HIERARCHICAL MODE JPEG COMPRESSION

The aim of this experimentation is to see what the JPEG Compression does to an image and what we have observed when we approach the Hierarchical Mode JPEG Compression. I have performed the image subsampling, up sampling at the respective places, encoding and decoding process at the expected times closely following the Architecture of Hierarchical JPEG. Below are observations.

1.1 Image subsampling.

- I have subsampled the image to 2 using imresize() function. The imresize() function resized the image into half and that is used for converting the RGB image to YUV format.
- The same procedure is applied for subsampling the image again to 2 using the imresize() function.

1.2 RGB to YUV conversion

- YUV colour format is attained by multiplying the R G and B values in an image to the corresponding matrix values.

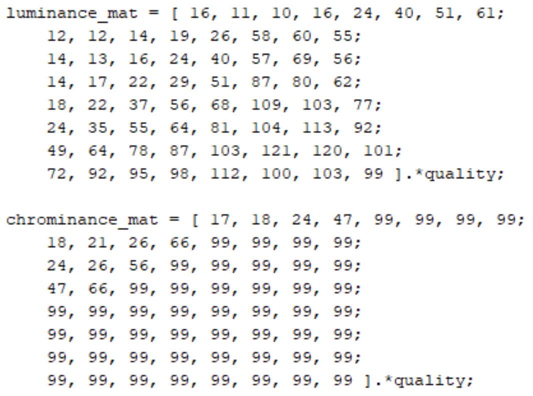
1.3 DCT matrix formation

- The 2D Discrete Cosine Transform matrix is created for an 8*8 block.
- 2D Discrete cosine transform matrix is calculated using the DCT equation formula.
- DCT-The process is repeated for every 8*8 block of image pixels using the formula D=TMT'. The 64 DCT coefficients are calculated for each image block, as the frequency becomes higher and higher when moved and received the highest frequency corresponding to the last image block.
- IDCT- The inverse DCT is applied after the decompression is performed by T*M*T.

2.1) Encoding & Decoding

- Both the encoding and decoding are performed using the Quantization and dequantization methods.
- The 8*8 block of DCT coefficients are now ready for performing the quantisation which does the encoding of the mage. The image is encoded by element wise multiplication with Luminance quantization table matrix for Y and element wise multiplication with chrominance quantization table matrix for U and V.

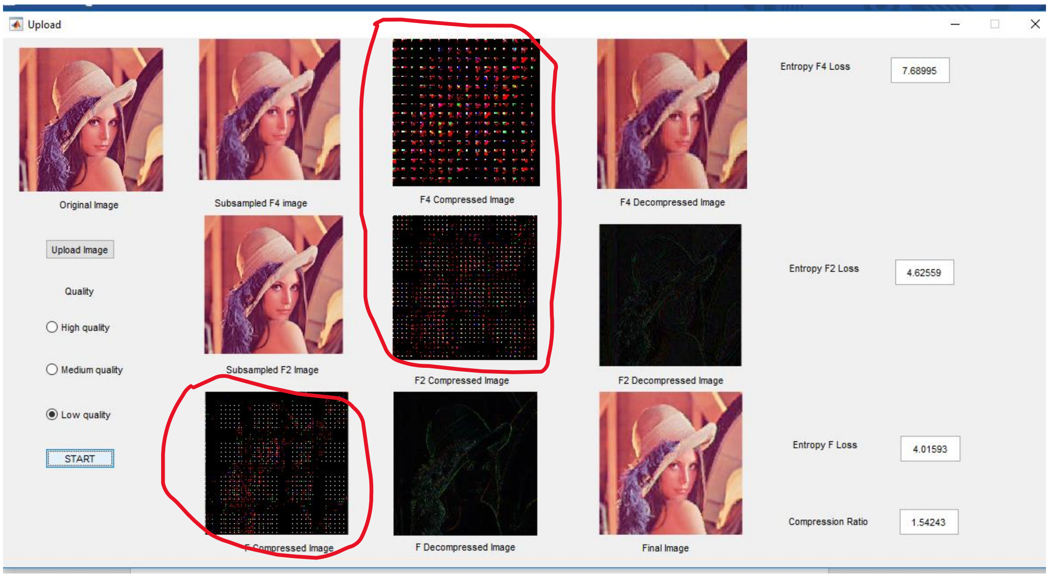
  Figure showing the Luminance and Chrominance Quantization and multiplied with quality received as High, Medium and Lower from the User GUI.



```
luminance_mat = [ 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 ].*quality;

chrominance_mat = [ 17, 18, 24, 47, 99, 99, 99, 99;
    18, 21, 26, 66, 99, 99, 99, 99;
    24, 26, 56, 99, 99, 99, 99, 99;
    47, 66, 99, 99, 99, 99, 99, 99;
    99, 99, 99, 99, 99, 99, 99, 99;
    99, 99, 99, 99, 99, 99, 99, 99;
    99, 99, 99, 99, 99, 99, 99, 99;
    99, 99, 99, 99, 99, 99, 99, 99 ].*quality;
```

