IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

NOVEL APPROACH FOR IMAGE SCALING USING INTERPOLATION TECHNIQUES

Dr. H S Prasantha (0000-0003-4739-517X)
Professor
Department of Electronics & Communication Engineering
Nitte Meenakshi Institute of Technology, Bangalore, Karnataka, India

Abstract: The Paper discusses the image scaling application required to zoom the image with interpolation. The paper also proposes the novel approach for image scaling and comparing the novel approach with the other existing interpolation techniques. The scope of the paper is limited to use the different image scaling techniques to fit the image on the display devices of different resolutions. Different image scaling techniques such as nearest neighbour and bilinear are considered for the comparison with the proposed novel approach. The proposed method, nearest neighbour and bilinear interpolation techniques is ported on a digital signal processor DM 642 to measure the computation complexity and memory requirements. The different image scaling techniques considered are compared on the basis of objective and subjective measures.

***Index Terms* – Interpolation, nearest neighbor, bilinear, bicubic, zoom.**

1. INTRODUCTION

Due to the restriction of resolution or display size of mobile devices, spatially downscaled or down sampled video/image is provided for the display. But user may not be satisfied the small sized video/image and may want to see the video/image with larger resolution. The better solution to satisfy the viewer is to define the region of interest to view the video/image. Even if the display size is same, semantically meaningful region can be defined with better resolution. In most of the videos/images, certain region is more important than the other regions in the video/image. For example, people in the picture are more meaningful than background.

The video/images, binary images or pseudo-binary images such as documents and signatures are major inputs required to process for mobile vision applications. To process these inputs on mobile devices, computational complexity or computation time and memory required to fit the image scaling techniques become crucial. Hence the fast interpolation techniques which gives reasonably good quality are required to process (scale) these inputs are required. When an image is geometrically transformed for scaling purpose, a pixel in the new image is often projected back to a point with non-integer coordinates in the original image.

To illustrate the necessity of fast algorithms for image scaling, consider the linear interpolation defined with four neighboring points with integer coordinates (Q11, Q12, Q21, and Q22). To estimate the value of point P from its four neighboring points with coordinates (Q11, Q12, Q21, and Q22), the linear interpolation in X direction is performed to obtain f(x, y)

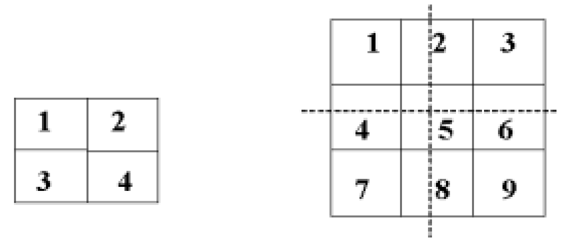
$$\boldsymbol{f(x,y) \approx \frac{f(Q_{11})}{(x_2-x_1)(y_2-y_1)}(x_2-x)(y_2-y) + \frac{f(Q_{21})}{(x_2-x_1)(y_2-y_1)}(x-x_1)(y_2-y) + \frac{f(Q_{12})}{(x_2-x_1)(y_2-y_1)}(x_2-x)(y-y_1) + \frac{f(Q_{22})}{(x_2-x_1)(y_2-y_1)}(x-x_1)(y-y_1)} \quad (1)$$

From the equation (1), we can estimate how many floating point multiplication and addition operations are required to find $f(x,y)$. For linear interpolation with $x_2 - x_1 = 1$ and $y_2 - y_1 = 1$, we need to calculate $f(Q_{11})(x_2 - x)(y_2 - y)$, $f(Q_{21})(x - x_1)(y_2 - y)f(Q_{12})(x_2 - x)(y - y_1)$ and $f(Q_{22})(x - x_1)(y - y_1)$, each of which requires two floating point subtraction and multiplication operations. Therefore, each pixel in the interpolated image will requires $(2\,x\,4 + 3)$ floating point additions (subtractions), and $(2\,x\,4)$ floating point multiplications.

