ISSN 2250-3153

A SNR based Comparative Study of JPEGmini and a Novel Proposed Algorithm

Anup Prabhu*, Jai Naik*, Kushaal Nayak*, Anjyot Talauliker*, Samarth Borkar**

* Department of Electronics & Telecommunication, Goa College of Engineering, Goa- India
** Assistant professor, Department of Electronics and Telecommunication Engineering, Goa College of Engineering, Goa, India

Abstract- In this paper a comparative study is has been done on JPEGmini (online service for compressing images) and a proposed algorithm for image compression. We are focusing on the Quantization method in image compression which results in the actual reduction of the size of the image. The proposed algorithm is based on modifying the Quantization method. Performance will be analyzed based upon the compression ratio and the image quality.

Index Terms- Quantization, Lossless, DCT, High resolution, JPEG, JPEGmini.

I. INTRODUCTION

The objective of image compression is to reduce irrelevancy and redundancy of the image data to store or transmit data in an efficient form thus saving storage costs and transmission time. Image compression can be lossless or lossy. Lossless compression means that you are able to reconstruct the exact original data from the compressed data whereas it is difficult in lossy compression. Here we are trying to compress image by quantizing DCT values of 8x8 pixel block.

II. LITERATURE REVIEW

In this section we have reviewed the existing tools and technologies in the field of JPEG standard. One of the things that the JPEG realized is that many digital images have very gradual changes in the intensity over most of the image. Besides this they realized that the human eye can only differentiate between similar shades of light-intensity, or luminance, to a certain extent[1].

The JPEG discovered that besides removing the most of the variations in luminance, most of the slight changes in color (from pixel to pixel) can be removed and still end up with a very good representation of the image. This way, instead of storing the individual pixel's color and intensity, only the gradual changes of color and luminance (across the picture) need to be stored which results in smaller file size. In order to get to a place where they could do this, the JPEG implemented The Discrete Cosine Transform.

The fundamental idea behind JPEG, and for that matter any picture compression is that you can take the values stored in a picture matrix and transform those numbers from one basis to another, where the new basis stores your relevant information in a more compact form. For the JPEG, the original basis was the two-dimensional spatial basis, where, as stated above, every entry in the picture matrix represents an actual square pixel which has a spatial position in the picture (e.g. Figure 1). The basis that would store the image values more compactly was the frequency basis, where the frequencies represent changes in the values of luminosity. Higher frequencies represent, quick changes if luminosity from pixel to pixel, and the low frequencies represent gradual changes across the entire picture. The way we get from one basis is through a transform, and the way we get from the spatial domain and into the frequency domain is through the Discreet Cosine Transform (DCT)[1].

Fig.1 Image and blocks of pixels
(source: JPEG Compression,Ben O'Hanen and Matthew Wisan)

JPEGmini:

JPEGmini is a photo recompression technology, which significantly reduces the size of photographs without affecting their perceptual quality. The technology works in the domain of baseline JPEG, resulting in files that are fully compatible with any browser, photo software or device that support the standard JPEG format.

JPEGmini is capable of reducing the file size of standard JPEG photos by up to 80% (5X), while the resulting photos are visually identical to the original photos. The JPEGmini algorithm imitates the perceptual qualities of the human visual system, ensuring that each photo is compressed to the maximum extent possible by removing redundancies, without creating any visual artifacts in the process. This enables fully automatic, maximal compression of photos with no human intervention required.

III. IMPLEMENTATION

The technique of Lossless compression is implemented on colour images having high resolution. The implementation part is done using Matlab software. The RGB image is read and stored in Matlab as a 3-dimensional matrix consisting of three 2-dimensional matrices, each comprising the respective pixel

ISSN 2250-3153

values of the R, G and B components. Next, colour space conversion[12] is done. Here the image is converted to the YUV color space from the RGB colour space[3].The 3-D matrix now consists of three Y(luminance), Cb & Cr (Chrominance) matrices. The Y(luminance) matrix is then taken and divided into 8x8 blocks i.e. blocks containing 8 rows and 8 columns of the Y matrix. The pattern followed here is that first the top leftmost 8x8 block is taken, then we move from left to right and then up to down. On each of the blocks 2-dimensional Discrete Cosine Transform is applied[4].The same process is then carried out on both the chrominance components. Thus we obtain a DCT matrix consisting of the transformed values of all the elements of the original matrix.

