species with different postures.

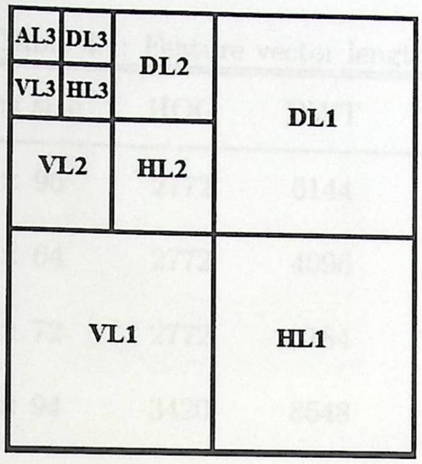


Figure 4.5: Three level 2D DWT.

4.3 Feature Extraction

Feature extraction is an important step in any detection and recognition system; because it is not worthy to use images directly in the problem, rather the use of features in any classification problem make it more effective and efficient. When extracting features from images, we transform them into a feature vector, which will be used to train the classifiers. We have used three popular feature extraction methods, which are DWT, GF and HOG features.

In DWT features, we decompose each input image by three-level 2D wavelet decomposition. Figure 4.5 shows the structure of DWT feature for each image, where HLK, VLK and DLK represent horizontal, vertical and diagonal direction information at level K, respectively. While AL3 represents the approximation at level three. Figure 4.6) shows the representation of an image in DWT space.

Gabor features are extracted using the parameters in Table 5.1. The feature representation of the input image in Figure 4.5 in gabor space is shown in Figure 4.7.

HOG features are extracted from R, G and B channels in a colored image. The channel with maximum orientation histogram is used to calculate the

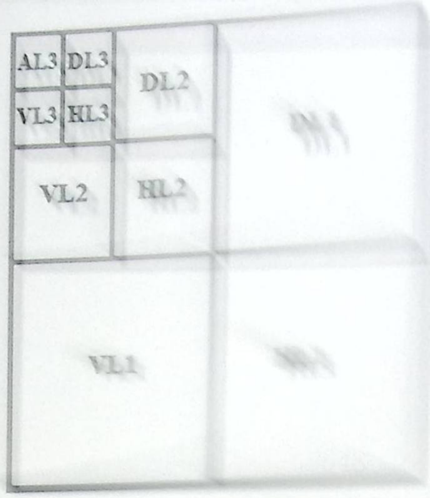


Figure 4.5: Three level 2D-DWT

4.3 Feature Extraction

Feature extraction is an important step in any detection and recognition system; because it is not worthy to use images directly ... the use of features in any ... figure out HOG features as they are in the 5th efficient. When extracting features from images ... feature vector, which ... feature vector varies according to the size of the patch popular feature ... ed features. The length of the feature vectors for each feature ... Table 4.1

In DWT feature ... wavelet decomposition ... each image, where ... agonal direction ... the approximation ... image in DWT ...

Input image DWT image

Figure 4.6: A sample image in DWT space.

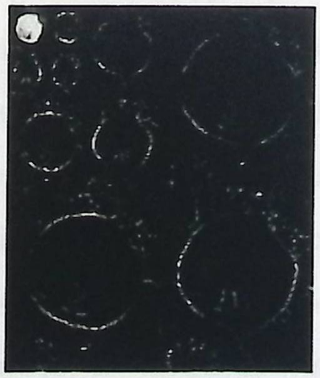
Table 4.1: Feature vector length.

Patch size	HOG	DWT	GF
64 × 96	2772	6144	6815
64 × 64	2772	4096	4639
72 × 72	2772	5184	5791
92 × 94	3420	8648	9427
112 × 118	5877	13216	14138
140 × 136	8604	19040	20213
168 × 176	14328	29568	30975
212 × 218	22635	46216	47914

HOG features. We cannot figure out HOG features as they are in the 5th dimension.

The length of the feature vector varies according to the size of the patch and the extracted features. The length of the feature vectors for each feature are shown in Table 4.1

Input image



DWT image

Figure 4.6: A sample image in DWT space.

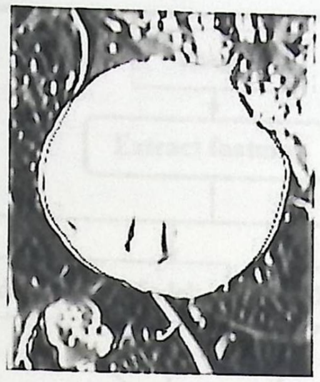


Figure 4.7: Image in gabor space.

4.4 Detection phase

In our proposed approach, we have considered the detection and recognition phases as classification problem. We have used linear SVM as a classifier, because it shows its efficiency in most classification problems [24, 5]. The flower detection and recognition system using SVM classifier is shown in Figure 4.8, where $D - Model_i$ denotes the set of detection models for flower species i. While $R - Model_i$ denotes the sets of recognition models that are used to recognize flower i.

The system starts with image acquisition, then features are extracted from the image after converting it to a set of overlapped patches. A set of detection models that are prepared during training a set of SVMs are used to classify the patches to either flower part(s) or not. The final decision about the location of flowers in the image is made using a classification criteria (1). Features are then extracted from the detected flowers in detection phase to be recognized. Recognition models are used to recognize the type of each detected patch. The classification criteria (2) are then used to make the final decision about the kind of the detected flower.

In this section, the detection phase is split into detection training phase and detection test phase. The following sections provide details about them.

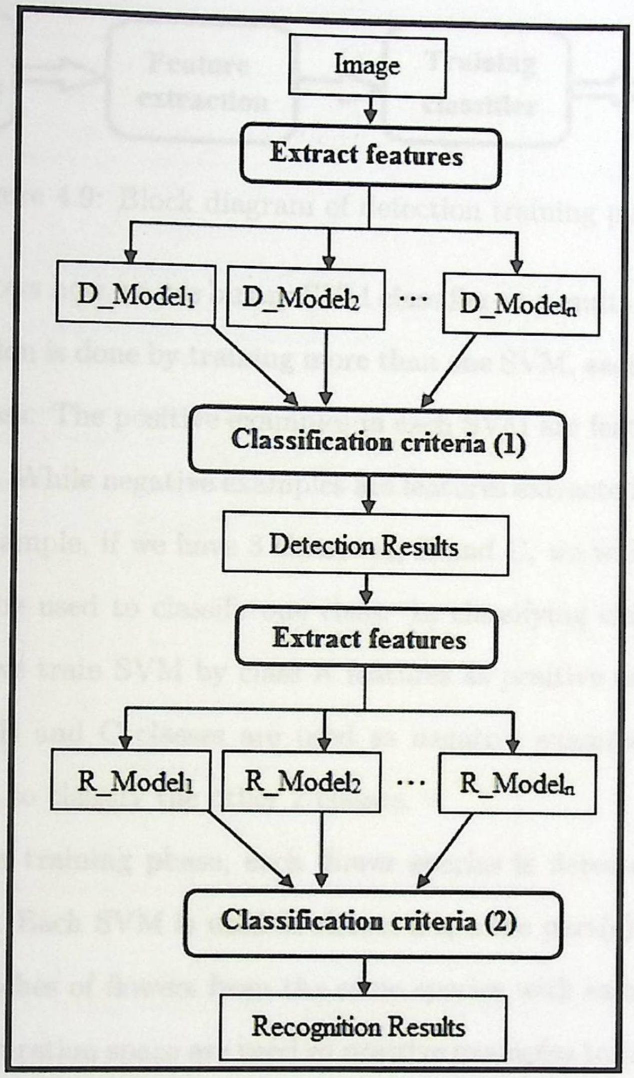


Figure 4.8: Overview of the detection and recognition system.

4.4.1 Detection training phase

The detection training phase is shown in Figure 4.9. Features that are extracted from the prepared patches in Section 4.2 are used to train a set of SVMs. The output of each SVM is a score that is used to determine whether the patch is a flower or not.

As we mentioned in Section 2.1.1, SVM is a binary classifier in nature, and our problem is a multi-class classification problem. Although there exist multi-class SVMs, we use the binary SVM for multi-classes for its simplicity.

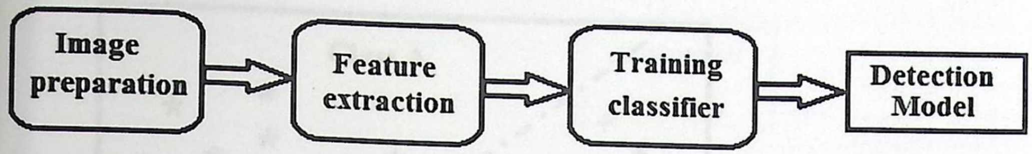


Figure 4.9: Block diagram of detection training phase

Figure 4.10 shows how we use binary SVM classifier as a multi-class classifier. The classification is done by training more than one SVM, each one is used to classify one class. The positive examples in each SVM are features extracted from one class. While negative examples are features extracted from all other classes. For example, if we have 3 classes A, B and C, we will have 3 SVMs each one will be used to classify one class. In classifying class A from the other classes, we train SVM by class A features as positive examples, while features from B and C classes are used as negative examples. The same process is used to classify the other 2 classes.

In detection training phase, each flower species is detected using more than one SVM. Each SVM is used to detect a specific part(s) of a flower or full flower. Patches of flowers from the same species with same posture and keypoint configuration space are used as positive examples to train one SVM. The negative examples are patches from background. Later, this trained SVM will be used to detect such part(s) or such flower in detection test phase.

4.4.2 Detection test phase

In this phase, the detection trained SVMs are used to detect flower species in full scene images, see Figure 4.11. A multi-scale sliding window slides each time 8 pixels across the rows of the full scene image and 8 pixels across the columns. The scale of the window varies as the patch scales vary. Features are extracted form the patch that is represented by the sliding window, then

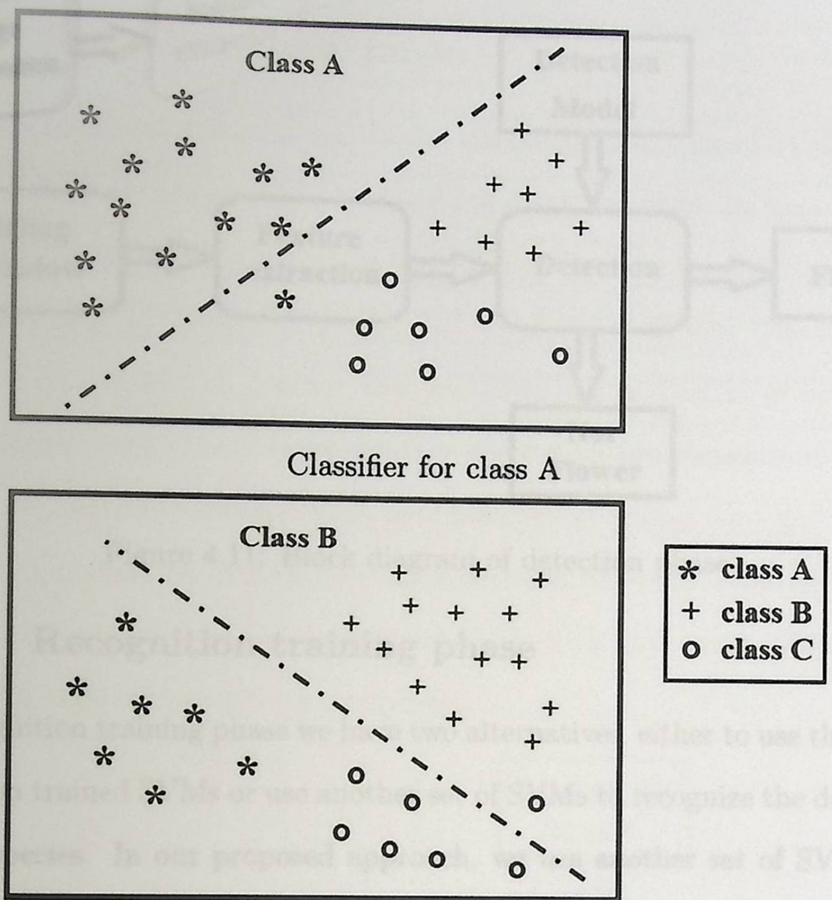


Figure 4.10: Using binary SVM classifier as multi-class classifier.

the features are fed to the detection trained SVMs. The result is a set of patches represent parts of flower(s). Classification criteria (1) in Figure 4.8, are used to determine the final detection decision, which are based on the results obtained from one or more $D - SVM_i$. The poselets implementation that is explained in Section 2.3 is used as implementation for classification criteria (1), which will detect flowers in images.

