DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
INDIAN INSTITUTE OF TECHNOLOGY-MADRAS

PR Assignment-I

Singular Value Decomposition
Eigen Value Decomposition
Linear and Non Linear Regression

Vikram Singh (Roll No: CS15D011)
Hitesh Gupta (Roll No: CS16M023)

August 24, 2017

Contents

Image Reconstruction

1 EVD

1.1 Square Image

There are 3 ways to perform EVD :-

- Converting to grayscale and then performing EVD
- Performing EVD separately on the 3 channels
- Concatenating 8bit R,G,B number to 24 bit number and then performing EVD

1.1.1 EVD on Gray Image

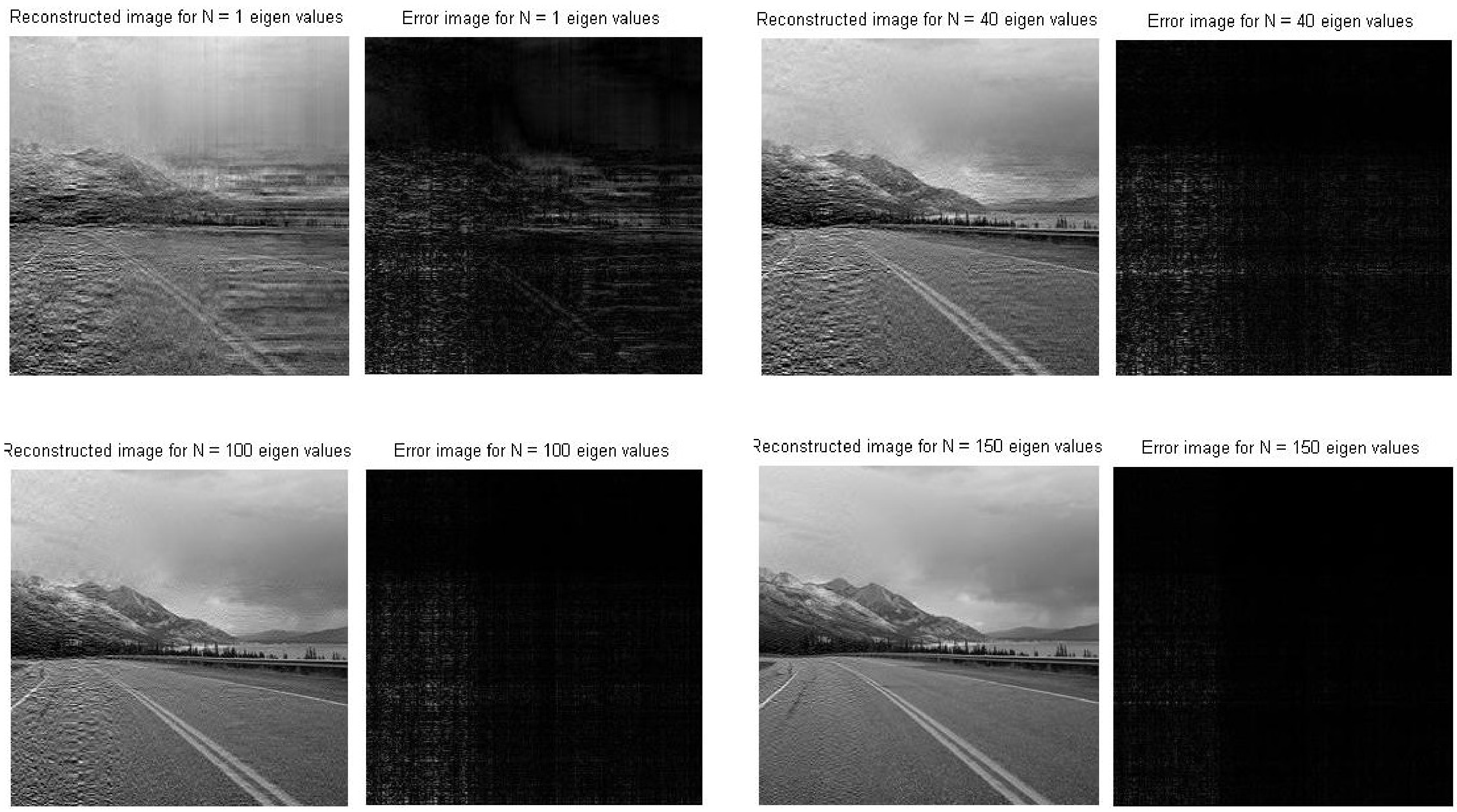


Figure 1.1: For N = 1,40,100,150

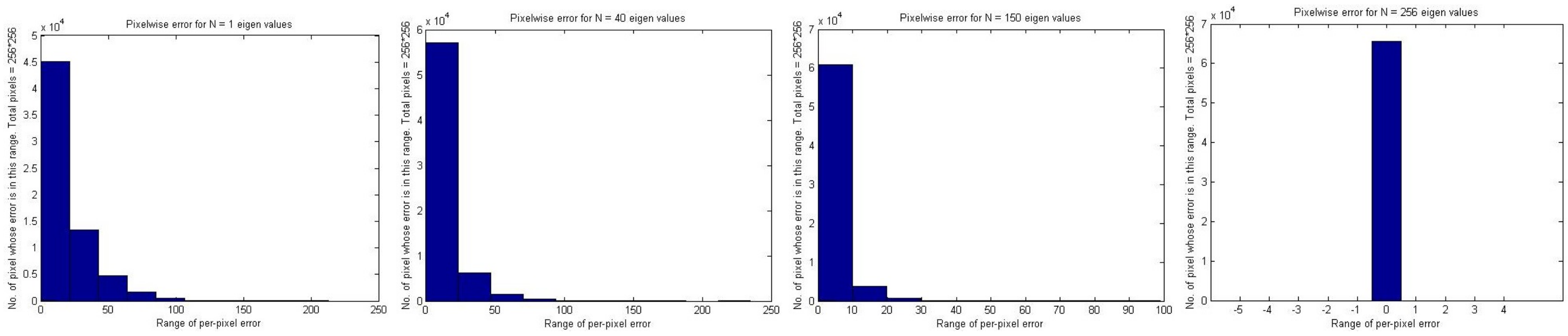


Figure 1.2: Histogram plot for N = 1,40,150,256 showing per pixel error in specific error range

Observations

- Sufficiently good reconstruction was obtained at N=150 and zero error for all 256 eigen values.
- The histogram plots shows that more number of pixels has less error and less number of pixels has high error. In N=256, all the pixels has zero error.
- Most reconstruction happens at N=1. After that the image sharpens for each subsequent value of N.

1.1.2 EVD ON EACH COLOR BAND SEPARATELY

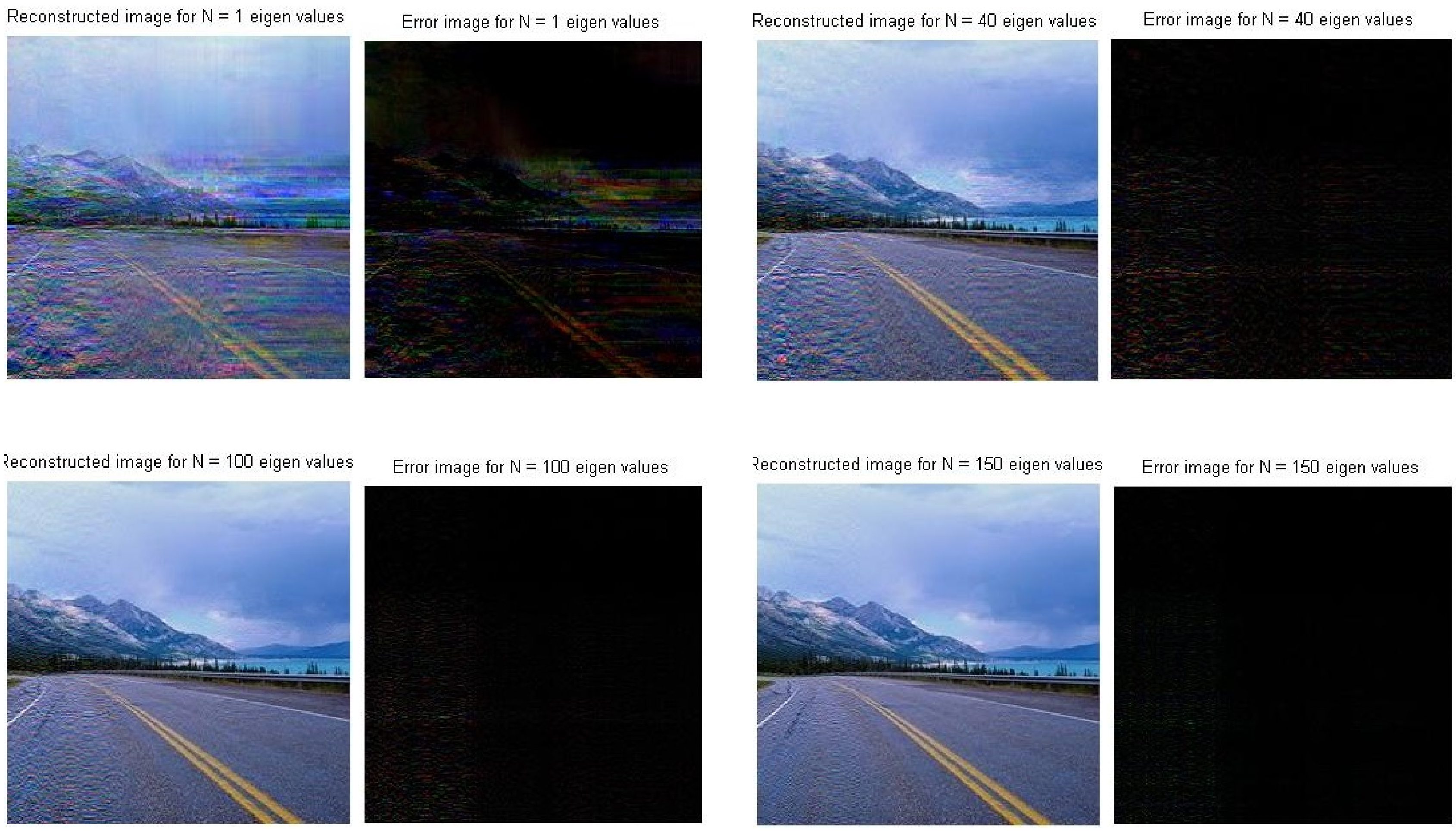


Figure 1.3: For N = 1,40,100,150

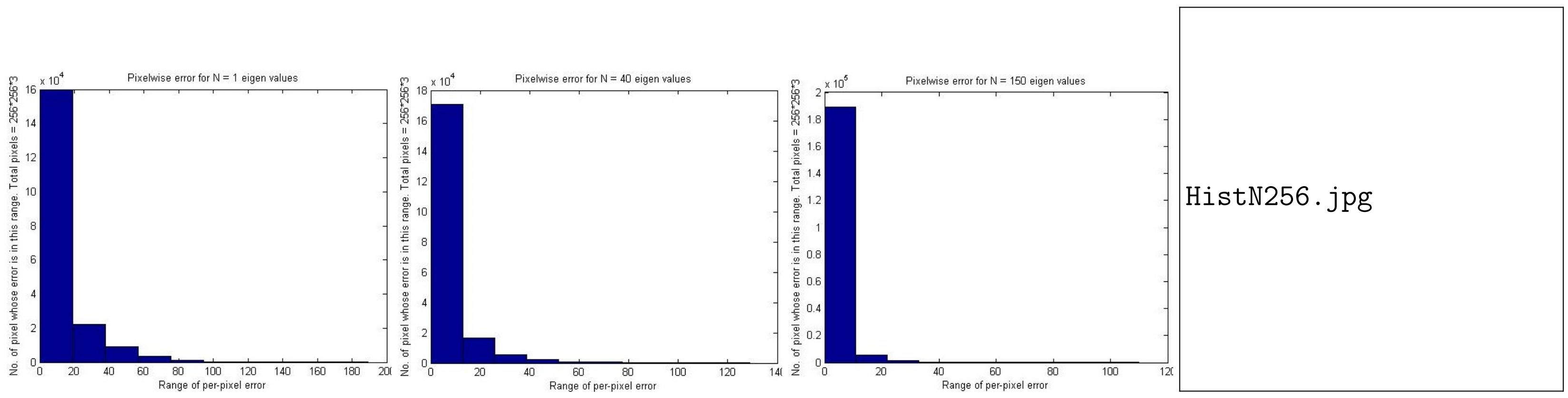


Figure 1.4: Histogram plot for N = 1,40,150,256 showing per pixel error in specific error range

Observations

- Sufficiently good reconstruction was obtained at N=150 and zero error for all 256 eigen values.
- The histogram plots shows that more number of pixels has less error and less number of pixels has high error. In N=256, all the pixels has zero error.
- There is no loss of data (color) in this technique.

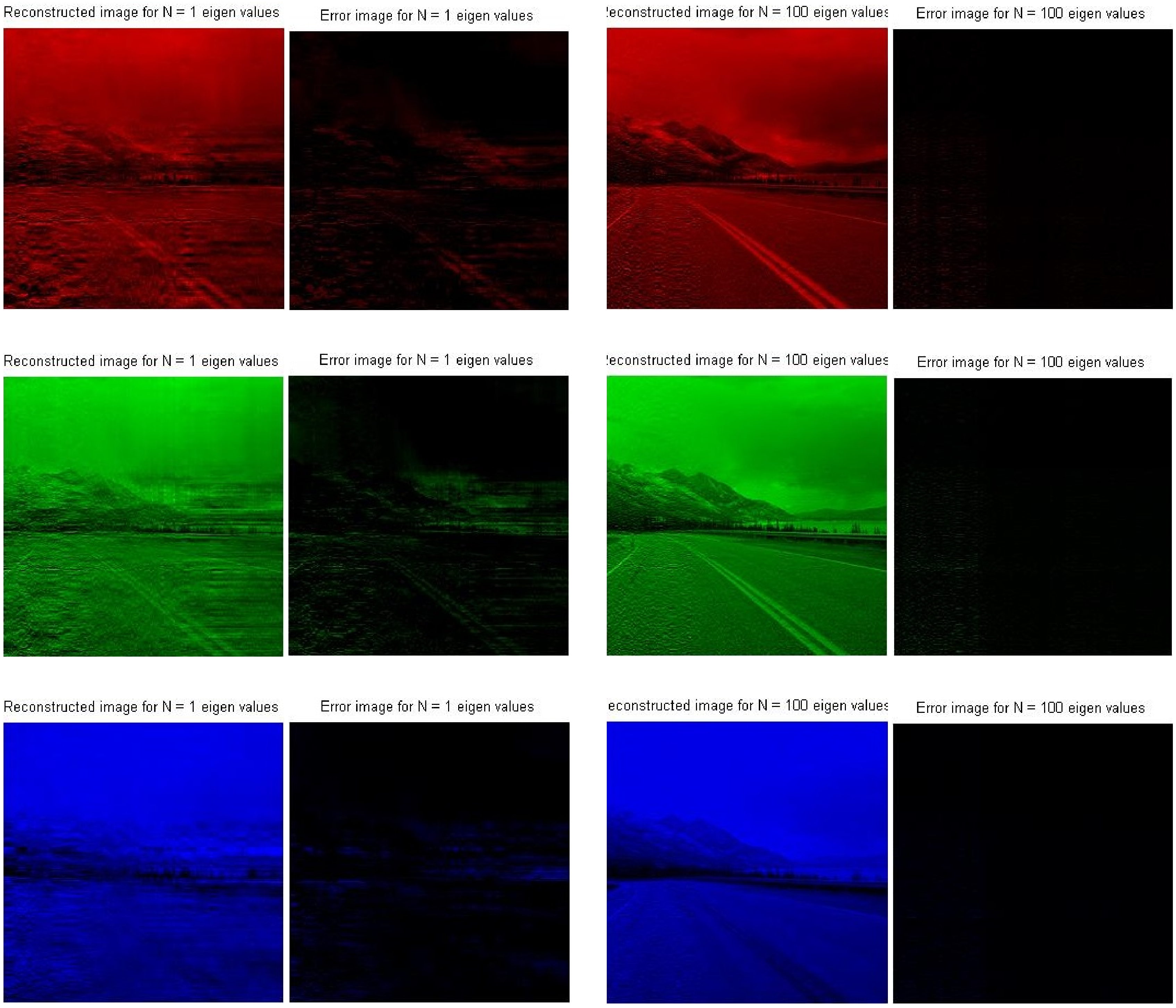


Figure 1.5: Reconstruction for N = 1 and 100 for all 3 colors

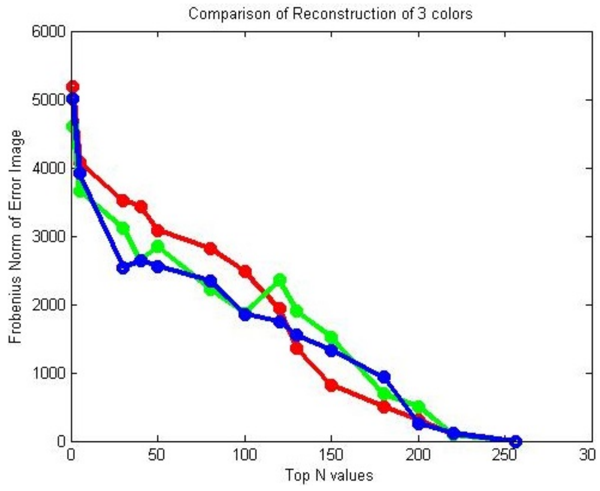


Figure 1.6: Comparison of Reconstruction error for R,G and B color

<u>Observations</u>

- The error in reconstruction for blue image is the least till N=100.
- Since the image is mostly blue in color, initially it got reconstructed faster then the red and green color got reconstructed faster than the blue color. It may be because the general idea of the image is blue but at pixel level, there is red and green also in them.

1.1.3 EVD ON 24 BIT NUMBER (RGB)

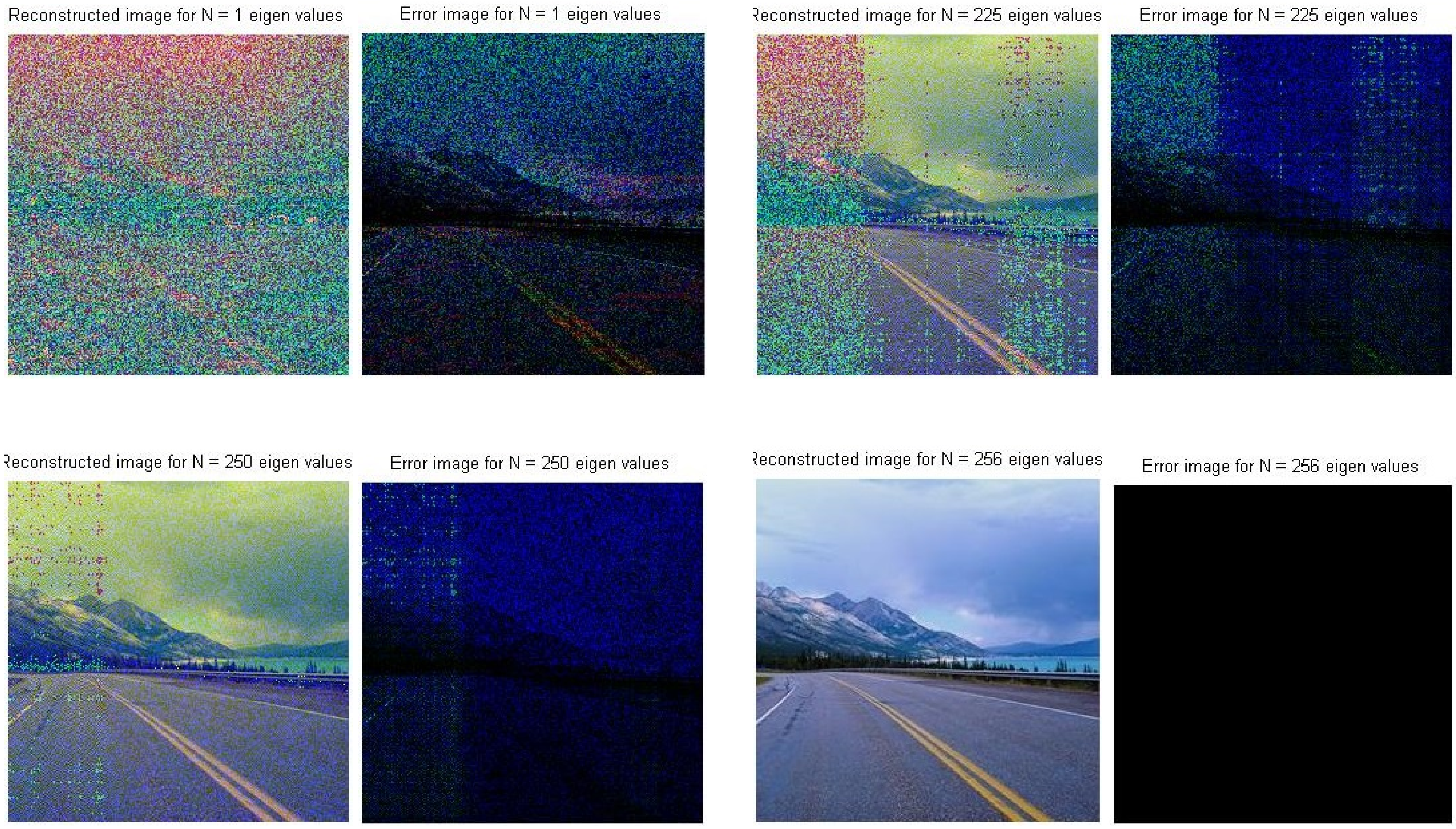


Figure 1.7: For N = 1,225,250 and 256(zero error)

1.1.4 EVD ON 24 BIT NUMBER (BGR)

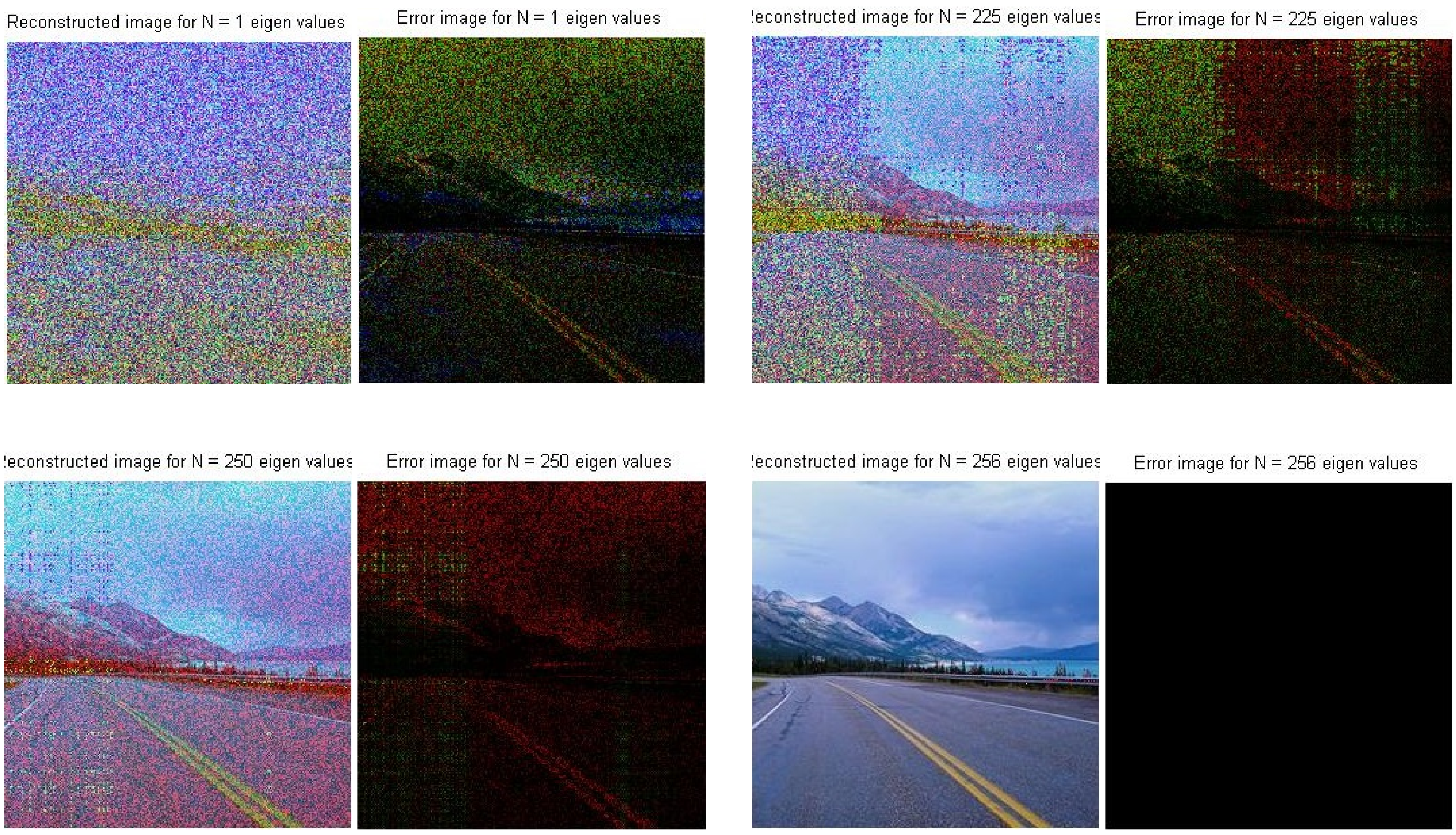


Figure 1.8: For N = 1,225,250 and 256(zero error)

Observations

- Sufficiently good reconstruction was obtained at N=256 for both the methods.
- In the RGB technique we can see that the red color(MSB) got reconstructed first, then green and then blue(LSB). The error images after N=200 are mostly blue
- In the BGR technique can see that the blue color(MSB) got reconstructed first, then green and then red (LSB). The error images are initially green and after N=200 they are mostly red.
- The reason of MSB being reconstructed first is possibly the matrix multiplication while reconstructing the image. When we use top values, the 24-bit number is mostly reconstructed, reconstructing the MSB. Then, if we take more eigen values, the 24-bit number approaches to actual number, reconstructing the LSB too.
- The histogram plots shows that more number of pixels has less error and less number of pixels has high error. In N=256, all the pixels has zero error.

