CHAPTER 6

FEATURE EXTRACTION AND SELECTION

Plant recognition and identification is very demanding in biology and agriculture as new plant discovery and the computerization of the management of plant species become more popular. The identification and recognition is a process resulting in the assignment of each individual plant to a descending series of related plants in terms of their common characteristics. The process is very time-consuming as it has been mainly carried out by botanists. Computer assisted plant recognition is still a very challenging task in computer vision due to the lack of proper models or representation schemes, a large number of variations of the plant species, and imprecise image preprocessing techniques, such as edge detection and contour extraction. The focus of computerized plant recognition is based on the extraction of stable features of leaves.

After enhancing and extracting the leaf image, the next step of CAP-LR is feature extraction and selection. The main aim of this step is two folds.

(i) Feature Extraction : Converts the image data into a representation that allows comparisons between leaf images by extracting leaf properties

(ii) Feature Selection: Selects the appropriate subset of features from the extracted features.

The task of the feature extraction is to obtain the most relevant information from the original data and represent that information in a higher dimensionality space. Feature selection is an important task that allows the determination of the most relevant features for pattern recognition. The objective of feature selection is three-fold:

- To improve the prediction performance of the predictors
- To increase the sped of the classification process
- To provide a better understanding of the underlying process that generates the data

The goal of feature selection is to reduce the dimensionality of vectors associated to patterns by selecting a subset of attributes smaller than the original. Further the task of feature selection is to improve the classifier performance by eliminating redundant features. In Phase III of the study, several features which improve the performance of the underlying classification model were selected. The features selected are categorized into five major groups. The details of the features selected in each group are shown in Table 6.1. Details regarding the same are discussed in the following subsections.

TABLE 6.1

FEATURES EXTRACTED

• **Geometric Features (3)** ▪ Eccentricity ▪ Extent ▪ Orientation • **Color (4)** • Mean • Standard Deviation • Skew • Kurtosis • **Texture (4)** • Energy • Entropy • Homogeneity • Variance • **Fractal Features (3)** • Average Fractal Dimension • Standard Deviation FD • Lacunarity	• **Leaf Specialized Features (14)** 1. Diameter 2. Physiological Length (PL) 3. Physiological Width (PW) 4. Area 5. Perimeter 6. Smooth Factor 7. Aspect Ratio 8. Form Factor 9. Rectangularity 10. Narrow factor 11. Perimeter ratio of diameter 12. Perimeter ratio of PL and PW 13. Ripple 14. Vein features

6.1. GEOMETRIC FEATURES

Three geometric features are considered in this study. They are Eccentricity, Extent and Orientation. The eccentricity is defined as the ratio of the length of main inertia axis of the ROI (EA) to the length of minor inertia axis of the ROI (EB) (Du *et al.*, 2007). This feature has the ability to differentiate the round and long leaf structure and is calculated using Equation (6.1).

$$E = \frac{E_A}{E_B} \tag{6.1}$$

Image extent is defined as the scalar that specifies the ratio of pixels in the region to pixels in the total bounding box. It is computed as the Area divided by the area of the bounding box. Here, area is defined as the actual number of pixels in the region and area of the bounding box is the smallest rectangle in the image. It is defined by a 4-tuple, (x_0, y_0, x_1, y_1) where (x_0, y_0) is the top left (northwest) corner of the rectangle, and (x_1, y_1) is the bottom right (southeast) corner. Generally, the area described by a bounding box will include the point (x_0, y_0), but it will not include the point (x_1, y_1) or the row and column of pixels containing the point (x_1, y_1).

Leaf orientation is defined as the angle between the axis exhibiting the minimum moment of inertia and the horizontal movements (Tzionas *et al.*, 2005). It can be calculated using the following Equation (6.2).

$$I(\theta) = \sum\sum[n - \bar{n})\cos\theta - (m - \bar{m})\sin\theta]^2 \tag{6.2}$$

where m, n $\in \Re$,resulting in the following angle θ as shown in Equation (6.3).

$$\theta = \frac{1}{2}\tan^{-1}\left[\frac{2\mu_{1,1}}{\mu_{2,0} - \mu_{0,2}}\right] \tag{6.3}$$

6.2. COLOR FEATURES

For RGB color space, the three features are extracted from each plane R, G, and B. The following statics are used to capture those moments. It is calculated using M and N which represents the dimension and total number of pixels in the image. P_{ij} is the value of color on i_{th} column and j_{th} row. The other moment used is termed as Standard Deviation. The standard deviation is the square root of the variance of the distribution. Another moment used for extracting features is known as Skewness. It is the measure of the degree of asymmetry in the distribution. The Equations for calculating the four selected color features are presented in Table 6.2.

TABLE 6.2

COLOR FEATURES

S.No.	Feature Name	Feature Calculation
1.	Mean	$\mu = \frac{1}{MN}\sum_{i=1}^{M}\sum_{j=1}^{N} P_{ij}$
2.	Standard Deviation	$\sigma = \sqrt{\frac{1}{MN}\sum_{i=1}^{M}\sum_{j=1}^{N}\left(P_{ij}-\mu\right)^2}$
3.	Skewness	$\theta = \frac{\sum_{i=1}^{M}\sum_{j=1}^{N}\left(P_{ij}-\mu\right)^3}{MN\sigma^3}$
4.	Kurtosis	$\gamma = \frac{\sum_{i=1}^{M}\sum_{j=1}^{N}\left(P_{ij}-\mu\right)^4}{MN\sigma^4}$

6.3 TEXTURE FEATURES

Texture of a plant may be due to having many veins in different directions or parallel lines of different colors. Classical Gabor filters gives rise to important difficulties when implemented in multiresolution space

(Zhang *et al.*, 2004; Fischer *et al.*, 2006). The texture features extracted from the segmented leaf images are energy, entropy, homogeneity and variance.

