Global Journal of Researches in Engineering
Electrical and Electronics Engineering
Volume 13 Issue 16 Version 1.0 Year 2013
Type: Double Blind Peer Reviewed International Research Journal
Publisher: Global Journals Inc. (USA)
Online ISSN: 2249-4596 & Print ISSN: 0975-5861

A Critical Performance Evaluation of Classification Methods with Modified JPEG Decompressed Multiband Images

By Ch. Ramesh, Dr. N.B. Venkateswarlu & Dr. J.V.R. Murthy

Aditya Institute of Technology and Management/JNTUK, India

Abstract- Effective utilization of bandwidth andstorage space is important in imaging applicationsincluding remote sensing. Remote sensingapplications use multi-sensory, multi-band, multiresolution images. Usually, remote sensingapplications uses image classification results fortheir analysis and decision making. In this paper wepropose a new JPEG based image compressionalgorithm based on filters. Proposed algorithmperformance is evaluated in relation to conventionalJPEG algorithm. In order to envisage the effect ofcompression on classification performance, MaximumLikelihood, Mahalanobis and Euclidean distanceclassifiers performance is evaluated with originalimage data, conventional JPEG compressed data andthe compressed image data with the proposedmethod. Experiments are carried out with manymulti-band images. Our experiments supports thatthe classification accuracies of compressed imagesare at par with original image data.

Keywords: joint photo experts group (jpeg), filters, maximum likelihood, mahalanobis, euclidean, confusion matrix, kappa coefficient.

GJRE-F Classification : FOR Code: 090699

Strictly as per the compliance and regulations of :

A Critical Performance Evaluation of Classification Methods with Modified JPEG Decompressed Multiband Images

Ch. Ramesh [α], Dr. N.B. Venkateswarlu [σ] & Dr. J.V.R. Murthy [ρ]

Abstract- Effective utilization of bandwidth andstorage space is important in imaging applicationsincluding remote sensing. Remote sensingapplications use multi-sensory, multi-band, multiresolution images. Usually, remote sensingapplications uses image classification results fortheir analysis and decision making. In this paper wepropose a new JPEG based image compressionalgorithm based on filters. Proposed algorithmperformance is evaluated in relation to conventionalJPEG algorithm. In order to envisage the effect ofcompression on classification performance, MaximumLikelihood, Mahalanobis and Euclidean distanceclassifiers performance is evaluated with originalimage data, conventional JPEG compressed data andthe compressed image data with the proposedmethod. Experiments are carried out with manymulti-band images. Our experiments supports thatthe classification accuracies of compressed imagesare at par with original image data.

Keywords: joint photo experts group (jpeg), filters, maximum likelihood, mahalanobis, euclidean, confusion matrix, kappa coefficient.

I. Introduction

In the recent years use of remote sensingsatellite data for urban planning, military,weather forecast, robotics, automatednavigation system, remote surveillance hasincreased by many fold in addition toconventional applications such as naturalresources management.

These applications involve acquisition, communication, storage and processing ofterrible number of images of earth surface.This situation is becoming more aggravatedbecause of increased pixel resolution, graylevel resolution, band resolutions and reducedrepetition cycle of satellite. All of these development demands more band width fordownlink lines of satellite in addition to moredisk space for storage.

In communications, data compression techniques under the name hood of imagecoding are widely used to reduce thecommunication bandwidth bottlenecks duringdata communication. For instance, JPEGstandard is used for still image compression [1], MPEG is used for video compression [2]. Also, while communicating data from satellites toground stations some compression methods areused [3].

Authors α σ ρ: Professor, Dept. of CSE, AITAM, Tekkali, A.P, India. e-mails: chappa_ramesh01@yahoo.co.in, venkat_ritch@yahoo.com, mjonnalagedda@gmail.com

A typical image processing system isas shown in Figure 1 that is commonlyemployed for remote sensing applications. It isvery common that most of the applicationscientists using original image data for theirprocessing. In majority of remote sensing applications, results of classification are theultimate interest [4].

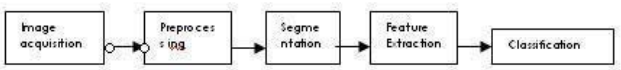

Figure 1 : A Topical Image Processing System

In this study, we propose to study howthe classification results will vary if we usecompressed image data instead of originalimage data. Usually applications such as landuse classifications assumes samples of a groupwill be having small random variations in theirpixel values while samples of different groupsto be having contrastingly different pixelvalues. Because of the increased pixel and graylevel resolutions, samples of a group may behaving similar pixel values. Moreover, they willbe having high level of spatial auto correlation.Evidently, majority of compression methods exploits this auto correlation to achieve highcompression ratios with acceptable PSNR (PeakSignal to Noise Ratio) values [5].

Our proposed Algorithm is based onfiltering concept [6]. In this Algorithm insteadof sending the original image, we send thefiltered image. In general, the number ofuseful DCT coefficients will become more if 8x8image blocks contain lot of variations in values,otherwise only few DCT coefficients will bemeaningful. If we apply filtering on image itgets smoothened, that is variation of the pixelvalues of a block reduces. It is attractive forpoint of view of compression as the number ofmeaningful DCT coefficients are going toreduce. Thus we are achieving compression benefit. We have compared the compression performance of our algorithm with conventional JPEG algorithm with variety ofmulti band images.

Also in this study, we evaluate theclassification performance of popularclassification algorithms like Maximum Likelihood, Mahalanobis and Euclidian distanceby taking original image data, conventional

JPEG compressed image data, compressed datathat is compressed by filter based JPEG imagecompression method.

Our paper work is organized as follows.Section II introduces the standard JPEGalgorithm. Section III explains proposedcompression algorithm. Section IV illustratesthe selected classification algorithms. Section Vincludes details about our experimentationsand results. Section VI contains conclusionsabout our research work

II. Brief Overview of JPEG Encoding/ Decoding System

JPEG is a well known standardized image compression technique. JPEG loses informationso the decompressed picture is not the same asthe original one. The main reason for use ofJPEG is to reduce the size of image files.Reducing image files is an important procedurefor transmitting files across networks orarchiving libraries. Usually JPEG can removethe less important data before thecompression; hence JPEG is able to compressimages meaningfully, which produces a hugedifference in the transmission time and the diskspace. Figure 2 shows the basic Architecture ofJPEG compression system. Here is a briefoverview of the JPEG compression system. [5]

The image is first subdivided into pixelblocks of size 8X8, which are processed left toright, top to bottom. As each 8X8 block or subimage is encountered, its 64 pixels are levelshifted by subtracting the quantity L/2, whereL is the Gray level resolution of the image . The2-D Forward Discrete Cosine Transform (FDCT)(Eq-1)[5]of the block is then computed,quantized using 64 corresponding step sizevalues from the quantization table in Figure 3[7]. After quantization the DCT coefficientsare rearranged in a zigzag sequence order as shown in the Figure 4. [7]

Since the one-dimensional reorderedarray generated under the zigzag pattern ofFigure 3 is qualitatively arranged according toincreasing spatial frequency, the JPEG codingprocedure is designed to take the advantage ofthe long runs of zeros that normally result fromthe reordering. In particular, the nonzero ACcoefficients (the term AC denotes all transformcoefficients with the exception of the zeroth orDC coefficient) are coded using a variablelengthcode that defines the coefficient's valueand number of preceding zeros. The DCcoefficient is difference coded relative to theDC coefficient of the previous sub image.

