CHAPTER 5

FEATURE EXTRACTION TECHNIQUES

Proper representation of digital image can be achieved by reducing the number of data points of the image signal in such a way that they maintain most of the information. The data points of image signal are known as features. These features can be achieved by using different feature extraction methods.

Retinal image databases are useful to design the neural network based classifier system. These databases are the collection of two classes of retinal images; normal and diabetic retinopathy affected retinal images (abnormal). It is expected that the classifier system must be capable to classify these images in specific class. The different features extracted from the images are used as input feature vectors to train the neural network classifier.

5.1 Feature Extraction

The extracted features of retinal images classified into two groups with their subclasses as shown in figure 5.1.

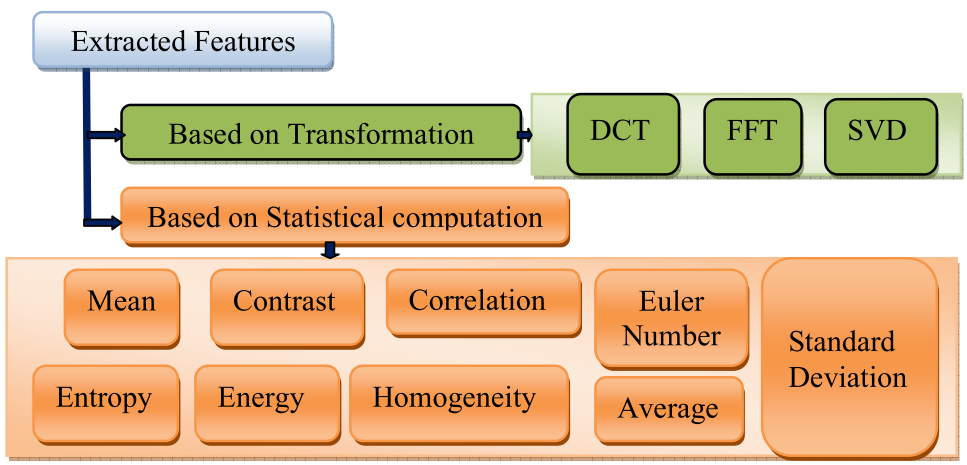

Fig. 5.1: Extracted features of retinal images

The transformation based features such as Discrete Cosine Transform (DCT), Fast Fourier Transform (FFT) and Singular Value Decomposition (SVD) and statistical based features, Entropy, Energy, Standard Deviation, Mean, Contrast, Homogeneity, Average, Correlation, and Euler number are extracted from retinal images. Each transformation based feature vector along with different statistical features formed a feature vector which is useful for training of the neural network.

5.2 Transformation based Feature Extraction

5.2.1 Discrete Cosine Transform (DCT)

Data compression is one of the important issues in today's computer applications world. Two classes of compression algorithms try to reduce the number of bits required to represent a signal. These classes are lossless and lossy. Compression algorithms work by removing redundancy in the signal. Fourier transformation is considered as basic compression algorithm in digital image processing. The Fourier transform plays an important role in a various digital image processing manipulations such as enhancement, analysis, restoration, and compression.

The discrete cosine transform (DCT) is closely related to the discrete Fourier transform. It is a separable linear transformation; that is, the two-dimensional transform is equivalent to a one-dimensional DCT performed along a single dimension followed by a one-dimensional DCT in the other dimension.

2-D DCT of each block is computed and the transform coefficients are quantized. Quantized coefficients are coded losslessly. The choice of quantization affects the transmission rate and distortion.

Advantages of DCT relative to the Discrete Fourier transform (DFT)

- It is real-valued and coefficients are nearly uncorrelated
- Energy compaction is achieved by maximum signal energy representation by a few coefficients.

It would be not possible to transmit a picture or video or even a decent quality audio signal without some kind of compression. One of the most significant methods of compression algorithms is the Discrete Cosine Transform (DCT).

For a given a sequence x[n], for n =0 to N-1,

The DCT and inverse DCT are defined as

$$X[k] = DCT\{x[n]\} = \sqrt{\frac{2}{N}} C[k] \sum_{n=0}^{N-1} x[n] \cos\left(\frac{\pi k(2n+1)}{N}\right), \quad \mathrm{k = 0,..., N - 1}$$

Eq. 5.1

$$x[n] = IDCT\{x[k]\} = \sqrt{\frac{2}{N}}\sum_{k=0}^{N-1} C[k]X[k]\cos\left(\frac{\pi k(2n+1)}{N}\right), \quad \mathrm{n} = 0,\ldots,\mathrm{N}-1$$

Eq. 5.2

DCT consider an image as a sum of sinusoids of varying magnitudes and frequencies. The *dct*2 function in the Image Processing Toolbox of MatLab calculates the two-dimensional discrete cosine transform of an image.

