CHAPTER 5

DESIGN OF LEAF IMAGE SEGMENTATION METHOD

Recent advances in imaging technologies with significant contributions from computer science, computer engineering and electrical have witnessed a revolutionary growth in the field of digital imaging. Revolutionary improvements in engineering and computing technologies have made it possible to acquire high-resolution images, to analyze structural and functional information for computer-assisted analysis, evaluation, and intervention.

Segmentation refers to the process of partitioning a digital image into multiple regions (sets of pixels also known as super pixels). The objective of segmentation is to simplify and/or change the representation of an image into something that is more meaningful and easier to analyze (Shapiro and Stockman, 2001). Image segmentation techniques work by locating objects and boundaries (lines, curves etc.) and assigning labels to every pixel such that pixels with the same label share certain visual characteristics.

The result of image segmentation is a set of regions that collectively cover the entire image, where each pixel in a region is similar with respect to some characteristic or computed property, such as color, intensity, or texture. Regions that are adjacent to each other are significantly different with respect to the same characteristic(s).

The second stage of the preprocessing phase is segmentation, which is used to separate the region of interest (ROI) that is, the leaf, from its background. This chapter discusses the segmentation technique used during the design of CAP-LR system. The chapter begins with an introduction to the concepts of image segmentation, followed by the description of the proposed algorithm.

5.1. OVERVIEW OF IMAGE SEGMENTATION

Image segmentation is the most important task in many image processing systems, such as pattern recognition, image retrieval and small surveillance. The result of segmentation is mainly used for image content understanding and visual object recognition (Singh,2004) through the identification of region of interest. The goal of segmentation is to simplify and/or change the representation of an image into something that is more meaningful and easier to analyze.

Image segmentation is used to locate objects and boundaries (lines, curves, etc.) in images and assigns a label to every pixel in an image, in a manner that pixels with the same label share certain visual characteristics. In medical imaging, the aim is to separate different parts of the anatomy, which is proving to be very challenging with the overwhelming number of visual patterns in an image. Thus, image segmentation has been, and still is, a relevant research area in Computer Vision (Zhang, 2010). Eventhough, several hundreds of segmentation algorithms have been proposed for natural images in the last 30 years, it is still evasive in botanical field.

Segmentation is the process of partitioning an image into non-intersecting regions such that each region is homogeneous and the union of no two adjacent regions is homogeneous and it can also be used to the process of isolating objects of interest from the rest of the scene. This pixel level processing is of paramount importance for many image processing applications (Musoko and Prochazka, 2004; Khouzani and Zadeh, 2005). These techniques play a vital role in CAP-LR processing systems, as they identify region of interest (leaf) that lead to content understanding and plant recognition. Segmentation appears to be a key issue in modern image analysis, enabling numerous botanical and agricultural applications. Segmentation is the process of assigning labels to pixels in 2-D images or voxels in 3-D images. The effect is that the image is split up into segments, otherwise called as regions or areas.

In plant recognition through leaf images, it is essential for quantification of outlined leaf structures and for visualization of relevant image data. Segmentation in this area includes plant identification, plant classification and disease identification. The problems of image segmentation and grouping remain as a great challenge in computer vision. From the period of the Gestalt movement in psychology, perceptual grouping plays an important and powerful role in human visual perception and several computational vision problems make good use of segmented images.

In general imaging science, segmentation is an important step in any image analysis process where an image is taken as input and the output is some detailed description of the scene or object. It has the same importance in leaf recognition also. Figure 5.1 depicts the steps involved in typical image analysis workflow showing segmentation as key step for succeeding image representation and recognition stages.

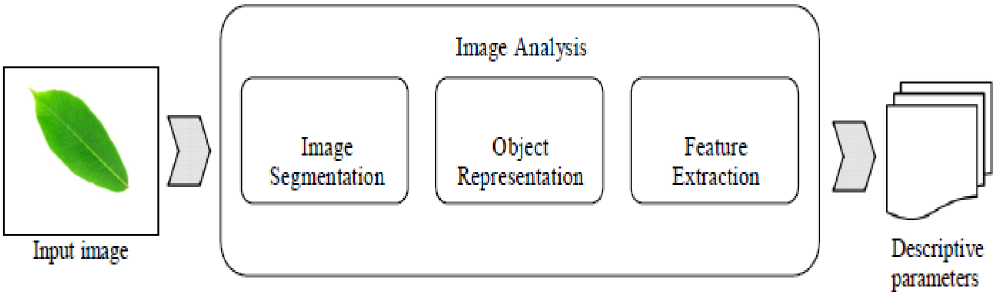


Figure 5.1 : Steps in Image Analysis

Classically, image segmentation is defined as the partitioning of an image into non-overlapping, constituent regions which are homogeneous with respect to some characteristic such as intensity or texture. Mathematically, if I is the domain of the image, then the problem of segmentation is to determine the sets $S_k \subset I$ whose union is the whole image I and S_k must satisfy Equation (5.1) and should be connected to each other.

$$I = \bigcup_{k=1}^{k} S_k \qquad (5.1)$$

where $S_k \cap S_j = \phi$ for $k \neq j$. In general, segmentation finds those sets that correspond to distinct structures or regions of interest in the image.

