Water Marking on Digital Image using Genetic Algorithm

Dr Cauvery N K
Department of Computer Science and Engineering, R V College of Engineering, Bangalore.

Abstract: Due to the explosion of data exchange on the Internet and the extensive use of digital media, it has been a great matter of interest in multimedia security and multimedia copyright protection. This work presents a secure algorithm for watermarking images, and a methodology for digital watermarking that may be generalized to video data. The watermark should be constructed as an independent and identically distributed. The main issues to be are considered during watermarking are Robustness and Fidelity. Robustness of the watermark implies the strength of the watermark against various image processing attacks. Fidelity is the factor which determines the quality of the image after embedding the watermark. The insertion of a watermark under this regime makes the watermark robust to signal processing operations and common geometric transformations provided that the original image is available and that it can be successfully registered against the transformed watermarked image. Genetic algorithm is used to identify the position for marking.

Key words: Water marking, Genetic algorithm, spatial and temporal domain.

Introduction: Watermarking is a concept of embedding a special pattern, watermark, into a document so that a given piece of information, such as the owner's or authorized consumer's identity, is indissolubly tied to the data. This information can later prove ownership, identify a misappropriating person, trace the marked document's dissemination through the network, or simply inform users about the rights-holder or the permitted use of the data. Digital watermarking is the process of embedding information into a digital signal in a way that is difficult to remove. The process of digital watermarking involves the modification of the original multimedia data to embed a watermark containing key information such as authentication or copyright codes. Thus embedded watermark should possess two key qualities, Robustness and Fidelity. Robustness implies the strength of the watermark to sustain any attacks. Fidelity accounts for the original quality of the watermark which is retained even after embedding the watermark. So in order to achieve both these constrains the concept of Genetic Algorithm is used. The Genetic Algorithm (GA) is a class of optimization algorithm that mimics the process of natural evolution. GA is an important optimization technique. Using the basic operations of GA the position to embed the watermark is found, such that the fidelity of the image is retained. Also embedding the watermark into multiple blocks of image, so that any image editing attacks does not remove the embedded watermark the robustness constraint can be achieved.

Digital Watermarking provides techniques to hide watermarks into digital content to protect it from illegal copy or reproduction. It should provide the qualities like imperceptibility, robustness, security of cover image. This work is an attempt to provide a conceptual understanding of the application of genetic algorithm to optimize the fidelity and Robustness of watermarked images. Genetic algorithm helps in searching appropriate locations in cover images to insert watermark. The fitness function is chosen so that optimal values are achieved for Robustness and Fidelity.

Caronni [1], suggests adding *tags*—small geometric patterns—to digitized images at brightness levels that are imperceptible. While the idea of hiding a spatial watermark in an image is fundamentally sound, this scheme may be susceptible to attack by filtering and re digitization. Tanaka *et al.* [2], [3], describe several watermarking schemes that rely on embedding watermarks that resemble quantization noise. Their ideas hinge on the notion that quantization noise is typically imperceptible to viewers. Macq and Quisquater [4], briefly discuss the issue of watermarking digital images as part of a general survey on cryptography and digital television. The authors provide a description of a procedure to insert a watermark into the least significant bits of pixels located in the vicinity of image contours.

Since it relies on modifications of the least significant bits, the watermark is easily destroyed. Further, their method is restricted to images, in that it seeks to insert the watermark into image regions that lie on the edge of contours. Chandramouli et al., [5], suggested using Multiple Description Coding (MDC) using which host image is fragmented into two sub images referred to as descriptions. These descriptions of the host image must be in such a way that some correlation exists between them. In this work, host image is represented as a sum of two sub images called even and odd descriptions. Genetic algorithms [6], [7], are search algorithms based on mechanics of natural selection and natural genetics. They combine survival of fittest among string structures with a structured yet randomized information exchange to form search algorithms with some of the innovative flair of human search. In this work the concept of multiple descriptions for images is used. The watermark is embedded in the spatial domain. Usually in spatial domain watermarking the fidelity of the image is the major constraint. By having multiple descriptions for the same image more than 50% of the pixels remain same after embedding the watermark. So fidelity constraint for an image is achieved by this technique. In order to achieve robustness the watermark is embedded into multiple blocks of an image. The position for embedding the watermark in each block is decided using the optimization technique of Genetic Algorithm.