Hence the interpolation of an image at VGA (640 × 480) resolution requires 640 × 480 × 11 =3,379,200 floating point additions, and 640 × 480 × 8 = 2,457,600 floating point multiplications. If the algorithm is tested for interpolation of an image of size 320×240 (Source Input Format) to VGA resolution on hardware, the process takes more time. The reason of the slow speed is that mobile devices often use software emulation to process floating point calculations instead of using a specific hardware floating point processor as is typical on a PC. Hence it is required to reduce the computational complexity to fit the algorithm for video/image scaling. Also the mobile devices have limited memory and it is required to estimate the memory requirements. Hence it is required to use the efficient image scaling algorithm for the mobile applications which is more efficient in-terms of computational requirements and memory requirements compared to the existing algorithms for image scaling.

II. RESEARCH METHODOLOGY

Proposed algorithm is based on the idea that any pixel in the new image will map back to, not a point, but a square in the old image. Simple example of expanding 2×2 pixel image by 1.5 in both the x and y axis to yield a 3×3 pixel image,



Old image Image with four original pixels overlaid onto the nine new pixels

Figure 1: Proposed approach for image scaling

The square will overlap pixels in the old image and new pixels are computed as the percentage of overlap of the old one. If the percentage of overlap for each of these pixels is known, it is possible to calculate the value of the new pixel by weighting each pixel from the old pixel by its overlap percentage. The weight factor indicates the weight-age given to old pixels to estimate the value of new pixels. If the percentage of overlap is not known, weight factor is defined.

Considering the pixels in the old image as o_1,o_2,o_3, o_4 and pixels in the new image as N_1, N_2, N_3 , N_4, N_5, N_6, N_7, N_8 and N_9

The new pixels can be found by measuring the percentage of overlapping of old pixels as shown in fig 5.2. The weight factor is defined to measure the percentage of overlapping of the pixels.

The new pixels N_1,N_3 ,$N_7$and N_9map back to the pixels in the old image ie.,
N_1=o_1 , N_3=o_2 , N_7=o_3 and N_9=o_4
N_2, N_4, N_5, N_6, N_8 can be computed by measuring the percentage of overlapping of the pixels o_1,o_2,o_3 and o_4. The percentage of overlapping is defined using weight factor. The new pixel N_2, N_4, N_5, N_6, N_8 depends on old pixel o_1,o_2,o_3, $o_4$and the weight factor. Efficient computation of the new pixel values can be done by using matrix method. The matrix method is chosen since the algorithm is ported on a DSP which consumes less time for the matrix multiplications.

Define matrix A given by

$$A = \begin{bmatrix} a_1 & a_2 \\ a_3 & a_4 \\ a_5 & a_6 \end{bmatrix}$$

Martix A and A^T are required to obtain 3 X 3 matrix represents the pixel values of the new image from the existing 2 X 2 matrix reperents the pixel values in the old image,

$$\begin{bmatrix} a_1 & a_2 \\ a_3 & a_4 \\ a_5 & a_6 \end{bmatrix} \begin{bmatrix} o_1 & o_2 \\ o_3 & o_4 \end{bmatrix} \begin{bmatrix} a_1 & a_3 & a_5 \\ a_2 & a_4 & a_6 \end{bmatrix} = \begin{bmatrix} N_1 & N_2 & N_3 \\ N_4 & N_5 & N_6 \\ N_7 & N_8 & N_9 \end{bmatrix}$$

$$(a_1 x\ o_1 + a_2\ x\ o_3)a_1 + (a_1 x\ o_2 + a_2\ x\ o_4)a_2 = N_1 \quad (2)$$
$$(a_1 x\ o_1 + a_2\ x\ o_3)a_3 + (a_1 x\ o_2 + a_2\ x\ o_4)a_4 = N_2 \quad (3)$$
$$(a_1 x\ o_1 + a_2\ x\ o_3)a_5 + (a_1 x\ o_2 + a_2\ x\ o_4)a_6 = N_3 \quad (4)$$
$$(a_3 x\ o_1 + a_4\ x\ o_3)a_1 + (a_3 x\ o_2 + a_3\ x\ o_4)a_2 = N_4 \quad (5)$$
$$(a_3 x\ o_1 + a_4\ x\ o_3)a_3 + (a_3 x\ o_2 + a_3\ x\ o_4)a_4 = N_5 \quad (6)$$
$$(a_3 x\ o_1 + a_4\ x\ o_3)a_5 + (a_3 x\ o_2 + a_3\ x\ o_4)a_6 = N_6 \quad (7)$$
$$(a_5 x\ o_1 + a_6\ x\ o_3)a_1 + (a_5 x\ o_2 + a_6\ x\ o_4)a_2 = N_7 \quad (8)$$
$$(a_5 x\ o_1 + a_6\ x\ o_3)a_3 + (a_5 x\ o_2 + a_6\ x\ o_4)a_4 = N_8 \quad (9)$$
$$(a_5 x\ o_1 + a_6\ x\ o_3)a_5 + (a_5 x\ o_2 + a_6\ x\ o_4)a_6 = N_9 \quad (10)$$

