P1-JPEG

Clara Rabasseda Cases

217030

1) Start a script called rgb_yuv.py and create a translator from 3 values in RGB into the 3 YUV values, plus the opposite operation. You can choose the 3 values, or open them from a text file, receive it from command line… feel free.

This exercise is done in a PyCharm script called *rgb_yuv.py.*

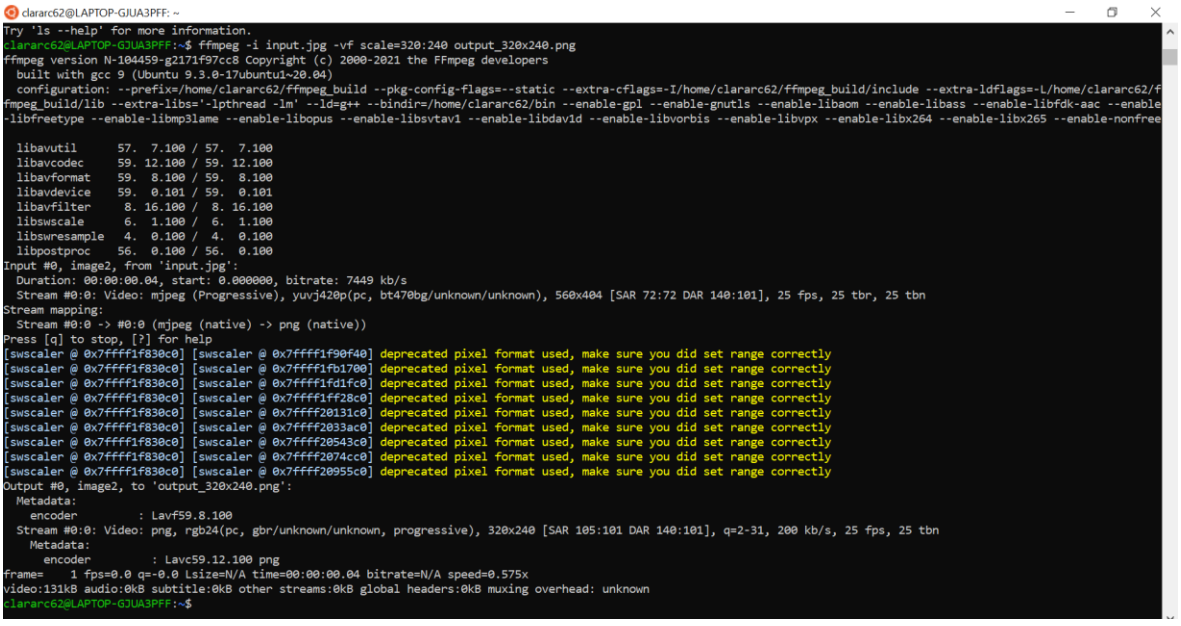
It asks to the user if he wants to convert form RGB to YUV or from YUV to RGB. Then, it asks for the number of each channel and finally shows the conversion.

2) Use ffmpeg to resize images into lower quality.

This exercise is done using the Ubuntu terminal. The used command is:

ffmpeg -i input.jpg -vf scale=320:240 output_320x420.png

The original image (called input.jpg) and the resulting output (called output_320x420.png) can be seen in the folder.

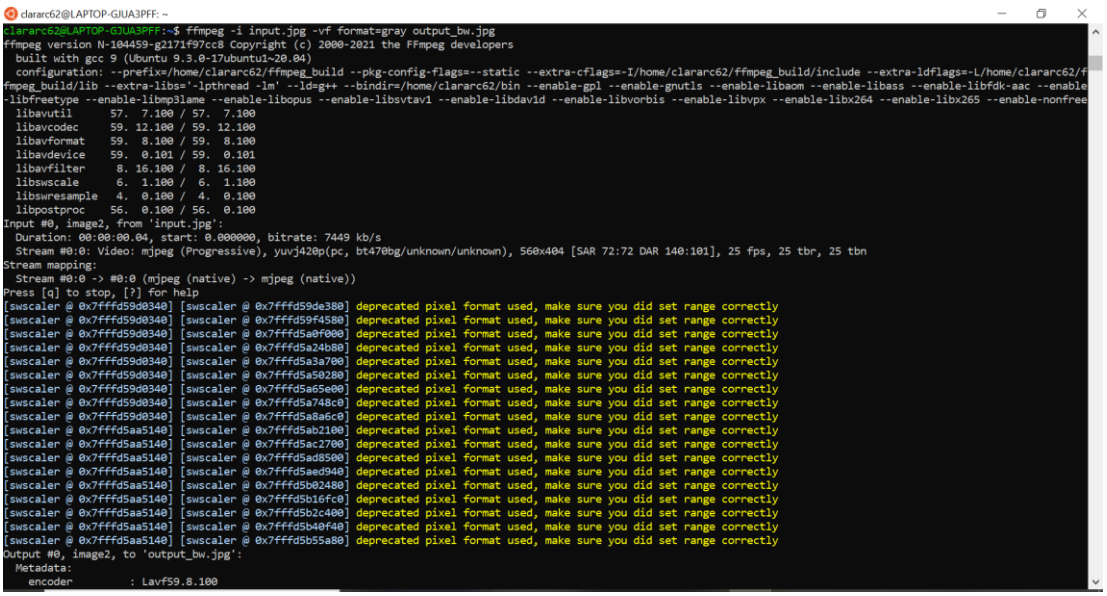


3) Use to transform the image into b/w.

Again, this exercise is done by the Ubuntu terminal. The used command is:

ffmpeg -i input.jpg -vf format=gray output_bw.jpg

By this way we obtain the original image (input.jpg) in black and white scale (output_bw.jpg)



4) Create a script which contains a function which applies a run-length encoding from a series of bytes given.

There are two scripts for this exercise.

The first one is a script called *runlength_encoding.py.* The algorithm takes an array (entered by the user) and each run of 0s is replaced by two characters in the compressed file: a zero to indicate that compression is occurring, followed by the number of zeros in the run.

On the other hand, in the script *RL.py* there is the run-length encoding algorithm that asks for strings to the user, and each of them that is repeated consecutively is counted and added to the final result as well as the corresponding string.

5) Create a script which can convert, can decode (or both) an input using the DCT. Not necessary a JPG encoder or decoder. A script only about DCT is OK too

This exercise is done in a PyCharm script called *DCT_array.py.* It takes an array entered by the user, and shows the original one, the DCT and the IDCT.

Furthermore, in *DCT_images.py* is done the conversion for JPG images. Hence, there are two functions, the 2D DCT and 2D IDCT. The input image is first converted to black and white scale and then the corresponding functions are applied. Finally, the original image and the reconstructed one are showed.