Computer Vision

e-Yantra , SVPCET Nagpur

Agenda

- Matplotlib
- Color Model
- 2D transformation

Matplotlib

- Matplotlib is a python package use for 2D graphic
- Matplotlib is a comprehensive library primarily used for creating static, animated, and interactive visualizations.
- Matplotlib was created by John D. Hunter.
- Matplotlib is open source and we can use it freely.

What is Data visualization?

- Presentation of data in graphical or pictorial format

Why we need data visualization?

- Human brain can process information easily when it is pictorial or graphical form
- It allows us to quickly interpret the data and adjust different variables to see their effect

Color Models

- A **Color Model** is a mathematical model describing the way colors can be represented as tuples of numbers, for example: triples in RGB
- Color spaces allows us to reproduce representations of color. Commons color spaces are:
  - RGB
  - CMYK
  - HSV
  - HSL
  - HSI

- Different color spaces have different applications and they cover varying degree of the color spectrum. For example, **RGB** is often used for Plasma Display or LED display while **CMYK** is the standard in printing industry

Converting Color Models:

- There are more than 150 color-space conversion methods available in OpenCV.
- Here we will use **cv2.cvtColor(src, code)**
- This method is used to convert an image from one color space to another.
- Parameters

  src: Source image, a numpy array.

  code: It specifies the TO and FROM for the conversion of color space. It follows the pattern sourceCode2destinationCode.

- Returns

  This method returns the converted image as specified in the code.

Codes belong to the enumeration cv::**ColorConversionCodes**. Some of the several codes available are:

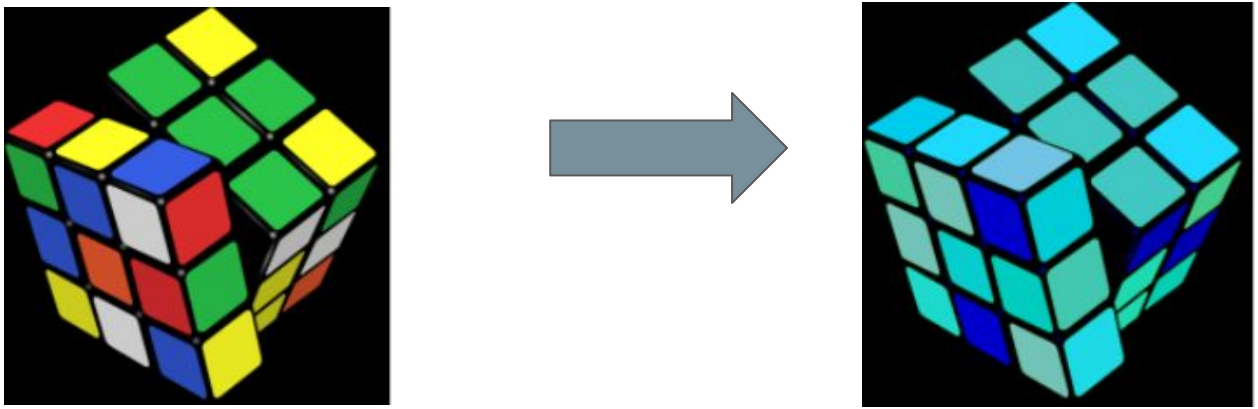
Code	int value	Result
COLOR_BGR2BGRA	0	Add alpha channel to RGB or BGR image
COLOR_RGB2RGBA	0	Add alpha channel to RGB or BGR image
COLOR_BGRA2BGR	1	Remove alpha channel from RGB or BGR image
COLOR_RGBA2RGB	1	Remove alpha channel from RGB or BGR image
COLOR_BGR2RGBA	2	Convert between RGB and BGR color spaces (with or without alpha channel)
COLOR_RGBA2BGR	3	Convert between RGB and BGR color spaces (with or without alpha channel)
COLOR_BGR2GRAY	6	Convert between BGR and grayscale
COLOR_RGB2GRAY	7	Convert between RGB and grayscale
COLOR_BGR2HSV	40	Convert BGR to HSV (hue saturation value)
COLOR_RGB2HSV	41	Convert RGB to HSV (hue saturation value)

All codes have their reverse conversions too (RGBA**2**BGRA <-> BGRA**2**RGBA) which can be obtained easily by interchanging the sourceCode with the destinationCode

❖ By default, OpenCV imports in **BGR** Color Model which stands for **B**lue, **G**reen, **R**ed. It is similar to the RGB Color Model. Some important properties of BGR are:

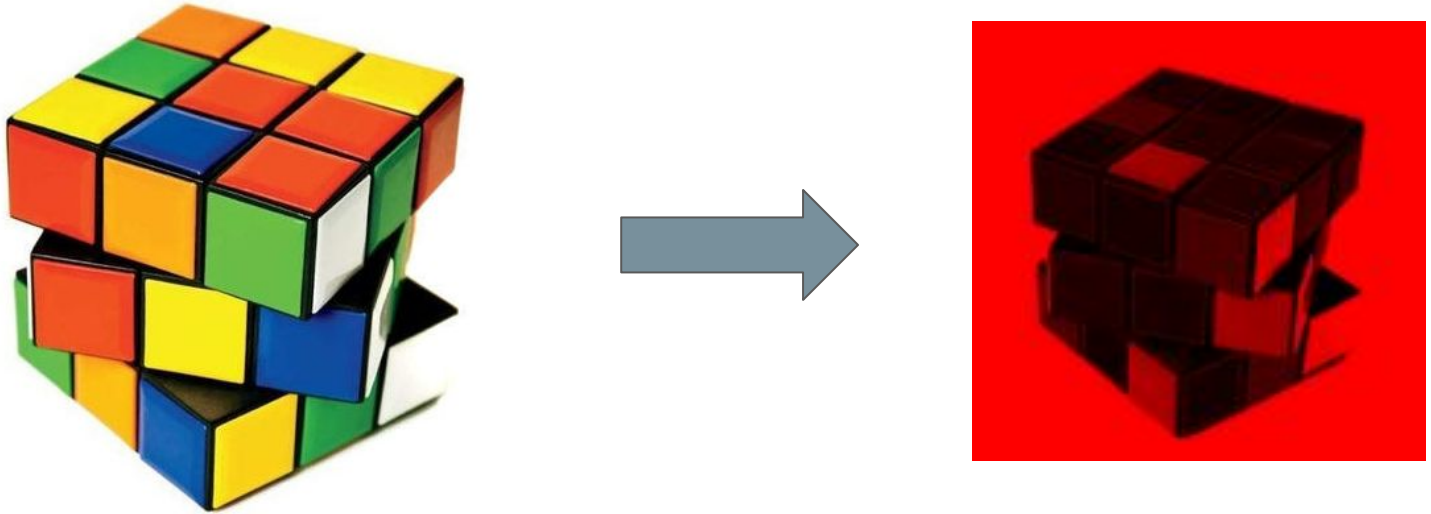
- It is an additive colorspace i.e. colors are obtained by a linear combination of Red, Green, and Blue values
- The three channels are correlated by the amount of light hitting the surface.

❖ Converting **RGB** to **HSV** Color Model which stands for **H**ue, **S**aturation and **V**alue .



Color Slicing

- Color slicing is the process of extracting certain channels from the given image



2 D Transformation

- We can perform various transformations on a 2D image. OpenCV has plenty of functions to help us realize the transformations

- Scaling (Resizing)
- Rotation
- Flip

Scaling

- Image resizing refers to the scaling of images
- It helps in reducing the number of pixels from an image
- It also helps in zooming in images
- OpenCV provides us several interpolation methods for resizing an image.
- OpenCV comes with a function **cv2.resize()** for this purpose

Choice of Interpolation Method for Resizing –

- cv2.INTER_AREA: This is used when we need to shrink an image.
- cv2.INTER_CUBIC: This is slow but more efficient.
- cv2.INTER_LINEAR: This is primarily used when zooming is required. This is the default interpolation technique in OpenCV.

Syntax: cv2.resize(src, dsize [, fx, fy, interpolation = INTER_LINEAR])

Parameters:

src: Source image, a numpy array

dsize: Output image size

fx(optional): scale factor along the horizontal axis

fy(optional): scale factor along the vertical axis

interpolation(optional): interpolation method. There are various Interpolation method available in OpenCV

Rotation

- cv2.rotate() method is used to rotate a 2D array in multiples of 90 degrees.

Syntax: cv2.rotate(src, rotate_code)

Parameters

src: Source image, a numpy array

rotate_code: Code to specify how to rotate the array

Code	Description
cv2.ROTATE_90_CLOCKWISE	Rotate 90° Clockwise
cv2.ROTATE_90_COUNTERCLOCKWISE	Rotate 90° CounterClockwise
cv2.ROTATE_180	Rotate 180°

Flip

- **cv2.flip()** method is used to flip a 2D array.
- The function cv::flip flips a 2D array around vertical, horizontal, or both axes

Syntax : cv2.flip(src, flip_code)

Parameters:

src: Source image, a numpy array

flip_code: Code to specify how to flip the array (*0 means flipping around the x-axis and positive value (for example, 1) means flipping around y-axis. Negative value (for example, -1) means flipping around both axes.*

Code	Description
Negative value (<0)	Flip around the both x-axis and y-axis
0	Flip around the x-axis
Positive value (>0)	Flip around the y-axis