Watermarking Concepts: There are various formats of images such as Joint Picture Experts Group (JPEG), Bitmap Images (BMP), Graphic Interchange Format (GIF), and Tagged Image File Format (TIFF). The basic component in all these formats is the pixel which builds up the image. In this work, pixels values are manipulated to embed the watermark. There are various image processing tools available viz Matlab, OpenCV etc. This project uses the OpenCV image processing functions to manipulate and build images. OpenCV is an open source computer vision library available. There are two domains by which a watermark can be embedded into an image, one in spatial domain and the other in frequency domain. In the spatial domain, one can simply insert watermark into a host image by changing the gray levels of some pixels in the host image. No transforms are applied to the host during watermark design or embedding. Combination with the host signal is based on simple operations, such as addition or replacement, and takes place directly in the pixel domain. Spatial watermarking can be applied using colour separation. In this way, the watermark appears in only one of the colour bands. This renders the watermark visibly subtle so that it is difficult to detect under regular viewing. However, the watermark appears immediately when the colours are separated for printing or xerography. This renders the document useless to the printer unless the watermark can be removed from the colour band. This approach is used commercially for journalists to inspect digital pictures from a photo-stockhouse before buying non-watermarked versions. Least significance bit modification is one of the spatial domain based watermarking scheme. In this technique the most straight-forward method of watermark embedding would be to embed the watermark into the least-significant-bits of the cover object. Given the extraordinarily high channel capacity of using the entire cover for transmission in this method, a smaller object may be embedded multiple times. Even if most of these are lost due to attacks, a single surviving watermark would be considered a success. LSB substitution however despite its simplicity brings a host of drawbacks. Although it may survive transformations such as cropping, any addition of noise or lossy compression is likely to defeat the watermark. An even better attack would be to simply set the LSB bits of each pixel to one, fully defeating the watermark with negligible impact on the cover object. Furthermore, once the algorithm is discovered, the embedded watermark could be easily modified by an intermediate party. In the frequency domain, one can insert watermark into the coefficients of a transformed image, for example, using the Discrete Fourier Transform (DFT), Discrete Cosine Transform (DCT) and Discrete Wavelet Transform (DWT), to be difficult to detect. In these methods, a watermark that one wishes to embed in distributed in overall domain of an original data, and the watermark, if it is once embedded, is hardly to be deleted. For transformed domain techniques, they have Hierarchical watermarking with Discrete Cosine Transform, sub- band watermarking techniques, modifying coefficient value in wavelet domain or after Discrete Fourier Transform. However, the embedded watermark in the coefficients of the transformed image will be somewhat disturbed in the process of transforming the image from its frequency domain to spatial domain because of deviations in converting real numbers into integers. The main strength offered by transform domain techniques is that they can take advantage of special properties of alternate domains to address the limitations of pixel-based methods or to support additional features. For instance, designing a watermarking scheme in the 8X8 Discrete Cosine Transform (DCT) domain leads to better implementation compatibility with popular image coding algorithms such as Joint Pictures Experts Group (JPEG). The scheme of embedding watermark in the frequency domain contains complex transformations and also cost in the loss of some of the pixel values. But as stated the earlier proposed schemes are less robust. In this work, the watermark is directly embedded into the spatial domain of the image. The robustness of the scheme is increased using Genetic algorithm. The problem of designing a feasible watermarking scheme can be viewed as an optimization problem with these conflicting goals: better robustness (watermark strength) and higher fidelity (media quality index). The fidelity requirement often limits the strength of the embedded signals, which consequently constraints the robustness of a watermarking scheme against common or malicious manipulations. Existing watermarking schemes make reasonable trade-offs among the two requirements. In this work Genetic Algorithm is used to optimize the Robustness and Fidelity requirements to watermark the image and to protect the quality of the image.