Figure showing the compressed images from three layers of JPEG Compression. This has been highlighted in red circles below.



Quality

○ High quality

○ Medium quality

● Low quality

Entropy F4 Loss 7.68995

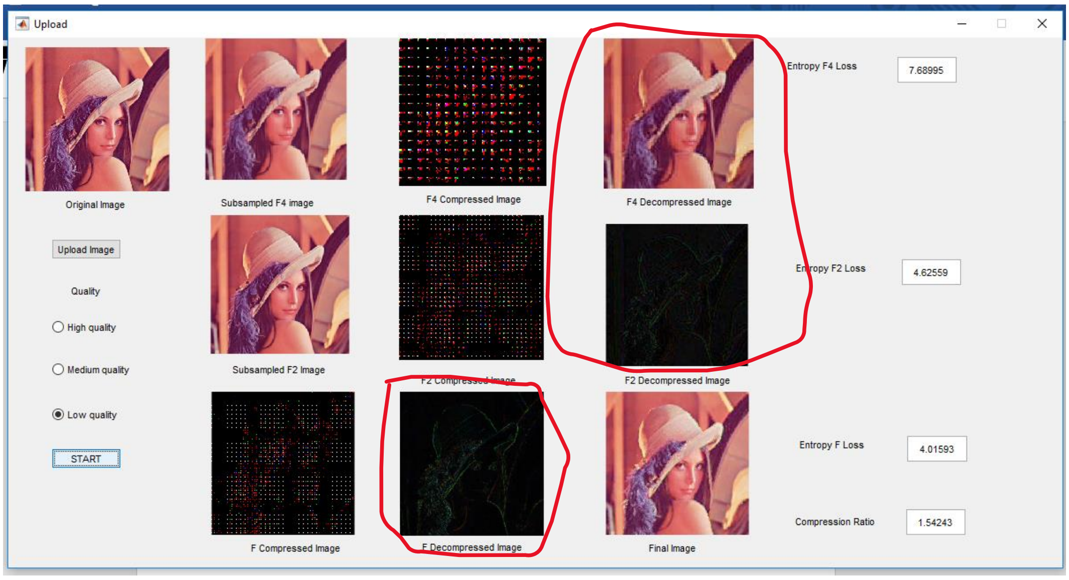
Entropy F2 Loss 4.62559

Entropy F Loss 4.01593

Compression Ratio 1.54243

- The 8*8 block of DCT coefficients are used for dequantization method for decoding the image. The image is decoded by element wise division with Luminance quantization table matrix for Y and element wise division with chrominance quantization table matrix for U and V.
- The decompressed image is now sent to IDCT and then converted back to RGB image and displayed as below.

