JPEG to Bitmap Image Converter

By Shane Davey and Angel Martinez

What is JPEG?

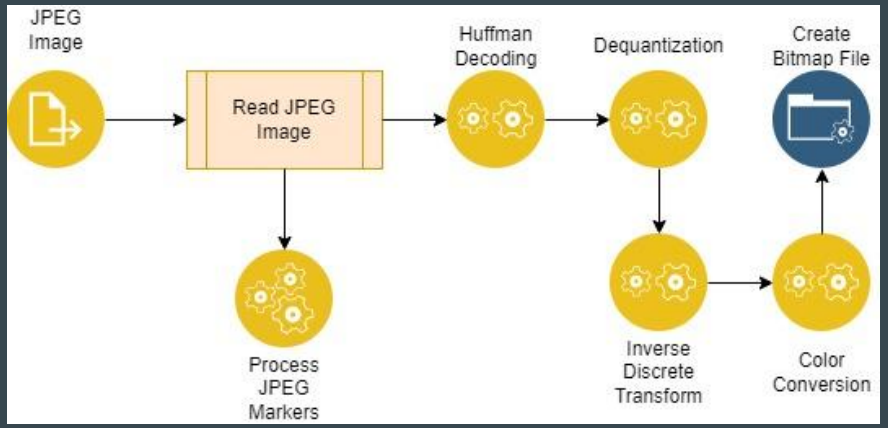
- “Joint Photographic Experts Group”
- Standard image format for compression
- Greatly reduces file size
- Great for sharing over the internet
- Must be encoded and decoded

What is Bitmap

- One of the file formats for images
- Bitmap images are not compressed
- Are built pixel-by-pixel
- Images size are larger than JPEG images

Decoding JPEG

- Read JPEG
- Process JPEG Markers
- Huffman Decoding
- Dequantization
- Inverse Discrete Transform
- Color Conversion
- Write to Bitmap



JPEG Markers

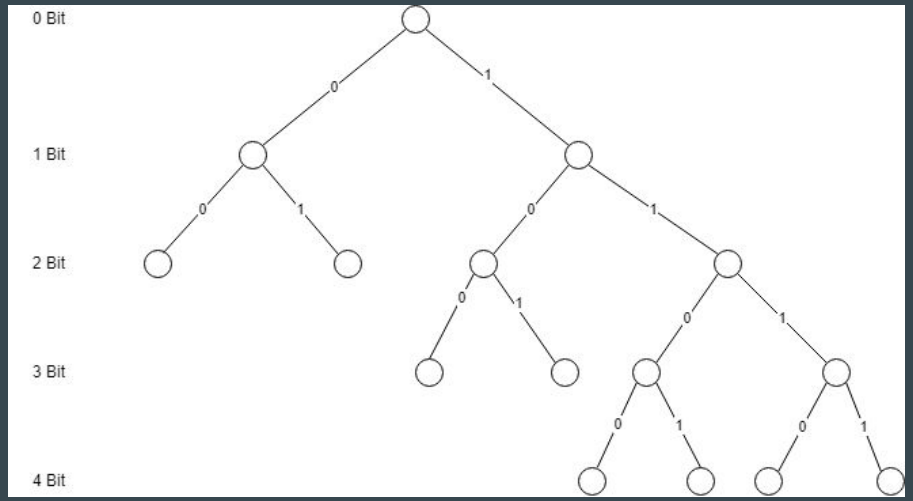
- The JPEG image header is made up of markers.

- JPEG markers examples
  - DQT, DHT, SOF, SOS, SOI

- JPEG markers importance

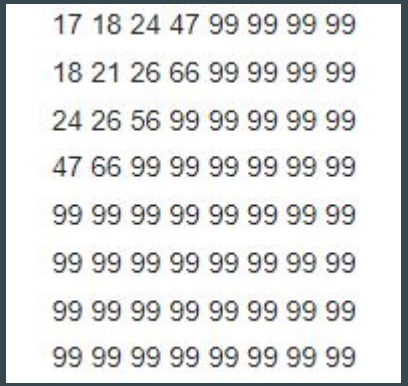
Huffman Decoding

- Compress/Decompress Data
- Frequency Based
- Symbols are fixed width
- Common symbols assigned shorter codes
- Uncommon symbols assigned longer codes
- Tree used for encoding must be saved