Comparison of the 4 methods

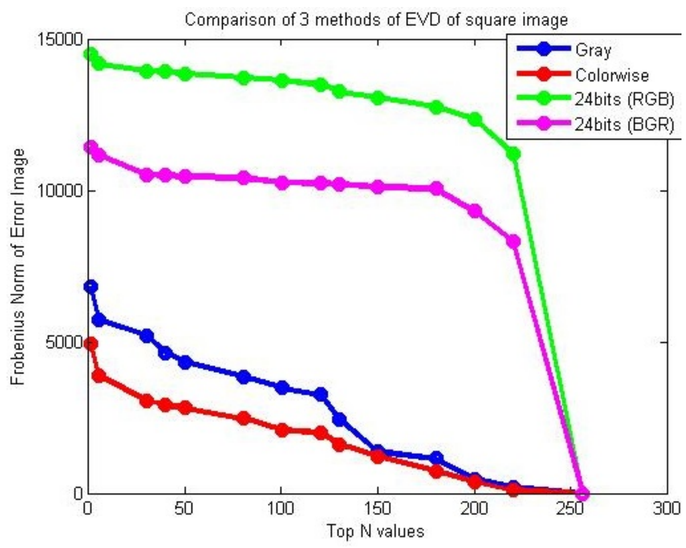


Figure 1.9: Comparison of different methods for EVD of square image

Method	Value of N	Old Size	New Size	Percentage Compression
Conversion to gray image	220	9.35 kB	9.18 kB	1.8%
Colorwise EVD	220	32.5 kB	10.5 kB	67.6%
Converting to 24bit number	256	32.5 kB	10.7 kB	66.1%

Table 1.1: Comparison of the size reduction in good reconstructed image

Observations and Inferences

- For square image, performing EVD on separate channel performs the best as there is no loss of color also.
- 24 bit (BGR) performs better than 24 bit (RGB) because the image is mostly blue so the error is less when blue(MSB) gets reconstructed first even for small values of N.
- With respect to the compression of the image, the most compression is by the 24-bit method.
- With respect to reconstruction, the colorwise methods outperforms the other two.

1.1.5 Random N for each method for Square image EVD

Random values were generated once and they were stored for all further experiments.

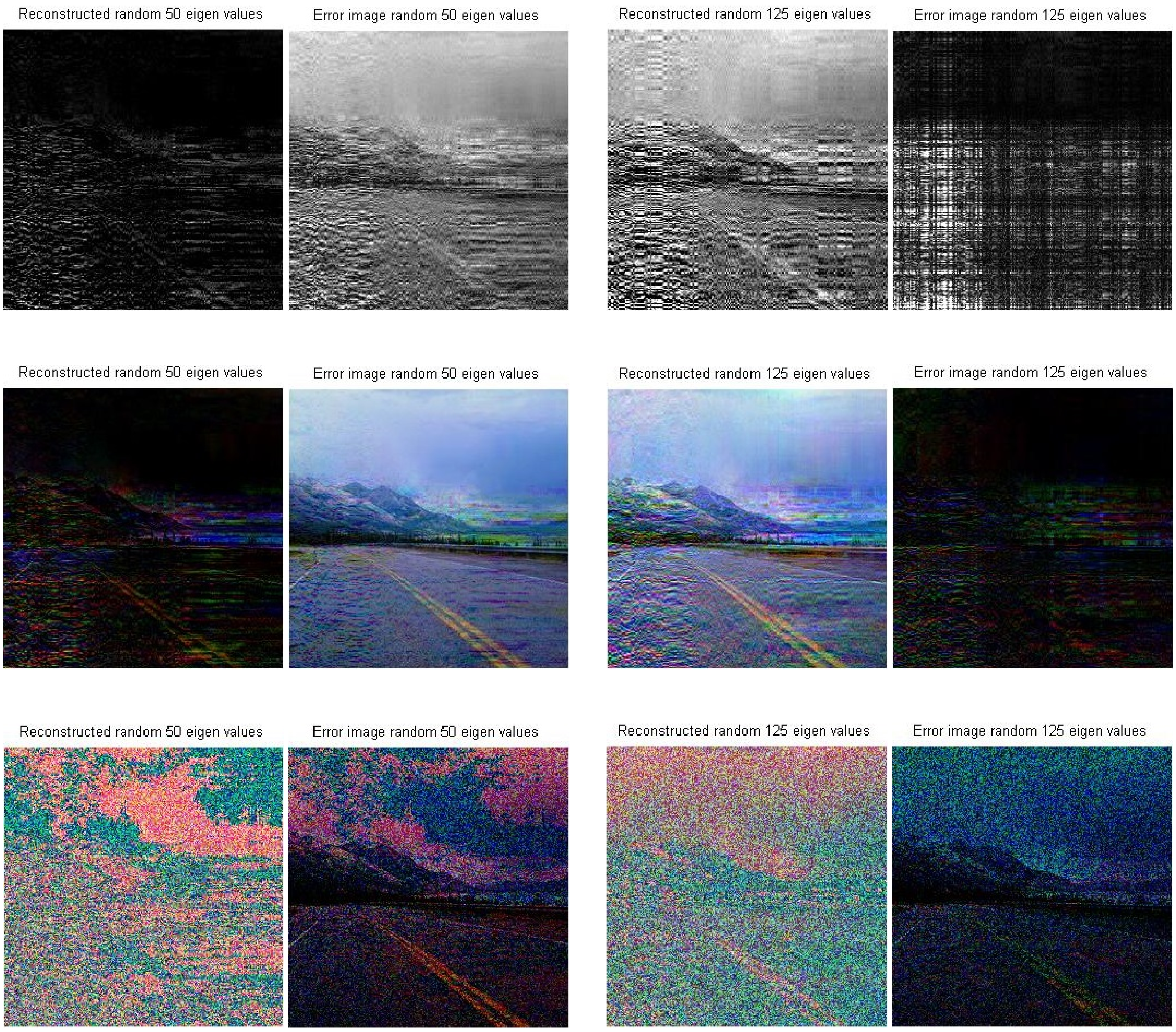


Figure 1.10: For Random N = 50 and 125 for different methods

Observations

- For random N, no specific pattern was obtained but bands and patches are visible which shows high energy eigen vectors are missing.
- The reconstruction happened at random pixels in the image which was expected since we have randomly selected the eigen values.
- As we get different random values and hence different reconstruction everytime, this is not a good way of reconstruction.

1.2 Rectangular Image

1.2.1 EVD on Gray Image

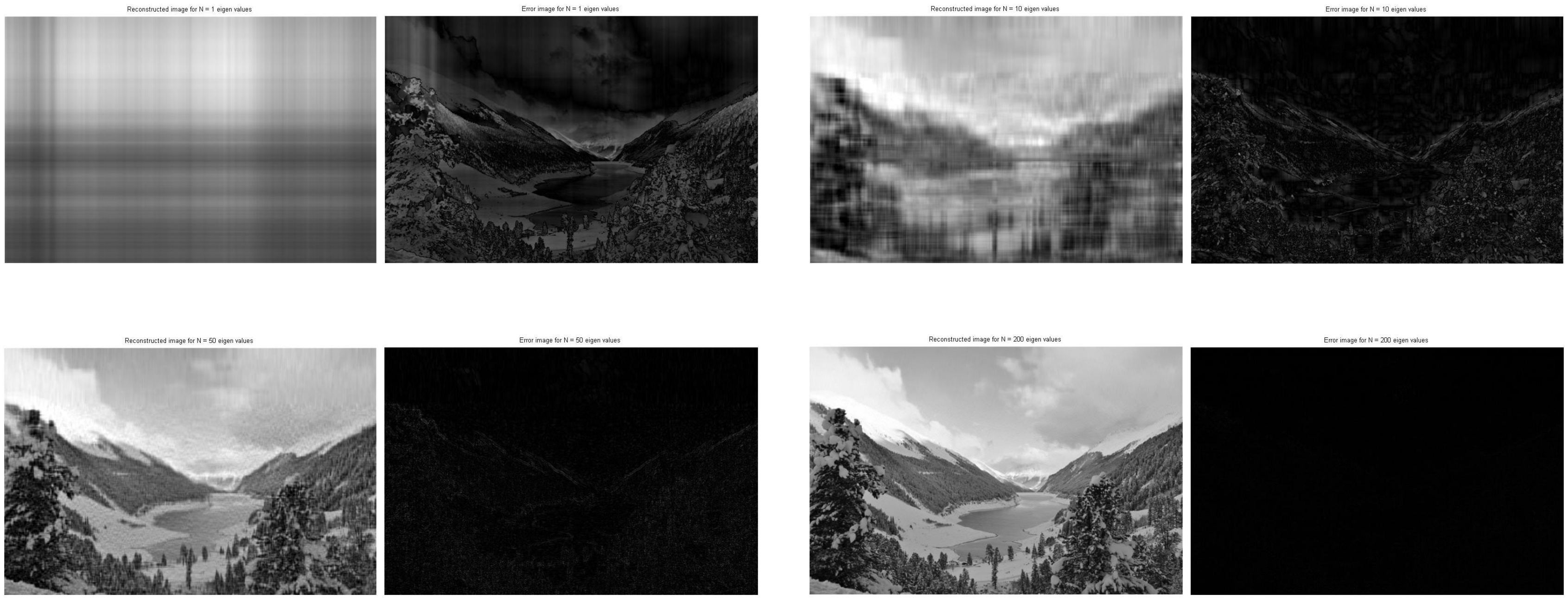


Figure 1.11: For N = 1,10,50 and 200(almost fully reconstructed)

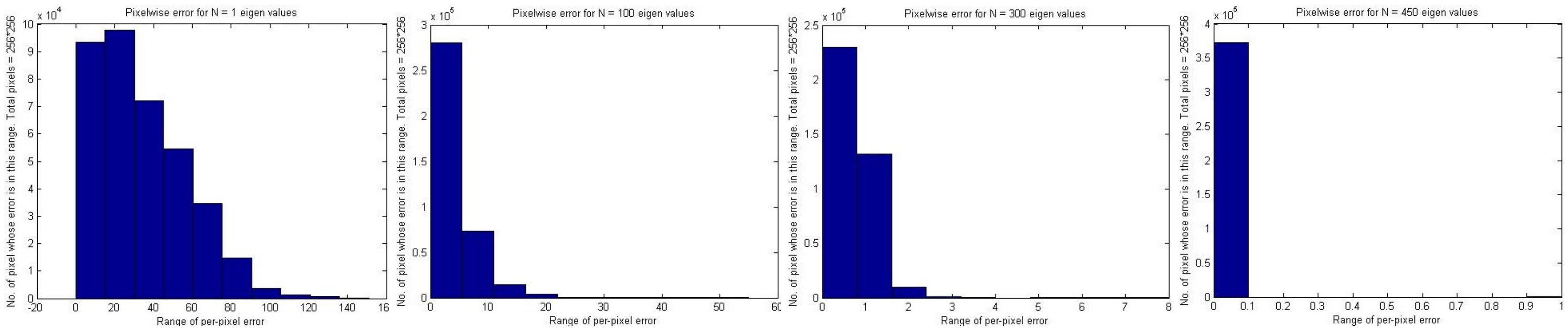


Figure 1.12: Histogram plot for N = 1,100,300,450 showing per pixel error in specific error range

1.2.2 EVD on each color band separately

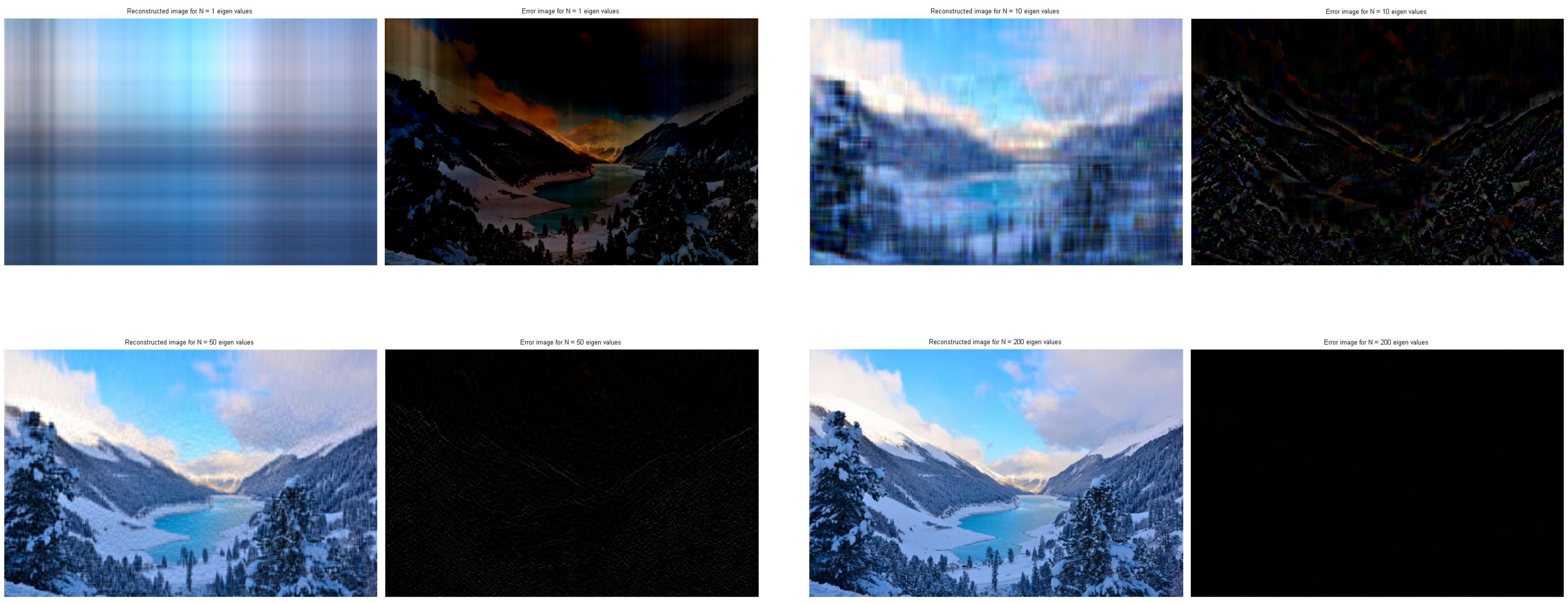


Figure 1.13: For N = 1,10,50 and 200(fully reconstructed)

Observations

- Sufficiently good reconstruction was obtained at N=200
- The histogram plots shows that more number of pixels has less error and less number of pixels has high error. In N=450, all the pixels has zero error.
- Most reconstruction happens at N=1. After that the image sharpens for each subsequent value of N.
- White and black bands are observed in the images which depicts the enrgy of the eigen values.

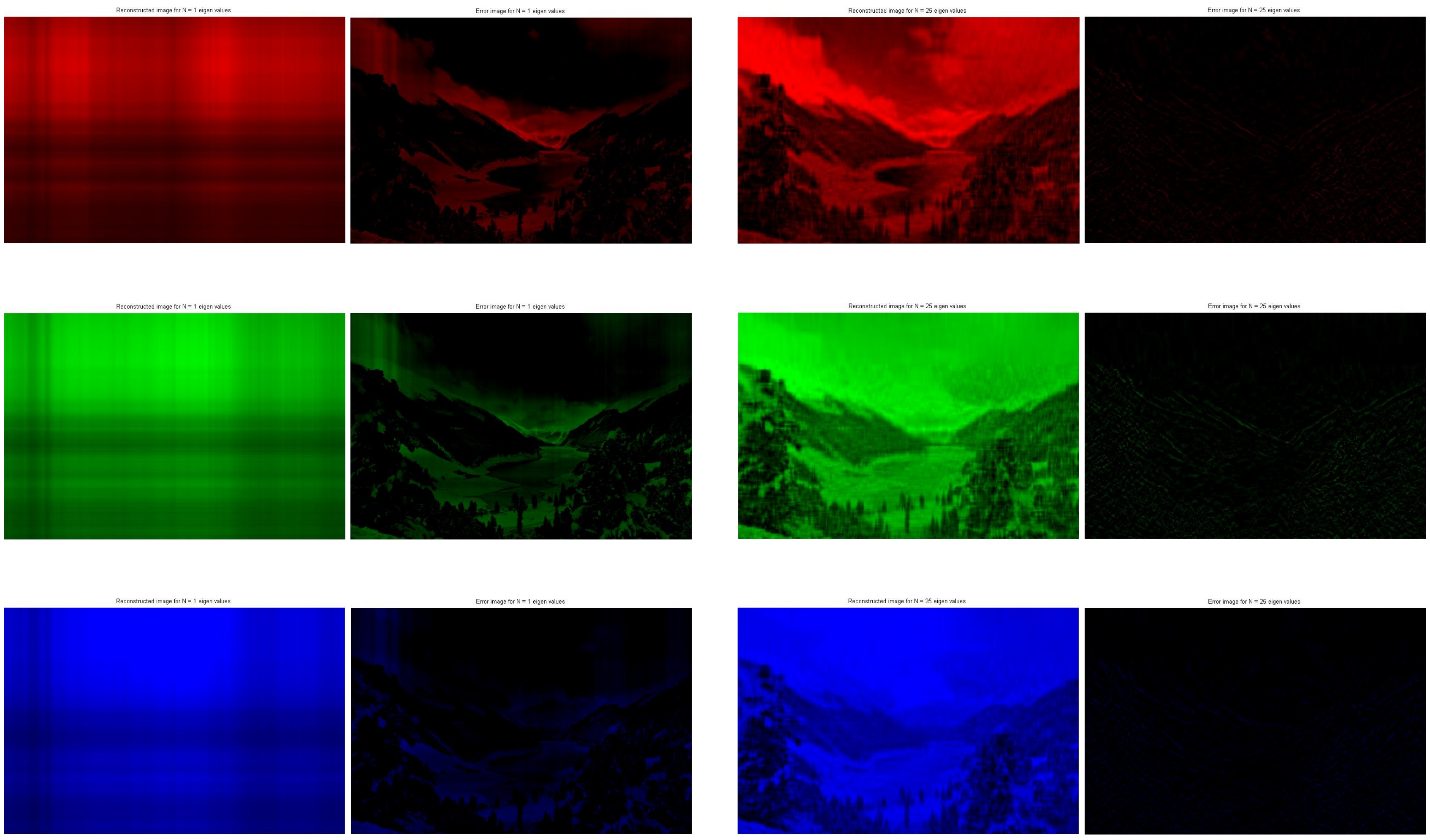


Figure 1.14: Reconstruction for N = 1 and 25 for all 3 colors

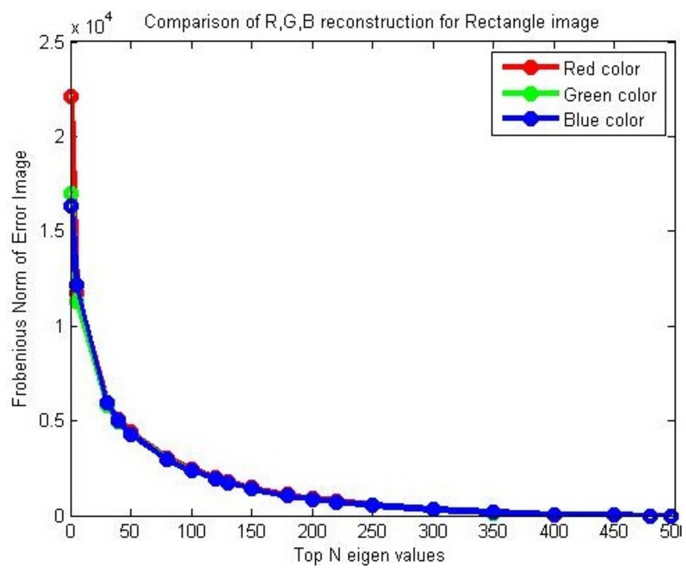


Figure 1.15: Comparison of Reconstruction error for R,G and B color

Observations

- At N=1, Blue color has the least error. This may be because the image is blue and the blue color is getting reconstructed faster initially like the square image.
- The frobenius norm of Error Image of each color follows exactly the same curve unlike square image.