5.2.2 Fast Fourier Transform(FFT)

In signal processing, any signal can be represented by a collection of sine waves of differing frequencies, magnitudes and phases. The transformation of a signal into its constituent sinusoids is known as the Fourier Transform [177].

In digital image processing, for discrete data, the computational basis of spectral analysis is the discrete Fourier transform (DFT). The DFT transforms time based data into frequency based data. The DFT of a vector x of length n is another vector y of length n:

$$Y_{\mathrm{p+1}} = \sum_{j=0}^{n-1} w^{jp} x_{\mathrm{j+1}}$$

Eq. 5.3

Where ω is the angular frequency, i is the imaginary unit, and p, j for indices that run from 0 to n–1. The indices p +1 and j+1 run from 1 to n.

Data in the vector x are assumed to be separated by a constant interval in time or space is given by $dt = 1/f_s$ where f_s is the sampling frequency. The DFT y is complex-valued. The absolute value of y at index p+1 measures the amount of the frequency which is $f = p\ (f_s / n)$ present in the data as shown in figure 5.2.

The first element of output (y), corresponding to zero frequency, is the sum of the data in x. This DC component is often removed from y so that it does not difficult to understand the positive frequency content of the data. Current signal and image processing applications would be impractical without DFT computing.

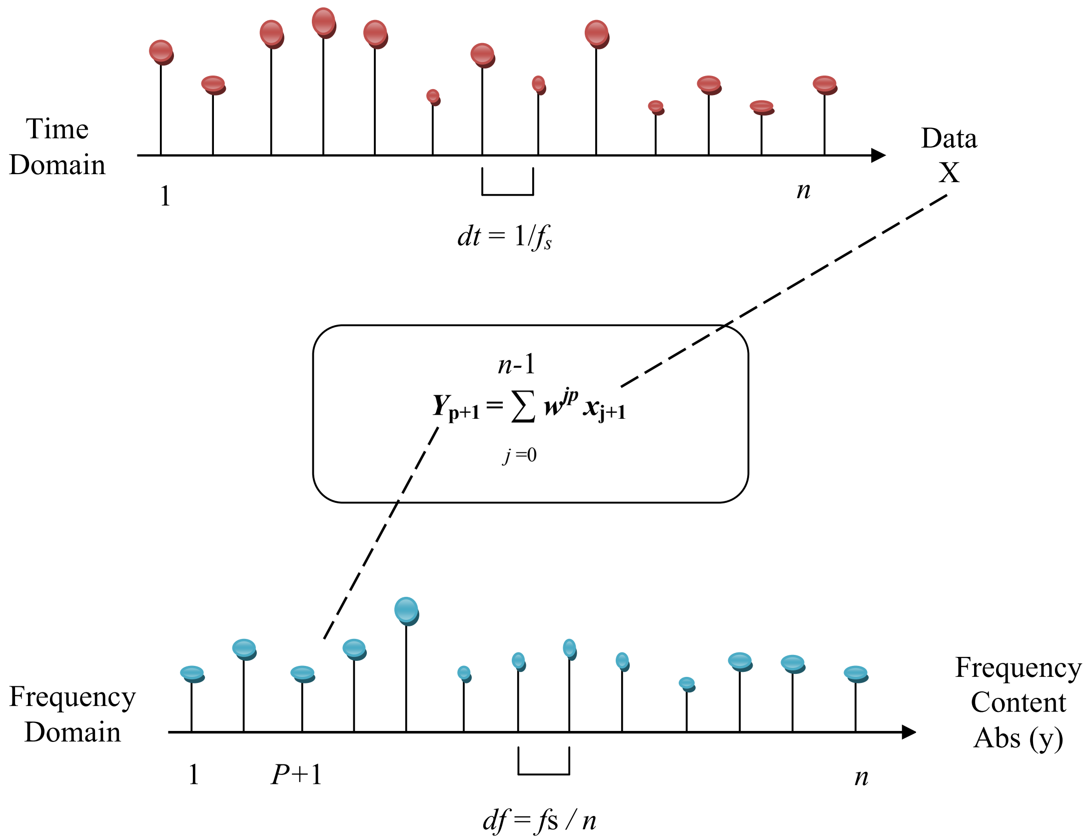

Fig. 5.2: Time and frequency domain representation of the signal

DFT requires n multiplications and n additions i.e. a total of $2n^2$ floating-point operations for data vector of length n. Therefore to compute a million-point DFT, computation time will be too much more. On the other hand Fast Fourier Transform (FFT) algorithms have computational complexity $O(n \log n)$ instead of $O(n^2)$.

