Image Compression based on Non-Parametric Sampling in Noisy Environments

Kishan Narotam & Nitesh Nana   School of Electrical & Information Engineering, University of the Witwatersrand

Objectives

To develop a robust scheme for image compression involving:

- Creating holes in an image
- Encode and transmit the image
- Simulate a noisy channel in order to introduce errors
- Identifying and filling the holes appropriately.

System Design

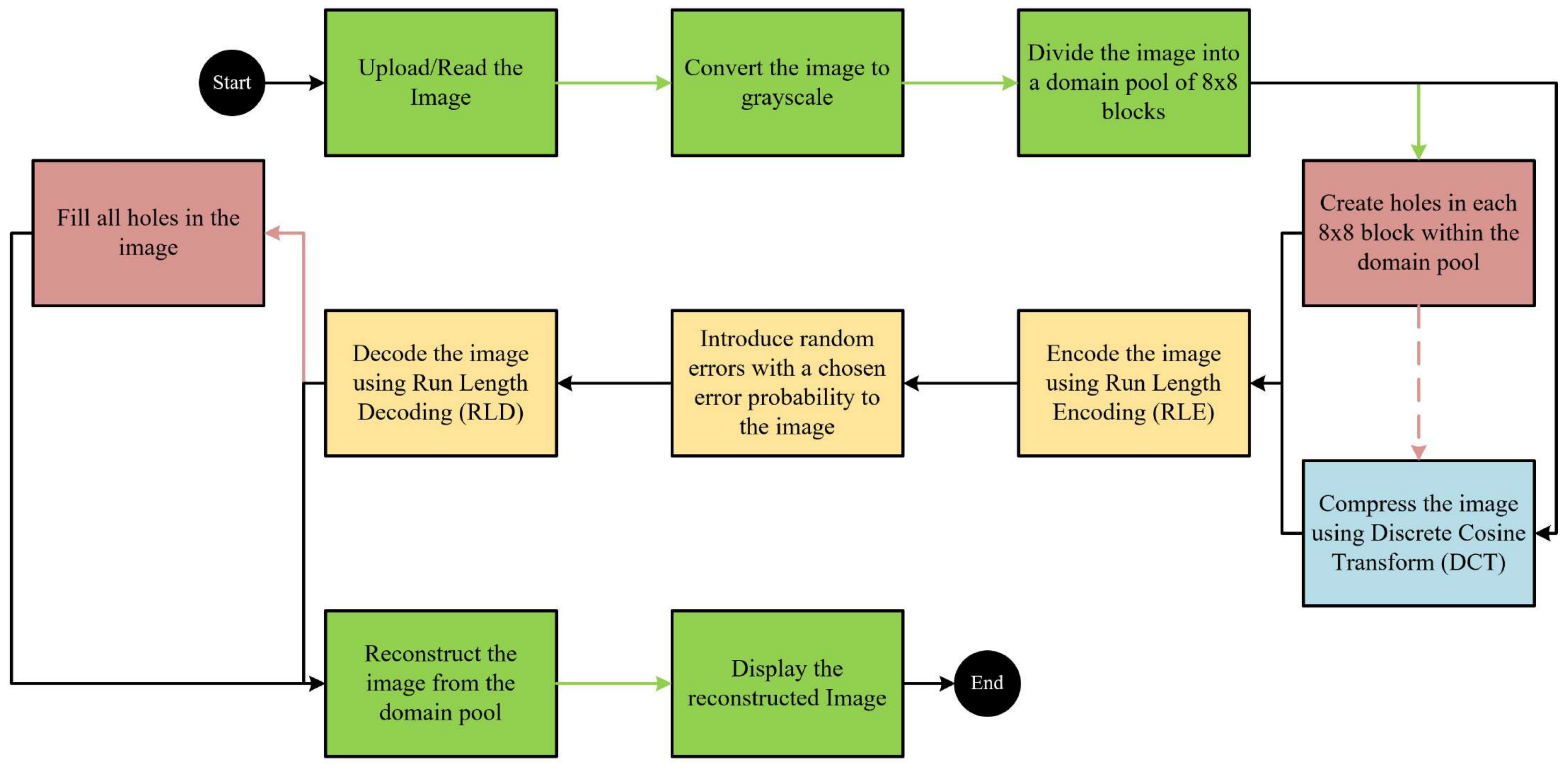


Figure 1: Image compression implementation overview

Algorithms

Algorithm 1 High-level algorithm for creating holes in `8x8` blocks

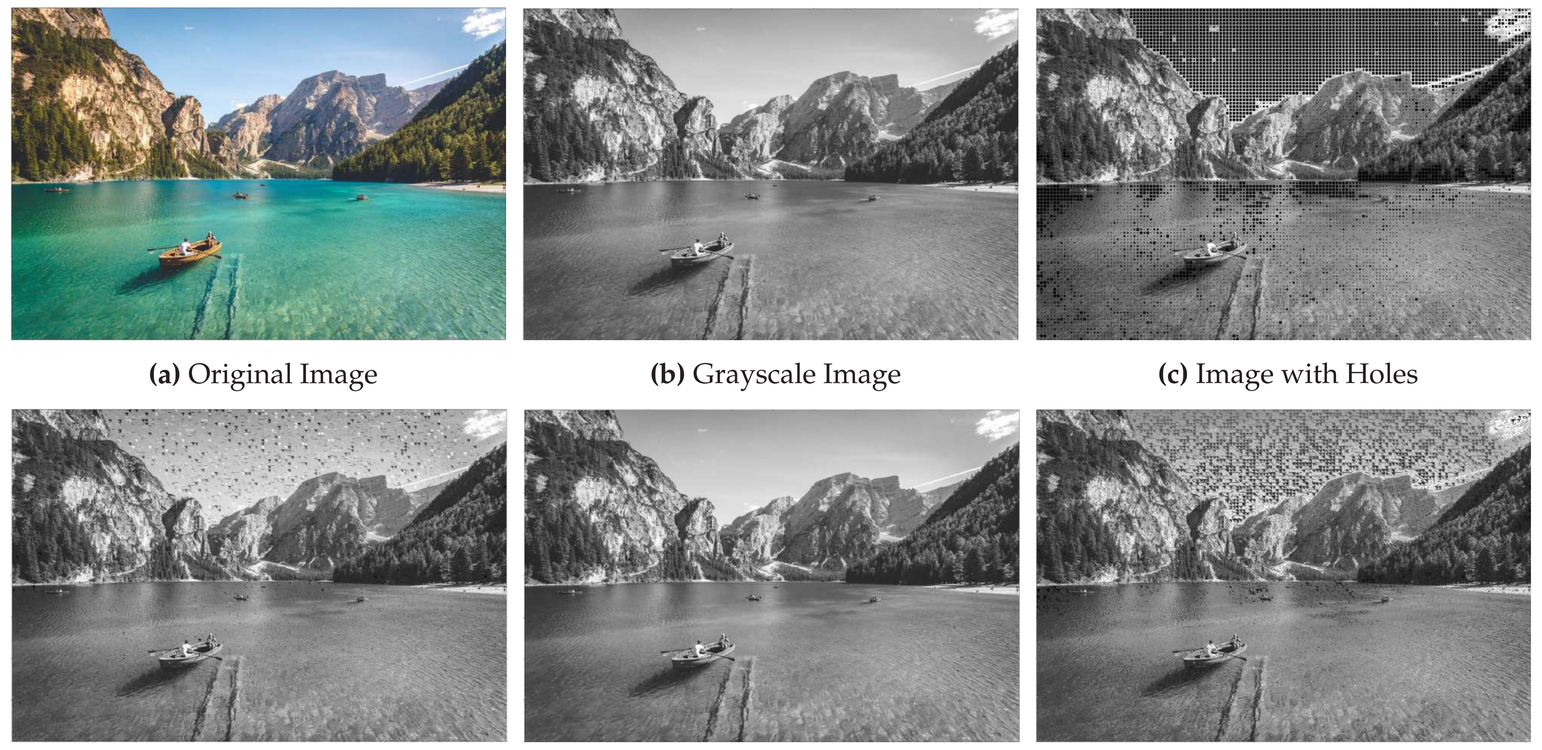
for every block in the domain pool **do**
  Set n=2
  Go to `nxn` square in `8x8` block
  Calculate average of pixels in `nxn`
  for every pixel in the `nxn` block **do**
    Check Chebyshev distance between pixels `(y[])` and average value `(x)`
  end for
  if Chebyshev distance between `x` and each value of `y[]` < 6 **then**
    Repeat for n=4
    if Chebyshev distance between `x` and each value of `y[]` < 6 **then**
      Repeat for n=6
      if Chebyshev distance between `x` and each value of `y[]` < 6 **then**
        Create hole in `6x6`
      else
        Create hole in `4x4`
      end if
    else
      Create hole in `2x2`
    end if
  else
    Move to next block in domain pool
  end if
end for

