

**PLANT IDENTIFICATION USING LOCAL INVARIANTS:
DENSE SIFT APPROACH**

by
Seyfettin Tolga Yıldıran

Submitted to the Graduate School of Engineering and Natural Sciences
in partial fulfillment of
the requirements for the degree of
Master of Science

Sabancı University

August 2015

PLANT IDENTIFICATION USING LOCAL INVARIANTS:
DENSE SIFT APPROACH

APPROVED BY

Assoc. Prof. Dr. Berrin YANIKOĞLU
(Thesis Supervisor)

Assoc. Prof. Dr. Erchan APTOULA

Assoc. Prof. Dr. Selim BALCISOY

DATE OF APPROVAL: 05 AUGUST 2015

© Seyfettin Tolga Yıldırım 2015
All Rights Reserved

Acknowledgments

My sincere gratitude is to my supervisor Berrin Yanikođlu for her excellent guidance, patience, and motivation.

I am grateful to my committee members, Assoc. Prof. Erchan Aptoula and Selim Balcısoy for taking the time to read and their useful comments and advice on my thesis.

I also would like to thank Sabancı University and TÜBİTAK for providing the necessary financial support for my graduate education¹.

I am also indebted to my colleagues and friends and my other fellow lab-mates from the Computer Vision and Pattern Analysis Laboratory for their assistance and suggestions.

Most importantly, none of this would have been possible without the love and patience of my family. I would like to express my heart-felt gratitude to my family who are always supporting me and encouraging me with their best wishes.

¹This work was partially supported by the Scientific and Technological Research Council of Turkey under Grants 113E499.

PLANT IDENTIFICATION USING LOCAL INVARIANTS:
DENSE SIFT APPROACH

Seyfettin Tolga Yıldırım

CS, M.Sc. Thesis, 2015

Thesis Supervisor: Berrin YANIKOĞLU

Keywords: SIFT, SWHH, SVMs, CLEF, Flower, ROI

Abstract

In this thesis, we investigate the use of Dense SIFT approach in automatic identification of plants from photographs. We concentrate on flowering plants and evaluate three alternative approaches. In the first one, we classify the plant directly using the dense SIFT method, using appropriate parameters that are found using experimental validation techniques.

In the second approach, we first identify the dominant colour in the photograph and use a separate classifier in each of the colour cluster. The second approach is intended to reduce the problem complexity and the number of classes handled by each classifier. In this approach, the classifier for red flowers will not know about a plant that does not flower in red; furthermore a plant that is only observed with red flowers will only be handled by that classifier.

In a third approach, we precede the second approach by adding a Region of Interest detector, in order to extract the flower color more reliably.

We find that enhancement of Dense SIFT features based identification is possible with saturation-weighted hue histogram based color clustering and region of interest detector. Using the proposed system, we obtain a 0.60 accuracy on the flower subset in the LifeCLEF 2014 database.

YEREL DEĞİŞKENLERLE BİTKİ TANIMA: YOĞUN SIFT YONTEMI

Seyfettin Tolga Yıldırım

CS, Yüksek Lisans Tezi, 2015

Tez danışmanı: Berrin YANIKOĞLU

Anahtar Kelimeler: Bitki tanıma, Yerel Değişmezler, SIFT, Doğunluk Ağırlıklı Renk Özü histogramı, SVMs, CLEF, Çiçek, İlgı Bölgesi

Özet

Bu tezde, bitki resimleri üzerinde Dense SIFT yönteminin farklı veri yapıları ile uygulanarak bitki türünün tanınmasını araştırdık. Çiçeklenmiş bitkileri üç farklı yöntem ile tanımlamak üzerine odaklandık. Birinci yaklaşım Dense SIFT yönteminin işlem görmemiş resimlere doğrudan uygulanması ile tanıma, ikinci yaklaşım renk özniteliklerine dayalı kümelere ayırdığımız veri parçalarında Dense SIFT uygulayarak tanıma, üçüncü yaklaşımda ise ilgi bölgesi yöntemi ile odak noktaları seçme işlemini ikinci yaklaşımın öncesinde uygulayarak tanımayı denedik.

Bu çalışmaların sonucunda Dense SIFT yöntemi ile bitki tanınmasında doğunluk ağırlıklı renk özü histogramı ve ilgi bölgesi yöntemleri kullanılarak iyileştirmenin mümkün olduğunu gözlemledik. Tasarlanan sistemi kullanarak, LifecLEF 2014 veritabanı çiçek alt kümesinde 0.60 doğru tanıma başarısını elde ettik.

Table of Contents

Acknowledgments	iv
Abstract	v
Özet	vi
1 Introduction	1
1.1 Scope	1
1.2 Motivation	2
1.3 Contributions	2
1.4 Outline	3
2 CLEF Plant Image Retrieval Campaigns	4
2.1 LifeCLEF 2014 Plant Identification Campaign	5
3 Classification with Local Invariants	8
3.1 Scale Invariant Feature Transform (SIFT)	8
3.1.1 Interest Point Detection	10
3.1.2 SIFT Descriptors	10
3.2 Dense SIFT	12
3.2.1 SIFT Descriptors	12
3.2.2 Effect of Grid Sizes	13
3.3 Bag of Words Representation	14
3.3.1 Size of the Dictionary	16
3.3.2 K-Means Clustering	17
3.4 Support Vector Machines (SVMs)	19
4 Color Based Classification	22
4.1 Outline	22
4.2 Flower Taxonomy	24
4.3 Color Descriptors	26
4.3.1 Saturation Weighted Hue Histogram	26
4.3.2 Saturation Weighted Hue Statistics	27
4.3.3 SWH Histogram Extraction	27
4.4 Color Clusters	30
4.5 Classification Using Random Forests	32
4.6 System Overview	32

5	Classification with Cropped Images by Saliency Map based ROI	
	Detection	36
5.1	System Outline	36
5.2	Region of Interest (ROI) (Saliency Map)	39
5.3	Salient Region Detection and Image Cropping	40
6	Evaluation	43
6.1	Data Set	43
6.2	Results	44
	6.2.1 System I - Plant Identification Using Dense SIFT	44
	6.2.2 System II and III - Color Classification	44
	6.2.3 System II - Plant Identification	47
6.3	Summary	49
7	Appendix	51
	Bibliography	53

List of Figures

2.1	LifeCLEF 2014 Plant identification task in different categories.	5
2.2	LifeCLEF 2014 Flower Data Set	7
3.1	System I	9
3.2	SIFT Keypoints [1]	11
3.3	SIFT Descriptors [2]	11
3.4	Dense SIFT	13
3.5	Dense Grid Sizes	14
3.6	Word Histogram	15
3.7	K-Means Clusters	17
3.8	System I Example	21
4.1	System II	23
4.2	"Narcissus" Flower Family	25
4.3	Achromatic Axis	27
4.4	(a) Colour image. (b) Hue of image (a). (c) Saturation of image (a). (d) Luminance of image (a).	28
4.5	Hue Histogram of <i>Figure 4.4.a</i>	29
4.6	Saturation-Weighted Hue Histogram of <i>Figure 4.4.a</i>	29
4.7	White Yellow Color Cluster	31
4.8	Gagea Granatelli Parl	33
4.9	Orchis Anthropophora	33
4.10	System II Example	34
5.1	Improved System II	38
5.2	Saliency Detection Algorithm [3]	40

5.3 ROI Samples; (a)left column Original images, (b)middle column Saliency
Maps, (c)right column ROI detected images 42

List of Tables

2.1	LifeCLEF Plant Identification Campaign 2014 data set	6
4.1	Flower Subset with Ground-truth	32
6.1	Flower Subset with Ground-truth	44
6.2	System I - Plant Identification	44
6.3	System II - Color Classification Results	45
6.4	System II with ROI - Color Classification Results	46
6.5	System II - Plant Identification Results	47
6.6	System II with ROI - Plant Identification Results	47
6.7	Results of All Systems	49

Chapter 1

Introduction

Identification of plants from photographs has been gaining interest, especially with the increase in hand-held devices. The problem has many application areas, ranging from helping botanists and the general public identify plants that they encounter, to identifying harmful species (e.g. poison ivy).

This thesis addresses one particular approach for this problem, namely the use of local invariants. Local invariants, such as the Scale Invariant Feature Transform (SIFT) is an approach that has been proposed in 1999 [4] and has been widely used in many type of object recognition problems challenges [5]. The SIFT approach is particularly suited for the plant identification problem where the object (the plant) may show variations common to other object recognition problems, namely color, lighting, pose, and scale variations, as well as large variations in shape due to age and leaf composition of the particular plant.

1.1 Scope

We propose a system to address the problem of plant identification from a given photograph, based on local invariants. We focus on flowering plant photographs; however the method is equally applicable to fruit-bearing plant photographs that show similar characteristics. Furthermore, the underlying components used in this thesis (bag of words representation with dense SIFT features) is applicable to all plant photographs.

We envision that the approach can be a specialized module of a complete plant identification system. Indeed, identifying a plant from pictures of its different organs

(e.g. flower, leaf, stem, overall view) or pictures taken in different formats (e.g. a partial photo versus a scanned leaf image) may require different approaches.

1.2 Motivation

Plant identification has been drawing more and more attention in recent years. There are several work that address sub-problems issues relating to agriculture (e.g. identifying diseased crop etc), and others that are more suitable for botanical applications (e.g. identifying poisonous plants).

Our work has started with the ImageCLEF Plant Identification campaigns organized since 2011 [6, 7]. The campaign is organized within CLEF (Conference and Labs of the Evaluation Forum) that organizes campaigns in order to benchmark progress in the general area of multi-lingual or multi-modal text and image retrieval problems.

1.3 Contributions

We compare two different plant Identification methods, geared especially for flowering plants. One of them is the state-of-art image classification method based on Bag of Words *BoW* model using dense SIFT features. In this case, we have a single classifier to classify all the flowering plant types considered by the system (namely the 212 plants in the LifeCLEF 2014 Flower data set). The second method uses a color classification step as a pre-classifier step, so as to divide the problem into smaller subproblems. In this case, the dense SIFT classifiers are trained within the plant species falling in each colour cluster. Finally an improved version of the second method works by selecting the Region of Interest (ROI) as the first step and proceeds with the cropped image, with the aim of better estimating the flower color group. In this method, we select the region of interest using the Saliency Map of raw images.

Our main contribution is to show the applicability of Dense SIFT features to the plant identification problem and showing that it can lead to state-of-art accuracy results. Our secondary contributions are showing that colour clustering and region

of interest detections help in this problem, further improving accuracy.

1.4 Outline

In Chapter 2, we describe the ImageCLEF 2014 Plant Identification campaign and the Flower data set.

In Chapter 3, the first system, named as Classification with Local Invariants is explained with all used approaches: Scale Invariant Feature Transform (*SIFT*), Dense SIFT, Bag Of Words *BoWs*, Support Vector Machines(*SVMs*).

In Chapter 4, we present the color classification operation, using the Saturation Weighted Hue Histogram(*SWHH*) and Random Forest classifier.

In Chapter 5, we propose an improved version for second method by automatically detecting the region of interest (ROI), so as to better estimate the color group of the flower.

In Chapter 6, results from all systems are reported with details.

Chapter 2

CLEF Plant Image Retrieval Campaigns

The ImageCLEF lab is a component of *CLEF*, the Cross Language Evaluation Forum that aims to systematically evaluate multi-modal and cross-language information retrieval systems, since 2011. The activities of the forum are presented in separate conference tracks [8] [9] [10] [11] [12] [7].

ImageCLEF organizes four main challenging tasks related to image annotation, for a wide range of source images and annotation objective. These are general multi-domain images for object or concept detection, as well as domain-specific tasks exist such as visual-depth images for robot vision and volumetric medical images for automated structured reporting. The goal of setting up these challenges is to support and promote cutting-edge research addressing the key challenges in the field.

In 2014, the LifeCLEF lab is split from the ImageCLEF with the goal of benchmarking challenges related to life and nature, in continuity of the image-based plant identification task. The considered tasks in 2014 are plant, bird, and fish classification. While most research is ran with a few hundreds of species/categories, LifeCLEF tries to increase the number of species to real life sizes. In 2014 and 2015, the plant retrieval campaign has reached a database containing almost all plant species of France.

2.1 LifeCLEF 2014 Plant Identification Campaign

Plant Identification is one of three main campaigns of LifeClef 2014, ran to benchmark progress in this problem. Given a plant photograph, the problem is to return the correct plant species among a ranked list of species. Hence, the problem is setup and evaluated as an image retrieval problem, rather than a classification problem.

The input images mostly consist of photographs captured from nature, while some are scanned leaves with a simple color background. The campaign includes seven view types, corresponding to seven subtasks: branch, flower, fruit, stem, leaf, entire plant, or leaf scan. Samples from different view types are depicted in Fig. 2.1.

