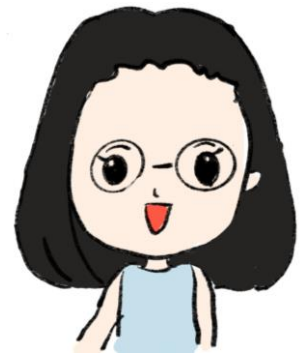




Lab#1

Image Representation, Visualization, and
Arithmetic Operation



Lab#1.1



1.1.1 : Image Color Order with different libraries

1.1.2 : Image Reshape

1.1.3 : Reduce Bit Depth using Quantization

1.1.4 : 3D Image Surface

Lab#1.1

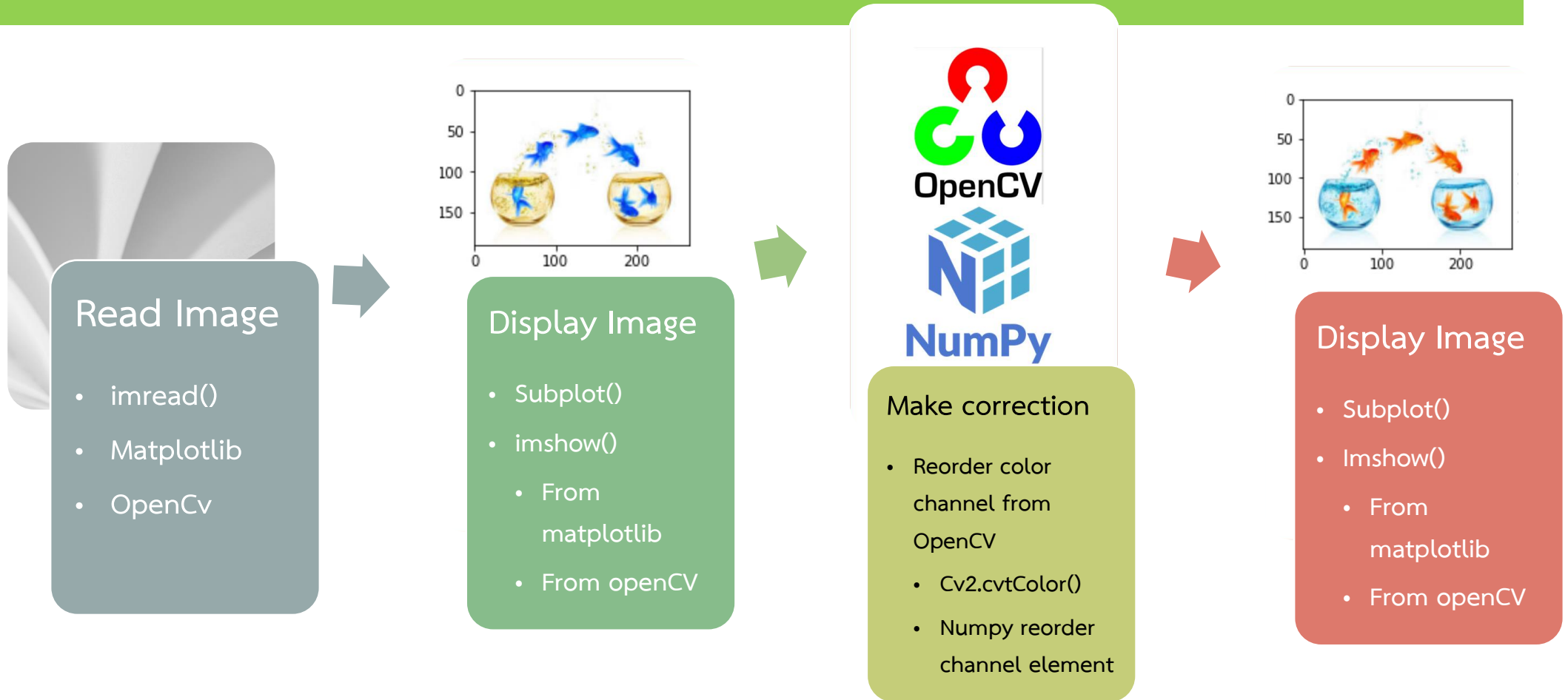


```
import cv2
import numpy as np
import math
import matplotlib.image as mpimg

%matplotlib notebook
from matplotlib import pyplot as plt
from mpl_toolkits.mplot3d import Axes3D
```

