CHAPTER V
IMAGE ENCRYPTION USING FUZZY SETS THEORY

5.1. Introduction

The soft computing algorithms are the fusion of paradigms designed to present solutions to the real world problems, which are not modelled or very difficult to be modelled mathematically. They are consortium of methodologies that work synergistically and provide in one form or another, flexible information and processing capability for handling real-life ambiguous situations. [148, 149] Their aim is to exploit the tolerance for imprecision, uncertainty, logic and partial truth in order to achieve close resemblance with human-like decision making. Soft computing is basically optimization technique to find solutions for problems which are very hard to solve.

Fuzzy sets theory is one of the most important branches of soft computing algorithms. Fuzzy sets were introduced by Lotfi A. Zadeh in 1965 to represent data and information possessing non-statistical uncertainties. It has been developed through various ways and in many disciplines. Fuzzy logic is an approach related to the fuzzy sets theory of soft computing algorithms based on “degree of truth” rather than the usual “true of false” (1 or 0) and Boolean logic on which the modern computer is based. This chapter presents two new algorithms for image encryption which use the fuzzy set theory to provide high protection to the images data from illegal intrusions. It is fast in the process of encryption and decryption. The decryption process does not induce any loss of image data, and it has ability of dealing with different format of images as well. The first algorithm based on fuzzy logic to provide security levels and their corresponding processing levels by generating random sequences for the encryption/decryption process. The second algorithm based on fuzzy graph to obtain highly encrypted image.
5.2. Basics Concepts of Fuzzy Sets Theory

The fuzzy sets theory can be defined as the theory which is related to the classes of objects with un-sharp boundaries in which membership is a matter of degree. Since 1992, the fuzzy set theory is known as the theory of neural nets and the field of evolutionary programming have become one of the important areas of ‘computational intelligence’ or ‘soft computing’. The strong relationship between these areas has naturally become close. In this section, however, the focus primarily will be on the notion of fuzzy set theory, fuzzy logic and fuzzy graph. The notion of a fuzzy set stems from the observation made by Zadeh (1965a) that “more often than not, the classes of objects encountered in the real physical world don’t have precisely defined criteria of membership”. [150-152] Fuzzy logic is brilliant engineering and data analysis approach proposed by Lotfi A. Zadeh. This section deals with the basic concept of fuzzy logic, and it includes some basic examples of fuzzy logic. Fuzzy logic is an approach based on ‘degrees of truth’ on which the modern computer is based. In addition, the fuzzy logic has two different meanings; one is wide and the other is narrow.

In the wide sense, fuzzy logic (FL) is almost equivalent with the theory of fuzzy sets. On the other hand, the narrow sense shows fuzzy logic as a logical system, which is an extension of multi-valued logic. In this perspective, fuzzy logic in its narrow sense is a branch of FL. Even in its more narrow definition, fuzzy logic differs both in concept and nature from traditional multi-valued logical systems. [153]

Fuzzy graph theory was introduced by Azriel Rosenfeld in 1975. Though he was very young, he has been growing fast and has numerous applications in various fields [154]. Later, Bhattacharya [155] gave some remarks on fuzzy graphs. Mordeson and Peng introduced some operations on fuzzy graphs. A new concept namely, graph structure was introduced by E. Sampathkumar [156].

*Definition:* A fuzzy graph is a pair $G(\sigma, \mu)$ where $\sigma$ a fuzzy subset of $S$ is, $\mu$ is a symmetric fuzzy relation on $\sigma$. The elements of $S$ are called the vertices or nodes of $G$ and the pair of vertices as edges in $G$. The underlying crisp graph of the fuzzy graph $G(\sigma, \mu)$ is denoted as $G^*: (S, E)$ where $E \subseteq S \times S$. The crisp graph $(S, E)$ is a special case of the fuzzy graph $G$ with each vertex and edge of $(S, E)$ having degree of membership.
5.3. Mechanism of Fuzzy Logic

The use of fuzzy logic needs at least two input magnitudes. If they are more, they would be better for fuzzy logic. Three basic steps are needed for applying fuzzy analysis:

- Fuzzification of the inputs
- Fuzzy-Inference
- Defuzzification of obtaining crisp output