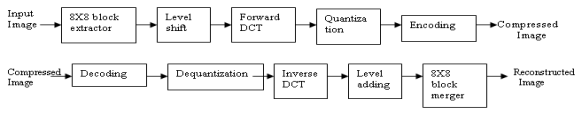

Figure 2 : Basic Architecture of JPEG Compression

The 2-D DCT is

$$C(u,v) = \alpha(u)\alpha(v)\sum_{x=0}^{N-1}\sum_{y=0}^{N-1} f(x,y)\cos\left[\frac{(2x+1)u\pi}{2N}\right]\cos\left[\frac{(2y+1)v\pi}{2N}\right] \quad (1)$$

for u, v = 0, 1, 2, ... , N -1

$$\alpha(u) = \begin{cases} \sqrt{1/N} & for \quad u = 0 \\ \sqrt{2/N} & for \quad u > 0 \end{cases} \quad (2)$$

$$\alpha(v) = \begin{cases} \sqrt{1/N} & for \quad v = 0 \\ \sqrt{2/N} & for \quad v > 0 \end{cases} \quad (3)$$

16	11	10	16	24	40	51	61
12	12	14	19	26	58	60	55
14	13	16	24	40	57	69	56
14	17	22	29	51	87	80	62
18	22	37	56	68	109	103	77
24	35	55	64	81	104	113	92
49	64	78	87	103	121	120	101
72	92	95	98	112	100	103	99

Figure 3 : Quantization Matrix [7]

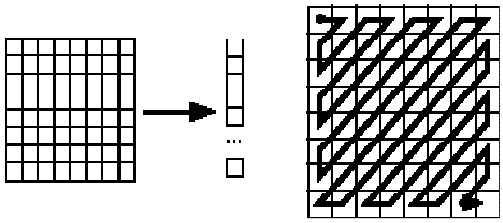

Figure 4 : Zigzag Séquencé [7]

The decompression process performs aninverse procedure. It decodes the Huffmancodes. Then, it makes the inversion of theQuantization step. In this stage, the decoderraises the small numbers by multiplying themby the quantization coefficients. The resultsare not accurate, but they are close to theoriginal numbers of the DCT coefficients. AnInverse Discrete Cosine Transform (IDCT) (Eq.4)[7] is performed on the data received from theprevious step. Finally add L/2 to each subimage. Place the sub images in their correctpositions.

$$\hat{f}(x,y) = \sum_{u=0}^{N-1}\sum_{v=0}^{N-1}\alpha(u)\alpha(v)C(u,v)\cos\left[\frac{(2x+1)u\pi}{2N}\right]\cos\left[\frac{(2y+1)v\pi}{2N}\right] \quad (4)$$

The error between the original imageand reconstructed image is calculated in termsof Peak signal to noise ratio (PSNR) = 10 log10

(L2/MSE) (5)

$$MSE = \frac{\sum_{x=0}^{m-1}\sum_{y=0}^{n-1}\left[\hat{f}(x,y) - f(x,y)\right]^2}{mXn} \quad (6)$$

MSE – Mean Squared Error

$\hat{f}(x, y)$ - Reconstructed Image

f (x, y) – Original Image

m x n – Size of the Image

III. New Mean, Median & Outlier based JPEG Algorithms

Mean filtering [8] is a simple, intuitiveand easy to implement method of imagesmoothing i.e. reducing the amount of variationbetween one pixel and the next or surroundingpixels. It is often used to reduce noise inimage. The idea of mean filtering is simply toreplace each pixel in an image with the meanvalue of its neighbors including itself. This hasthe effect of eliminating pixel values which areunrepresentative of their surroundings. Usually,3x3 neighborhoods of pixels are consideredwhile calculating mean filtered value of anypixel.

Median filter [9] is normally used to reducenoise in an image like the mean filter.However, it often does a better job than themean filter in preserving useful detail in theimage. Like the mean filter, the median filterconsiders each pixel in the image in turn andlooks at its neighbors to decide whether or notits representative of its surroundings. Insteadof simply replacing the pixel value with themean of neighboring pixel values, it replaces itwith the median of those values.

An outlier [10] is an observation that isnumerically distant from the rest of the data.In an image, a pixel value is very different fromits surrounding pixels, it can be called asoutlier.

From basic statistics, we know that apopulations sample values with someconfidence level can be given as mean ± C*P, where C is weighing factor (critical value) and P is standard deviation of the population. Table-1 shows the commonly used ConfidenceLevels and Corresponding Critical Values [11].In our outlier based algorithms, we take thesesimple confidence limits of normal distributionin deciding whether a pixel is outlier or not. Ifthe pixel is observed to be outlier with thegiven confidence level, we may retain else wemay take its mean filtered or median filtered value.

Table 1 : Confidence Levels Vs Critical values

Confidence Level	80%	90%	95%	98%	99%	99.8%	99.9%
Critical Values	1.28	1.645	1.96	2.33	2.58	3.08	3.27

In the following, we have listed thebasic Mean, Median and Outlier algorithms

a) MeanDCT Algorithm

- Apply mean filtering to the originalimage using 3x3 window
- Apply DCT on the mean filteredimages

b) MedianDCT Algorithm

- Apply median filtering to the original imageusing 3x3 window
- Apply DCT on the median filtered imagesOutlier

c) MeanDCT Algorithm

- Apply mean filtering with a little variation tothe given original image using 3x3 window. Foreach pixel, calculate average and standarddeviation of its neighboring 3x3 pixels. If apixels value is observed to be outlier (not inthe range of Mean ± C*P) then its value is takenas itself else mean is taken as its filtered value.
- Apply DCT on the outliermean filteredimage.

d) OutlierMedianDCT Algorithm

- Apply median filtering with a little variationto the given original image using 3x3 window.For each pixel, calculate median and standarddeviation of its neighboring 3x3 pixels. If apixels value is observed to be outlier (not inthe range of Median ± C*P) then its value istaken as itself else median is taken as itsfiltered value.
- Apply DCT on the outliermedian filteredimage.