1.1.1 : Image Color Order with different libraries

Reorder Image Color



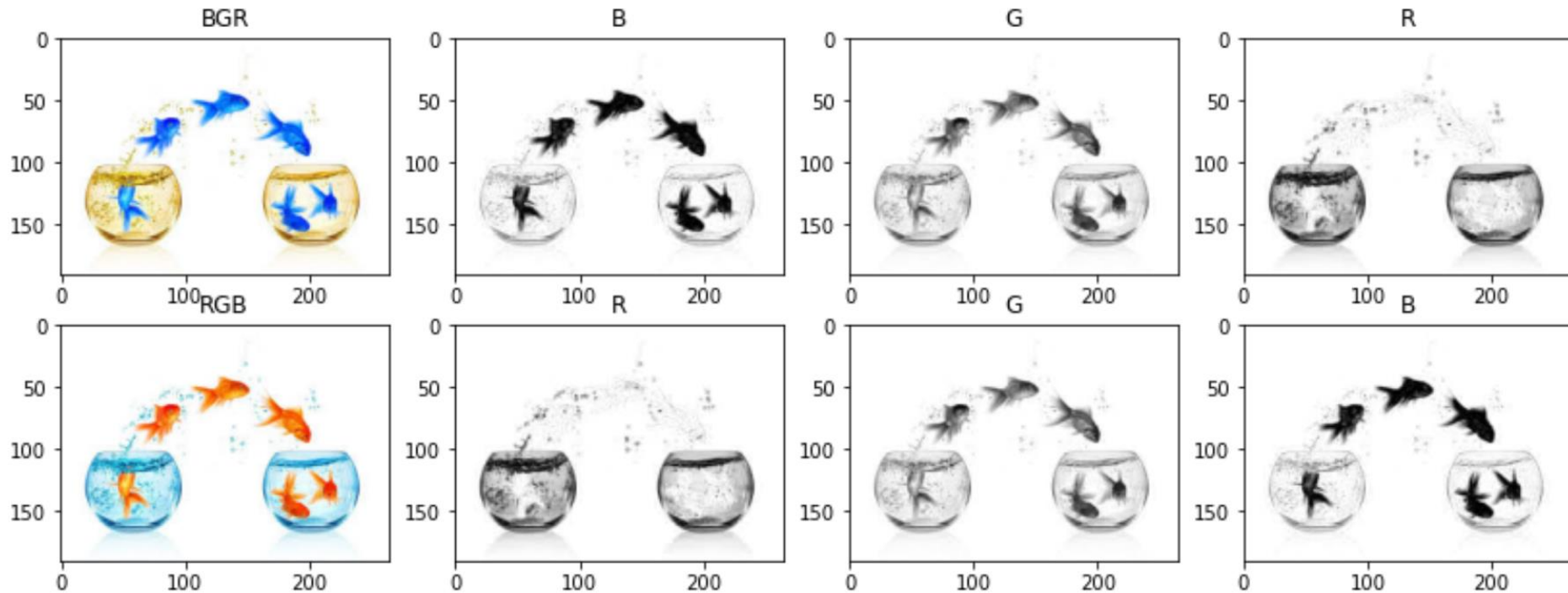
ศึกษาตัวอย่างการอ่านไฟล์ภาพ การแสดงผลภาพ และการจัดเรียง Channel สี ของภาพ

BGR vs RGB (ผลลัพธ์ 1.1.1)

1) แสดงภาพที่อ่านได้จาก OpenCV

From OpenCV
BGR

From numPy
Reorder
channel
Element
RGB



3) แสดงภาพ Xb, Xg, Xr เทียบกับ Xr, Xg, Xb โดยแสดงในรูป subplot เพื่อให้เห็นการเปรียบเทียบชัดเจน

2) แสดงภาพจากการทำ Manual Array Slicing (W) สลับ channel สีเอง

$W[:, :, 0] = X_r$; $W[:, :, 1] = X_g$; $W[:, :, 2] = X_b$

Grayscale image use cmap = 'gray'

1.1.2 : Image Reshape (การจัดโครงสร้างของมิติ Image Array)

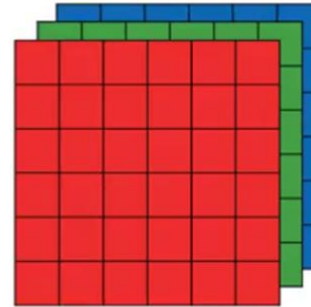
Reorder Channel



Reshape Channel

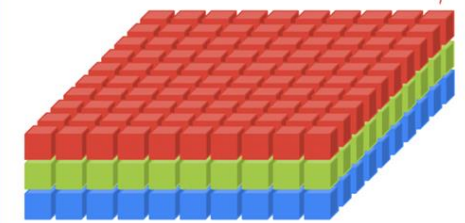
- From (X)
- (height, width, **channel**)
- -> Channel Last