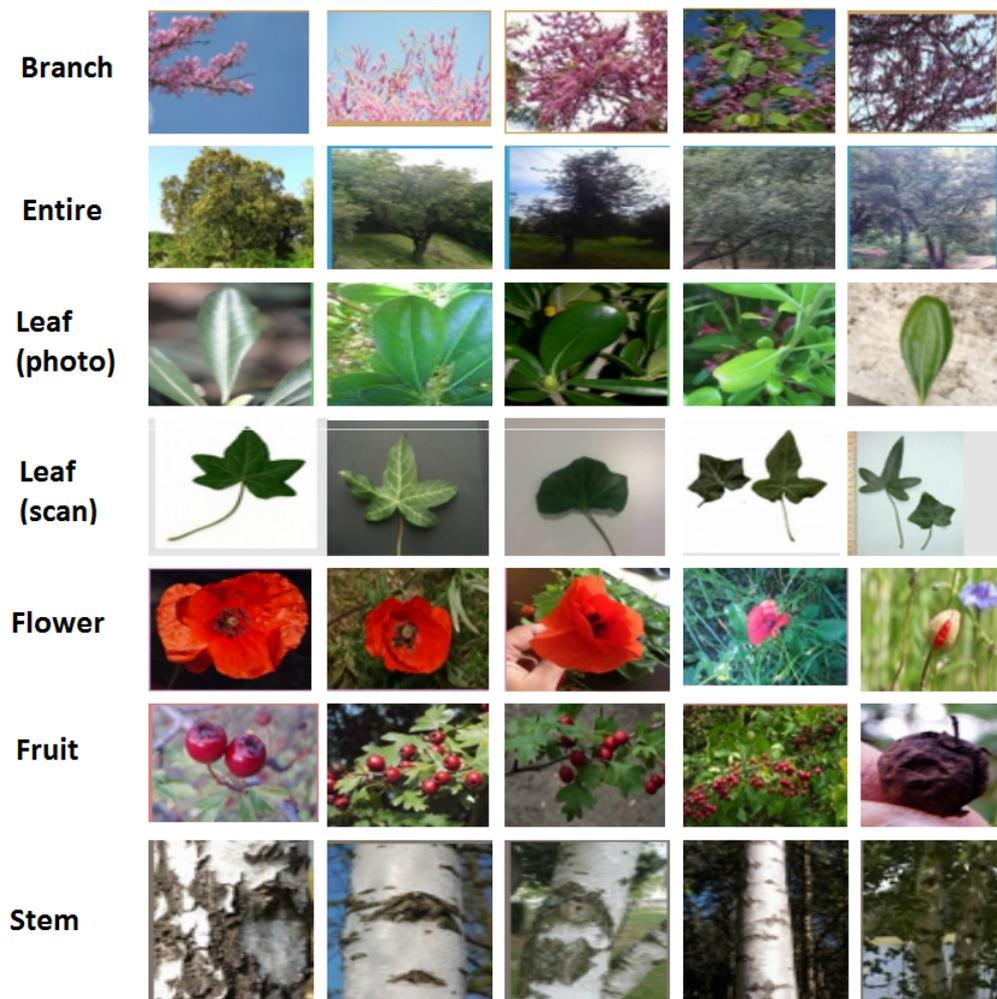


Figure 2.1: LifeCLEF 2014 Plant identification task in different categories.

The plant photographs are all collected in France, by a network of botanists and plant enthusiasts. The number of plant species are growing since the first organization in 2011 when there were 71 plant species, to 126 and 250 species in 2012 and 2013, respectively [8] [9] [10]. Since LifeClef 2014, the number of species is about 500, indicating a relatively large image retrieval problem [11] [12] [7].

The campaign has designated training and test data sets. The training data contains 47,815 images and test data contains 13,146 images [7]. Details with reference to train and test image numbers of each view type are represented in Table. 2.1,

Table 2.1: LifeCLEF Plant Identification Campaign 2014 data set

	Branch	Entire	Flower	Fruit	Leaf (photo)	Leaf (scan)	Stem	Total
Train	1987	6356	13164	3753	7754	11335	3466	47815
Test	731	2983	4559	1184	2058	696	935	13146

The meta data accompanying a photograph includes the location, date, and author information for the captured image. Furthermore, for some images, there is also a user-indicated quality of image parameter.

The campaign has two evaluation schemes: identification based on a single image (image-based) and identification based on several images of a plant (observation-based). In the first scheme, the input is a single image upon which a system should make its decision. In the second scheme, the decision is based on multiple photographs of an *individual plant* that is identified by a field (plantid) in the meta-data. This scheme allows a system to base its decision on a richer information and is still realistic as a user application.

In each case, the performance is measured as the average inverse rank of the correct species, in the returned list. Furthermore, each category is evaluated separately, to benchmark the progress in separate areas.

The flower subset contains images of flower view types and is used for evaluation and comparisons of retrieval systems in the thesis. This data set has 13,164 train and 4,559 test images with 483 plant species, as indicated in Table 2.1. Some examples from the dataset are shown in Figure 2.2

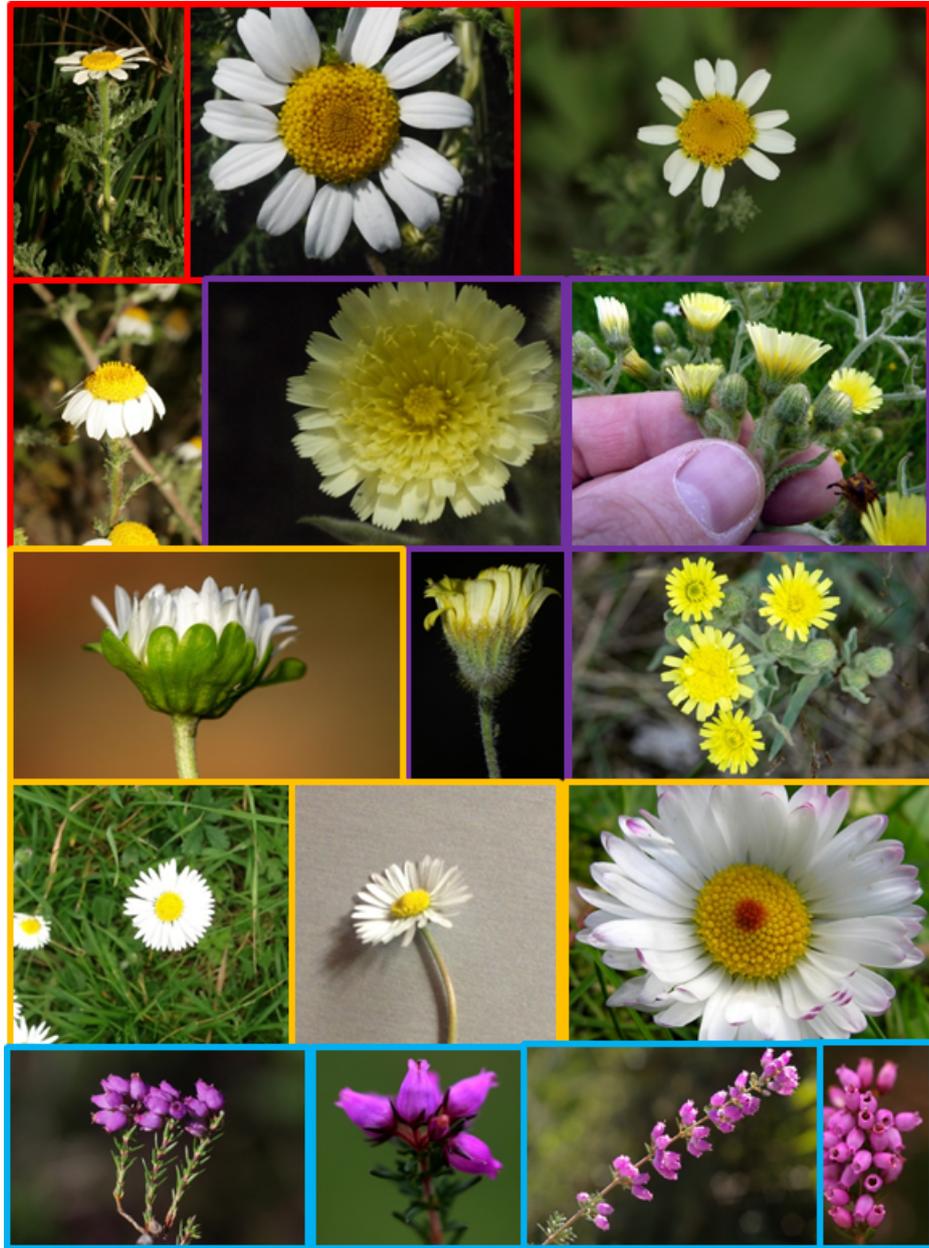


Figure 2.2: LifeCLEF 2014 Flower Data Set

Chapter 3

Classification with Local Invariants

In this chapter, we explain the base system, called *System I* along with its components, depicted in Figure 3.1.

The work flow of System I is as follows. First, Dense SIFT [4] [2] descriptors (features) are extracted from images. Secondly, a clustering and quantization step is used to build a dictionary of visual words from the observed features. The images are then described by the Bag of Words (*BoW*) [13] representation, consisting of a histogram of the chosen visual words. Finally a Support Vector Machine (*SVM*) [14] [15] [16] is trained to classify images according to their BoW histogram representation.

Each of these methods are explained in this chapter in order: The SIFT and Dense SIFT descriptors are explained in Sections 3.1-3.2; BoW method is described in Section 3.3; and SVMs are described in Section 3.4.

3.1 Scale Invariant Feature Transform (SIFT)

Scale Invariant Feature Transform (*SIFT*) is an widely used image descriptor for point based image matching researches since 1999 [4] [2]. SIFT descriptor are also used for point based matching in the computer vision. The main reasons of using SIFT descriptors is being invariant to translations, rotations and scaling in the spatial domain.

The SIFT method starts by detecting interest points from a gray-level image.

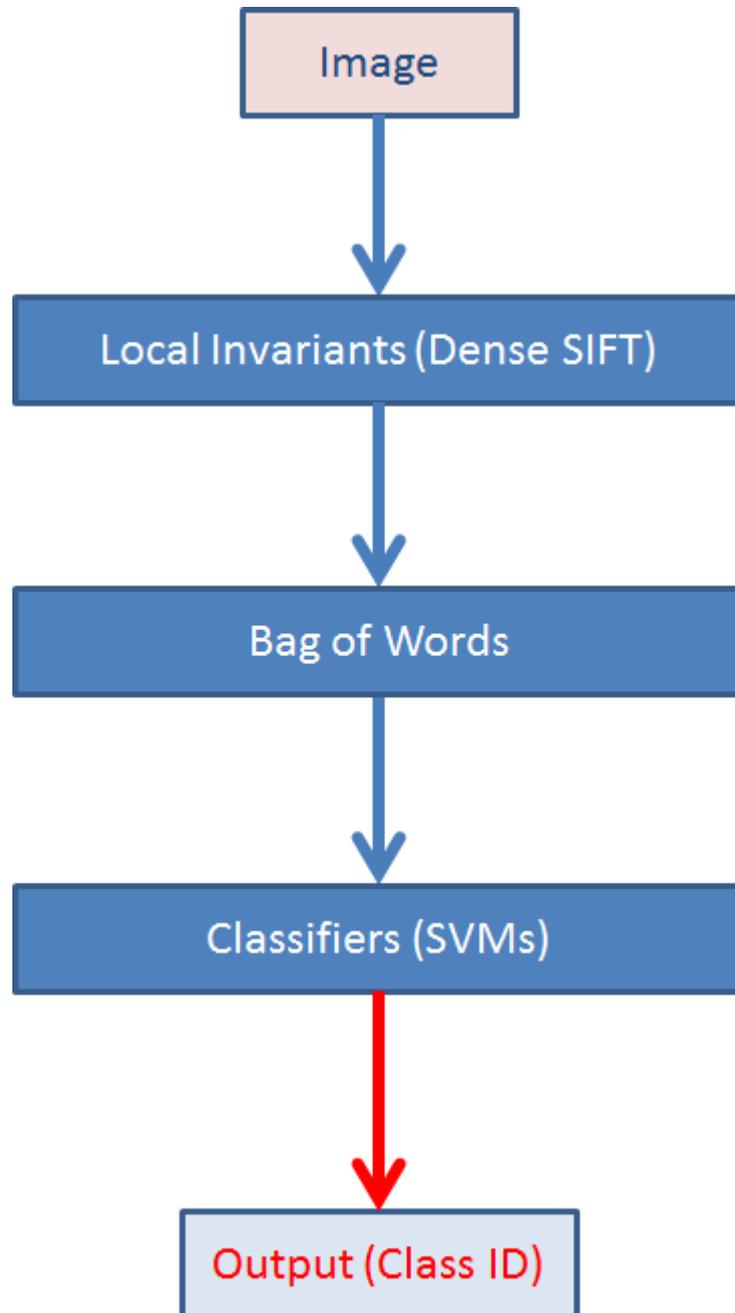


Figure 3.1: System I

The points are found as the local extrema of the difference of Gaussians in the scale-space. The descriptors are in turn extracted from these points, describing the local gradient information in a rotation-invariant manner and are used for matching corresponding interest points between different images. The SIFT descriptor has also been enriched for color images with various color features from spatial domain.

In this thesis, we use the method called the Dense SIFT approach where the interest point detection is bypassed and the SIFT descriptors are extracted at densely placed grids. This approach is shown to lead to better performance in some tasks in object categorization and texture classification [17] [18].

3.1.1 Interest Point Detection

Interest point detection is the most distinctive phase within various types of SIFT descriptors. This variety is generally originated from variation on the interest point detection method, such as the original SIFT descriptor by Lowe [4], Hessian Affine detector with SIFT descriptor [19] and Dense SIFT [17].

In the original SIFT descriptors by Lowe, interest points are computed from scale-space extrema of differences of Gaussians (*DoG*). At first A Gaussian pyramid is computed from gray-level images.

$$G(x, y; s) = \frac{1}{2\pi s} e^{-(x^2+y^2)/(2s)} \quad (3.1)$$

Secondly, difference of the adjacent levels in the Gaussian pyramid is computed, which is referred as DoG.

$$DOG(x, y; s) = L(x, y; s + \Delta s) - L(x, y; s) \approx \frac{\Delta s}{2} \nabla^2 L(x, y; s) \quad (3.2)$$

Finally, interest points are detected as extrema of these differences.

3.1.2 SIFT Descriptors

After interest point detection, the SIFT descriptor proposed by Lowe (1999, 2004) is computed from each interest point. The SIFT descriptor is illustrated in Fig. 3.3 and how the descriptor is computed is summarized below.