Genetic Algorithm: In this scheme, the image is decomposed into their colour components viz, R, G and B. One of these matrices are divided into odd and even banks of 8X8 each, such that pixel values present in the odd positions will go to the odd bank and similarly even bank. The data to be embedded is embedded into one of the banks say odd bank. The embedding position of the data to be watermarked into the image is found out using Genetic Algorithm. Using GA a key of 64-bit is generated by repeated application of the GA operators repeatedly for a certain number of generations or until the fitness requirement of the key is met. The main criteria behind the generation of this key are that the change introduced by embedding the watermark into it should bring about the least. The set bits in the key indicate the positions of the image where the watermark is embedded. The following requirements should be satisfied by watermarking algorithms.

i. Alterations introduced in the image should be perceptually invisible.
ii. A watermark must be undetectable and not removable by an attacker.
iii. A sufficient number of watermarks in the same image, detectable by their own key, can be produced.
iv. The detection of watermark should not require the original image information.

The main aim of this work is that the embedded watermark into the jpeg image is not perceivable to human eye and thus generated watermark is not accessible publicly. Robustness and Fidelity are the main points of concern. The application should be capable of handling the tradeoff between those two. This trade off is met when the position of embedding the watermark is optimally selected. This is achieved by the application of genetic algorithm which based on a particular fitness value. The functionality requirements mainly depict the functions that the system being developed should be able to perform and provide which includes: <u>Transparency</u>: Watermarking method shall be such that it is transparent to the end user. The watermarked content should be consumable at the intended user device without giving annoyance to the user.
<u>Security</u>: Watermark information shall only be accessible to the authorized parties. Only authorized parties shall be able to alter the Watermark content. Encryption can be used to prevent unauthorized access of the watermarked data.
Robustness: Watermarking must be robust enough to withstand all kinds for signal processing operations, “attacks” or unauthorized access. Any attempt, whether intentional or not, that has a potential to alter the data content is considered as an attack. Robustness against attack is a key requirement for Watermarking and the success of this technology for copyright protection depends on this.

The application will select specific blocks for embedding the watermark into the image file. The support tool OpenCV helps in getting the pixel values - R, G and B matrices - from the image into which watermark is embedded. The watermarking data characters are embedded into the pixels based upon the 1’s in the key block generated using GA. Thus the number of 1’s in the key block accounts to the performance of the watermarking scheme. The distribution of the 1’s in the key depends upon its fitness value which is calculated in the genetic algorithm process. This fitness value is the threshold percentage entered by the user which finally defines the quality of the image.

Design Constraints: These are those constraints that arose during the design of application. The length of the key block generated by Genetic Algorithm, size of the data which is to be embedded as watermark, domain of the image into which the watermark has to embedded and handling of images of different size. The length of the key block should be chosen such that the distribution of the 1's in the keys should be such that the change in the pixel positions from the original image block should be least. Thus even if there is a slight change in the pixel values, it is not perceivable by the eye. In addition to that, the size of the data blocks which is to watermarked needs to be appropriate. If the information to be inserted is very large, then it is difficult to embed that information within the blocks of the considered image. The watermark can be embedded either directly changing the pixel values of the image or by transforming the pixel values into the frequency domain. When pixel values are directly manipulated in spatial domain, there is chance of image getting distorted. Moreover the watermark embedded in the spatial domain can be easily attacked by image processing techniques. Thus it is embedded in various blocks of the image and GA is used in selection of the embedding position. In order to handle image of different size, an intermediate image of the same image is created and an auxiliary handler is assigned to it.