For fuzzification, a fuzzy set is required for each input. This set is a kind of linguistic variables such as relative terms: far, close, small, normal, high, cold, warm, hot, etc. Once fuzzy set is defined for each of the variables, the degree of membership should be calculated for every input. Minimum degree of membership is zero, and maximum is 1 or 100%. To calculate degree of membership, it is easy to use linear functions as presented in figure 5.1 bellow. [157]

![Figure 5.1 Calculation degree of membership using linear functions](image)

However, calculation of the degree of membership is not limited to linear functions only. Gaussian or some arbitrary functions can be used as it is shown in figure 5.2 bellow. In that sense, the number of possibilities is theoretically infinite.
After calculating the degrees of membership for every input and every fuzzy set, the step of (fuzzy-inference) should be started. All numerical data obtained in fuzzification process need to be mutually compared in ‘if-then’ fashion. This process is called fuzzy-inference. Overall number of ‘if-then’ rules is equal to the outputs of fuzzy set numbers for each input. For example, if we have two input values; input X has three fuzzy sets (negative, optimal and positive), and input Y has five fuzzy sets (negative, negative-optimal, optimal, optimal-positive and positive), the (15) ‘if-then’ rules are required to compare negative degree of membership of X with negative degree of membership of Y, negative degree of membership of X with negative-optimal degree of membership of Y and so forth. In fact, there is a need for comparing at least two numbers in every ‘if-then’ rule, but there are a lot of numbers and ‘if-then’ rules to be compared. There are various options to show how to determine output from compared numbers in each ‘if-then’ rule. Options are as follows:

- Choose minimum: \( \mu = \min\{X, Y, Z\} \)
- Choose maximum: \( \mu = \max\{X, Y, Z\} \)
- Choose complement \( \mu_c = 1 - \mu \)

Defuzzification is used for obtaining crisp output from numbers obtained in step fuzzy-inference. In order to calculate crisp output, we can use “centre of gravity” or some other method. Together with numbers from fuzzy-inference, we need to have additional parameters called “singleton values”. Singleton values can be achieved from experiments.
5.4. Application of Fuzzy Logic

Fuzzy logic addresses such applications perfectly as it resembles human decision making with an ability to generate precise solutions from certain or approximate information. [158-160] It fills an important gap in engineering design methods left vacant by purely mathematical approaches (e.g. linear control design), and purely logic-based approaches (e.g. expert systems) in system design. Fuzzy logic can be applied in various fields.

5.4.1. Fuzzy Logic in Image Processing

Fuzzy logic has rapidly become one of the most successful technologies for developing sophisticated image processing because of its simple use. The applying of fuzzy modelling to image processing is not new, a fact that Pal and King. [161] were quick to realize. It has been used to enhance digital image and signals processing. The Canon camera’s autofocus system is the best example of the efficient implementation of fuzzy technology. The autofocus, auto zooms and auto exposure systems for Minolta cameras use fuzzy controllers. Using fuzzy logic techniques, Sanyo and Canon camcorders are better in auto-white balancing system, auto exposure and autofocus system. Image stabilizer for camcorder of Matsushita has been implemented with fuzzy technology. Fuzzy inference was also used to improve image quality. Because of Fuzzy system technology, electro photography process of photocopying machine has been improved. The image quality of Sanyo copies has been improved by better toner supply control based on fuzzy control. Matsushita copies are also improved by better auto exposure fuzzy based control. Some other successful applications are hand written language recognition and voice recognition.

5.4.2. Fuzzy Logic in Control of Complex System

The aim of the application is to establish proper protection as well as to minimize the power used to protect the pipeline. Fuzzy control used as modelling the different soil conditions along the pipeline would be complex for a practical control system. The fuzzy control model was made with 3 anodes followed by 126 rules, which resulted in the controller having adequate performance in maintaining protection. [162]
5.4.3. Fuzzy Logic in Medical Diagnosis
Fuzzy methods are used in the automation of the medical diagnosis application. Using these methods, the medical result yield contains likelihood estimation rather than confirmation of presence or absence of disease. In practice, the recommendations for action suggested in guidelines are vague and conflicting. But these difficulties are well handled by the fuzzy methods [163, 164].