1.2.3 EVD ON 24 BIT NUMBER (RGB)

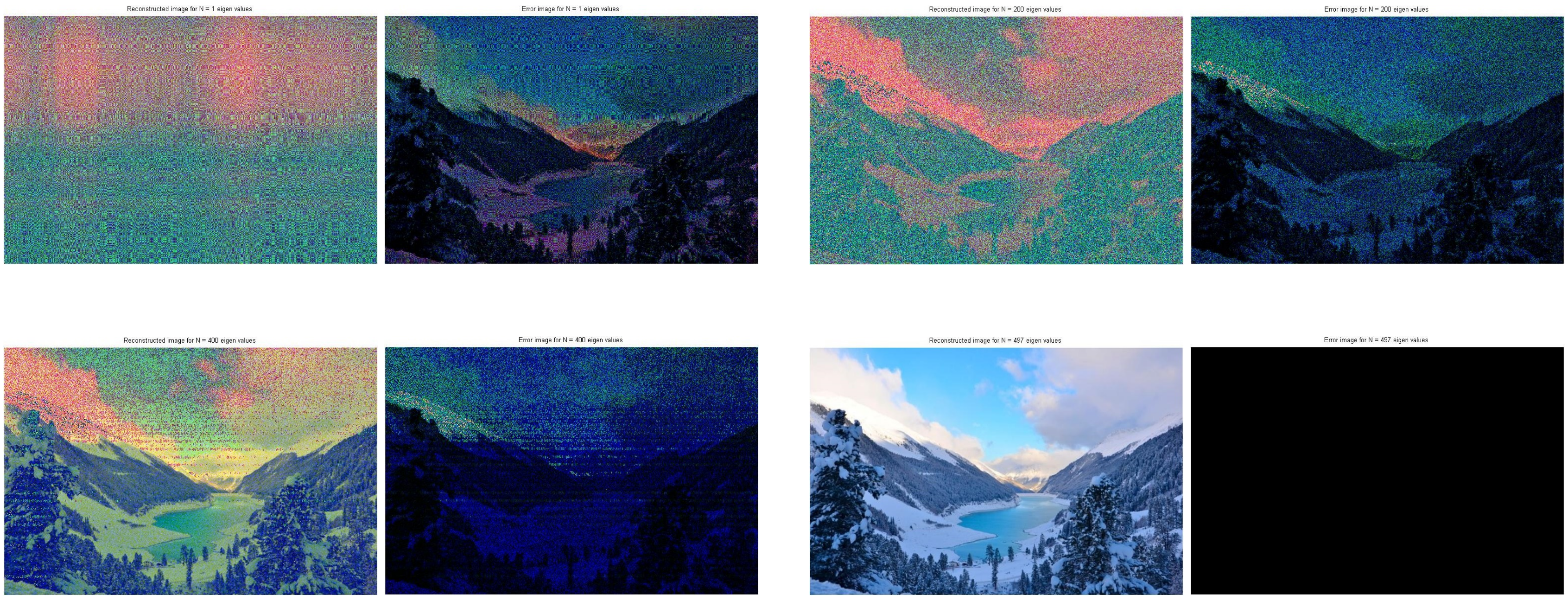


Figure 1.16: For N = 1,200,400 and 497(zero error)

1.2.4 EVD ON 24 BIT NUMBER (BGR)

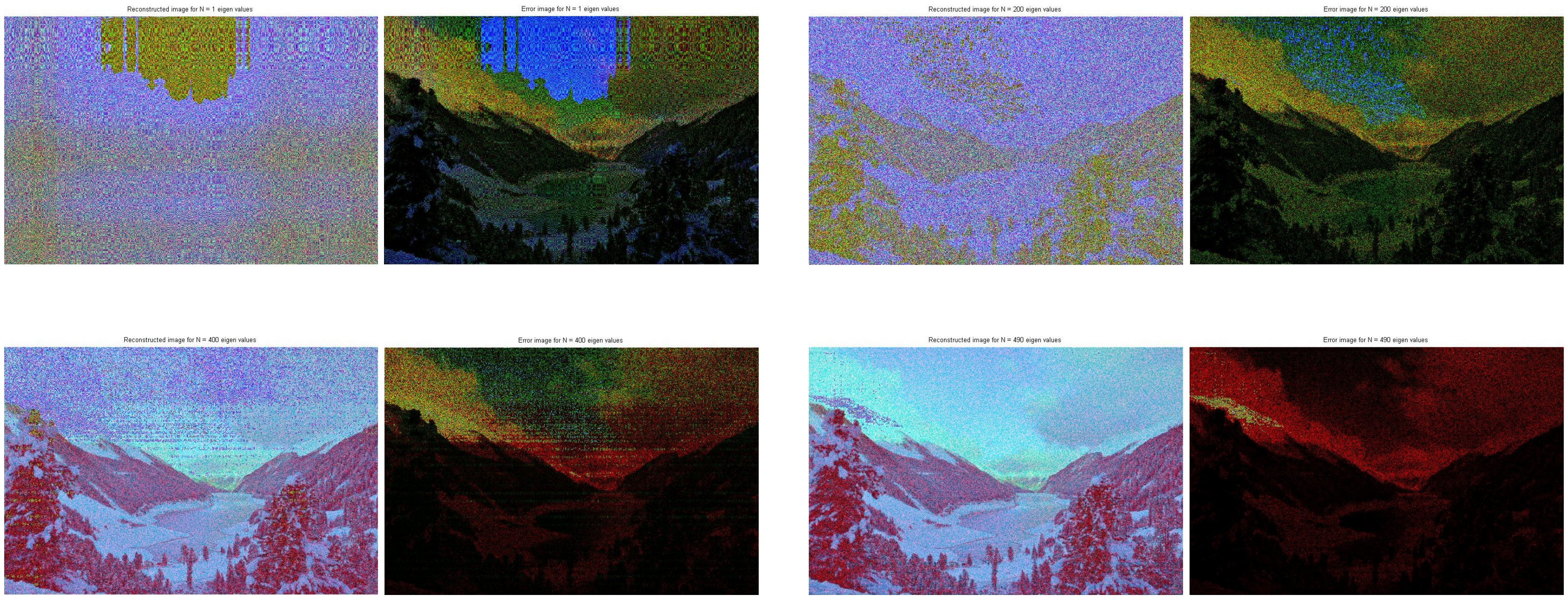
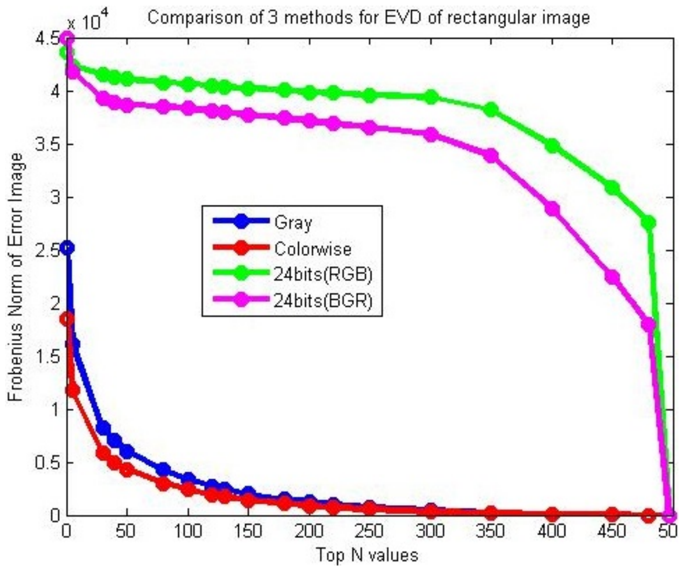


Figure 1.17: For N = 1,200,400,450 and 490

Observations

- Sufficiently good reconstruction was obtained at N=497 for both the methods.
- In the RGB technique we can see that the red color(MSB) got reconstructed first, then green and then blue(LSB). This can be seen as blue color in error images is initially less. Later, at higher N, there was only blue error left in the image.
- In the BGR technique can see that the blue color(MSB) got reconstructed first, then green and then red (LSB). This can be seen as red color in error images is initially less. Later, at higher N, there was only red error left in the image.
- The reason of MSB being reconstructed first is possibly the matrix multiplication while reconstructing the image. When we use top values, the 24-bit number is mostly reconstructed, reconstructing the MSB. Then, if we take more eigen values, the 24-bit number approaches to actual number, reconstructing the LSB too.
- The histogram plots shows that more number of pixels has less error and less number of pixels has high error. In N=497, all the pixels has zero error.



Method	N	Old Size	New Size	Percentage Compression
Gray	200	50.2 kB	48.6 kB	3.12%
Color-wise	200	97.0 kB	58.8 kB	40.21%
24-bit	497	97.0 kB	57.0 kB	41.63%

Figure 1.18: Comparison of different methods for EVD of Rectangular image

Observations and Inferences

- For rectangle image, performing EVD on separate channel performs the best. Converting to grayscale also is good but there is loss of color in that.
- 24 bit (BGR) performs better than 24 bit (RGB) because the image is mostly blue so the error is less when blue(MSB) gets reconstructed first even for small values of N.
- With respect to the compression of the image, the most compression is by the 24-bit method.
- With respect to reconstruction, the colorwise methods outperforms the other two.

1.2.5 Random N for each method for Rectangular image EVD

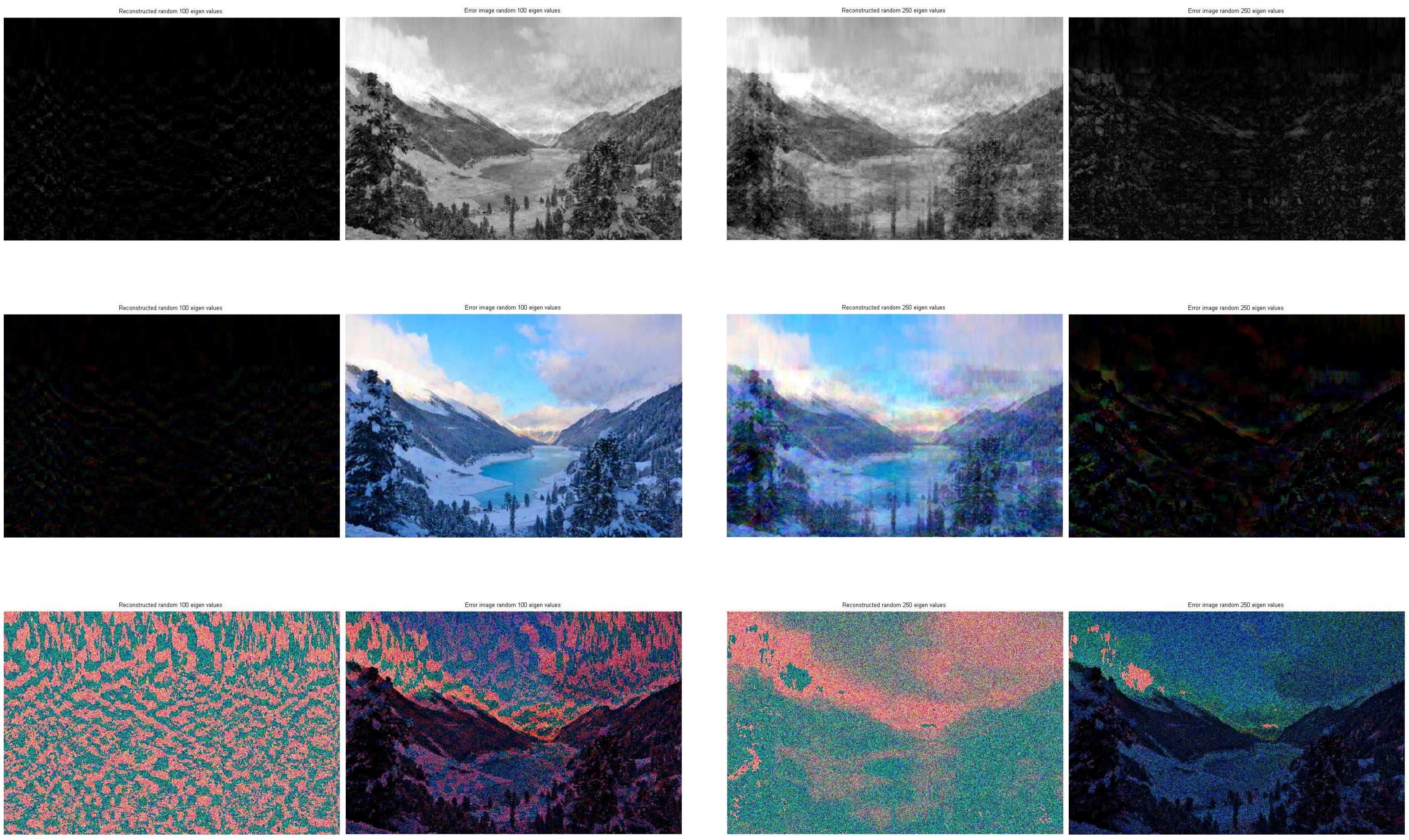


Figure 1.19: For Random N = 100 and 250 for different methods

2 SVD

2.1 SQUARE IMAGE

There are 3 ways to perform EVD :-

- Converting to grayscale and then performing EVD
- Performing EVD separately on the 3 channels
- Concatenating 8bit R,G,B number to 24 bit number and then performing EVD

2.1.1 SVD ON GRAY IMAGE

Figure 2.1: For N = 1,40,100,150

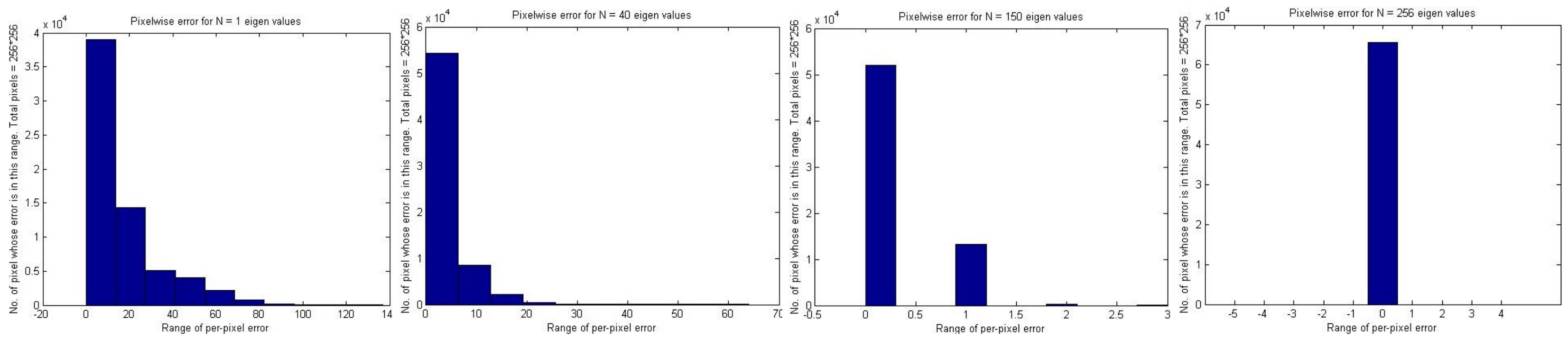


Figure 2.2: Histogram plot for N = 1,40,150,256 showing per pixel error in specific error range

2.1.2 SVD ON EACH COLOR BAND SEPARATELY

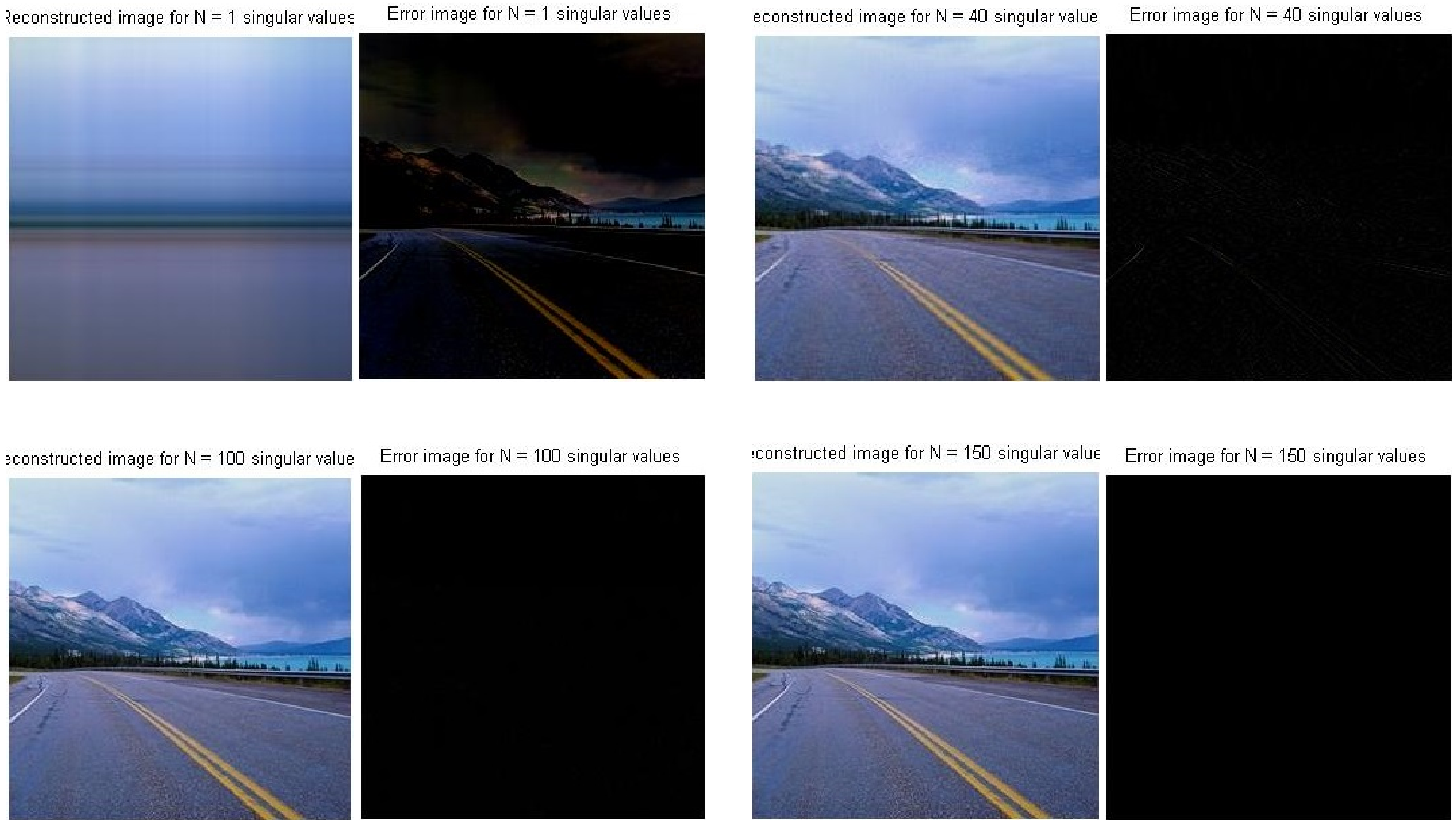


Figure 2.3: For N = 1,40,100,150(almost zero error)

Observations

- Sufficiently good reconstruction was obtained at N=150 and zero error for all 256 eigen values.
- The histogram plots shows that more number of pixels has less error and less number of pixels has high error. In N=256, all the pixels has zero error.
- Most reconstruction happens at N=1. After that the image sharpens for each subsequent value of N.
- There is no data loss (color) for this technique.
- SVD seems to perform much better than EVD as we got the reconstructed image at much less N than in EVD

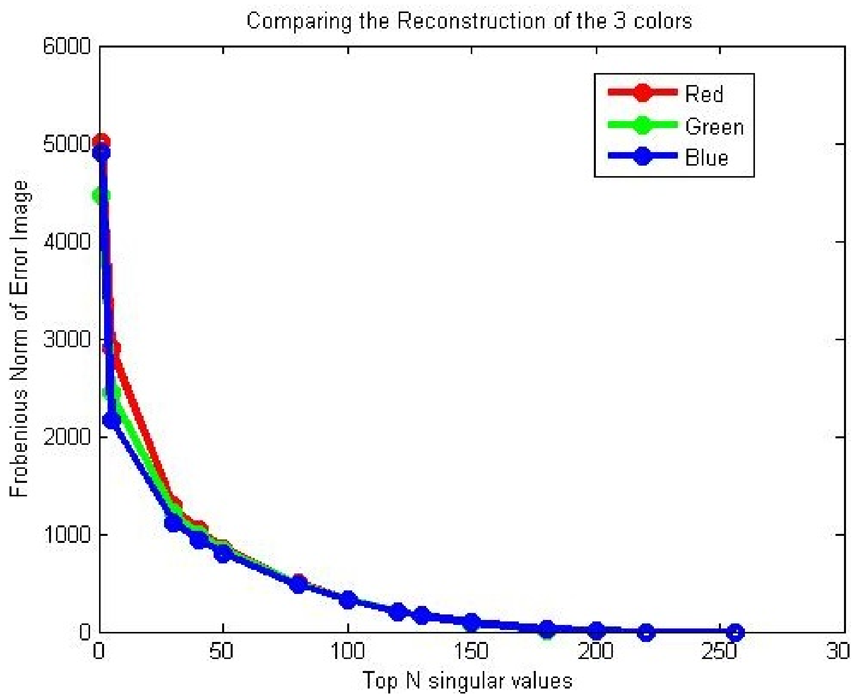


Figure 2.4: Comparison of Reconstruction error for R,G and B color

Observations

- The error in reconstruction for blue image is the least till N=50 and after that all 3 colors reconstruct with the same pace.
- Since the image is mostly blue in color, initially it got reconstructed faster. But later, when even the image got more sharp, all three colors started contributing equally, so they get reconstructed together.

.
.
.

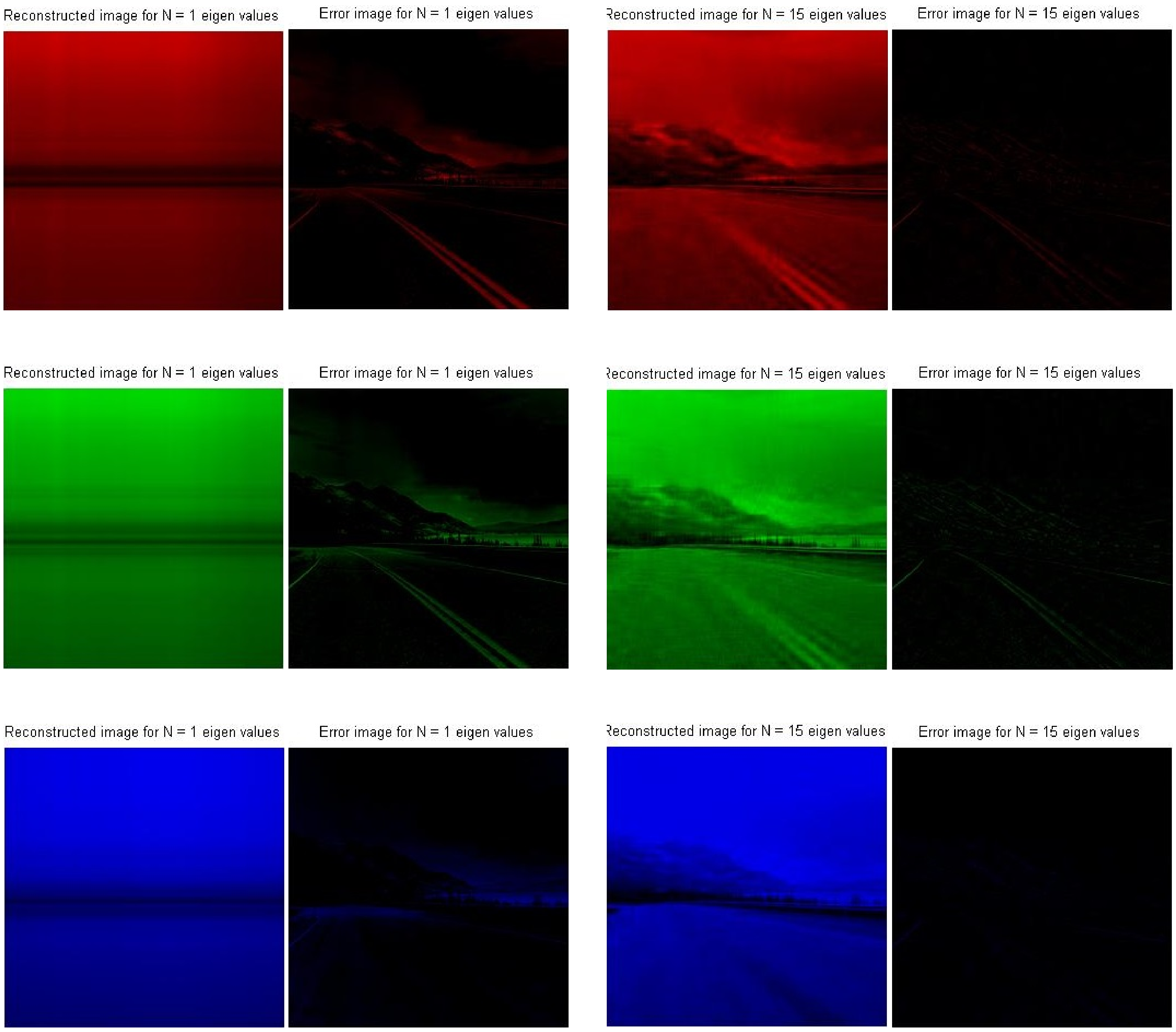


Figure 2.5: Reconstruction for N = 1 and 15 for all 3 colors

2.1.3 SVD ON 24 BIT NUMBER (RGB)

Figure 2.6: For N = 1,225,250 and 256(zero error)