The energy of an image is calculated as described below. To calculate energy (also called Uniformity) first the Angular Second Moment (ASM) is to be calculated. Both ASM and Energy use each P_{ij} as a weight for itself. High values of ASM or Energy occur when the window is very orderly.

$$\text{ASM equation} = \sum_{i,j=0}^{N-1} P_{i,j}^{2} \tag{6.3}$$

Energy is now calculated as the square root of the ASM as shown in Equation (6.4).

$$\text{Energy} = \sqrt{\text{ASM}} \tag{6.4}$$

The entropy is calculated using the formula

$$\text{Entropy} = \sum_{i,j=0}^{N-1} P_{i,j}(-\ln P_{i,j}) \tag{6.5}$$

The Homogenity feature is calculated as

$$\text{Homogeneity} = \sum_{i,j=0}^{N-1} \frac{P_{i,j}}{1+(i-j)^{2}} \tag{6.6}$$

Homogeneity is the most commonly used measure that increases with lesser contrast in the image window.

The variance for the horizontal and vertical directions is calculated as below.

$$\text{Variance} = \sigma_i^2 = \sum_{i,j=0}^{N-1} (i-\mu_i)(P_{i,j}),\ \sigma_j^2 = \sum_{i,j=0}^{N-1} (j-\mu_j)(P_{i,j}) \tag{6.7}$$

6.4. FRACTAL FEATURES

Fractals when used as image features and applied for pattern recognition problems have to be efficient and has been applied in many areas including video coding, medical imaging, natural texture analysis (Chan, 1990), classification (Sato *et al*., 1999), etc. The fractal dimension has also been used with wavelet analysis in feature extraction (Tang and Tao, 1999). Using fractals as a feature for plant identification through leaf image is a new concept that has been applied in the present study.

Fractal dimension expresses the (constant) rate at which new geometrical details appear as one zoom in an object. In this respect, the fractal dimension is a measure of the geometric irregularity of the object. Several estimators have been developed to compute this characteristic from raw data. These estimators were usually applied on 2D images, including in application to leaf images, leading to inaccurate estimations. Box counting method has been extensively used to estimate fractal dimension of objects embedded in the plane.

From the box dimensions and box count, three fractal features are extracted. Two features calculate the average and standard deviation of the fractal dimensions. These two features are widely used in image processing.

The fractal dimension does not characterize completely the geometric properties of fractal objects. For instance, even if plants P1 and P2 have the same identical fractal dimension, say for example, 2, they still show geometries with different gap structures. Lacunarity has been introduced as a complementary measure to reveal such characteristics. It is defined as the relative moment of order 2 of the distribution of local mass at scale δ when x varies inside the object bounding box (i.e. not only on the object) and is calculated using Equation (6.8).

$$L(\delta) = \frac{E(M(x,\delta)^2)}{E(M(x,\delta))^2} \qquad (6.8)$$

6.5. LEAF SPECIALIZED FEATURES

This section presents the extraction process of various leaf related features. A total of 14 leaf specialized features are considered in this research and a description of each of these features are presented below.

6.5.1. Vein Feature

The information of leaf veins plays an important role in identifying living plants. Botanical interpretation of the leaves needs data to be in a form that allows comparison between different leaves. Vein features allows this by transforming leaf image data into a set of coordinates affixed to a leaf (Zheng and Wang, 2010). The leaf vein system is a hierarchical entity with a main vein and side veins of possibly several orders. Plant leaves are well structured objects consisting of line-like veins and some parts in between as shown in Figure 6.1.

Figure 6.1 : Veins of a Leaf

The leaf vein feature extraction consists of two steps. The first step extracts or identifies the veins in the input leaf image while the second step extracts the required vein feature.

- **Step 1 : Vein Extraction**

The algorithm begins by transforming the color input leaf image into gray scale image. In the gray scale image, it was observed that the gray color overlaps in the leaf vein and background, that is, the pixels with the same gray value can belong to either the background or leaf vein. This problem is handled by the use of gray scale morphological operations.

There are six mathematical morphological operations, namely, erosion, dilation, opening, closing, bot-hot transformation and top-hat transformation (Park *et al.*, 2008). Let G be the converted gray scale image of input color leaf image and let 'S' be the structuring element. Table 6.3 describes the six morphological operations. From the gray scale image, it can be seen that the veins of the leaf are darker than its leaf background. This fact is exploited by applying the bot-hat transformation to obtain the gray difference between the leaf vein and its background. Then the following (Equation 6.9) morphological operation is performed.

$$R = (G \bullet S - G) - (G - G \circ S) \tag{6.9}$$

As the values of the gray differences are small between most of the parts in a leaf vein and their backgrounds, normally less than 60, the morphological operation resultant image R appears to be very dark and the effect of the gray morphology processing cannot be seen from the image R. To solve this problem, R is transformed to its inverse image. Thus the background leaf image appears almost white and the difference between vein and its background is well separated, thus solving the problem of gray scale color value overlapping.

To further optimize the veins of the leaf, a linear intensity adjustment in Equation (6.10) is applied.

$$R(x,y) = 255\frac{R(x,y) - \min_v}{\max_v - \min_v} \tag{6.10}$$

where $\min_v$ and $\max_v$ are the minimum and maximum gray scale values in R. This effect of applying the intensity adjustment operation makes the veins appear almost black and is well-separated from their background.

TABLE 6.3

MORPHOLOGICAL OPERATIONS

Erosion operation	$(G\Theta S)(s,t)=\min\{G(s+x,t+y)- S(x,y) \mid (s+x, t+y)\in DG, (x,y)\in DS$
Applying erosion operation on a gray image will reduce the brightness of the image and the darker parts in the original image will expand in the resultant image.	
Dilation operation	$(G\oplus S)(s,t)=\max\{G(s+x,t+y)+S(x,y) \mid (s+x,t+y)\in DG, (x,y)\in DS$
Applying dilation operation on a gray image will enhance the brightness of the image and the brighter parts in the original image will expand in the resultant image.	
Opening operation	$G \circ S = (G \Theta S) \oplus S$
Applying opening operation on a gray image can eliminate the little brighter regions in the image or weaken the brightness of these regions.	
Closing operation	$G \bullet S = (G \oplus S) \Theta S$
Applying closing operation on a gray image can eliminate the little darker regions in the image or increase the brightness of these regions.	
Top-hat transformation	$G – G \circ S$
Applying top-hat transformation on a gray image can extract the little brighter regions in the image or get the gray differences between these regions and their backgrounds.	
Bot-hat transformation	$G \bullet S - G$
Applying bot-hat transformation on a gray image can extract the little darker regions in the image or get the gray differences between these regions and their backgrounds.	

