VC Project Report

Project-2: Image Compression using Lapped Transform

***Group Members:***

|  |  |  |
| --- | --- | --- |
| Name | Matriculation No. | Email |
| Asad Ullah Haider | 66374 | asad-ullah.haider@tu-  ilmeanu.de |
| Muhammad Shariq | 65715 | shariq.kiet@tu- ilmenau.de |
| Yaseen Tahir | 66269 | yaseen.tahir@tu- ilmenau.de |

# Task:

The task was to compare the results of Lapped Transform compressed image with respect to the DCT compressed image by using Perceptual Similarity Metric and Compression Ratio.

# Step 1: Image Compression using DCT:

I have used the below link for this step as base code.

[image-compression/jpeg\_compression.py at main · Karanraj06/image-compression · GitHub](https://github.com/Karanraj06/image-compression/blob/main/jpeg_compression.py)

**Step 2: Image Compression using Lapped Transform:**

1. Encode an image (color or grayscale) using Lapped Transform and quantization

* Read the image
* Apply the lapped transform using a trained neural network model
* Quantize the transformed coefficients
* Flatten the quantized coefficients and retain the specified number of coefficients

1. Decode the encoded image back into an image

* Unflatten the encoded data
* Dequantize the coefficients
* Apply the inverse lapped transform to reconstruct the image

1. Encodes and decodes an image, then computes PSNR and compression ratio

* Encode the image using the lapped transform encoder
* Decode the image using the lapped transform decoder
* Calculate PSNR between the original and compressed images
* Calculate the compression ratio based on the size of the compressed data

1. Visualize the original and compressed images side by side, and save the encoded data

* Plot the original and compressed images using matplotlib
* Save the encoded image data to a text file

1. Save the compressed image to a specified path

* Write the compressed image data to a file using cv.imwrite

1. Output for original image vs Lapped Transform compressed image

* Obtained a PSNR and a compression ratio for the color image

**Step 3: Comparison of Lapped Transform and DCT Compression Results**

* Cloned the repository and installed the necessary dependencies for the Perceptual Similarity Metric too
* Imported the required models from perceptual\_similarity
* Compute perceptual similarity between the original and compressed images
* Initialized paths for both compressed and original images
* Computed perceptual similarity between the original and compressed images
* Calculated the file sizes of compressed and original images to determine the compression ratio
* Calculated bits per pixel (BPP) based on image dimensions
* Printed the perceptual similarity for both Lapped Transform and DCT
* Printed the results, including perceptual similarity, bits per pixel, and compression ratio

# Results:

|  |  |  |  |
| --- | --- | --- | --- |
| Compression Method | Perceptual Similarity | Compression Ratio | Bits Per Pixel (BPP) |
| |  | | --- | | Lapped Transform |  |  | | --- | |  | | 0.82 | 4 | 11.44 |
| DCT | 0.27 | 1.19 | 9.11 |

**Conclusion:**

Based on the metrics, the Lapped Transform compression method achieves a higher compression ratio compared to DCT while maintaining perceptual similarity. This suggests that the Lapped Transform method is effective for image compression, offering better compression with perceptually similar results.