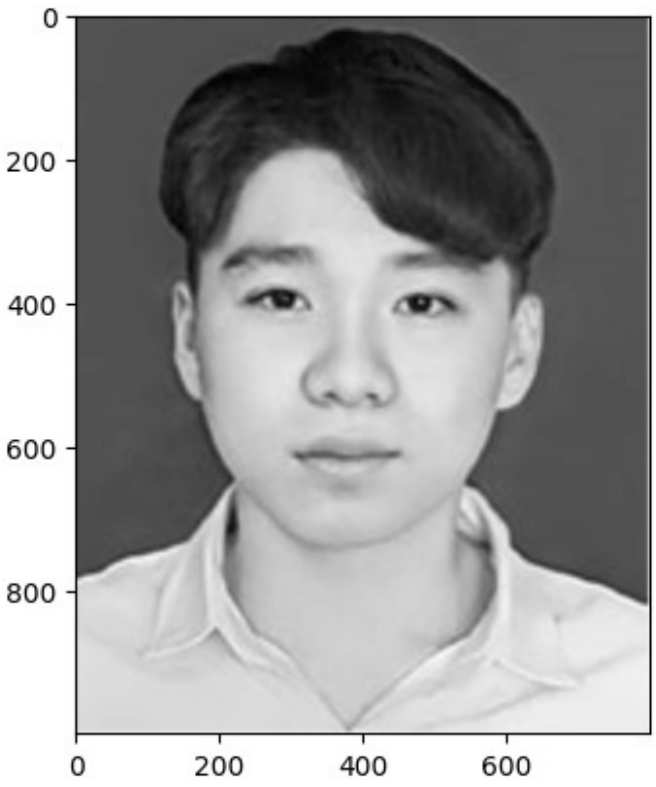
Import libraries

```
import cv2
import matplotlib.pyplot as plt
import numpy as np

path_img = r'D:\Image-Proccessing\pratice\week_1\img\download.png'

bgr_img = cv2.imread(path_img)
gray_img = cv2.imread(path_img, cv2.IMREAD_GRAYSCALE)

print('Kích thước ảnh màu',bgr_img.shape)
print('Kích thước ảnh xám',gray_img.shape)
```

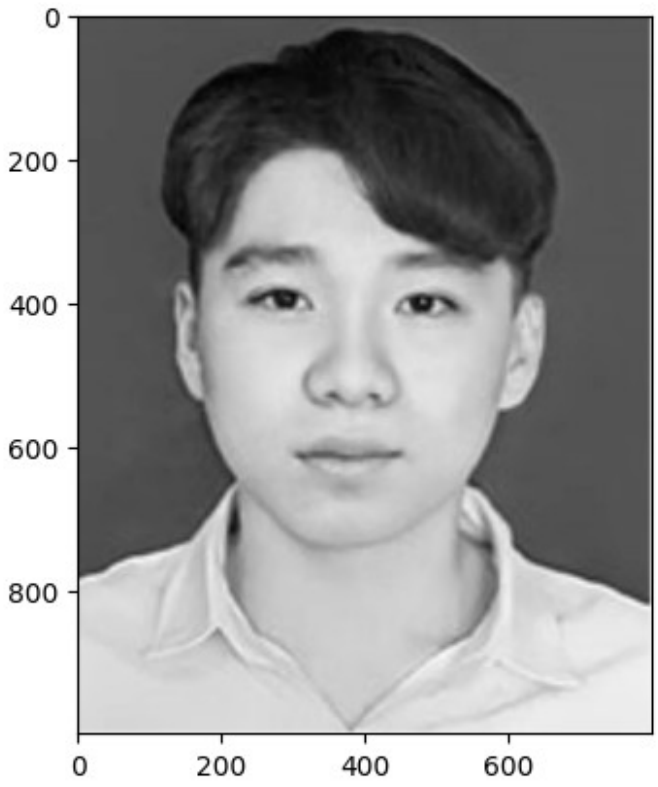
```
Kích thước ảnh màu (1000, 800, 3)
Kích thước ảnh xám (1000, 800)
```