2.1.4 SVD ON 24 BIT NUMBER (BGR)

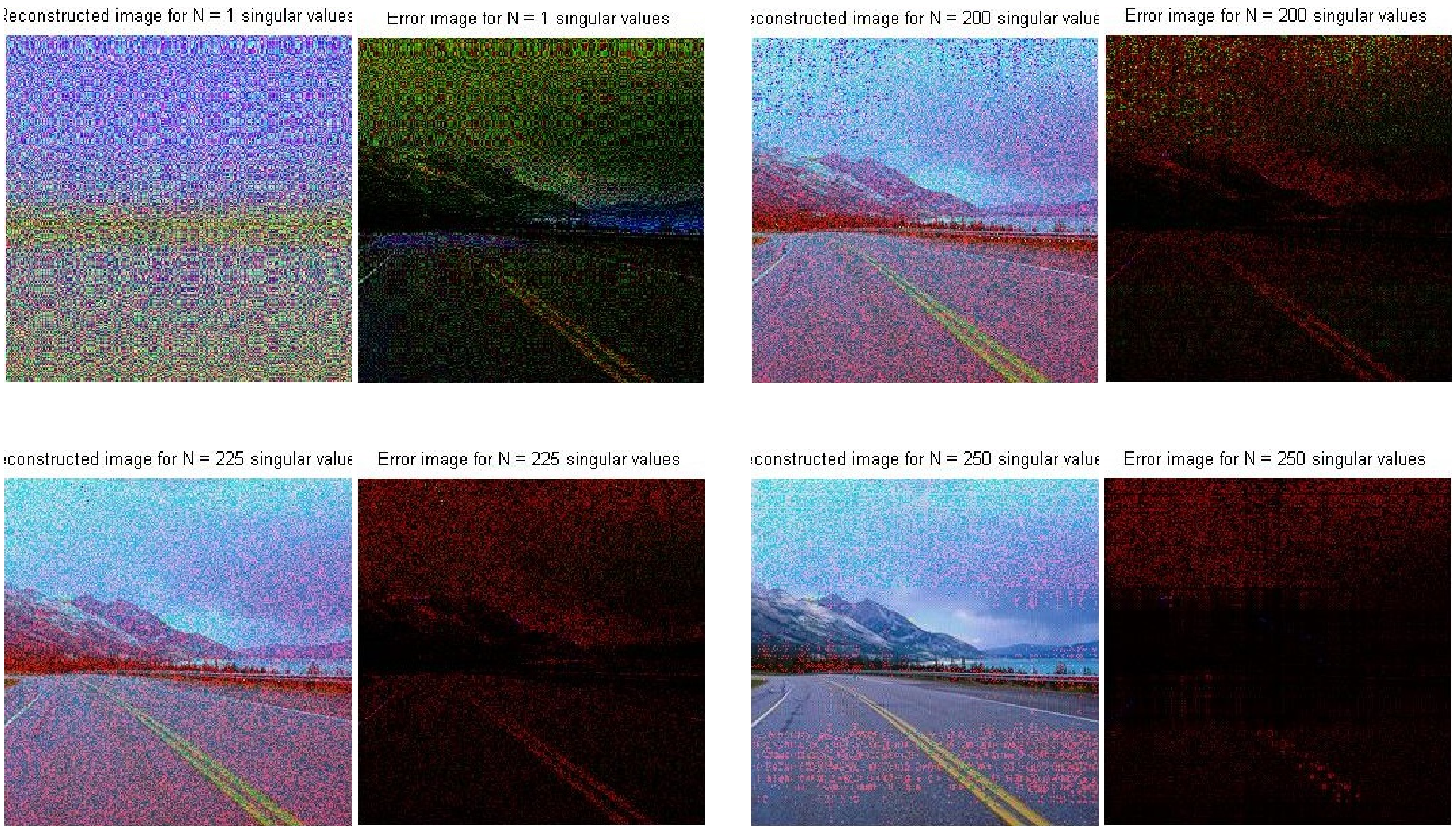


Figure 2.7: For N = 1,225,250 and 256(zero error)

Observations

- Sufficiently good reconstruction was obtained at N=256 for both the methods.
- In the RGB technique we can see that the red color(MSB) got reconstructed first, then green and then blue(LSB). The error images after N=200 are mostly blue
- In the BGR technique can see that the blue color(MSB) got reconstructed first, then green and then red (LSB). The error images are initially green and after N=200 they are mostly red.
- The reason of MSB being reconstructed first is possibly the matrix multiplication while reconstructing the image. When we use top values, the 24-bit number is mostly reconstructed, reconstructing the MSB. Then, if we take more eigen values, the 24-bit number approaches to actual number, reconstructing the LSB too.
- The histogram plots shows that more number of pixels has less error and less number of pixels has high error. In N=256, all the pixels has zero error.

Comparison of the 4 methods

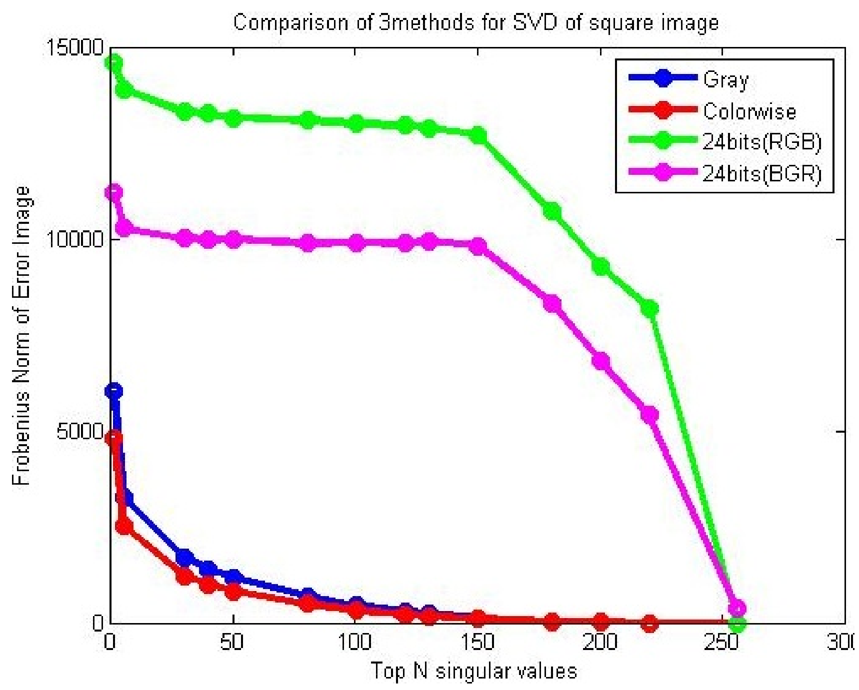


Figure 2.8: Comparison of different methods for SVD of square image

Method	Value of N	Old Size	New Size	Percentage Compression
Conversion to gray image	100	9.35 kB	8.94 kB	2.03%
Colorwise SVD	100	32.5 kB	10.4 kB	67.8%
Converting to 24bit number	256	32.5 kB	10.7 kB	66.1%

Table 2.1: Comparison of the size reduction in good reconstructed image

Observations and Inferences

- For square image, performing SVD on separate channel performs the best as there is no loss of color also.
- 24 bit (BGR) performs better than 24 bit (RGB) because the image is mostly blue so the error is less when blue(MSB) gets reconstructed first even for small values of N.
- With respect to the compression of the image, the most compression is by the 24-bit method.
- With respect to reconstruction, the colorwise methods outperforms the other two.

2.1.5 Random N for each method for Square image SVD

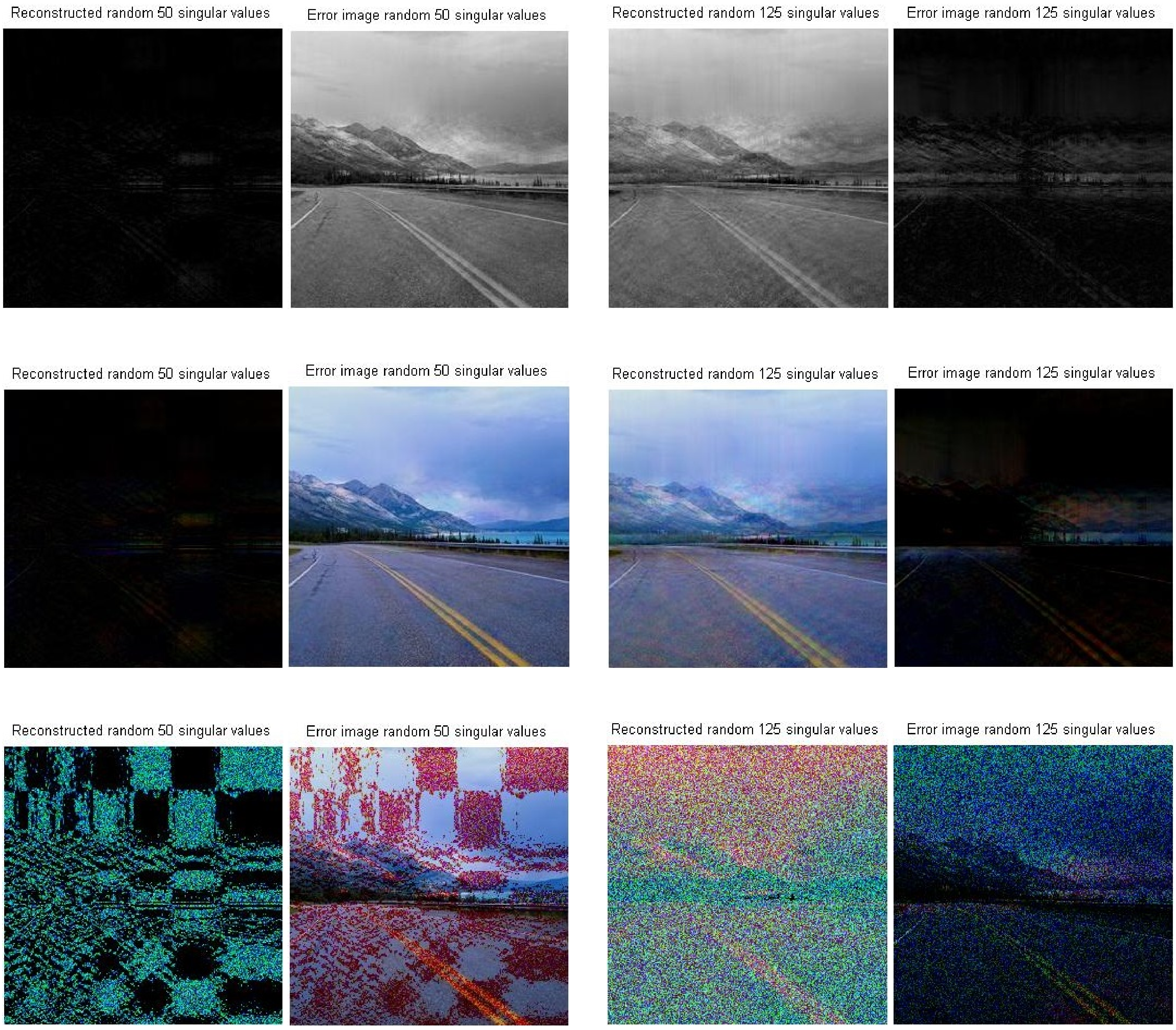


Figure 2.9: For Random N = 50 and 125 for different methods

Observations

- For random N, no specific pattern was obtained but bands and patches are visible which shows high energy singular vectors are missing.
- The reconstruction happened at random pixels in the image which was expected since we have randomly selected the singular values.
- As we get different random values and hence different reconstruction every time, this is not a good way of reconstruction.

2.2 Rectangular Image

2.2.1 SVD on Gray Image

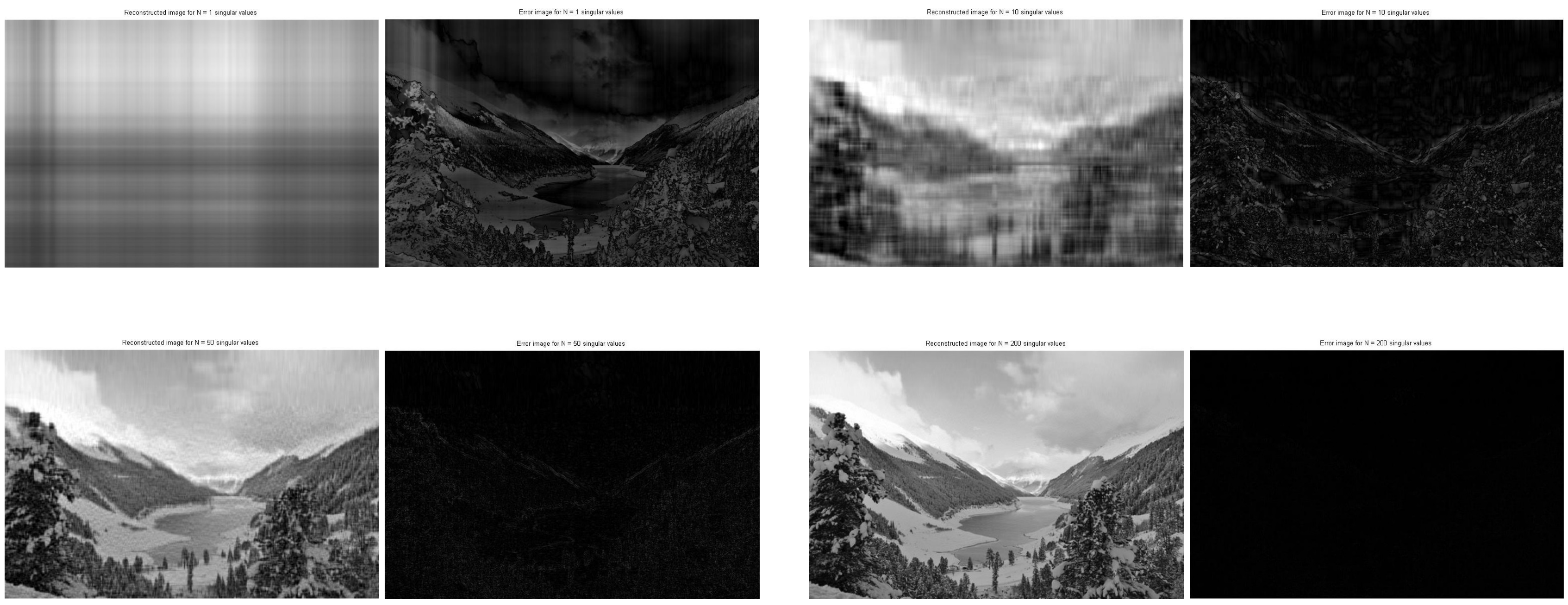


Figure 2.10: For N = 1,10,50 and 200(almost fully reconstructed)

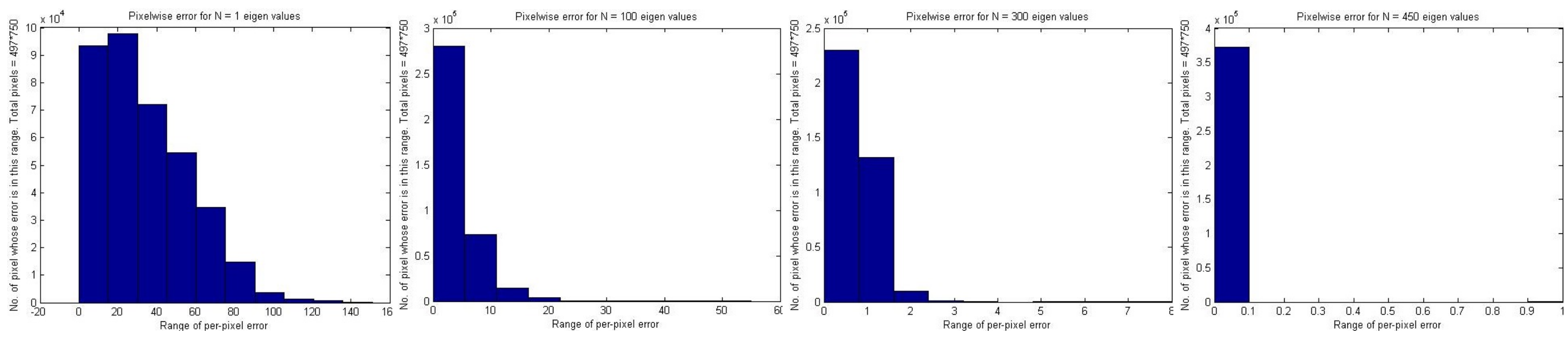


Figure 2.11: Histogram plot for N = 1,100,300,450 showing per pixel error in specific error range

2.2.2 SVD on each color band separately

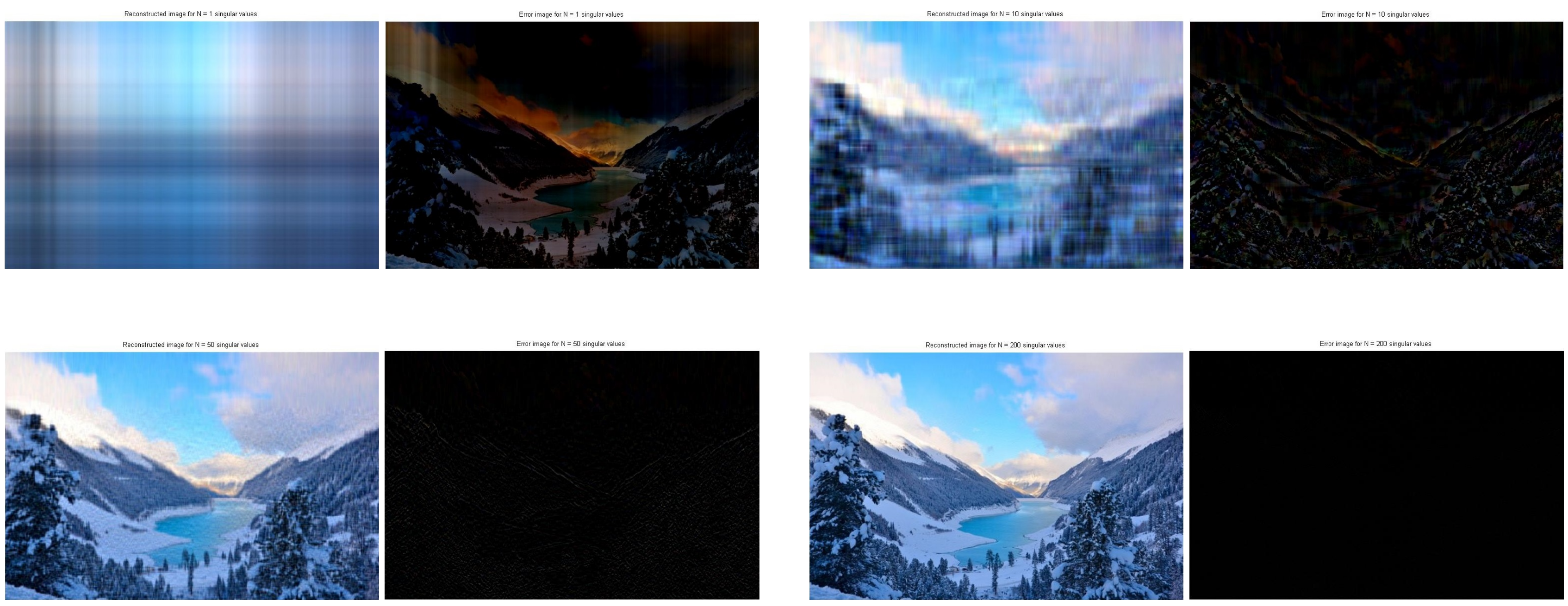


Figure 2.12: For N = 1,10,50 and 200(fully reconstructed)

Observations

- Sufficiently good reconstruction was obtained at N=200
- The histogram plots shows that more number of pixels has less error and less number of pixels has high error. In N=450, all the pixels has zero error.
- Most reconstruction happens at N=1. After that the image sharpens for each subsequent value of N.
- White and black bands are observed in the images which depicts the enrgy of the eigen values.

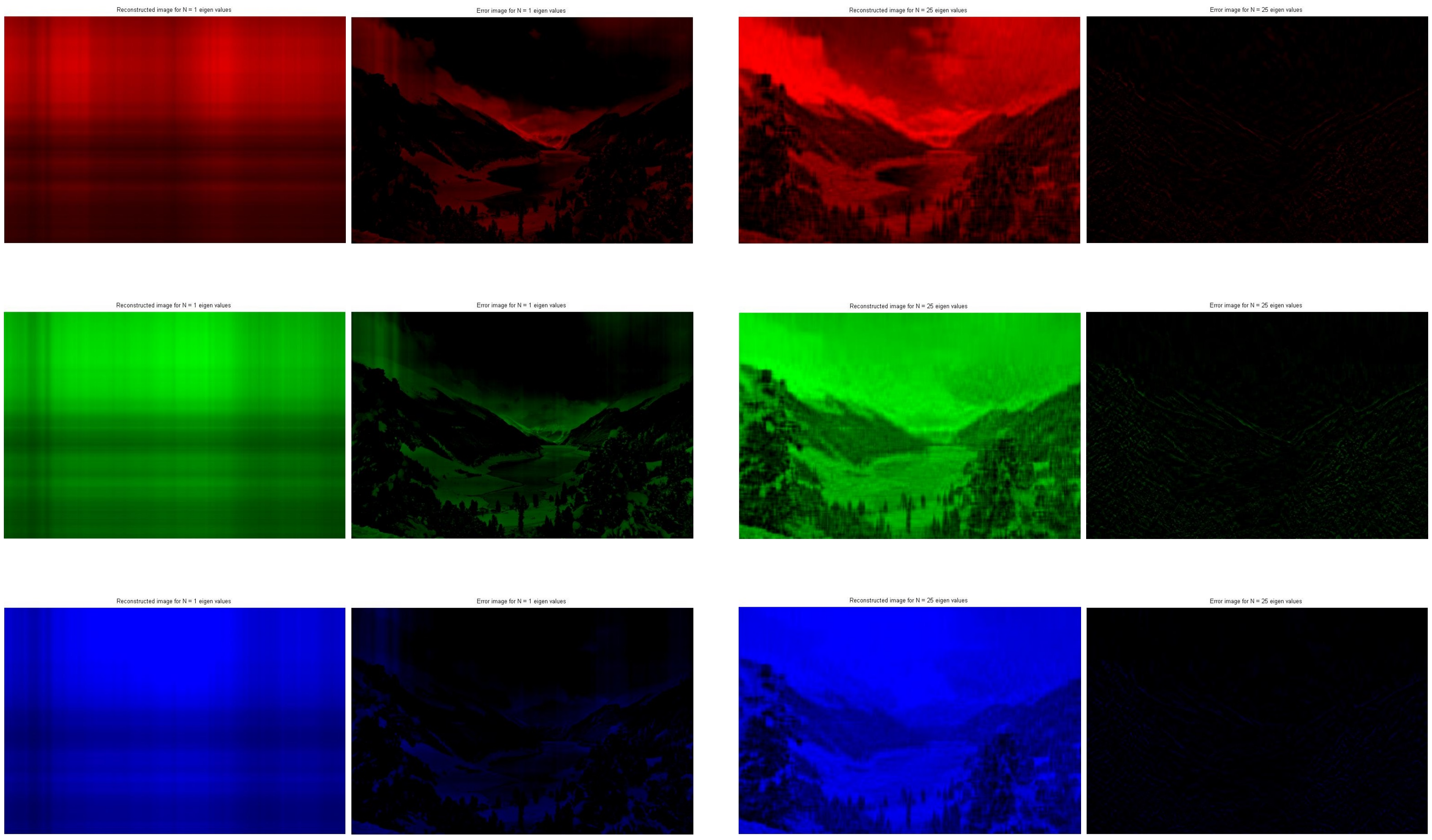


Figure 2.13: Reconstruction for N = 1 and 25 for all 3 colors

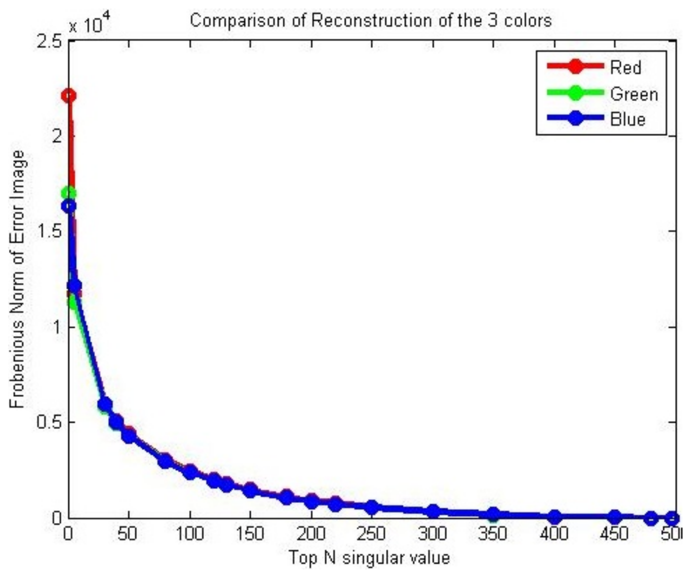


Figure 2.14: Comparison of Reconstruction error for R,G and B color

Observations

- At N=1, Blue color has the least error. This may be because the image is blue and the blue color is getting reconstructed faster initially like the square image.
- The frobenius norm of Error Image of each color follows exactly the same curve unlike square image.