5.4.4. Fuzzy Logic in Automotive Industry
Most of the automotive manufacturers are perusing fuzzy control concepts. Fuzzy control has been applied to control automatic transmission system, suspension system, engine system, climate system and antilock brake system. These systems are used to make the vehicle better, more efficient and safer to ride. [165]

5.4.5. Fuzzy Logic in Healthcare Industry
Fuzzy system technology has larger impact on healthcare industry. The biomedical applications are less due to inherent complexity and uncertainty of the systems as well as the risk involved. In biomedicine science, human knowledge, skills and experience are more important in diagnosis and treatment of diseases. This biomedical system is difficult to be managed because it has time to delay. World’s first fuzzy control in medicine is drug delivery system, which was developed and implemented to regulate blood pressure in post-surgical open heart patients at cardinal intensive care unit (ICU). Some other applications include determining the disease risk using fuzzy expert systems, medical control system and determination of drug dose. Classification of tissue and structure in electrocardiograms, classification of normal and cancerous tissues in brain magnetic resonance image are the other successful applications of fuzzy systems in medicine [166, 167].

5.4.6. Fuzzy Logic in Data Mining
Fuzzy logic is applied at various stages of Data Mining to impart human type reasoning in decision making. The human behaviour is always fuzzy in nature and it is very difficult to model human behaviour patterns with crisp Boolean Data Mining systems.
5.5. Properties of Fuzzy Logic

Some of the essential characteristics of fuzzy logic are related to the following properties:

• In fuzzy logic, exact reasoning is viewed as a limiting case of approximate reasoning.
• In fuzzy logic, everything is a matter of degree.
• In fuzzy logic, knowledge is interpreted as a collection of elastic or, equivalently, fuzzy constraint on a collection of variables.
• Inference is viewed as a process of propagation of elastic constraints.
• Any logical system can be fuzzified.

There are two main characteristics of fuzzy systems that give them better performance for specific applications.

• Fuzzy systems are suitable for uncertain or approximate reasoning, especially for the system with a mathematical model that is difficult to derive.
• Fuzzy logic allows decision making with estimated values under incomplete or uncertain information.

Elements of a fuzzy set may belong to the set, may not belong to the set, or may belong to a degree.

Membership of a crisp set is described by a bivalent condition; the membership of a fuzzy set is described by a multivalent condition.

Fuzzy sets are used to deal with uncertainty on one hand and to represent knowledge on the other. [168]
5.6. Proposed Algorithms Based on Fuzzy Sets Theory

5.6.1. Fuzzy Logic Algorithm

In this approach, a new algorithm is proposed to encrypt an image using the concept of fuzzy logic based on fuzzy pseudorandom number PN bit generator. This algorithm provides more secure image encryption, minimum loss of data, maximum speed and maximum distortion in encryption, with the addition of extra degree of protection.

The Steps of Encrypting an Image Using Fuzzy Logic Algorithm

1. Loading an image.
2. Determining the size of image (height and width).
   2.1. Checking the result of \((H \ mod \ 48)\) and \((W \ mod \ 48)\). If they are equal to zero then go to 3rd step otherwise do
   \[
   H = H + (48 - (H \ mod \ 48)) \quad \text{and} \quad W = W + [48 - (W \ mod \ 48)].
   \] (5.1)
3. Dividing the image into a set of blocks, each block size is (48).
4. Doing pixels permutation procedure.
   4.1. The permutation process for 48 successive pixels as:
   \(S_0(0), \ldots, S_0(11), S_1(0), \ldots, S_1(11), S_2(0), \ldots, S_2(11), S_3(0), \ldots, S_3(11)\)
   4.2. The permutation procedure is as follow:
   \[
   \text{Start:} \\
   \text{If } S_0(0) = S_0(1) \text{ then pixel (1) } \leftrightarrow \text{ pixel (2)} \\
   \text{If } S_0(1) = S_0(2) \text{ then pixel (2) } \leftrightarrow \text{ pixel (3)} \\
   \quad \vdots \\
   \text{If } S_3(9) = S_3(10) \text{ then pixel (46) } \leftrightarrow \text{ pixel (47)} \\
   \text{If } S_3(10) = S_3(11) \text{ then pixel (47) } \leftrightarrow \text{ pixel (48)} \\
   \text{End}
   \]
5. Encryption process for RGB

5.1. Encryption Process for the RED colour value:

The three sequences $S_0, S_1, S_2$ will be used in this process, where $S_0(0)$ will determine the direction of the R binary value rotation.

