**Student:** Le Minh Bang 黎明朋

**ID:** 313540015

**Video Compression Homework 1 - Color Space Conversion**

1. **Overview**

This assignment delves into the color transformation process of the "lena.png" image across the RGB, YUV, and YCbCr color models. The conversions were executed using Python without relying on pre-built libraries for color conversions. The output consists of eight grayscale images, each corresponding to an individual color channel (R, G, B, Y, U, V, Cb, and Cr). This task aims to deepen our understanding of how various color spaces depict image data and the benefits of isolating brightness (luminance) from color (chrominance) in image processing workflows.

Color spaces are a critical aspect of digital imagery, influencing how visuals are stored, processed, and displayed. While RGB is the standard for screens, it is not always ideal for compression or transmission tasks. Models like YUV and YCbCr distinguish brightness from color data, which enhances compression efficiency. This report outlines the manual process of converting the image from RGB to YUV and YCbCr and displays the grayscale representations of each channel.

1. **Insight into Color Spaces**
   1. **RGB Model**

* **Definition**: This model uses Red, Green, and Blue to represent colors.
* **Range**: Each color is between 0 and 255 in 8-bit images.
* **Use Case**: Commonly employed for display purposes, though it could be more suitable for applications needing the separation of brightness and color.
  1. **YUV Model**
* **Components**:
  + **Y (Luminance)**: Describes brightness levels.
  + **U and V (Chrominance)**: Convey color information.
* **Advantages**: This model enhances compression by decoupling brightness from color, as the human eye is more sensitive to brightness than color differences.
  1. **YCbCr Model**
* **Components**:
  + **Y (Luminance)**: Equivalent to Y in the YUV model.
  + **Cb and Cr (Chrominance)**: Represent the color difference components for blue and red, respectively.
* **Application**: Extensively utilized in video compression formats like JPEG and MPEG.
  1. **RGB to YUV and YCbCr Transformations**

The conversions between RGB, YUV, and YCbCr were conducted using specific mathematical formulas tailored for each color model.

1. **Python Implementation**
   1. **Libraries**

* **NumPy**: Employed for numerical operations.
* **Pillow (PIL)**: Used to handle image processing tasks.
  1. **Steps**

1. **Loading the Image**: The image "lena.png" was loaded via the PIL library and converted into a NumPy array.
2. **Channel Separation**: The R, G, and B channels were extracted from the image array.
3. **Conversion Formulas**: The conversion from RGB to YUV and YCbCr was executed using predefined mathematical expressions.
4. **Normalization and Clipping**: The U and V values were adjusted by adding 128, and all channel values were clipped within the range of 0–255.
5. **Data Conversion**: Arrays were converted into 8-bit unsigned integers for proper image representation.
6. **Saving the Images**: The grayscale images were saved for each channel.
7. **Output**

A total of eight grayscale images were produced, each representing one of the RGB, YUV, and YCbCr components:

1. **R.png**: Red channel.
2. **G.png**: Green channel.
3. **B.png**: Blue channel.
4. **Y.png**: Luminance component.
5. **U.png**: Chrominance U from YUV.
6. **V.png**: Chrominance V from YUV.
7. **Cb.png**: Chrominance Cb from YCbCr.
8. **Cr.png**: Chrominance Cr from YCbCr.
9. **Analysis**
   1. **Luminance (Y) Channel**

* The Y channel resembles a standard grayscale image, encapsulating brightness, which plays a significant role in human vision.
  1. **Chrominance Channels**
* **U and V (YUV)**: These channels represent the color variations, showing color differences with reduced intensity since humans are less sensitive to changes in chrominance.
* **Cb and Cr (YCbCr)**: These channels operate similarly to U and V, though they use different coefficients. They are vital in compression, where chrominance can be reduced.
  1. **RGB Channels**
* Each channel visualizes the contribution of red, green, and blue to the overall image, showcasing how the primary colors form the final image.

1. **Conclusion**

This exercise of manually transforming "lena.png" into different color spaces and generating individual grayscale images provided valuable insights into image representation. Separating luminance from chrominance is particularly beneficial for tasks like compression, where the bandwidth can be optimized by leveraging human visual system characteristics. Understanding these color models is crucial for anyone working in digital imaging and related areas, as the choice of color space significantly impacts the efficiency and quality of image processing operations.