A rectangular grid is centered at the interest point with the given scale and orientation these are adjusted for the interest point. From the experiments of Lowe,

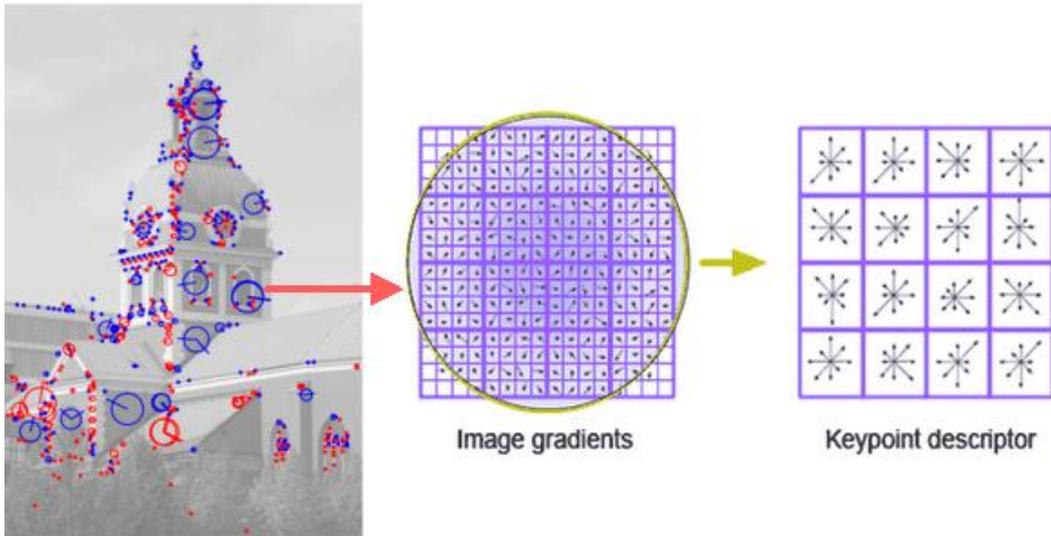


Figure 3.2: SIFT Keypoints [1]

4×4 grid size is thought to be a good size. After these, the local gradient direction with the scale of each interest point is computed as

$$\arg \nabla L = \text{atan2}(L_y, L_x) \quad (3.3)$$

for each point on this grid. Then, the gradient directions are coded within 8 discrete directions and histograms of each gradient directions are computed. In computing these histograms, each gradient direction is also weighted by the gradient magnitude, to give greater weights for image points:

$$|\nabla L| = \sqrt{L_x^2 + L_y^2} \quad (3.4)$$

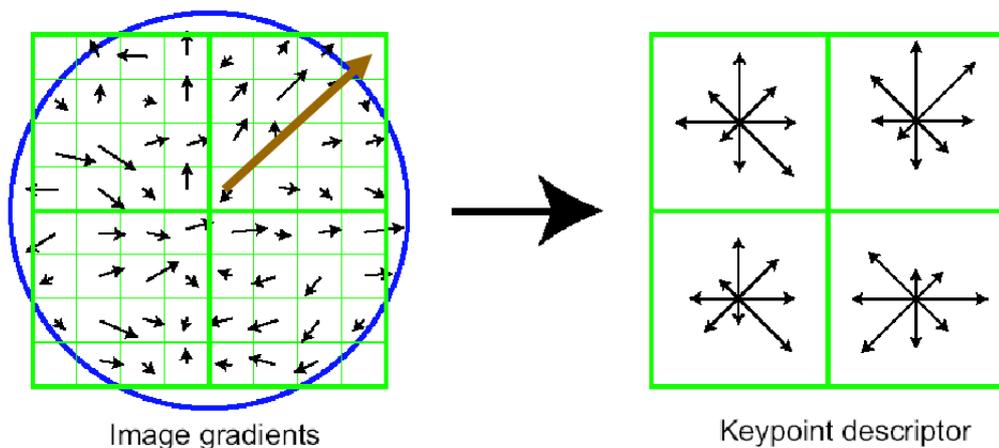


Figure 3.3: SIFT Descriptors [2]

The local histograms are calculated for all 4×4 grid points with these 8 discrete directions. These 8 discrete directions are combined for each interest points to extract $4 \times 4 \times 8 = 128$ dimensions image descriptor. This obtained image descriptor is named as the SIFT descriptor.

3.2 Dense SIFT

The Dense SIFT approach is directly derived from the SIFT approach. It differs from the SIFT approach in that the image descriptors are extracted from densely sampled grids over the image, rather than only at interest points. In fact, there is no interest point detection process in dense SIFT approach.

Recent researches such as [17] and [18] show that while applying the SIFT descriptor for object category classification or scene classification problems, the result on SIFT descriptors over dense grids in the image domain often provides better classification accuracies rather than computing the SIFT descriptor from interest points as obtained by an interest point operator. In other words, larger set of local image descriptors computed over dense grids often provide more information than corresponding descriptors extracted from limited number points evaluated by interest point operator.

This approach or improvement of SIFT descriptors was firstly proposed by Bosch et al. [20][21] and at the present time it is one of the state-of-the-art approach for image based object classification.

The most substantial reason of preferring Dense SIFT descriptors instead of SIFT descriptors is that with using Dense SIFT descriptors, the influence of texture characteristic increases compared to the influence of shape characteristics. When we consider our plant identification problem, the texture is often as important as the shape, which is why we preferred this approach.

3.2.1 SIFT Descriptors

In Dense SIFT approach, the descriptor extraction process is similar with SIFT descriptor extraction. The descriptors are extracted from the center points of grids, one per grid.

3.2.2 Effect of Grid Sizes

The size of these dense grids have crucial impact on the success of this approach: too big ones would loose detail and too small ones may not capture enough context. For this reason, we have evaluated different grid size alternatives to have optimal classification accuracy on our plant identification problem.

The size of plant images varies between 480 to 960 pixels both on vertical and horizontal dimensions. The size of the object of interest also varies among images: some images zoom on the petals of a flower, while others show the flower in a garden with leaves, stem, sky, etc. When we consider these different types of images, if we draw boxes around flowers, sizes of these imaginary flower boxes will occupy an area from 70x70 to 400x400 pixel in raw images. However our dense grids must be large enough to catch descriptive parts of these various sized image flowers.

After several observation over flower images, approximately 30x30 sized grids seem enough to carry informative parts of images. Therefore, we decided four types of dense grids size to evaluate effectiveness of sizes of dense grids. These are 16x16, 32x32, 48x48 and 64x64 pixel sized dense grids decided to work, Figure 3.5. Using all of them probably makes our carrying information potential much more stronger, however it computationally costly, and so choosing only two of them are enough for identification. As a result of several experiment with these parameters, we decide work with 16x16 and 32x32 pixel grid sizes.

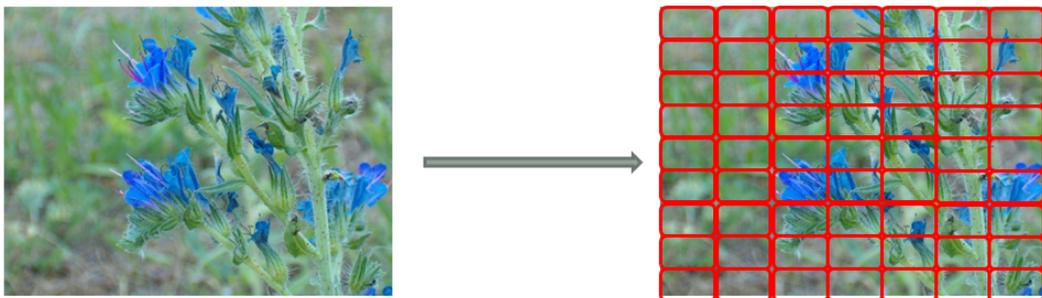


Figure 3.4: Dense SIFT

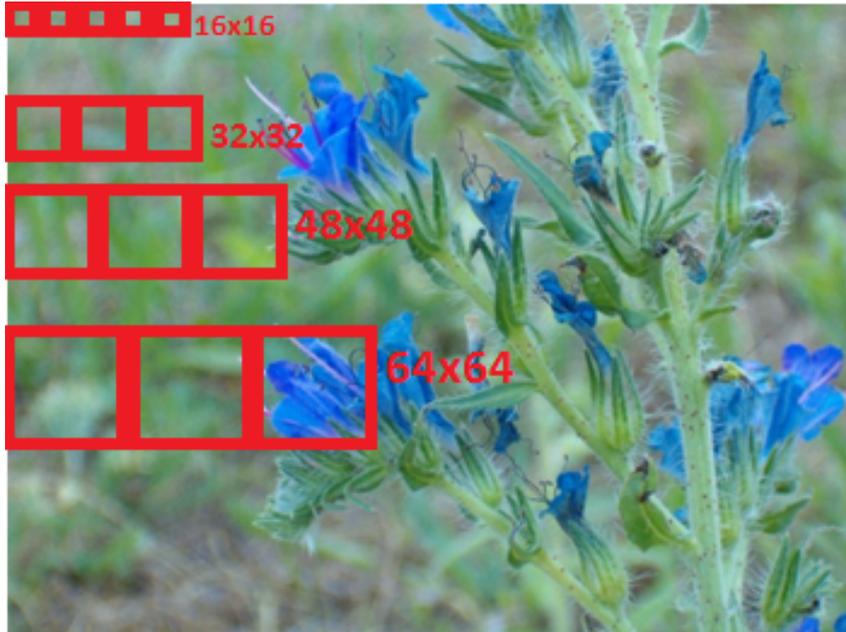


Figure 3.5: Dense Grid Sizes

3.3 Bag of Words Representation

The Bag of Words (BoW) method aims to summarize the contents of a document or image as an unordered set (bag) of primitive elements (words or visual words), regardless of the position of those elements in the whole document or image. In our case, the 128 dimensional SIFT descriptor space is clustered and each descriptor is denoted as belonging to one cluster. These enumerated clusters are the visual words [24] [25].

The BoW method starts by creating clusters from descriptors. We use the K-Means Clustering method for determination of clusters. Once the clusters are identified, we have what is called a *dictionary* or *codebook* of visual words. The descriptors of a given image can then be mapped to the nearest clusters, which is accomplished efficiently in our case using the KD-tree ([22]). This operation is called *coding*, in literature. The KD-tree is a multi-dimensional binary search tree that aims to represent the hierarchical clusters within data and thus reduces the match process between a descriptor and all the visual words, from linear to logarithmic.

The BoW method is then complete by computing a histogram of these visual words, *pooling* the information across the whole image. This approach aims to obtain a representation of the image that is robust towards spatial variations, while it also

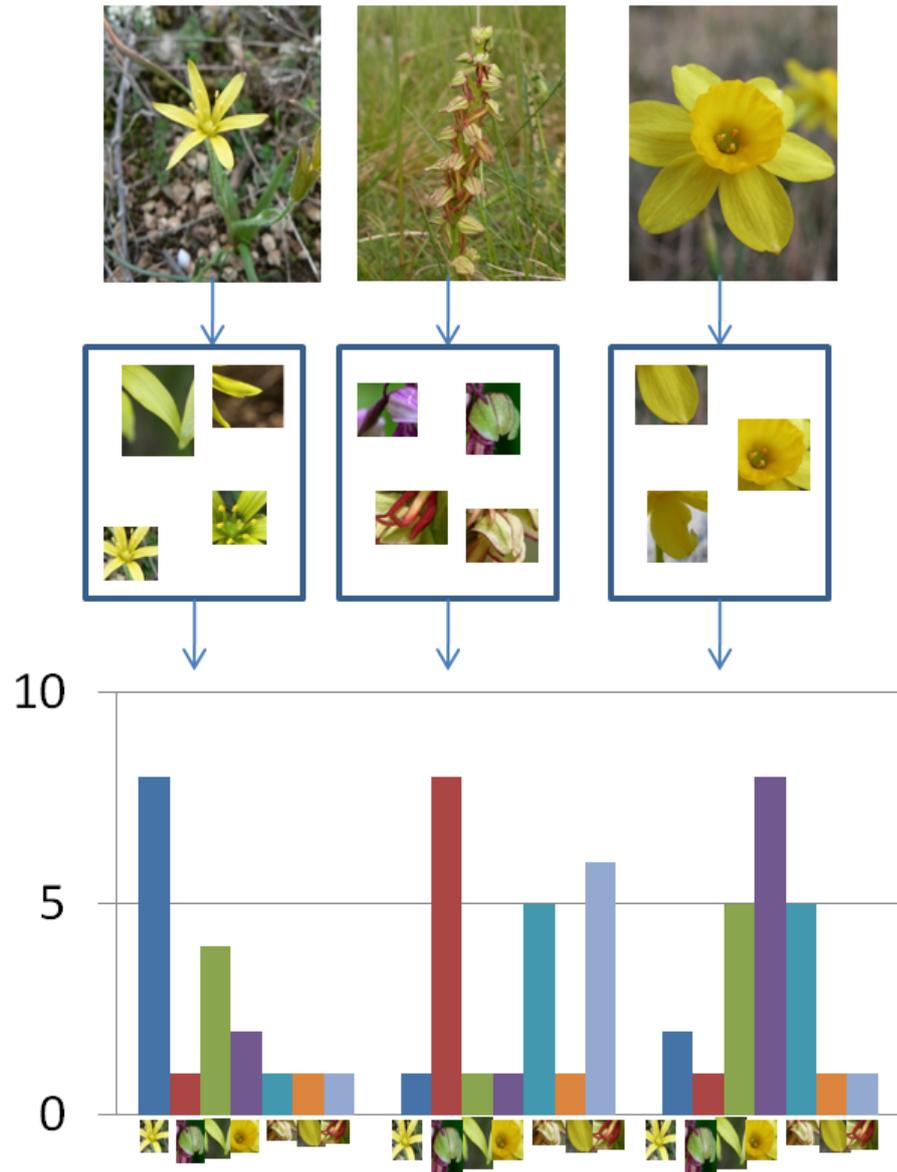


Figure 3.6: Word Histogram

makes it unlikely to discriminate between objects that differ only by their spatial arrangements. Figure 3.6 illustrates this method. These operations are called *coding* and *pooling* respectively.

3.3.1 Size of the Dictionary

The size of the visual words is a crucial parameter of the representation. A large number of visual words help better representation, however the resulting histogram may not properly cluster all the variations observed in a particular visual word. When we consider Figure 3.6, this can be understood easily. If we use 5 visual words instead of 7 for this three objects, green and blue words probably evaluated as same word. Then our histogram is weakened, and so identification accuracy can be reduced.

On the other hand, if we use large sizes of visual words, similar words can be coded with different visual words. Another disadvantage of using large visual words sizes is that some less frequently seen descriptor, such as descriptors from leaves, stems, ground, sky, etc. can be coded with new visual word.