Algorithm 2 High-level algorithm for filling holes in `8x8` blocks

Set n=2
Go to `nxn` square in `8x8` block
Calculate average of pixels in `nxn`
for every pixel in the `nxn` block **do**
  Check Chebyshev distance between pixels `(y[])` and average value `(x)`
end for
if Chebyshev distance between `x` and each value of `y[]` < 6 **then**
  Repeat for n=4
  if Chebyshev distance between `x` and each value of `y[]` < 6 **then**
    Repeat for n=6
    if Chebyshev distance between `x` and each value of `y[]` < 6 **then**
      Move to top left of nxn block
      Calculate average of pixels directly:above, left & above-left. Fill pixel with average value
      Proceed from left to right and top to bottom
    else
      Fill hole in `4x4`
    end if
  else
    Fill hole in `2x2`
  end if
else
  Move to next block in domain pool
end if

Results: Processing

The figures below and to the right detail the various steps involved in the compression of the image such as: converting to grayscale; creating holes; applying the DCT transformation; and the effect of noise in the final reconstruction.

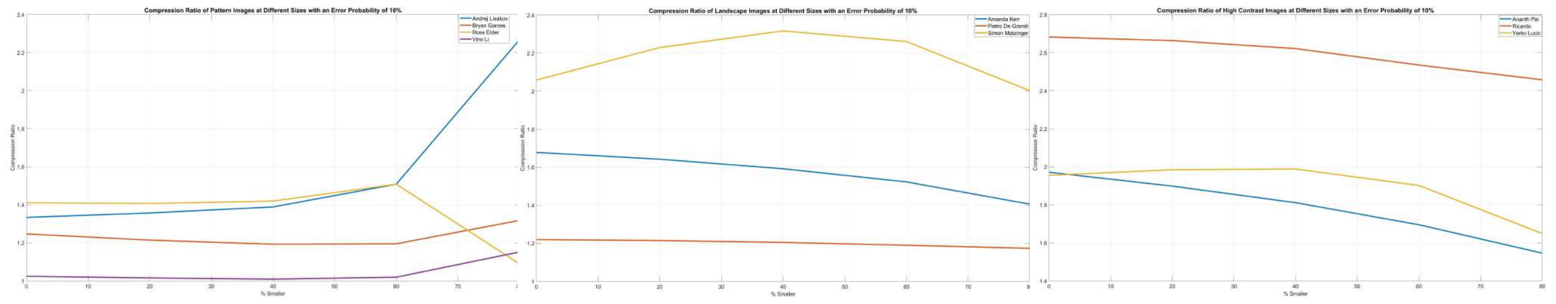


(a) Original Image (b) Grayscale Image (c) Image with Holes

(a) Bit Flip Error (50%) (b) No Errors (c) 1's Comp. Error (50%)