IV. Popular Classification Algorithms

a) Maximum Likelihood Classifier

Let $w_1, w_2, \ldots, w_m$denote m distinctpopulations (classes) with known d-dimensionalprobability density functions $p_1(X)$, $p_2(X)$, ... $p_m(X)$, respectively. The a priori probabilitiesthat an observation is selected frompopulations $w_1, w_2, \ldots, w_m$ are denoted by $q_l, q_2, \ldots, q_m$, respectively[12]. According tothe Bayesian ML classification rule, assumingequal costs for misclassifications, a random ddimensional pixel vector X is classified as classw_k

$$q_k p_k(X) = max\{q_i P_i(X)\} \text{ for } i = 1, 2 \ldots, m. \quad (7)$$

Assuming equal a priori probabilities for all theclasses, decision rule (7) becomes:

$$p_k(X) = \max\{p_i(X)\},\ i = 1, 2, \ldots, m \quad (8)$$

In Equations (7) and (8), the probability density$p_k(X)$ will be given as:

$$p_k(X) = \frac{1}{(2\pi)^{d/2}\left|\sum_k\right|^{1/2}} \quad (9)$$

$$X \exp[-1/2.(X - M_k)^T \sum_k^{-1} (X - M_k)].$$

Here, and, are the mean vector andcovariance matrices of the k^{th}class, and arecalculated from the training data. is asymmetric positive definite matrix. -1,k kare the inverse and determinant of the covariance matrix.

b) Mahalanobis Classifier

According to this classifier a d-dimensionalrandom pixel vector (X) will be assigned to thegroup to which it is nearest [13]. Each group ischaracterized by its mean vector, which iscalculated from training data. Nearness isdetermined by the Mahalanobis distancebetween the group mean and X. Inmathematical terms, the same classificationrule can be represented as:

$$X \in w_i \tag{10}$$

where i = (1, 2, . . . C) groups if $d_i(X) < d_j(X)$ forall j U i where

$$d_i(X) = (X-M_i)^T \Sigma^{-1}(X-M_i \tag{11}$$

and M_i s mean vector of i^{th}group indicatesvector should be transposed. -1 is the inverseof the pooled covariance matrix.

c) Euclidian Distance Classifier

According to this classifier a random ddimensionalpixel vector (X) will be assigned tothe group to which it is nearest [14]. Eachgroup is characterized by its mean vector,which is calculated from training data.Nearness is determined by the Euclideandistance between the group mean and X. Inmathematical terms, the same classificationrule can be represented as:

$$X \in w_i \tag{12}$$

Where i = (1, 2, . . . C) groups if $d_i(X) < d_j(X)$ forall j U i where

$$d_i(X) = (X-M_i)^T(X-M_i \tag{13}$$

and M_i is mean vector of i^{th} group. T indicatesvector should be transposed

V. Experimentations and Results

For the purpose of experimental work,Landsat TM data from USGS data base"www.usgs.gov" is used. Experiments arecarried out under MS Windows XP version 2002,SP3 edition. The experimental system isequipped with Intel core 2 Duo 2.60 GHzprocessor with 1 GB RAM. Using ERDAS Imagine8.6 (copy rightsÓ1991-2002, Lieca Geo systems)Training sites are labeled. Programs are writtenin C language under Microsoft Visual Studio2005 version 8.0.

We have carried out extensivesimulations with the selected images andproposed algorithms. Table 2 shows theCompression Benefit and PSNR values ofMeanDCT algorithm VsOutlierMeanDCTalgorithm. With all the images we found thatMeanDCT and OutlierMeanDCT algorithms havebetter compression ratios as compared toconventional JPEG coding. The PSNR loss inMeanDCT and OutlierMeanDCT algorithms isnegligible as compared to conventional JPEGcoding. While comparing MeanDCT and thecorresponding Outlier DCT, CompressionBenefits are observed to beMeanDCT>OutlierMeanDCT (for C=1.28 to 2.58).As the value of C increases in the Outlier,Compression Benefit increases. For C=3.08 to3.27 Compression Benefit in MeanDCT andOutlierMeanDCT is almost same. PSNR in MeanDCT<OutlierMeanDCT(for C=1.28 to 2.58).As the value of C decreases in the Outlier, PSNRincreases. For C=3.08 to 3.27 PSNR in MeanDCTand OutlierMeanDCT is almost same. Fig5 Showsthe sample (Owensvalley) Original image, JPEGcompressed image, Proposed compresse dimage (Mean filtered approach).

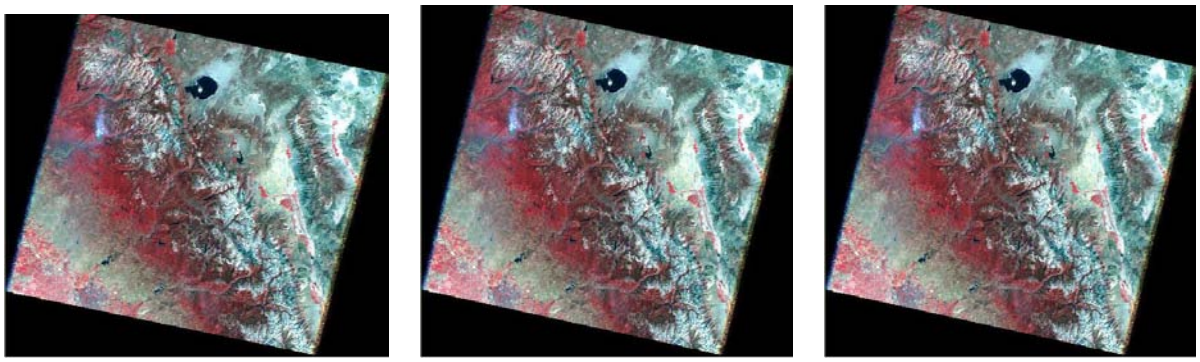

Figure 5 : Original Image JPEG Compressed Image Proposed Compressed Image (Mean filtered approach)

Table 2 : Compression benefit and PSNR loss of Mean & Outlier Mean DCT algorithm compared toconventional JPEG