Lazebnik et al. [25] report trying different word sizes or code-book sizes and argue that their spatial pyramid matching scheme is most effective when they choose visual word size as 200. We made several experiments to decide on the optimal word size for plant classifiers. These are 1200 visual words for plant classifier of System I, and 800 for each plant classifiers of System II and Improved System II.

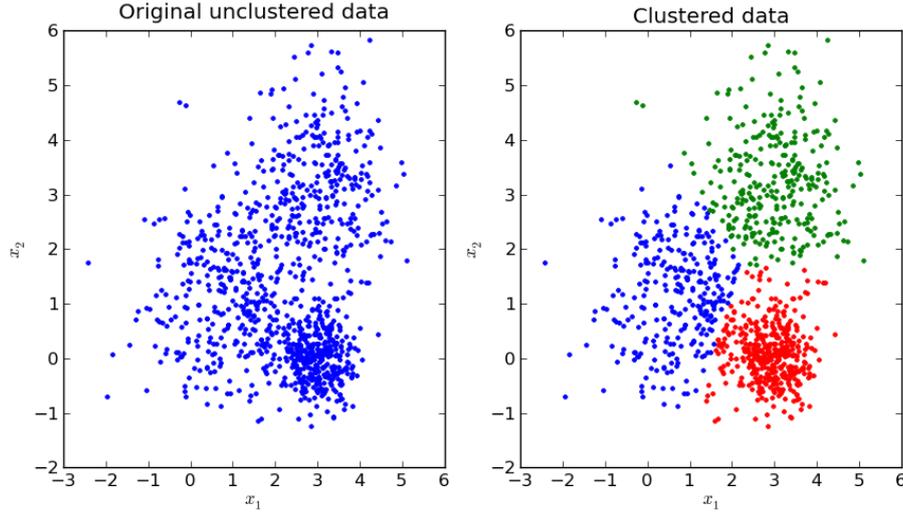


Figure 3.7: K-Means Clusters

3.3.2 K-Means Clustering

Clustering algorithms are traditionally viewed as unsupervised method for data analysis. The algorithm is presented with a set of data instances which has to be grouped according to some similarity notion. The algorithm has only access to set of features for each object, and no labels.

The K-Means clustering algorithm is first proposed by MacQuenn in 1967 [23] and it is commonly used to automatically partition a data set into k groups. It proceeds by randomly selecting k initial cluster centers and thereafter iteratively refining them as follows:

1. Each object is assigned to its closest cluster center.
2. After that each cluster center is updated to be the mean of its constituent instances.
3. Repeat until no change.

In our work, we initialize the cluster centers using randomly chosen instances from Dense SIFT descriptors. The clusters are created by considering a distance metric, such as the L^1 or L^2 norms, between Dense SIFT descriptors. We use 128 dimensional features and the L^2 norm, as it is thought to give a more robust measurement. An illustration of the input and output of the algorithm is given in Fig. 3.7.

The crucial factor on this method is deciding on the number of clusters, k . In

most cases, optimal amount of clusters are already known, therefore k can be set directly. In our case, the optimal number of clusters is unknown and is decided empirically.

Once we have a fixed-length representation of a given image, in terms of the observed frequencies of visual words in the dictionary, we use Support Vector Machines to classify the plant in the image, given its histogram.

3.4 Support Vector Machines (SVMs)

Support Vector Machines (*SVMs*) is state of the art classification technique in machine learning [24] [25]. It creates sets of hyperplanes in a high dimensional feature space. It is widely used for classification, regression and for other tasks. Intuitively SVM creates vectors to separate feature space by considering larger distance margin between hyperplanes. It requires the solution of the following optimization problem to create these vectors:

$$\min_{w,b,\xi} \frac{1}{2}w^T w + C \sum_{i=1}^l \xi_i \quad (3.5)$$

subject to,

$$y_i(w^T \phi(x_i) + b) \geq 1 - \xi_i,$$
$$\xi_i \geq 0$$

Solving this problem is a hard optimization problem and there are some libraries to cope with it and create classifiers. Therefore we use one of mostly preferred library that is LIBSVM.

LIBSVM has gained wide popularity in machine learning and many other classification areas. This library has been actively developed by Chih-Chung Chang and Chih-Jen Lin since the year 2000 [16]. It basically works as training with labeled train data set and learning classifier vectors with them. After that test data set is evaluated with these learned vectors. This is a supervised learning model.

There are some variations on vector types mainly categorised as linear and non-linear *SVMs*. Linear *SVMs* are very fast for training, however it is also limited to use an inner product to classify descriptors. Non-linear *SVMs* can generally obtain much better results by pre-transforming the data with homkernmap which computes an explicit feature map. In the lights of these facts we are using non-linear *SVMs*. Non-linear *SVMs* are also divided into several types and Radial Basis Function (*RBF*) is one of them and that is our preferred non-linear *SVM* classifier.

In the *RBF* model, the training vectors x_i are mapped into a higher dimensional space by the function ϕ . *SVMs* generates a linear separating hyperplane with the maximal margin in this higher dimensional space. $C \geq 0$ is the penalty parameter

of the error term. We are trying to use the optimal parameter C to train the whole training set.

$$K(x_i, x_j) = \phi(x_i)^T \phi(x_j) \quad (3.6)$$

K is called the kernel function for SVM and the kernel function for *RBF* is this;

$$K(x_i, x_j) = \exp(-\gamma \|x_i - x_j\|^2), \gamma > 0 \quad (3.7)$$

Here γ is kernel parameter for *RBF* which is widely used to optimize classifier vector.

Radial Basis Function (*RBF*) is a reasonable choice. The first reason is that this type of kernel is non-linearly mapping objects into a higher dimensional space. Unlike to the linear kernel, *RBF* can handle this type of classification problems, due to nonlinear relationship between class labels and descriptors. The second reason is the number of hyper parameters. This number influences the complexity of model selection. Indeed the polynomial kernel has more hyper parameters than the RBF kernel, these are \mathbf{r} , \mathbf{d} and γ .

$$K(x_i, x_j) = (\gamma x_i^T x_j + r)^d, \gamma \geq 0 \quad (3.8)$$

Equation 3.8 illustrates the kernel function of polynomial SVMs and its complexity.

We are creating a grid search model to decide optimal parameters for *RBF*. Various pairs of (C, γ) values are tried and the one with the optimal cross-validation accuracy is chosen for rest of the project. We found that trying exponentially growing sequences of C and γ is a practical method to decide best parameters. These parameter are

$$C = \{10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}, 10^0, 10^1, 10^2, 10^3, 10^4\} \quad (3.9)$$

and

$$\gamma = \{2^{-4}, 2^{-3}, 2^{-2}, 2^{-1}, 2^0, 2^1, 2^2, 2^3, 2^4\} \quad (3.10)$$

We have trained *RBF* model several times with these parameters to achieve optimal results, and finally we have decide to use more than one C parameters for different **SVMs** classifier. There is one *SVMs* classifier for *System I*, however we are using five *SVMs* for *System II* and *Improved System II*. γ is adjusted as 2^{-2} for all systems, however C is adjusted as 10 for *System I* and 100 for others.

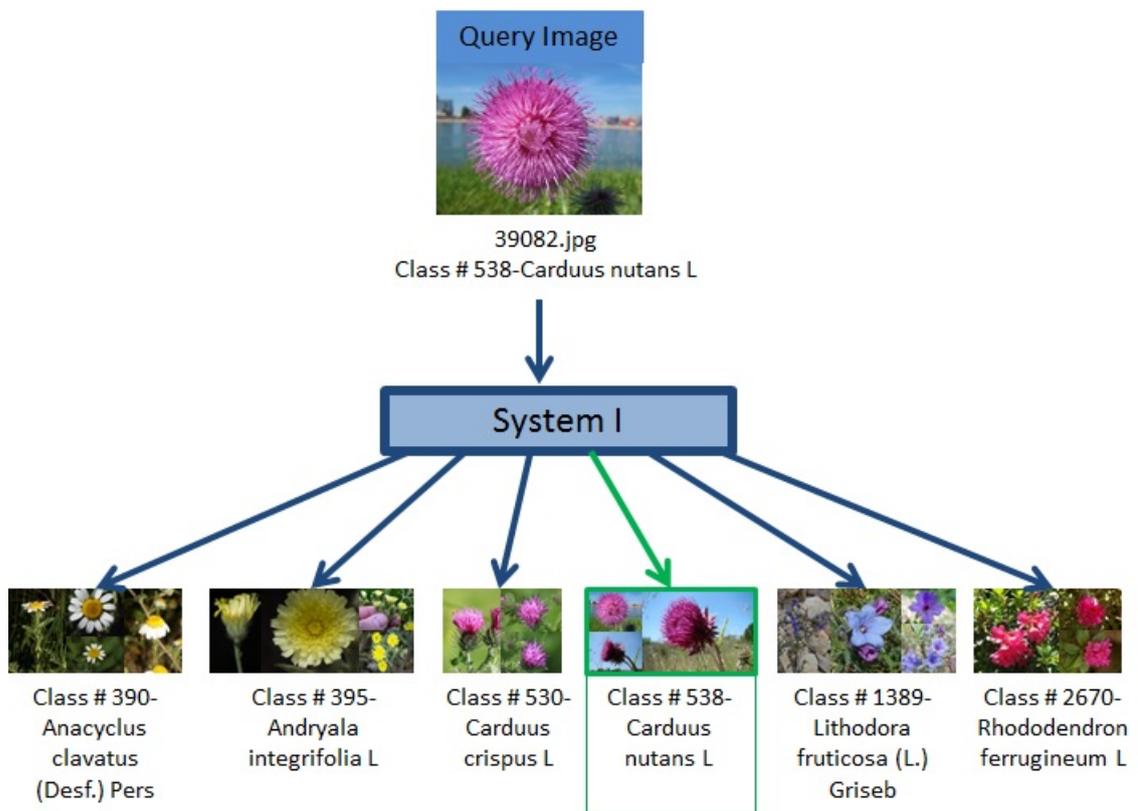


Figure 3.8: System I Example

Chapter 4

Color Based Classification

4.1 Outline

This is the second system, named as **System II**. This system is constructed by adding a color classification step before **System I**, with the goal of dividing the flower classification into smaller cases, according to the main color character of flowers.

The color classification is done using the Saturation Weighted Hue Histogram (*SWHH*) of raw images. After the color classification, where the input image is assigned into one of the 6 categories, the rest of the process flows just as in System I, with *only* plant species in that color cluster. Figure 4.1 illustrates System II.

There are two main reasons for color based classification stage. The first motivation is to reduce the number of flower classes, to facilitate the job of System I. There are approximately 500 flower species with very close flower textures or shapes, while after color clustering, each second-stage classifier deals with 50 - 200 species only. The second reason is necessity of much more efficient use of color features. We wanted to try the exploit the use of color information, as an alternative to extending the Dense SIFT with color [18].

We elaborate on these issues in Section 4.2.

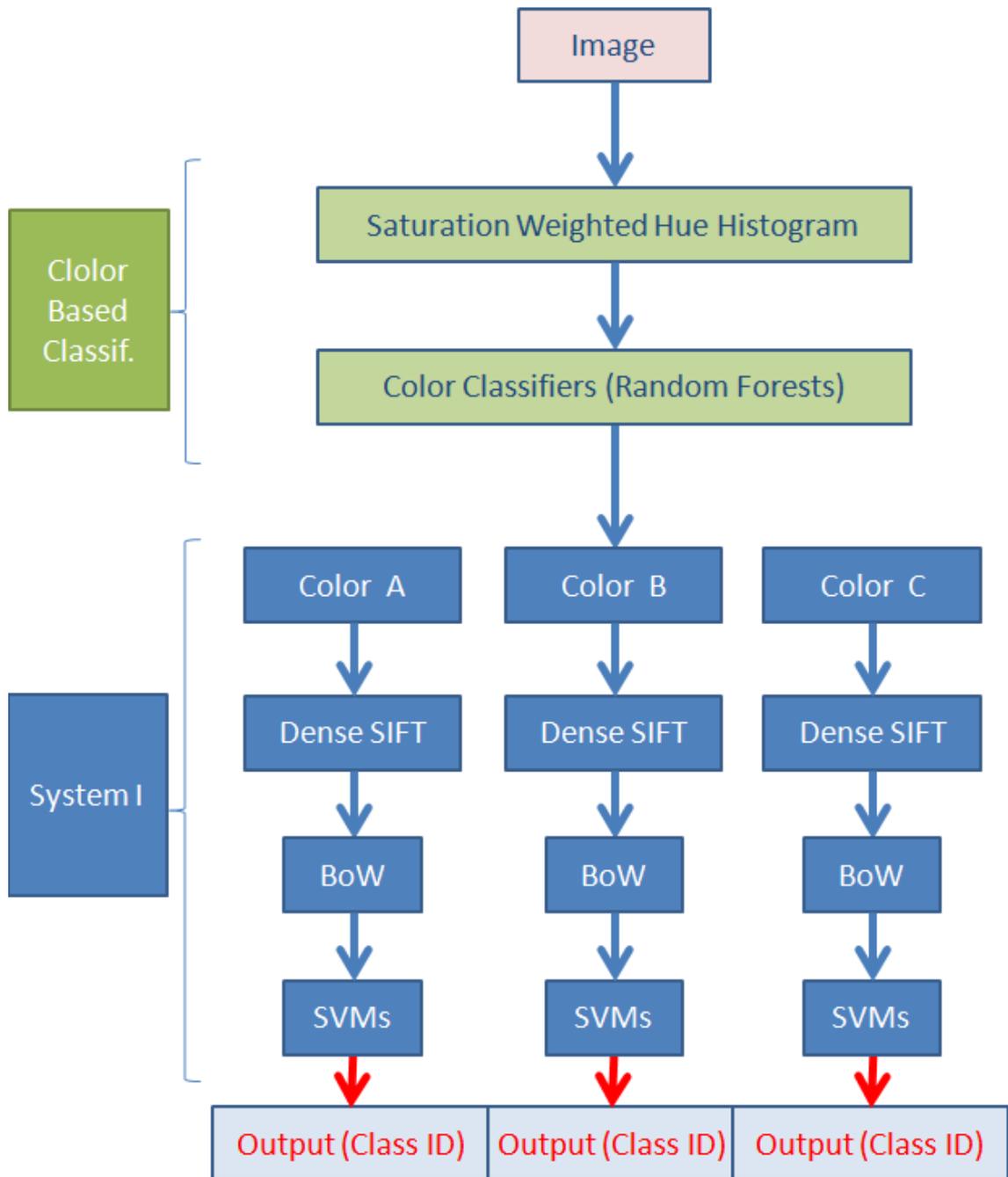


Figure 4.1: System II

4.2 Flower Taxonomy

At the top of flower genetic map comprising the LifeClef2014 flower set, there are 89 flower families which involves 305 kinds of flower genres in total, and at the bottom of the genetic map, these genres are divided into 483 types of flower species, as shown in Figure 4.2.

