Laia Duch Soler

Report Lab 2

In this lab, I have created a script called *convert_to_mpeg.py* where we can find all the necessary functions to perform lab 2. We also have a script called *rgb_yuv.py*, which contains all the implemented functions of lab 1. To verify that I have correctly performed the inheritance from another Python script, I am using the image 'eddie.jpg' again.

Task 1

The objective of the first task is to convert the video into an .mpeg file. The function *convert_to_mpeg* takes an input video and uses ffmpeg commands to convert the video into the desired format. The *parse_video_info* function parses video information using ffmpeg and saves it in a .txt file. The new video is saved as 'BBB.mpeg'.

Task 2

In this task, I create a function called *modify_resolution* where the resolution of the input video is modified. To perform the modification, ffmpeg is used. In this function, new values for height and width are provided as input parameters. The new video is saved with the name 'BBB_modified.mpeg'.

Task 3

In this task, I have created a function that allows you to change the chroma subsampling. To do so, we use the following ffmpeg command:

```
ffmpeg -i input.mp4 -c:v libx264 -vf format=yuv420p output.mp4
```

In this command, the video is encoded to 8-bit 4:2:0, meaning Cb' and Cr' are each subsampled at a factor of two, both horizontally and vertically."

The video is saved as 'BBB_subsampling.mp4'.

Task 4

In this task, we create a new function where an input video is processed to extract and display essential information. This function will output details such as width, height, codec name, bit rate, and duration of the video. To test the function, the original video ('BigBuckBunny.mp4') is used as input.

Task 5

We want to inherit functions from Lab 1. To do this, we import rgb_yuv to access the functions from this script. In the script, we can see how we call one of the functions from Lab 1.