5.2. IMPORTANCE OF SEGMENTATION IN CAP-LR

In recent years, plant life studies has seen an explosion of new and exciting applications which have lead to more cost-effective and efficient plant species identification and recognition. Recent advances in medical imaging devices have made it possible to provide the experts access to the data from remote places like hill regions and under water plants. These innovative explosions in hardware equipments have revolutionized the computer aided systems and are producing more rich, evocative and multidimensional information than ever before. This leads to the increase in discovering new information on plant kingdom. Image segmentation plays a crucial role in many such imaging applications**.**

Several approaches to segmentation have been proposed which can be categorized into cluster based techniques histogram based techniques (Zwiggelaar, 2010), compression based techniques (Biswas, 2003), edge detection, region growing methods (Chen and Georganas, 2008), Graph partitioning methods (Sumengen and Manjunath, 2006) and watershed transformation (Wenzhong and Xiaohui, 2010). Another popular method used is the curve evolution methods (Farzinfar *et al.*, 2010).

These methods evolve the initialized curve(s) to the boundaries of objects in an image. The evolution of the curves may be driven by image gradient information (Kichenassamy *et al.*, 1995; Caselles *et al.*, 1997), region information (Chan and Vese, 2001; Jehan-Besson *et* al., 2003), or their combination. The image segments generated are enclosed by closed contours and are unsupervised which make it an ideal candidate for image post processing. These methods are more suitable for simple images and their performance degrades both in terms of segmentation and complexity, with complicated images (Paragios and Deriche, 2000). In such cases, methods that depend on the internal characteristics of the images are preferred.

Segmentation used to extract the leaf from the input images that is difficult due to the complexity and variability of the shapes of interest (McInerney and Terzopoulos, 1996). Another difficulty is the lack of a ’golden standard’ method. The need for such a method arises as the manual delineation is very time consuming and operator dependent. Thus, automating the process of segmentation is an important and difficult task.

Image segmentation is a fundamental task in agriculture and computer graphics vision. It has been an active topic for the past 50 years, but is still considered immature in the field of horticultural image processing. It is considered as challenging field where there are increasing demands to understand the intricacies of the leaf images by various botanical applications where the number of visual patterns, quality, size of the image captured are also increasing in an overwhelming fashion. As a result, the usage of computers and algorithms for facilitating their processing and analysis has also increased. Segmentation of leaf images is considered challenging because these images contain different textured and coloured regions with varying background and are often subjected to illumination changes or environmental effects.

All these factors emphasize the fact that there is an urgent need in leaf recognition systems for a fast and reliable image segmentation model that requires minimum intervention from the user. Existing solutions for segmentation to extract the Region of Interest (ROI) regions face three major drawbacks. They are performance degradation when supplied with large sized images, degradation of segmentation accuracy due to the quality of the acquired image and speed of segmentation not meeting the standards of the modern equipments.

Although many methods are proposed, the search for an algorithm that accurately segment an arbitrary image using a single method is still considered difficult. In recent years, more and more attention has been paid to combine segmentation algorithms and information from multiple feature spaces (e.g. color, texture, and pattern) in order to improve segmentation results.

Using wavelets for segmenting images is a concept used in most of the studies by many researchers (Khalifa *et al.*, 2012; Priya and Gobu, 2013). However, the performance of these techniques comes down when presented with images that have varied representations of the same colour, which is often the case with leaf images. Figure 5.2 shows an example of such a situation.

Figure 5.2 : Examples of Leaf with Color Variation

In a color texture segmentation algorithm, a label is assigned to each pixel based on its color properties and texture properties. Although significant works have focused on developing algorithms based on either color (Cheng *et al.*, 2001; Comanicui and Meer, 2002) or texture features (Sengur *et al., 2008;* Sengur, 2008) separately, the area of combined color and texture segmentation remains open and active. To resolve these disadvantages, this study enhances Wavelet based method to include texture characteristics along with color features. These features were then used by a clustering based segmentation technique to extract the leaf image. This method is referred as 'Enhanced Wavelet based Segmentation using Clustering and Texture based Color Features (WCF Method)' in this thesis. The details of the algorithm are presented in the subsequent sections of this chapter.

5.3. PROPOSED SEGMENTATION METHOD

The proposed algorithm that is used to segment the leaf image from its background uses texture and color features to form a feature vector, which are then segmented using K means clustering algorithms. The algorithm makes use of wavelet frame decomposition during the extraction of texture features and L*u*v* color space to extract color features. The WCF method consists of five major steps and the algorithmic flow is presented in Figure 5.3.