Detailed Design: This part of the report deals with the algorithmic representation of the implementation of the project. The algorithmic representation for this project is given in two phases. Phase -1 is Genetic Algorithm Phase and Phase-2 is Embedding of Watermark.

Algorithm for Phase 1

Step 1: Start
Step 2: Accept the file name which contains the data to be watermarked from user.
Step 3: Accept the threshold percentage, TP from the user.
Step 4: Initialise the first generation by reading the characters from the input file. Totally forty characters for ten keys.
Step 5: Initialise generation count to zero, MAXGEN to 50.
Step 6: Crossover ().
Step 7: Increment the generation count.
Step 8: Mutation ().
Step 9: Fitness ().
Step 10: Sort the keys obtained.
Step 11: Copy the first ten keys of the present generation to next generation.
Step 12: Add the statistics of the generation to the file.
Step 13: Go to step 7 if generation count is less than MAXGEN.
Step 14: Add the keys of final generation to final key text file, FINAL_KEYS.
Step 15: End

Algorithm for Phase 2

Step 1: Accept the image filename
Step 2: Accept the Colour preference
Step 3: Divide the image into blocks of size 8X16
Step 4: Separate the odd pixel values from each block to an 8X8 matrix, MAT
Step 5: Read a key from FINAL_KEYS
Step 6: Depending upon the bit values in the key, the corresponding position in the 8X8 matrix is XORed with the character read from the input text file
Step 7: Write back the changed matrix back to the corresponding block
Step 8: Repeat Step 4 to Step 6 for each block
Step 9: Build the image from the matrix
Step 10: Stop

Implementation Details: The implementation consists of two phases. The first phase is the generation of keys. The keys generated in this phase are used for deciding the embedding positions for watermarks. To achieve this, the image must be divided into fixed size blocks. The second phase takes care of image handling and embedding of watermarks. The colour matrix for which the water mark has to be embedded can be decided at run time.

Key Generation Phase: The first phase of implementation makes use of object oriented programming (OOP) environment i.e. C++. The main entity of this phase is 'key' which is considered as a real world object. Thus making use of the OOP, the key can be bound as an object along with its related characteristics like fitness value, length and the number of set bits and those functions which can be used to manipulate them. Similarly each generation has certain number of keys. So treating each generation as an object, its element i.e. key which is defined as an object along with the its functionalities like cross over and mutation can be bound to each generation. Thus the simplicity in manipulating and maintaining the keys and each generation is achieved. The key used in each generation is implemented as 64bit value. The main reason for choosing this length is most of the times the images are handled by dividing the image into blocks of size 8X8. So there would be 64 pixels in each block. Thus, 64 bits can be properly fit into 64 pixels during image manipulation phase. To start with genetic algorithm process, it must contain an initial population. To provide the initial population, an input file containing the list of authors (any input file can be given) is taken and first generation keys are derived from this. The threshold value is defined as the percentage of bits in the key that can have their bit value ‘1’. Higher the threshold value, higher will be the number of embedding pixels and hence higher will be the water mark strength. But compromise has to be made with the image quality in certain scenarios. The threshold value for the image blocks is taken from the user. Based on the threshold value, the fitness function for the key is defined. The fitness function is calculated using the threshold value given. The threshold value is given in percentage. The fitness function is calculated as the ratio of number of ones in the key to the maximum number of one’s allowed in the key. Thus the threshold value entered signifies the maximum number ones that the user wants in the final key. Once the fitness function is defined and calculated, the different operations of Genetic Algorithm viz., cross-over and mutation are repeated till the optimal solution for key is obtained. The saturation can be achieved after ‘N’ number of generations. The maximum number of generations till which the GA operations can continue is fixed to 50. If the saturation is achieved in the earlier generations only, then the operations of GA are stopped at that particular generation.