4.5 Recognition phase

Recognition phase is also treated as classification problem and also has recognition training phase and recognition test phase.

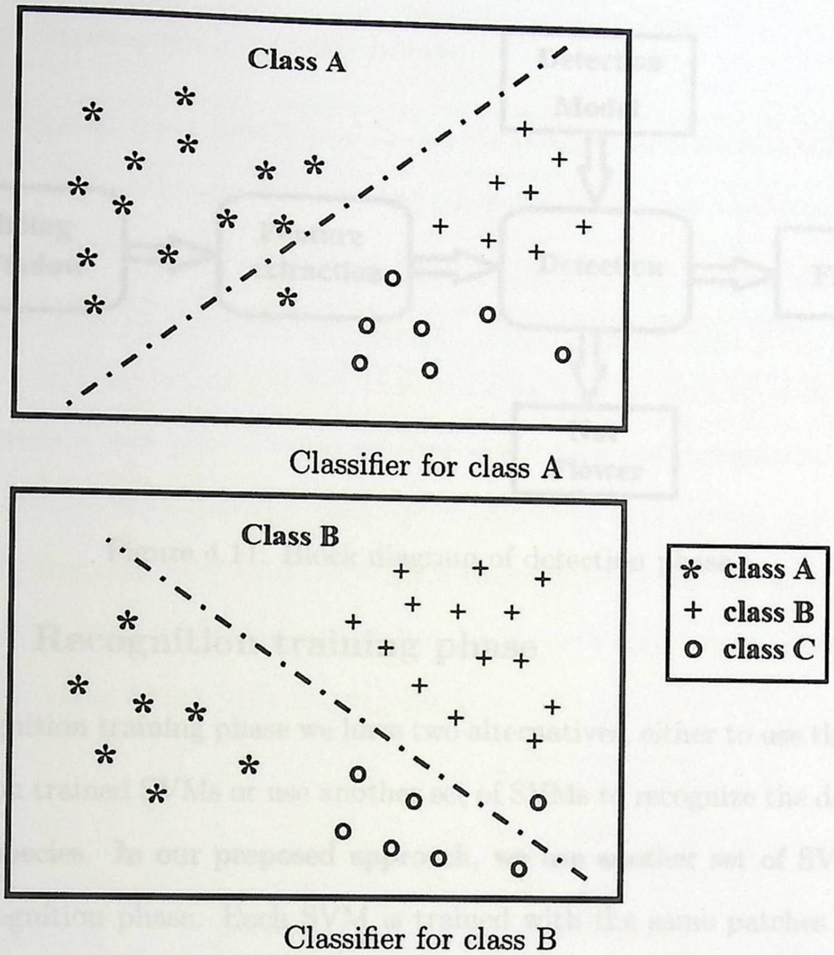


Figure 4.10: Using binary SVM classifier as multi-class classifier.

the features are fed to the detection trained SVMs. The result is a set of patches represent parts of flower(s). Classification criteria (1) in Figure 4.8, are used to determine the final detection decision, which are based on the results obtained from one or more $D - SVM_i$. The poselets implementation that is explained in Section 2.3 is used as implementation for classification criteria (1), which will detect flowers in images.

4.5 Recognition phase

Recognition phase is also treated as classification problem and also has recognition training phase and recognition test phase.

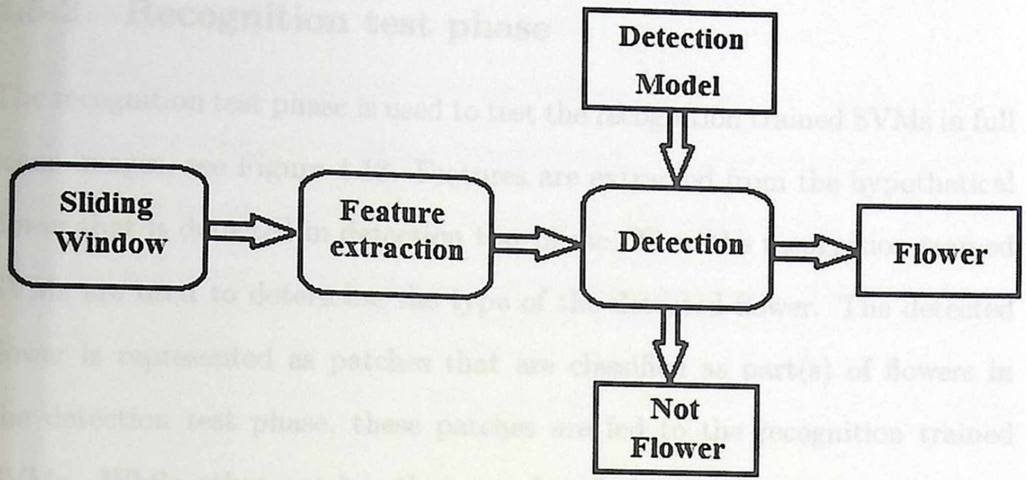


Figure 4.11: Block diagram of detection phase

4.5.1 Recognition training phase

In recognition training phase we have two alternatives, either to use the same detection trained SVMs or use another set of SVMs to recognize the detected flower species. In our proposed approach, we use another set of SVMs for the recognition phase. Each SVM is trained with the same patches that is used as positive examples in the detection phase. But the negative examples are patches from other species. Each SVM is trained to classify part(s) or full flower from one species, see Figure 4.8. The recognition training phase is shown in Figure 4.12. Features are extracted from the prepared patches to train recognition SVMs, which will be used to determine whether the patch belongs to the trained species or not.

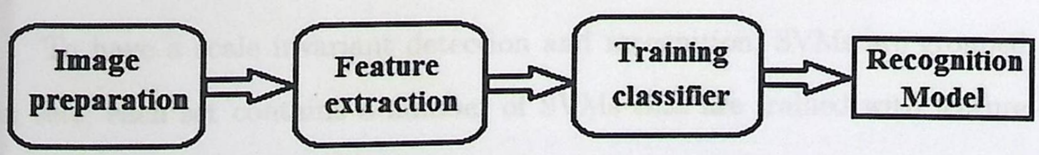


Figure 4.12: Block diagram of recognition training phase

4.5.2 Recognition test phase

The recognition test phase is used to test the recognition trained SVMs in full scene images, see Figure 4.13. Features are extracted from the hypothetical flower that is detected in detection test phase. Then the recognition trained SVMs are used to determine the type of the detected flower. The detected flower is represented as patches that are classified as part(s) of flowers in the detection test phase, these patches are fed to the recognition trained SVMs. While other patches that are classified as background patches are ignored. The final recognition decision is determined by classification criteria (2), which is voting from the recognition SVMs with high scores. The species with high votes from SVMs is recognized.

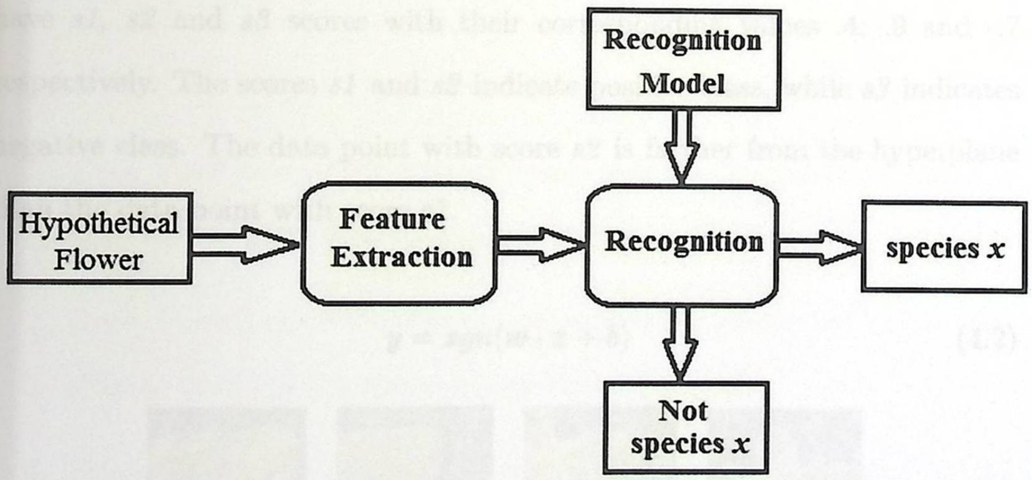


Figure 4.13: Block diagram of recognition phase

To have a scale invariant detection and recognition, SVMs are grouped in sets, each set contains a number of SVMs that are trained with feature vectors of same length, which are extracted from patches of same size. All possible postures of the flower species are taking into account in each set. In both detection and recognition phases, each flower species have more than one set of SVMs to detect and recognize it. Figure 4.14 shows a sample of examples that are used to train some SVMs belong to one set.

In both detection and recognition test phases, we use SVM scores to determine the strength of each activated patch in the test image. The score is calculated using Equation 4.1, where S_i is the score of the $i - th$ SVM, while w_i and b_i are calculated as in Equations 2.4 and 2.5, respectively.

$$S_i = w_i + b_i \tag{4.1}$$

SVM scores are used to determine the class of new data point x using Equation 4.2, where *sgn* is the sign functione. Scores are also used to express how far the new data point is from the hyperplane. Farther patches from hyperplane will get higher scores than closer patches. For Example, if we have *s1*, *s2* and *s3* scores with their corresponding values .4, .9 and -.7 respectively. The scores *s1* and *s2* indicate positive class, while *s3* indicates negative class. The data point with score *s2* is farther from the hyperplane than the data point with score s1.

$$y = sgn(w \cdot x + b) \tag{4.2}$$

SVM 1 SVM 2 SVM 3 SVM 4

Figure 4.14: Samples of training examples for Daffodil SVMs with dimensions 64 × 96.

4.6 Evaluation

In test phases, we have used three folds cross validation technique to evaluate the system. Images are divided into three equally groups, two-third of the images are used for training phase and one-third of the images are used as validation data in the test phase. The cross validation process is then repeated three times, each time the training and testing data are changed, but the percentage of them remains unchanged. The results that are obtained from the three folds are represented in a Receiver Operating Characteristic (ROC) curve.

We have used ROC curves to express the detection and recognition performance, which are used to illustrate the performance of a binary classifier. The outcomes of the binary classifier are scores that are calculated using Equation 4.1, but they are farther grouped into 4 groups which are:

1. **True Positives (TP):** The outcomes that are predicated by the classifier as true and the actual values of them are also true.

2. **False Positives (FP):** The outcomes that are predicated by the classifier as true and the actual values of them are false.

3. **True Negatives (TN):** The outcomes that are predicated by the classifier as false and the actual values of them are also false.

4. **False Negatives (FN):** The outcomes that are predicated by the classifier as false and the actual values of them are true.

All of the above binary classifier labels are used to calculate the following measurements that are used to plot the ROC curves:

1. **True Positive Rate (TPR):** Also called *sensitivity*, is the ratio of TP to the total number of positives, see Equation 4.3

2. **False Positive Rate (FPR):** The ratio of FP to the total number negatives, see Equation 4.4

$$TPR = \frac{TP}{TP + FN} \tag{4.3}$$

$$FPR = \frac{FP}{FP + TN} \tag{4.4}$$

The FPR measurement is equivalent to 1- *specificity* (SPC), which is a desired measurement in our work and it is calculated using Equation 4.5. The ROC curves plot the sensitivity vs. (1 - specificity) measurements, see Figure 4.15 [8]. Perfect classification in the ROC curve indicates 100% sensitivity and 100% specificity, which means that all validation data are truly predicted. Curve A represents excellent classifier, curve B is a good classifier, C1 and C2 curves are average classifiers, while curve D represents a random guess classifier with 50% true prediction rate.

$$SPC = \frac{TN}{FP + TN} \tag{4.5}$$

Another measurement is also considered, which are the area under curve (AUC) of the ROC curve. We use AUC to compare between two binary classifiers and measure the quality of them. The curve with the larger area yield a better binary classifier performance [28].