(i) Extraction of *texture features*

(ii) Extraction of color features

(iii) Creation of high dimensional feature space

(iv) Perform mean shift filtering to smooth and preserve edges

(v) *Use of K-Means clustering* to obtain segments

The algorithm begins by converting the input leaf image from RGB color space to L*u*v color space. This process produces the results with three color channels, L, u and v respectively. Simultaneously, the algorithm also applies wavelet transformation to obtain the four subbands LH(Low High), HL(High Low), LH(Low High) and HH(High Low). The next step classifies the coordinates into four texture classes for which the median energy is calculated. The combined texture and color features form a high dimensional feature space, which is converted to a low level space using a mean shift filtering algorithm. This result is then grouped by K-Means algorithm to segment the input image into ROI and background. The ROI is the output of the extracted leaf image.

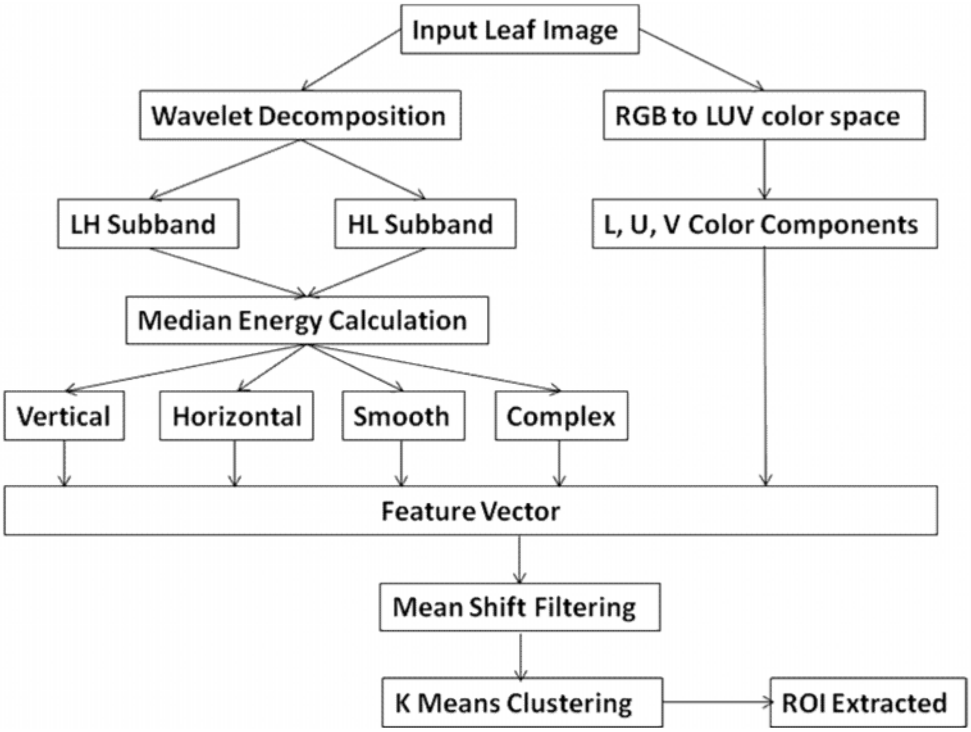


Figure 5.3 : Proposed WCF Method

5.3.1. Step 1 : Extraction of Texture Features

The WCF method uses the Discrete Wavelet Frame (DWF) decomposition technique, which is a variation of the Discrete Wavelet Transformation (DWT), for texture characterization. The WCF algorithm use DWF to classify the texture information for segmentation purpose. Several statistical approaches have been used to perform texture analysis for image segmentation in the segmentation history (Chen and Pavlidis, 1983; Kashyap *et al.*, 1982). These methods were enhanced by researchers (Porat *et al.*, 1989; Bovic *et al.*, 1990) to improve the information preservation capability of the texture features extracted. However, some disadvantages such as increased computational cost and irreversibility, which are inherent to those approaches, are common in these techniques. Experiments conducted by Malat (1989) and Rioul and Vetterli (1991) showed that these demerits can be eliminated using the wavelet transform.

In WCF method, the problem of texture classification and segmentation is approached with algorithms that are based on the concepts of DWF. The aim of the analysis is to determine corresponding characteristics to each texture content so that each is uniquely defined. Such a distinction takes place in the frequency domain, where the input image is decomposed to different frequency levels using the DWF. Once these characteristics are deduced, statistical properties are applied to conclude those features necessary to describe and classify the texture content. Although the philosophy to this approach has been introduced in 1980's (Unser, 1986), the proposed segmentation scheme differs in the methodology used for evaluating the texture parameters. The main advantages of the DWF representation are that it focuses on scale and orientation texture features and decomposes the image into orthogonal components, which are translation-invariant. This section first describes the DWF algorithm followed by the method used to classify the texture information.