The number of unknown in the above equations are a_1, a_2, a_3, a_4, a_5 and a_6. Also, $N_2, N_4, N_5, N_6$and N_8 are unknown quantities. But $N_2, N_4, N_5, N_6$and N_8 values depends on the percentage overlapping of the known pixels $N_1, N_3, N_7$and N_9 whose value can be found by using Gaussian elimination procedure.

The algorithm allows to choose $a_1 = 1$, $a_2 = 0$, $a_5 = 0$ and $a_6 = 1$. Choose a_3 and a_4 and the values depends on the weight factor. The assumption made reduces the number of unknowns and reduces the computational complexity.

III. IMPLEMENTATION DETAILS

Different image file formats of different resolutions are considered for the experimentation. The different file formats such a TIFF, JPEG, PNG, BMP, etc are considered for experimentation. Extensive experimentation is done using different file formats of different resolutions and on the different interpolation algorithms such as nearest neighbor and bilinear. Proposed method is considered and the experiments are conducted and the comparison of the result is done with nearest and bilinear interpolation techniques using different comparison parameters. The different comparison parameters considered are MSE, PSNR, quality index, average difference, maximum difference, structural content, normalized absolute error and normalized cross correlation. Also, the algorithms are compared by estimating the memory required to fit the code and computation time required for computation.

3.1 Mean Square Error (MSE)

The Mean Square Error measures the difference between the frames which is usually applied to Human Visual System. It is based on pixel-pixel comparison of the image frames. Minimizing the MSE is equivalent to least-squares optimization in a minimum energy sense, for which many mathematical tools are available. MSE is still popular despite its inability to reliably predict perceived quality across different scenes and distortion types.

$$d(X,Y) = \frac{\sum_{i-1\ \ j-1}^{m,n}(X_{i,j}-Y_{i,j})^2}{mn} \qquad (11)$$

3.2 Peak Signal to Noise Ratio (PSNR)

PSNR is measured on a logarithmic scale and depends on the mean squared error (MSE) of between an original and an impaired image or video frame, relative to $(2^n - 1)^2$ (the square of the highest-possible signal value in the image, where *n* is the number of bits per image sample).

$$PSNR_{db} = 10 log_{10}\left(\frac{(2^n-1)^2}{MSE}\right) \qquad (12)$$

PSNR can be calculated easily and quickly and is therefore a very popular quality measure, widely used to compare the 'quality' of images.

3.3 Quality Index

Image quality index measurement does not depend on the image being tested, the viewing conditions or the individual observers. More importantly it must be applicable to various image processing applications and provide meaningful comparison across different types of image distortions.

$$\boldsymbol{Q} = \frac{4\sigma_{xy}\overline{x}\overline{y}}{(\sigma^2{}_x+\sigma^2{}_y)[(\bar{x}^2)+(\bar{y}^2)]} \qquad (13)$$

Where

$$\bar{x} = \frac{1}{N}\sum_{i=1}^{N} x_i \ , \qquad \bar{y} = \frac{1}{N}\sum_{i=1}^{N} y_i \qquad (14)$$

$$\sigma^2{}_x = \frac{1}{N-1}\sum_{i=1}^{N}(x_i-\bar{x})^2,\ \sigma^2{}_y = \frac{1}{N-1}\sum_{i=1}^{N}(y_i-\bar{y})^2, \qquad (15)$$

$$\sigma_{xy} = \frac{1}{N-1}\sum_{i=1}^{N}(x_i-\bar{x})(y_i-\bar{y}) \qquad (16)$$

The dynamic range of Q is [-1,1].the best value of 1 is achieved if and only if $y_i = x_i$ for all i=1,2……….N. the lowest value of -1 occurs when $y_i = 2\bar{x} - x_i$ for all i=1,2…..N.

3.4 Average Difference

Average difference is calculated by subtracting the original image with that of reconstructed image, and dividing the sum of the resultant matrix with that of the size of the image.

$$average\ difference = \frac{error[i][j]}{M*N} \qquad (17)$$

Where, $error[i][j] = a[i][j] - b[i][j]$ M, N=size of the image.

