CHAPTER 6

BASIC_INTRINSIC_PATTERN_EXTRACTION_AND_INDEXING

The intensity variation pattern in an image is said to be texture. The texture of an image cannot be analyzed from a single pixel point; it requires the neighboring point’s intensity values. To analyze the texture, a mathematical model has to be derived with respect to its intensity values. Generally as per (Jain 1998) Texture handbook, texture mathematical modeling method is categorized as Statistical, Geometrical, Signal Processing, and Modal-based methods. In order to analyze the spatial distribution of pixel intensity values, Statistical based texture analysis models were used in this work.

In statistical texture analysis as per Manik & Andrew (2004), the intensity property of the image is represented by the non-deterministic properties of the distribution of grey levels of an image. The types of statistical approach are categorized with respect to the mathematical model such as first order model, second order model, and higher order model. The statistic calculated from the grey level histogram of the image is said to be first order texture calculation. Some of the texture components under this category are Mean, Standard deviation, skewness, and kurtosis. Instead of identifying these features from the raw grey scale image, image experts preferred to use transformed image such as Gradient image, Laplace image, and so on. The texture metrics based on second order statistics are autocorrelation function, spatial frequencies, edge frequencies, and co-
occurrence matrices. The metrics based on higher order statistics are Run length codes, Laws texture energy measures, Fractal texture descriptors, and Wavelet based texture calculation.

6.1 INTRINSIC PATTERN ANALYSIS

The main objective of BIP (Basic Intrinsic Pattern) algorithm is to determine the smoothness or roughness pattern of the flower image. Flower can be classified as per the flower head. There are three types of flower head: ray florets, disk florets, and combination of both as shown in Figure 4.5. The texture pattern of ray florets would be smooth and the patterns of disk florets are rough. The intensity variation pattern of the ray and disk florets are shown in Figure 6.1

![Figure 6.1 Ray and Disk florets intensity pixel variance](image)

Using statistical analysis based feature vector calculation the characteristic of the image texture pattern such as smoothness, coarse, grainy etc. can be calculated. Such calculation required second order statistical methods. Spatial gray level co-occurrence matrix, which was framed by Haralick (1979), estimates such image properties. This co-occurrence matrix is related to second-order statistics, which considers the relationship among
neighboring pixels. The author suggested the use of gray level co-occurrence matrices (GLCM) which have become one of the most well known and widely used texture features.

This method is based on the joint probability distributions of pairs of pixels. GLCM shows how often each gray level occurs at a pixel located at a fixed geometric position relative to each other pixel, as a function of the gray level. GLCM is a square matrix of dimension equal to the number of intensity levels in the image, for each distance d and orientation θ. Naturally the distance considered between two pixels would be one and angle of $0^\circ, 45^\circ, 90^\circ$ and $135^\circ$. The surface plot and the GLCM of the ray and disk flowers are shown in Figures 6.2(a) and (b) respectively. The pixel variance in disk flower is clearly visible through the surface plot; in ray, the variance is low.

The key objective of this work is to provide semantic meaning to the flower images by integrating both the visual feature vector and characteristic of the flower image. This $8 \times 8$ GLCM matrix cannot be included as such into the ontology attributes. As specified by Materka & Strzelecki (1998) some of the texture discrimination factors are calculated as shown in Figure 6.3.

From these values, it is difficult to conclude whether the flower has smooth or rough texture pattern. Thus, the issue to devise an intrinsic pattern algorithm is that the value has to be compressed and the classification needs to be genuine.
Figure 6.2 (a) GLCM of Disk Flower (b) GLCM of Ray Flower
For image compression and recognition, Principal Component Analysis (PCA) technique was used. Especially for face recognition system, PCA plays a major role. PCA is a kind of statistical method, which is used to reduce the large dimensionality of the data space to the smaller intrinsic dimensionality of feature vector space.