The second phase of implementation makes use of open Computer Vision (openCV) library for handling of images. It has many in built functions and data structures for handling the same. The implementation of application requires the image in such a way that all the pixels of the image should be accessible and it must be possible to manipulate them individually. The given image is divided into fixed size blocks in the beginning. The jpeg format of image is taken as input from the user. An approach known as Multiple Description Coding (MDC) is taken as the basis for handling of image. Initially the image is divided into two descriptions, known as odd description and even description. Each description is separated into two matrices. Only one of the descriptions is used for water marking and the other description is used as a reference. The advantage of this approach is that 50% of the image remains intact. Only the other half of the image is manipulated. So high fidelity for the image can be achieved. The description used for water marking is divided into fixed size blocks, each of size 8X8. Each block of image is considered for embedding of watermarks. The keys generated in the first phase are considered for deciding the embedding positions in each block.

Embedding Watermark: Final set of keys obtained from key generation phase are written in a file. These keys are read during the second phase. The key is checked for set bits in it. Whenever there is a ‘0’ bit in the key, the pixel corresponding to that key bit position is left as it is. But when there is a ‘1’ bit in the key, the corresponding pixel in the image block is considered for water marking purpose. The information to be embedded is considered for water marking with those pixel values. For each pixel, one character from information file is considered. The pixel value and the character information are XORed. Each image will be associated with three colour matrices, namely red, green and blue (RGB). In the implementation, any colour matrix can be considered for embedding the water mark information.

Challenges Encountered: This portion contains the main challenges that were faced during the coding phase of the project. **Key Generation phase (First phase):** One of the main challenges is to populate the initial generation. The keys of initial generation should have the bit values in a well distributed manner. There should not be long runs of zeros or ones as this would result in inefficient cross over results. In order to overcome this problem characters from the input text itself is considered for populating the initial generation. By this the keys generated after attaining the optimality will be a function of the input text file. This has got one more added advantage. For different input files different set of keys will be populated for initial generation and hence for the final generation. This makes the image processing attacks even more complex.

Choosing the fitness value is also one of the major challenges. The ultimate goal of the fitness function is to bring those key values which are good enough for deciding the watermark embedding positions. If there are long runs of ones, it will result in degradation of image quality. Hence, those keys are brought to the final generation which results in less degradation of image quality, thus achieving better fidelity. In this project, fitness function assigns the fitness value for the each key based on threshold value entered by the user. The keys thus generated are optimal keys, set bits of which indicate the positions for embedding watermark.

Embedding watermark (Second phase): In order to process the image, an attempt was made to use MATLAB. In this project VC++ is used for implementation. MATLAB provides the processed image details such as pixel values with '.m' extension. Processing '.m' files is not possible in VC++. In order to overcome this challenge, OpenCV is used which is compatible with VC++ and provides many user friendly functions to handle images. OpenCV has a data structure called IplImage which has many members as already explained in the earlier section. While building image, all these members have to be initialized. To achieve this, a copy of the input image is made and an image handler is initialized to it.

Results: This section contains the comparative study of the various inputs and their corresponding outputs. Different sets images are watermarked with various threshold values and watermark is applied to all three colour matrices.

Fig 1 shows the variations of the image baboon.jpg (Fig 1a) after embedding same watermark data successively (1b to 1e). After embedding watermark for the original image the resulting image is given as input to the next time and continued till fourth time. The watermark is embedded for R components every time.