The next step is the process of vein extraction. For this purpose, the otsu method of thresholding is used. Using otsu method, the veins are extracted by applying Equation (6.11).

$$d(x,y) = \begin{cases} 1 & R(x,y) \geq t \\ 0 & R(x,y) < t \end{cases} \qquad (6.11)$$

where t is the threshold and d(x,y) is the resultant binary image with white color representing the vein and black color representing the leaf background.

One problem of applying otsu threshold method is the appearance of random (isolated) points and broken veins (discontinuous lines). A 100% zoomed example of R is shown in Figure 6.2.

Figure 6.2 : Appearance of Random Points and Broken Veins

To solve these problems two simple procedures are used. A search procedure is initiated to search the black pixels. When a black pixel is found, the number of white pixels in its neighbourhood vicinity is counted. If this number is greater than a specified number (five used during experimentation), then the current pixel is considered as a leaf vein and is changed to white color (that is, its gray scale value is set to 1). Let this result be denoted as d'. To eliminate the random or isolated points, a morphological opening operation is calculated through the Equation (6.12).

$$e = d' \circ S \qquad (6.12)$$

The result after handling the random and broken veins in Figure 6.2 is presented in Figures 6.3 and 6.4.

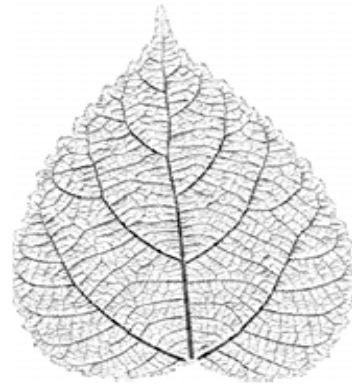
Figure 6.3 : Linking Broken Veins

Figure 6.4 : Elimination of Random or Isolated Points

The specified number in linking the discontinuous lines is related to the width of b structuring element in gray morphology processing. If the width is n, the specified number is (n-1)/2. The width of b structuring element in eliminating the isolated points is also related to the width of b structuring element in gray morphology processing. If the latter is n, the former should be n-2. A step-by-step illustration of the leaf vein extraction process is shown in Figure 6.5.

Input Leaf

Gray Scale



Gray Scale Morphology

Optimize Veins

Extract Veins

Processed Image

Leaf Vein

Figure 6.5 : Leaf Vein Extraction Process

- **Step 2 : Vein Feature Extraction**

The next step after vein identification is to extract its features that can be used to recognize the leaf and thus identify the plant. From the veins identified, the main vein is identified. This can be performed by measuring the distribution of the leaf veins. For this purpose a projection histogram in the horizontal and vertical directions is applied. First the main vein is extracted using projections in horizontal direction using a maximum point of histogram, while the leaf vein image is rotated 180 degrees. After the main vein extraction, the next step decides the direction of the leaf through projections in the vertical direction. Using this information, the centroid of the detected leaf region is calculated by the Equation (6.13).

$$C(x,y) = C\left(\frac{1}{N}\sum_{n=1}^{N} x_n, \frac{1}{N}\sum_{n=1}^{N} y_n\right) \tag{6.13}$$

where C(x, y) is the centroid coordinate of the leaf region image and N is the number of pixels on the detected leaf region. The distance is calculated by measuring the centroid of the leaf region to all points on the leaf contour as follows:

$$D(i) = \sqrt{\left|C_x - E(i)_x\right|^2 + \left|C_y - E(i)_y{}^2\right|} \tag{6.14}$$

where D(i) is the distance between the centroid of the leaf region and the i^{th} leaf contour pixel. C_x and C_y are the coordinates of the centroid of the leaf region and $E(i)_x$ and $E(i)_y$ are the coordinates of i^{th} leaf contour pixel.

Using the Projection Histogram, the longest vein (main vein) is identified, from which the following vein feature is extracted.

Ramification is the divergence of main veins of the leaf into small secondary veins. The number of ramifications of the main leaf is mainly used to measure the complexity of the leaf venation. By watering the main vein

from the end point of leaf stem, the number of the ramifications is calculated as the diffluent times of the water when it flows along the main vein. This is performed using a water filling algorithm proposed by Zhou and Huang (2008). Using this information the vein feature is calculated by the Equation (6.15).

$$\text{Vein Feature} = \frac{N_R}{L} \tag{6.15}$$

where N_R is the number of ramifications and L is the length of the main vein.

6.5.2. Other Leaf Features

Apart from the above feature several other features are also extracted, which use both leaf and vein details (Wu *et al.*, 2007). They are described below.

- **Diameter**: The diameter is defined as the longest distance between two points on the margin of the leaf. It is denoted as D.
- **Physiological Length:** The distance between the two terminals is defined as the physiological length. It is denoted as L_p.
- **Physiological Width:** Drawing a line passing through the two terminals of the main vein, one can plot infinite lines orthogonal to that line. The number of intersection pairs between those lines and the leaf margin is also infinite. The longest distance between points of those intersection pairs is defined at the physiological width. It is denoted as W_p. Since the coordinates of pixels are discrete, two lines are considered orthogonal if their degree is 90° ± 0.5°. The relationship between physiological length and physiological width is illustrated in Figure 6.6.
- **Leaf Area:** The value of leaf area is one of the easiest feature to evaluate. It is the process of just counting the number of pixels of binary value 1 on the smoothed leaf image. It is denoted as A.

- **Leaf Perimeter:** Denoted as P, leaf perimeter is calculated by counting the number of pixels consisting of leaf margin.

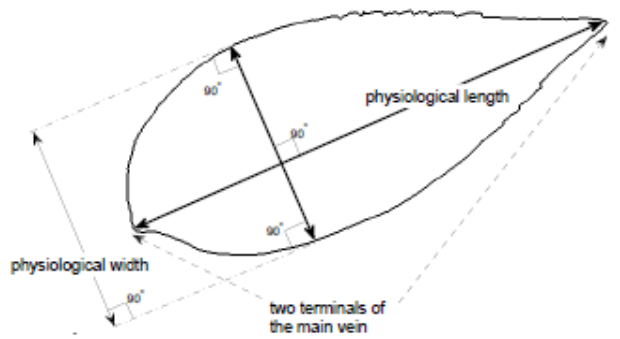


Figure 6.6 : Relationship between Physiological Length and Physiological Width

6.5.3. Morphological Leaf Features

Morphological features are normally derived from the geometrical leaf features. The various morphological features that can be extracted from the plant for leaf recognition are listed below:

- **Smooth factor:** The effect of noise in image area is used to describe the smoothness of leaf image. In normal practice, smooth factor is defined as the ratio between area of leaf image smoothed by 5 x 5 rectangular averaging filter and the one smoothed by 2 x 2 rectangular averaging filter.