- **Discrete Wavelet Frame Decomposition**

The fundamental tools used for building the processing of texture images are a group of filters and the concept of wavelet frames. A lowpass filter H(z) and its complementary highpass G(z) forms the basis for generating more filters by upsampling, so that the whole range of bands is covered. For these basic types of filters the following hold true, respectively.

$$\left.\begin{array}{ll} H(z) & = \dfrac{z^2 + 4z + 6 + 4z^{-1} + z^{-2}}{16} \\ G(z) & \quad zH(-z^{-1}) \end{array}\right\} \qquad (5.2)$$

in the frequency domain and

$$h(n) = Z^{-1}\ \{H(z)\}, \quad g(n) = (-1)^{1-n}\ h(1-n) \qquad (5.3)$$

in the time domain. In addition, the generated filters are characterized by locality, thus, taking advantage of the periodicity of signals. Such filters can form orthogonal wavelet base functions of the form:

$$\left.\begin{aligned}\varphi_{i,t}(k) &= 2^{i/2}h_i(k-2^i t)\\ \psi_{i,t}(k) &= 2^{i/2}g_i(k-2^i t)\end{aligned}\right\} \tag{5.4}$$

where φ, ψ are the wavelet base functions, i is the scale index and t is the translation index. And so the input signal can be decomposed into wavelet coefficients corresponding to different layers of frequency resolution. In order, however, to consider characteristics of texture, such as periodicity and translational invariance, the Discrete Wavelet Frames (DWF) is used to define a vector representing the filters necessary for decomposition at the different frequency levels. All of the above should be extended into 2-D so that it becomes functional for images with texture, the features of which must be extracted. This can be accomplished by forming wavelet bases which result from the cross product of separable bases in each direction, as in Equation (5.5)

$$\begin{aligned}\Phi(x,y) &= \varphi(x)\varphi(y) \quad & \zeta_1(x,y) &= \varphi(x)\psi(y)\\ \zeta_2(x,y) &= \psi(x)\varphi(y) \quad & \zeta_3(x,y) &= \psi(x)\psi(y)\end{aligned} \tag{5.5}$$

where Φ, ζ_1, ζ_2, ζ_3 are the 2-D wavelet base functions and φ, ψ are as defined in Equation (5.4). Thus, the computational complexity is reduced, as the rows and columns of the image are processed separately as though they were 1-D signals. The decomposition algorithm for images (2-D) is described in Equation (5.6).

$$\left.\begin{aligned}d_{1,i+1}(k,l) &= [h]_{2^i}(k)*[g]_{2^i}(l)+s_i(k,l)\\ d_{2,i+1}(k,l) &= [g]_{2^i}(k)*[h]_{2^i}(l)+s_i(k,l)\\ d_{3,i+1}(k,l) &= [g]_{2^i}(k)*[g]_{2^i}(l)+s_i(k,l)\\ s_{i+1}(k,l) &= [h]_{2^i}(k)*[h]_{2^i}(l)+s_i(k,l)\end{aligned}\right\} \tag{5.6}$$

where (k, l) is an image point, $[\ ]_m$ is upsampling with a factor of m, $d_{1,i+1}$, $d_{2,i+1}$, $d_{3,i+1}$ are the details of i+1 layer and s_{i+1} is the approximation of the decomposition.

Thus, unlike other decompositions, DWF is computationally inexpensive for the evaluation of low frequency components. Dissimilar to other wavelet-based approaches, the output of the filter banks is not sub-sampled in DWF decomposition between levels. This provides translation invariant texture description of input signal. This property yields a better estimation of texture statistics and more detailed characterization at region boundaries. DWF decomposition can be calculated by successive 1-D processing along the rows and columns of the image. Figure 5.4 illustrates the block diagram of one-level image decomposition and synthesis.

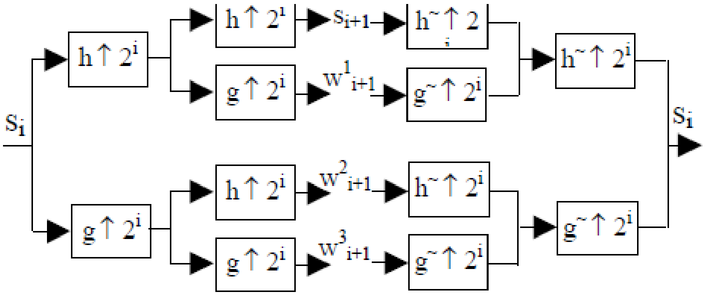


Figure 5.4 : Discrete Wavelet frame decomposition of the ith level

In the decomposition phase of 2-D DWFT, each row of the image is separately filtered by H and G. The resulting row-transformed image is then similarly filtered in the column direction, finally yielding four subbands at the first decomposition level (j=1). The three detail subbands contain the vertical (LH), horizontal (HL), and diagonal (HH) high frequency information, , while the approximation (LL) subband contains the low pass filtered version of the original image. This approximation subband is subsequently passed to the next level for further subband decomposition. Thus, a DWFT with N decomposition levels will have a total of 3N + 1 frequency subbands, all of them are of the same size.

- **Texture Classification**

The first step uses the DWF to one-level decomposition on the input image to obtain four subbands Figure (5.5) where L and H correspond to low-pass and high-pass filters respectively.

