

DIGITAL IMAGE AND VIDEO PROCESSING LAB - EXPERIMENT NO. 2

Submitted by Unnati Singh, 21EC39027

Overview:

This experiment involves the implementation of Python functions to read, write, and manipulate BMP images. The tasks include extracting BMP header information, saving image data, and performing colour channel manipulations on an image.

Files Included:

1. `Exp2_21EC39027.py`: Python script containing the following functions:
 - a. `readBMP(path)`: Reads a BMP file, extracts and prints the details of the file, and loads the image pixel array into memory.
 - b. `writeBMP(outputfilename, pixelarray, size)`: Writes the pixel array and height and width into a new BMP file.
 - c. `colourchannelmanipulation(filename, channel)`: Sets a specified colour channel to zero in the provided image.
2. `Exp2_21EC39027.ipynb`: Jupyter Notebook containing the same code.
3. `Exp2_21EC39027_report.pdf`: The experiment report detailing the objective, methodology, results, and conclusion.
4. `Input Images/`: Directory containing the input images.
5. `Output Images/`: Directory containing the output images obtained from the code.
6. `README.pdf`: This ReadMe file.

Functions Overview:

1. `readBMP(path)`: This function reads a BMP file and returns the BMP header and pixel array. It also prints key header details like image dimensions and bit depth.
2. `writeBMP(outputfilename, pixelarray, size)`: This function takes the pixel array and dimensions of the new image and writes them into a new BMP file specified by filename.
3. `colourchannelmanipulation(filename, channel)`: This function modifies the specified channel ('R', 'G', 'B') of the input image by setting it to zero and saves the resulting image.

Notes:

1. The script currently supports 24-bit BMP images and can handle 8-bit grayscale and 8-bit colour-indexed images.
2. Ensure that the input images are in BMP format. The script will output an error if the file format is unsupported.