CHAPTER 3
EXPERIMENTAL DATA COLLECTION

3.1 INTRODUCTION
The information image is embedded using one of the directions via.
i) start from left top of cover image, move to the right, go around and reach the starting point, ii) start from right top of cover image, move to the left, go around and reach the starting point, iii) start from bottom right, move to the top, go around and reach the starting point, iv) start from left bottom, go vertical, go around and reach the starting point, v) lay horizontally odd-even vi) lay vertically odd-even, vii) lay horizontally even-odd, viii) lay vertically even-odd, ix) lay 45° odd-even, x) lay -45° even-odd, xi) lay 45° even-odd, and xii) lay in circular form as shown in Figure 3.1.

Fig. 3.1 Direction of random distribution of information image in cover image
Sample cover images (Group 1) are shown in Figure 3.2. The total number of cover images considered is 1024. The cover and information images have been taken from the Matlab 7® image library. Information images (Group 2) are shown in Figure 3.3. Steganographic images for training (Group 3) are shown in Figure 3.4(a) and Figure 3.4(b). Steganographic images for testing (Group 4) are shown in Figure 3.5(a) and Figure 3.5(b). Group 3 and Group 4 have been obtained by embedding Group 2 images into Group 1 images. From these embedded images 512 has been chosen for training and 256 for testing. The general procedures followed for embedding information are mentioned below:

1. LSB, DCT, DWT, and DFT encoding schemes are considered for the experimental work.
2. 256-bit color images (one matrix) and true color images (3 matrices / planes i.e. red, green, blue) - are considered for hiding information.
3. Patterns are generated by considering 2 X 2 pixels from cover images, information image and steganographic image. Redundant 2 X 2 pixels are eliminated. Labeling of 2 X 2 pixels is done as 0.1 for cover image, 0.2 for information image and 0.3 for steganographic image.
4. The labeling of image types and their target value is shown in Table 3.1.

3.2 DATA CONVERSION
3.2.1 Normalization of the patterns

The patterns are normalized so that the values of the features from the cover images are in the range of 0 to 1, and the computational complexity is reduced. The normalization of the patterns is done by

\[ x_i = x_i / \text{xmax} \]  \hspace{1cm} (3.1)
Where $x_i$ is the value of a feature, and $x_{max}$ is the maximum value of the feature.

Fig. 3.2 Cover images taken from Matlab library (Group 1)

Fig. 3.3 Information images (Group 2)
<table>
<thead>
<tr>
<th>COVER IMAGES</th>
<th>INFORMATION IMAGES</th>
<th>STEGANOGRAPHIC IMAGES</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image1.png" alt="Image" /></td>
<td><img src="image2.png" alt="Image" /></td>
<td><img src="image3.png" alt="Image" /></td>
</tr>
<tr>
<td><img src="image4.png" alt="Image" /></td>
<td><img src="image5.png" alt="Image" /></td>
<td><img src="image6.png" alt="Image" /></td>
</tr>
<tr>
<td><img src="image7.png" alt="Image" /></td>
<td><img src="image8.png" alt="Image" /></td>
<td><img src="image9.png" alt="Image" /></td>
</tr>
<tr>
<td><img src="image10.png" alt="Image" /></td>
<td><img src="image11.png" alt="Image" /></td>
<td><img src="image12.png" alt="Image" /></td>
</tr>
<tr>
<td><img src="image13.png" alt="Image" /></td>
<td><img src="image14.png" alt="Image" /></td>
<td><img src="image15.png" alt="Image" /></td>
</tr>
</tbody>
</table>

Fig. 3.4 a) Steganographic images for training (Group 3)
<table>
<thead>
<tr>
<th>COVER IMAGES</th>
<th>INFORMATION IMAGES</th>
<th>STEGANOGRAPHIC IMAGES</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image1" alt="Cover Image" /></td>
<td><img src="image2" alt="Information Image" /></td>
<td><img src="image3" alt="Steganographic Image" /></td>
</tr>
<tr>
<td><img src="image4" alt="Cover Image" /></td>
<td><img src="image5" alt="Information Image" /></td>
<td><img src="image6" alt="Steganographic Image" /></td>
</tr>
<tr>
<td><img src="image7" alt="Cover Image" /></td>
<td><img src="image8" alt="Information Image" /></td>
<td><img src="image9" alt="Steganographic Image" /></td>
</tr>
<tr>
<td><img src="image10" alt="Cover Image" /></td>
<td><img src="image11" alt="Information Image" /></td>
<td><img src="image12" alt="Steganographic Image" /></td>
</tr>
<tr>
<td><img src="image13" alt="Cover Image" /></td>
<td><img src="image14" alt="Information Image" /></td>
<td><img src="image15" alt="Steganographic Image" /></td>
</tr>
<tr>
<td><img src="image16" alt="Cover Image" /></td>
<td><img src="image17" alt="Information Image" /></td>
<td><img src="image18" alt="Steganographic Image" /></td>
</tr>
</tbody>
</table>