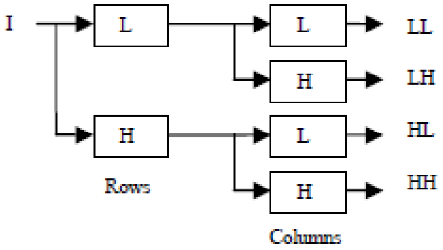


Figure 5.5 : Discrete Wavelet Frame decomposition

A texture is characterized by a set of median values of energy estimated in a local window at the output of the corresponding filter bank. The energy in a local window can be calculated using coefficients of DWF decompositions (LL, LH, HL, and HH) where the energy is defined as the square of the coefficients. The advantage of using median value is that it preserves the energy associated with texture between regions. The subbands at the output of filter bank in Figure (5.5) correspond to approximate, horizontal, vertical and diagonal components of the input image signal. The low-pass component is not used because a texture is better described through the higher frequency channels than through the approximation component.

A pixel in the textured region can be classified into one of four texture categories based on texture orientation (Chen *et al.*, 2003). They are

(i) Smooth (not enough energy in any orientation)

(ii) Vertical (dominant energy in vertical orientation)

(iii) Horizontal (dominant energy in horizontal orientation)

(iv) Complex (no dominant orientation).

Texture feature extraction consists of two steps. First, the energy of LH and HL subbands are classified into two categories (0 and 1) using K-means clustering algorithm. Second, a further classification is made using a combination of two categories in each subband LH and HL. The classification is performed using the simple rule-based procedure as given below.

$$\text{Pixel classified as} \rightarrow \begin{cases} \text{Smooth} & \text{Category} = 0 \text{ in both LH and HL subbands} \\ \text{Vertical} & \text{Category} = 0 \text{ in LH subband and Category} = 1 \text{ in HL subband} \\ \text{Horizontal} & \text{Category} = 1 \text{ in LH subband and Category} = 0 \text{ in HL subband} \\ \text{Complex} & \text{Category} = 1 \text{ in both LH and HL subbands} \end{cases}$$

Thus, according to the above rules, a pixel is classified as smooth if its category is 0 in both LH and HL subbands, vertical if its category is 0 in LH, and 1 in HL subbands, horizontal if its category is 1 in LH, and 0 in HL subbands and finally, complex if its category is 1 in both LH and HL subbands.

5.3.2. Step 2 : Extraction of Color Features

To obtain a meaningful segmentation the perceived color differences should correspond to Euclidean distances in the color space chosen to represent the features (pixels). An Euclidean metric, however, is not guaranteed for a color space (Wyszecki and Stiles, 1982). The spaces L*u*v* or L*a*b* were especially designed to best approximate perceptually uniform color spaces. In both cases, L*, the lightness (relative brightness) coordinate is defined the same manner and the two spaces differ only through the chromaticity coordinates. The dependence of all three coordinates on the traditional RGB color values is nonlinear. The metric of perceptually uniform color spaces is discussed in the context of feature representation for image segmentation in (Connolly, 1996). In practice, there is no clear advantage between using L*u*v* or L*a*b*, in the proposed algorithms and motivated by a linear mapping property, L*u*v color space is used.

In order to characterize the color content of a leaf image, the CIE L*u*v* color space is used. The L*u*v* color coordinate system has the advantage that it is designed to be perceptually uniform, meaning that the same distance in the color space leads to equal human color difference perception, i.e., images that are perceptually similar have the same chromaticity components. It also has the advantage that lightness L is distinct and independent from the chromaticity coordinates (a, b).

CIE L*u*v* is based on CIE Yu'v' and is a further attempt to linearise the perceptibility of unit vector color differences. This color model is a non-linear color space where the conversions are reversible. The information pertaining to coloring is centered on the color of the white point of the system. The non-linear relationship for Y* is intended to mimic the logarithmic response of the eye. The L, u, v components are obtained using Equation (5.7), (5.8) and (5.9). L* scales from 0 to 100 for relative luminance (Y/Yn) scaling 0 to 1.

$$L^* = \begin{cases} 116\left(\frac{Y}{Y_n}\right)^{1/3} - 16 & \text{if } \frac{Y}{Y_n} > 0.008856 \\ 903.3\left(\frac{Y}{Y_n}\right) & \text{if } \frac{Y}{Y_n} \leq 0.008856 \end{cases} \tag{5.7}$$

$$u^* = 13(L^*)(u' - u'_n) \tag{5.8}$$

$$v^* = 13(L^*)(v' - v'_n) \tag{5.9}$$

The quantities u_n' and v_n' refer to the reference white or the light source; for the 2° observer and illuminant C, $u_n' = 0.2009$, $v_n' = 0.4610$. Equations for u' and v' are given in Equations (5.10) and (5.11).

$$u' = 4X / (X + 15Y + 3Z) = 4x / (-2x + 12y + 3) \tag{5.10}$$

$$v' = 9Y / (X + 15Y + 3Z) = 9y / (-2x + 12y + 3) \tag{5.11}$$

The transformation from (u',v') to (x,y) is as follows:

$$x = 27u' / (18u' - 48v' + 36) \tag{5.12}$$

$$y = 12v' / (18u' - 48v' + 36) \tag{5.13}$$

The L*u*v* color space is derived from the CIE XYZ color space. The conversion process uses a two-step procedure.