Image		Conventional	Mean DCT	Outlier Mean DCT (C=3.27)	Outlier Mean DCT (C=3.08)	Outlier Mean DCT (C=2.58)	Outlier Mean DCT (C=2.33)	Outlier Mean DCT (C=1.96)	Outlier Mean DCT (C=1.645)	Outlier Mean DCT (C=1.28)
Bolivia1	No. of Bits	215085	172156	172144	172141	175585	178003	183439	19046	198835
	% of Saving	-	19.959	19.964	19.966	18.364	17.240	14.713	11.447	7.555
	PSNR	39.785	38.073	38.073	38.074	38.074	38.083	38.101	38.164	38.215
	% of Loss	-	4.303	4.303	4.300	4.300	4.277	4.232	4.074	3.946
Bolivia2	No. of Bits	275655	208384	208383	208394	212633	216319	225195	236059	250482
	% of Saving	-	24.404	24.404	24.400	22.862	21.525	18.305	14.364	9.132
	PSNR	37.939	36.872	36.872	36.872	36.88	36.89	36.914	36.971	37.064
	% of Loss	-	2.812	2.812	2.812	2.791	2.764	2.701	2.551	2.306
Bolivia3	No. of Bits	394036	274432	274432	274430	283304	291242	307235	325783	351372
	% of Saving	-	30.353	30.353	30.354	28.102	26.087	22.028	17.321	10.827
	PSNR	34.539	33.098	33.098	33.098	33.113	33.128	33.185	33.272	33.455
	% of Loss	-	4.172	4.172	4.172	4.128	4.085	3.920	3.668	3.138
Bolivia4	No. of Bits	623322	393045	393045	393045	398738	411789	442420	483673	526620
	% of Saving	-	36.943	36.943	36.943	36.030	33.936	29.022	22.403	15.513
	PSNR	30.959	29.219	29.219	29.219	29.225	29.247	29.283	29.373	29.652
	% of Loss	-	5.620	5.620	5.620	5.600	5.529	5.413	5.122	4.221
Bolivia5	No. of Bits	694304	429601	429601	429601	437597	451664	483706	525250	584315
	% of Saving	-	38.124	38.124	38.124	36.973	34.947	30.332	24.348	15.841
	PSNR	29.986	28.004	28.004	28.004	28.009	28.023	28.066	28.17	28.489
	% of Loss	-	6.609	6.609	6.609	6.593	6.546	6.402	6.056	4.992
Bolivia6	No. of Bits	321611	242123	242129	242127	243817	246467	252569	262095	277678
	% of Saving	-	24.715	24.713	24.714	24.188	23.364	21.467	18.505	13.660
	PSNR	37.221	33.598	33.598	33.598	33.598	33.597	33.600	33.612	33.654
	% of Loss	-	9.773	9.773	9.773	9.773	9.736	9.728	9.696	9.583

Bolivia7	No. of Bits	559107	357508	357508	357512	372414	386369	412008	441576	482908
	% of Saving	-	36.057	36.057	36.056	33.391	30.895	26.309	21.021	13.628
	PSNR	31.4	29.516	29.516	29.516	29.535	29.556	29.613	29.727	29.998
	% of Loss	-	6	6	6	5.939	5.872	5.691	5.328	4.464
Total	No. of Bits	3083120	2077249	2077242	2077250	2124088	2181853	2306572	2464900	2682210
	% of Saving	-	32.625	32.625	32.625	31.105	29.232	25.187	20.051	13.003
	PSNR	34.547	32.625	32.625	32.625	32.633	32.646	32.680	32.755	32.932
	% of Loss	-	5.563	5.563	5.563	5.540	5.502	5.404	5.187	4.674
Monolake1	No. of Bits	410486	329732	329729	329722	330006	331107	336165	344531	361041
	% of Saving	-	19.672	19.673	19.675	19.606	19.337	18.105	16.067	12.045
	PSNR	39.871	37.503	37.503	37.503	37.503	37.509	37.537	37.594	37.737
	% of Loss	-	5.939	5.939	5.939	5.939	5.924	5.853	5.710	5.352
Monolake2	No. of Bits	306006	257407	257409	257441	257713	258682	261126	266277	276333
	% of Saving	-	15.881	15.881	15.870	15.781	15.465	14.666	12.983	9.696
	PSNR	42.237	40.988	40.988	40.988	40.988	40.994	41.014	41.064	41.178
	% of Loss	-	2.957	2.957	2.957	2.957	2.942	2.895	2.777	2.507
Monolake3	No. of Bits	399635	324167	324170	324158	324460	325220	328846	335993	350594
	% of Saving	-	18.884	18.883	18.886	18.810	18.620	17.713	15.925	12.271
	PSNR	40.435	38.850	38.85	38.849	38.849	38.859	38.883	38.952	39.113
	% of Loss	-	3.919	3.919	3.922	3.922	3.897	3.838	3.667	3.269
Monolake4	No. of Bits	366063	300168	300130	300212	300264	300614	303531	309183	322696
	% of Saving	-	18.000	18.011	17.988	17.974	17.879	17.082	15.538	11.846
	PSNR	41.097	39.268	39.268	39.268	39.268	39.274	39.295	39.349	39.496
	% of Loss	-	4.450	4.450	4.450	4.450	4.435	4.384	4.253	3.895
Monolake5	No. of Bits	578965	451039	451032	451010	451400	452566	457120	468155	492798
	% of Saving	-	22.095	22.096	22.100	22.033	21.831	21.045	19.139	14.882
	PSNR	37.724	34.999	34.999	34.999	34.999	35.003	35.030	35.115	35.36
	% of Loss	-	7.223	7.223	7.223	7.223	7.212	7.141	6.916	6.266