6.2 PRINCIPAL COMPONENT ANALYSIS (PCA)

Principal component analysis is an uncomplicated statistical procedure to extract relevant principal information from complicated data sets. PCA uses orthogonal transformation to convert a set of correlated variables into a set of linearly uncorrelated values called principal component. Using the basic of PCA algorithm, the basic patterns of the given data sets are identified. The basic steps involved to identify the principal component are shown in block diagram -Figure 6.4

**Figure 6.3 GLCM matrix analysis**
Figure 6.4 Basic Steps of PCA Algorithm

The variance is the measure of the deviation from the mean value in single dimension. As per Equation (6.1), the variance is the spread of the data in the given dataset with the mean $\bar{X}$

$$\sigma^2 = \frac{\sum_{i=1}^{n}(X_i - \bar{X})}{(n-1)} \quad (6.1)$$

Covariance is the variance with respect to two values. If $(X,Y)$ are the two neighboring values, the $\text{cov}(X,Y)$ is calculated as shown in Equation (6.2)

$$\text{cov}(X,Y) = \frac{\sum_{i=1}^{n}(X_i - \bar{X})(Y_i - \bar{Y})}{(n-1)} \quad (6.2)$$

The zero covariance value means that the two-dimensional data are independent of each other. The negative value indicates that when one value increases the other value decreases and positive value covariance indicates
that both the value increase and decrease together. The Covariance matrix for 3-D value \((X,Y,Z)\) is represented as illustrated in Equation (6.3)

$$\text{cov}(X,Y,Z) = \begin{bmatrix}
\text{cov}(X,X) & \text{cov}(X,Y) & \text{cov}(X,Z) \\
\text{cov}(Y,X) & \text{cov}(Y,Y) & \text{cov}(Y,Z) \\
\text{cov}(Z,X) & \text{cov}(Z,Y) & \text{cov}(Z,Z)
\end{bmatrix}$$ (6.3)

### 6.3 BASIC INTRINSIC PATTERN EXTRACTION

To represent the texture pattern of the flower image an algorithm called Basic Intrinsic Pattern (BIP), was derived from the basic of PCA. The core objective of this approach is to reduce the computation and the size of the identified feature vector, which would represent the intrinsic pattern of the given image. As per Smith (2002) tutorial, PCA is a statistical model, which is used to identify some of the discrete patterns for a given dataset. From the identified pattern design, PCA implements a linear system, which is derived from applied linear algebra. This linear system is used to identify the potential feature vectors from the pattern design and formalize a tactic to analyze the continuity in data sets. Steps involved in BIP analysis are shown in the block diagram- Figure 6.5

![Figure 6.5 BIP Procedure](image-url)
6.3.1 Image Data Preparation

The Asteroideae flower families consist of more than 16,000 species. Among them, only 100 species are considered. To determine the Basic Intrinsic Pattern on each of the 100 flower images, first, nearly five images per each of the 100 flower species are collected. Let us consider Aaronsohnia as one amongst those flower species and let $AI$ be the Aaronsohnia flower image set which consists of five different sets of images of size 100x100. Initially these images are all quantized using EMK quantization algorithm, from which the prevalent color and their corresponding color blobs are extracted and saved as 100 x100 JPEG images as shown in Figure 6.6. This can be expressed as shown in Equation (6.4).

Figure 6.6 Aaronsohnia Flower Image data preparation

$$AI = \{AI_1 \cup AI_2 \cup AI_3 \cup AI_4 \cup AI_5\} \quad (6.4)$$

6.3.2 GLCM Cumulative Matrix Creation

Normally in Statistical Methods, various statistical measures are derived from the image intensity values. One of the most effective statistical models used for various statistical calculations is Grey level Co-occurrences Matrix, which is a matrix constructed upon the likelihood of a pair of grey pixel values in an image at random location and orientation.

The Grey Level Co-occurrence Matrix is said to be the second order texture static calculation. In the first order, the texture statistics are
calculated with respect to the current image pixel without considering the
neighbor pixel values. In second order calculation, the neighboring pixel’s
intensity plays a vital role in statistic calculation.

The GLCM is a matrix $X$, $(I \times J)$ whose number of rows and
columns depends upon the grey level of given input images. As in this work,
in the case all of the JPEG images of size 100x100, the $X$ matrix size is
always about $(8 \times 8)$, as here we first convert the JPEG color image to grey
level image. Thus, in general, for an image of size $M \times N$ $(100 \times 100)$ and gray
level of $G$ (8 level), the GLCM matrix for the input image $AI_1$ is calculated
by:

$$X(i,j | \Delta a,\Delta b) = S_p \Theta (i,j | \Delta a,\Delta b) \quad (6.5)$$

Where,

$\Delta a,\Delta b$ are called offset, which specifies the pixels intensity with
respect to its neighboring pixel’s intensity for a given set of angular
$(0^\circ, 45^\circ, 90^\circ, 135^\circ)$ displacement.