Chuyển đổi ảnh màu về ảnh xám

```
# Cách 1 : Sử dụng hàm cv2.cvtColor()
gray_img_2 = cv2.cvtColor(bgr_img, cv2.COLOR_BGR2GRAY)
plt.imshow(gray_img_2, cmap='gray')

plt.show()
```

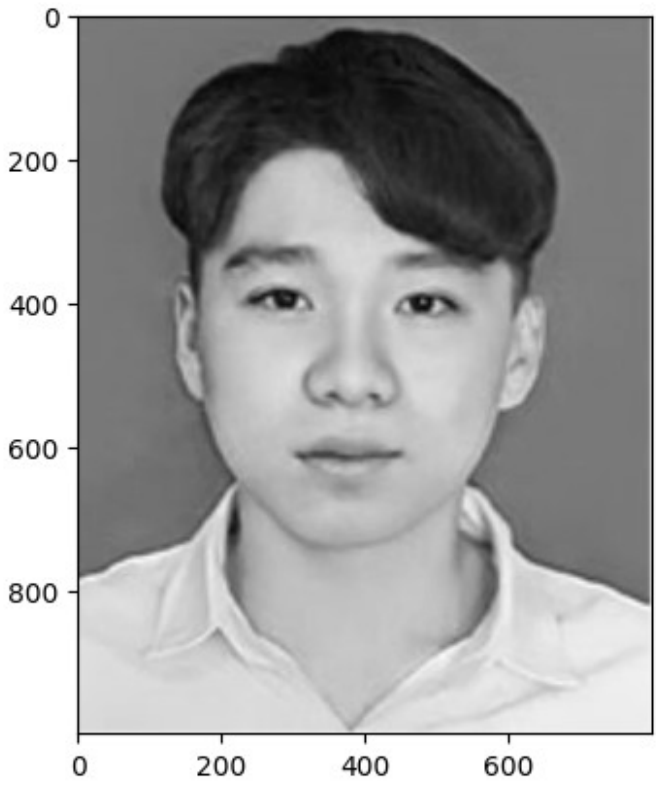



```
#Cách 2 : Sử dụng hàm cv2.imread() với flag=zero
gray_img_3 = cv2.imread(path_img, 0)
plt.imshow(gray_img_3, cmap='gray')
plt.show()
```

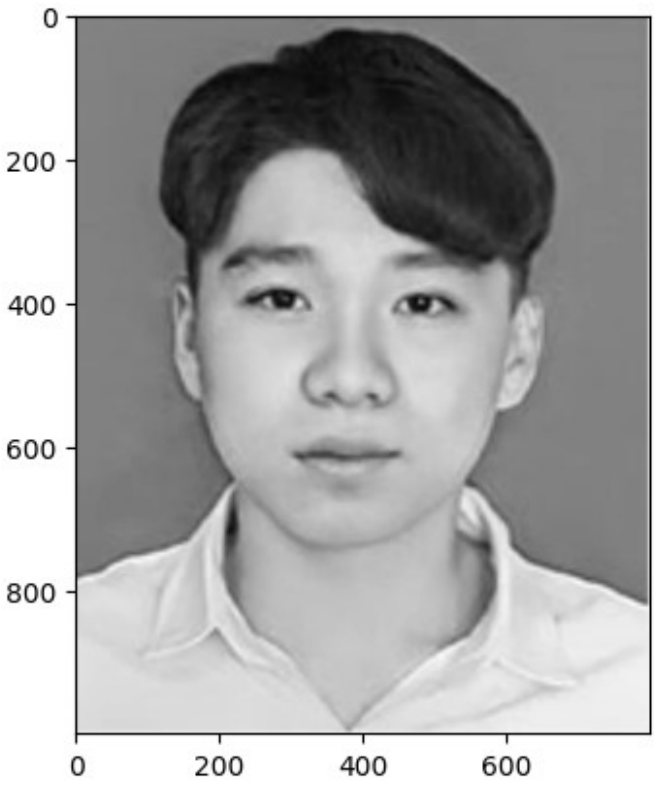



```
# Cách 3 : Phương pháp trung bình
image = cv2.imread(path_img)
rows, cols, _ = image.shape
gray_image = np.zeros((rows, cols), dtype=np.uint8)

for i in range(rows):
    for j in range(cols):
        # Lấy giá trị màu từ ảnh gốc
        blue, green, red = image[i, j]

        # Tính giá trị ảnh xám dựa trên công thức trọng số
        gray_value = int(0.07 * red + 0.72 * green + 0.21 * blue)

        # Gán giá trị vào ảnh xám
        gray_image[i, j] = gray_value

plt.imshow(gray_image, cmap='gray')
```

```
<matplotlib.image.AxesImage at 0x2571ce90d40>
```