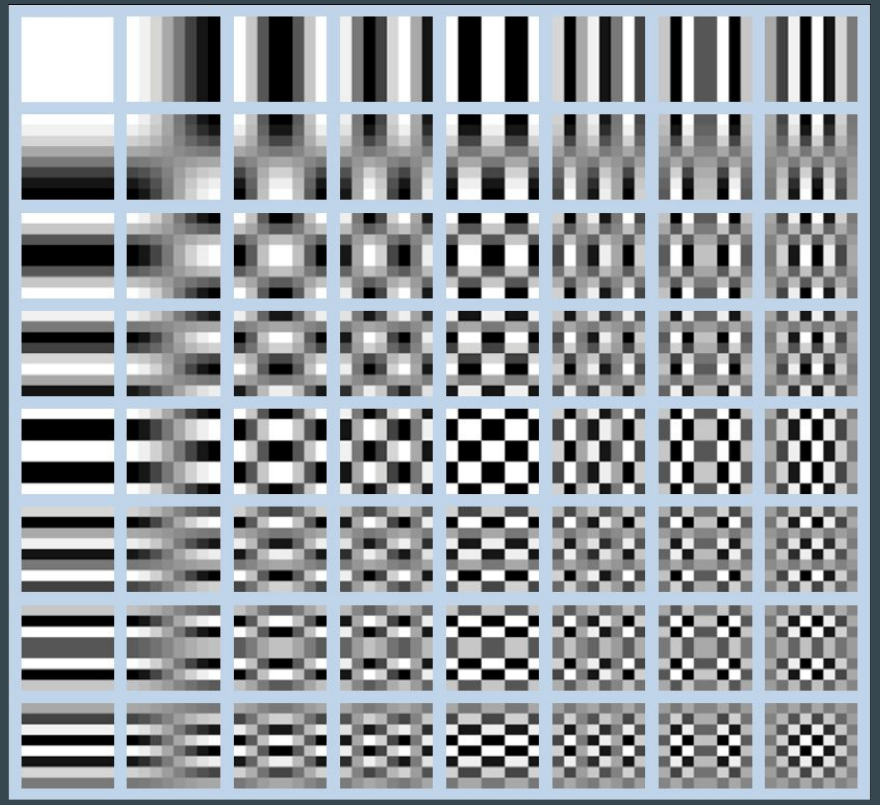
Dequantization

- Obtaining quantization tables
  - 16 bit or 64 bit
- ZigZag Principle
- Performing dequantization



Inverse Discrete Cosine Transform

- Vertical frequency left column
- Horizontal frequency top row
- DCT coefficients
- Retaining color data

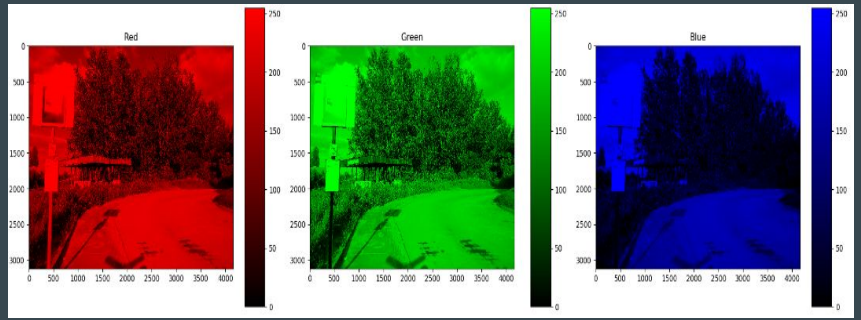


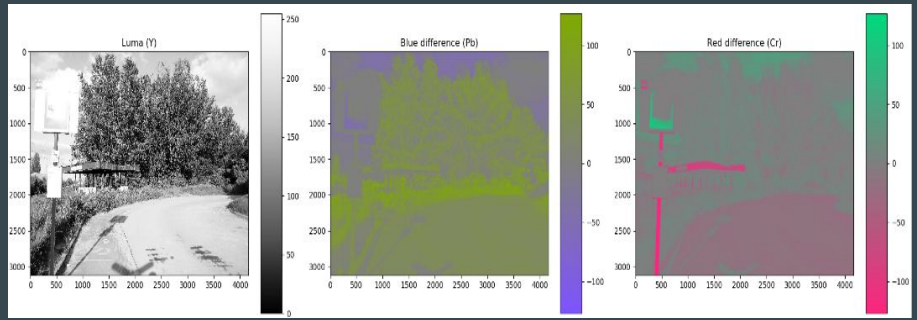
Color Conversion

- Bitmap color data - R, G, B
- JPEG color data - Y, Cb, Cr
- Conversion from one to another