3.5 Maximum Difference

Maximum difference is calculated by subtracting the original image with that of reconstructed image, and taking the maximum value among the obtained result.

$$maximum\ difference = a[i][j] - b[i][j] \qquad (18)$$

Where, $a[i][j] = original\ image, b[i][j] = recostructed\ image.$

Structural Content: Structural content is calculated using following formula,

$$structural\ content = \frac{imga[i][j]}{imgb[i][j]} \qquad (19)$$

Where,

$$imga[i][j] = a[i][j] * a[i][j] \qquad (20)$$

$$imgb[i][j] = b[i][j] * b[i][j] \qquad (21)$$

Where, $a[i][j] = original\ image, b[i][j] = recostructed\ image.$

3.6 Normalized Absolute Error

Normalized absolute error is calculated using following formula,

$$normalised\ absolute\ error = \frac{totalerr}{totalimg}, \qquad (22)$$

totalerr = $error[i][j]$, $totalimg = a[i][j]$.

$$error[i][j] = a[i][j] - b[i][j] \tag{23}$$
$$error[i][j] = abs(error[i][j]) \tag{24}$$
We have to take only absolute value of $error[i][j]$.

3.7 Normalized Cross correlation (NCC)

Normalized cross correlation is a mathematical computation that fulfills an essential role in finding the similarity of images image processing.

Normalized Cross Correlation is calculated using following formula,

$$NCC = \frac{totalimgab}{totalimgaa} \tag{25}$$

Where,

$$totalimgab = (a[i][j] * [i][j]) \tag{26}$$
$$totalimgaa = (a[i][j] * a[i][j]) \tag{27}$$

IV. EXPERIMENTAL RESULTS AND DISCUSSIONS

Experiments are conducted for different weights and for various set of input image and file formats. The sample result is displayed in the table 5.1. The weight factor a3=b2=0.4 and a4=b5=0.6 that is (0.4, 0.6) are considered for the proposed aproach and the sample results are displayed in the table 5.1. Experiments are repeated for different weight factors such as (0.1, 0.9), (0.2, 0.8), (0.3, 0.7), (0.6, 0.4), (0.7, 0.3), (0.8, 0.2) and (0.9, 0.1) and the comparison is done using differnt parameters. The computation complexity in clock cycles and memory complexity is listed in the table 4 by porting the proposed algorithm, nearest neighbor, bilinear interpolation algorithm on TI DSP. A profiling result gives the compuation complexity and memory complexity.

The table 1 shows the proposed technique result for different weight factors. It is seen that the selection of weight factor depends on the pixel relationship of the input image. If the pixels are more correlated, the weight factor of (0.5, 0.5) gives better result. For the result displayed the correlation factor is 0.9966 which is almost equal to 1. The comparison of the proposed approach is done with the other interpolation methods such as nearest neighbor and bilinear interpolation. The table 5.2 shows the comparison result of the proposed approach with nearest neighbor and bilinear interpolation.

Table 1: Comparison of the proposed technique for different weight factors

Parameter	Proposed approach With weights (0.1 & 0.9)	Proposed approach With weights (0.2 & 0.8)	Proposed approach With weights (0.3 & 0.7)	Proposed approach With weights (0.4 & 0.6)
Memory(H)	3944	3944	3944	3944
computation time	0.018565	0.018565	0.018565	0.018565
MSE	94.9462	76.9701	64.1648	56.4991
PSNR	28.3560	29.2676	30.0578	30.6104
Quality index	0.8279	0.8498	0.8667	0.8743
Maximum difference	109	100	91	93
Average difference	-0.1449	-0.1344	-0.1448	-0.1341
Structural content	1.0014	1.0026	1.0031	1.0037
Normalized absolute error	0.0290	0.0264	0.0243	0.230
Normalized cross correlation	0.9966	0.9966	0.9966	0.9966

Table 2: Comparison of the proposed technique with nearest neighbor and bilinear interpolation methods