- 1) แสดง shape ของภาพ RGB
- 2) แสดงภาพเฉดสีแดง (Xr)



Numpy

- transpose()
- moveaxis()
- reshape()



Result To (Y)

- (**channel**, height, width)
- -> Channel First

- 3) แสดง shape ของภาพผลลัพธ์
- 4) แสดงภาพเฉดสีแดงจากภาพผลลัพธ์
หลังจากผ่าน 3 ฟังก์ชัน

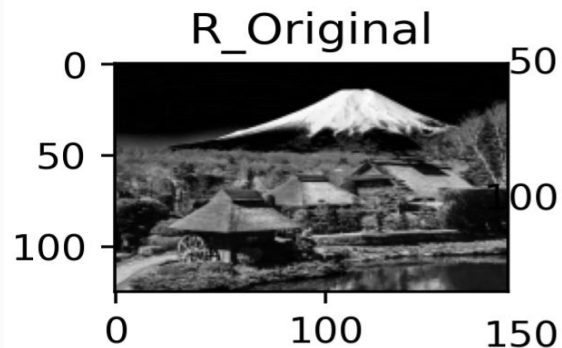
1.1.2 : Image Reshape (ผลลัพธ์ 1.1.2)

$X (H, W, CH) \rightarrow Y (CH, H, W)$

CH Last \rightarrow CH First

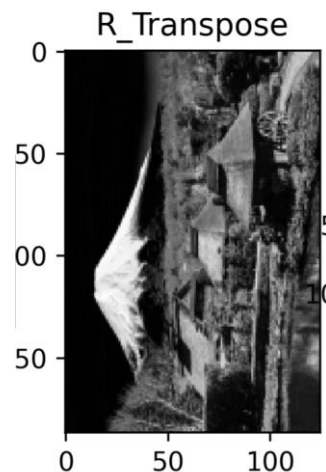
$X (H, W, CH)$

$X (125, 187, 3)$



$Y (CH, W, H)$

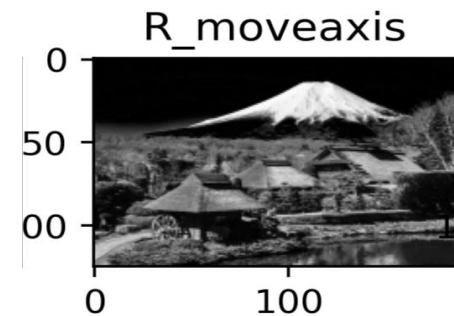
$Y (3, 187, 125)$



`imshow(Yr = Y[0, :, :])`

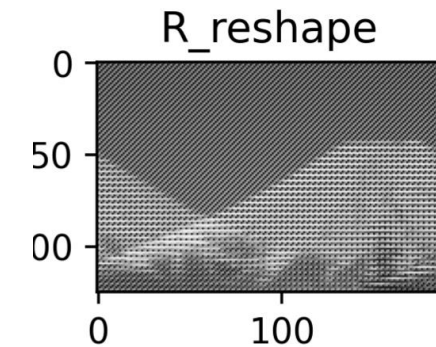
$Y (CH, H, W)$

$Y (3, 125, 187)$



$Y (CH, H, W)$

$Y (3, 125, 187)$



$X.shape = (H, W, CH) = (125, 187, 3)$

$Xr = X[:, :, 0]$

1) แสดง shape ของภาพ RGB (H, W ,3)

2) แสดงภาพเฉดสีแดง (Xr)

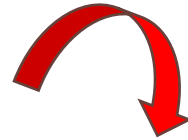
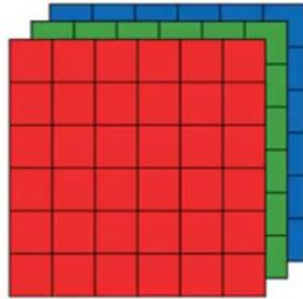
3) แสดง shape ของภาพผลลัพธ์ Y

4) แสดงภาพเฉดสีแดงจากภาพผลลัพธ์ Y

หลังจาก 3 ฟังก์ชัน (Transpose, Reshape, Moveaxis)

1.1.3 : Reduce Bit Depth using Quantization

X (H , W , CH)
X (125 , 187 , 3)