**Fig. 3.4 b) Steganographic images for training (Group 3)**
<table>
<thead>
<tr>
<th>COVER IMAGES</th>
<th>INFORMATION IMAGES</th>
<th>STEGANOGRAPHIC IMAGES</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image1.png" alt="Image 1" /></td>
<td><img src="image2.png" alt="Image 2" /></td>
<td><img src="image3.png" alt="Image 3" /></td>
</tr>
<tr>
<td><img src="image4.png" alt="Image 4" /></td>
<td><img src="image5.png" alt="Image 5" /></td>
<td><img src="image6.png" alt="Image 6" /></td>
</tr>
<tr>
<td><img src="image7.png" alt="Image 7" /></td>
<td><img src="image8.png" alt="Image 8" /></td>
<td><img src="image9.png" alt="Image 9" /></td>
</tr>
<tr>
<td><img src="image10.png" alt="Image 10" /></td>
<td><img src="image11.png" alt="Image 11" /></td>
<td><img src="image12.png" alt="Image 12" /></td>
</tr>
</tbody>
</table>

**Fig. 3.5 a) Steganographic images for testing (Group 4)**
<table>
<thead>
<tr>
<th>COVER IMAGES</th>
<th>INFORMATION IMAGES</th>
<th>STEGANOGRAPHIC IMAGES</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image1" alt="Cover Image 1" />.png</td>
<td><img src="image2" alt="Information Image 1" />.png</td>
<td><img src="image3" alt="Steganographic Image 1" />.png</td>
</tr>
<tr>
<td><img src="image4" alt="Cover Image 2" />.png</td>
<td><img src="image5" alt="Information Image 2" />.png</td>
<td><img src="image6" alt="Steganographic Image 2" />.png</td>
</tr>
<tr>
<td><img src="image7" alt="Cover Image 3" />.png</td>
<td><img src="image8" alt="Information Image 3" />.png</td>
<td><img src="image9" alt="Steganographic Image 3" />.png</td>
</tr>
<tr>
<td><img src="image10" alt="Cover Image 4" />.png</td>
<td><img src="image11" alt="Information Image 4" />.png</td>
<td><img src="image12" alt="Steganographic Image 4" />.png</td>
</tr>
</tbody>
</table>

Fig. 3.5 b) Steganographic images for testing (Group 4)
3.2.2 Selection of patterns for training

The numbers of classes, which are based on the classification range of the outputs, are decided. If only one output is considered, the range of classification is simple. If more than one output is considered, a combination criterion has to be considered. The total number of patterns is decided for each class. Out of these patterns, the number of patterns to be used for training the network is decided. The remaining patterns are used for testing the classification performance of the network. The patterns selected for training the network should be such that they represent the entire population of the data from the input images. The selection of patterns is done by

\[
E_i^2 = \frac{\sum_{j=1}^{n_i} (X_{ij} - \text{mean}(X_i))^2}{\sigma_i^2}
\]

(3.2)

Where,

\( E_i^2 \) is the maximum variance of a pattern,
nf is the number of features, and the value $\sigma_i^2$ is obtained by

$$\sigma_i^2 = \frac{\sum_{j=1}^{nf} (X_{ij} - \text{mean}(X)_j)^2}{L}$$

(3.3)

Where, L is the number of patterns.

The value of $E_i^2$ is found for each pattern. Patterns with maximum $E_i^2$ are chosen from each class for training the network. For each pattern target outputs are defined, so that during training, supervised algorithms can be used and during testing, the required output can be obtained by giving inputs. The total variance found by using equation (3.3) is categorized to know whether the patterns are similar to eliminate redundant patterns. Redundancy is identified if patterns total variance is less than particular range when any one of the pattern alone is considered.