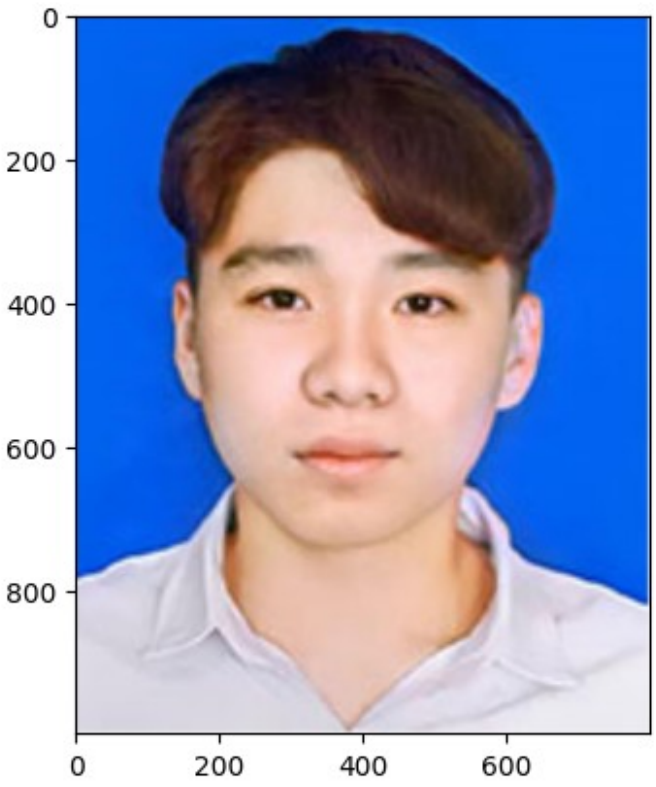



```python
# Cách 4 : phương pháp linear combination
def rgb2gray(rgb):
    # np.dot thực hiện phép nhân điểm giữa mảng RGB và vector trọng số [0.2989, 0.5870, 0.1140]
    # Trọng số này tương ứng với độ nhạy của mắt người đối với các kênh màu đỏ, xanh lá, và xanh dương.
    return np.dot(rgb[...,:3], [0.2989, 0.5870, 0.1140])


img_gray = rgb2gray(bgr_img)

plt.imshow(img_gray, cmap='gray')
plt.show()
```