Figure showing the decompressed images from three levels of Hierarchical JPEG Compression as shown highlighted in red circles below: -



Quality

○ High quality

○ Medium quality

● Low quality

Entropy F4 Loss 7.68995

Entropy F2 Loss 4.62559

Entropy F Loss 4.01593

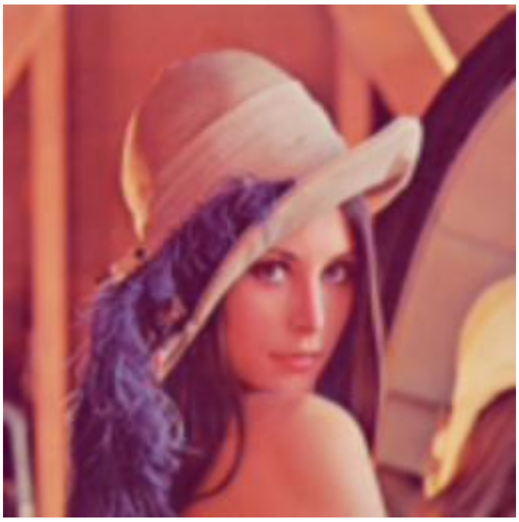
Compression Ratio 1.54243

2.2 Upsampling

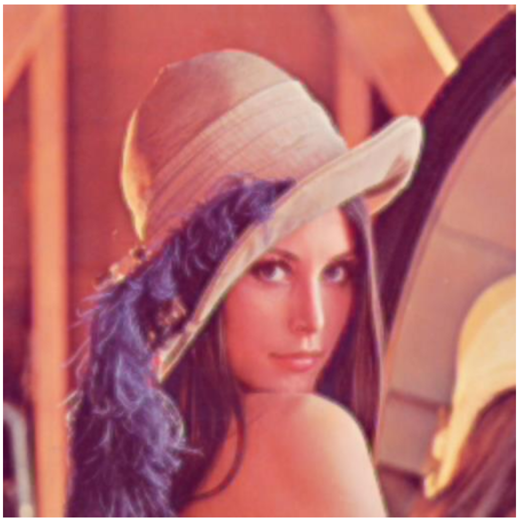
- F4dash-The decompressed image after F4 is now upsampled and interpolated using the imresize() function respectively.
- F2dash- The decompressed image after F2 is now upsampled and interpolated using the imresize() function respectively.

Figure 1&2 showing the up sampled F4 and F2 images. We could see that the information is lost more in F4 and reconstructed in the F2.

Sample Upsampled F4 Image



Sample Upsampled F2 Image



2.3 Finding the Differences D1 and D2.

- D1 difference is identified between the subsampled F4 image and the upsampled F4dash.
- D2 difference is identified between the subsampled F2 image and the upsampled F2dash.