- **Aspect ratio:** The aspect ratio is defined as the ratio of physiological length Lp to physiological width W_p (L_p/W_p).

- **Form factor:** This feature is used to describe the difference between a leaf and a circle. It is defined as $4\pi A/P^2$, where A is the leaf area and P is the perimeter of the leaf margin.

- **Rectangularity:** Rectangularity describes the similarity between a leaf and a rectangle. It is defined as L_pW_p/A, where L_p is the physiological length, W_p is the physiological width and A is the leaf area.

- **Narrow factor:** Narrow factor is defined as the ratio of the diameter D and physiological length L_p (D/L_p).

- **Perimeter ratio of diameter:** Ratio of perimeter to diameter, representing the ratio of leaf perimeter P and leaf diameter D, is calculated by P/D.

- **Perimeter ratio of physiological length and physiological width:** This feature is defined as the ratio of leaf perimeter P and the sum of physiological length L_p and physiological width W_p ($P/(Lp + Wp)$).

6.6. RIPPLE FEATURE

The ripples features describe the fluctuation of the leaf boundary, which can be calculated by finding the differences between the leaf image and the averaged boundary leaf image (Lurstwut and Pornpanomchai, 2011). It is a feature that is used frequently to identify diseases in seeds and fruits and, in this study, is proposed as a leaf featue. The averaged leaf boundary can be calculated by the leaf boundary coordinates in order to find the range of the boundary. The range of boundary formula is calculated by the following Equation (6.16).

$$R = B / 10 \tag{6.16}$$

where R = range of leaf boundary, B = length of leaf boundary. After obtaining the differences, a morphological opening operation is performed to filter out the narrow area and remove any object that has fewer than 10 pixels. From this ripple image, two ripple features are calculated. The first is the Ripples counting (The ripples are the remaining objects in the ripple image) and second is the Ripples pixels counting (This process counts all the white pixels in all ripples).

Thus, the feature extraction process of CAP-LR extracts three geometric features, four color features, four texture features, three fractal features and fourteen leaf specialized features. This study, in order to increase the accuracy

of leaf recognition for plant identification, proposes to combine these features. These features form the feature space to be used by the classifier. The research work analyzes the advantages obtained by combining the selected features and compares them with the single feature. The method of combining features is presented in the next section. The proposed method performs fusion and selection simultaneously, thus improving the time parameter of the operation.

6.7. PROPOSED FEATURE FUSION AND SELECTION ALGORITHM

Feature selection is an important task that allows the determination of the most relevant features for pattern recognition. The extracted features are normalized or reduced by selecting appropriate features to improve the classification accuracy (Pornpanomchai *et al.*, 2011). A good feature selection results in

- Faster training and better generalization.
- Removes redundant leaf images.
- Focuses recognition to a small set of properties.
- Displays the final classified outcome.

In general, most of the selected features are either partially or completely irrelevant or redundant to the classified target. The main goal of feature selection is to identify only those features which can provide sufficient information for accurate classification and can provide maximum discrimination between classes.

While fusion feature set increases the accuracy of recognition, one major drawback is the curse of high dimensionality, which increases the time complexity. Reducing the dimensions of the feature space not only reduces the computational complexity, but also increases the estimated performance of the classifiers.

The study proposes a technique that performs both the operations together by combining the results of Genetic Algorithm (GA) based feature fusion combined with feature selection and Kernel based Principal Component Analysis (KPCA) to obtain a feature space that can be forwarded to train the classifier for leaf recognition for plant identification. This section first presents GA based feature fusion combined with feature selection algorithm followed by KPCA algorithm. Finally, the proposed method that is used to combine the advantages of both these algorithms is presented.

The method used to combine the GA based algorithm and KPCA is illustrated in Figure 6.7. The goal here is to effectively utilize useful information from different feature selection methods to select better feature subsets with smaller size and/or higher classification performance in comparison with the existing methods.

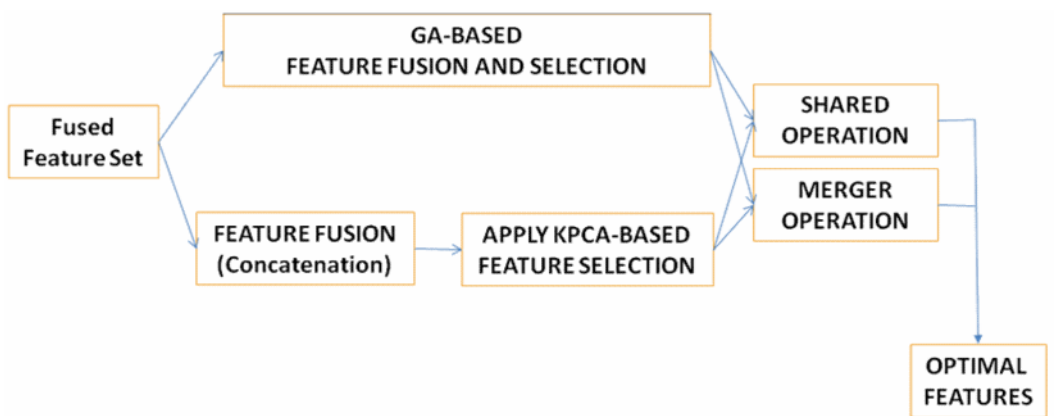


Figure 6.7 : Proposed Fusion and Selection Algorithm

The proposed feature fusion based feature selection method consists of two stages. The first stage uses a GA-based feature fusion and selection algorithm to combine the various categories of features and to select appropriate feature subset.