(ผลลัพธ์ 1.1.3)



$$Q_{level} = 2^{Bit_Depth}$$

$$Q = \text{floor}(\text{NormValue}(Si) * Q_{level})$$

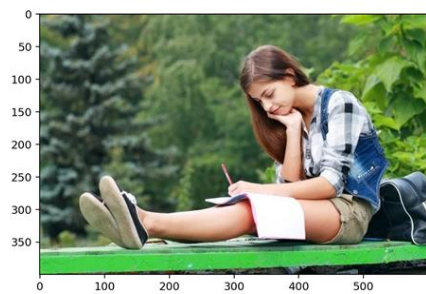
$$= \text{floor}\left(\left(\frac{Si - S_{min}}{S_{max} - S_{min}}\right) * Q_{level}\right)$$

- 1) เปลี่ยนภาพสี เป็นภาพเฉดเทา (RGB to Gray)
- 2) เขียนโปรแกรมคำนวณการปรับจำนวนเฉดสี (quantization ของทุกตำแหน่งพิกเซล ตามสมการ)
- 3) แสดงภาพเฉดสีเทาผลลัพธ์จากการปรับจำนวนเฉดสี (Bit_depth = 8 -> 4)

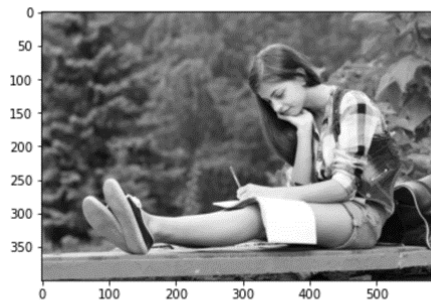
♦ Reduce
Image
Bit Depth
(Quality)

1.1.4 : 3D Image Surface (แสดงพื้นผิว 3 มิติ ของภาพ Grayscale)

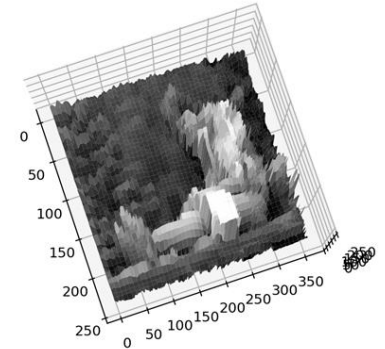
3D Image Surface



2D Image



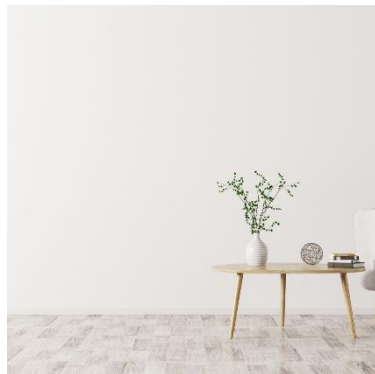
- Convert to grayscale image
- Reduce Dimension
 - `Cv2.resize()`
 - ประมาณ 200 x 200 พิกเซล



3D Surface Display

- Meshgrid สร้าง array พิกัดตำแหน่งพิกเซล
 - `numpy.mgrid()`
- 3D surface plot
 - `plot_surface()`