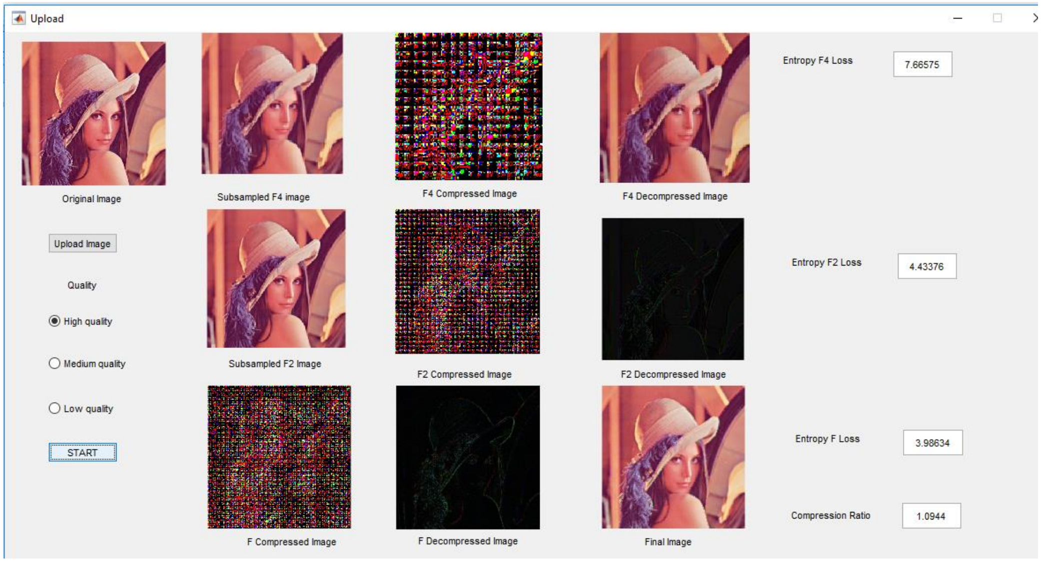
3) Entropy Loss

- Entropy loss has been calculated by calculating the product of array elements of the image and calculating the count of the histogram populated as a numeric array of elements. Divide the product with the counts. Using the above value, we can calculate the entropy loss of the image using the sum of the value with multiplying the log2(value).
- It is been observed that the F4, F2 and F0 values are not constantly decreasing. It is changing for few images.

4) Graphical User Interface (GUI)

- A Graphical User Interface has been created for this JPEG compression.
- I have used the guide GUI function in matlab to achieve this.
- An upload image has push button has been created to upload the any type of image format.
- Have used the handles function to get the filename and the image.

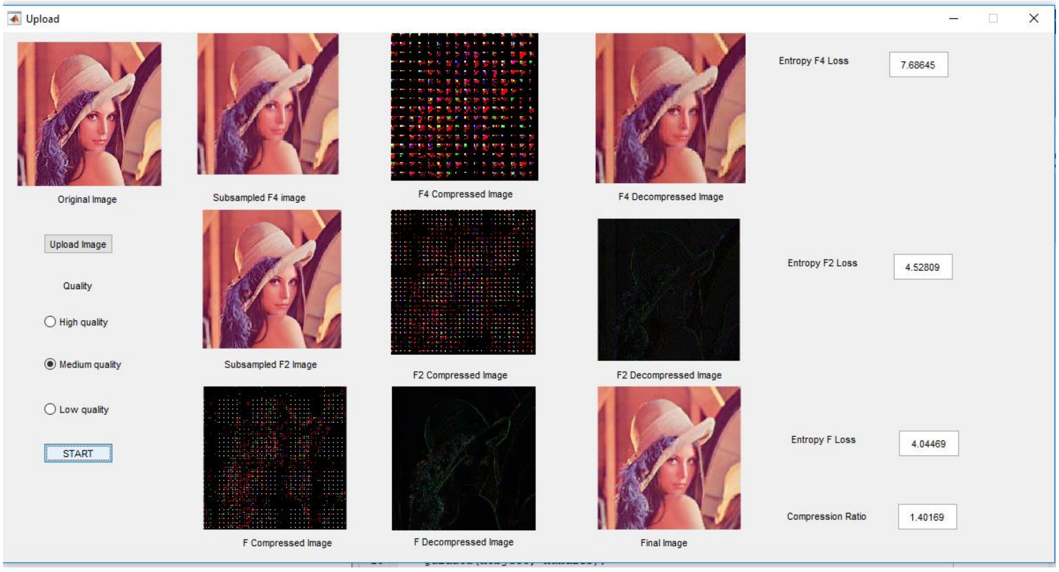
- All the encoded and the decoded images that are received for F4, F2 and F0 are displayed in the GUI. I have used the handles.axes function to display the images in the expected axes in the GUI.
- The entropy loss that has been calculated is displayed in the GUI using the handles.edit function.
- The compression ratio of the original image and the final image is calculated and displayed in the GUI using the handles.edit function.
-
- High Quality Pixel Reconstructed Image.