2.2.3 SVD ON 24 BIT NUMBER (RGB)

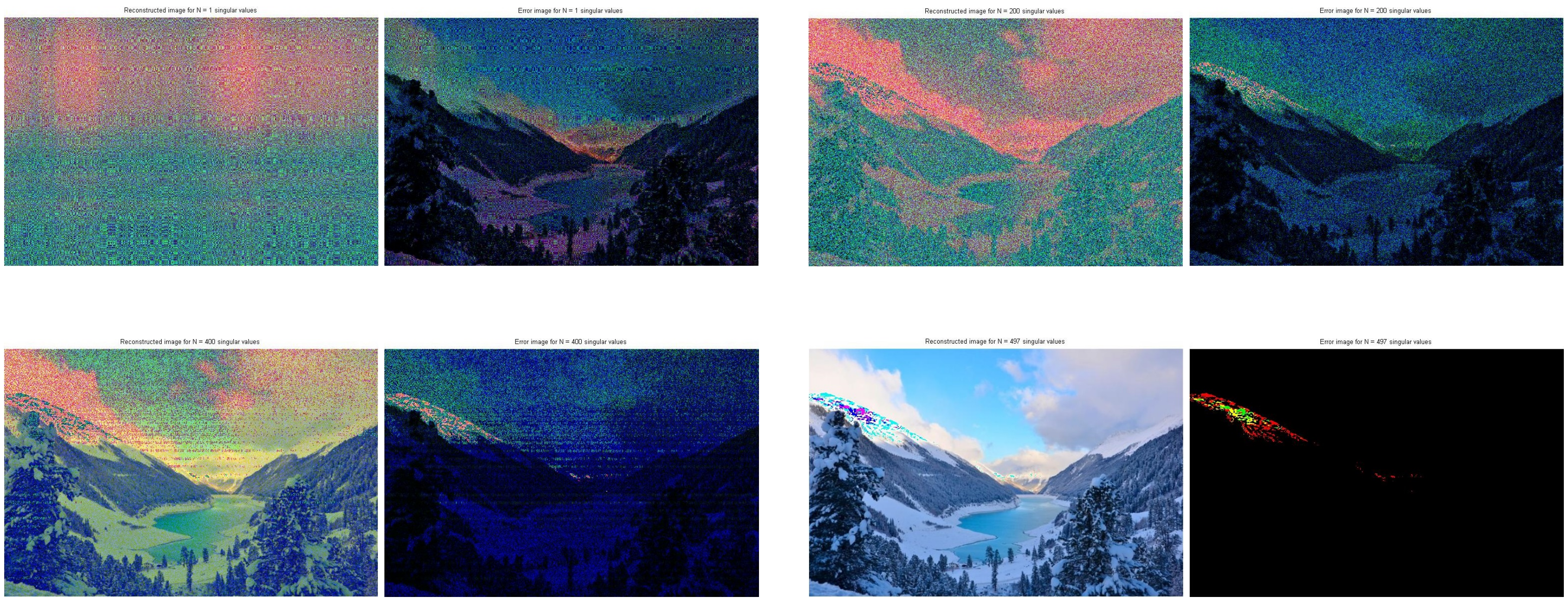


Figure 2.15: For N = 1,200,400 and 497(non-zero error)

Observations

- Sufficiently good reconstruction was obtained at N=497 for both the methods.
- In the RGB technique we can see that the red color(MSB) got reconstructed first, then green and then blue(LSB). This can be seen as blue color in error images is initially less. Later, at higher N, there was only blue error left in the image.
- In the BGR technique can see that the blue color(MSB) got reconstructed first, then green and then red (LSB). This can be seen as red color in error images is initially less. Later, at higher N, there was only red error left in the image.
- The reason of MSB being reconstructed first is possibly the matrix multiplication while reconstructing the image. When we use top values, the 24-bit number is mostly reconstructed, reconstructing the MSB. Then, if we take more eigen values, the 24-bit number approaches to actual number, reconstructing the LSB too.
- The histogram plots shows that more number of pixels has less error and less number of pixels has high error. In N=497, all the pixels has zero error.

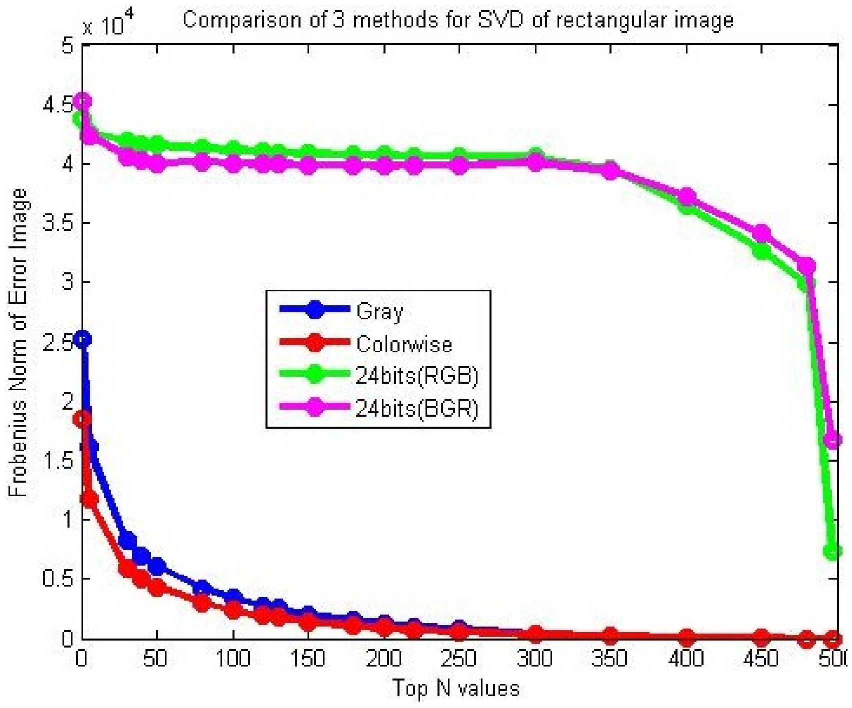


Figure 2.16: Comparison of different methods for SVD of Rectangular image

Method	Value of N	Old Size	New Size	Percentage Compression
Conversion to gray image	200	50.2 kB	48.78 kB	2.82%
Colorwise SVD	200	97.0 kB	59.1 kB	39.0%
Converting to 24bit number	497	97.0 kB	61.3 kB	36.8%

Table 2.2: Comparison of the size reduction in good reconstructed image

Observations and Inferences

- For rectangle image, performing SVD on separate channel performs the best. Converting to grayscale also is good but there is loss of color in that.
- 24 bit (BGR) performs worse than 24 bit (RGB).
- With respect to the compression of the image, the most compression is by the 24-bit method.
- With respect to reconstruction, the colorwise methods outperforms the other two.

2.2.4 Random N for each method for Rectangular image SVD

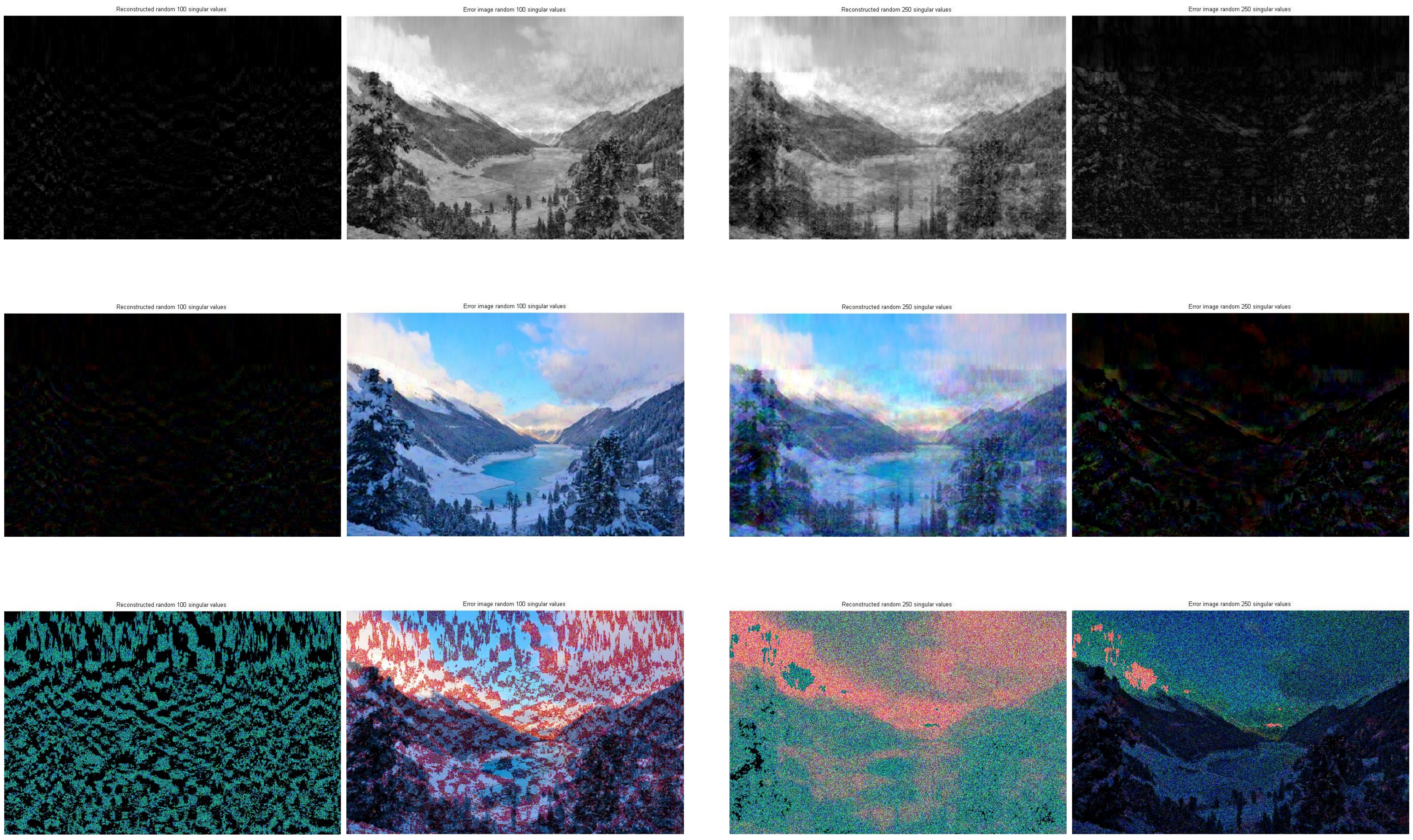


Figure 2.17: For Random N = 100 and 250 for different methods

Observations

- For random N, no specific pattern was obtained but bands and patches are visible which shows high energy eigen vectors are missing.
- The reconstruction happened at random pixels in the image which was expected since we have randomly selected the eigen values.
- As we get different random values and hence different reconstruction everytime, this is not a good way of reconstruction.

3 COMPARISON BETWEEN SVD AND EVD

3.1 SQUARE IMAGE

3.1.1 CONVERTING TO GRAY IMAGE

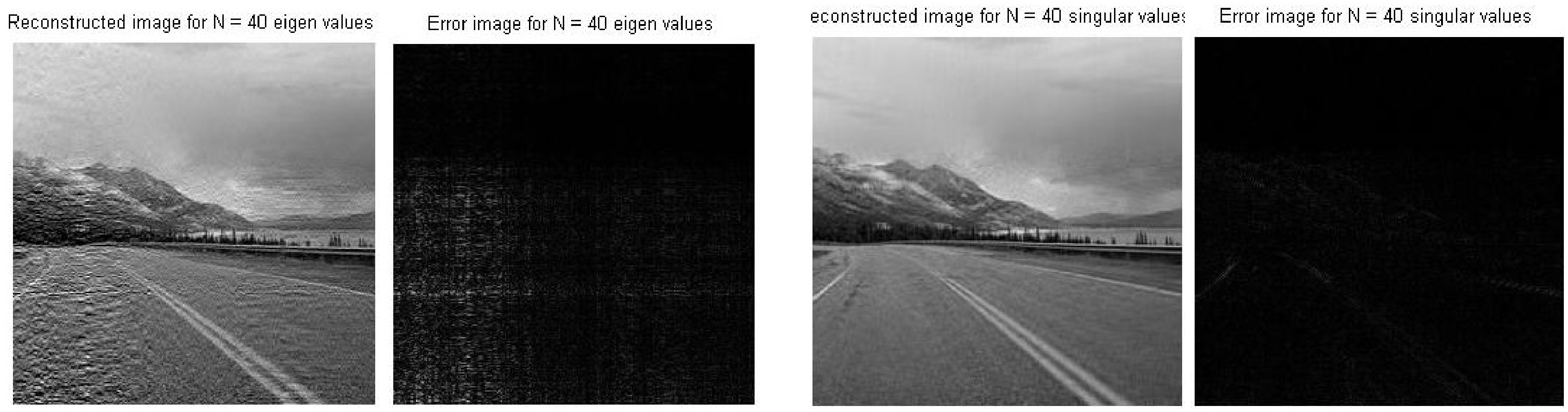


Figure 3.1: EVD (left) and SVD (right) for gray square image for N = 40

3.1.2 SEPARATING EACH COLOR BAND

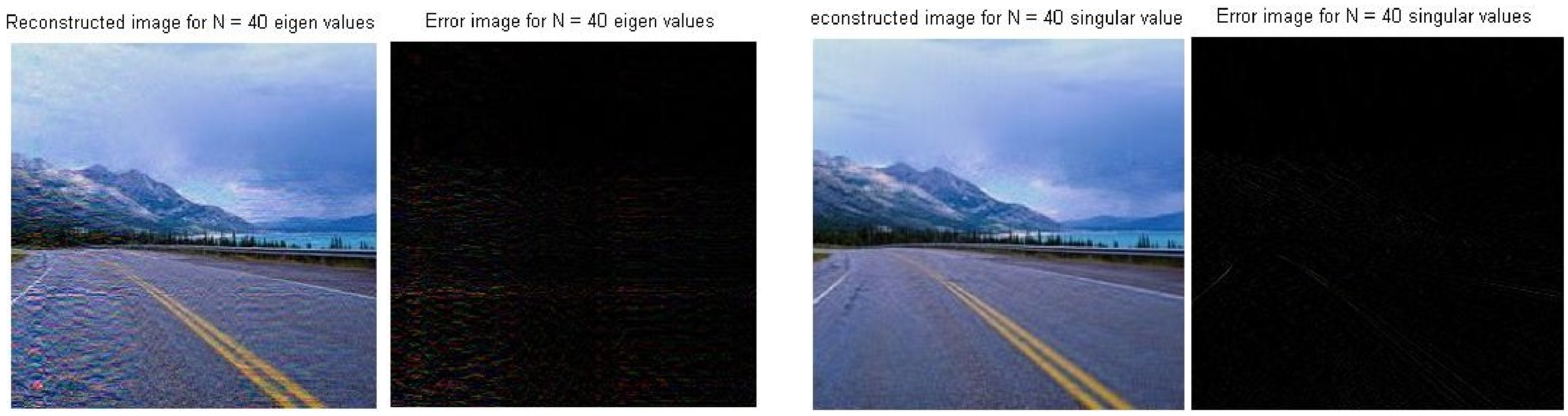


Figure 3.2: EVD (left) and SVD (right) for colorwise square image for N = 40

3.1.3 CONVERTING 8BIT R,G,B COLOR TO ONE 24 BIT NUMBER

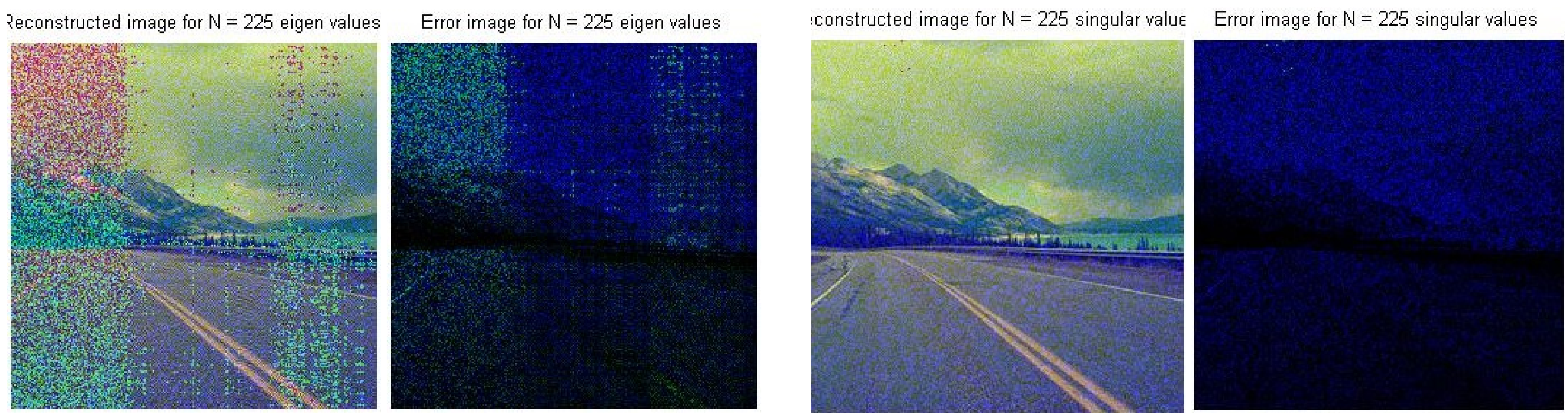


Figure 3.3: EVD (left) and SVD (right) for 24-bit square image for N = 225

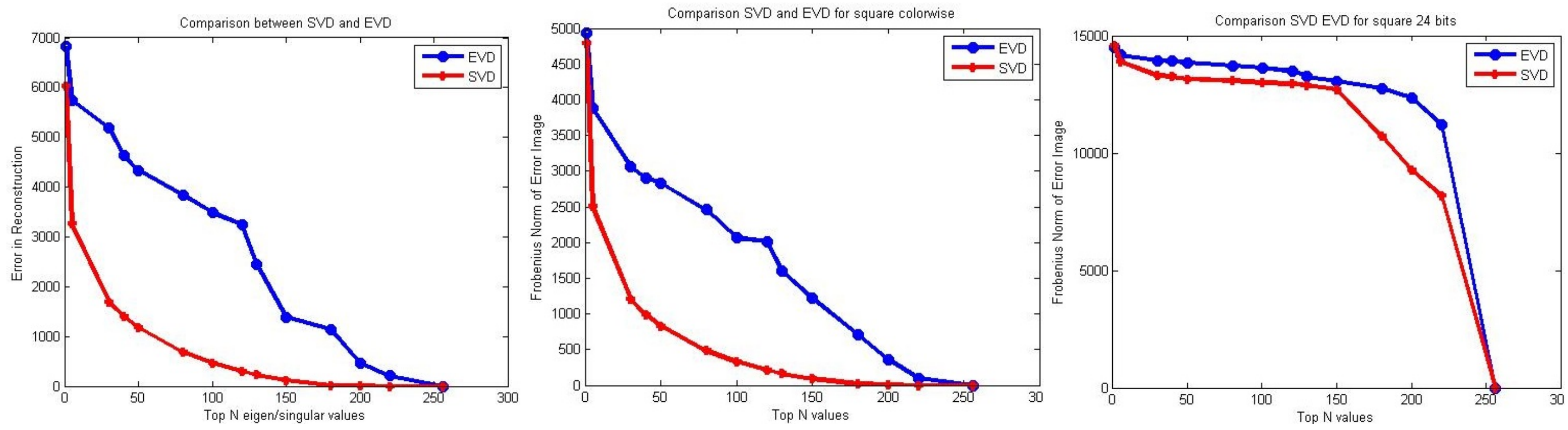


Figure 3.4: Comparison graphs for EVD and SVD for each method gray,colorwise,24bits