The second method uses a simple concatenation method where all features are joined together with the target label and the high dimensionality

and feature subset selection is handled using a KPCA based algorithm. The results of the two algorithms are further combined with the following operators

(i) Using Shared Operator ($\cap$)

(ii) Using Merger Operator ($\cup$)

The shared operator performs the intersection operation and selects only those features that are selected by both the algorithms. The merger operator on the other hand works similar to union operation and selects all distinct features from both the algorithms.

6.7.1. FEATURE FUSION AND SELECTION USING GENETIC ALGORITHM

The steps involved in the GA based feature fusion and selection algorithm is given in Figure 6.8 (Tan *et al.*, 2008). The main idea behind this approach is to combine the results of feature selection algorithms on different datasets to form a feature pool, which are then fused or combined using Genetic Algorithm.

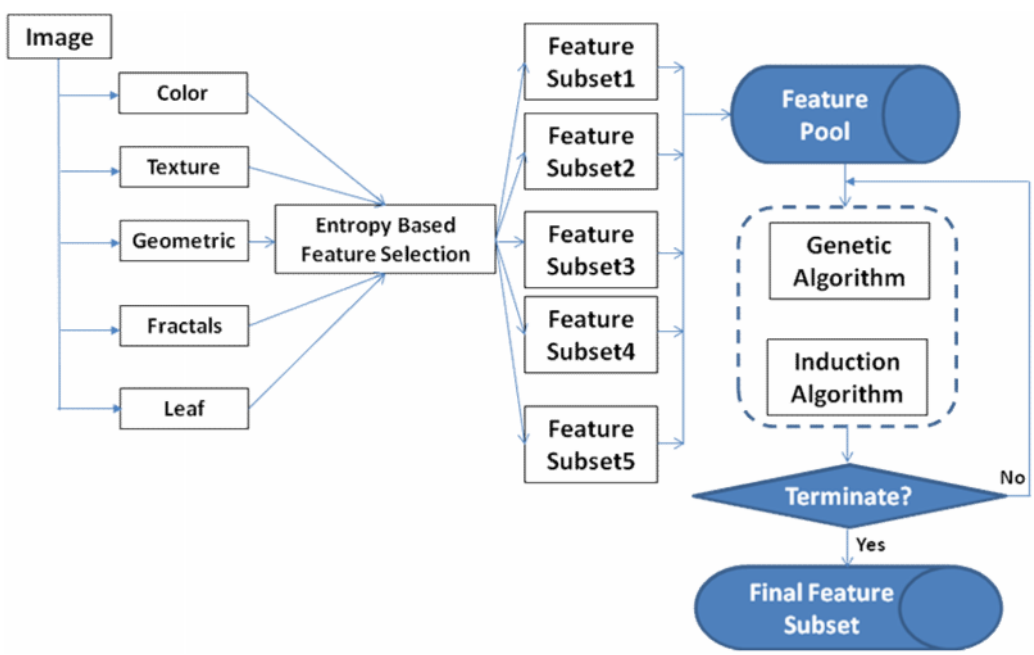


Figure 6.8 : Feature Selection and Fusion Algorithm

In the above diagram, feature pool is defined as the collection of candidate features selected by the entropy based feature selection method. This is forwarded as input to GA to find the nearest feature subset. In general, GA based feature selection algorithm selects features related to only one category of subsets. In this study the GA based algorithm is used to fuse the features selected from different categories.

To construct the feature pool, the entropy based feature selection algorithm is used. The entropy-based method (Dash and Liu, 1999) depends on the fact that entropy is lower for orderly configurations and higher for disorderly configurations. The basic idea of this method is to filter out those features whose expression distributions are relatively random. For the remaining features, this method can automatically find some cut points in these features' value ranges such that the resulting expression intervals of every feature can be maximally distinguished. If every expression interval induced by the cut points of a feature contains only the same class of samples, then this partitioning by the cut points of this feature has an entropy value of zero. This is an ideal case. The smaller the feature's entropy will result in more discriminatory value. The obtained entropy values are sorted in ascending order and only those features with lowest entropy values are considered.

The entropy measure of a data set of N instances is calculated using Equation (6.17).

$$E = -\sum_{i=1}^{N}\sum_{j=1}^{N}(s_{ij} \text{ x } \log s_{ij} + (1-s_{ij}) \text{ x } \log(1-s_{ij})) \qquad (6.17)$$

where $s_{ij} = e^{-\alpha x D_{ij}}$ and $\alpha = \frac{-\ln 0.5}{\bar{D}}$. Here S_{ij} is the similarity measure based on distance between two instances x_i and x_j with all numeric features (similarity between two instances with nomial features is measured using Hamming distance) and α is a parameter. D_{ij} is the Euclidean distance between the two instances and $\bar{D}$ is the average distance among the instances.

After selecting relevant features, a ranking process that uses a threshold value to select the top n-ranking features as optimal features will be used. The selection of 'n' in this method is very critical. A very low or very high value will have a negative impact on the result of feature selection and thus on the result of classification. To overcome this problem, the GA is used in this study.

- **Genetic Algorithm**

Genetic algorithms (GAs) provide a learning method inspired by evolutionary biology. GAs are the most popular class of evolutionary algorithms that use mechanisms such as reproduction, mutation, crossover (also called recombination), natural selection and survival of the fittest to simulate biological evolution (Holland, 1992).

Genetic algorithms have been successfully applied to a wide variety of scientific and engineering optimization or search problems. They can search spaces of hypotheses containing complex interacting parts, where the impact of each part on an overall hypothesis is difficult to model (Mitchell, 1997). The relative insensitivity of GAs to noise, and the requirement of no domain knowledge make them a powerful tool to optimize the process of classification, especially when the domain knowledge is costly to exploit or unavailable (Vafaie and Jong, 1992). Many researches demonstrate the advantages of the GAs for feature selection (Lu *et al.*, 2008; Bhanu and Lin, 2003; Kharrat *et al.*, 2011).

Genetic algorithms begin the search for solutions in a population of initial hypotheses that traditionally are generated at random. Each hypothesis, called an individual or a chromosome, represents a potential solution of the problem. Individuals are encoded as bit strings whose interpretation depends on applications(Cao,2003). Typically, individuals are represented in binary as strings of zeros and ones. The initial population then evolves in generations. In each generation, every individual of the current population is evaluated according to the fitness function F, which is a predefined numerical measure

for the problem at hand. A new population is generated by stochastically selecting the current fittest individuals. Some of the selected individuals are modified to produce new offspring individuals by mutating and recombinating parts of them.