If $S_0(0) = 0$ R will be rotated left

If $S_0(0) = 1$ R will be rotated right

Number of rotated position of R will be determined according to:

$$Nr_{R}p = S_1(1) + 2 \cdot S_1(2) + 2^2 \cdot S_1(3)$$

(5.2)

The resulting rotated R will be bitwise XOREd with the decimal value of $S_2(4 \ldots 11)$

5.2. Encryption Process for the GREEN colour value:

The value of $S_2(0)$ will determine the direction of the G binary value rotation.

If $S_2(0) = 0$ G will be rotated left

If $S_2(0) = 1$ G will be rotated right

Number of rotated position of G will be determined according to:

$$Nr_{G}p = S_0(1) + 2 \cdot S_0(2) + 2^2 \cdot S_0(3)$$

(5.3)

The resulting rotated G will be bitwise XOREd with the decimal value of $S_1(4 \ldots 11)$

5.3. Encryption Process for the BLUE colour value:

The value of $S_1(0)$ will determine the direction of the B binary value rotation.

If $S_1(0) = 0$ B will be rotated left

If $S_1(0) = 1$ B will be rotated right

Number of rotated position of B will be determined according to:

$$Nr_{B}p = S_2(1) + 2 \cdot S_2(2) + 2^2 \cdot S_2(3)$$

(5.4)

The resulting rotated B will be bitwise XOREd with the decimal value of $S_0(4 \ldots 11)$

6. Repeating the 4th and 5th steps for all blocks

7. Getting encrypted block

8. Getting encrypted image
5.6.2. Fuzzy Graph

In this approach, a new algorithm is proposed to encrypt an image using the concept of fuzzy graphs obtained from a matrix of image’s pixels, and then they are used to encrypt an image. This algorithm provides more secure image encryption, minimum loss of data, maximum speed and maximum distortion in encryption with the addition of extra degree of protection.

The Steps of Encrypting an Image Using Fuzzy Graph Algorithm

1. Loading an image.

2. Determining the size of image (height and width).
   2.1. Checking the result of \((H \mod 5)\) and \((W \mod 5)\). If they are equal to zero then go to 3rd step otherwise do
   
   \[
   H = H + (5 - (H \mod 5)) \quad \text{and} \quad W = W + (5 - (W \mod 5)).
   \]  
   (5.5)

3. Dividing the image into a set of blocks, each block size is (8x8).

   Considering a block \(B(w \times h)\) where \(w\) and \(h\) are width and height of \(B\).

4. Extracting the RGB components and store them as \(R, G\) and \(B\) matrix
   
   \[
   R = R(r \star c), \quad G = G(r \star c), \quad B = R(r \star c).\]  
   (5.6)

4.1. Finding the membership for each pixel as: \(P(i,j)/255\).

5. Doing interval-valued fuzzy graphs for each block as:
   
   \[
   X = \left\{ \left( \frac{PR(P_1, P_2, P_3, \ldots, P_n)}{\mu(P_1, P_2, P_3, \ldots, P_n)} \right), \left( \frac{PG(P_1, P_2, P_3, \ldots, P_n)}{\mu(P_1, P_2, P_3, \ldots, P_n)} \right), \left( \frac{PB(P_1, P_2, P_3, \ldots, P_n)}{\mu(P_1, P_2, P_3, \ldots, P_n)} \right) \right\}
   \]
   (5.7)

   \[
   Y = \left\{ \left( \frac{PR(P_1, P_2, P_3, \ldots, P_n)}{\mu(R)} \right), \left( \frac{PG(P_1, P_2, P_3, \ldots, P_n)}{\mu(G)} \right), \left( \frac{PB(P_1, P_2, P_3, \ldots, P_n)}{\mu(B)} \right) \right\}
   \]

Where \(X\) and \(Y\) are an interval-valued fuzzy set, \(P_1, P_2, P_3, \ldots, P_n\) are the pixels in the block, \(PR, PG\) and \(PB\) are the pixels of RED, GREEN and BLUE components in each block. \(\mu(P_1, P_2, P_3, \ldots, P_n)\) the membership of pixels in each block and \(\mu(R), \mu(G), \mu(B)\) is the membership of RED, GREEN and BLUE components which
are calculated by dividing the summation of pixels for each component dividing by number of pixels in each block.