Monolake6	No. of Bits	249582	232905	232911	232913	232925	233052	234078	234701	235876
	% of Saving	-	6.681	6.679	6.678	6.673	6.623	6.211	5.962	5.491
	PSNR	46.608	37.557	37.557	37.557	37.557	37.557	37.558	37.559	37.56
	% of Loss	-	19.419	19.419	19.419	19.419	19.419	19.417	19.415	19.412
Monolake7	No. of Bits	439079	352864	352873	352899	353207	354337	358195	365769	381841
	% of Saving	-	19.635	19.633	19.622	19.557	19.299	18.421	16.696	13.035
	PSNR	39.828	38.134	38.134	38.134	38.134	38.139	38.162	38.235	38.418
	% of Loss	-	4.253	4.253	4.253	4.253	4.240	4.182	3.999	3.540
Total	No. of Bits	2749816	2248277	2248254	2248355	2250035	2255578	2279061	2324609	2421179
	% of Saving	-	18.239	18.239	18.236	18.175	17.973	17.119	15.463	11.951
	PSNR	41.114	38.186	38.185	38.185	38.185	38.190	38.211	38.266	38.408
	% of Loss	-	7.121	7.124	7.124	7.124	7.111	7.060	6.927	6.581
Owensvalley1	No. of Bits	537518	360478	360483	360486	367990	378051	399052	423410	460890
	% of Saving	-	32.936	32.935	32.935	31.539	29.667	25.760	21.228	14.255
	PSNR	32.756	30.116	30.116	30.116	30.127	30.161	30.224	30.299	30.575
	% of Loss	-	8.059	8.059	8.059	8.026	7.922	7.729	7.500	6.698
Owensvalley2	No. of Bits	409426	288904	288897	288887	293676	299851	313895	331761	358355
	% of Saving	-	29.436	29.438	29.440	28.271	26.763	23.332	18.969	12.473
	PSNR	35.207	33.626	33.626	33.626	33.638	33.649	33.685	33.749	33.939
	% of Loss	-	4.490	4.490	4.490	4.456	4.425	4.323	4.141	3.601
Owensvalley3	No. of Bits	558702	371381	371378	371376	377700	388065	410282	438638	479219
	% of Saving	-	33.527	33.528	33.528	32.396	30.541	26.565	21.489	14.226
	PSNR	32.590	30.696	30.696	30.696	30.713	30.737	30.785	30.871	31.12
	% of Loss	-	5.811	5.811	5.811	5.759	5.685	5.538	5.274	4.510
Owensvalley4	No. of Bits	565589	374161	374168	374159	379834	390527	413718	443804	485971
	% of Saving	-	33.845	33.844	33.846	32.842	30.952	26.851	21.532	14.077
	PSNR	32.645	30.63	30.63	30.63	30.65	30.682	30.729	30.812	31.035
	% of Loss	-	6.172	6.172	6.172	6.111	6.013	5.869	5.614	4.931

Owensvalley5	No. of Bits	991131	583964	583964	583964	590799	606683	650116	708184	801412
	% of Saving	-	41.081	41.081	41.081	40.391	38.788	34.406	28.547	19.141
	PSNR	28.087	24.942	24.942	24.942	24.949	24.969	25.034	25.178	25.613
	% of Loss	-	11.197	11.197	11.197	11.172	11.101	10.869	10.357	8.808
Owensvalley6	No. of Bits	478658	346282	346277	346290	346905	348258	354345	370218	403209
	% of Saving	-	27.655	27.656	27.653	27.525	27.242	25.971	22.655	15.762
	PSNR	35.711	29.586	29.586	29.586	29.593	29.595	29.620	29.825	30.554
	% of Loss	-	17.151	17.151	17.151	17.131	17.126	17.056	16.482	14.440
Owensvalley7	No. of Bits	708596	449957	449957	449957	458283	470221	498865	536339	595087
	% of Saving	-	36.500	36.500	36.500	35.325	33.640	29.598	24.309	16.018
	PSNR	30.718	28.584	28.584	28.584	28.62	28.655	28.703	28.801	29.095
	% of Loss	-	6.947	6.947	6.947	6.829	6.715	6.559	6.240	5.283
Total	No. of Bits	4249620	2777127	2775122	2775119	2815187	2881656	3040273	3252354	3584143
	% of Saving	-	34.649	34.649	34.697	33.754	32.190	28.457	23.467	15.659
	PSNR	32.530	29.740	29.740	29.740	29.755	29.778	29.826	29.934	29.974
	% of Loss	-	8.576	8.576	8.576	8.530	8.459	8.312	7.980	7.857

Table 3 shows the Compression Benefit and PSNR values of Median DCT algorithm Vs Outlier Median DCT algorithm. With all the images we found that Median DCT and Outlier Median DCTalgorithms have better compression ratios as compared to conventional JPEG coding. The PSNR loss in Median DCT and OutlierMedianDCT algorithms is very less as compared to conventional JPEG coding. While comparing MedianDCT and the corresponding Outlier DCT, Compression Benefits are observed tobe MedianDCT>OutlierMedianDCT(for C=1.28 to 2.58). As the value of C increases in the Outlier,Compression Benefit increases. For C=3.08 to 3.27 Compression Benefit in MedianDCT andOutlierMedianDCT is same. PSNR in MedianDCT<OutlierMedianDCT(for C=1.28 to 2.58). As the value ofC decreases in the Outlier, PSNR increases. For C=3.08 to 3.27 PSNR in MedianDCT and OutlierMedianDCT is almost same.

Table 3 : Compression benefit and PSNR loss of Median & Outlier Median DCT algorithm compared toconventional JPEG

Bolivia1	No. of Bits	215085	178234	178240	178242	181353	183473	188038	194726	204035
	% of Saving	-	17.133	17.130	17.129	15.683	14.697	12.575	9.465	5.137
	PSNR	39.785	38.432	38.432	38.432	38.488	38.53	38.624	38.886	39.219
	% of Loss	-	3.400	3.400	3.400	3.260	3.154	2.918	2.259	1.422

Bolivia2	No. of Bits	275655	216504	216510	216502	220253	223670	232032	242533	257069
	% of Saving	-	21.458	21.456	21.459	20.098	18.858	15.825	12.015	6.742
	PSNR	37.939	36.860	36.860	36.860	36.914	36.963	37.094	37.307	37.562
	% of Loss	-	2.844	2.844	2.844	2.701	2.572	2.227	1.665	0.993
Bolivia3	No. of Bits	394036	283023	283023	283030	292258	300355	316259	334144	359377
	% of Saving	-	28.173	28.173	28.171	25.829	23.774	19.738	15.199	8.795
	PSNR	34.539	32.819	32.819	32.819	32.889	32.961	33.157	33.418	33.864
	% of Loss	-	4.979	4.979	4.979	4.777	4.568	4.001	3.245	1.954
Bolivia4	No. of Bits	623322	423592	423592	423592	428393	439439	467525	505592	556812
	% of Saving	-	32.042	32.042	32.042	31.272	29.500	24.994	18.887	10.670
	PSNR	30.959	29.228	29.228	29.228	29.247	29.298	29.412	29.635	30.093
	% of Loss	-	5.591	5.591	5.591	5.529	5.365	4.996	4.276	2.797
Bolivia5	No. of Bits	694304	469136	469136	469136	477077	490552	520174	558956	614893
	% of Saving	-	32.430	32.430	32.430	31.287	29.346	25.079	19.494	11.437
	PSNR	29.986	27.936	27.936	27.936	27.964	28.016	28.139	28.382	28.900
	% of Loss	-	6.836	6.836	6.836	6.743	6.569	6.159	5.349	3.621
Bolivia6	No. of Bits	321611	266018	266004	266014	267316	269347	274662	283086	297933
	% of Saving	-	17.285	17.290	17.287	16.882	16.250	14.598	11.978	7.362
	PSNR	37.221	35.806	35.806	35.806	35.818	35.833	35.868	35.934	36.136
	% of Loss	-	3.801	3.801	3.801	3.769	3.729	3.635	3.457	2.915
Bolivia7	No. of Bits	559107	375504	375504	375497	391368	405654	430574	460136	500472
	% of Saving	-	32.838	32.838	32.839	30.001	27.446	22.988	17.701	10.487
	PSNR	31.400	29.245	29.245	29.245	29.321	29.399	29.573	29.855	30.386
	% of Loss	-	6.863	6.863	6.863	6.621	6.372	5.818	4.920	3.229
Total	No. of Bits	3083120	2212011	2212009	2212013	2258018	2312490	2429264	2579173	2790591
	% of Saving	-	28.254	28.254	28.254	26.761	24.995	21.207	16.345	9.488
	PSNR	34.547	32.903	32.903	32.903	32.948	33	33.123	33.345	33.737
	% of Loss	-	4.758	4.758	4.758	4.628	4.477	4.121	3.479	2.344