Some of these selected individuals are passed on to the next generation intact. The new population is then used in the next iteration of the algorithm. Random search strategies powered by the genetic operators (mutation and crossover) are designed to move the population away from local optima where many algorithms (e.g., greedy hill climbing) face hindrance. In the GA based image fusion and selection method, there are several operations that need to be determined. They are chromosome encoding, fitness function, selection, crossover and mutation.

- **Chromosome Encoding**

In chromosome encoding, a binary encoding scheme is used where a binary bit string represents an individual. Each individual represents a feature subset. The individuals are encoded by L-bit binary vectors (Figure 6.9). The bit with value 1 in a vector represents the corresponding feature being selected, while the bit with value 0 means the opposite. The length of each chromosome is determined by the number of features N. Thus, in the encoding scheme used, the chromosome is a bit string whose length is determined by the number of parameters in the image. Each parameter is associated with one bit in the string. If the ith bit is 1, then the ith parameter is selected. Otherwise that component is ignored. Each chromosome thus represents a different parameter subset.

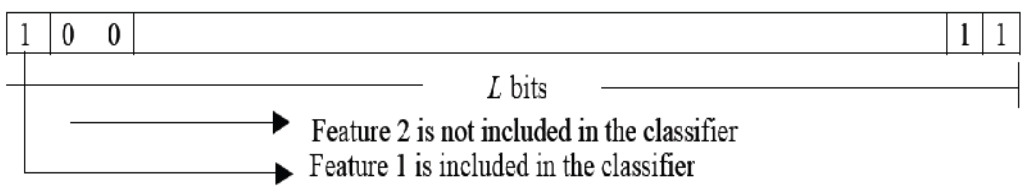


Figure 6.9 : Encoding of Feature Subset in GA - A L-Dimensional Binary Vector

- **Fitness function**

The genetic algorithm is designed to optimize two objectives:

(i) Maximize classification accuracy of the feature subset and

(ii) Minimize the number of features selected.

For this purpose, the following fitness function (Equation 6.18) is used.

$$F = w * c(x) + (1 - w) * (1/s(x)) \quad (6.18)$$

where x is a feature vector representing a feature subset selected and w is a parameter between 0 and 1. The function is composed of two parts. The first part is a weighted classification accuracy c(x) from the classifier and the second part is weighted size s(x) of the feature subset represented by x.

For a given w, the fitness of an individual x is increased as the classification accuracy of the x increases, and decreased as the size of x increases. Increasing 'w' indicates a priority given to classification accuracy over size. On the other hand, reducing the value of 'w' will give more penalties on the size of x. By adjusting w, a tradeoff between the accuracy and the size of the feature subset obtained can be achieved.

- **Induction algorithm**

The genetic algorithm is independent of the inductive learning algorithm used by the classifier. Different induction algorithms, such as Naïve Bayes, artificial neural network, and decision trees can be flexibly incorporated into the proposed method. In this paper, a two stage hybrid classification is used.

- **Genetic operators**

Three genetic operators were used during feature selection. They are selection, crossover and mutation.

Roulette wheel selection is one of the most popular selection methods for genetic algorithm and is used by the study. Roulette wheel selection probabilistically selects individuals from a population for later breeding. The probability of selecting individual h_i is determined by:

$$P(h_i) = \frac{F(h_i)}{\sum_{i=1}^{p} F(h_i)} \qquad (6.19)$$

where $F(h_i)$ is the fitness value of h_i. The probability that an individual will be selected is proportional to its own fitness and is inversely proportional to the fitness of the other competing hypothesis in the current population. The router wheeler procedure used is given below (Figure 6.10).

1. Calculate accumulative probabilities for the ith individual using Equation (6.20).

$$p_i = \sum_{j=1,t} P(j) \quad , j = 1...t \qquad (6.20)$$

2. Generate a random number r within [0, 1]

3. Select the ith individual if $p_{i-1} < r < p$

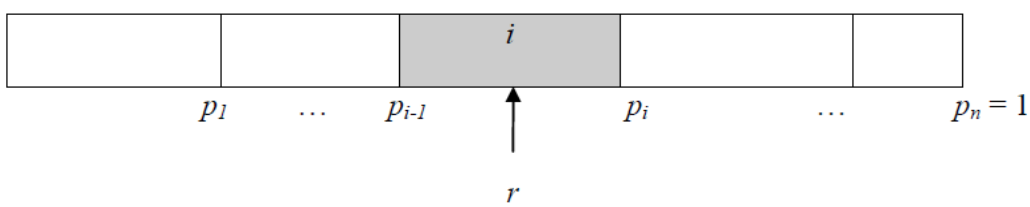


Figure 6.10 : Roulette Wheel Selection

Mutation and crossover are two of the most commonly used operators with genetic algorithm that represent individuals as binary strings. Mutation operates on a single string and generally changes a bit at random. Crossover operates on two parent strings to produce two offspring.

6.7.2. The KERNEL PRINCIPAL COMPONENT ANALYSIS

Most of the studies use Principal Component Analysis (PCA) (Cao *et al.*, 2003) method, which is a linear feature extraction method, to solve the problem of feature selection during classification. However, inspite of its popularity, the PCA algorithm can extract only linear features while the chart features contain both linear and non-linear features. To solve this problem, Kernel Principal Component Analysis (KPCA) (Scholkopf *et al.*, 1998a, 1998b), a non-linear presentation of linear PCA, is used in this study. KPCA maps the data space of inputs to a high dimension eigenspace, and transforms to calculate the eigenvalue and eigenvector of the kernel matrix. The projection of inputs in eigenvalue is transformed to calculate the linear combination of kernel function, thus the computation is reduced. The process of KPCA algorithm is given below.