$S_p = \text{Scalar product},$

$\Theta = \text{Occurrences of intensity with respect to the given offset}.$

$$S_p = \frac{1}{(M-\Delta a)(N-\Delta b)} \quad (6.6)$$

$$\Theta(i, j | \Delta a, \Delta b) = \sum_{m=1}^{N-\Delta a} \sum_{n=1}^{M-\Delta b} Q \quad (6.7)$$

$$Q = \begin{cases} 1 & \text{if \ } AI_i(m, n) = i \text{ and } AI_i(m + \Delta a, n + \Delta b) = j \\ 0 & \text{Elsewhere} \end{cases} \quad (6.8)$$
Thus, through Equations (6.5) to (6.8), the GLCM matrix of size (8x8) for each image in AI set is calculated separately and combined in a single matrix \( X(AI) \) as expressed in Equation (6.9). The resultant matrix of \( X(AI) \) will be of size 64x8.

\[
X(AI) = X(AI_1 \cup AI_2 \cup AI_3 \cup AI_4 \cup AI_5)
\]  

(6.9)

### 6.3.3 Eigen Value Generation

Eigen vector of a matrix is said to be a non-zero vector from the reference (Olver 2003), such that by multiplying the matrix by this vector will provide a matrix, which will be parallel to the original vector. The Eigen value of each Eigen vector is factor values by which the vectors are scaled when multiplied by the matrix. An \((M\times N)\) matrix will have \(M\) Eigen values.

To determine the pattern effectively, the resultant matrix has to be normalized with respect to a mean image. So, from the set of five images one such image \( AI_3 \) is considered as a mean image and the resultant matrix \( X(AI) \) is subtracted from the \( X(AI_3) \) matrix value as shown in Equation (6.10).

\[
X(AI)_m = X(AI_3) - X(AI_1 \cup AI_2 \cup AI_3 \cup AI_4 \cup AI_5)
\]  

(6.10)

Now, for this matrix \( X(AI)_m \), which is of size \((64x8)\), Eigen values are computed by implementing the traditional algorithm in matrices. The Eigen values can be computed only for square matrices. To identify the Eigen values for the non-square matrix, there is a need to form a square gram matrix \( K \). The \( K \) matrix for \( X(AI_m) \) can be computed as written in Equation (6.11)

\[
K_{ Ai_m } = X(AI_m) . X(AI_m)^T
\]  

(6.11)
Now, the matrix size of $X(AI_m) = (64 \times 8)$

The matrix size of $X(AI_m)^T = (8 \times 64)$

Thus the resultant matrix size would be $K_{Alm} = (64 \times 64)$

From this square matrix $(64 \times 64)$ the Eigen values are computed

6.3.4 Potential Eigen Value Selection

For a Square matrix of $(n \times n)$, there would be ‘$n$’ number of Eigen values. So, here totally 64 Eigen vector values were computed. From these non-zero values, we can select the largest among them. This can be determined as potential Eigen vector values. Thus using these steps, almost 5 Eigen vector values are identified for all the 100 different flower species and indexed separately.

The abstract algorithm of Basic Intrinsic Pattern indexing is shown in Algorithm 6.1.

---

**Algorithm 6.1: Basic Intrinsic Pattern Indexing**

**Input:** Prevalent color indexed Images

**Output:** Positive BIP feature vector for each flower class

1) Begin

2) Prevalent color extracted images are converted to 8 level grey scale image of size $(100 \times 100)$

3) For each flower species collect 5 instance of images say

   $AI = (AI_1 \cup AI_2 \cup AI_3 \cup AI_4 \cup AI_5)$

4) For each $AI$ determine the grey level co-occurrences matrix of size $(8 \times 8)$
for eight grey levels. The resultant matrix would be:

5) \( X(AI) = X(AI_1 \cup AI_2 \cup AI_3 \cup AI_4 \cup AI_5) \)