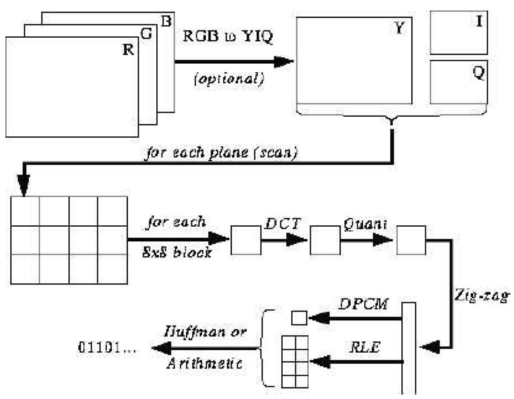

Fig-2: Basic block diagram for Image Compression
(source:Basic JPEG Compression Pipeline, Cardiff University)

The next step in compressing the image is quantizing each element of the transformed matrix. This is done by dividing the 8x8 blocks of the transformed matrix by a quantization matrix of the same size. The 8x8 blocks are taken in a similar fashion as described above. Here we have used two separate quantization matrices for the luminance and the two chrominance components. The use of two different matrices is based on the fact that the Human Visual System is more sensitive to luminance as compared to the colour components. The chrominance components convey information about colour and hence can be quantized more effectively (suppressed more) than the luminance components. In the DCT matrix the low frequency components are present in the top left part of the matrix and the higher frequency components in the lower right part. Eye is most sensitive to low frequencies (upper left corner) and less sensitive to high frequencies (lower right corner)[5], these higher frequency values can be quantized to zero. In the DCT matrix the first element of each 8x8[6] block will consist of the dc component. This dc coefficient is usually very high in magnitude as compared to the rest of the 63 values. The compressed values are then rounded. The following equation sums up the quantization process, where c_{ijk} is the value in dct matrix and q_{ij} is from quantization matrix.[7]

$$u_{ijk} = Round[c_{ijk}/q_{ij}] \qquad ......(1)$$

The maximum absolute error (MAE) is calculated as

$$MAE = max\ |f(x,y)\ - f\ '(x,y) \qquad ......(2)$$

Where $f\ (x,\ y)$ is the original image data and $f'(x,\ y)$ is the compressed image value.

The formulae for calculated image matrices are:

$$MSE^{[9]} = \frac{1}{M*N}\sum_{n=1}^{N}\sum_{m=1}^{M} f(x,y) - f\ '(x,y) \qquad ......(3)$$

$$RMSE = \sqrt[2]{MSE} \qquad ......(4)$$

Where M and N are the matrix dimensions in x and y, respectively.
Signal-to noise-ratio (SNR) measures are estimates of the quality of a reconstructed image compared with the original image

$$SNR = 10\ log \left|\frac{\sum_{n=1}^{N}\sum_{m=1}^{M} f(x,y)^2}{M*N*MSE}\right| \qquad ......(5)$$

$$PSNR^{[13]} = 20\ log \left|\frac{255}{RMSE}\right| \qquad .....(6)$$

To find the most efficient quantization matrix we first tried to find out on average what is the value of the dc coefficient. After implementing the dct on various images we came to the conclusion that the dc coefficient can range from anywhere between few hundred to about a thousand in magnitude. Since the dc coefficient contains most of the information, it has to be quantized by a very small value. The following quantization matrix was used to quantize the image:

2	4	6	8	10	12	14	16		
		4	6	8	10	12	14	16	18
		6	8	10	12	14	16	18	20
Q	=	8	10	12	14	16	18	20	22
		10	12	14	16	18	20	22	24
		12	14	16	18	20	22	24	26
		14	16	18	20	22	24	26	28
		16	18	20	22	24	26	28	30

The dc coefficient is divided by two and the rest of the values are quantized by increasing multiples of two. Next all the values except the top left 4x4 block are increased by a factor of two and applied to the chrominance matrices. According to the quality of the image and the reduction in the size we proceed with the next iteration. If the quality is maintained we increase the values of the quantization matrix. We also checked the size compressed image. If the size started to increase after initially reducing we accordingly tried to adjust the quantization matrix. Thus we have tried to obtain an optimum quantization matrix which gives high compression ratio and also almost no visible loss in the image quality.