Parameter	Nearest neighbor interpolation	Bilinear interpolation	Proposed approach Weights (a3, a4) = (0.4,0.6)	Proposed approach weights (a3, a4) = (0.6,0.4)
Memory	1048	1200	3944	3944
computation time (sec)	0.0136287	0.856	0.018565	0.01856569
MSE	98.5860	10.4	22.8421	22.9372
PSNR	28.1927	37.97	34.5434	34.5254
Quality index	0.8724	0.999	0.9556	0.9544
Maximum difference	138	39	66	66
Average difference	-0.0640	-0.0294	-0.0324	-0.0353
Structural content	0.9986	1.0028	1.0052	1.0052
Normalized absolute error	0.0397	0.0171	0.0186	0.0187
Normalized cross corr	0.9967	0.9982	0.9965	0.9965

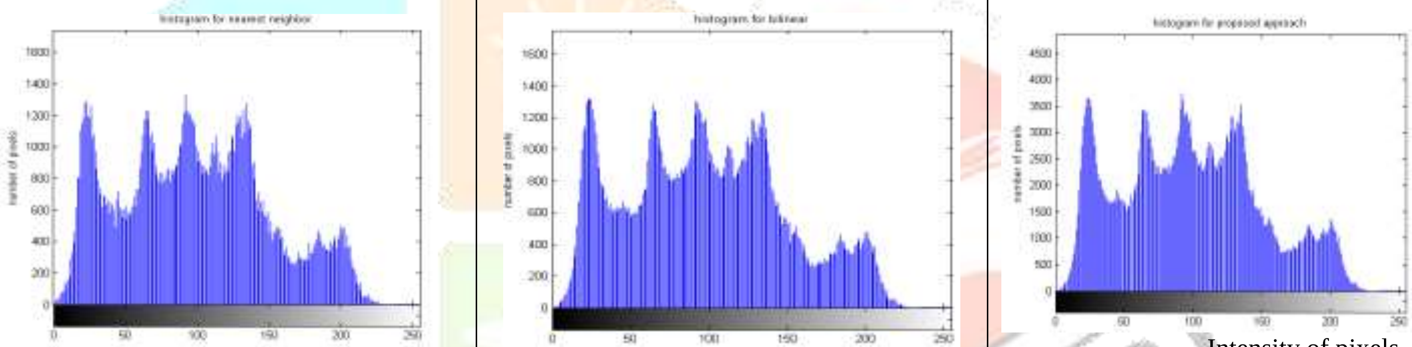
Table 3: Comparison of proposed technique with existing techniques for 2x2 images with scaling factor 1.5

Method	Substitution Operations	Multiplication operation	Addition operation
Nearest neighbor	9	0	0
Bilinear interpolation	1	32	24
Proposed technique	4	5	5

The table 2 and 3 indicates that the proposed technique is better than nearest neighbor in terms of PSNR and Quality index but consumes more computation time than the nearest neighbor technique. The proposed technique is better than the bilinear interpolation in terms of computation time which is illustrated in table 2. But the PSNR and quality index is less compared to bilinear interpolation. The proposed algorithm is a tradeoff between the quality and the computation complexity. Proposed approach consumes more memory since the algorithm is implemented using matrix multiplication method. The figure 2 shows the original image and image histogram.

Nearest neighbor interpolation | Bilinear interpolation | Proposed approach for interpolation

Figure 2: Image scaling result for different interpolation for a scaling factor of 1.5



Intensity of pixels | Intensity of pixels | Intensity of pixels

Nearest neighbor interpolation | Bilinear interpolation | Proposed approach

Figure 3: Image scaling histogram for different approaches for a scaling factor of 1.5

The image scaling result and its histogram using nearest neighbor technique for a scaling factor of 1.5 is shown in figure 2 and 3. The different interpolation techniques such as nearest neighbor, bilinear and proposed approach are ported on DSP under noisy conditions. The different type of noise considered is Gaussian, speckle, Poisson and salt & pepper. The table 4 shows the result for different scaling algorithms under the effect of noise. The table 4 shows that the proposed approach is better than the other interpolation techniques under the effect of noise. The proposed approach has higher value of quality index compared to nearest and bilinear interpolation techniques.