6. Obtaining the fuzzy graph for each component in each block.
   
   6.1. Considering the vertices of block as the entries of matrix and connecting the adjacent edge to make grid graph like graph.

   6.2. Applying the sigmoid function as:  
   
   \[
   \text{sig}(X; Y, C) = \frac{1}{1 + e^{-Y(X-C)}}
   \]

   Where \( X \) and \( Y \) are an interval-valued fuzzy set and \( C \) is the center of graph

7. Encrypting the block:
   
   \[
   E_{i,j} = \text{mod} \left( FG_{i,j} \times 10^3, 256 \right) \text{XOR} \ P_{i,j}.
   \]

8. Repeating 4th - 7th steps for all blocks to get encrypted image

5.7. Results and Discussion

In this section, several experiments are performed to test the effect of the proposed algorithms on image encryption. The proposed approaches are implemented and applied to three images, which have different types and sizes, to verify its adaptability, quality, security and speed. The results are illustrated in this section to confirm the properties of proposed approaches based on genetic algorithm and to compare them with the quality of image encryption. The security analysis performed in this approaches is visual analysis, histogram analysis, correlation coefficient. All these experiments prove that the proposed approaches in this chapter fulfil all the results completely and rightly. In this subsection, the discussions and analyses of the encryption algorithms based on fuzzy logic and fuzzy graph are presented. The following discussions and analyses are performed on original, encrypted and decrypted images. In these experiments, the algorithm parameters are set as follows: the images sizes are different; the length of images blocks is (48×48) pixels in fuzzy logic and (8×8) pixels in fuzzy graph. Image encryption and decryption tests have been carried out using standard images of different sizes in gray scale and colour.
Encrypted and decrypted outputs have been obtained from fuzzy logic and fuzzy graph algorithms. It may be noted that the encrypted image obtained from fuzzy logic and fuzzy graph algorithms does not resemble the original image.

![Image of Car: Original, Encrypted by Fuzzy Logic, Encrypted by Fuzzy Graph, Decrypted by Fuzzy Logic, Decrypted by Fuzzy Graph]

**Figure 5.3** Encryption and decryption of Image ‘Car’ by fuzzy logic and fuzzy graph

5.7.1. **Visual Analysis:**

The purpose of visual testing is to highlight the presence of the similarities between plain image and its cipher. Figure 5.3 shows that the encrypted images do not contain any features of the plain images. After the visual testing had been performed on some images, which have different sizes and formats, it showcased that there is no perceptual similarity.

5.7.2. **Statistical Analysis**

A statistical analysis of encrypted images provides much information about the security of a cipher with reference to statistical attacks that could be launched. This is shown by a test performed on the histograms of the enciphered images and on the correlations of the adjacent pixels in the ciphered image.

5.7.2.1. **Histogram Analysis**

The histogram analysis is used to illustrate the confusion and diffusion properties in the encrypted image for testing purposes. The histogram of the plain image ‘Lena’ and that of the encrypted image by using proposed approaches are shown in figures
5.4 – 5.7. While comparing the two, the histogram of the encrypted image is fairly uniform and is significantly different from that of the original image. The encrypted images transmitted are not affected by any attacker.

![Figure 5.4 Original Image of Lena and Its RGB Histogram](image1)

![Figure 5.5 Encrypted Image of Lena and Its RGB Histogram by fuzzy logic](image2)
Encryption and Decryption Image Using Multiobjective Soft Computing Algorithm

Figure 5.6 Original Image of Lena and Its RGB Histogram

Figure 5.7 Encrypted Image of Lena and Its RGB Histogram by fuzzy graph
5.7.2.2. Correlation Coefficient Analysis

The correlation coefficient displays the relationship between two neighbouring pixels. If correlation between two pixels is nearly 1, the image Pixels are highly correlated, but if it is nearly 0, the image pixels are highly uncorrelated. In the proposed method, the experiments on the images prove that the original image correlation is nearly one, and cipher image correlation is nearly zero. So, this saves the system from statistical attacks. Tables 5.1 and 5.2 show the results of experiments. The formula of correlation coefficient of two adjacent pixels is as follows:

\[
C_r = \frac{\sum_{j=1}^{N} x_j y_j - \sum_{j=1}^{N} x_j \sum_{j=1}^{N} y_j}{\sqrt{\sum_{j=1}^{N} x_j^2 - (\sum_{j=1}^{N} x_j)^2} \sqrt{\sum_{j=1}^{N} y_j^2 - (\sum_{j=1}^{N} y_j)^2}}
\]  

(5.10)

Where \(x\) and \(y\) are the values of two adjacent pixels in the image and \(N\) is the total number of adjacent pixels selected from the image.