6) Let \( X(AI_3) \) be the mean image, then compute \( X(AI)_m \) the resultant matrix would be of size \((64 \times 8)\)

\[
X(AI)_m = X(AI_3) - X(AI_1 \cup AI_2 \cup AI_3 \cup AI_4 \cup AI_5)
\]

7) Normalize the \( X(AI)_m \) to square matrix using Square Gram matrix concept, now \( K_{AIm} \) would be of size \((64 \times 64)\)

\[
K_{AIm} = X(AI_m) \cdot X(AI_m)^T
\]

8) Compute 64 Eigen values from \( K_{AIm} \) matrix

9) Select first five positive Eigen values as BIP feature vector value for that flower species

10) End

6.4 RESULTS AND IMPLEMENTATION

These algorithms are implemented on Asteroideae flower family image dataset to categorize the flower image as ray flower or disk flower. For the experimental purpose we have taken nearly 100 species. For each species five instances of flower images are collected, thus with a total of 500 images this algorithm was tested.

The first Eigen vector of ray and disk flowers is shown in Table 6.1. The size of ray (smooth) flower is 8 bit whereas for disk (rough) flower it is 7 bit. The identified Asteroideae flowers with respect to their intrinsic pattern are listed as shown in Figure 6.7.
Table 6.1 BIP of certain Ray and Disk flower

<table>
<thead>
<tr>
<th>Ray Flower</th>
<th>BIP</th>
<th>Disk Flower</th>
<th>BIP</th>
</tr>
</thead>
<tbody>
<tr>
<td>Adenoglossa</td>
<td>69214778</td>
<td>Aaronsohnia</td>
<td>5538111</td>
</tr>
<tr>
<td>Chrysanthemum</td>
<td>67625076</td>
<td>Cotula barbata</td>
<td>8854932</td>
</tr>
<tr>
<td>glandiforum</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Cladanthus</td>
<td>36441960</td>
<td>cotula turbinata</td>
<td>2524231</td>
</tr>
<tr>
<td>arabicus</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Coleostephus</td>
<td>40718140</td>
<td>Foveolina</td>
<td>2704441</td>
</tr>
<tr>
<td>myconis</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Coleostephus</td>
<td>48291890</td>
<td>Oncosiphon</td>
<td>4499811</td>
</tr>
<tr>
<td>multicoulis</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Cota tincoria</td>
<td>77946708</td>
<td>Lonas</td>
<td>3370212</td>
</tr>
<tr>
<td>Glossopappus</td>
<td>35672406</td>
<td>Filifolium sibiricum</td>
<td>4619571</td>
</tr>
<tr>
<td>Heteranthemis</td>
<td>53136974</td>
<td>Gonospermum</td>
<td>6668841</td>
</tr>
<tr>
<td>Lepidophorum</td>
<td>35300960</td>
<td>Handelia</td>
<td>5518843</td>
</tr>
<tr>
<td>Prolongoa</td>
<td>85023644</td>
<td>Otanthus</td>
<td>6600899</td>
</tr>
<tr>
<td>Ursinia</td>
<td>13253078</td>
<td>Pentzia</td>
<td>5729341</td>
</tr>
</tbody>
</table>

One of the prime objectives to devise this BIP algorithm is to reduce the size of the texture feature vector. The size of the BIP feature vector is compared with the SIFT, SURF, GLCM, MPEG 7’s edge histogram descriptor and Gabour histogram as shown in Figure 6.8. The performance analysis of the BIP is done by retrieving alike texture pattern for the input query image. The Recall comparison is shown in Figure 6.9.
Figure 6.7 Basic Intrinsic Pattern based flower image indexing
Figure 6.8 Feature size quantization analysis

Figure 6.9 Retrieval rate with respect to Texture
6.5 SUMMARY

In this chapter, the texture based feature extraction named as Basic Intrinsic Pattern extraction algorithm was elaborated. The intrinsic patterns of the flower image with respect to smooth and rough texture pattern were all analyzed. The GLCM based feature evaluation has not provide any texture pattern information that was specified in this chapter to state the justification for the need of a new algorithm. As these features are used to build an image ontology, the size quantization is also one of the necessities of the devised BIP algorithm. The PCA based BIP algorithm formulation and the results were discussed in this chapter.