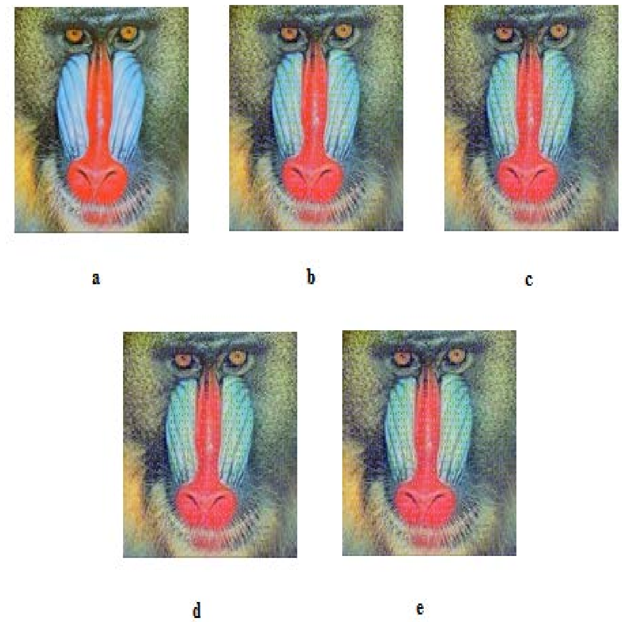

Fig 1 Image of baboon.jpg after embedding same watermark data four times in succession

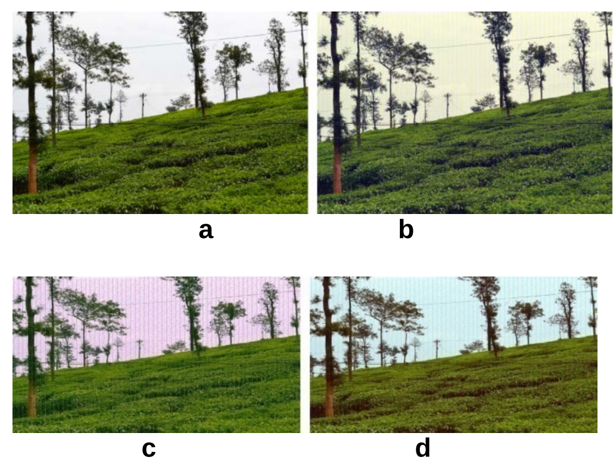

Fig 2 Image green.jpg after embedding watermark for R, G & B components with the threshold value of 25%

Fig 2b to 2d shows the embedding of watermark for the image green.jpg (Fig 2a) with the change in the colour matrix i.e R, G and B, for which watermark is embedded. The threshold value of 25% and the watermark data is kept constant every time.

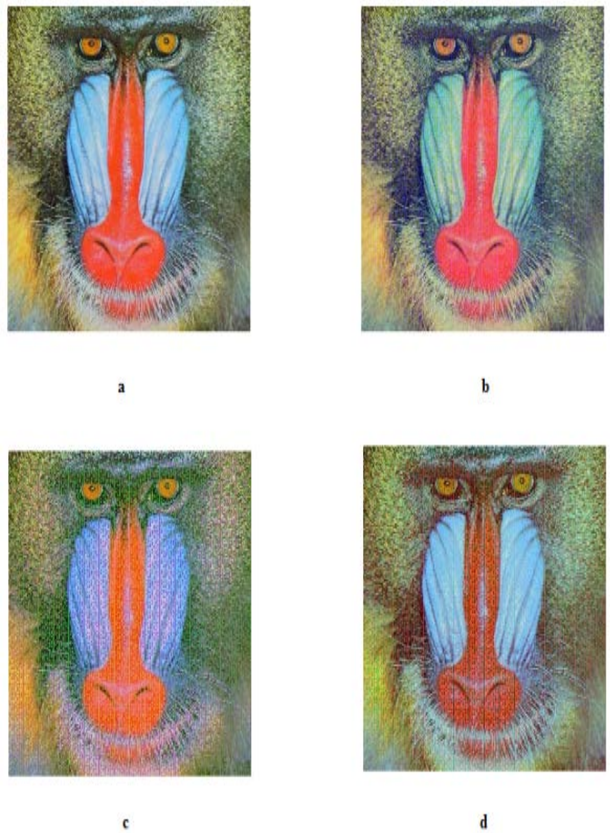

Fig 3 Image baboon256.jpg after embedding watermark for R, G & B components with the threshold value of 80%

Fig 3b to 3d shows the embedding of watermark for the image baboon256.jpg (Fig 3a) with the change in the colour matrix i.e R, G and B, for which watermark is embedded. The threshold value of 80% and the watermark data is kept constant every time.