Table 4: Comparison of the scaling algorithms under the effect of noise

Scaling algorithms	Noise added	MSE	PSNR	Quality Index
Nearest	Gaussian	1321	16.9205	0.9378
	Speckle	1906	15.3296	0.9453
	Poisson	378.0865	22.3549	0.9020
	Salt & pepper	2159	14.7870	0.9646
Bilinear	Gaussian	541.0906	20.7981	0.9133
	Speckle	774.0885	19.2429	0.9080
	Poisson	158.6316	26.1269	0.9908
	Salt & pepper	863.4177	18.7686	0.9776
Proposed approach	Gaussian	1058	17.8839	0.9174
	Speckle	1533	16.2730	0.9363
	Poisson	254.6695	24.0710	0.9688
	Salt & pepper	1759	15.6761	0.9730

V. SUMMARY AND CONCLUSIONS

The experimental results shows that the proposed approach works better than the nearest neighbor interpolation in terms of quality of the image (PSNR) and better than the bilinear interpolation in terms of computational complexity (computation time). Also, the proposed approach works better under noisy conditions. The different noises such as Gaussian, speckle, Poisson and salt & pepper noises are considered to know the susceptibility of the proposed approach. The proposed approach is compared with nearest neighbor and bilinear interpolation methods.

The proposed approach can be modified to obtain the performance of the nearest neighbor interpolation and bilinear interpolation. The proposed approach can be used to scale-up the QCIF format or CIF format to any resolution such as QVGA, VGA, etc. The proposed approach can be used iteratively to convert from the given image/video format to required display resolution by padding with necessary number of zeros or duplication required number of rows and columns.

REFERENCES

1. Prasantha H.S, Shashidhara H.L and K.N.Balasubramanya Murthy, "Comparative analysis of different interpolation schemes in image processing", *International Conference on Advanced communication Systems (ICACS)*, India, pp.17–24, Jan 2007.
2. Prasantha H.S, Shashidhara H.L and K.N.Balasubramanya Murthy, "Image Scaling comparison using Universal Image Quality Index", *International Conference on Advances in computing, control and Telecommunication Technologies (ICACCTT)*, India, pp.859-863, Dec 2009.
3. Zhou Wang, et.al (2002), "A universal image quality Index", IEEE signal processing letters, Vol XX, No.Y, 2002.
4. Thomas M. Lehmann, *et. al.* (1999), "Survey: Interpolation Methods in Medical Image Processing", IEEE Transactions on medical imaging, VOL. 18, NO. 11.
5. Shi zaifeng, *et. al.* (2008), "A novel nonlinear scaling method for video images", International conference on computer science & software engineering", pp.357-360.
6. Shuai yuan, *et. al.* (2005), "High accuracy WADI image interpolation with local gradient features", Proceedings of 2005 International symposium on intelligent signal processing and communication system, Hongkong.
7. S. Battiato, *et. al.* (2002), "A locally adaptive zooming algorithm for digital images", Image and Vision Computing , 20 (2002), pp.805–812
8. Ran Gao, *et. al.* (2009), "Image zooming algorithm based on partial differential equations technique", International Journal of Numerical analysis and modeling, Volume 6, Number 2, pp.284-292.
9. Ryuji Matsuoka, *et. al.* (2008), "Comparison of Image Interpolation Methods Applied to Least Squares Matching", CIMCA 2008.
10. Gunasheela K S, H S Prasantha, "Satellite image compressiondetailed survey of the algorithms", Proceedings of ICCR in LNNS Springer, 2017, vol. 14, pp. 187-198.
11. H. S. Prasantha, H. L. Shashidhara, and K. N. Balasubramanya Murthy. Image compression using SVD. In Proceedings of the International Conference on Computational Intelligence and Multimedia Applications, pages 143–145. IEEE Computer Society, 2007.
12. Raghavendra.M.J, Prasantha.H.S and S.Sandya, "Image Compression Using Hybrid Combinations of DCT SVD and RLE", International Journal of Computer Techniques, Volume 2 Issue 5-2015.
13. Gunasheela K S, H S Prasantha, "Compressive sensing for image compression: survey of algorithms", Proceedings of Emerging Research in Computing, Information, Communication and Applications, ERCICA, Springer publication, Bengaluru, 2018
14. K N Shruthi, B M Shashank, Y. SaiKrishna Saketh, H.S Prasantha and S. Sandya, "Comparison Analysis Of A Biomedical Image For Compression Using Various Transform Coding Techniques", IEEE, pp. 297-303, 2016
15. Raghavendra.M.J, Prasantha.H.S and S.Sandya, "DCT SVD Based Hybrid Transform Coding for Image Compression", International Journal of Recent and Innovative Trends in computing and communication. 2015