Finally, accuracy measurement is used to evaluate the system performance on full scene images. The accuracy (ACC) is calculated using Equation 4.6

$$ACC = \frac{TP + TN}{TP + TN + FP + FN} \tag{4.6}$$

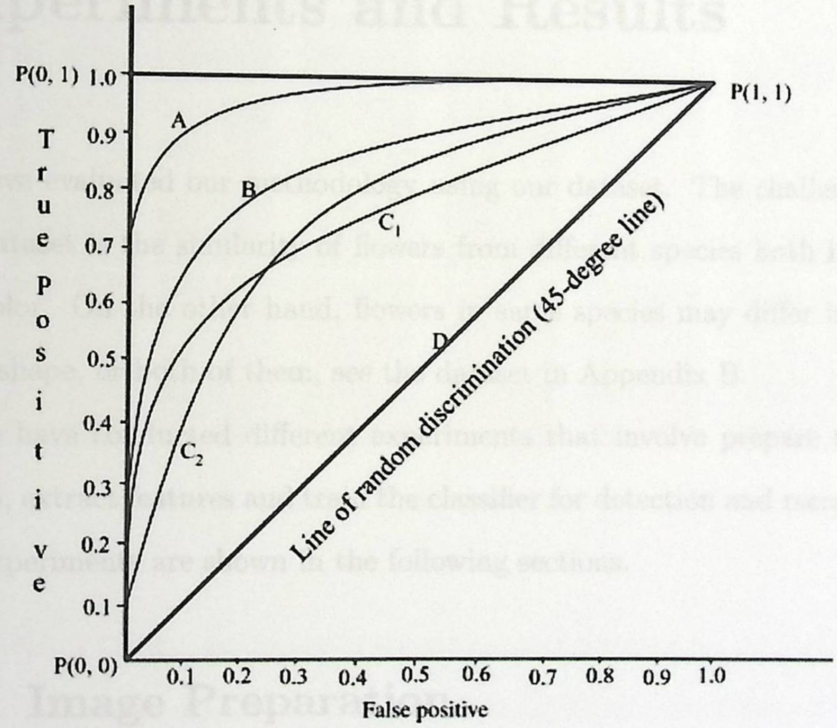


Figure 4.15: The ROC curve space [17]

Chapter 5

Experiments and Results

We have evaluated our methodology using our dataset. The challenging in this dataset is the similarity of flowers from different species both in shape and color. On the other hand, flowers in same species may differ in either color, shape, or both of them, see the dataset in Appendix B.

We have conducted different experiments that involve prepare training images, extract features and train the classifier for detection and recognition. The experiments are shown in the following sections.

5.1 Image Preparation

Images in the dataset are full scene, which means we need to divide them into smaller patches in order to be used for classification. As we have mentioned previously in Section 4.2, each positive example patch includes a part or more of a flower or even a full flower. A set of patches with same flower part representation is used as one cluster. To prepare these patches different approaches are conducted. Approach 1, Approach 2 and Approach 3 are used to prepare patches with size 96×64 and 64×64, where each patch include flower part(s) as in poselets. Approach 4 is used to prepare patches with full flower representation. These images preparation approaches are:

Approach 1 (Manually extracted patches): Firstly, patches are extracted manually from the full scene images. Positive and negative examples are extracted in the same way. As we can see in Figure 5.1, all patches with same flower posture are well extracted, with a slight variation between them.

Figure 5.1: Manually prepared image patches

Approach 2 (Semi-automatically extracted patches): Each full scene image is manually annotated with a number keypoints, see Figure 5.2. Keypoints are used to annotate parts of flowers, so the number of keypoints varies between flower species. An overlapping multi-scale sliding window is used over the full scene image, which slides each time 8 pixel across the rows and 8 pixels across the columns. A patch with keypoint(s) represents a part of a flower or full flower, this patch is saved as a positive example, while a patch without keypoints represents a background that is saved as negative example. In Figure 5.3, each row represents a sample of patches belongs to same cluster and will be used as positive examples to train one classifier.

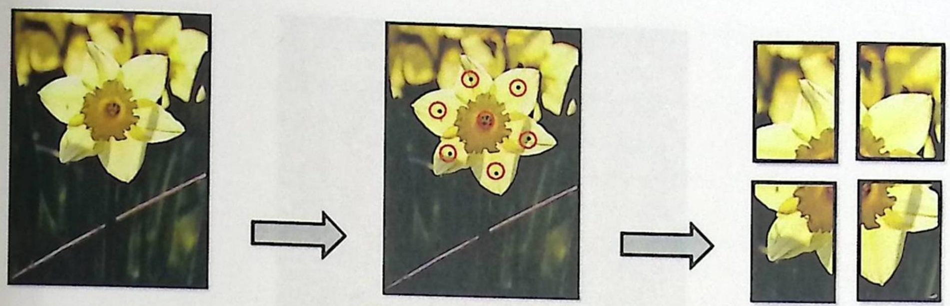


Figure 5.2: Extracting patches semi-automatically.

Figure 5.3: Samples of semi-automatically extracted patches

Approach 3 (9-regions patches): We divide each flower into 9 regions by adding another 9 keypoints on the flower together with the previous keypoints. Figure 5.4 shows the 9 regions, where a region is represented by a keypoint in the center of it, which is marked manually. Also an overlapping multi-scale sliding window slides over the full scene image in the same manner in Approach 2. The resulted patches are then assigned to one of these regions. To do this, the euclidean distance (see Equation 5.1) is calculated between each keypoint in the resulted patch and all keypoints that is used to represent the regions. Equation 5.2 is used to assign patches to one region, where p is the patch keypoint and q_i is the keypoint of the i-th region. In this equation, The region with minimum distance is selected. In this exper-

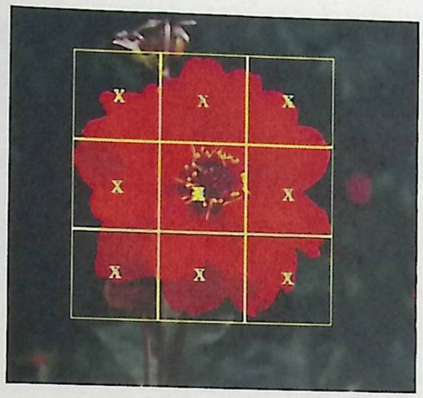


Figure 5.4: Manually dividing a flower into 9 regions

iment, all patches within one region is used to train one SVM to represent that region.

$$|p, q| = \sqrt{(q_1 - p_1)^2 + (q_2 - p_2)^2} \tag{5.1}$$

$$R_i = \min_{d} \left\{ |p, q_i| \right\} \tag{5.2}$$

Approach 4 (Full flower patches): In this experiment, patches include full flower representation. To prepare the patches, we annotate each flower in the full scene image with 2 keypoints. The keypoints represent the top left corner and bottom right corner of the hypothesized bounding rectangle for each flower, see Figure 5.5. The keypoints are used to find the window width and height for each flower. Because flowers vary in size, we need different window sizes to have scale invariant detection and recognition. The size of the windows is determined using K nearest neighbour clustering approach, where windows width and height are clustered into 8 clusters, which are 72×72, 92×94, 112×118, 140×136, 168×176, 216×218 and 296×203.

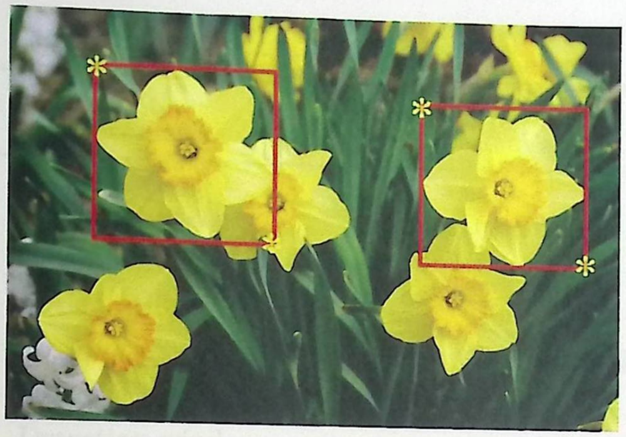


Figure 5.5: Full flower patches

5.2 Feature Extraction

Features are extracted from images to represent them during training and testing phases. In our work, we have three feature extraction methodologies, GF, DWT and HOG features, which are extracted from patches in gray scale color space. A feature vector is extracted from each patch to represent it in the training phase. The associated parameters for each feature extraction methodology are shown in Table 5.1. We use the same parameters as in [31]. In addition, we modified the GF and DWT parameters as shown in the Table 5.1 because they lead to best accuracy based on initial experiments.

5.3 Detection Phase

We have used the linear SVM as a classifier for the detection phase. As what is mention previously in Section 2.1.1, the classifier needs to be trained on positive and negative examples. In detection, each SVM is used to detect a part or more of one species. Thus, a set of SVMs is used to represent and detect one species. In this phase, we have two approaches:

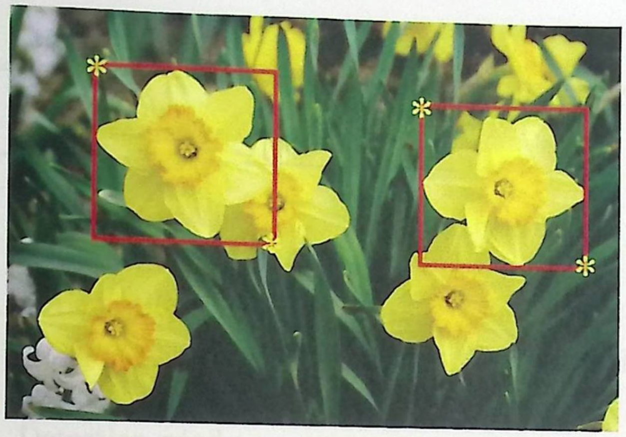


Figure 5.5: Full flower patches

5.2 Feature Extraction

Features are extracted from images to represent them during training and testing phases. In our work, we have three feature extraction methodologies, GF, DWT and HOG features, which are extracted from patches in gray scale color space. A feature vector is extracted from each patch to represent it in the training phase. The associated parameters for each feature extraction methodology are shown in Table 5.1. We use the same parameters as in [31]. In addition, we modified the GF and DWT parameters as shown in the Table 5.1 because they lead to best accuracy based on initial experiments.

5.3 Detection Phase

We have used the linear SVM as a classifier for the detection phase. As what is mention previously in Section 2.1.1, the classifier needs to be trained on positive and negative examples. In detection, each SVM is used to detect a part or more of one species. Thus, a set of SVMs is used to represent and detect one species. In this phase, we have two approaches:

Table 5.1: Feature extraction parameters.

Features	Parameter	Value
HOG	Colour space	RGB
	Gradient filter	[-1, 0, 1]
	Orientation bins	9
	Block size	16x16 pixel
	Cell size	8x8 pixel
	Detection window	64x96
Gabor Filter	Colour space	Gray scale
	Wavelength (λ)	0.5
	Orientation (θ)	0
	Phase shift (ψ)	30
	Aspect ratio (γ)	0.5
DWT	Colour space	Gray scale or one of the RGB channels
	Type	2D wavelet decomposition
	Levels	3
	Wavelet name	Daubechies 2

Approach 1 (Mixed negative patches): The linear SVMs are trained with positive examples that are patches within same species and same posture to represent that part(s). While negative examples are a combination of patches from other species and patches from the background. Figure 5.6 shows a sample of positive and negative examples used to train one SVM.

Approach 2 (Background negative patches): In this approach, we only use patches from the background as negative example to train the SVMs, see Figure 5.7. The positive examples still the same as the positive example in the first approach. In this approach, each SVM is only used to classify whether the patch belongs to a species or a background regardless of its type.

Positive examples (Patches from the same species)

Negative examples (Patches from background and other species)

Figure 5.6: A sample of positive and negative examples in Approach 1

Figure 5.7: A sample of negative examples in experiment 2

5.4 Recognition Phase

In recognition phase, we also identify and discriminate between species using sets of linear SVMs. Thus the recognition task is considered as a classification problem. We have also two approaches:

Approach 1 (Recognition based on detection): We firstly depend on the results obtained from *mixed negative patches* detection phase. As we use SVMs to detect 1 type of a flower, the same SVMs can be used to recognize the species by using SVMs scores, which are calculated using Equation 4.1. SVMs with high scores are used to vote for the type of the detected species.

Approach 2 (Recognition based on new SVM sets): In this approach, we have built other set of SVMs for recognition phase. Here, each species is represented by a new set of SVMs, where the positive examples are the same as the examples that are used in the detection phase. The negative examples are just patches from other species, we don't use the background patches in the negative examples. During test phase, the SVMs of the recognition phase are just used to recognize the detected parts, not a sliding window all over the full scene image. The patches that contribute to determine the location of one flower species are fed to the SVMs of recognition phase. Then, the maximum number of SVMs that vote to a specific flower species are used to determine the final recognition decision.