Monolake1	No. of Bits	410486	350758	350767	350766	350935	351841	356379	364857	380685
	% of Saving	-	14.55	14.548	14.548	14.507	14.286	13.181	11.115	7.259
	PSNR	39.871	39.084	39.084	39.084	39.086	39.099	39.149	39.251	39.461
	% of Loss	-	1.973	1.973	1.973	1.968	1.936	1.810	1.555	1.028
Monolake2	No. of Bits	306006	270223	270260	270298	270633	271462	274155	274958	288663
	% of Saving	-	11.693	11.681	11.669	11.559	11.288	10.408	10.146	5.667
	PSNR	42.237	41.791	41.791	41.791	41.794	41.805	41.835	41.895	42.02
	% of Loss	-	1.055	1.055	1.055	1.048	1.022	0.951	0.809	0.513
Monolake3	No. of Bits	399635	343728	343749	343743	344127	345205	348596	355357	369757
	% of Saving	-	13.989	13.984	13.985	13.889	37.794	12.771	11.079	7.476
	PSNR	40.435	39.742	39.742	39.742	39.745	39.760	39.797	39.881	40.075
	% of Loss	-	1.713	1.713	1.713	1.706	1.669	1.577	1.370	0.890
Monolake4	No. of Bits	366063	318485	318461	318468	318653	319234	322400	328482	340987
	% of Saving	-	12.997	13.003	13.001	12.951	12.792	11.927	10.266	6.850
	PSNR	41.097	40.428	40.429	40.428	40.432	40.441	40.477	40.571	40.76
	% of Loss	-	1.627	1.625	1.627	1.618	1.596	1.508	1.279	0.820
Monolake5	No. of Bits	578965	483941	483935	483932	484197	485070	489545	501222	524503
	% of Saving	-	16.412	16.413	16.414	16.368	16.217	15.444	13.427	9.406
	PSNR	37.724	36.431	36.431	36.431	36.432	36.439	36.488	36.626	36.957
	% of Loss	-	3.427	3.427	3.427	3.424	3.4063	3.276	2.910	2.033
Monolake6	No. of Bits	249582	245642	245642	245642	245647	245771	246794	247364	248020
	% of Saving	-	1.578	1.578	1.578	1.578	1.526	1.117	0.8888	0.625
	PSNR	46.608	46.536	46.536	46.536	46.536	46.536	46.537	46.54	46.542
	% of Loss	-	0.154	0.154	0.154	0.154	0.154	0.152	0.145	0.141
Monolake7	No. of Bits	439079	376122	376122	376134	376452	377326	381171	388516	403764
	% of Saving	-	14.338	14.338	14.335	14.263	14.064	13.188	11.515	8.042
	PSNR	39.828	39.074	39.074	39.074	39.074	39.080	39.114	39.207	39.407
	% of Loss	-	1.893	1.893	1.893	1.893	1.878	1.792	1.559	1.057

Total	No. of Bits	2749816	2388899	2388936	2388983	2390644	2395909	2419040	2464756	2556379
	% of Saving	-	13.125	13.123	13.122	13.122	12.870	12.029	10.366	7.034
	PSNR	41.114	40.44	40.441	40.44	40.442	40.451	40.485	40.567	40.746
	% of Loss	-	1.639	1.636	1.639	1.634	1.612	1.529	1.330	0.895
Owensvalley1	No. of Bits	537518	392626	392626	392622	399523	409238	428771	451032	485562
	% of Saving	-	26.955	26.955	26.956	25.672	23.865	20.231	16.089	9.665
	PSNR	32.756	30.489	30.489	30.489	30.540	30.630	30.782	30.962	31.469
	% of Loss	-	6.920	6.920	6.920	6.765	6.490	6.026	5.476	3.929
Owensvalley2	No. of Bits	409426	310144	310145	310145	314136	319462	332498	348448	373454
	% of Saving	-	24.249	24.248	24.248	23.274	21.973	18.789	14.893	8.785
	PSNR	35.207	33.821	33.821	33.821	33.862	33.912	34.016	34.171	34.497
	% of Loss	-	3.936	3.936	3.936	3.820	3.678	3.382	2.942	2.016
Owensvalley3	No. of Bits	558702	403360	403360	403357	408933	417707	437272	463259	502047
	% of Saving	-	27.804	27.804	27.804	26.806	25.236	21.734	17.082	10.140
	PSNR	32.590	30.813	30.813	30.813	30.865	30.929	31.049	31.239	31.652
	% of Loss	-	5.452	5.452	5.452	5.293	5.096	4.728	4.145	2.878
Owensvalley4	No. of Bits	565589	405223	405223	405218	410226	420224	441258	468824	509673
	% of Saving	-	28.353	28.353	28.354	27.469	25.701	21.982	17.108	9.886
	PSNR	32.645	30.842	30.842	30.842	30.892	30.965	31.08	31.278	31.688
	% of Loss	-	5.523	5.523	5.523	5.369	5.146	4.793	4.187	2.931
Owensvalley5	No. of Bits	991131	650771	650771	650771	657617	672431	712566	768429	860138
	% of Saving	-	34.340	34.340	34.340	33.649	32.155	28.105	22.469	13.216
	PSNR	28.087	25.070	25.070	25.070	25.091	25.139	25.284	25.562	26.233
	% of Loss	-	10.741	10.741	10.741	10.666	10.495	9.979	8.989	6.600
Owensvalley6	No. of Bits	478658	377688	377690	377697	378455	379666	385816	404543	435834
	% of Saving	-	21.094	21.093	21.092	20.934	20.681	19.396	15.483	8.946
	PSNR	35.711	29.467	29.467	29.467	29.478	29.482	29.527	29.996	31.897
	% of Loss	-	17.484	17.484	17.484	17.454	17.442	17.316	16.003	10.680

