Àlex Serrano Marull
230278

P1 – JPEG AND FFMPEG

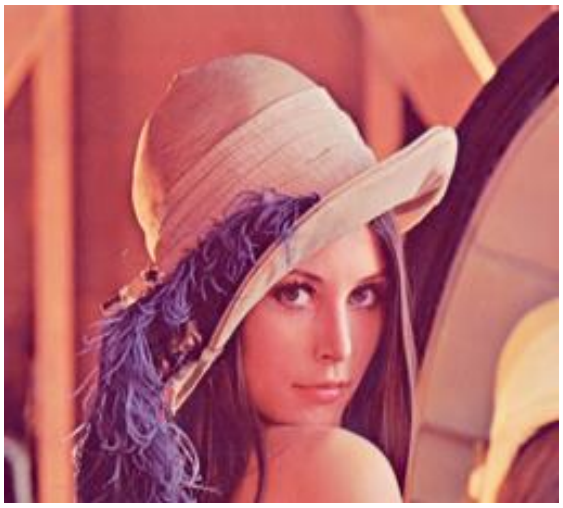
2) Use FFmpeg to resize images to lower quality.

In order to complete this task, I used an FFmpeg command that allowed us to resize the image to a lower quality. Namely, I applied a scaling factor of 0.5 (256 pixels from the original 512).

```
alexserrano@LAPTOP-SGMM4KG8:/mnt/c/users/34645/desktop$ ffmpeg -i Lenna.png -vf scale=256:-1 Lenna_rezised.png
ffmpeg version 4.4.2-0ubuntu0.22.04.1 Copyright (c) 2000-2021 the FFmpeg developers
  built with gcc 11 (Ubuntu 11.2.0-19ubuntu1)
  configuration: --prefix=/usr --extra-version=0ubuntu0.22.04.1 --toolchain=hardened --libdir=/usr/lib/x86_64-linux-gnu
--incdir=/usr/include/x86_64-linux-gnu --arch=amd64 --enable-gpl --disable-stripping --enable-gnutls --enable-ladspa --e
nable-libaom --enable-libass --enable-libbluray --enable-libbs2b --enable-libcaca --enable-libcdio --enable-libcodec2 --
enable-libdav1d --enable-libflite --enable-libfontconfig --enable-libfreetype --enable-libfribidi --enable-libgme --enab
le-libgsm --enable-libjack --enable-libmp3lame --enable-libmysofa --enable-libopenjpeg --enable-libopenmpt --enable-libo
pus --enable-libpulse --enable-librabbitmq --enable-librubberband --enable-libshine --enable-libsnappy --enable-libsoxr
--enable-libspeex --enable-libsrt --enable-libssh --enable-libtheora --enable-libtwolame --enable-libvidstab --enable-li
bvorbis --enable-libvpx --enable-libwebp --enable-libx265 --enable-libxml2 --enable-libxvid --enable-libzimg --enable-li
bzmq --enable-libzvbi --enable-lv2 --enable-omx --enable-openal --enable-opencl --enable-opengl --enable-sdl2 --enable-p
ocketsphinx --enable-librsvg --enable-libmfx --enable-libdc1394 --enable-libdrm --enable-libiec61883 --enable-chromaprin
t --enable-frei0r --enable-libx264 --enable-shared
  libavutil      56. 70.100 / 56. 70.100
  libavcodec     58.134.100 / 58.134.100
  libavformat    58. 76.100 / 58. 76.100
  libavdevice    58. 13.100 / 58. 13.100
  libavfilter     7.110.100 /  7.110.100
  libswscale      5.  9.100 /  5.  9.100
  libswresample   3.  9.100 /  3.  9.100
  libpostproc    55.  9.100 / 55.  9.100
Input #0, png_pipe, from 'Lenna.png':
  Duration: N/A, bitrate: N/A
  Stream #0:0: Video: png, rgb24(pc), 512x512, 25 fps, 25 tbr, 25 tbn, 25 tbc
File 'Lenna_rezised.png' already exists. Overwrite? [y/N] Y
Stream mapping:
  Stream #0:0 -> #0:0 (png (native) -> png (native))
Press [q] to stop, [?] for help
Output #0, image2, to 'Lenna_rezised.png':
  Metadata:
    encoder         : Lavf58.76.100
  Stream #0:0: Video: png, rgb24(pc, gbr/unknown/unknown, progressive), 256x256, q=2-31, 200 kb/s, 25 fps, 25 tbn
    Metadata:
      encoder         : Lavc58.134.100 png
frame=    1 fps=0.0 q=-0.0 Lsize=N/A time=00:00:00.04 bitrate=N/A speed=1.19x
video:178kB audio:0kB subtitle:0kB other streams:0kB global headers:0kB muxing overhead: unknown
```