5.5 Results

During our evaluation, we are interested in measuring the accuracy of the detection and the recognition that are achieved by the system. In training phase, we have used three-fold cross validation technique. We are interested in demonstrating two types of results, the results of using patches during training phase and the results of applying the system on full scene images. The obtained results are discussed below.

5.5.1 Time

The system is implemented under MATLAB environment in a personal computer with 2 GB RAM and 2.30 GHz CPU speed. The runtime depends on the size of test image and the number of the hypothetical flowers in each image. To have accurate time measure, each experiment tested 3 times and the average time is taken.

Experiments show a positive relationship between the size of the image and between the number of the hypothetical flowers in the test image. The average detection runtime is 12 seconds for HOG features and 8 seconds for DWT features when using part(s) patches. For full flower patches, the average runtime is 8.6 seconds for HOG features and 6.6 seconds for DWT features. Figures 5.8 and 5.9 show the relationship between the image size and detection time.

5.5.2 Test the system using patches

The proposed approach is tested on classifying patches as flower or background (detection phase). Also, it is tested on classifying the patches to determine whether they belong to a certain flower species type or not (recognition phase). The results of detection and recognition phases are shown below.

Detection phase: The proposed approach is firstly tested on detecting the parts of the flowers, where the trained classifiers are used to classify patches to either a flower part(s) or background. Number of SVMs depends

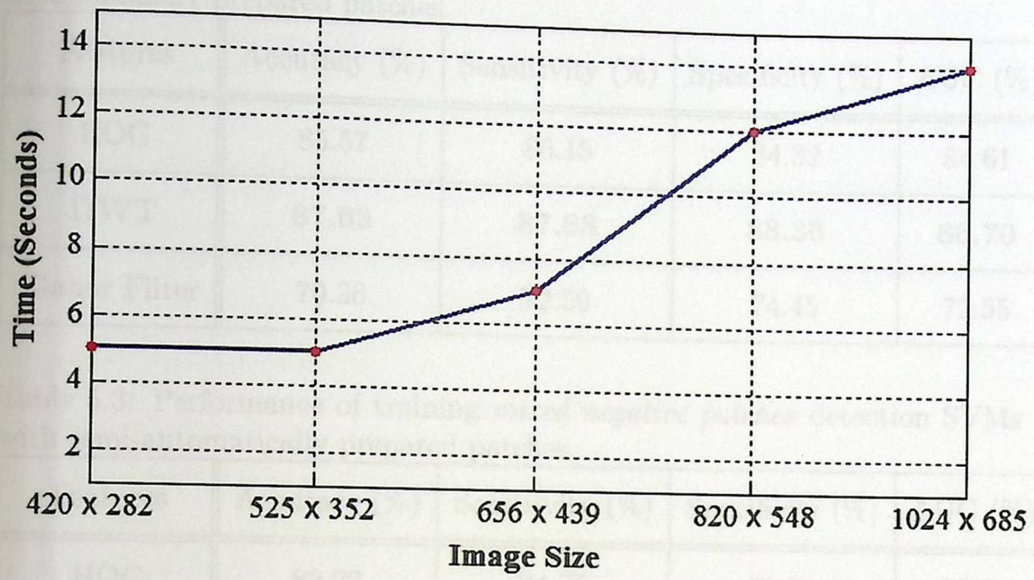


Figure 5.8: The relationship between the image size and detection based on HOG features.

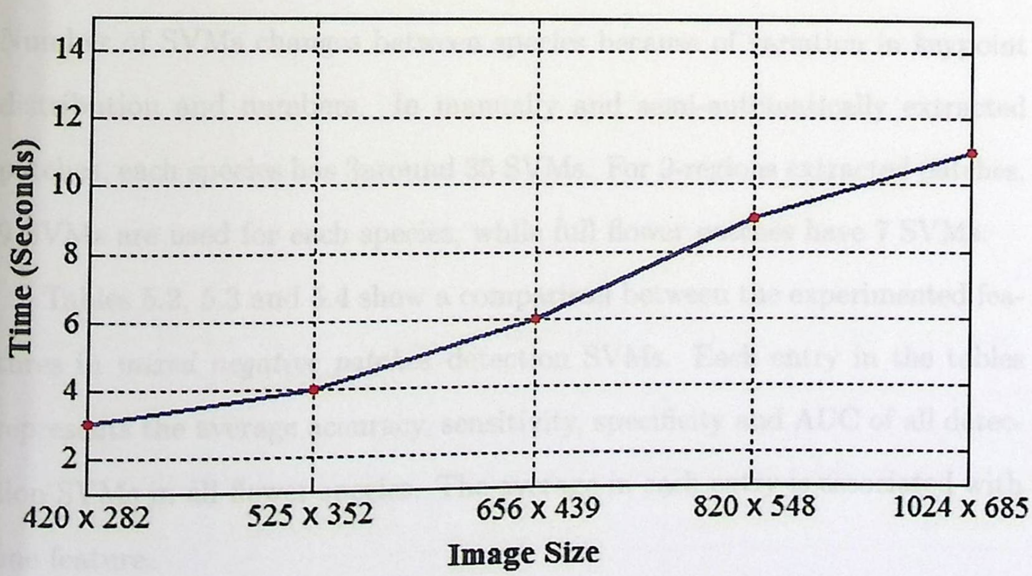


Figure 5.9: The relationship between the image size and detection based on DWt features.

Table 5.2: Performance of training *mixed negative patches* detection SVMs with manually prepared patches.

Features	Accuracy (%)	Sensitivity (%)	Specificity (%)	AUC (%)
HOG	85.57	86.15	84.32	84.61
DWT	**87.63**	**87.68**	**88.36**	**86.70**
Gabor Filter	70.36	72.59	74.45	72.55

Table 5.3: Performance of training *mixed negative patches* detection SVMs with semi-automatically prepared patches.

Features	Accuracy (%)	Sensitivity (%)	Specificity (%)	AUC (%)
HOG	82.22	84.75	81.34	82.49
DWT	**85.73**	**86.47**	**86.46**	**85.25**
Gabor Filter	67.16	69.27	70.14	68.35

on the image preparation approach and the kind of flowering plant species. Number of SVMs changes between species because of variation in keypoint distribution and numbers. In manually and semi-automatically extracted patches, each species has 3around 35 SVMs. For 9-regions extracted patches, 9 SVMs are used for each species, while full flower patches have 7 SVMs.

Tables 5.2, 5.3 and 5.4 show a comparison between the experimented features in *mixed negative patches* detection SVMs. Each entry in the tables represents the average accuracy, sensitivity, specificity and AUC of all detection SVMs in all flower species. The average in each entry is associated with one feature.

A comparison between the experimented features on training *background negative patches* detection SVMs to detect flower parts on the prepared patches are shown in Tables 5.5, 5.6, 5.7 and 5.8. For more details about the

Table 5.4: Performance of training *mixed negative patches* detection SVMs with 9-regions patches.

Features	Accuracy (%)	Sensitivity (%)	Specificity (%)	AUC (%)
HOG	81.35	83.45	79.89	80.36
DWT	**83.39**	**85.37**	**84.49**	**83.28**
Gabor Filter	64.24	67.69	68.36	66.19

Table 5.5: Performance of training *background negative patches* detection SVMs with manual patches.

Features	Accuracy (%)	Sensitivity (%)	Specificity (%)	AUC (%)
HOG	88.49	90.75	89.51	90.19
DWT	**90.65**	**91.78**	**91.94**	**91.87**
Gabor Filter	74.16	76.69	78.05	77.38

results of each species, see Appendix B.

From the obtained results, we can conclude that DWT features with gray scale images provide better performance in most cases. ROC curves for some *background negative patches* detection phase are shown in Figures 5.10, 5.11 and 5.12. The results consider the usage of the image preparation approaches.

Table 5.6: Performance of training *background negative patches* detection SVMs with semi-automatic patches.

Features	Accuracy (%)	Sensitivity (%)	Specificity (%)	AUC (%)
HOG	87.15	88.65	87.63	88.24
DWT	**88.78**	**90.10**	**89.14**	**89.68**
Gabor Filter	70.34	72.60	72.54	73.85

Table 5.7: Performance of training *background negative patches* detection SVMs with 9-regions patches.

Features	Accuracy (%)	Sensitivity (%)	Specificity (%)	AUC (%)
HOG	85.45	86.26	86.68	86.34
DWT	**86.65**	**88.39**	**87.41**	**87.62**
Gabor Filter	68.69	70.36	70.38	69.97

Table 5.8: Performance of training *background negative patches* detection SVMs with full flower patches.

Features	Accuracy (%)	Sensitivity (%)	Specificity (%)	AUC (%)
HOG	87.25	87.56	88.45	87.17
DWT	**92.15**	**93.42**	**91.43**	**92.97**

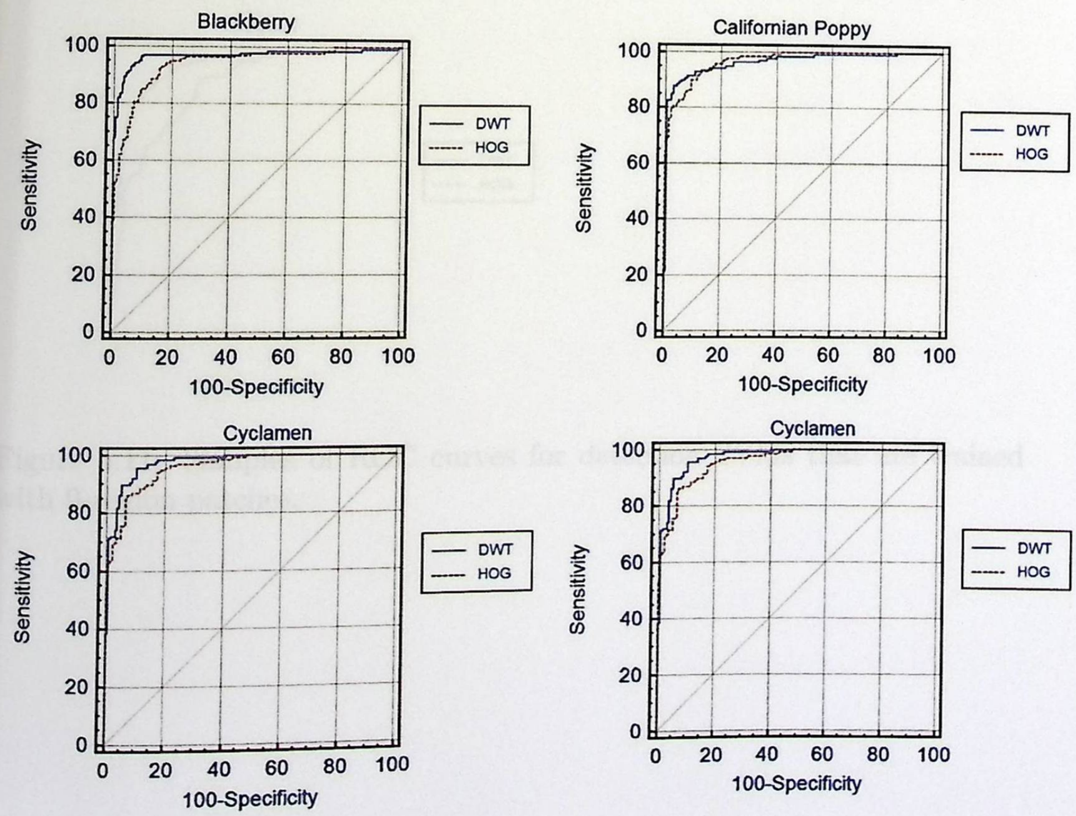


Figure 5.10: Samples of ROC curves for detection SVMs that are trained with semi-automatic patches.