● High quality
○ Medium quality
○ Low quality

Entropy F4 Loss 7.66575
Entropy F2 Loss 4.43376
Entropy F Loss 3.98634
Compression Ratio 1.0944

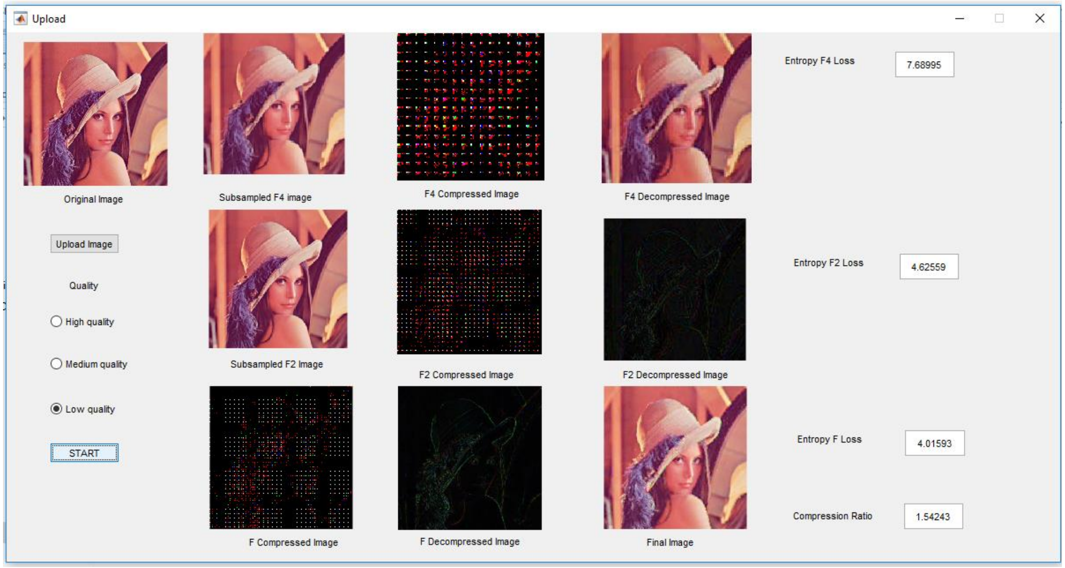
- Medium Quality Pixel Reconstructed Image



○ High quality
● Medium quality
○ Low quality

Entropy F4 Loss 7.68645
Entropy F2 Loss 4.52809
Entropy F Loss 4.04469
Compression Ratio 1.40169

- Lower Quality Pixel Reconstructed Image



Quality

○ High quality

○ Medium quality

● Low quality

Entropy F4 Loss 7.68995

Entropy F2 Loss 4.62559

Entropy F Loss 4.01593

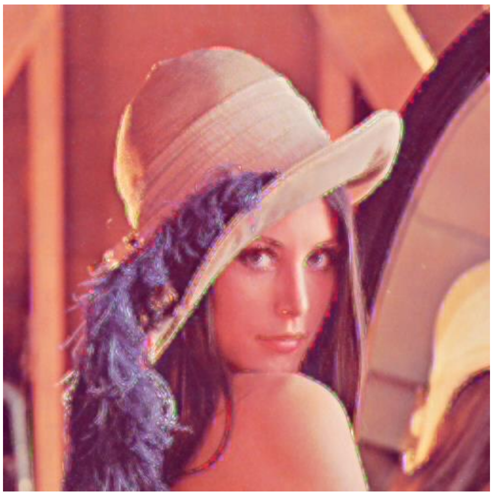
Compression Ratio 1.54243

- The Compression ratio is calculated by taking the Image1 bytes/Image2 bytes. This gives the compressed ratio from the Original Image to the final Compressed Image.