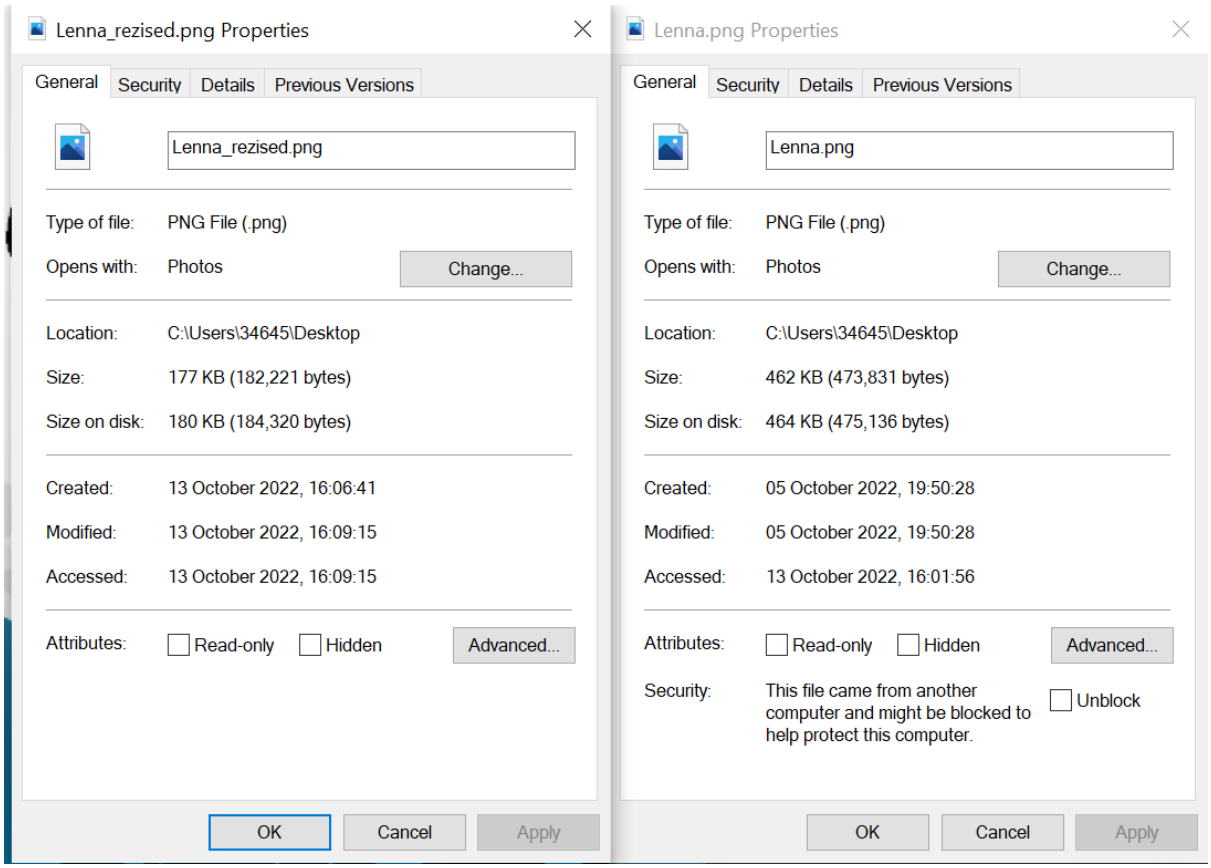
These were the results of performing this operation:

Lenna.png

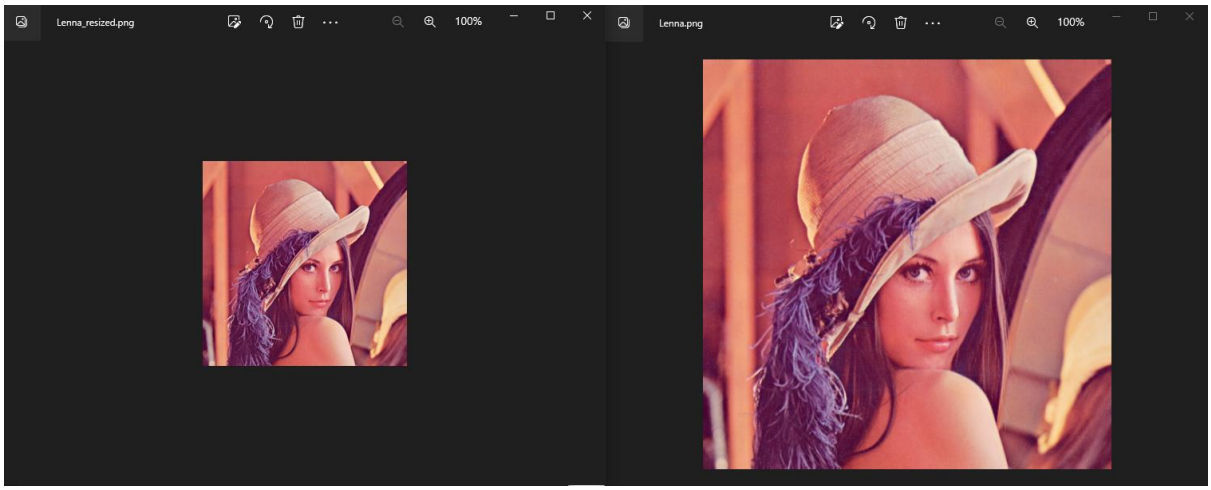


Lenna_resized.png

The change might not be appreciated if no zoom is applied, but by checking their properties I made sure that their quality was different:



By opening them together, we can see that indeed, one is half of the other:



3) Use FFmpeg to transform the Lenna image into b/w. Do the hardest compression you can and comment on the results.

For this exercise, I found two ways of converting an image to B/W using FFmpeg. The first one was by desaturating the image (setting the hue to 0) and the other was by simply transforming it to grayscale. Both commands are shown as follows:

```
alexserrano@LAPTOP-SGMM4KG8:/mnt/c/users/34645/desktop$ ffmpeg -i Lenna.png -vf hue=s=0 Lenna_bw.png
```

```
alexserrano@LAPTOP-SGMM4KG8:/mnt/c/users/34645/desktop$ ffmpeg -i Lenna.png -vf format=gray Lenna_grayscale.png
```

And the corresponding results:

Lenna desaturated (bw)

Lenna grayscale

However, since we were asked to perform the hardest compression we could, I also tried doing the same operations to the already compressed images. Meaning, I applied the desaturation to the grayscaled image and vice versa. It only worked when applying the grayscale to the desaturated image. Seeing the results, I can say that the grayscale operation does a harder compression than the desaturation one