Lab#1.2



1.2.1 : Color Model Visualization

1.2.2 : Image Addition

1.2.3 : Image Bitwise Operation

Libraries

- `import cv2`
- `import numpy as np`
- `from matplotlib import pyplot as plt`

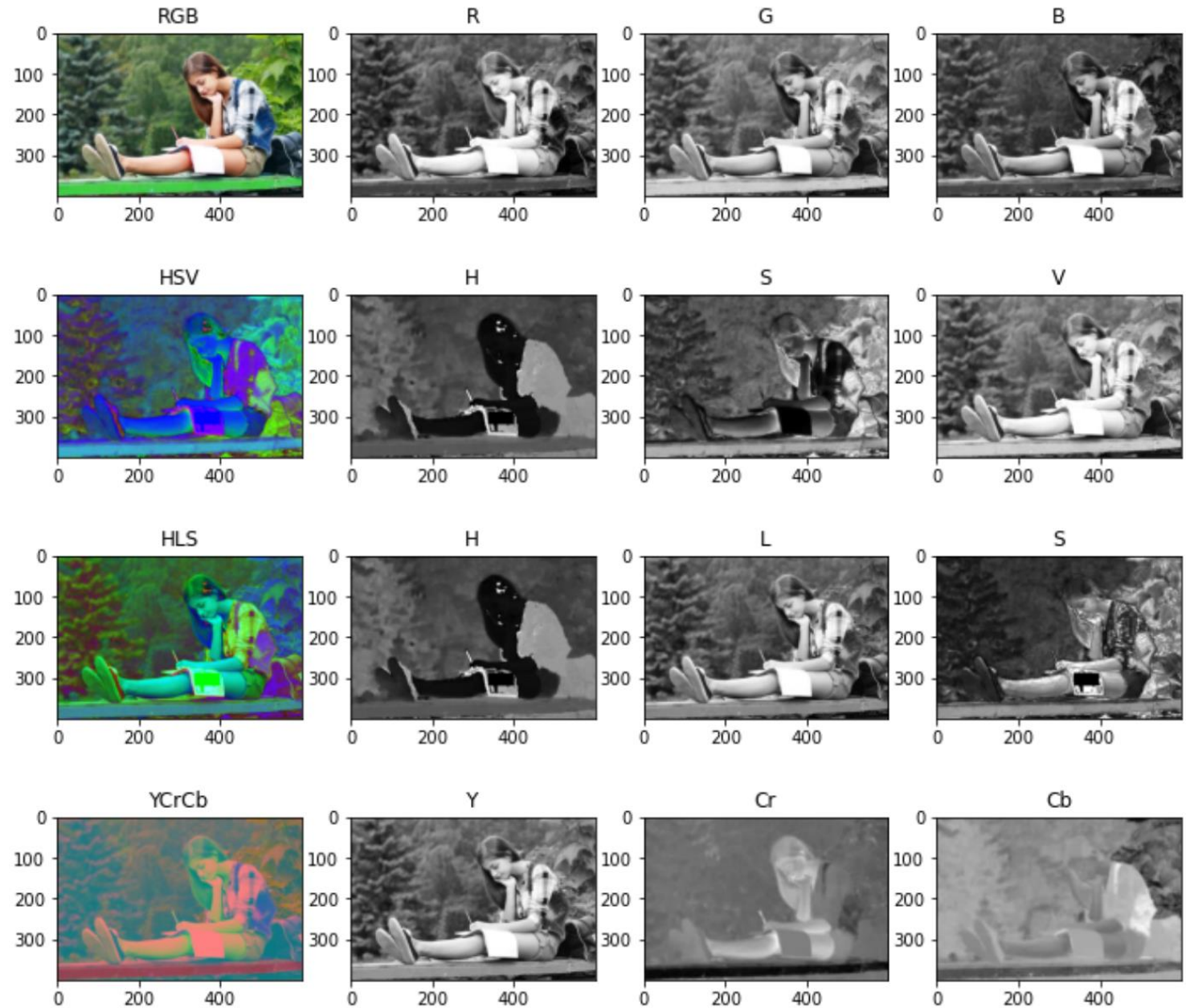
1.2.1 Color Model Visualization

☒ Color Model Conversion

Using `cv2.cvtColor()`

- RGB
- HSV
- HLS
- YCrCb

☒ Visualize each color channel



1.2.2 Image Addition

☑ Create weight array

$W_1, W_2 = [0, 1]$

Steps เพียงพอ จะสร้าง รูป ผลบวก

☑ Write added images to a video file

$$\text{Im_addition} = w_1 \text{Im}_1 + w_2 \text{Im}_2$$

$$w_1 + w_2 = 1.0$$

1

- Create weight array
- $W_1, W_2 = [0, 1]$
- Steps เพียงพอ สร้างภาพผลบวก อย่างน้อย 20 ภาพ

2

- Call OpenCV Video Writer Object (MP4V)
- `cv2.VideoWriter_fourcc()`
- `cv2.VideoWriter()`
- กำหนดพารามิเตอร์ frame rate (fps) ให้เพียงพอ จะเห็นผลการบวกภาพ ที่ค่อยๆเปลี่ยนจากภาพหนึ่ง ไปเป็นอีกภาพหนึ่ง

3

- `Write()` ภาพผลบวกเข้าวิดีโอ ทีละภาพ เข้าวิดีโอ
- กำหนดให้ เขียนให้ภาพที่ 1 ปรากฏ แล้วค่อยๆ เปลี่ยนเป็นภาพที่ 2
- จากนั้น ให้ภาพที่ 2 ค่อยๆ หายไป และเปลี่ยนเป็นภาพที่ 1

4

- `release()`

5

- ระวัง color model ที่ควร write ลง video เมื่อใช้ openCV

1.2.2 Image Addition

- ☑ Create weight array

$w_1, w_2 = [0, 1]$

Steps เพียงพอ จะสร้าง รูป ผลบวก

- ☑ Write added images to a video file

$$\text{Im_addition} = w_1 \text{Im}_1 + w_2 \text{Im}_2$$

$$w_1 + w_2 = 1.0$$



1.2.3 Image Bitwise AND operation

- ☑ Create Image Mask
 - = create zeros array
 - = Fill 255 in Image Mask at Pixel x, pixel y of Object area
- ☑ Use bitwise AND operation `Bitwise_AND()` to mask only object area from original image

$$Im_obj_area = Bitwise_AND(Im, Im_{mask})$$