Step 1 : Converts RGB to XYZ color space

Step 2 : Converts XYZ to L*u*v color space using equations 5.14.

An inverse procedure is used to get back the original RGB color spaced image.

- **RGB-XYZ Conversion**

RGB values in a particular set of primaries can be transformed to and from CIE XYZ via a 3x3 matrix transformation and involve a set of three linear-light components (tristimulus values) that conform to the CIE color-matching functions. The CIE XYZ color space is a special set of tristimulus values, where in XYZ any color is represented as a set of positive values. To convert from XYZ to RGB, the Equation 5.14 is used. The range for valid R, G, B values is [0,1]. The inverse transformation matrix is given in Equation (5.15).

$$\begin{bmatrix} R \\ G \\ B \end{bmatrix} = \begin{bmatrix} 3.240479 & -1.537150 & -0.498535 \\ -0.969256 & 1.875992 & 0.041556 \\ 0.055648 & -0.204043 & 1.057311 \end{bmatrix} * \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \qquad (5.14)$$

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} 0.412453 & 0.357580 & 0.180423 \\ 0.212671 & 0.715160 & 0.072169 \\ 0.019334 & 0.119193 & 0.950227 \end{bmatrix} * \begin{bmatrix} R \\ G \\ B \end{bmatrix} \qquad (5.15)$$

- **XYZ-CIE LUV Conversion Process**

The transformation from CIE LUV to XYZ is performed as follows:

$$u' = u / (13L^{*}) + u_n \qquad (5.16)$$

$$v' = v / (13L^{*}) + v_n \qquad (5.17)$$

$$Y = ((L^* + 16) / 116)^3 \quad (5.18)$$

$$X = -9Yu' / ((u' - 4) v' - u'v') \quad (5.19)$$

$$Z = (9Y - 15v'Y - v'X) / 3v' \quad (5.20)$$

5.3.3. Step 3 : Feature Space Generation

This step generates the feature vector such that every pixel in the image has p-dimensional feature vector ,which includes spatial (x,y), color (gray-level or L*u*v values) and texture (smooth, vertical, horizontal or complex) information.

5.3.4. Step 4 : Mean Shift Filtering Algorithm

The result of Step 3 is a high dimensional feature space. It is a well-known fact that when the number of distinct feature vectors is large, the size of the feature space is reduced by grouping nearby vectors into a single cell (Comaniciu and Meer, 2002). In general, conventional clustering techniques are used for this purpose. But, these traditional clustering techniques (Jain and Dubes, 1988) when used for feature space analysis are reliable only if the number of clusters is small and known a priori. Estimating the number of clusters from the data is computationally expensive and not guaranteed to produce satisfactory result.

Another drawback of these techniques is that, most often, they work on the assumption that the individual clusters obey multivariate normal distributions. The parameters of the mixture are then estimated by minimizing an error criterion. For example, large classes of thresholding algorithms are based on the Gaussian mixture model of the histogram (Sahoo *et al.*, 1988). However, there is no theoretical evidence that an extracted normal cluster necessarily corresponds to a significant image feature. On the contrary, a strong artifact cluster may appear when several features are mapped into partially overlapping regions.

Nonparametric density estimation (Fukunaga, 1990) avoids the use of the normality assumption. The two families of methods, Parzen window, and k-nearest neighbors, both require additional input information (type of the kernel, number of neighbors). This information must be provided by the user, and for multi-modal distributions it is difficult to guess the optimal setting. Thus, a reliable general and simple non-parametric density estimation algorithm technique for feature space analysis is required. For this purpose, a Mean Shift Filtering algorithm is used.

Numerous nonparametric clustering methods have been described in the literature and they can be classified into two large classes: density estimation and hierarchical clustering. Hierarchical clustering techniques either aggregate or divide the data based on some proximity measure. Jain and Dubes (1988) have provided a survey of hierarchical clustering methods. The hierarchical methods tend to be computationally expensive and the definition of a meaningful stopping criterion for the fusion (or division) of the data is not straightforward.

The rationale behind the density estimation based nonparametric clustering approach is that the feature space can be regarded as the empirical probability density function (p.d.f.) of the represented parameter. Thus, in the feature space, dense regions correspond to local maxima of the p.d.f. After determining the location of a mode, the cluster associated with it is delineated based on the local structure of the feature space (Herbin et al., 1996; Touzani and Postaire, 1989; Wilson and Spann, 1990). The WCF method uses the mean shift filtering procedure for improving the clustering process and the details are explained below.

Mean shift filtering is a data clustering algorithm commonly used in computer vision and image processing. For each pixel of an image (having a spatial location and a particular color), the set of neighboring pixels (within a spatial radius and a defined color distance) is determined. The new spatial center (spatial mean) and the new color mean value are calculated for this set of neighbor pixels, which serve as the new center for the next iteration. This procedure is iterated until the mean value converges. At the end of the iteration, the final mean color is assigned to the starting position of that iteration. Application domains include cluster analysis in computer vision and image processing.