Fig. 3.6 a) Cover image b) Message image c) Stego image

3.3 BIT PLANE ANALYSIS

A binary image is a digital image that has only two possible values for each pixel. Binary images are, also called bi-level or two-level. The names black-and-white, monochrome or monochromatic are often used, but may also designate any images that have only one sample per pixel, such as gray scale images. Binary images often arise in digital image
processing as masks or as the result of certain operations such as segmentation, thresholding and dithering. Every single pixel in an 8-bits gray level digital image consists of 8 bits. Gray-level images are also encoded as a 2D array of pixels, using eight bits per pixel, where a pixel value of 0 usually means “black” and a pixel value of 255 means “white”, with intermediate values corresponding to varying shades of gray. An 8-bit monochrome image can also be thought of as a collection of bit-planes, where each plane contains a 1-bit representation of the image at different levels of detail. An image can be represented with maximum intensity value of 255. The value 255 can be represented by binary values “11111111”. Each bit is treated as a plane as in (Table 3.2).

<table>
<thead>
<tr>
<th>Plane</th>
<th>Bit value</th>
<th>Upper nibble (Foreground in an image)</th>
<th>Lower nibble (Background in an image)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1&lt;sup&gt;st&lt;/sup&gt;</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>2&lt;sup&gt;nd&lt;/sup&gt;</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>3&lt;sup&gt;rd&lt;/sup&gt;</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>4&lt;sup&gt;th&lt;/sup&gt;</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>5&lt;sup&gt;th&lt;/sup&gt;</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>6&lt;sup&gt;th&lt;/sup&gt;</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>7&lt;sup&gt;th&lt;/sup&gt;</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>8&lt;sup&gt;th&lt;/sup&gt;</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

In actual practice, the person involved in retrieving the message will not know the pattern of hiding information that has been used during steganography. Keeping this in view, a method has been proposed to find out the presence of information in the cover image. This is identified by bit plane analysis. The cover and message images are shown in Figure 3.6(a) and Figure 3.6(b) respectively.

Random least significant bit (LSB) embedding method (Table 3.3) is used to produce the stego image that is shown in Figure 3.6(c). Bit
planes of cover and stego image which are shown in figure 3.6(a) and Figure 3.6(b) are analyzed. This approach is straight forward and general, so that one can clearly see hidden information with varying densities.

<table>
<thead>
<tr>
<th>Table 3.3 Least significant bit embedding</th>
</tr>
</thead>
<tbody>
<tr>
<td>Carrier image</td>
</tr>
<tr>
<td>Category</td>
</tr>
<tr>
<td>Original binary value</td>
</tr>
<tr>
<td>Equivalent decimal value</td>
</tr>
</tbody>
</table>

**Information embedding**

<table>
<thead>
<tr>
<th>Method 1</th>
<th>Information embedded in lower 4 bits</th>
</tr>
</thead>
<tbody>
<tr>
<td>Method 2</td>
<td>Information can be embedded in any of the one or two or three lower bits</td>
</tr>
</tbody>
</table>

| Combined binary value | 1 1 1 1 1 0 0 1 |
| Category | UN<sub>c</sub> | UN<sub>i</sub> |
| Decimal value | 249 |

### 3.4 FEATURE EXTRACTION

Transforming the input data into the set of features is called feature extraction. It is a special form of dimensionality reduction. During embedding process, features of typical natural images will be violated. Hence feature based universal steganalysis includes 2 phases. First is to obtain the features that are able to capture statistical changes present in the image after the embedding process. Next the classification
algorithm should be chosen that accepts the selected features as input and informs whether the given image is stego or non stego. A good feature should be consistent, accurate, and monotonic in capturing statistical changes left by the embedding process. Various researchers proposed different methods for feature extraction. Some of them are given in the review of literature in section 2.8.

When the input data to an algorithm is too large to be processed and when redundant in formations are available (e.g. the same measurement in both feet and meters) then the input data will be transformed into a reduced representation that produces set of features. These are also called as feature vector. Feature set should extract the relevant information from the input data in order to perform the desired task using this reduced representation instead of the full size input. Figure 3.7 represents the selection of feature vector for the sample cover images (Group 1) given in Figure 3.2. This feature vector is given as the input to the neural network classifier that informs whether the hidden information is present or not.

Fig. 3.7 Feature extraction
In this research, for feature extraction 2 pixels are taken and converted into gray scale values. To reduce the computational complexity these values are normalized. (i.e.) the value of the features will be in a range of 0 to 1. Hence two normalized gray scale values are used in the input layer of BPA and RBF as features. No hidden layer is used in FUBPA. The number of nodes in the input layer of FUBPA is 8 which is the binary representation of each intensity value of a pixel in the (cover) image. The number of nodes in the output layer is 1 that determines if image is stego or non stego (cover). The output of the FUBPA is ‘0’ or ‘1’ where ‘0’ represents no hidden message and ‘1’ represents a hidden message.