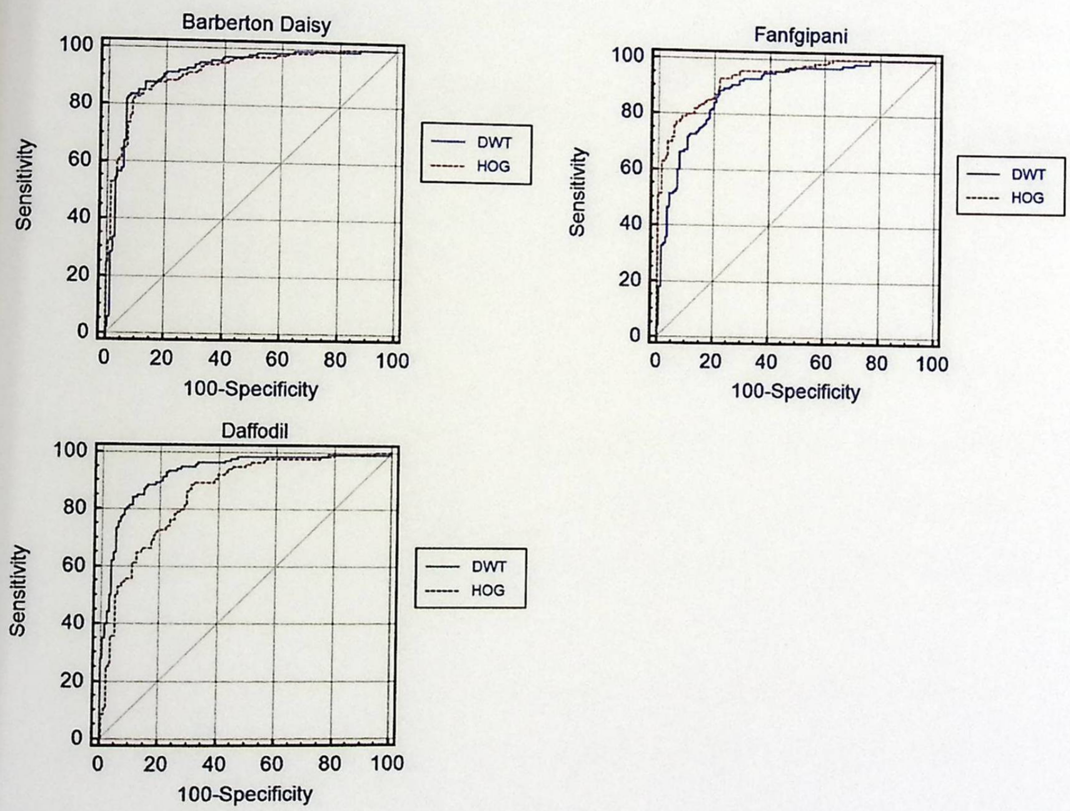


Figure 5.11: Samples of ROC curves for detection SVMs that are trained with 9-region patches.

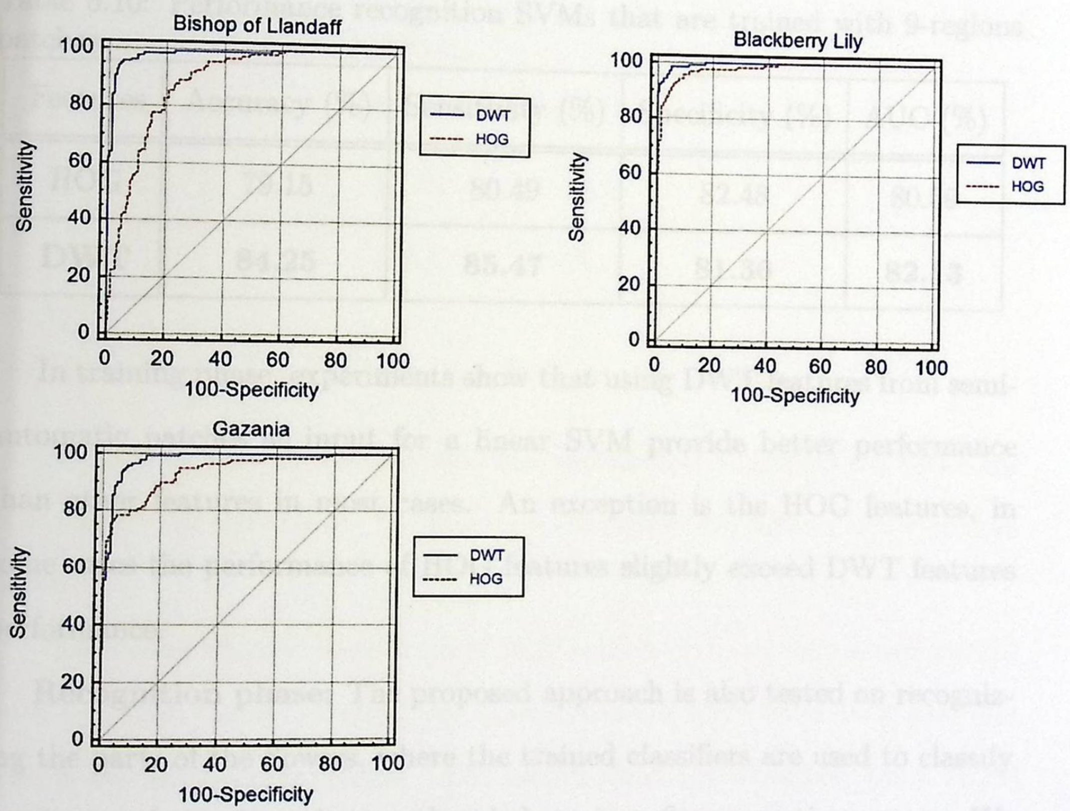


Figure 5.12: Samples of ROC curves for detection SVMs that are trained with full flower patches.

Table 5.9: Performance of recognition SVMs that are trained with semi-automatic patches.

Features	Accuracy (%)	Sensitivity (%)	Specificity (%)	AUC (%)
HOG	81.34	83.15	84.25	84.35
DWT	**86.27**	**87.42**	**85.34**	**86.64**

Table 5.10: Performance recognition SVMs that are trained with 9-regions patches.

Features	Accuracy (%)	Sensitivity (%)	Specificity (%)	AUC (%)
HOG	79.15	80.49	82.48	80.59
DWT	**84.25**	**85.47**	**81.36**	**82.13**

In training phase, experiments show that using DWT features from semi-automatic patches as input for a linear SVM provide better performance than other features in most cases. An exception is the HOG features, in some cases the performance of HOG features slightly exceed DWT features performance.

Recognition phase: The proposed approach is also tested on recognizing the parts of the flowers, where the trained classifiers are used to classify patches to determine whether they belong to a flower species or not. We ignored manual patches and GF features because they provide bad results in detection phase. The performance of *recognition based on detection* is explained in the following section because it depends on detection SVMs.

Performance of the experimented features in new recognition SVMs are shown in Tables 5.9, 5.10 and 5.11 .

Table 5.11: Performance recognition SVMs that are trained with full flower patches.

Features	Accuracy (%)	Sensitivity (%)	Specificity (%)	AUC (%)
HOG	82.75	81.98	84.17	82.19
DWT	**84.12**	**87.67**	**86.18**	**85.78**

Table 5.12: Performance of *mixed negative patches* detection SVMs trained with manually extracted patches.

Features	Accuracy (%)
HOG	55.2
DWT	58.6

5.5.3 Test the system on full scene images

In this section, we test the detection and recognition phases on full scene images. The SVMs are tested to detect and recognize full flowers not scattered parts.

Detection Phase: Each trained SVM is used to detect its trained part(s) on a test (non-trained) image. We ignored the GF features as they give bad results during training phase. The results that are obtained from *mixed negative patches* SVMs are shown in Tables 5.12, 5.13 and 5.14. Full flower patches are not tested in this approach.

Table 5.13: Performance of *mixed negative patches* detection SVMs trained with semi-automatically extracted patches.

Features	Accuracy (%)
HOG	82.4
DWT	85.6

Table 5.14: Performance of *mixed negative patches* detection SVMs trained with 9-regions patches.

Features	Accuracy (%)
HOG	85.8
DWT	87.7

Table 5.15: Performance of *background negative patches* detection SVMs trained with semi-automatically extracted patches.

Features	Accuracy (%)
HOG	83.8
DWT	85.6

The results of using *background negative* detection SVMsare shown in the Tables 5.15, 5.16 and 5.17 and. Manually extracted patches are ignored because of their bad results in the previous approach.

Recognition Phase: The detected species from detection phase then will be recognized during recognition phase. The results of recognition phase that depend on detection are shown in Table 5.18. Full flower patches are not tested in this approach.

The results of using a new set of SVMs for recognition phase are shown in the Tables 5.19, 5.20 and 5.21.

Table 5.16: Performance of *background negative patches* detection SVMs trained with 9-regions patches.

Features	Accuracy (%)
HOG	80.1%
DWT	82.4%

Table 5.17: Performance of *background negative patches* detection SVMs trained with full flower patches.

Features	Accuracy (%)
HOG	85.2%
DWT	88.9%

Table 5.18: Performance of *recognition based on detection*, where SVMs are trained with semi-automatic patches.

Features	Accuracy (%)
HOG	56.7
DWT	60.5

Table 5.19: Performance of *recognition based on new SVMs sets* that are trained with semi-automatic patches.

Features	Accuracy (%)
HOG	74.8
DWT	78.6

Table 5.20: Performance of *recognition based on new SVM sets* are trained with 9-regions patches.

Features	Accuracy (%)
HOG	72.5%
DWT	75.3%

Table 5.21: Performance of *recognition based on new SVMs sets* that are trained with full flower patches.

Features	Accuracy (%)
HOG	77.1
DWT	81.7

Table 5.22: Comparison between our proposed recognition approach and other publications

Publication	Accuracy (%)
Nilsback (2009)	76.3%
Kanan (2010)	71.4%
Ito (2010)	53.9%
Chai (2011)	90.0%
Ours	81.7%

Comparing our works with the works that are mentioned in [7], his experiments performance superior the performance of our recognition approaches. But their experiments suppose the existence of flowers in the image and flowers must also be focused. This means no detection phase in his work. Table 5.22 shows a comparison between our work and the related works in recognition task.

5.6 Discussion

In image preparation approaches, the manually extracted patches provide better performance during detection training phase due to the slight variation between training patches. The resulted patches from Approach 1 represent specific cases without having a consideration of other images in the test phase. Thus, learning process will be easier and will provide better performance than the performance of the detection SVMs that are trained with semi-manually extracted patches from Approach 2. In the later approach, there are more variation between patches, which will actually represent them in test images. It is clear in Figure 5.3 the existence of variations between patches belong to same clusters. Thus, we certain that patches have general representation not special case representation.

By comparing the results of using patches from Approach 2 and patches from Approach 3 in training detection phase classifiers, we notice that the performance of the detection SVMs using Approach 3 patches decreased. The patches that are resulted from Approach 3 have a very large variation between them, which make the learning process harder and thus will result in a slightly decreased performance. The purpose of Approach 3 is to decrease the number of SVMs that are used to detect each flower species. The reduction decreases the performance of the system; because the variation between patches in Approach 3 larger than variation between patches in Approach 2.

Using patches from other species as negative examples in the detection phase causes an excellent results during training phase, but this causes a bad results in the recognition phase during the test phase. The reduction in the performance is due to the confusion of using flower parts as negative examples. The confusion is resulted form training one SVM with examples as positives and in the same time, another SVMs is trained with a subset of that examples as negatives. To resolve the problem, we use patches only from the background as negative examples, which enhances the performance of recognition results. The performance enhancement is due to the absence of positive examples as negative examples, which reduces confusion between

SVMs. Also, this enhances detection rate, as some non-background patches will be considered as negatives, which means they are the same as background patches and this will not happen here.

Poselets detecting approach has advantage of having more than one SVM that votes for the location of flowers. But poselets require large number of SVMs. When using a mutli-scale sliding window to detect full flower patches, we have a reduction in SVMs number and time.

Chapter 6

Conclusion and future work

6.1 Conclusion

We have built a vision-based system for flower species detection and recognition using linear SVM as a classification tool to detect and recognize a flowering plant species. We experimented 4 types of patches preparation approaches, which are manually prepared patches, semi-automatically, 9-region patches and full flower prepared patches. Performance of the full flower patches superior the other patches in all experiments. We experimented 3 features namely, HOG, DWT and GF. Performance of DWT superior the other 2 features in all experiments.

The experiment of using linear SVM that is trained with background patches as negative examples superior the performance of training linear SVM with patches from background and other species. We used another set of linear SVMs to recognize the type of the flowering plant rather than depending on the detection phase results.

The propose approach is tested in our dataset, which includes 10 types of flowering plant species. Experiments show that DWT with linear SVM outperform other experimented features in both accuracy and runtime. We have achieved an accuracy rate in detecting 10 flower species reaches about 88 ± 2 %.

6.2 Future work

In this section some ideas can be used that may improve the current work. The ideas are:

1. Speed up the system either by changing he features as they take relatively long time to be extracted or other changes that may speed up the system runtime.