During High Quality Pixel Compression – Compression Ratio -1.09

Original Image

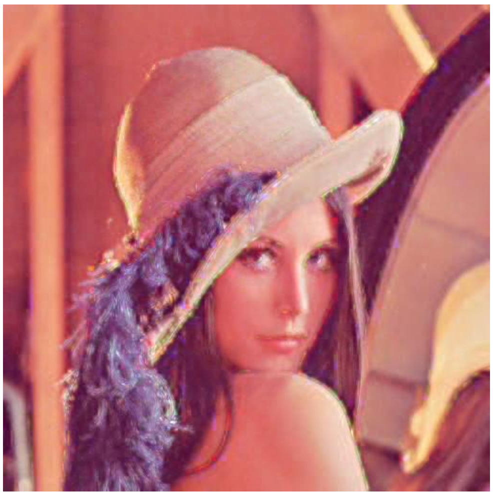
Final Image



- Mid-Level Quality Compression- Compression Ratio- 1.40

Original Image

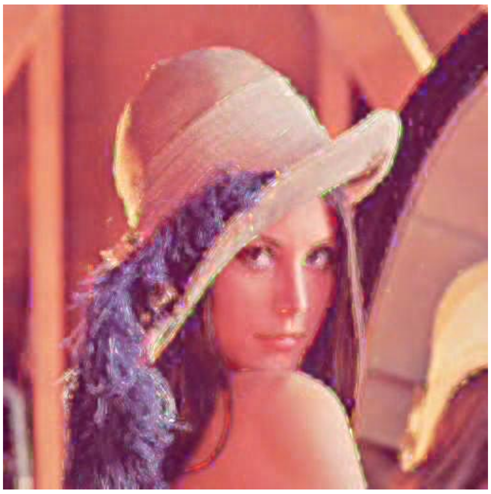
Final Image



- Low Quality Compression – Compression Ratio- 1.542

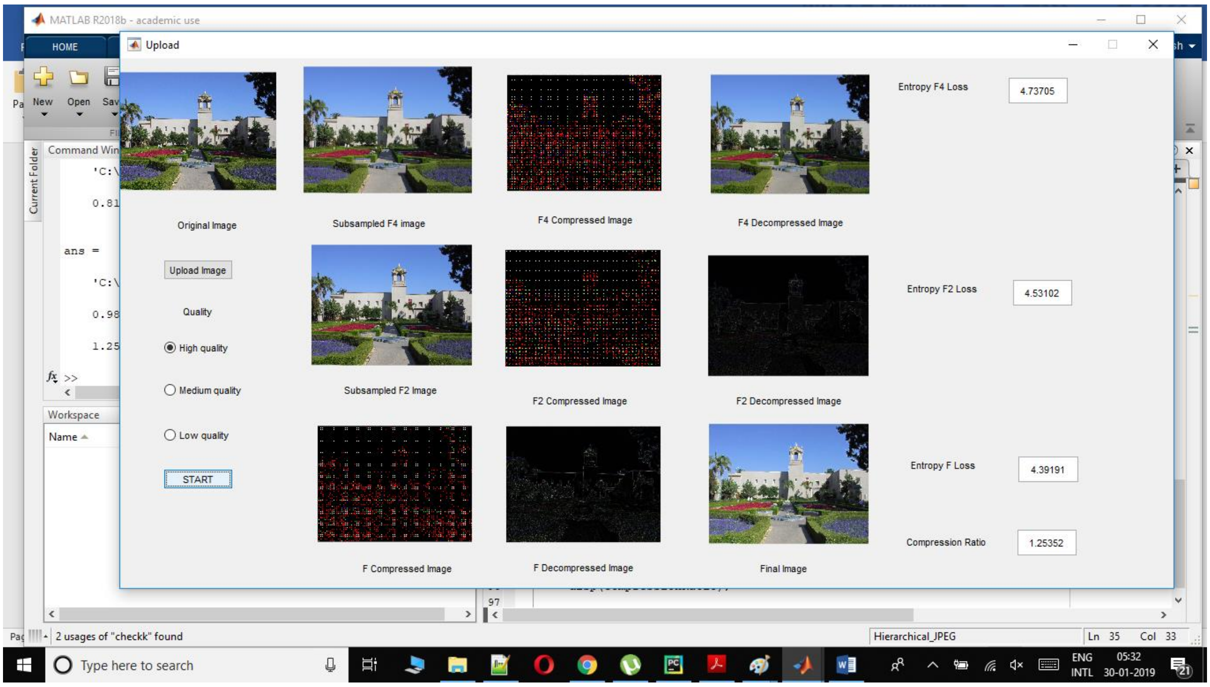
Original Image

Final Image



As we could observe that the quality of the image is distorted as the Compression is increased.