As can be seen in this figure, while similarity within a flower family can be low, similarities within the flower genus can be very high. In particular, flowers within a genus are often similar in their petals, carpels, and stamens, while they differ mainly in color. For example the flower genus, *Narcissus*, has four flower species: *Narcissus assoanus* Dufour (# 5806), *Narcissus dubius* Gouan (# 5810), *Narcissus poeticus* L. (# 5817) and *Narcissus pseudonarcissus* L. (# 5819). As they are from same flower genus, the shapes of their petals and texture characteristics are similar, while their main differences are in the color of their petals and stamens. The significance of color features thus becomes indispensable for the fine grained identification of flower species.

Indeed, when we consider results from identification with *System I*, these four species of *Narcissus* are misclassified into similar species under the same genus. By processing with color based classification step, they are clustered at different color clusters (one of them is classified as red color cluster, while others are classified into the white and yellow cluster). As each cluster is handled by separate plant identification systems, they each have to deal with fewer number of classes.

Color can also be used alongside the SIFT descriptors, rather than in a pre-classification step. Indeed, there is an extension of SIFT features with color. However, as some researchers point out (e.g. Wengert et al. [26]), this may not result in the best way of combining these three information sources (texture, shape and color).

As a simultaneous solution to both problems (large number of classes and incorporating color), we have decided to use new color descriptors and classifier stage. In other words, we use the color information at first clustering (classification) stage, after that Dense SIFT descriptors from gray-scale images are used for second (final) classification step. Therefore, we have fewer number of species for each Dense SIFT operation tasks and color and shape/texture differences between flowers with similar

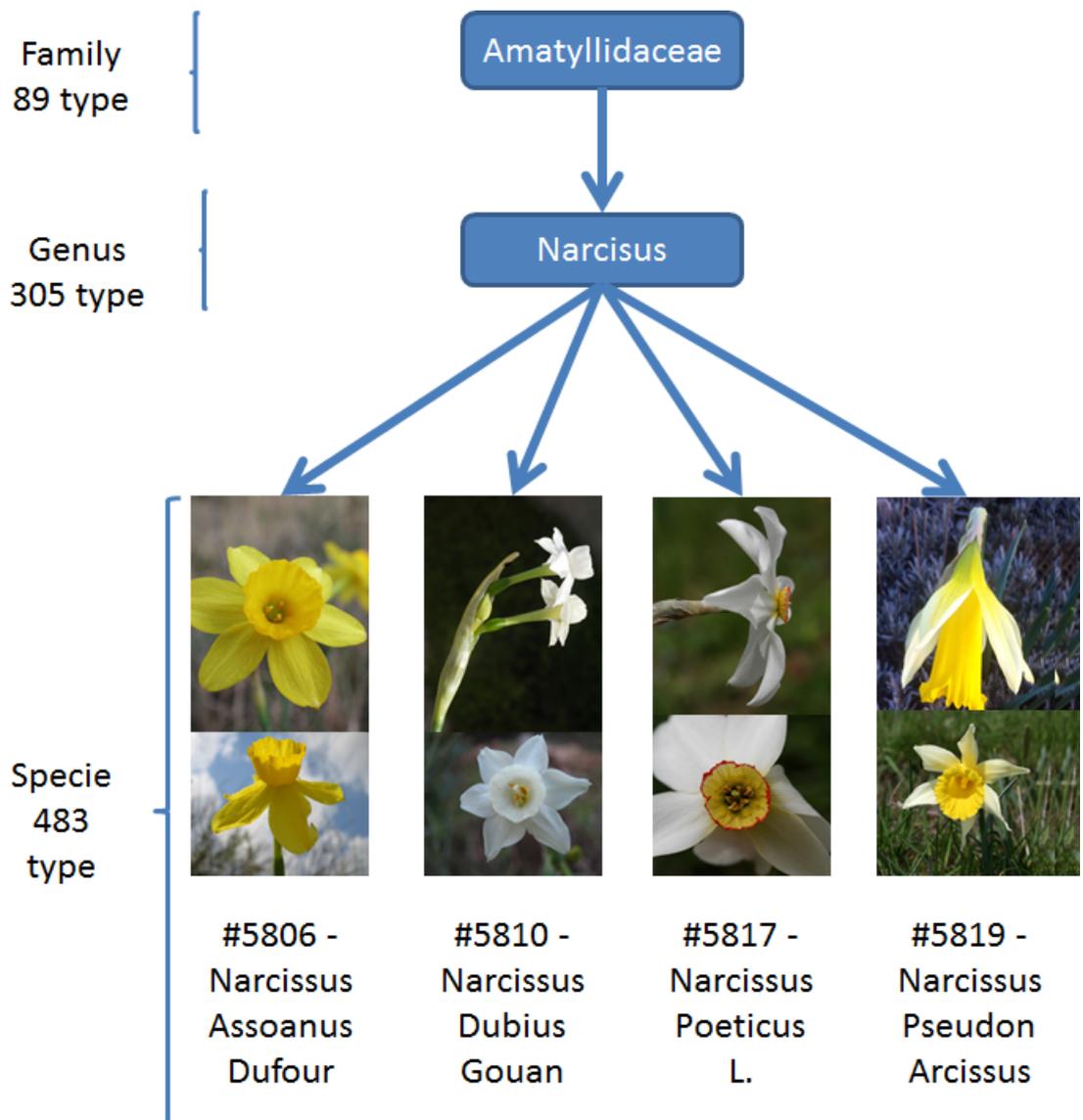


Figure 4.2: "Narcissus" Flower Family

shape, texture characteristics have become much more recognizable.

4.3 Color Descriptors

As color descriptors, we have considered Color Moments (*CM*) and Saturation Weighted Hue Histograms (*SWHH*) based on our prior experience in plant identification [27] [28] [29]. After several experiments with these descriptors, such as CM descriptors from the whole size of images, dense grid of images, SWHH from the whole size of images, we preferred to use SWHH from the whole image, as the chosen color descriptor. Comparison of these experiments and results of them are discussed in detail at Section 4.6. As a classifier of this stage, we have decided to use Random Forest classifier due to successful results on similar problems.

4.3.1 Saturation Weighted Hue Histogram

We have selected the *Saturation-Weighted Hue Histogram* due to its desirable characteristics such as matching the human color clustering better than RGB statistics.

Hue, Luminance, and Saturation (HSL) form a 3D-polar coordinate color space which is an alternative representation to the RGB space. This representation is constructed by first placing an axis between the origin and the point $[R_{max}, G_{max}, B_{max}]$ in the RGB space, defining the *achromatic axis*. As shown in Figure 4.3, the color space is fully described by these terms:

- **Brightness or Luminance** $L \in [0, 1]$: gives the position on the achromatic axis.
- **Hue** $H \in [0^\circ, 360^\circ]$: is an angular measure around the achromatic axis with respect to an origin at pure red.
- **Saturation** $S \in [0, 1]$: the distance from the achromatic axis.

There are different color space models whose are defined with these three components of the HSL color space. We are using the *Improved Hue, Luminance and Saturation (IHLS)* space that is one of these space models [30]. The main reason



Figure 4.3: Achromatic Axis

of using this color space is related to the importance of the hue component in our problem.

4.3.2 Saturation Weighted Hue Statistics

We used saturation weighted hue statistics due to determinant attribute of hue values on flower images. The statistics are calculated with hue values that are weighted by their corresponding saturation values. The main reason of weighting the hue value is for compensating the weakness of hue statistics for weakly saturated colors (gray-values).

Our calculation of the statistics are based on Allan Hanbury's calculations [31].. In that work, Hanbury argued about some disadvantages being due to the correlation between hue and saturation values. This correlation is taken care of by this weighting operation.

4.3.3 SWH Histogram Extraction

Hue histograms are usually used as image feature to classify chromatic images. Achromatic and near-achromatic pixels are mostly eliminated in these histograms, because in these pixels the hue term is fuzzy. Conversely, the saturation term is ineffective in classification of achromatic and chromatic colors. Therefore we can derive a weight for differentiating between chromatic and achromatic colors by combining these histograms. In other words, higher saturated pixels which are the more colourful can have higher weighting in the hue histogram than lower saturated the less colourful pixes.



Figure 4.4: (a) Colour image. (b) Hue of image (a). (c) Saturation of image (a). (d) Luminance of image (a).

The saturation-weighted hue histogram is calculated with equation 4.1,

$$W_\theta = \sum_x S_x \delta_{\theta H_x} \quad (4.1)$$

where x is a pixel in the image. H_x and S_x are the hue and saturation term at a point x and δ_{ij} is the Kronecker delta function.

When we compare the hue histogram of image in Figure 4.4.b and the saturation-weighted hue histogram in Figure 4.4.c, we can see main differences on low saturated areas, these are black pixels in the Figure 4.4.c. This scattered hue histogram can not meet with the expectation of representation for color differences. The resulting color varies from red, through yellow, green, cyan, blue, magenta, back to red and these colors are illustrated with different levels of gray pixels respectively from black to white in the Figure 4.4.b. However the saturation-weighted hue histogram reduces this problem and the amplitudes of the peaks are successfully reduced. So it leads to a more robust histogram.

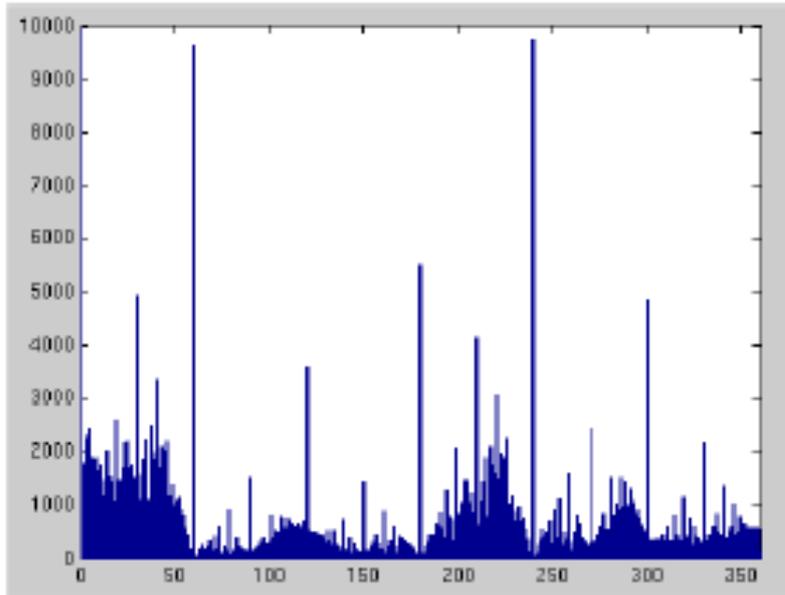


Figure 4.5: Hue Histogram of *Figure 4.4.a*

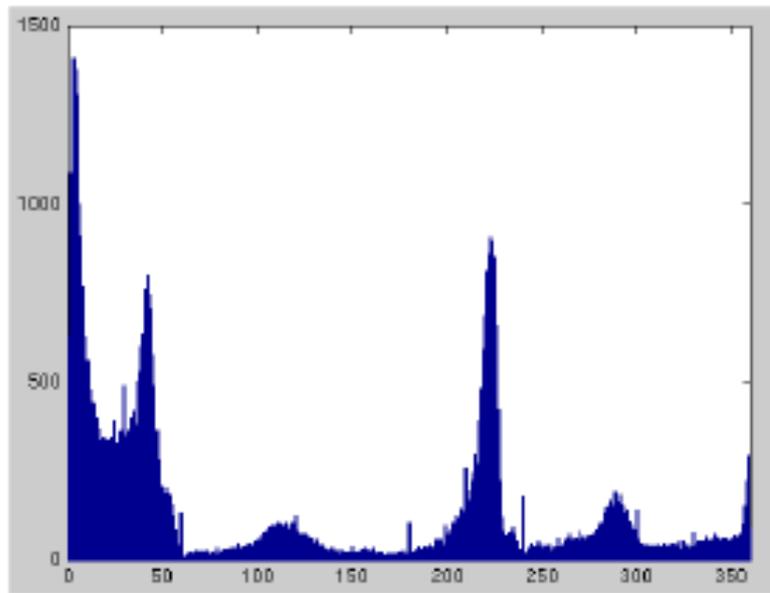


Figure 4.6: Saturation-Weighted Hue Histogram of *Figure 4.4.a*

4.4 Color Clusters

A significant decision for System II is the decision of which color clusters to have. Too many clusters would correspond a more difficult classification task and too few clusters would not really serve its purpose.

While doing this we considered many heuristically color clustering splits. We started with the 9-color clusters scheme with white, yellow, red, blue, purple, green, pink, orange, cyan. However in that case, there were many species classified in 2 or 3 different color clusters, due to the presence of multiple colors in many flowers (e.g. yellow center, pink surround). Therefore we decided combining the most intersected color clusters.

After several empirical color clustering experiments, five main color clusters are determined. These are the least intersecting colors clusters that are created by considering most intersecting color clusters from the previous scheme:

- Red Color Cluster,
- Blue & Purple Color Cluster,
- Pink & Magenta Color Cluster,
- White & Yellow Color Cluster,
- Green Color Cluster.

As can be seen from Table 4.1, color clusters of the training set contains a varying number of 40 to 155 species in each cluster. Both training and test sets contain images and color ground-truths from all flower species within color cluster.