2. Experiment different color spaces such as HSV and LAB colors.

3. Enhance the system performance in recognition.

4. Expand the dataset to accommodate another flowering plant species.

5. Use the system in mobile applications.

Appendix A

Dataset

The dataset has 704 images divided into 10 categories with 60 to 124 images per category. For each category, two-third training images are predefined, while the rest is left for testing. A subset of images for each category is shown in the following:

Barberton Daisy

Bishop of Llandaff

Californian Poppy

Snowdrop

Blackberry Lily

Cyclamen

Daffodil

Fangipani

Gazania

Zantedeschia

Appendix B

Detailed Results

B.1 Detection phase

For the *mixed negative patches* detection phase, Figures B.1, B.2 and B.3 show the performance of detection SVMs for each flower species, which are trained by manually, semi-automatically and 9-regions extracted patches, respectively. Full flower patches are not tested in this experiment. Each bar in the figures represents the average AUC for all detection SVMs that are used to detect the associated flower species.

The obtained results of *background negative patches* detection SVMs are shown in Figures B.4, B.5, B.6 and B.7. Full patches are not tested on GF features. The figures show the performance of detection SVMs for each flower species, which are trained by manually, semi-automatically, 9-regions and full flower extracted patches, respectively.

B.2 Recognition phase

The obtained results of *recognition based on new SVMs* are shown in Figures B.8, B.9 and B.10. The figures show the performance of recognition SVMs

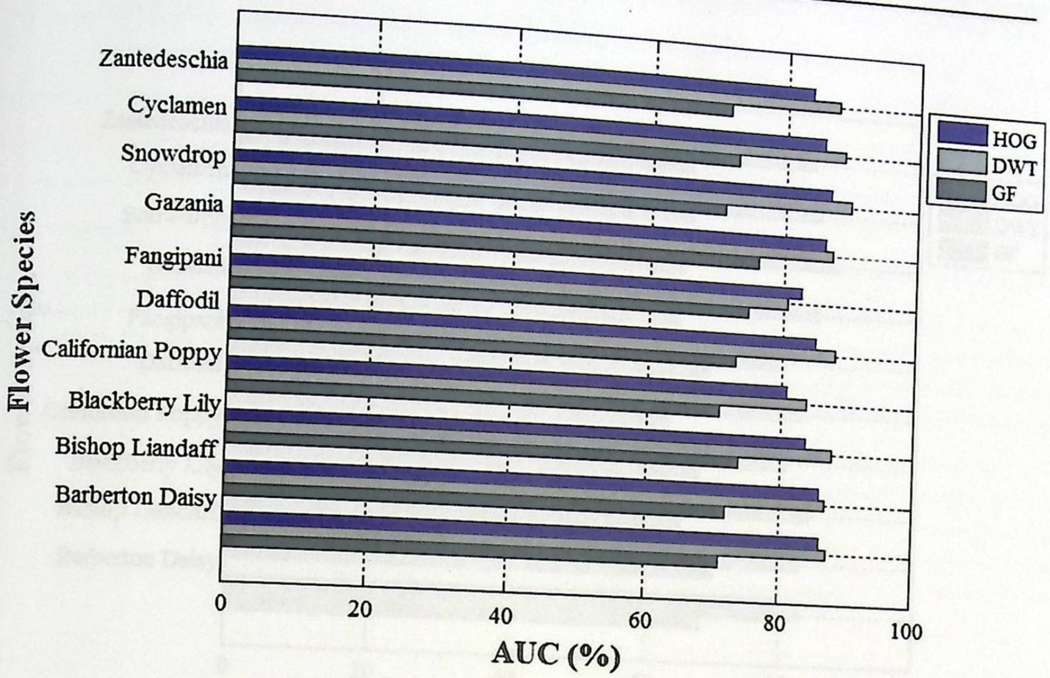


Figure B.1: Performance of training *mixed negative patches* detection SVMs for each flower species that are trained by manually extracted patches

for each flower species, which are trained by semi-automatically, 9-regions and full flower extracted patches, respectively.

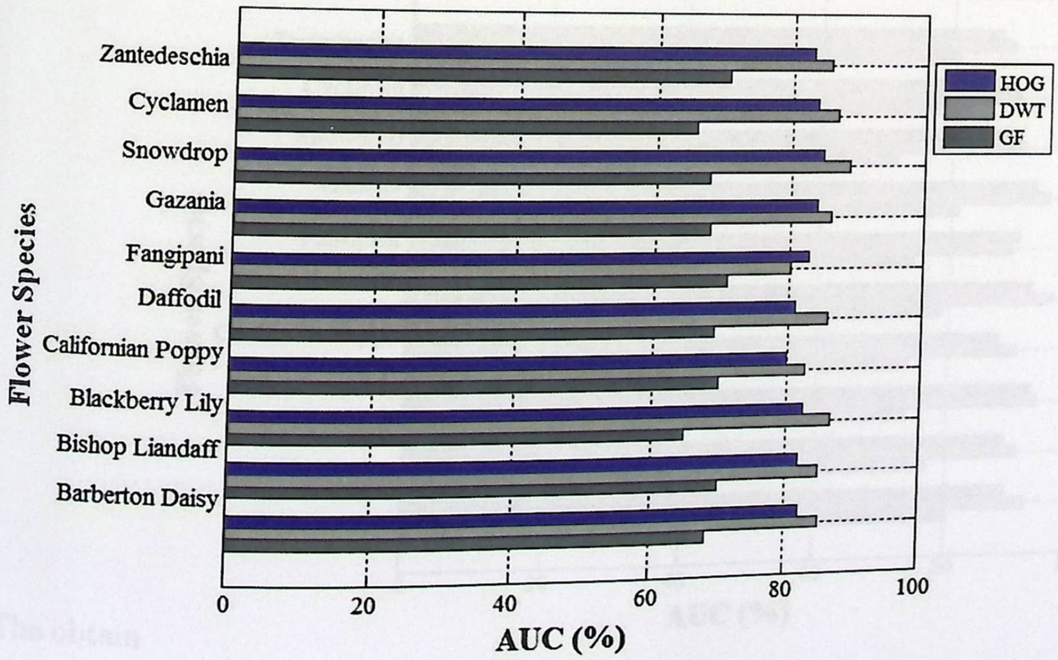


Figure B.2: Performance of training *mixed negative patches* detection SVMs for each flower species that are trained by semi-automatically extracted patches

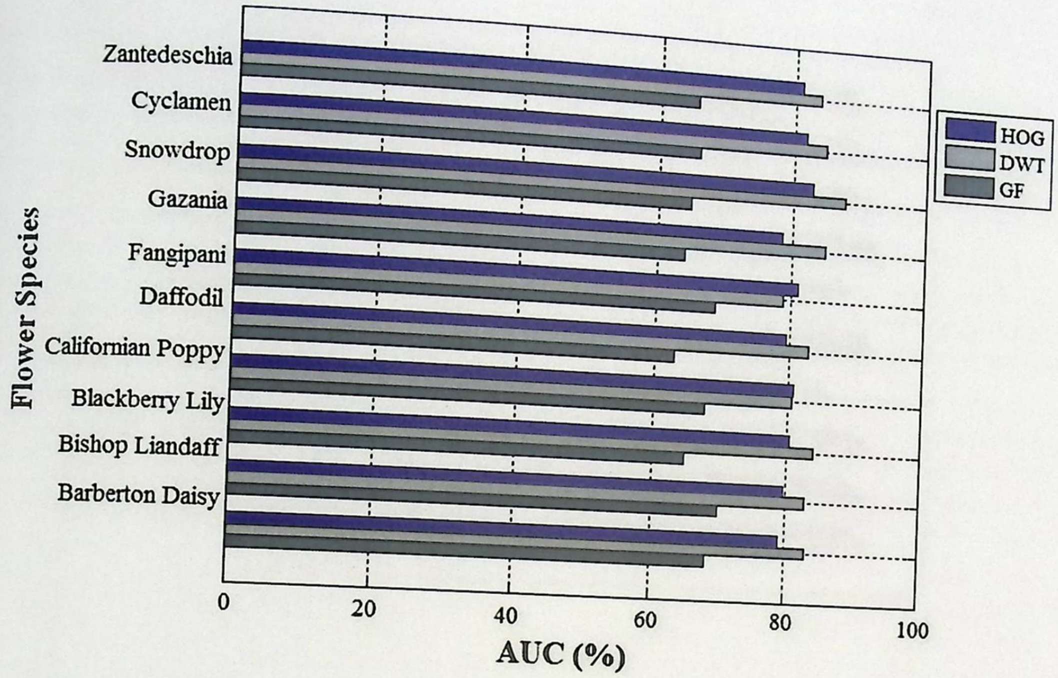
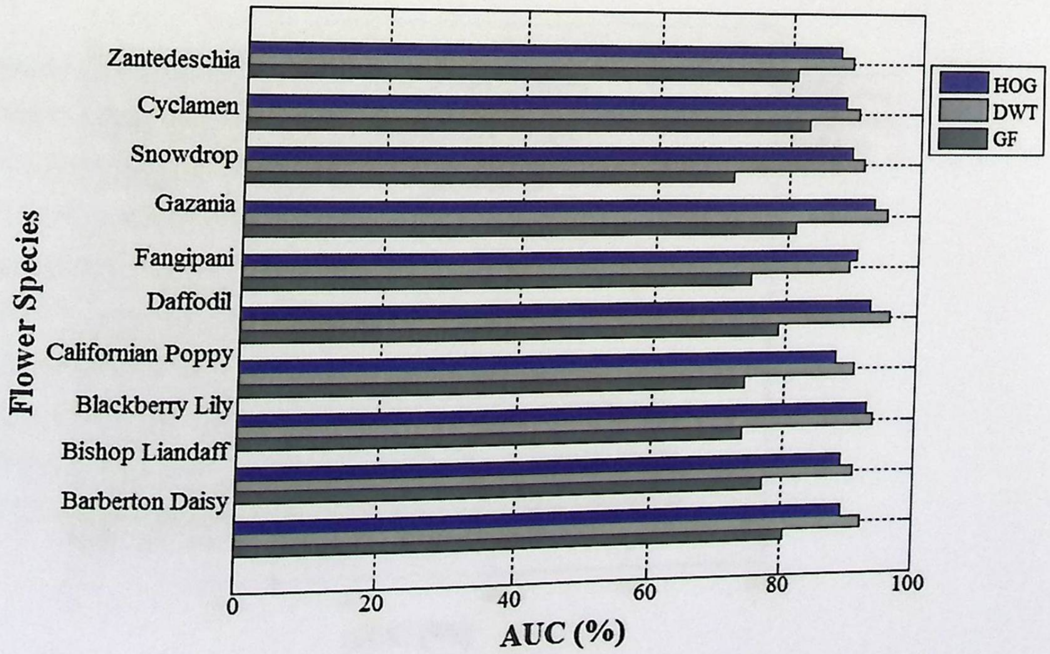


Figure B.3: Performance of training *mixed negative patches* detection SVMs for each flower species that are trained by 9-regions extracted patches



The obtain

Figure B.4: Performance of training *background negative patches* detection SVMs with manual patches.

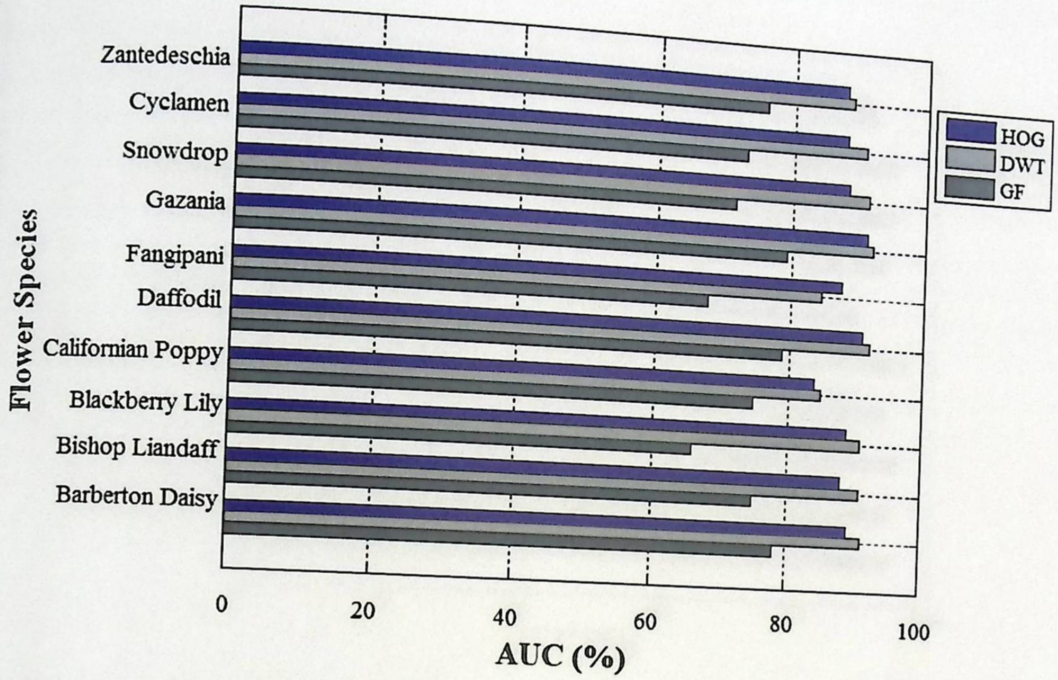


Figure B.5: Performance of training *background negative patches* detection SVMs with semi-automatic patches.