The equation (3.5) is used to calculate the correlation coefficients in the horizontal, vertical and diagonal planes of the image. The calculated results are listed in Tables 5.1 and 5.2.

Table 5.1 Adjacent pixels correlation coefficients analysis by fuzzy logic

<table>
<thead>
<tr>
<th>Direction</th>
<th>Plain image of Lena</th>
<th>Cipher image of Lena</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Red</td>
<td>Green</td>
</tr>
<tr>
<td>Horizontal</td>
<td>0.9878</td>
<td>0.9860</td>
</tr>
<tr>
<td>Vertical</td>
<td>0.9868</td>
<td>0.9897</td>
</tr>
<tr>
<td>Diagonal</td>
<td>0.9871</td>
<td>0.9795</td>
</tr>
</tbody>
</table>

Table 5.2 Adjacent pixels correlation coefficients analysis by fuzzy graph

<table>
<thead>
<tr>
<th>Direction</th>
<th>Plain image of Lena</th>
<th>Cipher image of Lena</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Red</td>
<td>Green</td>
</tr>
<tr>
<td>Horizontal</td>
<td>0.9785</td>
<td>0.9754</td>
</tr>
<tr>
<td>Vertical</td>
<td>0.9811</td>
<td>0.9764</td>
</tr>
<tr>
<td>Diagonal</td>
<td>0.9763</td>
<td>0.9645</td>
</tr>
</tbody>
</table>
5.7.3. Sensibility Analysis

5.7.3.1. NPCR and UACI Analysis

The NPCR is measures the change rate of the number of the cipher-image pixels when only one pixel of the plain-image is modified. The (UACI) means unified average changing intensity. Index measures the average intensity of differences between two images. Let us assume two ciphered images $C_1$ and $C_2$ whose corresponding plain images have only one-pixel difference. The gray-level values of ciphered images $C_1$ and $C_2$ at row $i$, column $j$ are labeled as $C_1(i,j)$ and $C_2(i,j)$ respectively. NPCR and UACI are defined as:

$$\text{NPCR} = \frac{\sum_{i=1}^{W} \sum_{j=1}^{H} R(i,j)}{W \times H} \times 100\% \quad (5.11)$$

$$R(i,j) = \begin{cases} 1 & C_1(i,j) \neq C_2(i,j) \\ 0 & C_1(i,j) = C_2(i,j) \end{cases}$$

$$\text{UACI} = \frac{\sum_{i=1}^{W} \sum_{j=1}^{H} \frac{|C_1(i,j) - C_2(i,j)|}{255}}{W \times H} \times 100\% \quad (5.12)$$

Where $W$ and $H$ are the width and height of the image. In this example, the selected pixel is the last pixel of the plain-image. The NPCR and UACI at previous approaches are calculated and listed in Tables 5.3 and 5.4. The data show that the performance is satisfactory for the proposed approaches.

<table>
<thead>
<tr>
<th>Lena Image</th>
<th>NPCR</th>
<th>UACI</th>
</tr>
</thead>
<tbody>
<tr>
<td>NPCR</td>
<td>99.5793</td>
<td>0.1178</td>
</tr>
<tr>
<td>UACI</td>
<td>99.6247</td>
<td>0.0527</td>
</tr>
<tr>
<td>Blue</td>
<td>99.5990</td>
<td>0.0298</td>
</tr>
</tbody>
</table>

Table 5.3 NPCR and UACI by fuzzy logic

<table>
<thead>
<tr>
<th>Lena Image</th>
<th>NPCR</th>
<th>UACI</th>
</tr>
</thead>
<tbody>
<tr>
<td>NPCR</td>
<td>99.6049</td>
<td>0.01689</td>
</tr>
<tr>
<td>UACI</td>
<td>99.6168</td>
<td>0.0540</td>
</tr>
<tr>
<td>Blue</td>
<td>99.6049</td>
<td>0.0322</td>
</tr>
</tbody>
</table>