$$
\begin{aligned}
Y' &= 0 + (0.299 \cdot R'_D) + (0.587 \cdot G'_D) + (0.114 \cdot B'_D) \\
C_B &= 128 - (0.168736 \cdot R'_D) - (0.331264 \cdot G'_D) + (0.5 \cdot B'_D) \\
C_R &= 128 + (0.5 \cdot R'_D) - (0.418688 \cdot G'_D) - (0.081312 \cdot B'_D)
\end{aligned}
$$

$$
\begin{aligned}
R'_D &= Y' + 1.402 \cdot (C_R - 128) \\
G'_D &= Y' - 0.344136 \cdot (C_B - 128) - 0.714136 \cdot (C_R - 128) \\
B'_D &= Y' + 1.772 \cdot (C_B - 128)
\end{aligned}
$$





Writing to Bitmap

- Structure of bitmap images
  - Header, InfoHeader, ColorTable
- Write the pixel rows of the bitmap image
- Padding bytes

```
//Set the Bitmap Header
bitmap_file.put('B'); //Signature
bitmap_file.put('M'); //Signature
write_4_byte(bitmap_file, bitmap_file_size); //Write the file size
write_4_byte(bitmap_file, 0); //Unused values
write_4_byte(bitmap_file, 0x1A); //Offset from beginning of file to the beginning of the bitmap data
//Info Header
write_4_byte(bitmap_file, 12); //Size of the bitmap info header
write_2_byte(bitmap_file, header->jpeg_width); //Pixel width of bitmap
write_2_byte(bitmap_file, header->jpeg_height); //Pixel height of bitmap
write_2_byte(bitmap_file, 1); //Number of planes
write_2_byte(bitmap_file, 24); //Number of bits per pixel
```

Difficulties and Challenges

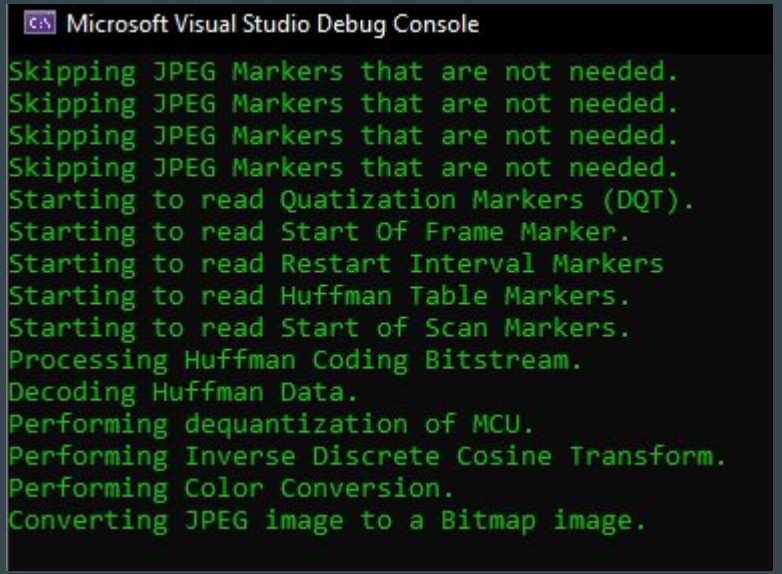
- JPEG file format
- Decoding the JPEG
- Not using existing libraries
- Lots of Testing

Results

- New bitmap image is created

- Image is not compressed
  - JPEG file size: 49.8 KB,
  - New bitmap file size: 263 KB

- Image went through all the steps of decoding



Demo Output

JPEG image

Bitmap image

Questions?