Multimedia İmage Assignment Report

Ferzan Kara

Introduction

This report analyzes the structure and processing of a digital image (baby.jpg) using Python with the Pillow, NumPy, and Matplotlib libraries. The assignment involves manipulating the image through five steps: analyzing dimensions and compression, brightening the image, removing red shades, and simulating JPEG compression with chroma and full downsampling. Results are visualized and discussed to highlight the effects of each operation.

1. Image Dimensions and Compression Rate

The image was opened using the Pillow library to extract its dimensions. Assuming an RGB model (24-bit, 3 bytes per pixel), the uncompressed size was calculated and compared to the actual JPEG file size to determine the compression rate.

Results:

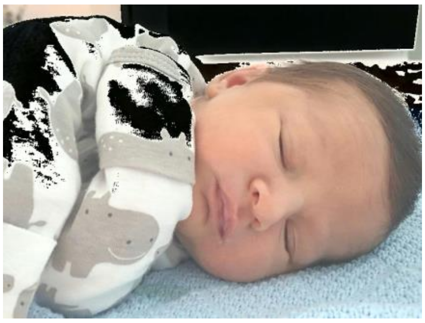
- **Image Size**: 756 × 567 pixels
- **Uncompressed Size**: 756 × 567 × 3 = 1,286,676 bytes
- **Compressed Size**: ~110,600 bytes (obtained via os.path.getsize)
- **Compression Rate**: ~11.63 (uncompressed size ÷ compressed size)

The high compression rate reflects JPEG's efficiency in reducing file size while maintaining visual quality.

2. Brightening the Image Using the Y Component

The image was converted from RGB to YCbCr color space, where the Y component represents brightness, and Cb/Cr carry color information. The Y component was increased by 50 units (clipped to 0–255) to enhance brightness, and the image was converted back to RGB.

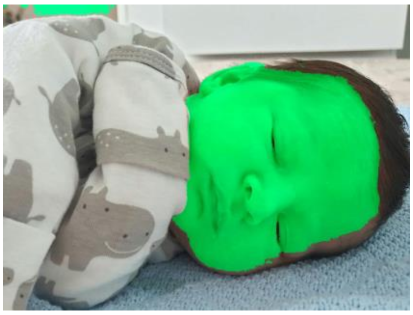
Results:
The brightened image (baby_bright.jpg) is visibly lighter, with no significant changes to color tones. This confirms that manipulating the Y component effectively controls brightness without directly affecting chroma.



3. Detecting and Removing Red Shades

To isolate red areas, the image was converted to YCbCr, and pixels with Cr values between 140 and 180 (indicative of red tones) were identified. These pixels' Cr values were set to zero, and the image was converted back to RGB.

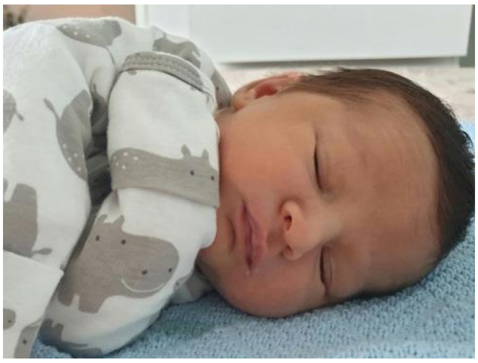
Results:
The output image (baby_no_red.jpg) lacks red tones, with affected areas (e.g., skin regions) appearing greenish due to the absence of the Cr component. This demonstrates the Cr channel's role in representing red hues.



4. JPEG Compression Simulation (Cb and Cr Downsampling)

JPEG compression reduces file size by downsampling chroma channels (Cb and Cr), as the human eye is less sensitive to color details. The Cb and Cr channels were downsampled by a factor of 2 (width and height halved) using bilinear interpolation, then upsampled to the original size. The image was reconstructed using the original Y channel and upsampled Cb/Cr channels.

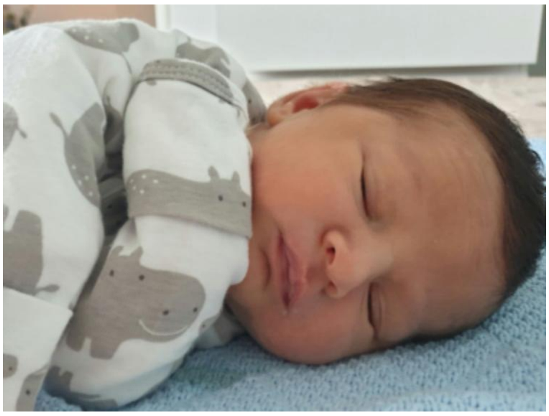
Results:
The resulting image (baby_jpeg_simulated.jpg) shows a slight loss in color fidelity, particularly in areas with fine color details, but the overall structure and brightness remain intact. This aligns with JPEG's ability to maintain visual quality during compression.



5. Full Downsampling (Y, Cb, Cr)

All three components (Y, Cb, Cr) were downsampled by a factor of 2 and then upsampled to the original size using bilinear interpolation. The image was reconstructed in RGB.

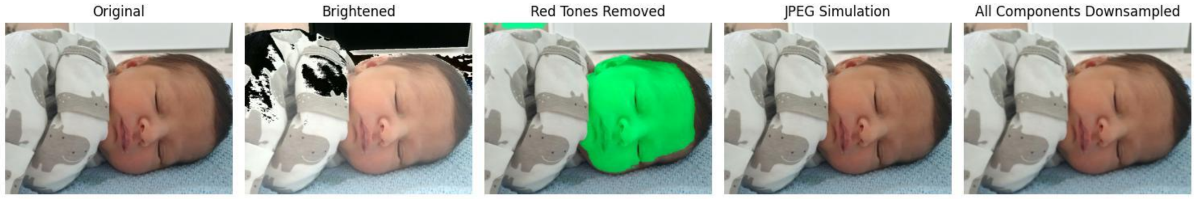
Results:
The output image (baby_all_downsampled.jpg) exhibits noticeable blurring and loss of sharpness compared to Step 4. Degrading the Y (luminance) channel significantly impacts image quality, highlighting its critical role in preserving detail.



Visualization

A comparison image (results_comparison.jpg) was generated to display the original and processed images side by side, illustrating the visual effects of each step (see Figure 1).

Figure 1: Side-by-side comparison of original, brightened, red-removed, JPEG-simulated, and fully downsampled images.



Conclusion

This assignment demonstrated key digital image processing techniques using Python:

- Extracting image properties and calculating compression rates.
- Converting between RGB and YCbCr color spaces.
- Modifying brightness and color channels.
- Simulating JPEG compression through chroma and full downsampling.

Each step was implemented, visualized, and analyzed. The results mirror real-world multimedia techniques, such as JPEG compression, and emphasize the importance of the luminance (Y) channel in maintaining image quality.