Mean shift is a non-parametric feature-space analysis iterative technique that has more robustness than other clustering-based image segmentation algorithms like K-Means. The mean shift image segmentation algorithm (Fukunaga and Hostetler, 1975), later adapted by Cheng (Cheng, 1995), considers a joint domain representation that includes spatial and range domains. An image is represented as a two dimensional lattice where the space of the lattice is known as spatial domain and the gray-level or color information is represented in the range domain.

Every pixel in the image can be considered as a p-dimensional vector where p=1 in gray-level and p=3 for color images. The dimension of joint domain representation becomes d=p+2. Using this representation, the mean shift filtering is performed to smooth the image and to preserve the region boundaries based on color and spatial information. However, in cases where colors in region boundaries are similar, this representation will not be sufficient and additional features are needed. In this study, this problem is addressed and the algorithm extends the mean shift feature space by integrating texture feature to improve the segmentation.

The mean shift estimate of gradient of a density function and the associated iterative procedures of mode seeking has been developed by Fukunaga and Hostetler (1975). The property of data compaction of the mean shift has been exploited in image segmentation. Based on the idea of iteratively shifting a fixed size window to the average of the data points within, Comaniciu and Meer (2002) developed a simple method of image segmentation. The mean shift procedure can be obtained by successively computing the mean shift vector $M_h(x)$, and translating the window $S_h(x)$ by $M_h(x)$.

$$M_h(x) = \frac{1}{n_x} \sum_{x_i \in s_h(x)} x_i - x \qquad (5.21)$$

The mean shift vector always points towards the direction of the maximum increase in the density f(x), so it can define a path leading to a local density maximum.

$$f(x) = \frac{1}{nh^d} \sum_{i=1}^{n} K\left(\frac{x - x_i}{h}\right) \qquad (5.22)$$

Let $\{x_i\}_{i=1..n}$ be an arbitrary set of n points in the d-dimensional Euclidean space R^d. f(x) is the multivariate kernel density estimate with the kernel K(x) and the window $S_h(x)$ radius h. The mean shift filtering procedure is:

- For each j = 1…n
  - Initialize k = 1 and $y_k = x_j$
  - Compute $y_{k+1} = \frac{1}{n_k} \sum_{x_i \in s_h(y_k)} x_i,\ k \leftarrow k+1$
  - Assign $z_j = \left(\mathbf{x}_j^s, \mathbf{y}_{conv}^r\right)$
- Let $\{z_j\}_{j=1..n}$ be the *d*-dimensional original and filtered image in the spatial-range domain. The *s* and *r* denote the spatial and range parts of the vectors, respectively. The last assignment specifies that the filtered data at the spatial location of x_j will have the range components of the points of convergence y_{conv}.
-

5.3.5. Step 5 : KMeans Algorithm

The converted low dimensioned grouped pixels from mean shift algorithm are then used as input to K Means algorithm. The K Means clustering generates a specific number of disjoint, flat (non-hierarchical) clusters. It is well suited for generating globular clusters. The K Means method is an unsupervised, non-deterministic and iterative method with the following properties.

- There are always K clusters.
- The clusters formed do not overlap and are non-hierarchical.
- Each and every element of a cluster is nearest or closer to its cluster than any other cluster because closeness does not always involve the 'center' of clusters.

The traditional K Means algorithm is given in Figure 5.6. The most widely used convergence criteria (objective function) for the K Means algorithm is minimizing the SSE (Sum of Squared Error).

$$\mathrm{SSE} = \sum_{j=1}^{k} \sum_{x_i \in c_j} \| x_i - \mu_j \|^2 \tag{5.23}$$

where $\mu_j = \frac{1}{n_j} \sum_{x_i \in c_j} x_i$ denotes the mean of cluster c_j and n_j denotes the number of instances in c_j. The KMeans algorithm always converges to a local minimum and depends on the starting cluster centroids. The K Means algorithm updates cluster centroids till local minimum is found.

Step 1 : Begin with a decision on the value of k = number of clusters

Step 2 : Put any initial partition that classifies the data into k clusters.

1. Take the first k data as single-element clusters
2. Assign each of the remaining (N-k) data to the cluster with the nearest centroid. After each assignment, recompute the centroid of the gaining cluster.

Step 3 : Take each data in sequence and compute its distance from the centroid of each of the clusters. If a sample is not currently in the cluster with the closest centroid, switch this sample to that cluster and update the centroid of the cluster gaining the new sample and the cluster losing the sample. The distance metric used is Euclidean distance (Equation 3.2).

$$d_{ij} = \sqrt{\sum_{k=1}^{n} (x_{ik} - x_{jk})^2} \tag{3.2}$$

Step 4 : Repeat step 3 until convergence is achieved, that is, until a pass through the training sample causes no new assignments.

Figure 5.6 : Traditional K Means Algorithm

Thus, the sequential steps of WCF method for segmenting the leaf image from its background can be summarized as follows.

Step 1 : Transform image from RGB color space to L*u*v color space and decompose each color channel L, u and v.