Results: Simulation

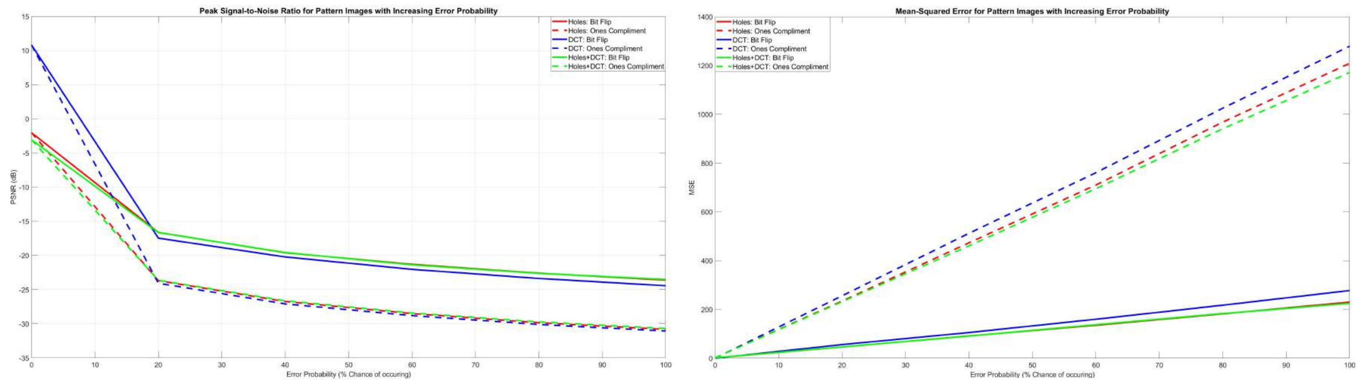
Figure 4 and the subsequent sub-figures test the created *Holes* algorithm on different images. Pattern images are chosen for the repetitive colour and smooth features; landscape images are chosen for the texture features and intense detail; high contrast images are chosen for its combination of both repetitive colour, smooth and texture features.



(a) Pattern Images (b) Landscape Images (c) High Contrast Images

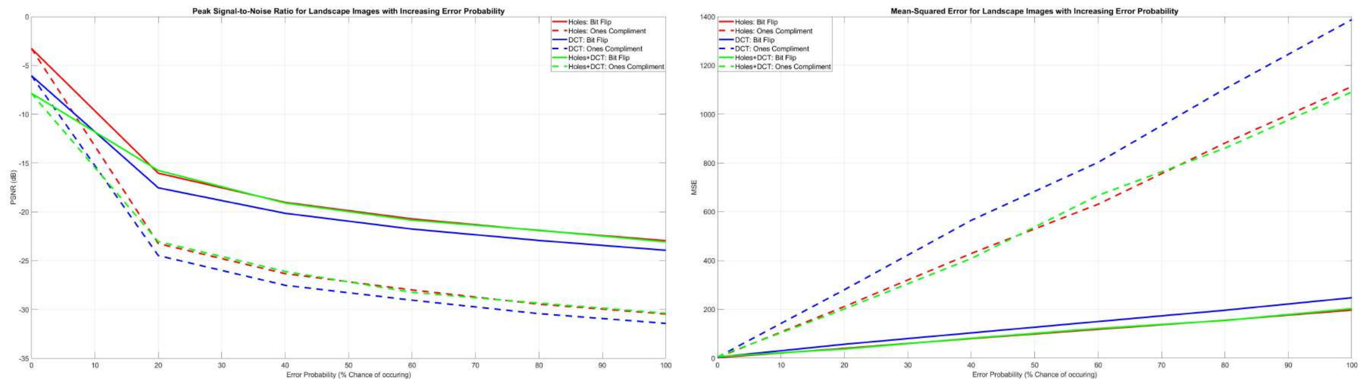
Figure 4: Compression Ratio vs Same Image at Different Sizes

An error analysis is carried out on the different image types calculating the peak signal-to-noise ratio (PSNR) and mean squared error (MSE) of the reconstructed images. The PSNR is a dimensionless number expressed on a logarithmic decibel scale, to identify the perceived errors noticeable by the human vision. The MSE is the cumulative squared error of the compressed image against the original image [1].