Owensvalley7	No. of Bits	708596	492900	492901	492901	501063	512030	537843	571450	626345
	% of Saving	-	30.439	30.439	30.439	29.287	27.740	24.097	19.354	11.607
	PSNR	30.718	28.639	28.639	28.639	28.723	28.790	28.917	29.132	29.595
	% of Loss	-	6.768	6.768	6.768	6.494	6.276	5.863	5.163	3.655
Total	No. of Bits	4249620	3032712	3032716	3032711	3069953	3130758	3276024	3475985	3793053
	% of Saving	-	28.635	28.635	28.635	27.759	26.328	22.910	18.204	10.743
	PSNR	32.530	29.877	29.877	29.877	29.922	29.978	30.094	30.334	31.004
	% of Loss	-	8.155	8.155	8.155	8.017	7.845	7.488	6.750	4.691

Confusion matrix [15] is used to assessthe accuracy of an image classification. Thestrength of a confusion matrix is that it identifiesthe nature of the classification errors, as well astheir quantities. In confusion matrix rowscorrespond to classes in the test set, columnscorrespond to classes in the classification result.The diagonal elements in the matrix representthe number of correctly classified pixels of eachclass. The off-diagonal elements representmisclassified pixels. The overall accuracy iscalculated as the total number of correctlyclassified pixels divided by the total number of test pixels.

Another measure which can be extractedfrom a confusion matrix is the kappa coefficient[16] which is a popular measure to estimateagreement in categorical data. The motivation ofthis measure is to extract from the correctlyclassified percentage the actual percentageexpected by chance. Thus, this coefficient iscalculated as

$$K = \frac{P_0 - P_e}{1 - P_e} \qquad (14)$$

P_0 is observed agreement = $\frac{\text{Total number of correctly classified pixels (diagonal elements)}}{\text{Total number of test pixels}}$

P_e is the Expected agreement = $\left[\frac{\frac{cm^1 xrm^1}{n} + \frac{cm^2 xrm^2}{n} + \ldots\ldots\ldots \frac{cm^n xrm^n}{n}}{n}\right]$ (15)

cm1,, cm2, ... cmnare the column 1, 2 .....n marginals

rm^1 ,rm^2, ... rm^n are the row 1, 2 --------- nmarginals n is the total number of test pixels.

The higher the value of kappa, the betterthe classification performance. If all informationclasses are correctly identified, kappa takes thevalue 1. As the off-diagonal entries increase, thevalue of kappa decreases. For classification two data sets were used. Onewas a 1048 X 920 (Owensvalley image) LandsatTM with all 7 bands. The second data setcontained 500 samples of four ground types,Dense scrub, Rock, Forest and Open scrub of thesame scene. This second data set is used to observe the classification accuracy. All the 2000set patterns were classified simultaneously withthe Maximum Likelihood, Mahalanobis andMinimum distance classifiers. Classificationperformance of all the classifiers is displayed intables 4, 5 & 6. Figure 6 shows the Overallaccuracy of all the classifiers. It is observed thatclassification performance on proposedcompression images is almost same as JPEGstandard compression images and original images.

Table 4 : Confusion matrix for Maximum Likelihood classifier

Original Image						
Spectral Class	Correct Classification (%)	Number of Samples used	Classified as group			
			1	2	3	4
1. Dense scrub	99.9	500	499	1	0	0

2. Rock	100	500	0	500	0	0
3. Forest	94.4	500	0	28	472	0
4. Open scrub	100	500	0	0	0	500
Misclassification= 1.15 %	Overall accuracy= 98.55%			Kappa coefficient=0.9806		
Conventional JPEG Compression image						
Spectral Class	Correct Classification (%)	Number of Samples used	Classified as group			
			1	2	3	4
1. Dense scrub	97.6	500	488	12	0	
2. Rock	100	500	0	500	0	0
3. Forest	90.2	500	1	48	451	0
4. Open scrub	99.2	500	1	3	0	496
Misclassification= 3.25%	Overall accuracy= 96.75 %			Kappa coefficient=0.9567		
Proposed compressed image (Mean filtered approach)						
Spectral Class	Correct Classification (%)	Number of Samples used	Classified as group			
			1	2	3	4
1. Dense scrub	96.8	500	484	16	0	0
2. Rock	100	500	0	500	0	0
3. Forest	77	500	0	115	385	0
4. Open scrub	99.6	500	2	0	0	498
Misclassification= 6.65 %	Overall accuracy=93.35%			Kappa coefficient=0.9113		

Table 5 : Confusion matrix for Mahalanobis classifier

Original Image						
Spectral Class	Correct Classification (%)	Number of Samples used	Classified as group			
			1	2	3	4
1. Dense scrub	100	500	500	0	0	0
2. Rock	95.4	500	20	477	3	0
3. Forest	100	500	0	0	500	0
4. Open scrub	100	500	0	0	0	500
Misclassification= 1.15 %	Overall accuracy= 98.85 %			Kappa coefficient=0.9846		
Conventional JPEG Compression image						
Spectral Class	Correct Classification (%)	Number of Samples used	Classified as group			
			1	2	3	4
1. Dense scrub	98.2	500	491	6	3	0
2. Rock	93	500	23	465	12	0
3. Forest	96.8	500	8	8	484	0
4. Open scrub	99.8	500	1	0	0	499
Misclassification= 3.05 %	Overall accuracy=96.95%			Kappa coefficient=0.9593		
proposed compression image(Mean filtered approach)						
Spectral Class	Correct Classification (%)	Number of Samples used	Classified as group			
			1	2	3	4
1. Dense scrub	98.6	500	493	4	1	2
2. Rock	78.4	500	81	392	27	0
3. Forest	96.2	500	18	1	481	0
4. Open scrub	100	500	0	0	0	500
Misclassification= 6.7%	Overall accuracy= 93.3 %			Kappa coefficient=0.9106		