Step 2 : Obtain gray scale image from RGB color space.

Step 3 : Use wavelet frame transformation to decompose the gray scale image into sub-bands (LL, LH, HL, and HH) and use the LH and HL subbands for further processing.

Step 4 : Texture Feature Extraction

(a) Calculate the median energy for LH and HL subbands coefficients in a local window. The size of the window should be large enough to capture the local texture characteristics.

(b) Use K-means clustering algorithm to classify the energy values in two classes for each subband.

(c) Using these information, classify the pixels in textured region into four groups, namely, smooth, vertical, horizontal and complex.

Step 5 : Generate the feature vector such that every pixel in the image has p-dimensional feature vector which includes spatial (x,y), color (gray-level or L*u*v values) and texture (smooth, vertical, horizontal or complex) information.

Step 6 : Filter the image using mean shift algorithm in higher dimensional feature space, which includes spatial, color and texture information. The filtering operation can be controlled by setting the spatial window radius (hs) and color range (hr). The filter output (convergence point in mean shift algorithm) is determined by color as well as texture information unlike in standard mean shift filtering. This provides better discrimination between regions where colors are similar but texture is different.

Step 7 : Segment the output image using K-Means clustering algorithm.

5.4. EXPERIMENTAL RESULTS

This section presents the segmentation results obtained by the wavelet based segmentation model and the proposed WCF Method. The evaluation is performed by analyzing the speed of segmentation and visual analysis.

To analyze the time complexity of the proposed algorithm, the time taken to segment the leaf image was calculated in seconds and the result is presented in Figure 5.7.

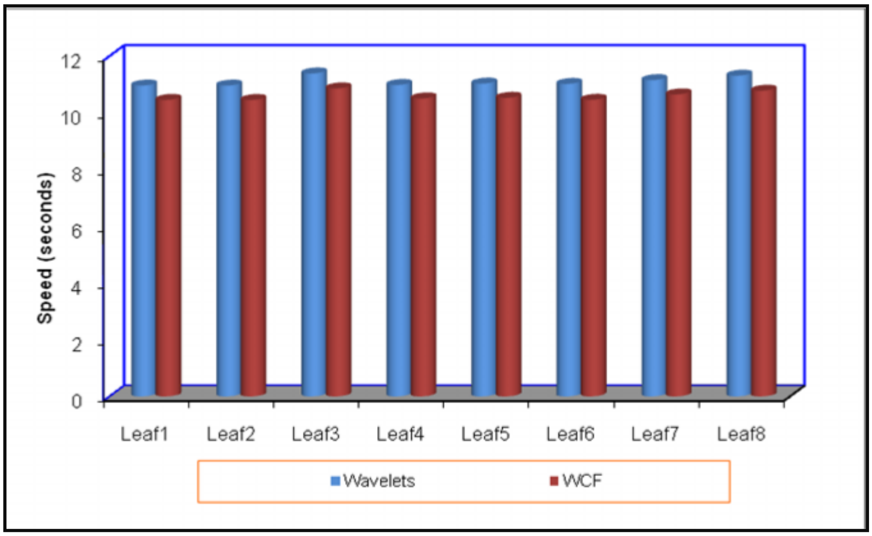


Figure 5.7 : Speed of Segmentation (Seconds)

The speed of the proposed algorithm is also increased considerably when compared with the existing wavelet based algorithm. On an average the existing algorithm took 11.08 seconds to segment the leaf image, while the proposed algorithm took only 10.56 seconds to extract the leaf from its background. This shows that the enhanced WCF algorithm is efficient in the process of segmentation.

Figures 5.8 and 5.9 show the visual results of the 8 test images, (4 belonging to standard dataset and the rest belonging to real dataset) respectively and the results of the entire dataset is shown in **Appendix C**.

From the visual results, it is clear that the proposed WCF method that enhances the operation of wavelet based segmentation method through the use of texture based color features, mean shift filtering algorithm and K-means clustering algorithm produces improved segmentation results.

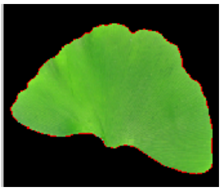
Enhanced Images	Wavelet	WCF
		
Leaf 1		
		
Leaf 2		
		
Leaf 3		
		
Leaf 4		

Figure 5.8: Segmentation Results of Standard Dataset

Enhanced Images	**Wavelet**	**WCF**
		
Leaf 5		
		
Leaf 6		
		
Leaf 7		
		
Leaf 8		

Figure 5.9 : Segmentation Results of Real Dataset

5.5. CHAPTER SUMMARY

The second step of CAP-LR system is the process of segmenting the leaf image from its background. For this purpose, the wavelet based method was enhanced to use color and texture features along with a filtering method and clustering algorithm. The filtering method used was mean-shift filtering algorithm, while the clustering algorithm used was K-Means algorithm. From the results, it can be seen that the proposed algorithm performs segmentation to separate foreground and background to extract the ROI (leaf). The next step of CAP-LR system is the feature extraction and classification task, which is explained in chapter **Feature Extraction and Selection.**