Table 5.4 NPCR and UACI by fuzzy graph
5.7.3.2. Peak Signal to Noise Ratio (PSNR) and Mean Square Error (MSE)

Peak signal to noise ratio (PSNR) and mean square error (MSE) are very useful measurements of performance the quality of reconstruction of encryption codes (e.g., for image encryption). The signal in this case is the original image, and the noise is. When comparing the plain image, cipher image and decrypted image. PSNR is an approximation to human perception of reconstruction quality. Although a higher PSNR generally indicates that the reconstruction is of higher quality PSNR is most easily defined via the mean square error (MSE) as:

$$PSNR = 10 \log_{10} \frac{255^2}{MSE}$$  \hspace{1cm} (5.13)

Where the mean square error (MSE) is defined as:

$$MSE = \frac{1}{MN} \sum_{i=1}^{M} \sum_{j=1}^{N} [I(i,j) - I'(i,j)]^2$$  \hspace{1cm} (5.14)

Where M and N are the number of pixels in the frame, I(i,j) and I'(i,j) are the number of pixels in the original and decrypted image, respectively. Greater PSNR value (>30dB) reveals better image quality. For encrypted image, smaller value of PSNR is expected. The results of PSNR and MSE for different images by using fuzzy logic and fuzzy graph algorithm are listed in Tables 5.5 and 5.6.

<table>
<thead>
<tr>
<th>Image</th>
<th>PSNR and MSE by fuzzy logic</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lena</td>
<td>8.4569</td>
</tr>
<tr>
<td>Bird</td>
<td>3.1830e+03</td>
</tr>
<tr>
<td>Car</td>
<td>9.0017</td>
</tr>
<tr>
<td>Lena</td>
<td>3.2765e+03</td>
</tr>
</tbody>
</table>

Table 5.5 PSNR and MSE by fuzzy logic

<table>
<thead>
<tr>
<th>Lena Image</th>
<th>PSNR and MSE by fuzzy graph</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lena</td>
<td>9.0017</td>
</tr>
<tr>
<td>Lena</td>
<td>3.2765e+03</td>
</tr>
</tbody>
</table>

Table 5.6 PSNR and MSE by fuzzy graph
5.7.4. Average Time of Encryption and Decryption

The executing time of encryption and decryption process by proposed approaches based on genetic algorithms has been implemented and conducted using Matlab-7.10.0 (R2010a) in AMD V 140 CPU @ 2.30 GHz with Windows-7 operating system. In this case, three separate images having the different type’s formats of JPG, TIF and BMP respectively and sizes 225 × 225 pixels, 194 × 259 pixels and 1153 × 2050 pixels have been used to measure the encryption time for each image by using the suggested approaches. The encryption time obtained using these images are given in tables 5.7. The average time of encryption is achieved by fuzzy logic and fuzzy graph while encrypting different images of different sizes and formats. This shows that the fuzzy graph algorithm is 1.07 times less than fuzzy logic algorithms. The result of average time is shown in Table 5.7 and Figure 5.8.

Table 5.7 Average time of encryption and decryption by using fuzzy logic and fuzzy graph algorithms with three images different in size, type and formats

<table>
<thead>
<tr>
<th>Image / Approach</th>
<th>Fuzzy Logic</th>
<th>Fuzzy Graph</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lena</td>
<td>16.477</td>
<td>14.936</td>
</tr>
<tr>
<td>Bird</td>
<td>15.741</td>
<td>14.525</td>
</tr>
<tr>
<td>Car</td>
<td>56.645</td>
<td>53.812</td>
</tr>
</tbody>
</table>

Figure 5.8 Average time of encryption and decryption images by using fuzzy logic and fuzzy graph algorithms
According to the above mentioned figures and tables, the proposed approaches can categorize the encryption algorithms on the basis of its ability to secure the confidential data against security attacks and its processing speed.

5.8. Conclusion

This chapter proposed two approaches of image encryption based on fuzzy sets theory. The first is described in section 5.6.1. The second is described in section 5.6.2. These approaches have been tested by different analysis measurements and showed their capability and quality for encrypting and decrypting images as demonstrated in tables and figures above.