Given an input data set denoted by matrix $X_{M \times m}$, where X consists of features dataset $\{x_i\}$ and i= 1...m, where $x_i \in R$ and $\sum_{i=1}^{m} x_i = 0$. The KPCA algorithm first performs non-linear transformation via a non-linear function κ to map the original input vector x_i to a high dimensional feature space F and then calculates linear PCA in $\kappa(x_i)$, whose dimension is assumed to be larger than m . The mapping function are represent as $\kappa: x \rightarrow \kappa(x) \in F$, where $\kappa(x_i)$ is sample of F and $\sum_{i=1}^{m} \kappa(x_i) = 0$.The covariance matrix of the sample in F is C (Equation 6.21) and corresponding eigenvalue problem is given as $\lambda V^{\kappa} = C^{\kappa}V^{\kappa}$.

$$C = \frac{1}{m} \sum_{i=1}^{m} \kappa(x_i)\kappa(x_i)' \qquad (6.21)$$

The corresponding eigenvector V_i is then given as follows (Equation 6.22)

$$V_i = \sum_{i=1}^{m} \alpha_i \kappa(x_i) \qquad (6.22)$$

where α_i is the eigenvalues. The kernel function is taken as $K_{i,j} = <\kappa(x_i) . \kappa(x_j)>$ for all i, j = 1,2...m . Now, Equation (6.21) can be transformed to the eigenvalue problem as Equation (6.23).

$$\lambda\alpha_i = K\alpha_i, \text{ where } i = 1, 2, \ldots, m \quad (6.23)$$

Finally, based on the estimated α_I, the principal components for x_i are calculated using Equation (6.24).

$$S(i) = u_i^T \kappa(x_i) = \sum_{i=1}^{m} \alpha_i(j)\kappa_{i.j} \quad (6.24)$$

In the kernel method, the inner product of feature vectors is replaced to a nonlinear kernel function, through which a nonlinear mapping of the feature vector to a high dimensional space is performed in an implicit manner.

6.8. EXPERIMENTAL RESULTS OF FEATURE FUSION AND SELECTION ALGORITHMS

The experiments to evaluate the algorithms of Phase III of CAP-LR are as follows.

1. Identify the feature category that has maximum impact
2. Effect of the two proposed leaf features, vein feature and ripple feature on leaf classification
3. Identify the efficient algorithm among the two proposed feature selection algorithm

To identify the feature category that has maximum impact, the individual features selected by the feature selection method was analyzed. The same experiments were also used to compare the performance of the proposed feature selection algorithms with the conventional counterparts. The conventional algorithms are the techniques which are used to build the proposed feature selection algorithms.

Out of the 14 features used, two features, namely, enhanced vein feature and ripple feature are the two new features proposed to be used along with other chosen features for leaf recognition in CAP-LR. To analyze the effect of these two features, two feature sets, one having all 14 features and another having only 12 features without vein and ripple features were constructed. These two feature sets were then used to classify the leaf images using three conventional classifiers, namely, Back Propagation Neural Network, Support Vector Machine and Wavelet Neural Networks. The performance was analyzed using recognition rate (%) and the results are presented in Figure 6.11.

From the results, it can be seen that the two proposed leaf specialized features have positive impact on recognition rate. While using 14 leaf features dataset, the BPNN showed 6.78% efficiency gain, while it was 4.19% with SVM and 5.63% with WNN. Thus, it is proved that the two leaf specialized features provide positive contribution.

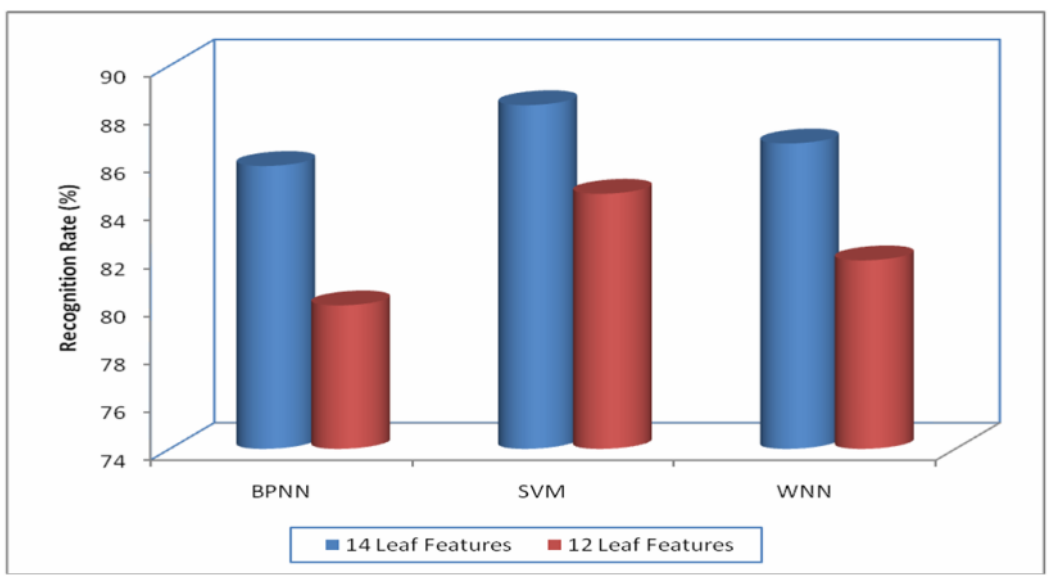


Figure 6.11 : Effect of Leaf Features on Classification

Motivated by these results, and to further improve the efficiency of the recognition, the 14 leaf feature set was fused with the other feature categories, namely, geometric, color, texture and fractals. The fusion would thus result with four feature sets, namely, Geometric and Leaf Feature set (GLFS), Color and Leaf Feature set (CLFS), Texture and Leaf Feature set (TLFS) and Fractals and Leaf Feature set (FLFS) as shown in the following Figure (6.12).

