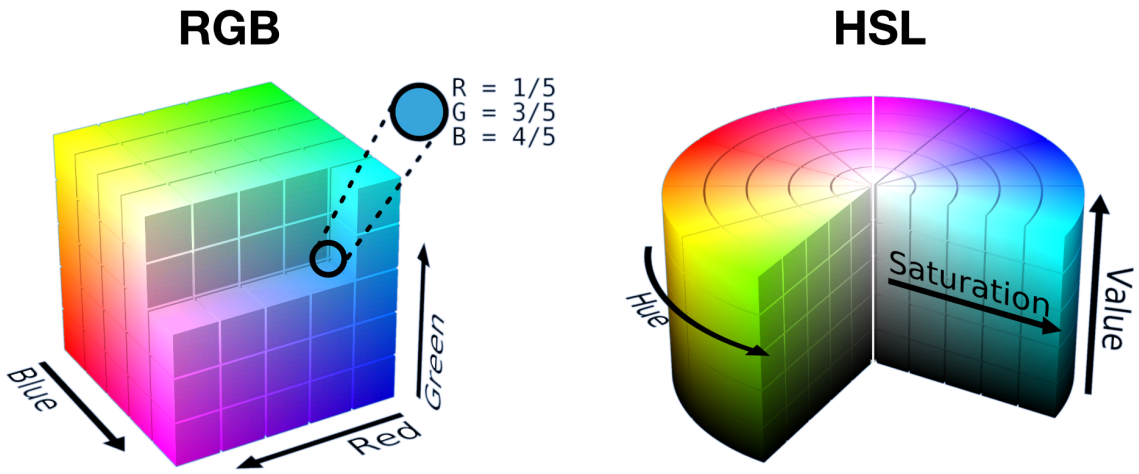
Image Processing for Computer Vision
Session 3

collected

Topics

- Color Space
- Image Formats
- Libraries for working with image
- Loading images
- Color Space Conversion

Color Space



collected

A **color space** is a specific organization or representation of colors, describing how colors are encoded as numerical values. It defines a range of colors and how they can be represented in a digital image.
Each color space is suited to specific applications, and the choice of color space significantly impacts image processing tasks.

Common Color Spaces

1. **Grayscale (single channel)**
2. **RGB (Red, Green, Blue)**
3. **RGBA (A- Alpha) -** Useful for object segmentation, overlays, and blending in CV tasks.
4. **HSV (Hue, Saturation, Value) -** ideal for detecting objects of a specific color under different lighting conditions, used in color based detection and segmentation
5. **YUV / YCbCr Y:** Luminance U/Cb: Chrominance Blue V/Cr: Chrominance Red Projection - effective in low light and shadows
6. **Lab (CIE-Lab) L:** Lightness(0-100) a: Green-to-red(-100, +100). b: Blue-to-yellow(-100, +100).

Why Use Different Color Spaces?

1. **Image Processing:**
    - Tasks like edge detection or filtering often work better in Grayscale.
    - HSV is useful for isolating colors.
2. **Compression:**
    - YUV reduces data by focusing on luminance over chrominance.

3. **Perceptual Accuracy:**
    - Lab is used when human perception matters.

Image Formats

collected

Images come in various formats, each with unique properties suited for specific use cases. JPEG and PNG are the most commonly used formats.

JPEG (JPG): Lossy compression, commonly used for photos, reduces file size at the cost of quality.

PNG: Lossless compression, supports transparency (RGBA), useful for overlays, icons, and web graphics.

Necessary Libraries for Image in python:

There are many libraries in python to work with images. Most common libraries are: OpenCV, Matplotlib, Pillow, Skimage etc.

OpenCV: used for computer vision and image processing, transformations, edge detection, feature extraction etc.

Matplotlib: Visualization and plotting library, with support for displaying images, annotating, adding overlays etc.

Pillow: General-purpose image manipulation.

Scikit-Image (skimage): image processing and feature extraction

Image Loading Using OpenCV

Syntax: cv2.imread(<image_path>)

Image path: location of the image
There are two types of paths that you can use:

1. Relative path: location of your image relative to your working/current directory
2. Absolute path: location of your image on the system

Displaying loaded image:
For plotting the loaded image we will use matplotlib.

Keep in mind:

1. Be sure about the image location you are trying to load
2. Image is loaded as numpy array
3. OpenCV loads an image in BGR format by default
4. Matplotlib show an image in 'Viridis' color map by default

Color Space conversion

collected

Syntax: image_cs = cv2.cvtColor(<image>, <conversion_code>)
Here image is the loaded image array and image_cs color space converted image.

Example: COLOR_BGR2RGB this is used to convert the BGR image to RGB
Most commonly used conversion codes are:
COLOR_BGR2RGB
COLOR_RGB2GRAY
COLOR_BGR2GRAY

You will find a list of all conversion codes for openCV here .