Neural network training is done with 512 images. Training set consists of 2 classes (cover images and stego images). This training is very much useful to learn the nature of the image as well as to enhance the detection accuracy. The algorithm is tested with 256 images. It detects whether secret information is present or not by comparing background pixels of stego image and cover image. If the pixel value of the cover image differs with stego image then the stego image embeds some secret data. The detection accuracy is 94.5%.

<table>
<thead>
<tr>
<th>Table 3.4 3*3 Pixels</th>
</tr>
</thead>
<tbody>
<tr>
<td>x-1,y-1</td>
</tr>
<tr>
<td>x,y-1</td>
</tr>
<tr>
<td>x+1,y-1</td>
</tr>
</tbody>
</table>

Sample 3*3 pixel values are given in Table 3.4. These pixels can be given as input to the neural network during training process. Figure
3.8 shows how the pixels are given as input during the training process for BPA neural network.

Fig. 3.8 Training for BPA

3.5 RESULTS AND DISCUSSION

Least significant bits are the deserving places in an image to hide the message data, because, their alterations results unnoticeable loss of quality, gives no clue to human eye. The other important nature in LSBs is that, they are completely random in terms of their overall significance to the complete image. The perception of LSBs as a binary image in isolation will appear scattered such that the values make no difference and look very random. Images from the natural scenes mostly include objects which contain color changes because of the varying intensities of visible light subjected on the objects. Shadows can act as patterns for such type of images. The pixel values of the image decrease by very small amounts as the shadow gets stronger. It can be assumed that the LSBs of an image are more structured posing a huge weakness that can be exploited through visual attacks.

If original cover is available then a forensic expert obtaining the LSB planes of both the original and the suspect image can conclude the cleanness of the suspect image by computing the difference between the two obtained LSB planes one from the other. They will then be left with an image similar to that in where the black regions of the image
represent the values that do not change between both images, and the white regions represent secret message detection. The 8 bit planes of both the original and stego image are shown in Figure 3.9 and Figure 3.10 respectively. The difference of corresponding 8 bit planes of original and suspect image is computed and shown in Figure 3.11. White regions are observed within the bit plane images in the Figure 3.11. It can also be observed that the lower bit plane holds much secret information compared to its next higher plane. It is clear that the suspect file is embedded with secret information.

The steganalyst can therefore conclude that the suspect image has been secretly embedded, and they can even identify the locations of the modified pixels. But in practical, the original cover will not be available during steganalysis in most of the cases. To produce a more guaranteed analysis, it is needed to combine several steganalysis methods to fix a suspect image which is discussed in subsequent chapters.

Fig. 3.9 a) The first plane of carrier image
Fig. 3.9 b) The second plane of carrier image

Fig. 3.9 c) The third plane of carrier image
Fig. 3.9 d) The fourth plane of carrier image

Fig. 3.9 e) The fifth plane of carrier image
Fig. 3.9 f) The sixth plane of carrier image

Fig. 3.9 g) The seventh plane of carrier image
Fig. 3.9 h) The eighth plane of carrier image

Fig. 3.10 a) The first plane of stego image
Fig. 3.10 b) The second plane of stego image

Fig. 3.10 c) The third plane of stego image
Fig. 3.10 d) The fourth plane of stego image

Fig. 3.10 e) The fifth plane of stego image
Fig. 3.10 f) The sixth plane of stego image

Fig. 3.10 g) The seventh plane of stego image
The eighth plane of stego image

Embedded message is detected in plane 1

Fig. 3.10 h) The eighth plane of stego image

Fig. 3.11 a) Difference of carrier and stego- Plane 1
Embedded message is detected in plane 2

Fig. 3.11 b) Difference of carrier and stego- Plane 2

Embedded message is detected in plane 3

Fig. 3.11 c) Difference of carrier and stego- Plane 3
Embedded message is detected in plane 4

Fig. 3.11 d) Difference of carrier and stego- Plane 4

Embedded message is detected in plane 5

Fig. 3.11 e) Difference of carrier and stego- Plane 5
Embedded message is detected in plane 6

Fig. 3.11 f) Difference of carrier and stego- Plane 6

Embedded message is detected in plane 7

Fig. 3.11 g) Difference of carrier and stego- Plane 7
3.6 SUMMARY

This chapter has presented LSB embedding of information image into a cover image. The normalization of patterns and selection of patterns for training neural networks is presented. A general method of analyzing the presence of information in an image using bit plane method is given. The amount and density of information present in each plane is also presented. Chapter 4 presents the implementation of BPA for steganalysis.