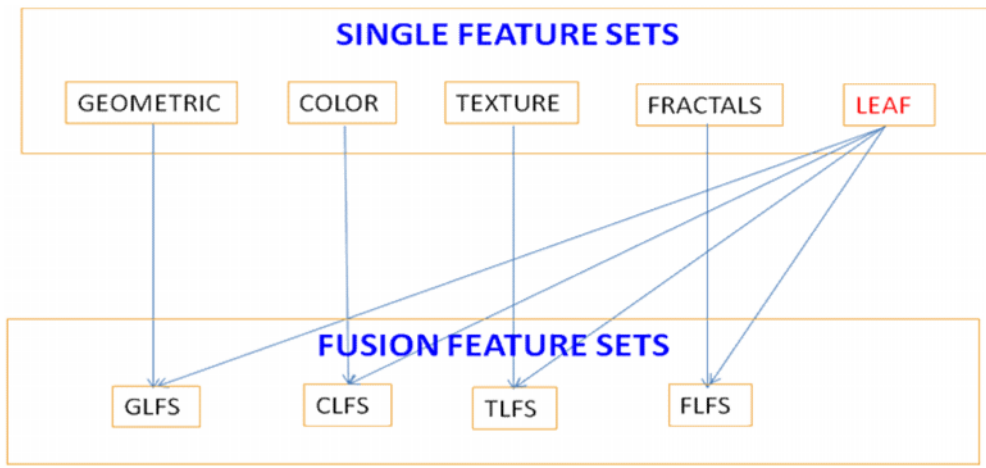


Figure 6.12 : Single Feature and Fusion Feature Sets

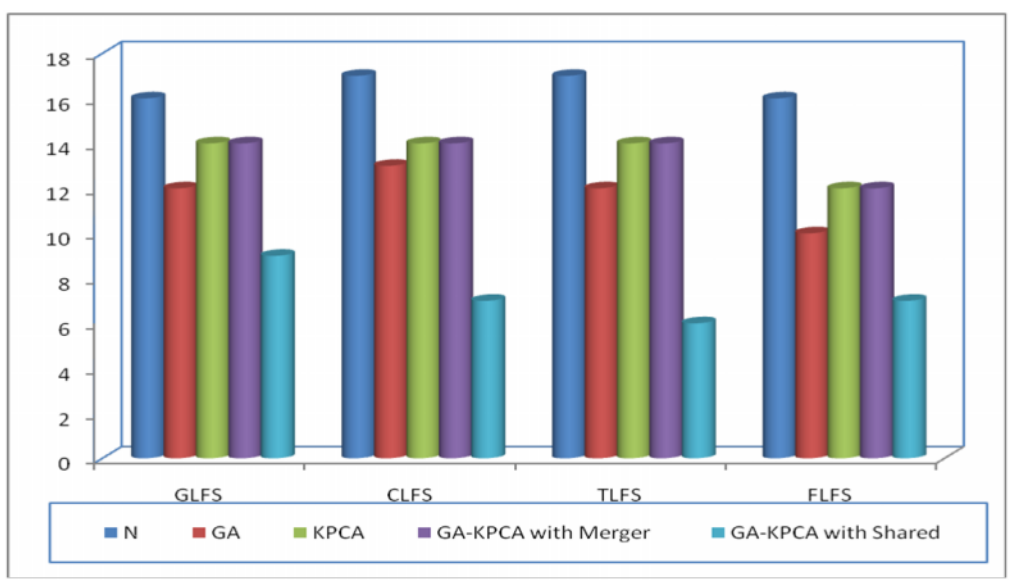


Figure 6.13 : Effect of Feature Fusion on Number of Features selected

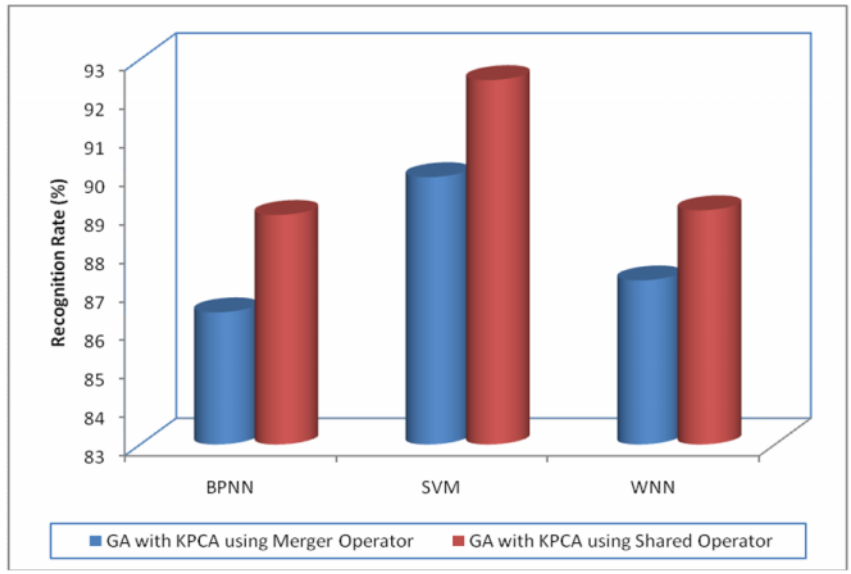


Figure 6.14: Effect of Proposed Feature Fusion and Selection Algorithms

The effect of feature fusion and selection on the number of features selected is presented in Figure 6.13. The results from this figure again show that the proposed algorithms have satisfied the goals of feature selection techniques.

Further experiments were also conducted to analyze the performance of the two proposed feature fusion and selection algorithm, namely, GA and KPCA with shared operator and GA and KPCA with merger operator with respect to recognition rate. Figure 6.14 presents this result. From the figure, it can be seen that GA intersection KPCA algorithm performs better than GA union KPCA. Hence GA and KPCA combined with intersection operator prove more efficient than all the three single level classifiers. The BPNN showed a 2.84% efficiency gain while using GA $\cap$ KPCA algorithm, SVM showed 2.73% while WNN increased the efficiency by 2.04% when compared with GA $\cup$ KPCA.

6.9. CHAPTER SUMMARY

This chapter presented details regarding the procedures used by CAP-LR for feature extraction and selection. The study extracts five categories of features, namely, geometric, color, texture, fractals and leaf features. During experimentation, it was found that the leaf features have positive impact on recognition process and hence, fusion was performed by combining leaf features with other four types. Two types of fusion techniques are considered. The first is the simple and straightforward concatenation method and the second is based on Genetic Algorithm. One problem encountered while using concatenation method is the curse of dimensionality, which was solved using Kernel Principal Component Analysis. For feature selection, the results from GA and KPCA are combined using two Boolean operators, namely, $\cap$ (shared features) and $\cup$ (merged features). As GA $\cap$ KPCA algorithm produced better results, it was decided to use this technique in the subsequent recognition steps of CAP-LR. These features are then used to classify the leaf images. The method used for classification are described in the next chapter, **Design of Leaf Image Classification Models for Plant Identification.**