The software used for image processing was Matlab from Mathworks, Inc.

For a comparative study we used JPEGmini online service for compressing images. We compressed the same images using our proposed algorithm i.e. using our quantization matrix. The following are the results that we have obtained:

Table 1
Comparison of Compression Ratio

	JPEGmini	Proposed Algorithm
Image 1 (9.62 MB)	3.25	5.37
Image 2 (4.7 MB)	3.64	6.37
Image 3 (2.35 MB)	4.7	5.54

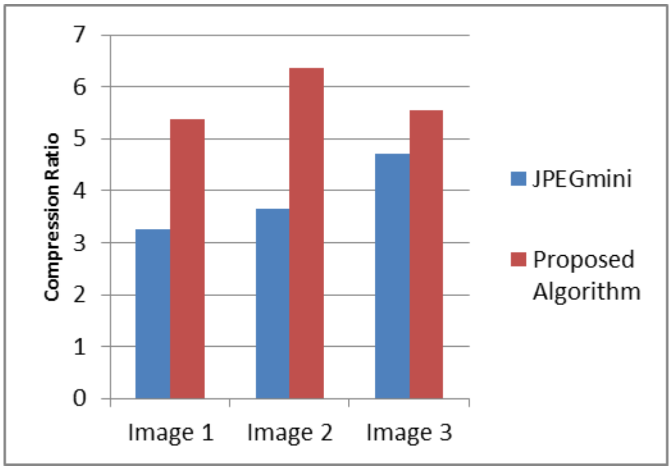

Fig. 3: Comparison of Compression Ratio

The following are results for Signal to Noise Ratio(SNR):

Table 2: Comparison of SNR

	JPEGmini	Proposed Algorithm
Image 1 (9.62 MB)	130.06	118.24
Image 2 (4.7 MB)	118.68	117.24
Image 3 (2.35 MB)	129.89	112.91

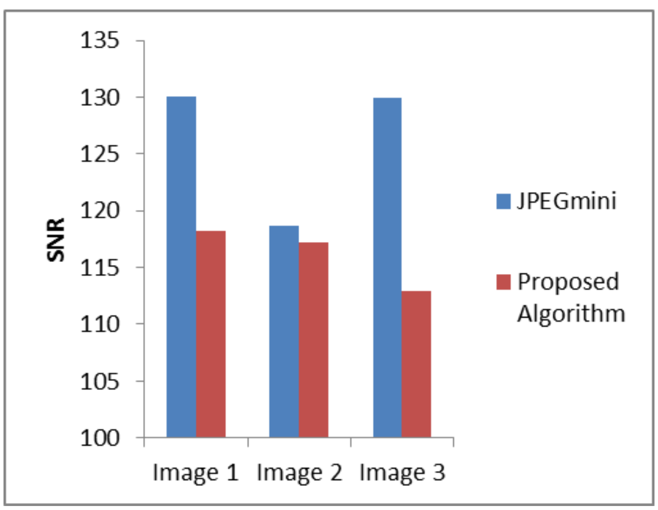

Fig. 4: Comparison of SNR

Usually high values for Compression Ratio (the ratio of original image size to that of compressed image size) and SNR are desired. With the obtained results for Compression Ratio we can say that our algorithm gives better compression than JPEGmini. On the other hand JPEGmini has better values of SNR (Signal to Noise Ratio) as mentioned in Table 2. Still the compressed images obtained through our approach do not show any loss in visual quality. Though we get less SNR values the compressed images have lesser size as compared to JPEGmini and there is no visual loss of data.