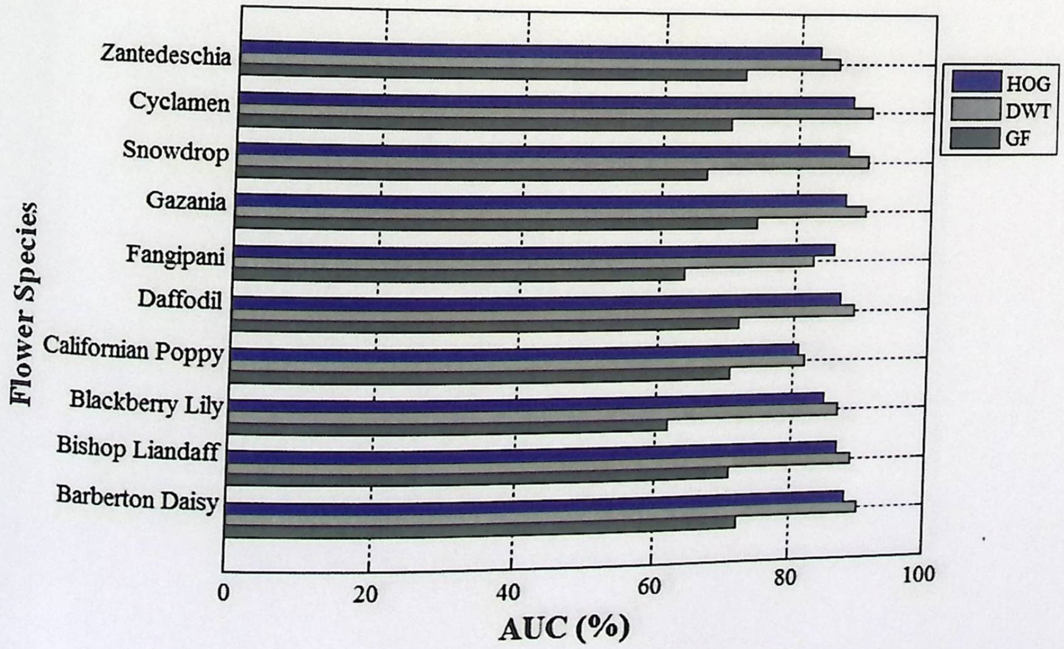


Figure B.6: Performance of training *background negative patches* detection SVMs with 9-regions patches.

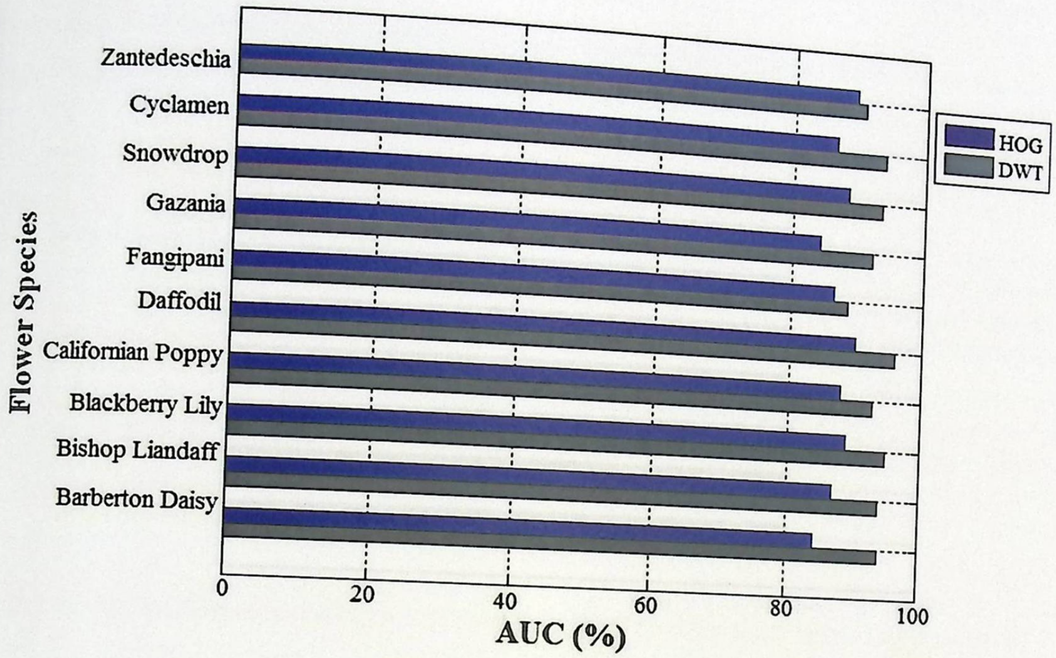


Figure B.7: Performance of training *mixed negative patches* detection SVMs for each flower species that are trained by full flower patches

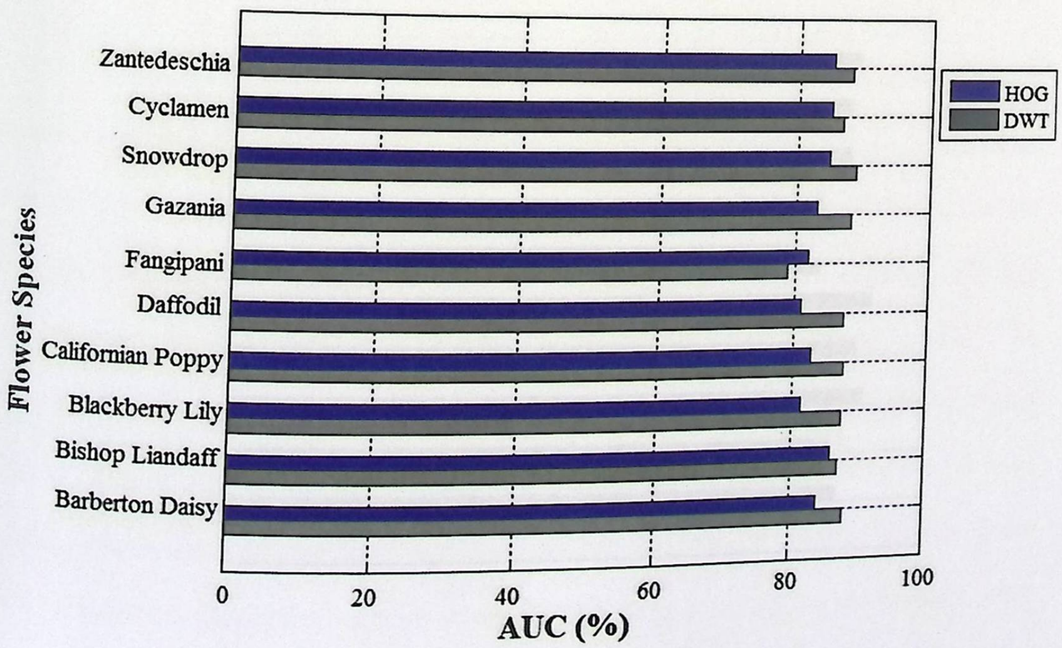


Figure B.8: Performance of recognition SVMs for each flower species that are trained by semi-automatically extracted patches

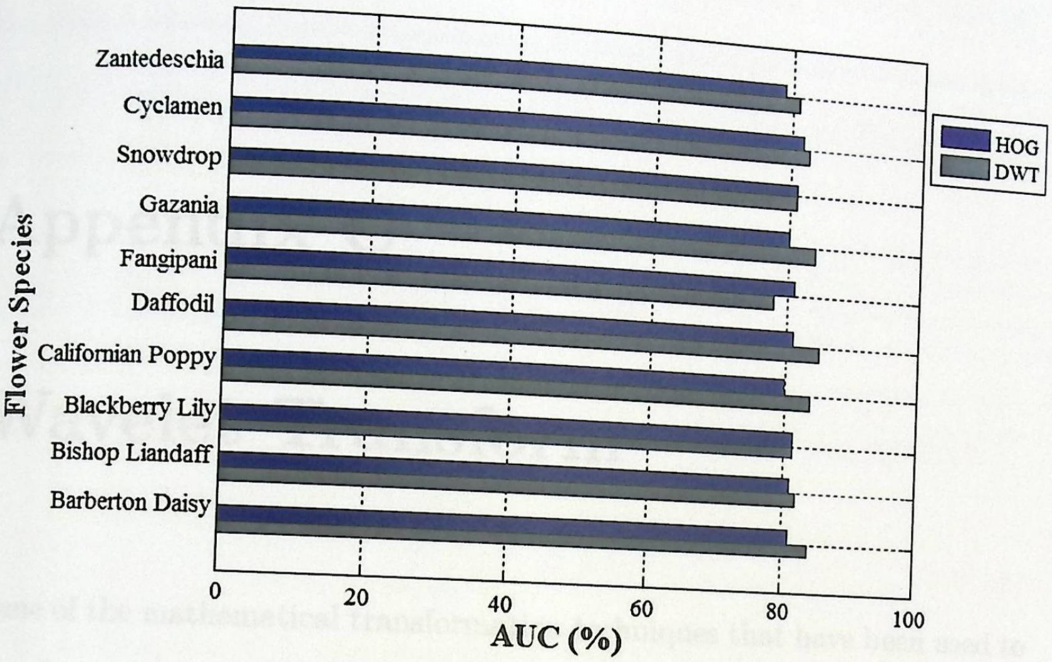


Figure B.9: Performance of training recognition SVMs for each flower species that are trained by 9-regions extracted patches

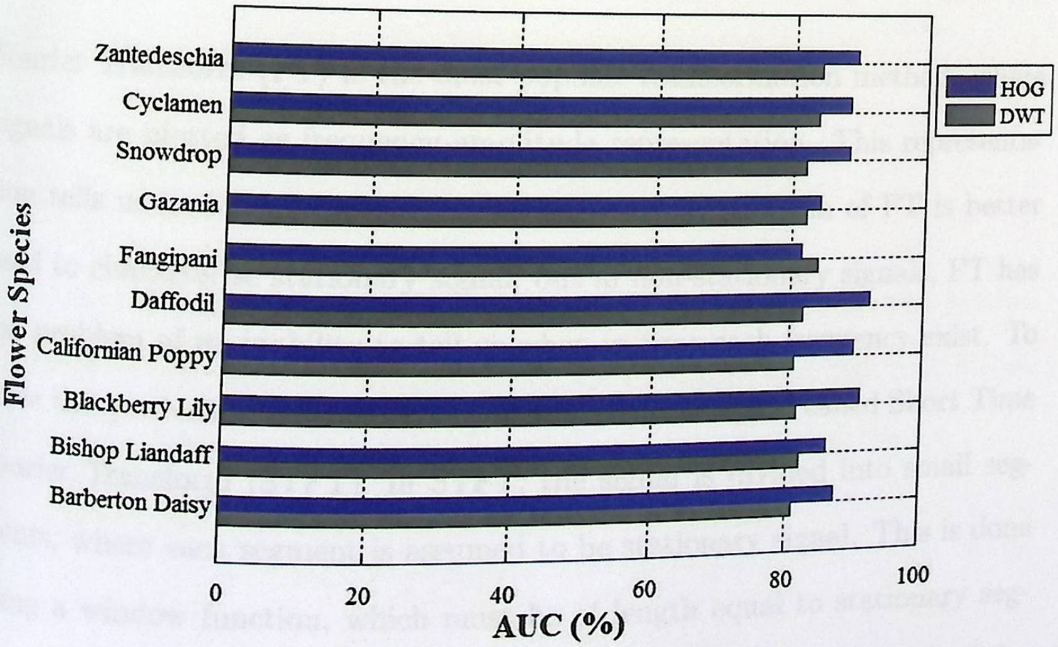


Figure B.10: Performance of training recognition SVMs for each flower species that are trained by full flower extracted patches

Appendix C

Wavelet Transform

Some of the mathematical transformation techniques that have been used to transform images from time-domain into other domains are explained in this appendix.