The White & Yellow color cluster which is the largest cluster, includes 155 flower species from among the 483 flower species in the full data set. Some of the species inside this cluster are shown in Figure 4.7. The sizes of the other color clusters are listed in Table 4.1.

Even this color scheme, there are also some intersections among the 5 color clusters. Specifically, while the total number of species for test set is 306, the total of the species in Table 4.1 is 381. For example, a flower species named as *Gagea Granatelli Parl* has 8 flower images shown in Figure 4.8. After color based

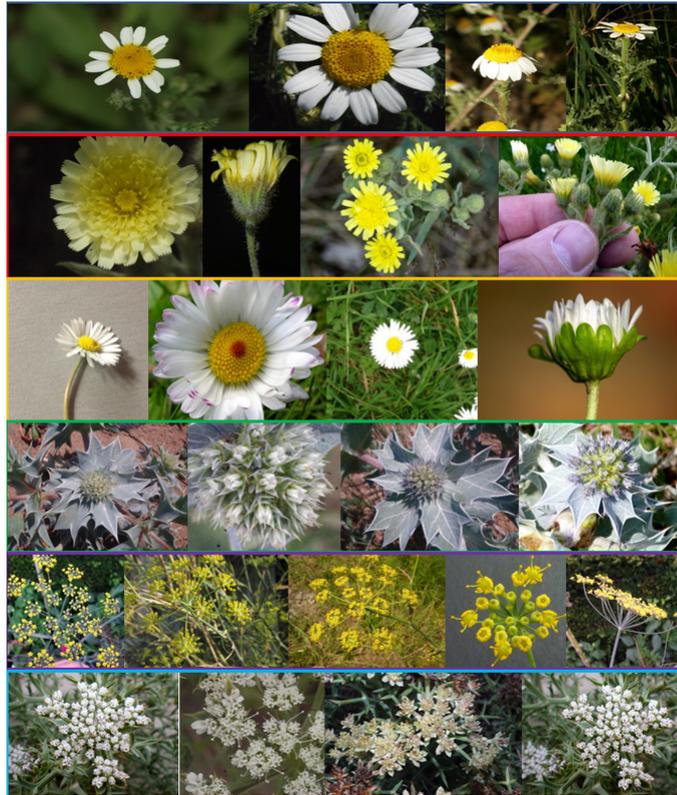


Figure 4.7: White Yellow Color Cluster

classification operation is done, 6 of these are classified as *White & Yellow Color C.* and 2 of them (top left and middle images) are classified as *Green Color C.*. However, we allow for some intersection so as not to have harsh colour boundaries; and this intersection does not necessarily cause a problem in the plant identification phase.

Another example and different type of problem is represented in Figure 4.9 for the flower specie *Orchis Anthropophora*. This flower species is classified into three different clusters during the color clustering. Out of the 28 flower image in this class, 2 of them were classified as *Green C.*; another 2 were classified as *Pink & Magenta C.* and the rest were classified as *White & Yellow C.* This situation is not a misclassification problem apparently, as seen in Figure 4.9, as there green and magenta versions of this species.

Table 4.1: Flower Subset with Ground-truth

	Train Images	Species	Test Images	Species
Red	200	41	100	41
Blue & Purple	400	64	200	57
Pink & Magenta	400	93	200	78
White & Yellow	400	235	200	155
Green	328	50	128	50
Overall	1728	483	828	381

4.5 Classification Using Random Forests

We use a Random Forest classifier, to detect each color cluster. Random Forests are proposed by Breiman [32], and are among the most commonly used ensemble learning techniques. They are known to have good generalization performance.

Plant classifiers for each color clusters are trained with flower species from these color clusters. Color clusters of the training set contains a varying number of 40 to 155 species in each cluster.

Often, one color classifier returns positive and others are negative; while occasionally two color classifiers returns positive and rarely all of them returns negative. We select the final cluster according to the the largest response output by the classifiers, indicating their strength of beliefs.

The run-time performance of the random forest is quite fast. This is the main reason to use Random Forests as color classifier. Training time is around 15 minutes for training set with 2000 images by Weka (data mining software in java) [33]. Testing time is approximately 15 seconds per 100 images.

4.6 System Overview

A sample input/output for System II is shown in Figure 4.10. In brief, initially the query image is classified into the correct color cluster. Then, the plant is



Figure 4.8: Gagea Granatelli Parl

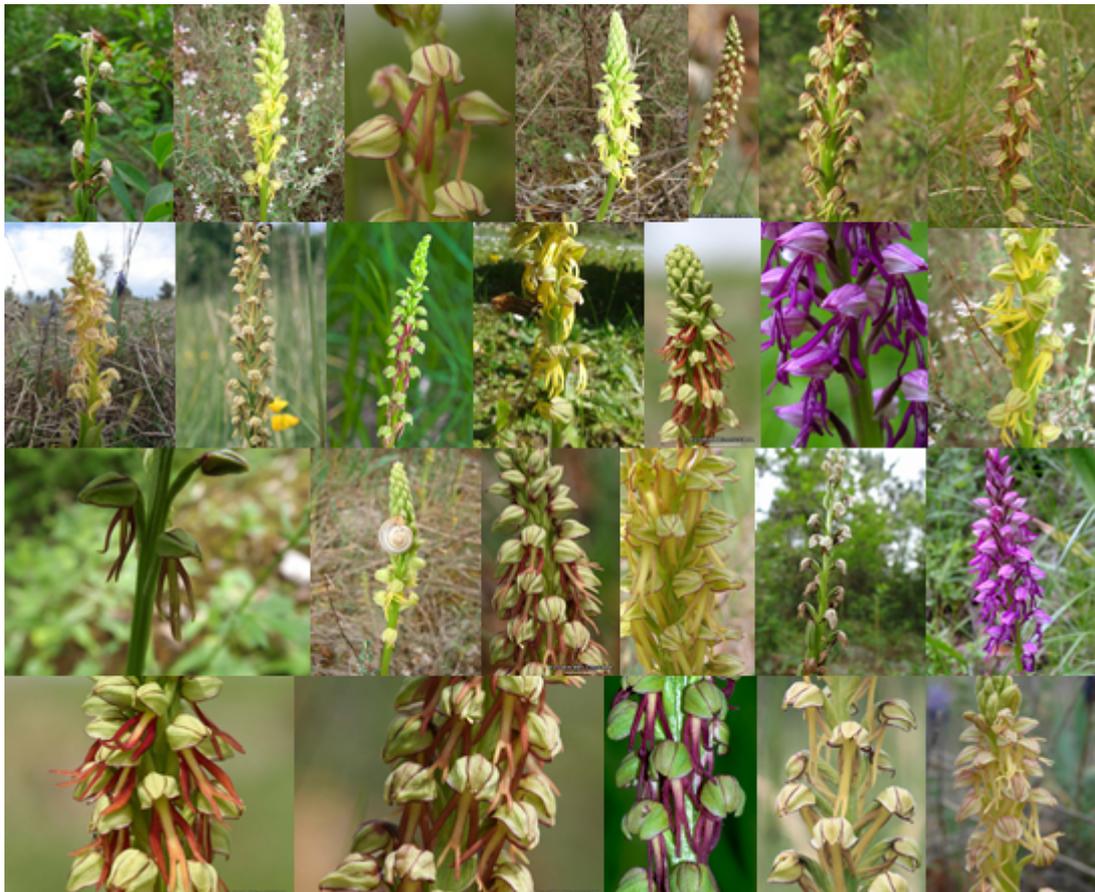


Figure 4.9: Orchis Anthropophora

identified by *System I* that is trained for that color cluster.

In this example, the query image (39082.jpg) is firstly classified by the color classifier into the Pink & Magenta cluster. After that, the identification system of the Pink & Magenta cluster recognizes the image identity as belonging to 538 - *Carduus Nutans* species, where the initial number is just the system assigned ID for that species.

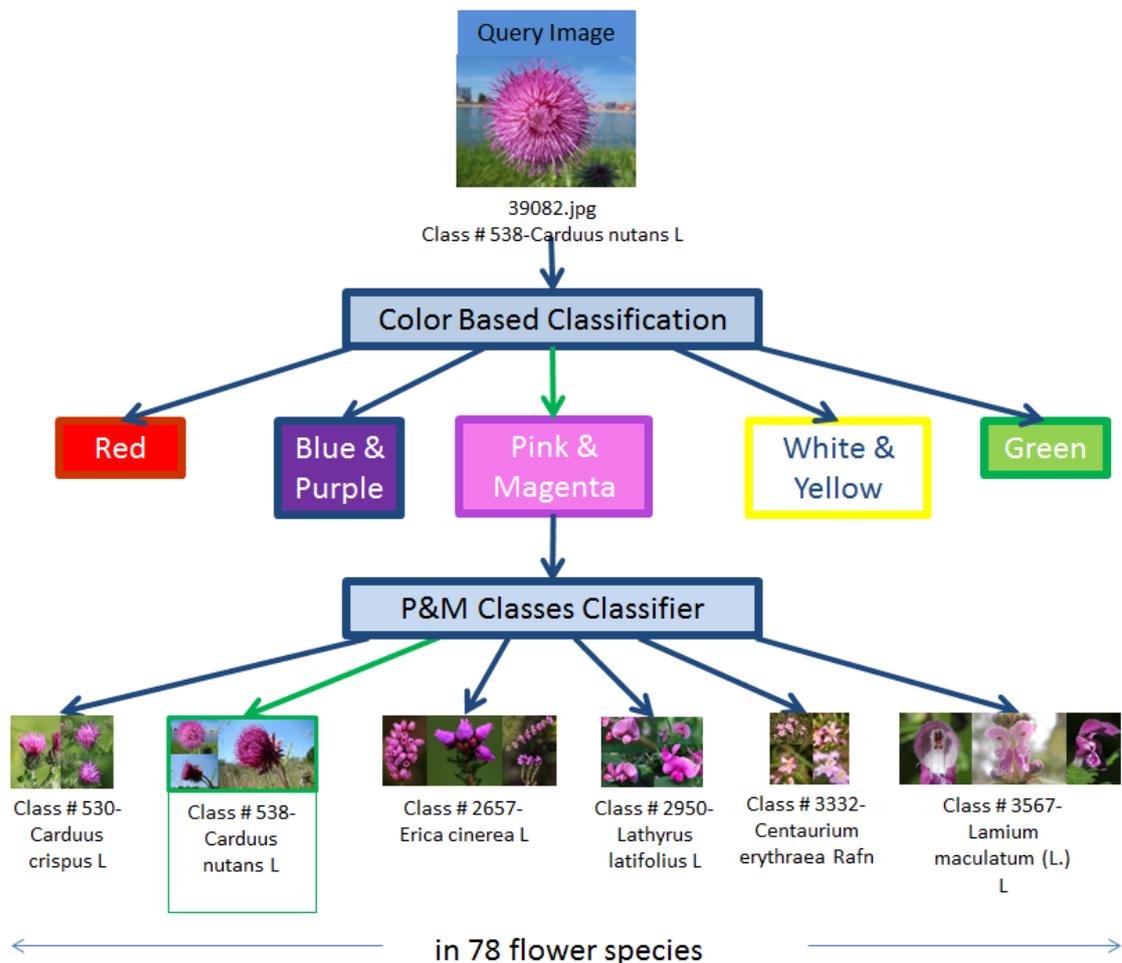


Figure 4.10: System II Example

For testing the color classification step, we use the test set that is labelled manually as to the correct color cluster and measure the Precision and Recall values within each cluster. Detailed analysis about precision and recall results are examined in Chapter 6.

The main disadvantage of this system is that a wrong classification at color based classification step may cause a drastic effect on plant identification accuracy.

However, as we allow some overlap between clusters, the errors are often not drastic. Furthermore, reducing the number of species from 483 to 78 reduces the size of the identification problem for System I considerably, and results in much more accurate plant identification overall.

Chapter 5

Classification with Cropped Images by Saliency Map based ROI Detection

5.1 System Outline

This system is an improved version of **System II**. It is constructed by adding an image cropping (region selection) phase just before System II. The goal by constructing this phase is focusing on characteristic parts of image in addition to getting rid of frequently seen background objects such as leaves, stem, ground, etc by cutting out them.

The most important problem for this preprocess phase is choosing the area with required flower components. In literature it is called as *Region of Interest(ROI) Selection*. After numerous literature reviews related with that problem, we have decided to work out with Saliency Map based ROI selection algorithms, due to the fact that most accurate results are gathered with this method during our experiments on flower data set.

In this model firstly saliency map from raw images is extracted. Secondly band-pass and Gaussian filters are applied on saliency map of images to get rid of noise, artifacts and negligible pieces of flower. Finally a smart cropping process is applied onto raw images by using these filtered maps. As shown in Figure 5.1, this process is applied before *System II*. Rest of the processes are the same as work flow of *System*

II. The main difference between *Improved System II* and *System II* is processing on saliency based Region of Interests cropped images.

