

Instructions:

This assignment requires programming in Python. Please submit this assignment individually through Canvas. Please submit a zipped folder including a document file and a python file. Your python codes should be inside a file named main.py, and the report and the answer to the questions should be brought in document(.doc or pdf)

## Part1

Q1. Please download the data from gyroscope sensor in the link below([gyroData.txt](#)

[Download gyroData.txt](#)

).

Put this in the same folder as your python code. Load the data into your python code. In order to open the file use the following piece of code.

```
from numpy import loadtxt
```

```
dataStr=loadtxt("gyroData.txt", delimiter="," ,dtype='str' )
```

```
data=dataStr.astype(int)
```

This data is the output from the accelerometer in x-axis. The range of data is integer values from -8 up to 7 (16 integers).

Q2. Use the following code ( [sortCount.py](#)

[Download sortCount.py](#)

) to find how frequent each integer value shows up in the dataset. Fill out the table below with the numbers on how frequent each level shows up inside the dataset.

level	-4	-3	-2	-1	0	1	2	3
number								

Q3. Draw a chart to get Huffman code representation for this data set, assigning lower number of bits to more frequent levels. Put a snapshot of your Huffman algorithm steps.

Q4. Write down the code representations you derived in part3 inside the below table. Knowing the number of times each level happens in the data , sum the entire number of bits you need to represent the total data.

level	-4	-3	-2	-1	0	1	2	3
bit representatio n								

## Part2

Q1. Install the discrete wavelet transform package :

*sudo apt install python3-pywt*

Use the pywt package to compute the 3<sup>th</sup> order Haar wavelet transform of the signal from part 1 ([gyroData.txt](#)

[Download gyroData.txt](#)

).

Q2. The energy of the signal is defined as the sum of its squared samples. Compute the energy of the original signal . Also compute the energy in approximate and detail coefficients in each level of decomposition separately. Discuss how the energy is distributed in the decomposed signal at each level of decomposition.

## Part3

Q1.Go to the course files section of Canvas and download the image. "emma.png" "



You will need opencv package to calculate the discrete cosine transform. Install opencv as

```
sudo apt install python3-opencv
```

Use 'cv2.imread' function to load the picture "emma.png" file.

To display the image you should install the matplotlib package.

```
sudo apt install python3-matplotlib
```

Use 'plt.imshow' to display the image. Put the snapshot of the image in your report.

**Q2.** Your image has 2D matrices for each color of red, green, and blue. Get the red matrix and apply a 2-dimensional Discrete Cosine Transform . You may use the **cv2.dct** command to extract the DCT coefficient matrix. Visualize the DCT coefficient matrix using the logarithmic scale of the coefficients. Put a snapshot of the coefficient matrix image inside your report

**Q3.** Inside the coefficients matrix, the energy of the signal is concentrated where the absolute value of coefficients are larger. Ignore the smaller coefficients by setting those

coefficients which have absolute value of less than 10 to 0. How many of the coefficients are kept? What is the compression ratio? The ratio of number of significant coefficients to the total number of coefficients is the compression ratio.

**Q4.** Use Inverse Cosine Transform to reconstruct the compressed image using the new coefficient matrix.

**Q5.** Display the reconstructed image and check for the quality of the picture. The introduced error inside the compressed image can be measured using MSE (mean squared error ). Extract the MSE.