The following two images are obtained from JPEGmini compression and our approach respectively:

Fig.4 Compressed image using JPEGmini (Image 3)

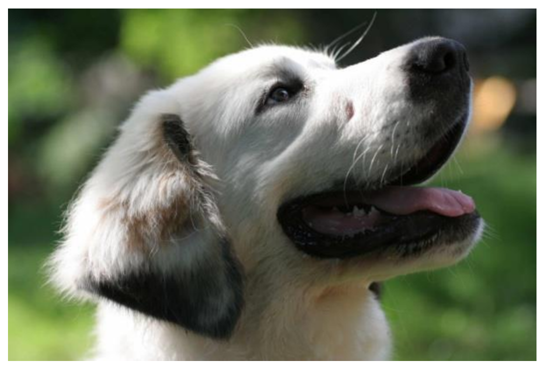

Fig.5 Compressed image using proposed algorithm(Image 3)

Both the images look exactly the same, but our results prove that better compression is achieved using our approach.

IV. CONCLUSION

Our proposed algorithm for compressing images gives higher compression than JPEGmini without any degradation of images. On an average improvement in the compression ratios of about 50% is obtained from our algorithm. Additionally the algorithm gives very good results for high resolution images, as compared to low resolution ones.

ACKNOWLEDGMENT

We wish to thank Prof. (Dr.) R.B. Lohani, Principal and head of Electronics and Telecommunication Department, of Goa College of Engineering for constantly motivating us in our journey through this project and for his support and encouragement.

We would like to thanks our parents who taught us the value of hard work by their own examples. They rendered us enormous support during the whole period of our life.

REFERENCES

[1] JPEG Compression,Ben O'Hanen and Matthew Wisan December 16, 2005.

[2] DCT quantization matrices visually optimized for individual Images by Andrew B. Watson, MS 262-2 NASA Ames Research Center Moffett Field, CA 94035-1000.

[3] A Comparative Study of Image Compression Methods by Ashwin Swaminathan, Gaurav Agarwal.

[4] An introduction to JPEG compression using MATLAB, Arno Swart 30 October, 2003.

[5] Discrete Cosine Transform,by Ken Cabeen & Peter Gent.Math 45 College of Redwoods .

[6] An introduction to JPEG compression using MATLAB, Arno Swart 30 October, 2003.

[7] Wallace, Gregory k. The JPEG still picture compression standard. IEEE Transaction on Consumer Electronics.

[8] Lossless Image Compression, by C.M. Liu, Perceptual Signal Processing Lab, College of Computer Science, National Chiao-Tung University.

[9] Quantization by Yao Wang, polytechnic University, Brooklyn, New York 11201.

[10] Offical miniJPEG site: http://www.jpegmini.com

[11] Estimation with Minimum Mean Square Error Alan V. Oppenheim and George C. Verghese, 2010

[12] Colour Space Conversions drian Ford (ajoec1@wmin.ac.uk <defunct>)and Alan Roberts (Alan.Roberts@rd.bbc.co.uk).

[13] PSNR-BASED OPTIMIZATION OF JPEG-BASELINE COMPRESSION ON COLOR IMAGES, Linfeng Guo, BrainMedia LLC, New York, New York 10016, USA

AUTHORS

First Author – Anup Prabhu was born in 1991. He is pursuing his BE Degree from the Department of Electronics and Telecommunication Engineering Goa University, Goa, India. His research interests are in Image processing and Video Processing., Email: anupprabhu22@gmail.com

Second Author – Jai V. Naik was born in 1992. He is pursuing his BE Degree from the Department of Electronics and Telecommunication Engineering Goa University, Goa, India. His research interests are in Image processing and Video Processing. Email: jvnaik06@gmail.com

Third Author – Kushaal Nayak was born in 1991. He is pursuing his BE Degree from the Department of Electronics and Telecommunication Engineering Goa University, Goa, India. His research interests are in Image processing and Video Processing., Email: kushaaln.00@gmail.com

Fourth Author – Anjyot Talauliker was born in 1992. He is pursuing his BE Degree from the Department of Electronics and Telecommunication Engineering Goa University, Goa, India. His research interests are in Image processing and Video Processing. Email: anjyott5@gmail.com

Fifth Author – Samarth Borkar received his B.E. and M.E., degrees in Electronics & Telecommunication Engineering from Goa University. Presently he is serving as Assistant Professor at E&TC Dept of Goa College of Engineering. His research area includes Image and Video Processing. He is member of IETE and has published and presented numerous papers at National and international conferences. Email:Samarth@gec.ac.in