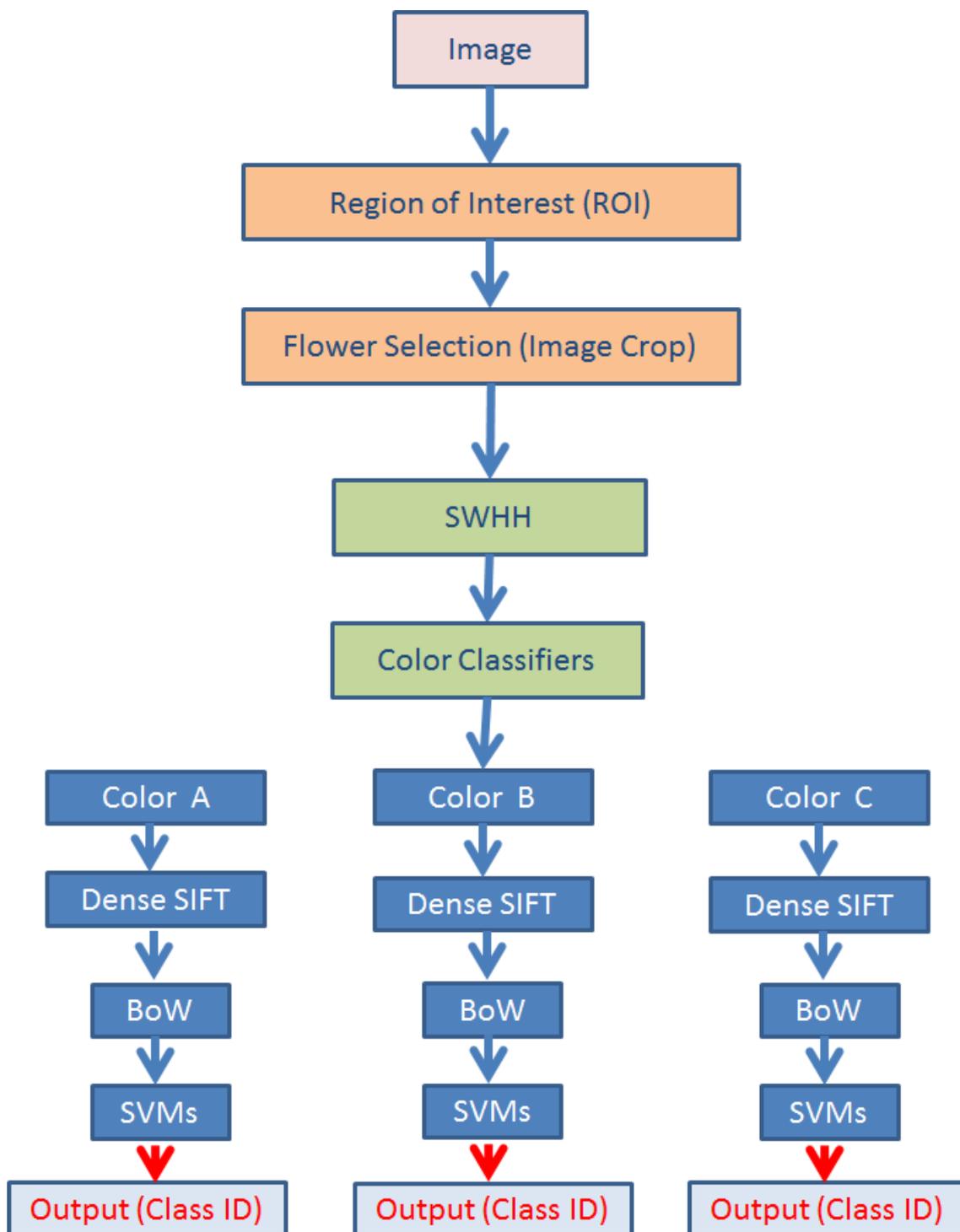


Figure 5.1: Improved System II

5.2 Region of Interest (ROI) (Saliency Map)

There are several approaches for extracting Saliency Map in the literature. We preferred to evaluate a CVPR work by Achanta et al titled as "*Frequency-tuned Salient Region Detection*" [3]. The reason why this approach is favoured is that it is easy to implement and computationally fast, robust and accurate. In addition, this approach benefits from features of color and luminance.

The goal of this approach is to compute the degree of saliency of each pixel with respect to its neighbourhood. Substantial components of this approach are color and lightness properties with center versus surround approach. In the literature, most of the saliency map extraction approach utilize the similar center versus surround method. The crucial point in this method is the size of the neighborhood used for computing saliency. In our case we have preferred the entire image as the neighborhood area. Empirically this preference provides us with more spatial frequency than state-of-the-art methods. Eventually, it results as uniformly highlighted salient regions with well defined borders.

Briefly, this algorithm finds the Euclidean distance between pixel vector of Gaussian filtered image with the mean vector for the input image, as shown at Equation 5.1.

$$S(x, y) = \| I_\mu - I_{w_{hc}}(x, y) \| \quad (5.1)$$

In the formula, S represents the saliency map for an image I . I_μ is the mean image feature vector, $I_{w_{hc}}$ is the corresponding image pixel vector value in the Gaussian blurred version of the original image to eliminate fine texture details and artifacts. We are only interested in the magnitude of the difference; hence, we are just using the L_2 norm of the difference. This provides computational efficiency.

As demonstrated in Figure 5.2, at the first place input image I is used for calculations of image average or mean vector and Gaussian blurred image. The next step is extracting Region of Interest points by calculating differences of these vectors. Eventually, an highlighted gray-scale map as saliency map is gathered as depicted in the final image of Figure 5.2. The next step after the map extraction is segmentation of region with interest points.

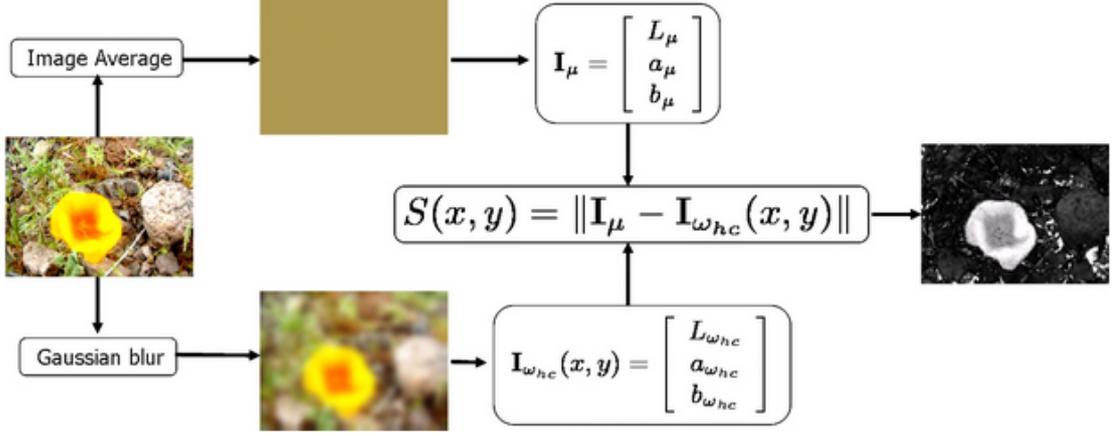


Figure 5.2: Saliency Detection Algorithm [3]

5.3 Salient Region Detection and Image Cropping

This process is the final phase in Region of Interest detection approach. We accomplish this step for images with saliency maps. As illustrated in Figure 5.2, we have gray-scale saliency map images. In these images, intensities of pixels indicate the relevancy of the pixel regarding searched object. Relying onto this evidence, we have developed our region selection algorithm briefly by cropping the area associated to these pixels.

Firstly, we create a mapping matrix from saliency map. Each element of this matrix maps the consecutive 40×40 pixels from saliency map. In other words, a binary matrix M whose size equals to one fortieth of saliency map size, is generated and each value represents corresponding 40×40 area from saliency map.

Secondly, the energy of these 40×40 areas is calculated and energy of entire image is as shown at Equation 5.2.

$$\xi = \sum_i^{N_i} S(i)/N_i \quad (5.2)$$

If the energy of the selected area is higher than optimized k value times energy of entire image, then we set corresponding M value as 1 , otherwise 0 .

$$M_i = \begin{cases} 1 & \text{if } \varepsilon_i \geq \varepsilon_I * k \\ 0 & \text{if } \varepsilon_i < \varepsilon_I * k \end{cases} \quad (5.3)$$

This k value is empirically optimized and it is set to 1.7 in order to be used in our thresholding process. After that, Gaussian thresholding is applied to get rid of noise, artifacts and some negligible pieces with 5×5 Gaussian filter. Then, the Gaussian blurred M matrix is thresholded with mean value of M .

$$g(x, y) = \frac{1}{2\pi\sigma^2} * e^{-\frac{x^2+y^2}{2\sigma^2}} \quad (5.4)$$

In the final phase, borders of a box comprehending all positive values of final M matrix are derived. In sum, corresponding box area from raw images are selected as Region of Interest points and cropped to process with *System II*.

Images in Figure 5.3 illustrate saliency maps and corresponding cropped images from randomly chosen images at Flower Set. Saliency maps from raw images are extracted and they seem as convincing as most of rest images in the flower set. Cropped (or saliency based selected) parts are also quiet successful; eventhough, at some images cropping do not seem perfectly successful since some components of flowers, especially small parts of petals, are deleted. However; it doesn't constitute a big problem in our work, and overall the RoI helps with experimental results.



Figure 5.3: ROI Samples; (a)left column Original images, (b)middle column Saliency Maps, (c)right column ROI detected images

Chapter 6

Evaluation

In this chapter we evaluate and report all classification results of three systems proposed in the thesis.

6.1 Data Set

We use the training portion of the flower data set from LifeClef2014 Plant Identification campaign to evaluate the proposed systems, containing 13,164 flower images. We have split roughly 1/3 of this data as test set and rest of them as training set due to absence of meta-data (ground-truth) for official test set. The resulting data set contains 8,798 training images and 4,366 test images, containing 483 and 381 different plant species, respectively.

Since there is no ground-truth about the dominant color of a flower among the meta-data, we have manually labelled a subset of the flower data set. As shown in the last column of Table 6.1, the subset includes 1,728 training images and 828 test images from 306 plant species. As can be seen in this table, the distribution of images and plant species vary for different color clusters. Furthermore, some flower species have images on more than one color clusters. Hence, the total number of species in Table 6.1 is higher than 306, which is the total number of different species in the test data. However, as indicated before, we believe that the small amount of overlap between different color groups is necessary and is makes it worthwhile to do a color clustering.

Table 6.1: Flower Subset with Ground-truth

	Train Images	Species	Test Images	Species
Red	200	41	100	41
Blue & Purple	400	64	200	57
Pink & Magenta	400	93	200	78
White & Yellow	400	235	200	155
Green	328	50	128	50
Overall	1728	483	828	381

6.2 Results

6.2.1 System I - Plant Identification Using Dense SIFT

The first set of results, shown in Table 6.2, belong to System 1, where the plants in the photographs are classified in one step. We give the results both for the full data set, and separately for the subset for which we manually assigned color clusters, so as to be able to compare with System II. As can be seen, plant identification accuracy for the full dataset is around 27%, while that for the subset is around 35%.

Table 6.2: System I - Plant Identification

	Train Images Count	Test Images Count	Number of Species	Accuracy
Full dataset	13,164	4,559	483	27.00%
Color groundtruth subset	1,728	828	306	34.54%

6.2.2 System II and III - Color Classification

Table 6.3 reports the results of the color classification used in System II in terms of precision and recall measures. For color clustering in a given cluster, the precision

value indicates the proportion of correctly classified images among all images that are put in that cluster, while the Recall value indicates the ratio of the images put in that cluster among all images that indeed belong to that cluster.

High precision and recall ratios are observed for the first three color clusters (Red, Blue&Purple, Pink&Magenta) color clusters. while the White&Yellow color cluster has somewhat lower precision and the Green color cluster has low recall values, respectively. We see that since the background is also green in most of the flower photographs, this cluster has the least overall accuracy. However, the weighted average precision is 92.83%, which is quite satisfactory.

Color classification results of the System II with region of interest detection, is similar to that of System II for the first three color clusters, as shown in Table 6.4. However, there is an improvement in the White&Yellow and Green color clusters. Overall, the color classifier precision increases from 92.83% with System II, to 94.72% with saliency based ROI selection method.

Table 6.3: System II - Color Classification Results

	Test Images	Classified As	True Positives	Precision	Recall
Red	100	108	98	90.74%	98.00%
Blue & Purple	200	200	195	97.50%	97.50%
Pink & Magenta	200	203	196	96.65%	98.00%
White & Yellow	200	232	191	82.32%	96.00%
Green	128	85	83	97.64%	64.84%
Overall	828	828	764	92.83%	92.26

Table 6.4: System II with ROI - Color Classification Results

	Test Images	Classified As	True Positives	Precision	Recall
Red	100	105	98	93.33%	98.00%
Blue & Purple	200	201	197	98.01%	98.50%
Pink & Magenta	200	202	198	98.02%	99.00%
White & Yellow	200	212	195	91.98%	97.50%
Green	128	108	97	89.81%	75.78%
Overall	828	828	785	94.72%	94.80%

6.2.3 System II - Plant Identification

The plant identification rates within each color cluster is given in Table 6.5. These rates show what percentage of all images in the test set are recognized correctly. For instance, out of the 100 images that are the Red cluster of the test set, 69 of them are correctly identified. Note that the corresponding 31% (100-69%) error includes both the color classification and plant identification results. In that sense, 69% accuracy in that cluster is very satisfactory.

Table 6.5: System II - Plant Identification Results

	Test Images	Classified As	Correctly Identified	Identification Accuracy
Red	100	108	69	69.00%
Blue & Purple	200	200	89	44.50%
Pink & Magenta	200	203	108	54.00%
White & Yellow	200	232	114	57.00%
Green	128	85	45	35.15%
Overall	828	828	425	51.32%

Table 6.6: System II with ROI - Plant Identification Results

	Test Image Numbers	Color Clustering Results	True Classified Numbers	Identification Accuracy
Red	100	105	73	73.00%
Blue & Purple	200	201	92	46.00%
Pink & Magenta	200	202	117	58.50%
White & Yellow	200	212	119	59.50%
Green	128	108	53	41.40%
Overall	828	828	454	54.83%

Plant identification results of the System II with region of interest detection are reported in Table 6.6. If we compare these classifiers with plant classifiers of System II, we see that the overall plant identification accuracy is increased from 51.31% to 54.83%. There are two main factors on this improvement; first of all color classification is more accurate, as the color classification is less affected by background. Second of all, we obtain a much more robust feature representation (BoW dense SIFT) due to ROI selection.

6.3 Summary

Evaluation of System I, System II and Improved System II are completed and comparisons of accuracies are demonstrated at Table 6.7. The accuracy means the percentage of correctly retrieved or classified images among all test subset as a final identification results of entire systems. 51% accuracy means that roughly 51 % of all images in the flower subset with color ground-truth are classified correctly among 306 flower species. As can be seen, there are great performance differences between System I and others.

Specifically, the accuracy of System I is 34.54% and our proposal color based preclassification process improves this system’s accuracy by almost 17% and 19% with System II and Improved System II, respectively. The second factor is efficient use of SVMs by dividing main classification problem into subsets or subproblems with prior classification tools.