Snapshots

Fig 4 shows Generations 1, 2, 10 and 11 with the details for each key such as their equivalent hex value, number of set bits and the fitness value for each key. The saturation is attained at generation 11.

```
                                      NO Of ones    FitnessValue(%)

Generation  1
Key1   aeca00004e860000                   16          100
Key2   2e8600004eae0000                   16          100
Key3   f67600004a040000                   15          93
Key4   4e0e000086ce0000                   15          93
Key5   86160000866e0000                   14          87
Key6   504e0000866e0000                   14          87
Key7   04b60000866e0000                   14          87
Key8   16260000866e0000                   14          87
Key9   1622000026ae0000                   13          81
Key10  86b2004086160100                   13          81

Average        89

Generation  2
Key1   ae8680004e2c0000                   16          100
Key2   aeca00004e860000                   16          100
Key3   4e0e000086ce0000                   15          93
Key4   f67600004a040000                   15          93
Key5   504e0000866e0000                   14          87
Key6   04b60000866e0000                   14          87
Key7   9622000026ae0000                   14          87
Key8   16160000866e0000                   14          87
Key9   16260000866e0000                   14          87
Key10  04b60000866e0000                   14          87

Average        90

........

........

Average        99

Generation  10
Key1   acae81a05e2c0200                   16          100
Key2   162700008e6a0020                   16          100
Key3   ae8680004e2c0000                   16          100
Key4   aeca00004e860000                   16          100
Key5   16260001876e0000                   16          100
Key6   960e000086ce0020                   16          100
Key7   962700008e6a0000                   16          100
Key8   1626000186ce0040                   16          100
Key9   4a1e40008e6a0000                   16          100
Key10  960e00008e6a0000                   15          93

Average        99

Generation  11
Key1   16260001876e0000                   16          100
Key2   1626000186ce0040                   16          100
Key3   960e00008e6a0020                   16          100
Key4   4a1e40008e6a0000                   16          100
Key5   1626000186ce0040                   16          100
Key6   962f0200aeea0000                   16          100
Key7   960e000086ce0020                   16          100
Key8   16260001876e0000                   16          100
Key9   aeca00004e860000                   16          100
Key10  ae8680004e2c0000                   16          100

THRESHOLD PERCENTAGE    25
Saturation achieved at Generation  11
```

Fig 4 Key summary file for the threshold value of 25%

```
                                  NO Of ones   FitnessValue(%)

Generation  1
Key1  aece000086ce0000                18         35
Key2  aeca000026ae0000                17         33
Key3  aeca0000866e0000                17         33
Key4  aee200000eae0000                17         33
Key5  86b2000056f60000                17         33
Key6  ce4e000076860000                17         33
Key7  aeca00004e860000                16         31
Key8  2e8600004eae0000                16         31
Key9  f67600404a040100                15         29
Key10  4e0e000086ce0000               15         29

Average        32


Generation  2
Key1  f6b2000056f60000                20         39
Key2  ce4e00007e860000                18         35
Key3  c6b2000056f60000                18         35
Key4  aece000086ce0000                18         35
Key5  ce4e000076860000                17         33
Key6  86b2000056f60000                17         33
Key7  aee200000eae0000                17         33
Key8  aeca0000866e0000                17         33
Key9  2eca8000262c0000                17         33
Key10  b67600404a040100               16         31

Average        34


............

............


Generation  49
Key1  feea87b096f684ad                36         70
Key2  f7ea49b096b6c8e9                35         68
Key3  f7a8c9b096f6cca9                35         68
Key4  feea87b096f694a9                34         66
Key5  aee249b096f6ccad                34         66
Key6  f7a885b096f6cca9                34         66
Key7  f7a885b096f6c4a9                33         64
Key8  f7a885b096f6c4a9                33         64
Key9  f7a885b096f694a9                33         64
Key10  f7a885b096f6c4a9               33         64

Average        66


Generation  50
Key1  f7e8c5b096f6ccad                37         72
Key2  feea87b096f684ad                36         70
Key3  f7a885b096f6ccad                35         68
Key4  f7a8c9b096f6cca9                35         68
Key5  f7ea49b096b6c8e9                35         68
Key6  f7a885b096f6cca9                34         66
Key7  f7a885b096f6cca9                34         66
Key8  aee249b096f6ccad                34         66
Key9  feea87b096f694a9                34         66
Key10  f7a885b096f6c4a9               33         64

Average        67


 THRESHOLD PERCENTAGE     80
 Saturation achieved at Generation   50
```