3.2 Rectangular Image

3.2.1 Converting to Gray Image

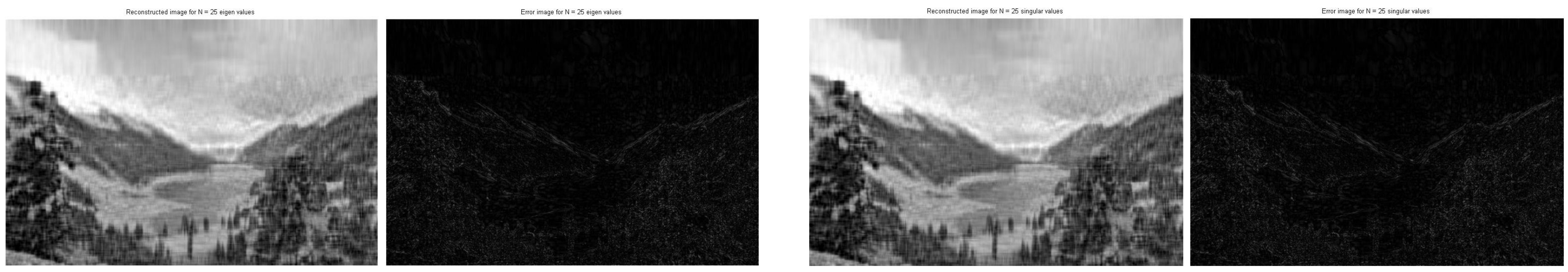


Figure 3.5: EVD (left) and SVD (right) for gray rectangular image for N = 25

3.2.2 Separating each color band

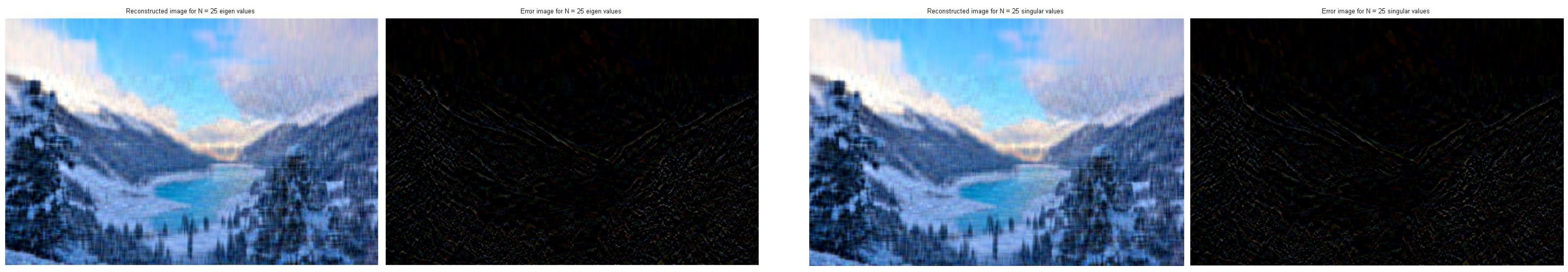


Figure 3.6: EVD (left) and SVD (right) for gray rectangular image for N = 25

3.2.3 Converting 8bit R,G,B color to one 24 bit number

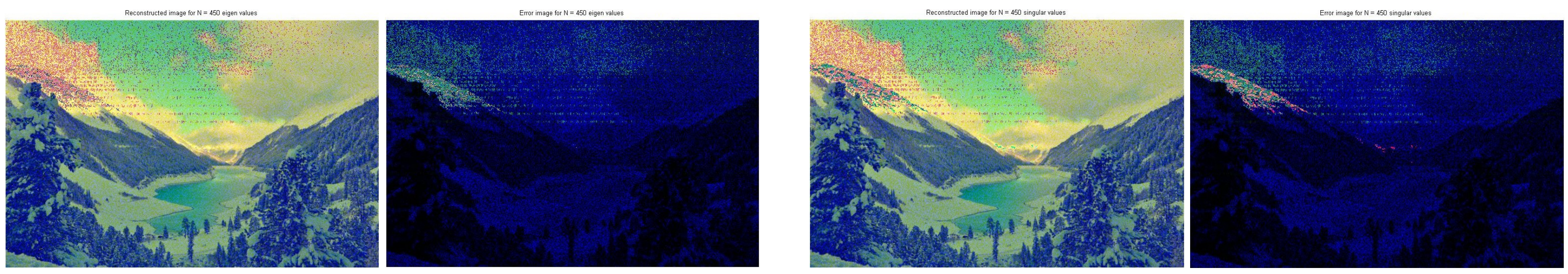


Figure 3.7: EVD (left) and SVD (right) for gray rectangular image for N = 450

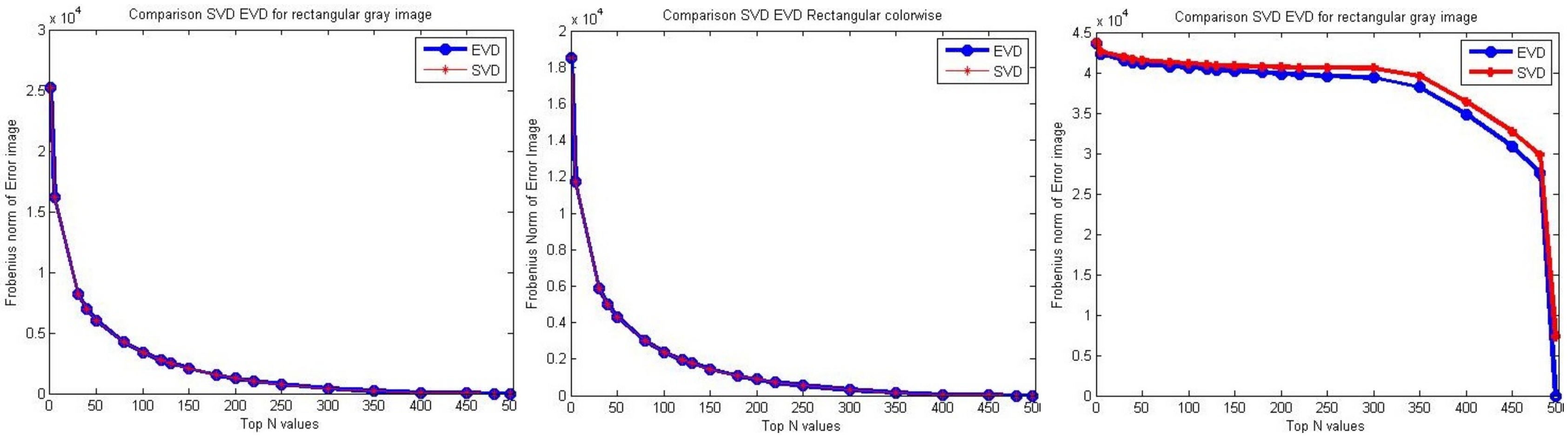


Figure 3.8: Comparison graphs for EVD and SVD for each method gray,colorwise,24bits

Method	Value of N	Old Size	New Size	Percentage Compression
EVD_square_gray	220	9.35 kB	9.18 kB	1.8%
EVD_square_colorwise	220	32.5 kB	10.5 kB	67.6%
EVD_square_24bits	256	32.5 kB	10.7 kB	66.1%
SVD_square_gray	100	9.35 kB	8.94 kB	2.03%
SVD_square_colorwise	100	32.5 kB	10.4 kB	67.8%
SVD_square_24bits	256	32.5 kB	10.7 kB	66.1%
EVD_rectangle_gray	200	50.2 kB	48.6 kB	3.12%
EVD_rectangle_colorwise	200	97 kB	58.8 kB	40.12%
EVD_rectangle_24bits	497	97 kB	57.0 kB	41.63%
SVD_rectangle_gray	200	50.2 kB	48.78 kB	2.82%
SVD_rectangle_colorwise	200	97.0 kB	59.1 kB	39.0%
SVD_rectangle_24bits	497	97.0 kB	61.3 kB	36.8%

Table 3.1: Comparison of compression of all the techniques

Observations and Inferences

- Singular values are always real and positive and eigen values can be negative and complex (complex roots existed in conjugate pairs).
- SVD performs better than EVD and best method is performing SVD on the channels separately and then concatenating.
- Eigen vectors are orthogonal for the square image but for the rectangle image, after doing A*A' , the eigen vectors are not orthogonal.
- The SVD and EVD are found to be closely related as :
  - The left-singular vectors of A are eigen vectors of AA^T
  - The right-singular vectors of A are eigen vectors of $A^T A$.
  - The non-zero singular values of A (found on the diagonal entries of Îč) are the square roots of the non-zero eigenvalues of both $A^T A$ and AA^T
- That is why when we do AA^T for performing EVD on rectangular image, SVD and EVD exactly behaves same. Their error curve overlaps.
- Computing the pseudo-inverse employ the SVD (It is formed by replacing every non-zero diagonal entry by its reciprocal and transposing the resulting matrix) that is why for rectangular images EVD and SVD are behaving similarly.

4 Linear and Non Linear Regression

4.1 Two Dimensional Data

4.1.1 About the Data

Terminology used:
X: The first dimension of the given data representing the independent variable.
T: The second dimension of the given data representing the dependent variable.
Y: Noiseless version of given data T.
$\hat{Y}$: Approximated value of Y, calculated using the polynomial derived using regression.

Below are some facts about the given data derived by visual inspection and related plots of data.

1. Data consists of 100 points in 2 Dimensions X and T, where T shares a non linear relationship with X.
2. The distribution of X is continuous uniform (almost) within the range [-4, 4].
3. The data is sorted on X dimension.
4. The distribution of T is Gaussian with μ = 6.1379 and σ^2 = 1.0342e+03.
5. Dimension Y seems to have a Gaussian noise added to it. The parameters of the noise are unknown as at this step.

Function	Value
Var(X)	5.496045
Cov(X,T)	66.109607
CorCoeff(X,T)	0.876869

Table 4.1: Statistical measures of the data

4.1.2 Objective

Given a 2 dimensional data X and noisy data T with unknown relationship between X and T. The objective is to identify a relationship (polynomial) between X and Y (Y is noiseless T) so that the relationship can closely approximate Y for any valid value of X, As the relationship derived will be a closed approximation of the actual relationship so the values predicted by the polynomial will be referred as $\hat{Y}$ which is a close approximation of unknown Y.

4.1.3 Preprocessing

Preprocessing: In the preprocessing step the data is divided into 3 parts which are used for the purpose of training, validation and testing. The division was done in the ratio 70:20:10 for training, validation and test sets. The data points in the three subsets were selected at uniform intervals from the given global data so as to maintain the underlying distributions and other properties of global data in all the three subsets because the data is sorted on X axis.

4.1.4 Simple Linear Regression

Values of Y were estimated in a way so as to have a linear relationship with the values of X. As the relationship between the two random variables was linear so the plot of X with $\hat{Y}$ consisted of a line. Weight values were calculated using:

$$\boxed{w = (\Phi'.\Phi)^{-1}.\Phi'.T}$$

where Φ is a 100 x 2 design matrix made up with first column of all ones and second column is filled up with X values.

$$\boxed{\phi = [\bar{1} \quad \bar{X}]}$$

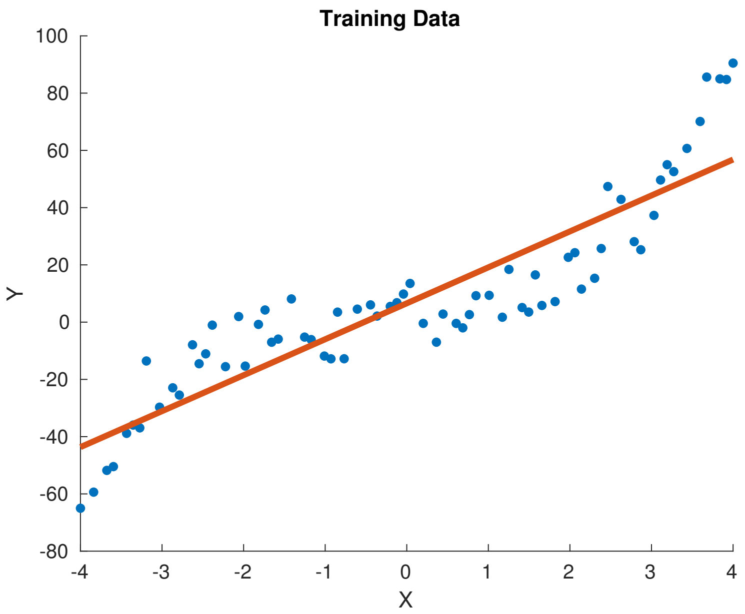
The prediction on all data sets was made using:

$$\boxed{\hat{Y} = \Phi.w}$$

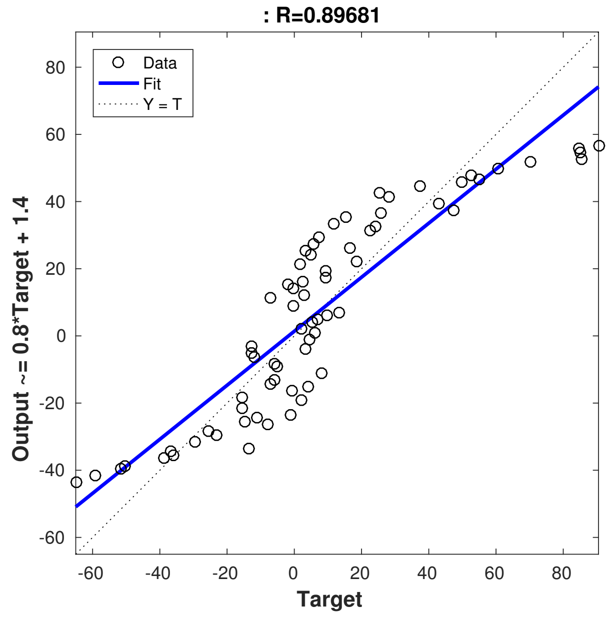
4.1.5 Simple Non Linear Regression

Values of Y were estimated in a way so as to have a non linear relationship with the values of X. As the relationship between Y and X was not linear so the plot consisted of a curve instead of a straight line. Everything was same as in simple linear regression except for the Phi which was

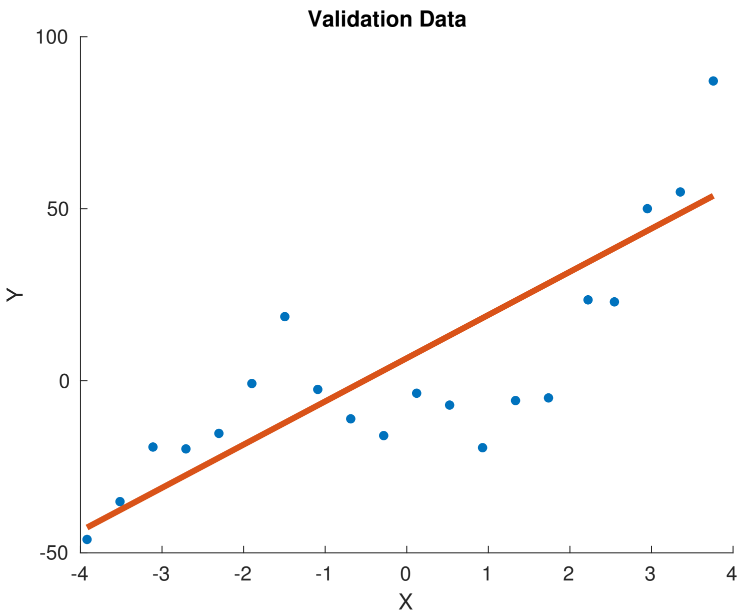
$$\boxed{\phi = [\bar{1} \quad \bar{X} \quad \bar{X^2} \quad \bar{X^3}]}$$



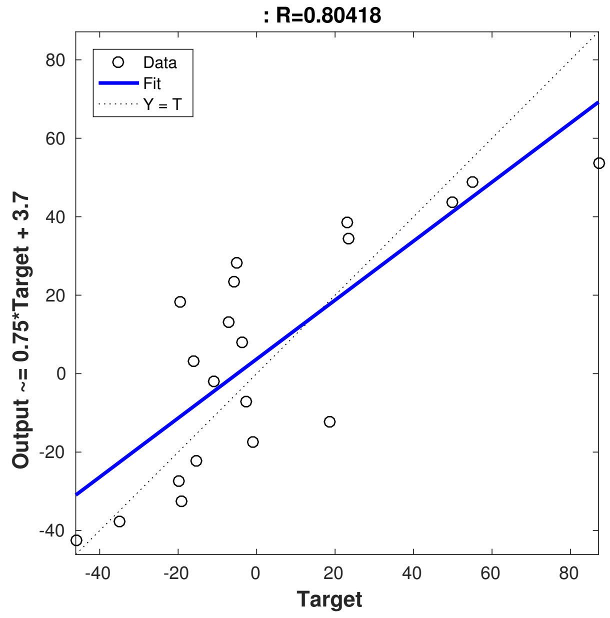
(a) Train: Actual Data and Approximated Function



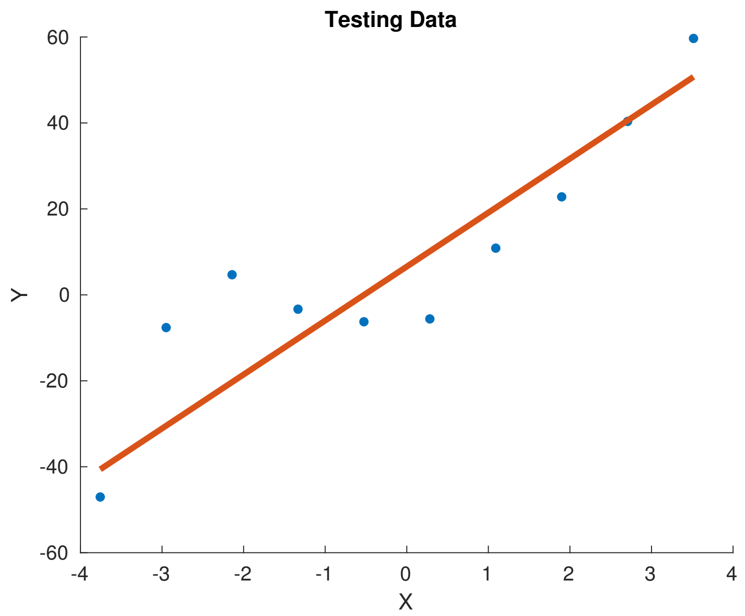
(b) Train: Scatter Plot-Target Vs Model Output



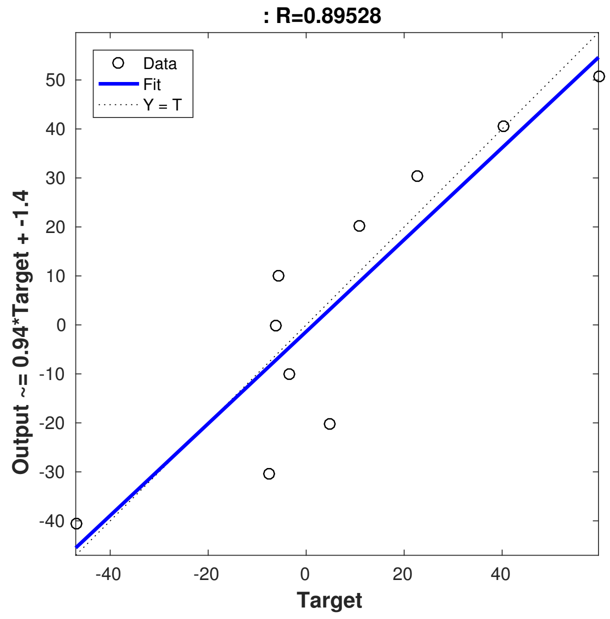
(c) Val: Actual Data and Approximated Function



(d) Val: Scatter Plot-Target Vs Model Output

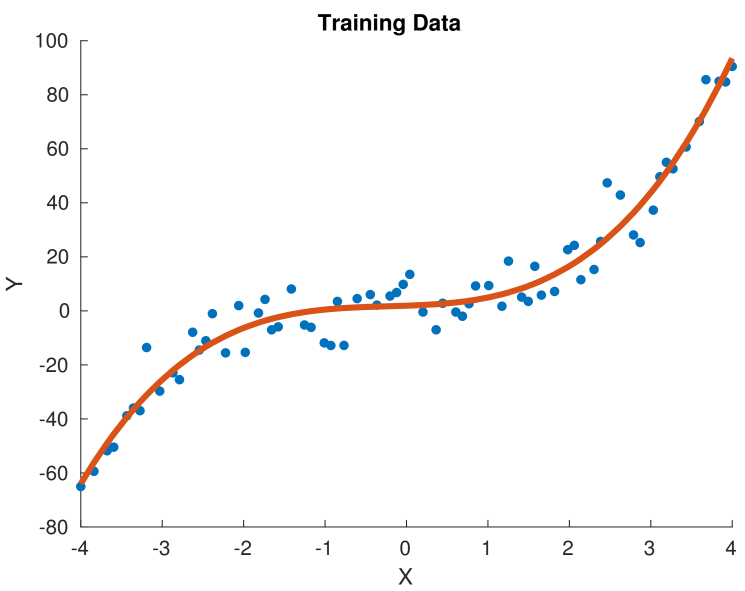


(e) Test: Actual Data and Approximated Function

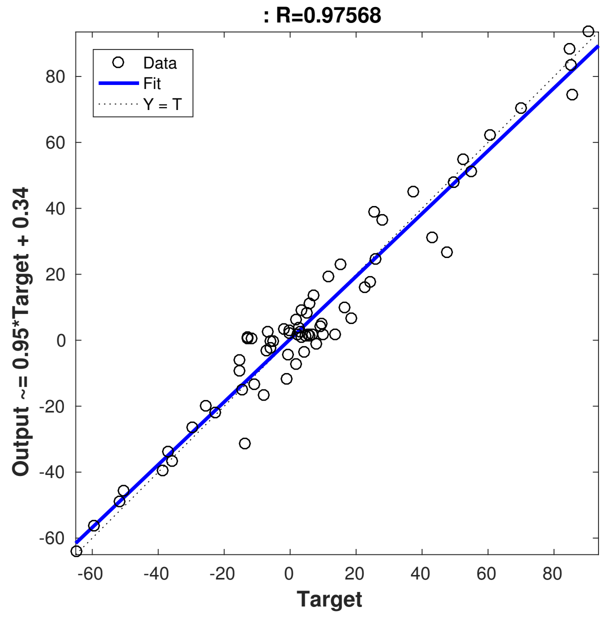


(f) Test: Scatter Plot-Target Vs Model Output

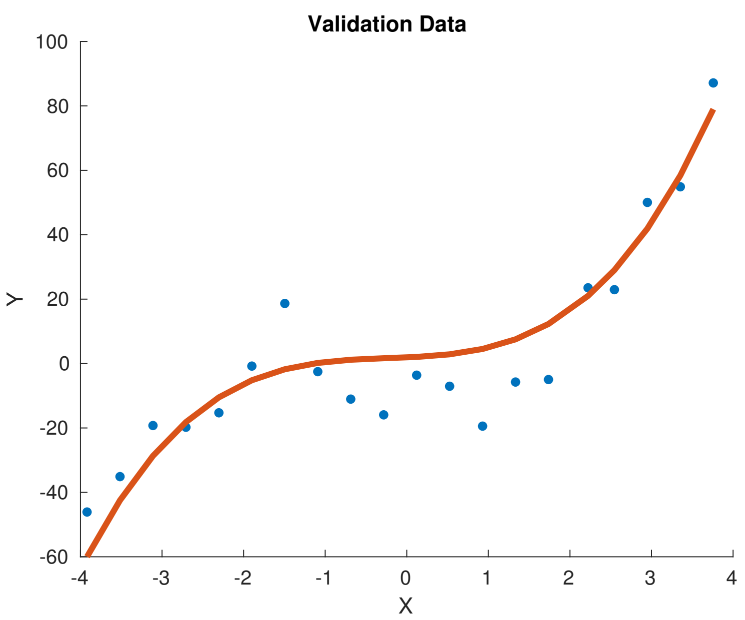
Figure 4.1: 2D Data Plots at Degree 1



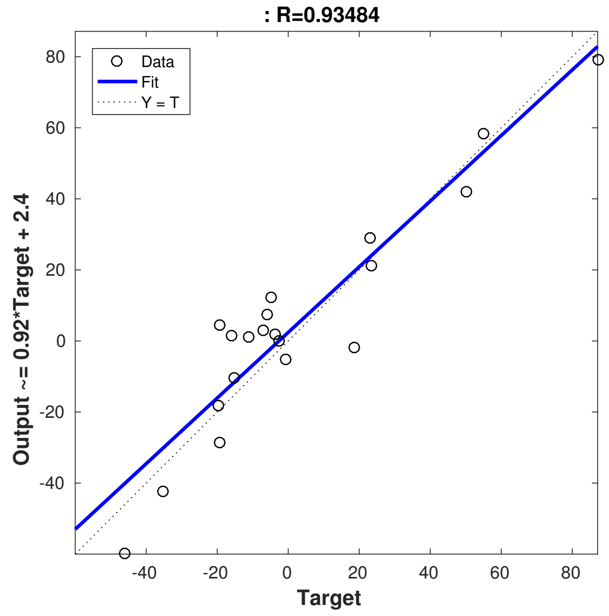
(a) Train: Actual Data and Approximated Function



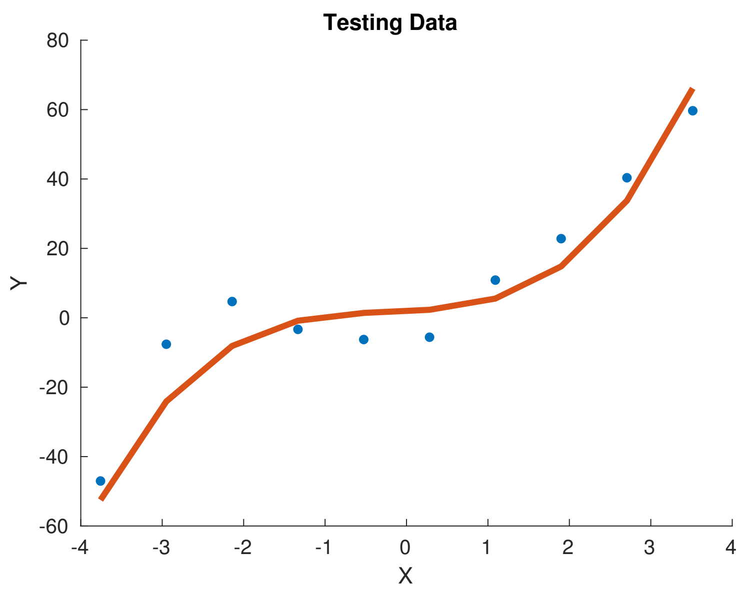
(b) Train: Scatter Plot-Target Vs Model Output