Table 6.7: Results of All Systems

	System I	System II	Improved System II
	Accuracy	Accuracy	Accuracy
Red	-	69.00%	73.00%
Blue & Purple	-	44.50%	46.00%
Pink & Magenta	-	54.00%	58.50%
White & Yellow	-	57.00%	59.5%
Green	-	35.15%	41.40%
Overall	34.54%	51.32%	54.83%

We participated the Plant Identification campaign at 2013 and 2014 with System I and alternative models of System II. We have 3rd rank with System I on the flower category of campaign 2013 [10]. However, we used an alternative model of System II at 2014. In this model we are using time-era based classification instead of color based classification for the classification before the plant classifiers. We have 4th rank with this model against more experienced competitors than 2013 [11][12]. The ranks are decided by score between 1 to 0 from inverse of the rank of the correct

species. The top scores are respectively 0,339 and 0,22 and our score is 0,118 which means almost 20% correctly classified images. If we compare the results of System II with results of the top competitors, we can claim that System II could get good results almost as good as the top result.

Chapter 7

Appendix

WEKA wrapper in MATLAB for Random Forest

```
1 function Random_Forest_with_SWHH()  
2  
3 clear all; close all;  
4  
5 %% Initializing  
6 % adding the path to matlab2weka codes  
7 addpath([pwd filesep 'matlab2weka']);  
8 % adding Weka Jar file  
9 if strcmp(filesep, '\')% Windows  
10     javaaddpath('C:\Program Files\MATLAB\R2014b\toolbox\Weka  
        -3-6\weka.jar');  
11 end  
12 % adding matlab2weka JAR file that converts the matlab  
        matrices (and cells)  
13 % to Weka instances.  
14 javaaddpath([pwd filesep 'matlab2weka' filesep 'matlab2weka.  
        jar']);  
15  
16 %% Loading SWHH Dataset  
17 load SWHH_train
```

```

18 load SWHH_test
19
20 % numerical class variable
21 featName = {'SWHH'};
22 feature_train = SWHH_train(:,1:12);
23 class_num_t = SWHH_train(:,13);
24
25 feature_test = SWHH_test(:,1:12);
26 class_num_v = SWHH_test(:,13);
27
28 % converting to nominal variables (Weka cannot classify
    numerical classes)
29 class_train = cell(size(class_num_t));
30 uClass_num = unique(class_num_t);
31 tmp_cell = cell(1,1);
32 for i = 1:length(uClass_num)
33     tmp_cell{1,1} = strcat('class_', num2str(i-1));
34     class_train(class_num_t == uClass_num(i),:) = repmat(
        tmp_cell, sum(class_num_t == uClass_num(i)), 1);
35 end
36 clear uClass_num tmp_cell i
37
38 class_test = cell(size(class_num_t));
39 uClass_num = unique(class_num_t);
40 tmp_cell = cell(1,1);
41 for i = 1:length(uClass_num)
42     tmp_cell{1,1} = strcat('class_', num2str(i-1));
43     class_test(class_num == uClass_num(i),:) = repmat(
        tmp_cell, sum(class_num == uClass_num(i)), 1);
44 end
45 clear uClass_num tmp_cell i
46

```

```

47
48 % Choosing a regression tool to be used
49 % -----
50 % classifier = 1: Random Forest Classifier from WEKA
51 % classifier = 2: Gaussian Process Regression from WEKA
52 % classifier = 3: Support Vector Machine from WEKA
53 % classifier = 4: Logistic Regression from WEKA
54 classifier = 1;
55
56 %% Train and Test
57 idxCV = ceil(rand([1 N])*K);
58 actualClass = cell(size(class_num_v,1),1);
59 predictedClass = cell(size(class_num_v,1),1);
60 for k = 1:K
61
62     %performing regression
63     [actual_tmp, predicted_tmp, probDistr_tmp] =
        wekaClassification(feature_train, class_train,
        feature_test, class_test, featName, classifier);
64
65     %accumulating the results
66     actualClass(idxCV == k,:) = actual_tmp;
67     predictedClass(idxCV == k,:) = predicted_tmp;
68     clear feature_train class_train feature_test class_test
69     clear actual_tmp predicted_tmp probDistr_tmp
70 end
71 clear idxCV k

```

Bibliography

- [1] [Online]. Available: <http://www.codeproject.com/KB/recipes/619039/SIFT.JPG>
- [2] D. G. Lowe, “Distinctive image features from scale-invariant keypoints,” *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004. [Online]. Available: <http://dx.doi.org/10.1023/B:VISI.0000029664.99615.94>
- [3] R. Achanta, S. Hemami, F. Estrada, and S. Ssstrunk, “Frequency-tuned Salient Region Detection,” in *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR 2009)*, 2009, pp. 1597 – 1604, for code and supplementary material, click on the url below. [Online]. Available: http://ivrg.epfl.ch/supplementary_material/RK_CVPR09/index.html
- [4] D. G. Lowe, “Object recognition from local scale-invariant features,” in *ICCV*, 1999, pp. 1150–1157. [Online]. Available: <http://computer.org/proceedings/iccv/0164/vol%202/01641150abs.htm>
- [5] M. Everingham, S. M. A. Eslami, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, “The pascal visual object classes challenge: A retrospective,” *International Journal of Computer Vision*, vol. 111, no. 1, pp. 98–136, Jan. 2015.
- [6] H. Goëau, P. Bonnet, A. Joly, N. Boujemaa, D. Barthelemy, J. Molino, P. Birnbaum, E. Mouysset, and M. Picard, “The clef 2011 plant image classification task,” in *CLEF 2011 working notes*, Amsterdam, The Netherlands, 2011.
- [7] A. Joly, H. Müller, H. Goëau, H. Glotin, C. Spampinato, A. Rauber, P. Bonnet, W.-P. Vellinga, and B. Fisher, “Lifeclef 2014: multimedia life species identification challenges,” in *Proceedings of CLEF 2014*, 2014.

- [8] P. Forner, J. Gonzalo, J. Kekäläinen, M. Lalmas, and M. de Rijke, Eds., *Multilingual and Multimodal Information Access Evaluation - Second International Conference of the Cross-Language Evaluation Forum, CLEF 2011, Amsterdam, The Netherlands, September 19-22, 2011. Proceedings*, ser. Lecture Notes in Computer Science, vol. 6941. Springer, 2011. [Online]. Available: <http://dx.doi.org/10.1007/978-3-642-23708-9>
- [9] T. Catarci, P. Forner, D. Hiemstra, A. Peñas, and G. Santucci, Eds., *Information Access Evaluation. Multilinguality, Multimodality, and Visual Analytics - Third International Conference of the CLEF Initiative, CLEF 2012, Rome, Italy, September 17-20, 2012. Proceedings*, ser. Lecture Notes in Computer Science, vol. 7488. Springer, 2012. [Online]. Available: <http://dx.doi.org/10.1007/978-3-642-33247-0>
- [10] P. Forner, H. Müller, R. Paredes, P. Rosso, and B. Stein, Eds., *Information Access Evaluation. Multilinguality, Multimodality, and Visualization - 4th International Conference of the CLEF Initiative, CLEF 2013, Valencia, Spain, September 23-26, 2013. Proceedings*, ser. Lecture Notes in Computer Science, vol. 8138. Springer, 2013. [Online]. Available: <http://dx.doi.org/10.1007/978-3-642-40802-1>
- [11] E. Kanoulas, M. Lupu, P. D. Clough, M. Sanderson, M. M. Hall, A. Hanbury, and E. G. Toms, Eds., *Information Access Evaluation. Multilinguality, Multimodality, and Interaction - 5th International Conference of the CLEF Initiative, CLEF 2014, Sheffield, UK, September 15-18, 2014. Proceedings*, ser. Lecture Notes in Computer Science, vol. 8685. Springer, 2014. [Online]. Available: <http://dx.doi.org/10.1007/978-3-319-11382-1>
- [12] A. Joly and H. Goëau, “Lifeclef plant identification task 2014,” in *Working Notes for CLEF 2014 Conference, Sheffield, UK, September 15-18, 2014.*, 2014, pp. 598–615. [Online]. Available: <http://ceur-ws.org/Vol-1180/CLEF2014wn-Life-JolyEt2014a.pdf>
- [13] H. M. Wallach, “Topic modeling: Beyond bag-of-words,” in *Proceedings of the 23rd International Conference on Machine Learning*, ser. ICML

- '06. New York, NY, USA: ACM, 2006, pp. 977–984. [Online]. Available: <http://doi.acm.org/10.1145/1143844.1143967>
- [14] T. Joachims, “Training linear svms in linear time,” in *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD '06. New York, NY, USA: ACM, 2006, pp. 217–226. [Online]. Available: <http://doi.acm.org/10.1145/1150402.1150429>
- [15] C.-J. Hsieh, K.-W. Chang, C.-J. Lin, S. S. Keerthi, and S. Sundararajan, “A dual coordinate descent method for large-scale linear svm,” in *Proceedings of the 25th International Conference on Machine Learning*, ser. ICML '08. New York, NY, USA: ACM, 2008, pp. 408–415. [Online]. Available: <http://doi.acm.org/10.1145/1390156.1390208>
- [16] C.-C. Chang, “Libsvm.” <http://www.csie.ntu.edu.tw/~cjlin/libsvm/>, 2008. [Online]. Available: <http://www.csie.ntu.edu.tw/~cjlin/libsvm/>
- [17] C. Liu, J. Yuen, A. Torralba, J. Sivic, and W. T. Freeman, “Sift flow: Dense correspondence across different scenes,” in *Proceedings of the 10th European Conference on Computer Vision: Part III*, ser. ECCV '08. Berlin, Heidelberg: Springer-Verlag, 2008, pp. 28–42. [Online]. Available: http://dx.doi.org/10.1007/978-3-540-88690-7_3
- [18] A. J. Chavez, “Image classification with dense sift sampling: An exploration of optimal parameters,” Ph.D. dissertation, Manhattan, KS, USA, 2012, aAI3513432.
- [19] M. Perdoch, O. Chum, and J. Matas, “Efficient representation of local geometry for large scale object retrieval,” in *2009 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2009), 20-25 June 2009, Miami, Florida, USA*, 2009, pp. 9–16. [Online]. Available: <http://dx.doi.org/10.1109/CVPRW.2009.5206529>
- [20] A. Bosch, A. Zisserman, and X. Muñoz, “Scene classification via plsa,” in *Proceedings of the 9th European Conference on Computer Vision - Volume Part IV*, ser. ECCV'06. Berlin, Heidelberg: Springer-Verlag, 2006, pp. 517–530. [Online]. Available: http://dx.doi.org/10.1007/11744085_40

- [21] A. Bosch, X. Muñoz, and R. Martí, “Review: Which is the best way to organize/classify images by content?” *Image Vision Comput.*, vol. 25, no. 6, pp. 778–791, Jun. 2007. [Online]. Available: <http://dx.doi.org/10.1016/j.imavis.2006.07.015>
- [22] J. L. Bentley, “Multidimensional binary search trees used for associative searching,” *Commun. ACM*, vol. 18, no. 9, pp. 509–517, Sep. 1975. [Online]. Available: <http://doi.acm.org/10.1145/361002.361007>
- [23] J. Macqueen, “Some methods for classification and analysis of multivariate observations,” in *In 5-th Berkeley Symposium on Mathematical Statistics and Probability*, 1967, pp. 281–297.
- [24] B. E. Boser, I. M. Guyon, and V. N. Vapnik, “A training algorithm for optimal margin classifiers,” in *Proceedings of the Fifth Annual Workshop on Computational Learning Theory*, ser. COLT ’92. New York, NY, USA: ACM, 1992, pp. 144–152. [Online]. Available: <http://doi.acm.org/10.1145/130385.130401>
- [25] C. Cortes and V. Vapnik, “Support-vector networks,” in *Machine Learning*, 1995, pp. 273–297.
- [26] C. Wengert, M. Douze, and H. Jégou, “Bag-of-colors for improved image search,” in *Proceedings of the 19th International Conference on Multimedia 2011, Scottsdale, AZ, USA, November 28 - December 1, 2011*, 2011, pp. 1437–1440. [Online]. Available: <http://doi.acm.org/10.1145/2072298.2072034>
- [27] S. T. Yildiran, B. A. Yanikoglu, and E. Abdullah, “Plant identification using local invariants,” in *2014 22nd Signal Processing and Communications Applications Conference (SIU), Trabzon, Turkey, April 23-25, 2014*, 2014, pp. 2094–2097. [Online]. Available: <http://dx.doi.org/10.1109/SIU.2014.6830674>
- [28] B. A. Yanikoglu, E. Aptoula, and S. T. Yildiran, “Sabanci-okan system at imageclef 2013 plant identification competition,” in *Working Notes for CLEF 2013 Conference, Valencia, Spain, September 23-26, 2013.*, 2013. [Online]. Available: <http://ceur-ws.org/Vol-1179/CLEF2013wn-ImageCLEF-YanikogluEt2013.pdf>

- [29] B. A. Yanikoglu, S. T. Yildiran, C. Tirkaz, and E. Aptoula, “Sabanciokan system at lifeclef 2014 plant identification competition,” in *Working Notes for CLEF 2014 Conference, Sheffield, UK, September 15-18, 2014.*, 2014, pp. 771–777. [Online]. Available: <http://ceur-ws.org/Vol-1180/CLEF2014wn-Life-YanicogluEt2014.pdf>
- [30] A. Hanbury and J. Serra, “Mathematical morphology in the hls colour space,” In Proceedings of British Machine Vision Conference, Tech. Rep., 2002.
- [31] A. Hanbury, “Circular statistics applied to colour images,” in *Computer Vision Winter Workshop*, Valtice, Czech Republic, February 2003, pp. 55–60.
- [32] L. Breiman and E. Schapire, “Random forests,” in *Machine Learning*, 2001, pp. 5–32.
- [33] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten, “The weka data mining software: an update,” *SIGKDD Explor. Newsl.*, vol. 11, no. 1, pp. 10–18, 2009.