Table 6 : Confusion matrix for Minimum distance classifier

Original Image						
Spectral Class	Correct Classification (%)	Number of Samples used	Classified as group			
			1	2	3	4
1. Dense scrub	97.2	500	486	1	0	13
2. Rock	71.8	500	118	359	5	18
3. Forest	100	500	0	0	500	0
4. Open scrub	100	500	0	0	0	500
Misclassification= 7.75%	Overall accuracy=92.25%		Kappa coefficient=0.8966			
Conventional JPEG compression image						
Spectral Class	Correct Classification (%)	Number of Samples used	Classified as group			
			1	2	3	4
1. Dense scrub	93.6	500	468	4	5	23
2. Rock	51.2	500	101	256	69	74
3. Forest	98.6	500	7	0	493	0
4. Open scrub	100	500	0	0	0	500
Misclassification= 14.15%	Overall accuracy=85.85 %		Kappa coefficient=0.811			
proposed compression image(Mean filtered approach)						
Spectral Class	Correct Classification (%)	Number of Samples used	Classified as group			
			1	2	3	4
1. Dense scrub	91	500	455	5	6	34
2. Rock	48.8	500	79	244	90	87
3. Forest	96	500	18	2	480	0
4. Open scrub	99.6	500	2	0	0	498
Misclassification= 16.15%	Overall accuracy= 83.85 %		Kappa coefficient=0.784			

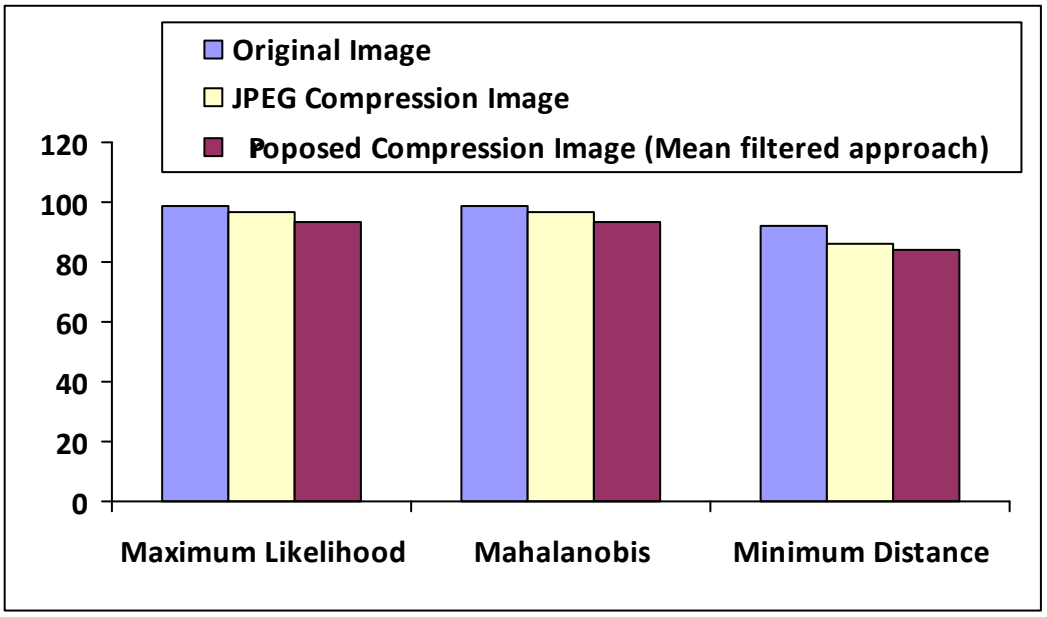

Figure 6 : Overall Accuracy of Classifiers

VI. Conclusions

In this paper, a new filtering basedJPEG compression algorithm is proposed. Wehave compared our proposed algorithm withStandard JPEG compression. From ourexperiments it is evident that our approachgives better compression ratios compared toStandard JPEG. The PSNR resulting from ourapproach is slightly less than Standard JPEGapproach. Also the Classification accuracy oforiginal images, Conventional JPEGcompression images and proposed compression
images are almost same.

If a typical satellite mission goal isclassification only, then we can sendcompressed images from satellite which savesbandwidth requirements of a satellite mission.Also, storage requirement reduces by manyfolds as we will be storing compressed imagesonly. This indirectly reduces powerrequirement needs of the storage system. Inaddition, loading and storing of images takesless time compared to original images, thus response times of imaging systems increases.

References Références Referencias

1. Darrel Hankerson, Greg A. Harris, Peter, D.Johnson Jr," Introduction to information theoryand data compression" second edition. A CRCpress company.
2. John Watkinson" MPEG Handbook" Taylor & Francis, 2004.
3. Bormin Huang,Antonlo J.Plaza,Joan Serra-Sagrista, Chulhee Lee, Younsong LI,Shen-En Qlan Editors Proceedings of SPIE "Satellite Datacompression, communications, and ProcessingVI",.3-5 august 2010 San Dlego, California, United States.
4. John R.Jensen "Introductory Digital Image Processing A Remote Sensing Perspective"Second Edition, Prentice Hall.
5. Cung Nguyen "Detecting Computer –InducedErrors in Remote-Sensing JPEG CompressionAlgorithms".
6. Ch. Ramesh, Dr. N.B. Venkateswarlu, Dr.J.V.R. Murthy "Filter Augmented JPEGAlgorithms: A Critical Performance Study forImproving Bandwidth" IJCA, Vol 60-No-17, ISSN:0975-8887, December 2012, Impact factor:0.821.
7. R.C.Gonzalez and R.E.Woods "Digital Image Processing", 2nd Edition Addison Wesley, USAISBN: 0-201-60078, 1993sz.
8. http://homepages.inf.ed.ac.uk/rbf/HIPR2/mean.htm
9. http://homepages.inf.ed.ac.uk/rbf/HIPR2/median.htm
10. http://en.wikipedia.org/wiki/outlier
11. http://www.stat.yale.edu/courses/1997-98/101/con-fint.htm
12. NB Venkateswarlu& PSVSK Raju, "Three Stage ML Classifier", Pattern Recognition, Vol24, No 11,pp. 1113-1116, 1991.
13. N.B. Venkateswarlu & PSVSK Raju, "Winograd' smethod:a perspective for somepattern recognition problems" PatternRecognition Letters 15(1994) 105-109.
14. NB Venkateswarlu & PSVSK Raju "A newfast classifier for remotely sensed imagery,International Journal of Remote Sensing 1993,Vol. 14. No.2, pp. 383-389.
15. Morton.J.Canty "Image Analysis,Classification and Change Detection in Remote Sensing: With Algorithms for ENVI/IDL", A CRCpress company.
16. Anthony J. Viera, MD; Joanne M. Garrett, PhD" Understanding Interobserver Agreement:The Kappa Statistic" Family Medicine ResearchSeries 360 May 2005.

This page is intentionally left blank