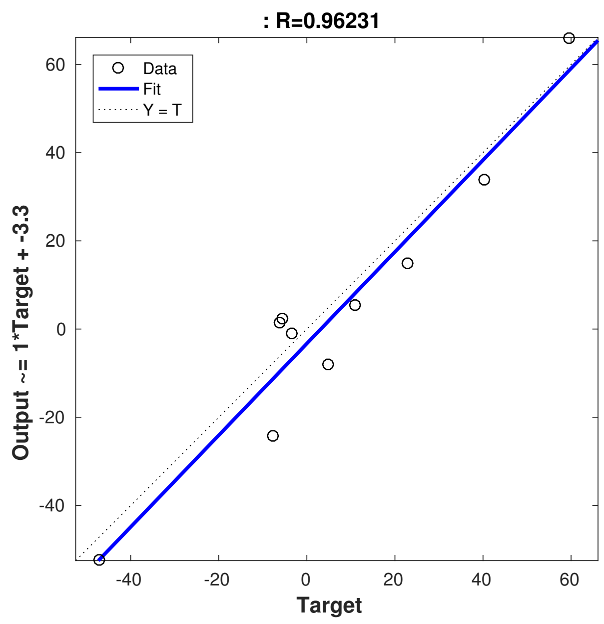
(c) VaL: Actual Data and Approximated Function



(d) VaL: Scatter Plot-Target Vs Model Output

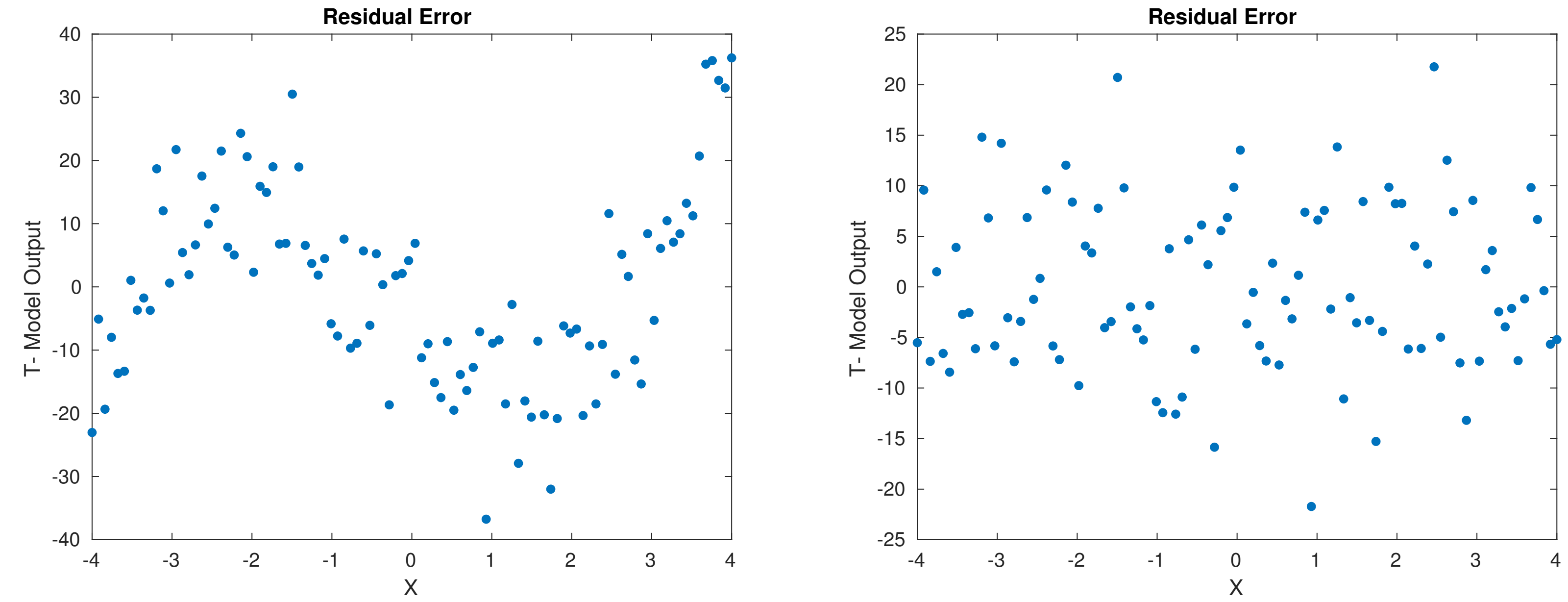


(e) Test: Actual Data and Approximated Function



(f) Test: Scatter Plot-Target Vs Model Output

Figure 4.2: 2D Data Plots at Degree 3



(a) R.Error Vs X at Degree 1 (b) R.Error Vs X at Degree 3

Figure 4.3: Residual Error at degree 1 and 3

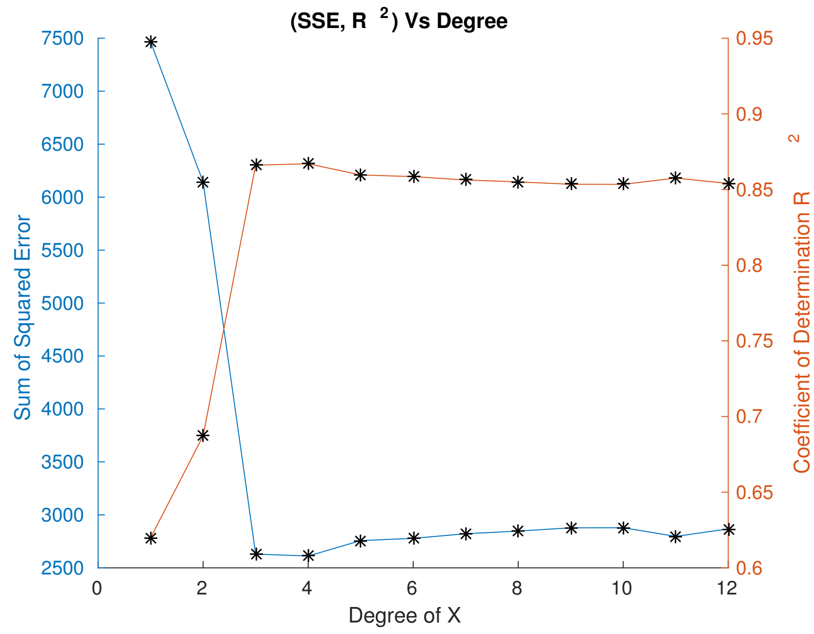


Figure 4.4: Degree Vs Error / COD

	Degree 1	Degree 3
Cov(X,$\hat{Y}$)	66.109607	66.109607
Var($\hat{Y}$)	795.204605	967.328825
CorrCoeff(X,$\hat{Y}$)	1	0.906677
Mean of R.Error	0	0
Sum of R.Error	0	0
Var of R.Error	239.006212	66.881992
Error in Training Set	14618.50	3588.27
Error in Validation Set	7466.65	2629.94
Error in Test Set	1747.62	770.40
Polynomial Obtained	$\hat{Y} = 6.5747 + 12.5573X$	$\hat{Y} = 1.9121 + 1.0642X + 0.8027X^2 + 1.1637X^3$

Table 4.2: SSE & SM on various data sets at degree 1 and 3

4.1.7 Inferences and Observations

1. A tilted Z shape was seen in the residual plot at degree 1, indicating that linear model may not be the best. The random shape seen in the residual plot at degree 3, indicates that cubic model is better.
2. Least error was obtained at degree 4. However the reduction in error was not much significant as compared to error obtained at degree 3 and so we decided to keep the model complexity to 3.
3. We desired to have Output = A * Target + B where A = 1 and B = 0. but what we got in different data sets at degree 3 is Output = 0.95 * Target + 0.34 for training, Output = 0.92 * Target + 2.4 for validation, Output = 1 * Target + -3.3 for test.
4. Mean and sum of residual error is found to be 0 in all outcomes of experiments performed.
5. Co-variance of X and $\hat{Y}$ does not change with change in degree and it is same as Co-variance of X and T.
6. Reduction in the variance of Residual error in cubic model is suggestive of the fact that the cubic model has less spread of errors which means the cubic model is better and limiting the range of error.
7. Errors obtained on Training, Validation and Test data sets clearly indicates that the cubic model is better than the linear model.
8. As the complexity of the model is increasing the coefficients associated with X are reducing in numeric value.

4.2 THREE DIMENSIONAL DATA

4.2.1 ABOUT THE DATA

Terminology used:
$\bar{X}$: The dimensions (Features) of the given data except for the last dimension. $\bar{X}$ is made up of X_1 and X_2
T: The last dimension of the given data.
Y: Noiseless version of given data T.
$\hat{Y}$: Approximated value of Y, calculated using the polynomial derived using regression.
Below are some facts about the given data derived by visual inspection and related plots of data.

1. Data consists of 500 points in 3 Dimensions X_1, X_2 and T where T shares a non linear relationship with $\bar{X}$.
2. The distribution of X_1 and X_2 are continuous uniform (almost) within the range [-4, 4].
3. X_1 and X_2 dimensions are not sorted.
4. The distribution of T is Gaussian with $\mu = 3.1581$ and $\sigma^2 = 827.3928$.
5. Dimension T seems to have a Gaussian noise added to it. The parameters of the noise are unknown as at this step.

Function	**Value**	**Function**	**Value**
Var(X1)	5.1861	Cov(X1,X2)	-0.0245
Var(X2)	5.0933	CorrCoef(X1,T)	0.8934
Cov(X1,T)	58.5197	CorrCoef(X2,T)	-0.0132
Cov(X2,T)	-0.8590	CorrCoef(X1,X2)	-0.0048

Table 4.3: Statistical measures of the 3D data

4.2.2 OBJECTIVE

Given a 3 dimensional data X_1,X_2 and noisy T with unknown relationship between $\bar{X}$ and T. The objective is to identify a relationship (polynomial) between the $\bar{X}$ and Y (Y is noiseless T) so that the relationship can closely approximate Y for any valid value of X, As the relationship derived will be a closed approximation of the actual relationship so the values predicted by the polynomial will be referred as $\hat{Y}$ which is a close approximation of unknown Y.

4.2.3 PREPROCESSING

Preprocessing: In the preprocessing step the data is divided into 3 parts which are used for the purpose of training, validation and testing. The division was done in the ratio 70:20:10 for training, validation and test sets randomly for 1000 times. The division which generated best results on validation data was used in rest of the experiments. The data points in the three subsets can be selected at uniform intervals from the given global data, however this may generate poor results as the data was not sorted on X_1 and X_2 dimensions and this is why we used multiple random generations.

4.2.4 MULTI LINEAR REGRESSION

Values of Y were estimated in a way so as to have a linear relationship with the values of $\bar{X}$. As the relationship between Y and $\bar{X}$ was linear so the plot consisted of a plane. Weight values were calculated on training set using:

$$\boxed{w = (\Phi'.\Phi)^{-1}.\Phi'.T}$$

where Φ is the design matrix and

$$\boxed{\phi = [\bar{1} \quad \bar{X}_1 \quad \bar{X}_2 \quad \bar{X}_1^2 \quad \bar{X}_2^2 \quad \bar{X}_1\bar{X}_2]}$$

The prediction was made on training validation and test set using:

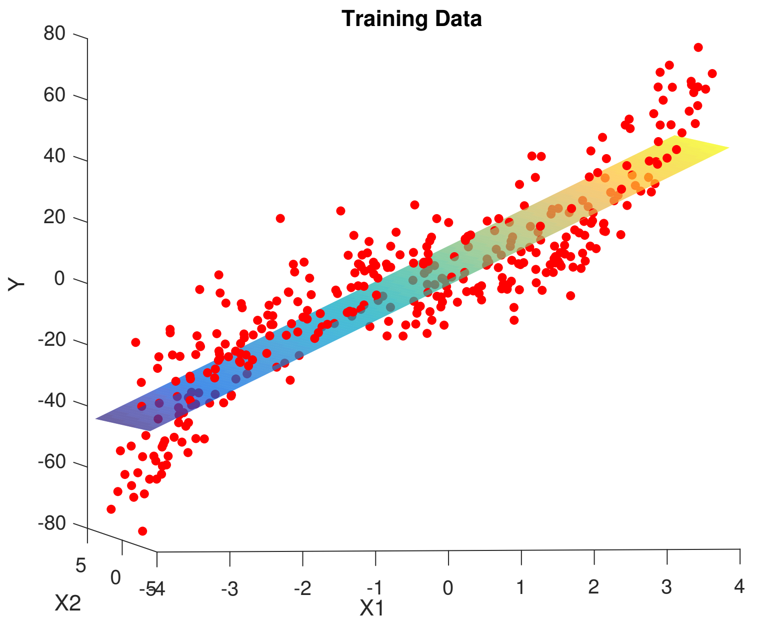
$$\boxed{\hat{Y} = \Phi.w}$$

4.2.5 MULTI NON LINEAR REGRESSION

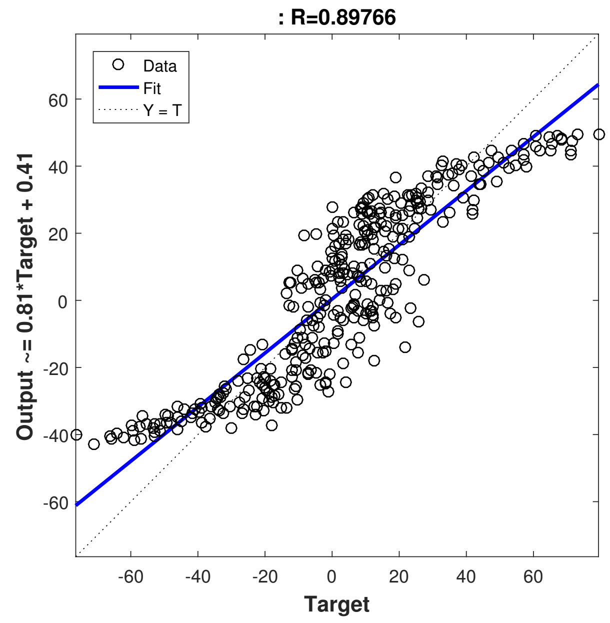
Values of Y were estimated in a way so as to have a non linear relationship with the values of $\bar{X}$. As the relationship between Y and $\bar{X}$ was not linear so the plot consisted of a plane with bends. Everything was same as in multi linear regression except for the Phi which was

$$\boxed{\phi = [\bar{1} \quad \bar{X}_1 \quad \bar{X}_2 \quad \bar{X}_1^2 \quad \bar{X}_2^2 \quad \bar{X}_1\bar{X}_2 \quad \bar{X}_1^3 \quad \bar{X}_2^3 \quad \bar{X}_1\bar{X}_2^2 \quad \bar{X}_1^2\bar{X}_2]}$$

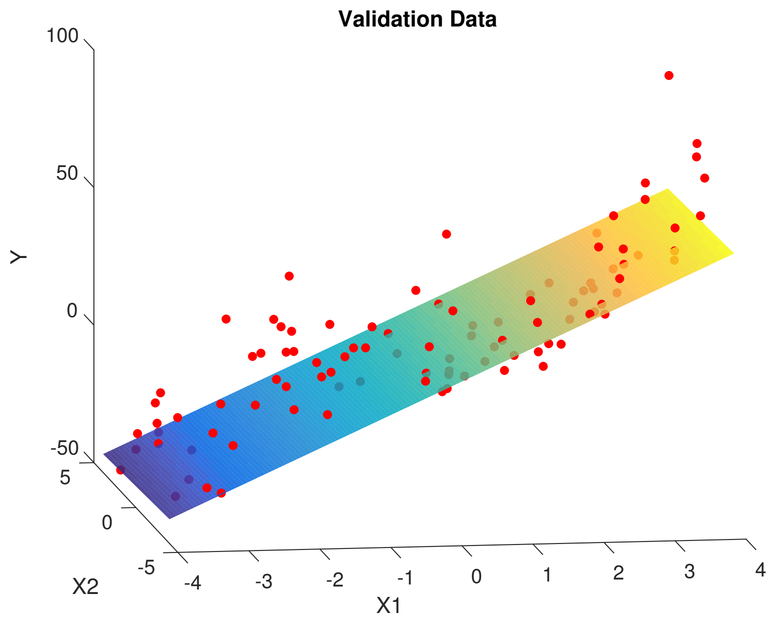
4.2.6 Plots



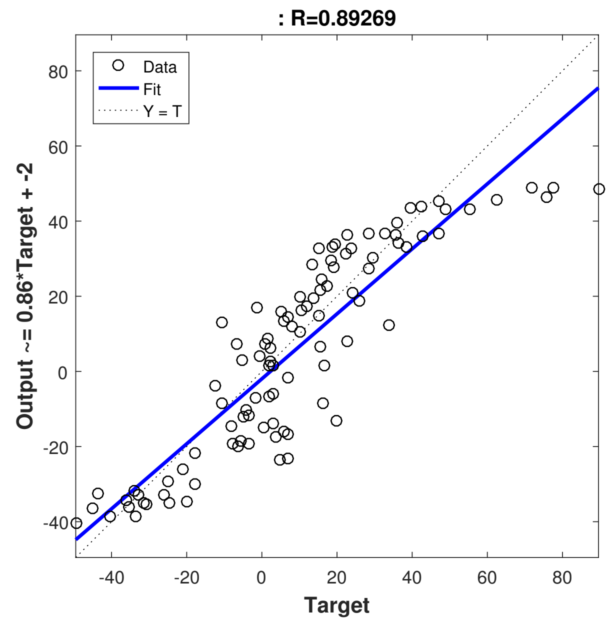
(a) Actual Data and Approximated Function



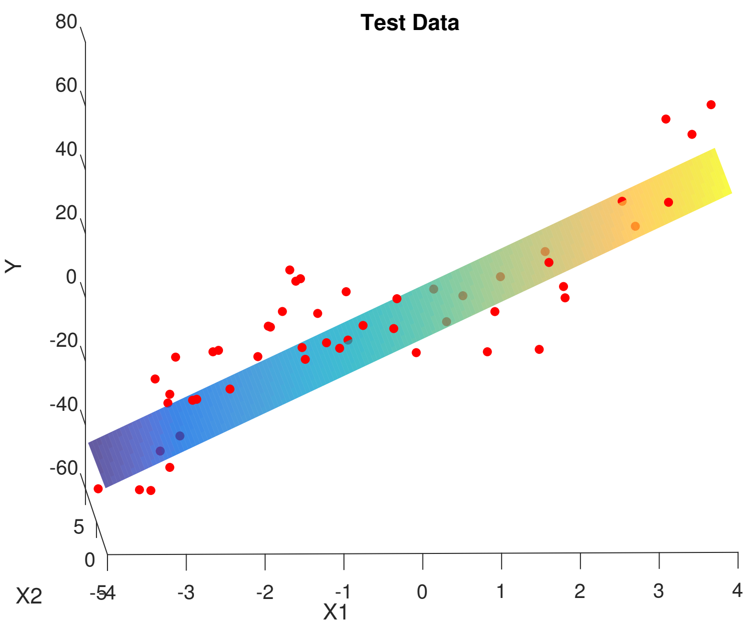
(b) Scatter Plot: Target Vs Model Output



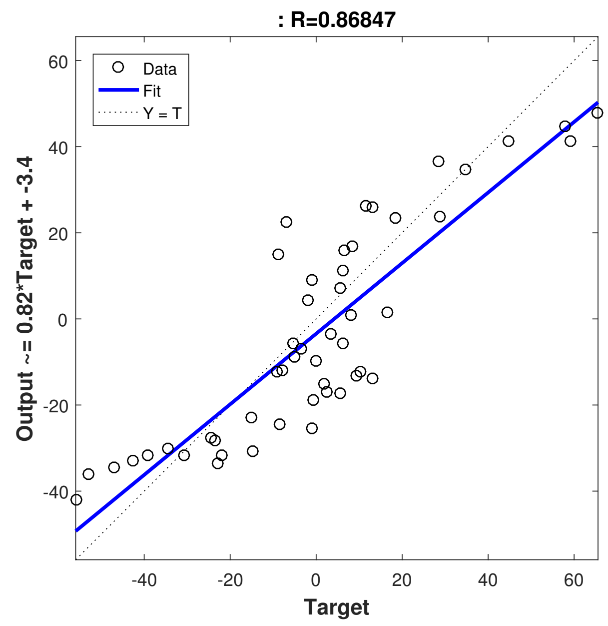
(c) Actual Data and Approximated Function



(d) Scatter Plot: Target Vs Model Output

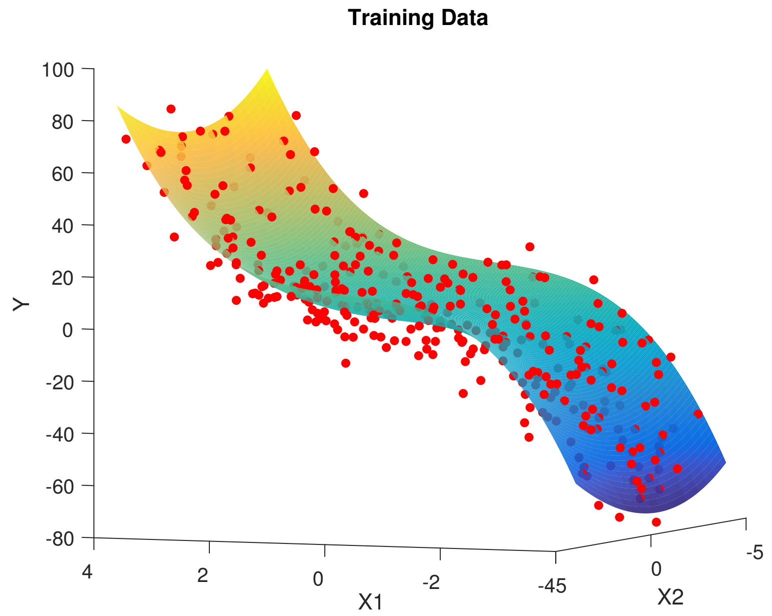


(e) Actual Data and Approximated Function

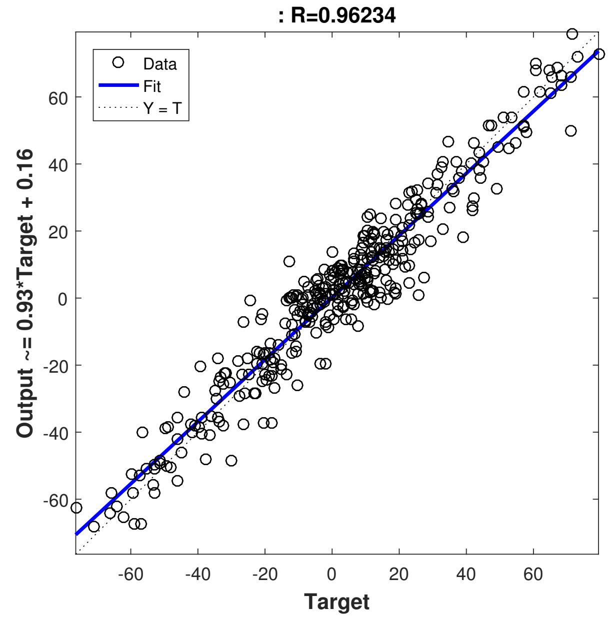


(f) Scatter Plot: Target Vs Model Output

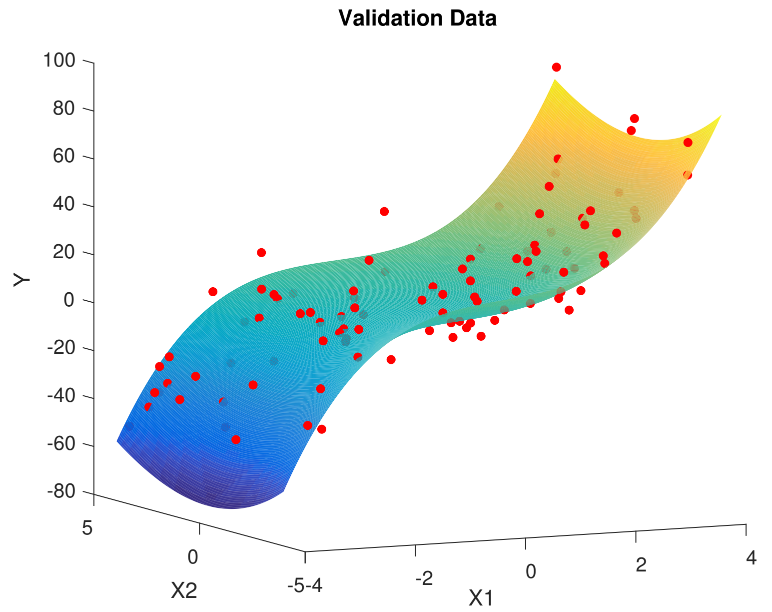
Figure 4.5: 3D Data at Degree 1



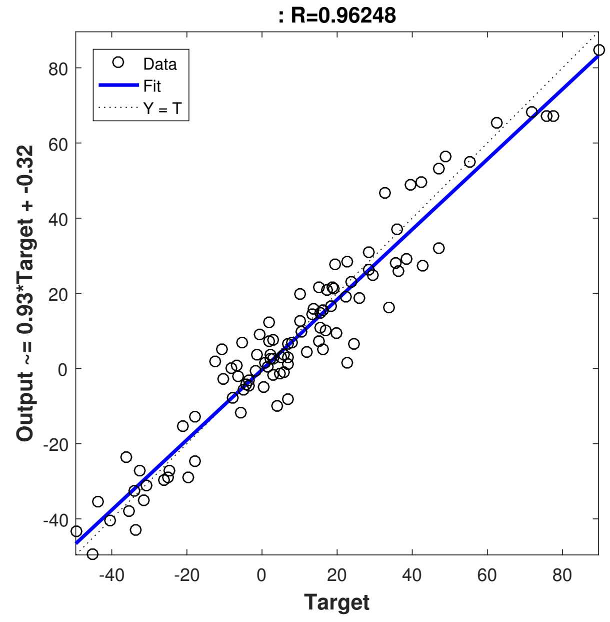
(a) Actual Data and Approximated Function