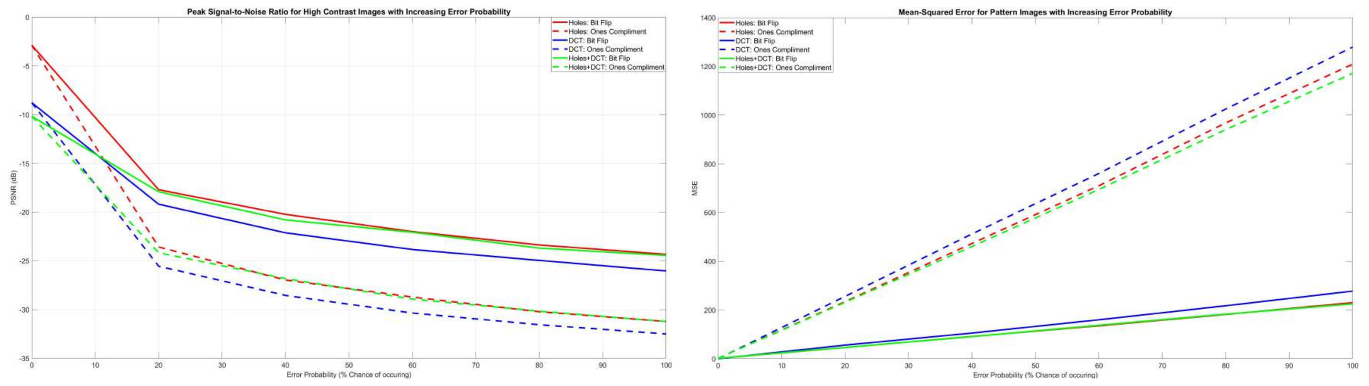
(a) PSNR for all Pattern Images (b) MSE for all Pattern Images

Figure 5: Pattern Images Error Analysis



(a) PSNR for all Landscape Images (b) MSE for all Landscape Images

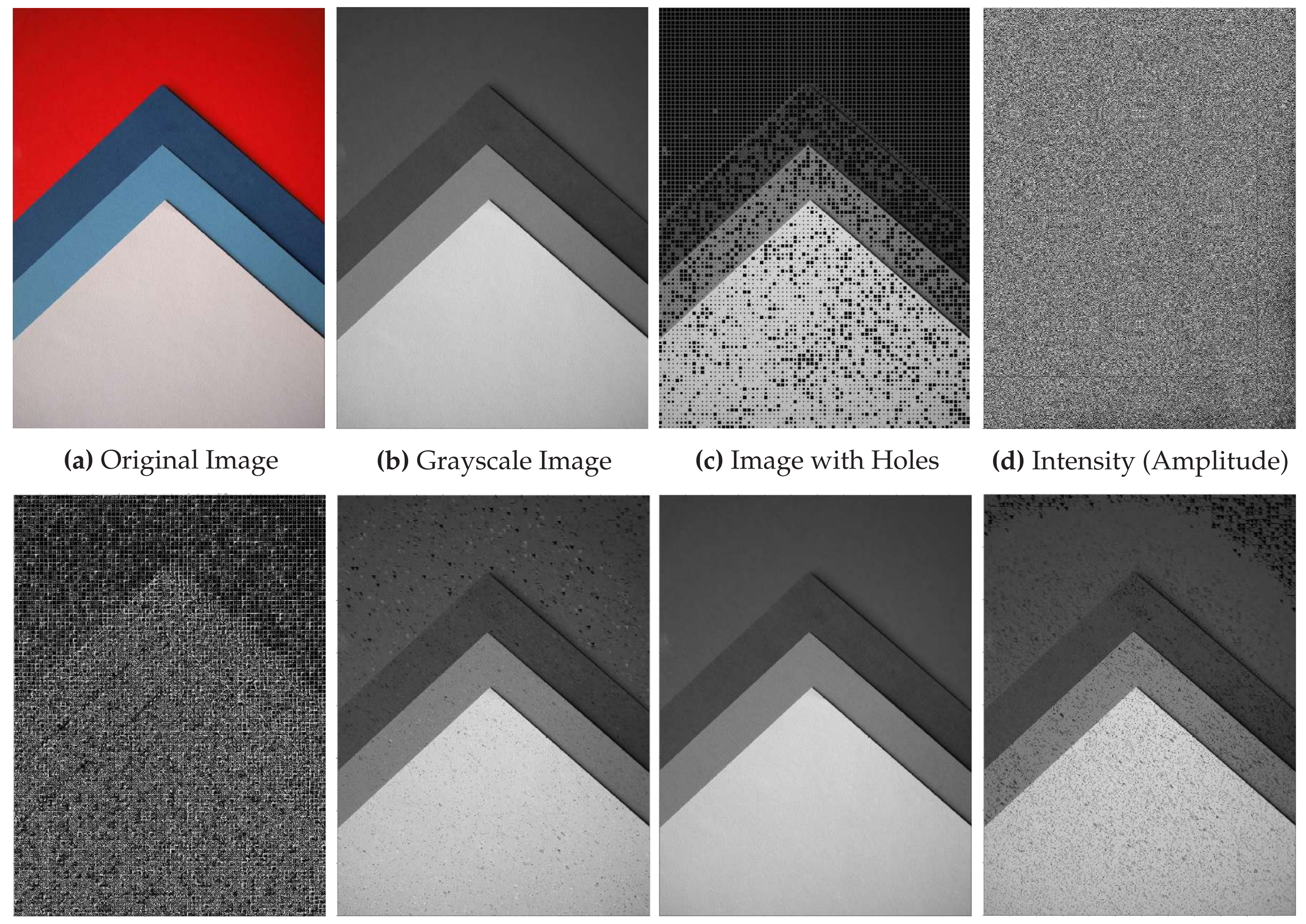
Figure 6: Landscape Images Error Analysis