There is a fast algorithm for computing the Discrete Fourier which is known as Fast Fourier Transform (FFT). The Fourier transform represented an image as a sum of complex exponentials of varying magnitudes, frequencies, and phases.

If f (m, n) is a function of two discrete spatial variables m and n, then the two-dimensional Fourier transform of f (m, n) is defined by the relationship as

$$F(\omega 1, \omega 2) = \sum_{m=-\infty}^{\infty} \sum_{n=-\infty}^{\infty} f(m,n) e^{-j\omega_1 m} e^{-j\omega_2 n} \qquad \text{Eq. 5.4}$$

Where - f (m, n) is an original image

- F (w1, w2) *is* the frequency-domain representation of f (m, n) i.e. transformed image.

The Fourier transform is a representation of an image as a sum of complex exponentials of varying magnitudes, frequencies, and phases.

5.2.3 Singular Value Decomposition (SVD)

Singular Value Decomposition is a factor matrix as shown in below figure 5.3.

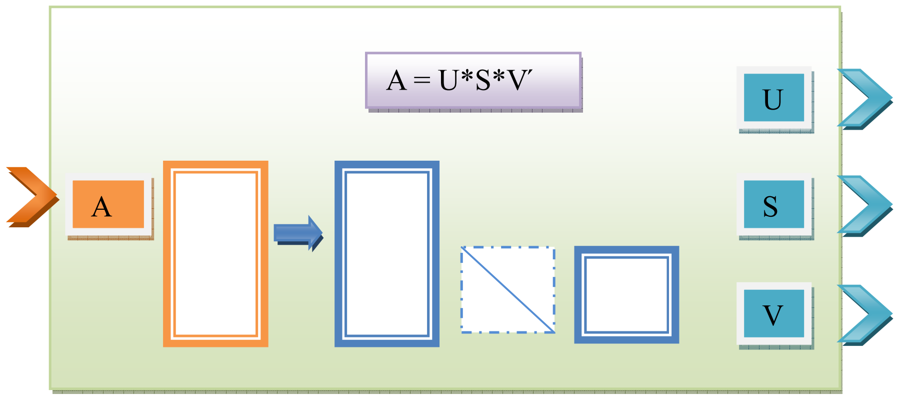

Fig. 5.3: Block Diagram of Singular Value Decomposition

The Singular Value Decomposition block factors the M-by-N input matrix A such that

$$A = U.\textit{diag}(S).V^* \qquad \text{Eq. 5.5}$$

Where, U is an M-by-P matrix, V is an N-by-P matrix, S is a length-P vector and P is defined as min (M, N)

When

- M = N, U and V are both M-by-M unitary matrices
- M > N, V is an N-by-N unitary matrix, and U is an M-by-N matrix whose columns are the first N columns of a unitary matrix
- N > M, U is an M-by-M unitary matrix, and V is an N-by-M matrix whose columns are the first M columns of a unitary matrix

In all cases, S is a 1-D vector of positive singular values having length P. Length-N row inputs are treated as length-N columns. The first (maximum) element of output S is equal to the 2-norm of the matrix A. The output is always sample based.

5.3 Statistical Feature Extraction

5.3.1 Entropy

Texture analysis considers to the description of regions of the images by their texture contents. Texture analysis attempts to compute sensitive features described by the terms like rough, silky, etc in the context of an image. In this case, the roughness refers to the variations in the brightness values or gray levels. Some of the most commonly used texture measures are derived from the Gray Level Co-occurrence Matrix (GLCM). The GLCM tabulates frequently arises different combinations of pixel brightness values (gray levels) occur in a pixel pair in an image.

The texture analysis plays an important role into several new functions such as range, standard deviation, and entropy. Entropy is a statistical measure of randomness that can be used to characterize the texture of the input image.

Low entropy images have a very little contrast and large number of pixels with the same or analogous values. An image that is perfectly flat will have entropy of zero. Therefore, they can be compressed to a relatively small size. On the other hand, high entropy images have high contrast from one pixel to the next and as a result cannot be compressed as much as low entropy images.

The entropy of an image is calculated using the equation

$$\text{Entropy} = \sum P_i \, Log_2 \, (P_i) \qquad \text{Eq 5.6}$$

where P_i is the probability for the difference between two neighboring pixels is equal to i, and Log_2 is the base 2 algorithm.