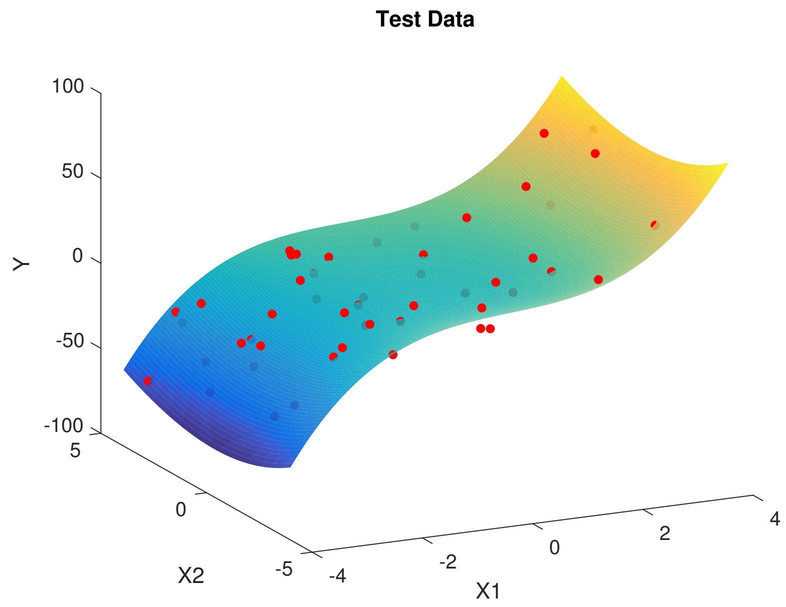
(b) Scatter Plot: Target Vs Model Output



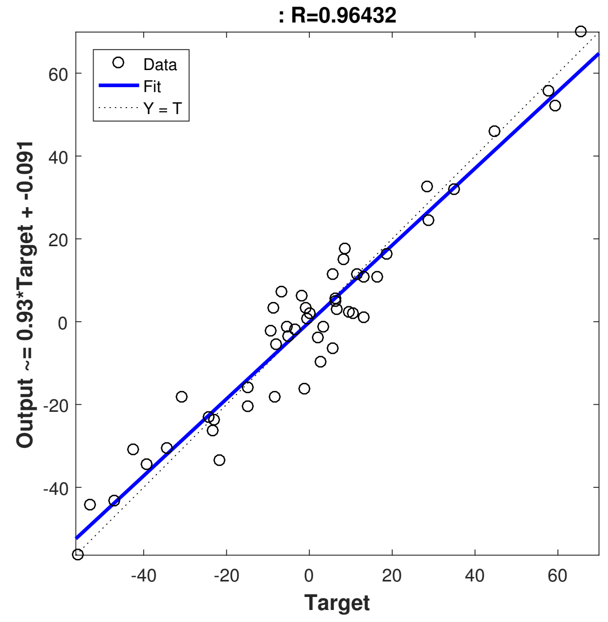
(c) Actual Data and Approximated Function



(d) Scatter Plot: Target Vs Model Output

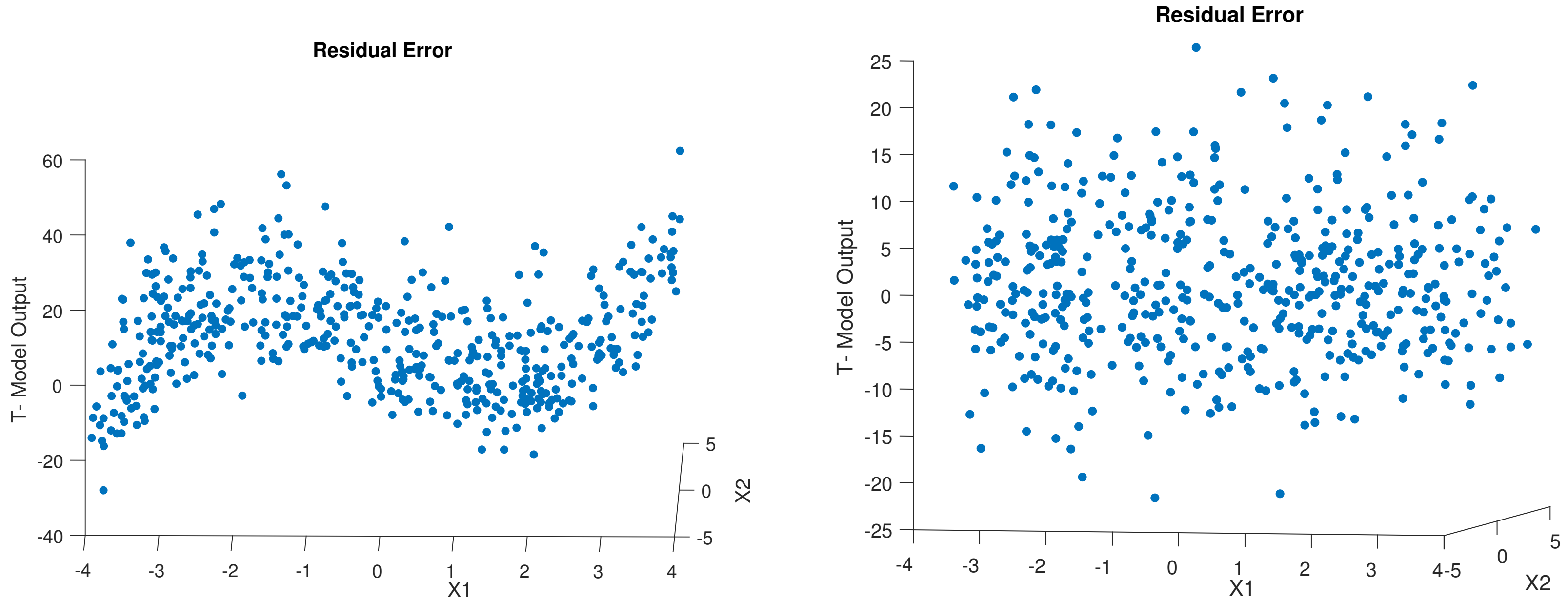


(e) Actual Data and Approximated Function



(f) Scatter Plot: Target Vs Model Output

Figure 4.6: 3D Data at Degree 3



(a) R.Error Vs X at Degree 1

(b) R.Error Vs X at Degree 3

Figure 4.7: Residual Error at Degree 1 and 3

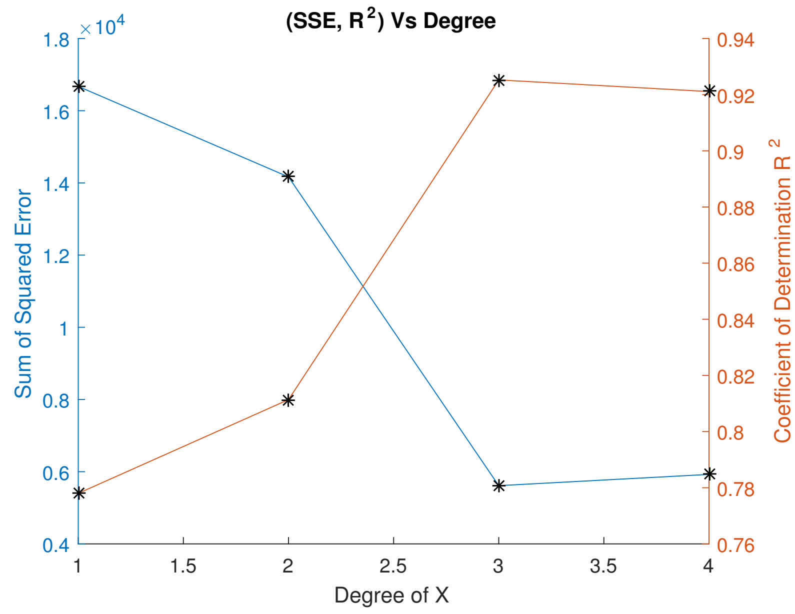


Figure 4.8: Degree Vs Error / COD

	Degree 1	Degree 3
Mean of R.Error	0	0
Sum of R.Error	0	0
Var of R.Error	166.987063	60.277834
Cov(X1, $\hat{Y}$)	58.519691	58.519691
Cov(X2, $\hat{Y}$)	-0.859010	-0.859010
Var($\hat{Y}$)	660.405702	767.114931
CorrCoeff(X1, $\hat{Y}$)	0.999949	0.927798
CorrCoeff(X2, $\hat{Y}$)	-0.014811	-0.013743
Error in Training Set	58173.044038	22139.870977
Error in Validation Set	16673.334221	5613.789308
Error in Test Set	9183.478714	2401.957234
Polynomial Obtained	$\hat{Y} = 3.77 - 11.55X_1 - 0.268X_2$	$\hat{Y} = -0.406 + 2.1008X_1 + 0.197X_2 + 0.0285X_1^2 + 1.0311X_2^2 - 0.097X_1X_2 + 0.993X_1^3 - 0.0289X_2^3 + 0.0114X_1X_2^2 + 0.0098X_1^2X_2$

Table 4.4: SSE & SM on various data sets at degree 1 and 3

4.2.7 Inferences and Observations

1. A tilted Z shape was seen in the residual plot of degree 1, indicating that linear model may not be the best. A random shape was seen in the residual plot of degree 3, indicating that cubic model is better.

2. We desired to have Output = A * Target + B where A = 1 and B = 0 but what we got in different data sets at degree 3 is Output = 0.93 * Target + 0.16 for training, Output = 0.93 * Target + -0.32 for validation, Output = 0.93 * Target + -0.091 for test.

3. Mean and sum of residual error is found to be 0 in all outcomes of experiments performed.

4. Co-variance of X_1 and $\hat{Y}$ does not change with change in degree and remains same as co-variance of X_1 and T. Similarly Co-variance of X_2 and $\hat{Y}$ does not change with change in degree and remains same as co-variance of X_2 and T.

5. Reduction in the variance of Residual error in cubic model is suggestive of the fact that the cubic model has less spread of errors which means the cubic model is better and limiting the range of error.

6. Errors obtained on Training, Validation and Test data sets clearly indicates that the cubic model is better than the linear model.

7. As the complexity of the model is increasing the coefficients associated with X are reducing in numeric value, in the polynomial obtained.

4.3 Multi Dimensional Data

4.3.1 About the Data

Terminology used:
$\bar{X}$: The dimensions (Features) of the given data except for the last dimension. $\bar{X}$ is made up of X_1, X_2.....X_m.
T: The last dimension of the given data.
Y: Noiseless version of given data T.
$\hat{Y}$: Approximated value of Y, calculated using the polynomial derived using regression.
Below are some facts about the given data derived by visual inspection and related plots of data.

1. Data consists of 1500 points in 61 Dimensions from X_1 to X_{60} and T where T is dependent on $\bar{X}$.

2. The distribution of X_1 to X_{60} are continuous uniform (almost) within [0, 1000] range.

3. Elements of $\bar{X}$ are not sorted.

4. The distribution of T is Gaussian with $\mu = 16.5045$ and $\sigma^2 = 2.1221$.

4.3.2 Objective

Given a 61 dimensional data X_1 to X_60 and T with unknown relationship between $\bar{X}$ and T. The objective is to identify a relationship (polynomial) between X and Y (Y is noiseless T) so that the relationship can closely approximate Y for any valid value of X, As the relationship derived will be a closed approximation of the actual relationship so the values predicted by the polynomial will be referred as $\hat{Y}$ which is a close approximation of unknown Y.

4.3.3 Constraints

1. Parameters should be such that the sum of the squares of the differences between T and Y should be minimum. The same should be calculated on validation data (if available)

2. Model Complexity should be as less as possible that is the degree of the polynomial that establishes relationship between $\bar{X}$ and Y should be minimum.

4.3.4 Preprocessing

Preprocessing: In the preprocessing step the data is divided into 3 parts which are used for the purpose of training, validation and testing. The division was done in the ratio 70:20:10 for training, validation and test sets randomly for 1000 times. The division which generated best results on validation data was used in rest of the experiments. The data points in the three subsets can be selected at uniform intervals from the given global data, however this may generate poor results as the data was not sorted on $\bar{X}$

4.3.5 Multi Linear Regression

Values of Y were estimated in a way so as to have a linear relationship with the values of $\bar{X}$. Weight values were calculated using:

$$\boxed{w = (\Phi'.\Phi)^{-1}.\Phi'.T}$$

where Φ is a 1500 x 61 design matrix made up with first column of all ones followed by $X_1 to X_{60}$ from column 2 to column 61. The prediction was made using:

$$\boxed{\hat{Y} = \Phi.w}$$

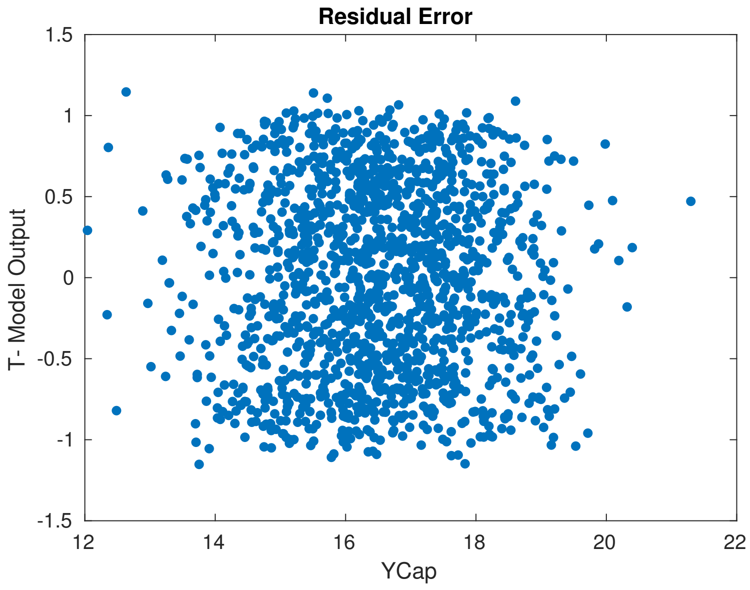
4.3.6 Multi Non Linear Regression

Transformations were applied on the data to establish a non linear relationship between $\bar{X}$ and T. The transformations were as follows:

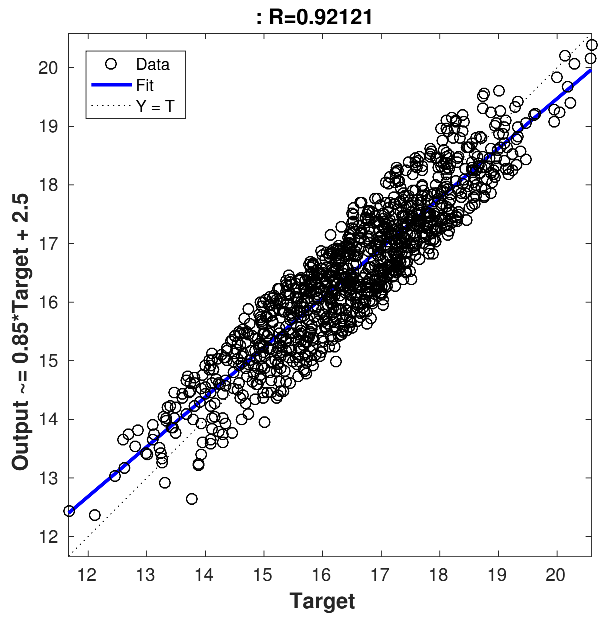
p3cm|p3cm|p3cm| height Transformation Value of
Exponential Model T = $log_{10}(T)$ $\hat{Y}$ = $10^{\hat{Y}}$
Quadratic Model T = $\sqrt{T}$ $\hat{Y}$ = $\hat{Y}^2$
Reciprocal Model T = 1 / T $\hat{Y}$ = $1/\hat{Y}$
Logarithmic Model X = $log_{10}(X)$ in input X = $log_{10}(X)$ in Phi
Power Model T = $log_{10}(T)$, X = $log_{10}(X)$ in input $\hat{Y}$ = $10^{\hat{Y}}$, X = $log_{10}(X)$ in Phi

Table 4.5: Sum of Squared Errors with different data division strategies

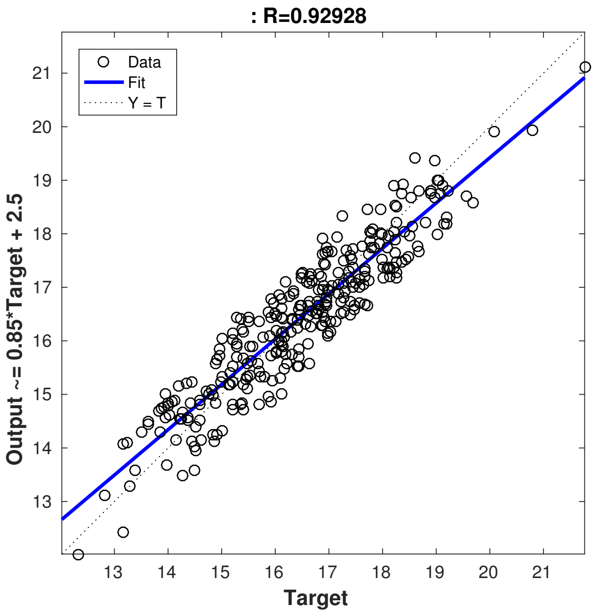
4.3.7 Plots



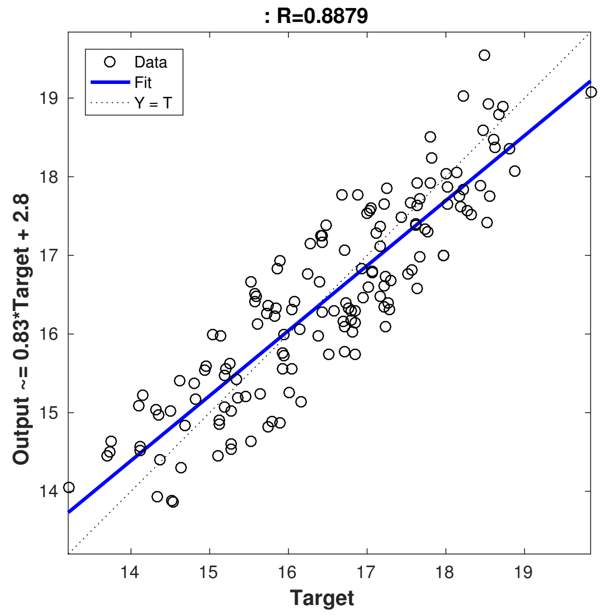
(a) Residual Error



(b) Training Data



(c) Validation Data



(d) Test Data

Figure 4.9: Scatter Plots: Target Vs Model Output & Residual Error Plot

Table 4.6: Results on M-D Data Set

Table 4.7: SSE & SM on various data sets at degree 1

	Degree 1
Mean of R.Error	0
Sum of R.Error	0
Var of R.Error	0.320196
Var($\hat{Y}$)	1.801952
Training Error	332.181863
Validation Error	97.932217
Test Error	58.703682

Table 4.8: Transformation Results on Validation Data Set

Transformation	Coefficient of Determination
Standard Linear Regression	0.92928
Exponential Model	0.92846
Quadratic Model	0.92936
Reciprocal Model	0.92313
Logarithmic Model	0.81765
Power Model	0.82203

4.3.8 Polynomial Obtained

$$
\begin{aligned}
\hat{Y} = {} & 1.16857654453193 + 0.000395339096877153X_{1} + 0.000837607778234081X_{2} + 0.000734721718402832X_{3} + 0.000244560693212635X_{4} \\
& + 0.000629972177575423X_{5} + 0.000968136888154934X_{6} + 1.84288701009953e\text{-}05X_{7} + 0.000800562870478720X_{8} + 0.000894481706280926X_{9} \\
& + 0.000776229460202362X_{10} + 0.000848395705598894X_{11} + 0.000105398267526311X_{12} + 0.000491194581704745X_{13} + 0.000884310394790405X_{14} \\
& + 0.000346567998069323X_{15} + 0.000612216960689304X_{16} + 0.000693906708662689X_{17} + 0.000829942894616016X_{18} + 9.84922478606350e\text{-}06X_{19} \\
& + 0.000524500019936386X_{20} + 0.000900315949730675X_{21} + 0.000268024067215296X_{22} + 0.000151561415980286X_{23} + 0.000140655368862916X_{24} \\
& + 5.33389720279987e\text{-}05X_{25} + 0.000569779630422640X_{26} + 2.83883855605223e\text{-}05X_{27} + 0.000788611074300828X_{28} + 0.000109640933875909X_{29} \\
& + 0.00103610720672667X_{30} + 0.000383356943847576X_{31} + 0.000649105794593866X_{32} + 0.000189168728847915X_{33} + 0.000141533434606643X_{34} \\
& + 5.79405730637930e\text{-}05X_{35} + 0.000382758509983242X_{36} + 0.000838118731910503X_{37} + 0.000472279515519067X_{38} + 0.000966007148596139X_{39} \\
& + 0.000990417445236907X_{40} + 0.000602015134978149X_{41} + 0.000173605510107681X_{42} + -0.000135657106833344X_{43} + 0.000144081259601460X_{44} \\
& + 0.000488661487489951X_{45} + 0.000309068521415047X_{46} + 0.000976589869477464X_{47} + 0.000807813561068018X_{48} + 0.000366879609879799X_{49} \\
& + 0.000884833027976732X_{50} + 0.000913011116448139X_{51} + 0.000639448737174116X_{52} + 0.000753392475361519X_{53} + 0.000806536876653471X_{54} \\
& + 0.000754998064867540X_{55} + 0.000343852873347084X_{56} + 0.000447908842199054X_{57} + 2.51126167495331e\text{-}05X_{58} + 2.52980399411891e\text{-}05X_{59} \\
& + 0.000497088167957866X_{60}
\end{aligned}
$$

4.3.9 Inferences and Observations

1. Central cluster was seen on residual plot of degree 1, indicating that a linear model may be good for this case.

2. We desired to have Output = A * Target + B where A = 1 and B = 0 but what we got in different data sets at degree 1 is Output = 0.85 * Target + 2.5 for training, Output = 0.85 * Target + 2.5 for validation, Output = 0.83 * Target + 2.8 for test.

3. Mean and sum of residual error is found to be 0.

4. As per the polynomial obtained X_0 dominate in the prediction of Y. The coefficients of all other X are very small.

5. Due to total number of dimensions being very high, it was not possible to manually make a model of higher degree. However other models were used to establish a non linear relationship between $\bar{X}$ and Y such as Exponential model, Quadratic model and so on. It was found that Quadratic model gave the highest coefficient of determination, better than the standard linear regression. However the improvement in accuracy was not much significant and to keep the model complexity less, we decided to go with the Standard linear regression.