The case is been studied for various of the images and are shown in the below figures respectively.



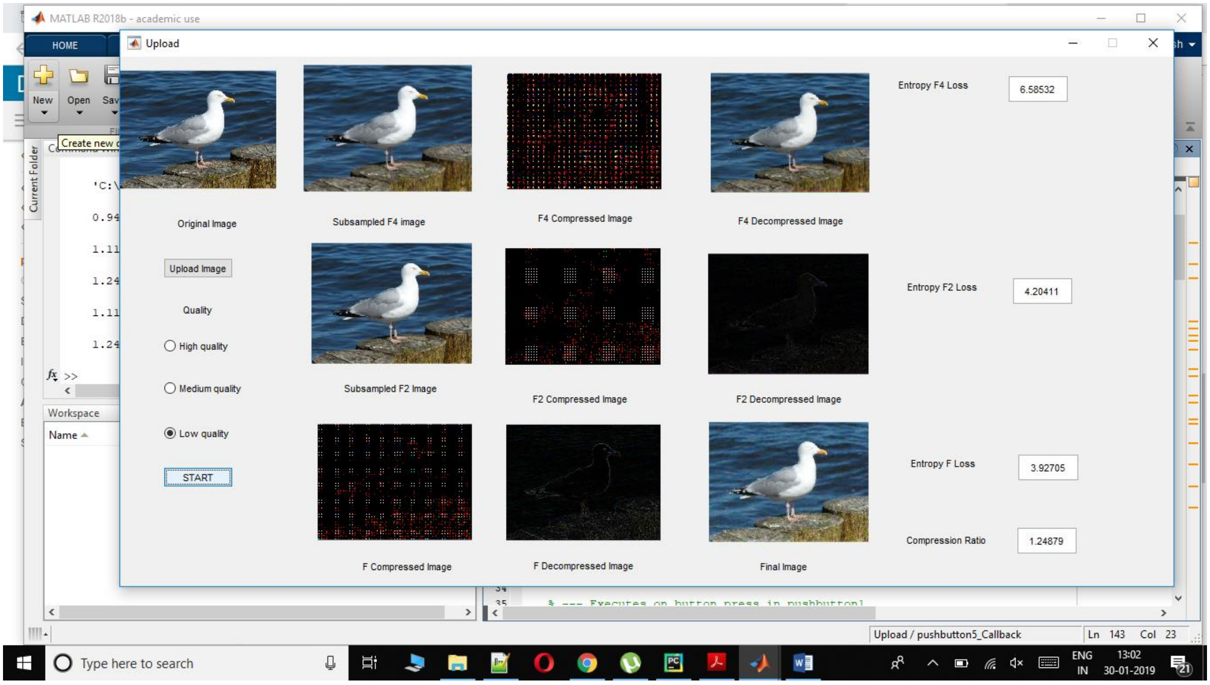
Quality
● High quality
○ Medium quality
○ Low quality

Entropy F4 Loss 4.73705

Entropy F2 Loss 4.53102

Entropy F Loss 4.39191

Compression Ratio 1.25352



Quality
○ High quality
○ Medium quality
● Low quality

Entropy F4 Loss 6.58532

Entropy F2 Loss 4.20411

Entropy F Loss 3.92705

Compression Ratio 1.24879