(a) PSNR for all High Contrast Images (b) MSE for all High Contrast Images

Figure 7: High Contrast Images Error Analysis

Results: Processing



(a) Original Image (b) Grayscale Image (c) Image with Holes (d) Intensity (Amplitude)

(a) Quantized (b) Bit Flip Error (50%) (c) No Errors (d) 1's Comp. Error (50%)

Formulas

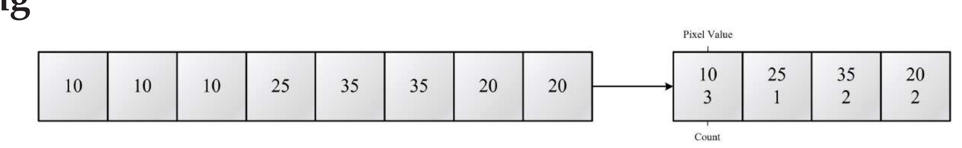
The Discrete Cosine Transform Formula

$$D(i,j) = \frac{1}{\sqrt{2N}} C(i)C(j) \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} p(x,y) \cos\left[\frac{(2x+1)i\pi}{2N}\right] \cos\left[\frac{(2y+1)j\pi}{2N}\right] \qquad (1)$$

The Chebyshev Distance Formula

$$D(p,q) = \max_i(|p_i - q_i|) \qquad (2)$$

Run Length Encoding



Compression Ratio

$$CR = \frac{No.\ of\ bits\ in\ uncompressed\ image}{No.\ of\ bits\ transmitted\ after\ encoding} \qquad (3)$$

Future Work

The designed algorithm for image compression can be improved in numerous ways, including:

- Utilizing parallel computing and programming to speed up the processing time of the algorithm
- A neural network can be trained on multiple images so that holes can be created in the larger picture as opposed to smaller 8x8 blocks within the image
- A neural network trained on multiple images at different compression depths can determine the correct check value
- A trained neural network can ultimately reconstruct an image and improve detail and quality of low-quality images.

Conclusion

This project is a proof of concept that an image compression technique to both create and fill holes is possible and viable to use in noisy environments. This was proven by:

- Creating a novel holes creation algorithm
- Using an additional well known compression scheme in addition to the holes algorithm
- Simulating a simple channel and introducing random errors to the channel
- Filling of holes and successful reconstruction of original image

Overall success criteria of the algorithm:

- Achieved compression ratios of an average of: 1.303765 (Pattern); 1.6475 (Landscape); 2.0909 (High Contrast)
- Average PSNR of: $-14.111325dB$ (Pattern); $-16.7051dB$ (Landscape); $-15.7210dB$ (High Contrast)
- Average MSE of: 25.904334 (Pattern); 52.7247 (Landscape); 57.5855 (High Contrast)

Acknowledgements

The pattern, landscape and high contrast images are taken from `Unsplash.com`, and the images used in this poster are by Andrej Lisakov and Pietro De Grandi.
The engineers would like to thank Professor Fambirai Takawira for his assistance as supervisor for this project.

References

[1] Uthayakumar, J. et al; *A survey on data compression techniques: From the perspective of data quality, coding schemes, data type and applications*; Journal of King Saud University - Computer and Information Sciences (2018)