Fig 5 Key summary file for the threshold value of 80%

Fig 5 shows Generations 1, 2, 49 and 50 with the details for each key such as their equivalent hex value, number of set bits and the fitness value for each key. The saturation is attained at generation 50.

Fig 6: output file

Fig 6 is the output for a specific file with red colour to be used for embedding.

Inferences

i. When the watermark is embedded for the same image repetitively, the quality of the image gets reduced.
ii. When the watermark is embedded considering the different colour matrices of image, the change in fidelity of the image varies.
iii. The change in image quality depends on the property of the image in addition to color matrices.
iv. The quality of the image changes with the change in threshold value. Higher the threshold value, higher is the degradation in the image quality.

Conclusion: In this project, attempts have been made to demonstrate the utility of genetic algorithm in the area of improving the fidelity and robustness of digital watermarking. The role of fitness function proposed is to ensure the optimization of fidelity or robustness. As the genetic algorithms are very promising in field of optimization applications so they may be employed in digital watermarking area also to optimize its desirable characteristics. Several variations in genetic algorithm may be tried and tested for performance in fidelity and robustness optimization area which forms the further scope of research.

Limitations: There are certain limitations regarding the features implemented by the project. The main limitations include the following.

i. The scope of the project is restricted only to images.
ii. The positions for embedding watermark are decided statically without considering image properties.
iii. The watermark is embedded to every block of the image. Though for higher threshold values, higher robustness is achieved, the fidelity of the image gets reduced with the

increasing threshold values.

iv. If the input image has more percentage of some particular color, then the color matrix to which the watermark is embedded decides the image quality.

Further Enhancements: The application can be enhanced to decide the embedding positions based on the pixel values dynamically.

i. The watermarking embedding scheme can be extended to work in frequency domain.

ii. None of the current watermarking schemes can resist all attacks. In order to tackle this problem, a hybrid watermarking scheme can be used which includes number of watermarking techniques that can resist most of the attacks.

References

[1] G. Caronni, “Assuring ownership rights for digital images,” in *Proc. Reliable IT Systems, VIS’95*.

[2] K. Matsui and K. Tanaka, “Video-steganography,” in *Proc. IMA Intellectual Property Project*, 1994, vol. 1, pp. 187–206.

[3] K. Tanaka, Y. Nakamura, and K. Matsui, “Embedding secret information into a dithered multi-level image,” in *Proc. 1990 IEEE Military Communications Conf.*, 1990, pp. 216–220.

[4] B. M. Macq and J.-J. Quisquater, “Cryptology for digital TV broadcasting,” in *Proc. IEEE*, vol. 83, pp. 944–957, 1995.

[5] Chandramouli R, Graubard Benjamin and Richard Collin R, “A multiple description frameworkfor oblivious watermarking,” in *Proceedings of Security, Watermarking and Multimedia,* 2001.

[6] Zhicheng Wei, Hao Li, Jufeng Dai, Sashuang Wang “ Image Watermarking Based on Genetic Algorithm” IEEE Transaction, 2006.

[7] Chien-Chang Chen and Cheng-Shian Lin “ A Genetic algorithm based nearly optimal image authentication approach”. International Journal of Innovative computing, Information and Control, Volume 3, No. 3, pp 631-640, June 2007.