C.1 Fourier Transform

Fourier Transform (FT) is the most popular transformation method, where signals are plotted as frequency-amplitude representation. This representation tells us how much each frequency exists. The problem of FT is better used to characterise stationary signal, but in non-stationary signals, FT has the problem of its inability to tell us when in time each frequency exist. To solve this problem, a revised version of FT was developed, called Short Time Fourier Transform (STFT). In STFT, the signal is divided into small segments, where each segment is assumed to be stationary signal. This is done using a window function, which must be of length equal to stationary segment in the original signal. The problem here is in choosing the width of the window function, which also affect time and frequency resolution. Narrow windows will provide good time resolution, but poor frequency resolution, and vice versa for wide windows. In frequency resolution problem, we only know the frequency bandwidth without have any information about the spectral components. To solve window function problem, Wavelet Transform can be used.

C.2 Continuous Wavelet Transform

The wavelet mother function in CWT must fulfil the following condition:

$$\int_{-\infty}^{\infty} \psi(t)\, dt = 0 \tag{C.1}$$

From Equation C.1, the convolution should be done at every position and every scale, which is considered a costly process. Fortunately, images are stored discretely in computers (band-limited), thus we do not need to compute the continuous version of wavelet transform. So we have to use the Discrete Wavelet Transform (DWT) in image analyses, in which the transformation can be scaled and translated in discrete steps.

C.3 Discrete Wavelet Transform

DWT depends on two types of functions, called scaling functions and wavelet functions. These functions are associated with low pass and high pass filters, respectively. The discrete wavelet function is expressed as follows:

$$\psi_{m,n} = \frac{1}{\sqrt{a_0}^m}\psi\left(\frac{t - nb_0{a_0}^m}{{a_0}^m}\right) \tag{C.2}$$

The variables n and m are used to control the expansion and translation of the wavelet. While variables a_0 and b_0 are usually set to 2 and 1, respectively. Thus we can rewrite Equation C.2 as:

$$\psi_{m,n} = \frac{1}{\sqrt{2^m}}\psi\left(\frac{t-n2^m}{2^m}\right) \tag{C.3}$$

When DWT is used to transform a continuous signals $x(t)$, the wavelet transform is defined as:

$$T_{m,n} = \int_{-\infty}^{\infty} x(t)\frac{1}{a_0{}^{m/2}}\psi\left(a_0{}^{-m}t - nb_0\right) dt \tag{C.4}$$

This is used to measure the similarity between the wavelet at the current scale and the signal itself. It can be calculate using the inner product:

$$T_{m,n} = \langle x, \psi_{m,n}\rangle \tag{C.5}$$

$T_{m,n}$ are the detail wavelet coefficients.

C.3.1 Orthonormal Wavelets

In orthonormal wavelets, the wavelets are pairwise orthogonal and all of unit energy. This type of wavelets can be expressed as:

$$\int_{-\infty}^{\infty} \psi_{m,n}(t)\,\psi_{m',n'}(t)\,dt = \begin{cases} 1 & \text{if } m = m' \text{and } n = m' \\ 0 & otherwise \end{cases} \tag{C.6}$$

Orthonormal wavelets are desirable because they simplify the computation of wavelet coefficients. From Equation C.6, the product of any wavelet with all other wavelets equal zero. This yields non-redundant transform, in which the reconstruction of signals can be done without redundancy.

C.3.2 Wavelet Scaling Function

The wavelet scaling function is considered as a band-pass filter [wiki], where it is used to have a finite number of wavelets. It is associated with the signal and can be expressed as follows:

$$\phi_{m,n}(t) = 2^{-m/2}\phi\left(2^{-m}t - n\right) \tag{C.7}$$

It also must satisfy the admissibility condition:

$$\int_{-\infty}^{\infty} \phi(t)\, dt = 1 \tag{C.8}$$

To produce the approximate coefficients from scaling function, the signal is convolved with it as follows:

$$S_{m,n} = \int_{-\infty}^{\infty} x(t)\, \phi_{m,n}(t)\, dt \tag{C.9}$$

The scaling function must have the following property:

$$\phi(t) = \sum_{k} c_k \phi(2t - k) \tag{C.10}$$

where $\phi(2t-k)$ represents the compressed version of $\phi(t)$, which is compressed by a factor of 2 and is translated by k [h]. The variable c_k represents the scaling coefficient that must satisfy the following condition:

$$\sum_{k} = c_k = 2 \tag{C.11}$$

The mother wavelet function is constructed using the scaling function, the construction process is done using the following equation:

$$\psi_{m,n}(t) = \sum_k (-1)^k c_{N_k-1-k}\phi(2t-k) \tag{C.12}$$

Assume we have N_k coefficients, the sum of them is zero. The Equation C.12 can be rewritten as:

$$\psi_{m,n}(t) = \sum_{k=0}^{N_k-1} b_k\phi(2t-k) \tag{C.13}$$

where b_k is:

$$b_k = (-1)^k c_{N_k-1-k} \tag{C.14}$$

Referring to Equation C.7 and C.10, the scaling function at index $m+1$ (next scaling function) can be calculated as follows:

$$\phi_{m+1,n}(t) = \frac{1}{\sqrt{2}}\sum_k c_k\phi_{m,2n+k}(t) \tag{C.15}$$

The associated wavelet function is expressed as:

$$\psi_{m+1,n}(t) = \frac{1}{\sqrt{2}}\sum_k b_k\phi_{m,2n+k}(t) \tag{C.16}$$

Thus, the scaling function and the wavelet function are composed of shifted scaling functions, each is factored by its scaling coefficients.

Bibliography

[1] P. Addison. *The illustrated wavelet transform handbook: introductory theory and applications in science, engineering, medicine, and finance.* IoP, 2002.

[2] B. Askham. Flower power, 2008.

[3] E. Bayro-Corrochano and N. Arana-Daniel. Clifford support vector machines for classification, regression, and recurrence. *Trans. Neur. Netw.*, 21(11):1731–1746, November 2010.

[4] P. Belhumeur, D. Chen, S. Feiner, D. Jacobs, W. Kress, H. Ling, I. Lopez, R. Ramamoorthi, S. Sheorey, S. White, and L. Zhang. Searching the world's herbaria: A system for visual identification of plant species. pages 116–129. 2008.

[5] L. Bourdev and J. Malik. Poselets: Body part detectors trained using 3d human pose annotations. In *IEEE 12th International Conference on Computer Vision, ICCV 2009, Kyoto, Japan, September 27 - October 4, 2009*, pages 1365–1372. IEEE, 2009.

[6] C. Burges. A tutorial on support vector machines for pattern recognition. *Data Min. Knowl. Discov.*, 2(2):121–167, 1998.

[7] Y. Chai, V. Lempitsky, and A. Zisserman. Bicos: A bi-level co-segmentation method for image classification. In *IEEE International Conference on Computer Vision*, 2011.

[8] C. Chang and C. Lin. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2:27:1–27:27, 2011. Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.

[9] H. Demuth. *Neural Network Toolbox 6*, 1992.

[10] P. Felzenszwalb, R. Girshick, and D. McAllester. Cascade object detection with deformable part models. pages 2241–2248. IEEE, 2010.

[11] P. Felzenszwalb, R. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part-based models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(9):1627–1645, 2010.

[12] T. Fletcher. Support vector machines explained, 2099.

[13] G. Flitton, T. Breckon, and N. Megherbi. Object recognition using 3D SIFT in complex CT volumes. In *Proc. British Machine Vision Conference*, pages 11.1–12, September 2010.

[14] Y. Freund. Boosting a weak learning algorithm by majority. *Information Computing*, 121(2):256–285, September 1995.

[15] R. Gonzalez and R. Woods. *Digital Image Processing.* Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 2nd edition, 2001.

[16] S. Gunn. Support vector machines for classification and regression, May 1998.

[17] Ch. Heng. Machine learning notes series 2: Offline impact assessment, 2004.

[18] T. Hsu, C. Lee, and L. Chen. An interactive flower image recognition system. *Multimedia Tools Appl.*, 53(1):53–73, May 2011.

[19] S. Ito and S. Kubota. Object classification using heterogeneous co-occurrence features. In *Proceedings of the 11th European conference on Computer vision: Part II*, ECCV'10, pages 209–222. Springer-Verlag, 2010.

[20] C. Kanan and G. Cottrell. Robust classification of objects, faces, and flowers using natural image statistics. In *The Twenty-Third IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2010, San Francisco, CA, USA, 13-18 June 2010*, pages 2472–2479. IEEE, 2010.

[21] C. Kanan and G. Cottrell. Robust classification of objects, faces, and flowers using natural image statistics. In *The Twenty-Third IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2010, San Francisco, CA, USA, 13-18 June 2010*, pages 2472–2479. IEEE, 2010.

[22] L. Zhangand J. Kong, X. Zeng, and J. Ren. Plant species identification based on neural network. In *Proceedings of the 2008 Fourth International Conference on Natural Computation - Volume 05*, ICNC '08, pages 90–94. IEEE Computer Society, 2008.

[23] S. Maji L. Bourdev and J. Malik. Describing people: A poselet-based approach to attribute classification. In *Proceedings of the 2011 International Conference on Computer Vision ICCV 2011*, pages 1543–1550. IEEE, 2011.

[24] T. Brox L. Bourdev, S. Maji and J. Malik. Detecting people using mutually consistent poselet activations. In *Proceedings of the 11th European conference on Computer vision: Part VI*, ECCV'10, pages 168–181. Springer-Verlag, 2010.

[25] D. Lowe. Object recognition from local scale-invariant features. In *Proceedings of the International Conference on Computer Vision-Volume 2 - Volume 2, ICCV '99*. IEM. Andriluka, S. Roth, B. Schiele.EE Computer Society, 1999.

[26] B. Schiele M. Andriluka, S. Roth. Pictorial structures revisited: People detection and articulated pose estimation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009.

[27] M. Mahmoud, M. Dessouky, S. Deyab, and F.. Elfouly. Comparison between haar and daubechies wavelet transformions on fpga technology. In *Proceedings of world academy of science engineering and technology*, volume 20, Apri 2007.

[28] S. Maji, L. Bourdev, and J. Malik. Action recognition from a distributed representation of pose and appearance. In *Proceedings of the 2011 IEEE Conference on Computer Vision and Pattern Recognition*, CVPR '11, pages 3177–3184. IEEE Computer Society, 2011.

[29] M. Wieling N. Petkov. Gabor filter for image processing and computer vision, 2008.

[30] D. Nasien, S. Yuhaniz, and H. Haron. Statistical learning theory and support vector machines. In *Proceedings of the 2010 Second International Conference on Computer Research and Development*, ICCRD '10, pages 760–764. IEEE Computer Society, 2010.

[31] D. Navneet and T. Bill. Histograms of oriented gradients for human detection. In *In CVPR*, pages 886–893, 2005.

[32] M. Nilsback. *An Automatic Visual Flora - Segmentation and Classification of Flowers Images - Master thesis.* 2009.

[33] M. Nilsback and A. Zisserman. Automated flower classification over a large number of classes. In *Proceedings of the Indian Conference on Computer Vision, Graphics and Image Processing*, Dec 2008.

[34] M. Nilsback and A. Zisserman. 102 category flower dataset, 2009.

[35] M. Nilsback and A. Zisserman. 17 category flower dataset, 2009.

[36] R. Poliker. The wavelet tutorial, 2001.

[37] X. Ren, A. Berg, and J. Malik. Recovering human body configurations using pairwise constraints between parts. In *Proceedings of the Tenth IEEE International Conference on Computer Vision (ICCV'05) Volume 1 - Volume 01*, ICCV '05, pages 824–831. IEEE Computer Society, 2005.

[38] E. Schapire. The strength of weak learnability. *Machine Learning*, 5(2):197–227, 1990.

[39] D. Sebald and J. Bucklew. Support vector machine techniques for nonlinear equalization. *Trans. Sig. Proc.*, 48(11):3217–3226, November 2000.

[40] F. Siraj, M. Salahuddin, and S. Yusof. Digital image classification for malaysian blooming flower. In *Proceedings of the 2010 Second International Conference on Computational Intelligence, Modelling and Simulation*, CIMSIM '10, pages 33–38. IEEE Computer Society, 2010.

[41] V. Vuarnoz. *Flower recognition-Master Thesis.* 2010.

[42] J. Walker. *A Primer on Wavelets and their Scientific Applications.* Charman and Hall/CRC, 1999.