Shannon's entropy plays a key role in information theory called as measure of uncertainty. The entropy of a random variable is defined in terms of its probability distribution and can be depicted to be a good measure of randomness or uncertainty.

5.3.2 Mean

Mean is used for getting the pixel values or statistics of image signal. It gives the average or mean value of array or signal. In MatLab software, mean2 is used for getting average or mean of matrix elements of image signals.

For example, B = *mean*2 (A) computes the mean of the values in A where the input image A can be numeric or logical. The output image B is a scalar of class double.

5.3.3 Standard deviation

There are two common definitions for the standard deviation ***s*** of a data vector ***x***.

$$s = \left(\frac{1}{n-1} \sum_{n-1}^{n} \left(x_i - \bar{x} \right)^2 \right)^{\frac{1}{2}} \quad \text{Eq 5.7}$$

$$s = \left(\frac{1}{n} \sum_{n-1}^{n} \left(x_i - \bar{x} \right)^2 \right)^{\frac{1}{2}} \quad \text{Eq 5.8}$$

Where, $\bar{x} = \frac{1}{n} \sum_{i=1}^{n} x_i$

The *n* is the number of elements in the sample. The two forms of the equation differ only in *n*-1 versus *n* in the divisor.

The 2-D Standard Deviation block calculates the standard deviation of each M-by-N input matrix or of a sequence of inputs. This block's functionality is different from the signal processing block. It set Standard Deviation block, which computes the standard deviation of each column in the input. For solely real or solely imaginary inputs, the standard deviation is the square root of the variance and is given by the following equation:

$$y = \sigma = \sqrt{\frac{\sum_{i=1}^{M} \sum_{j=1}^{N} (u_{ij} - \mu)^2}{MN - 1}} \quad \text{Eq 5.9}$$

Where, μ is the mean of the input matrix u. For complex inputs, the block outputs the total standard deviation of the input matrix, which is the square root of the total variance.

$$\sigma = \sqrt{\sigma_{\mathrm{Re}}{}^2 + \sigma_{\mathrm{Im}}{}^2} \quad \text{Eq 5.10}$$

The total standard deviation is not equal to the sum of the real and imaginary standard deviations.

5.3.4 Contrast

It is a measure of the intensity difference between a pixel and its neighbor over an image.

$$\sum_{i,j} |i - j|^2 \, p(i, j) \quad \text{Eq. 5.11}$$

It is easy to understand contrast visually. In high contrast image, definite edges and the unlike elements of the images are easily visible. In a low contrast image, all the colors are almost the same and it is difficult to distinguish the details in the images.

5.3.5 Correlation

The operation called correlation is closely related to convolution. In correlation, the value of an output pixel is also computed as a weighted sum of neighboring pixels. The difference is that the matrix of weights, in this case called the correlation kernel, is not rotated during the computation.

It is a measure of how a pixel is associated to its neighbor over the whole image.

$$\sum_{i,j} \frac{(i-\mu_i)(j-\mu_j)p(i,j)}{\sigma_i \sigma_j} \qquad \text{Eq. 5.12}$$

Correlation coefficient a measure of the degree of linear relationship between two variables, frequently labeled X and Y. Where as in regression the importance is on forecasting one variable from the other, in correlation the emphasis is on the degree to which a linear model may exhibits the relationship between two variables. In regression the interest is directional, one variable is predicted and the other is the predictor; in correlation the interest is non-directional, the relationship is the critical aspect.

5.3.6 Energy

It returns the sum of squared elements in the GLCM (Gray level to co-variance matrix). The range of energy is in between 0 and 1.Energy is 1 for a constant image. Energy of an image is represented by

$$\sum_{i,j} p(i,j)^2 \qquad \text{Eq. 5.13}$$

5.3.7 Homogeneity

It returns a value that measures the closeness of the distribution of elements in the GLCM to the GLCM diagonal. Homogeneity is 1 for a diagonal GLCM.

$$\sum_{i,j} \frac{p(i,j)}{1+|i-j|} \qquad \text{Eq 5.14}$$

5.3.8 Euler Number

The Euler number is a measure of the topology of an image. It is defined as the total number of objects in the image minus the number of holes in those objects. You can use either 4- or 8-connected neighborhoods. The *bweuler* function returns the Euler number for a binary image in MatLab [178].

For example, computes the Euler number for the circuit image, using 8-connected neighborhoods.

```
BW1 = imread('circbw.tif');
eul = bweuler(BW1,8)

eul =
    -85
```

Here, the Euler number is negative, indicating that the number of holes is greater than the number of objects.