Image Processing

Assignment 5

Pouria melek khayat

Exercise 1

We implement a basic image compression algorithm using the Discrete Cosine Transform (DCT) and quantization techniques.

1. **display_images**(images)

- Description: This function displays a list of images in a single figure.

- Input: images - a list of images to be displayed.

- Output: None.

2. **encode_image**(image, block_size, quant_matrix, loss_factor)

- Description: This function encodes an image using the specified block size, quantization matrix, and lossiness factor.

- Input:

  - image - the original image to be encoded.

  - block_size - the size of the blocks used for encoding.

  - quant_matrix - the quantization matrix used for quantizing the DCT coefficients.

  - loss_factor - the lossiness factor that adjusts the amount of loss during compression.

- Output: A list of compressed blocks representing the image.

3. **decode_image**(compressed_blocks, block_size, quant_matrix, image_shape, loss_factor)

- Description: This function decodes a list of compressed blocks back into an image using the specified block size, quantization matrix, image shape, and lossiness factor.

- Input:

  - compressed_blocks - a list of compressed blocks representing the image.

  - block_size - the size of the blocks used during encoding.

  - quant_matrix - the quantization matrix used during encoding.

  - image_shape - the shape of the original image.

  - loss_factor - the lossiness factor used during encoding.

- Output: The reconstructed image.

4. **extract_blocks**(image, block_size)

- Description: This function extracts non-overlapping blocks from the image based on the specified block size.

- Input:

  - image - the original image.

  - block_size - the size of the blocks.

- Output: A list of extracted blocks.

5. **apply_dct**(block, channels) - Description: This function applies the Discrete Cosine Transform (DCT) to each channel of a block.

- Input:

  - block - a block of image data.

  - channels - the number of channels in the block.

- Output: The block with DCT coefficients applied.

6. **quantize_dct**(dct_block, quant_matrix, loss_factor)

- Description: This function quantizes the DCT coefficients of a block using the specified quantization matrix and lossiness factor.

- Input:

  - dct_block - a block with DCT coefficients.

  - quant_matrix - the quantization matrix.

  - loss_factor - the lossiness factor.

- Output: The quantized block.

7. **dequantize_dct**(quantized_block, quant_matrix, loss_factor)

- Description: This function dequantizes the quantized DCT coefficients of a block using the specified quantization matrix and lossiness factor.

- Input:

  - quantized_block - a block with quantized DCT coefficients.

  - quant_matrix - the quantization matrix.

  - loss_factor - the lossiness factor.

- Output: The dequantized block.

8. **apply_idct**(block)

- Description: This function applies the Inverse Discrete Cosine Transform (IDCT) to each channel of a block.

- Input: block - a block of image data.

- Output: The block with IDCT applied.

9. **combine_blocks**(blocks, image_shape, block_size)

- Description: This function combines the reconstructed blocks into a single image based on the specified image shape and block size.

- Input:

  - blocks - a list of reconstructed blocks.

  - image_shape - the shape of the original image.

  - block_size - the size of the blocks used during encoding.

- Output: The reconstructed image.

10. **get_image_size**(image)

  - Description: This function calculates the size of the image in megabytes.

  - Input: `image` - the image to calculate the size of.

  - Output: The size of the image in megabytes.

11. **pad_image**(image)

  - Description: This function pads the image to make its dimensions divisible by the block size.

  - Input: image - the original image.

  - Output: The padded image.

The code provides a basic implementation of image compression using the DCT and quantization techniques. By adjusting the block size and lossiness factor, the user can control the trade-off between image quality and compression ratio.

Notebook link

https://colab.research.google.com/drive/1qNL6vO5QMrTXr2dlPVURiX9lvQhSnpnc#scrollTo=sH48BVcpH1TS