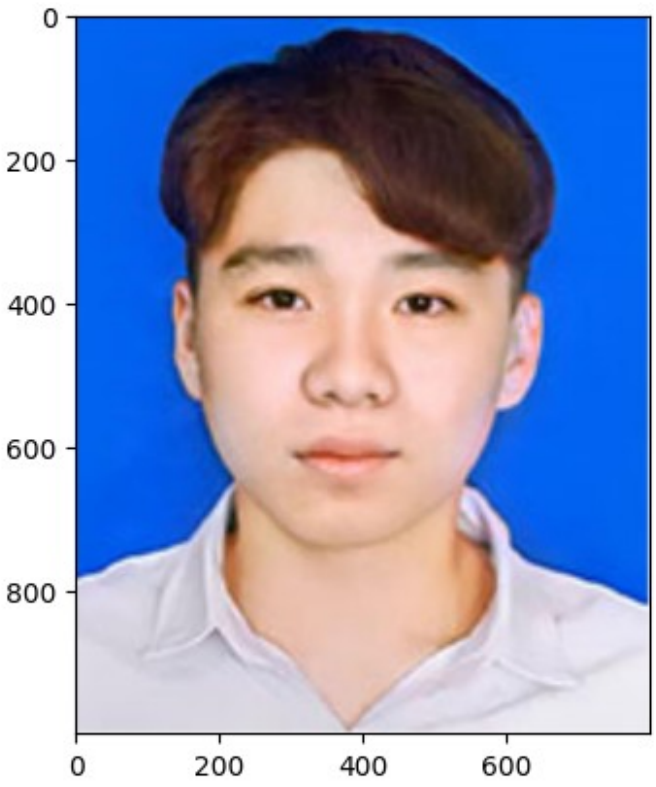


BGR to RGB conversion

```
img = cv2.imread(path_img)
img = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
plt.imshow(img)
plt.show()
```



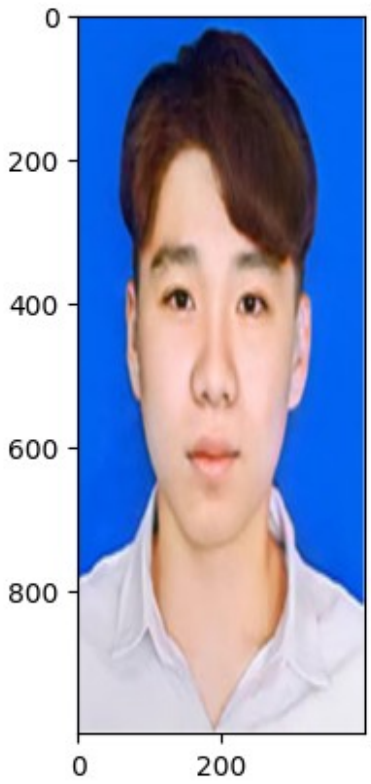

```
# BGR to RGB conversion
img_bgr = cv2.imread(path_img)
img_rgb = cv2.cvtColor(img_bgr, cv2.COLOR_BGR2RGB)
plt.imshow(img_rgb)
plt.show()
```



RESIZE image

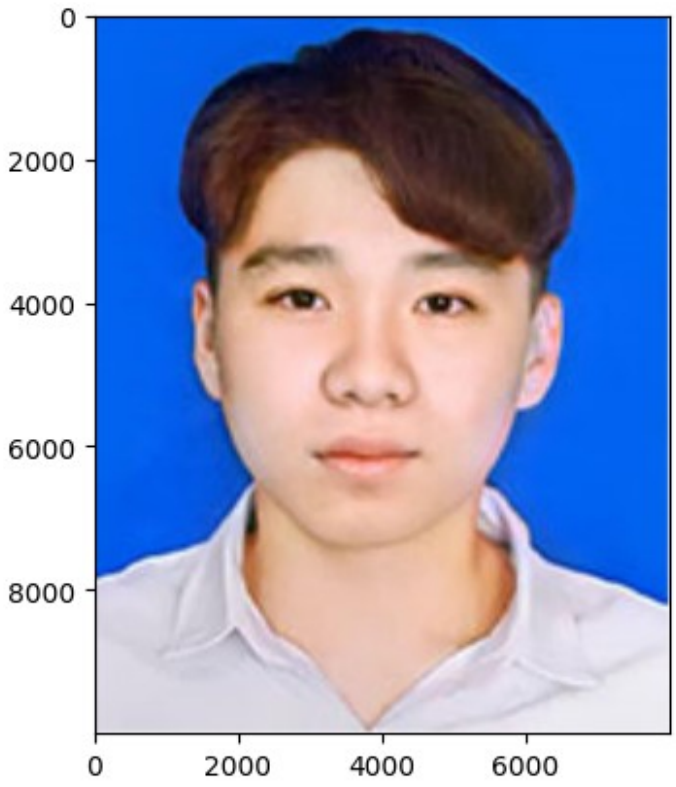
```
fx = 0.5
fy = 1.0
img_resized1 = cv2.resize(img_rgb, dsize=None, fx=fx, fy=fy)

print('Kích thước ảnh sau khi resize:', img_resized1.shape)
plt.imshow(img_resized1)
plt.show()
```

```
Kích thước ảnh sau khi resize: (1000, 400, 3)
```

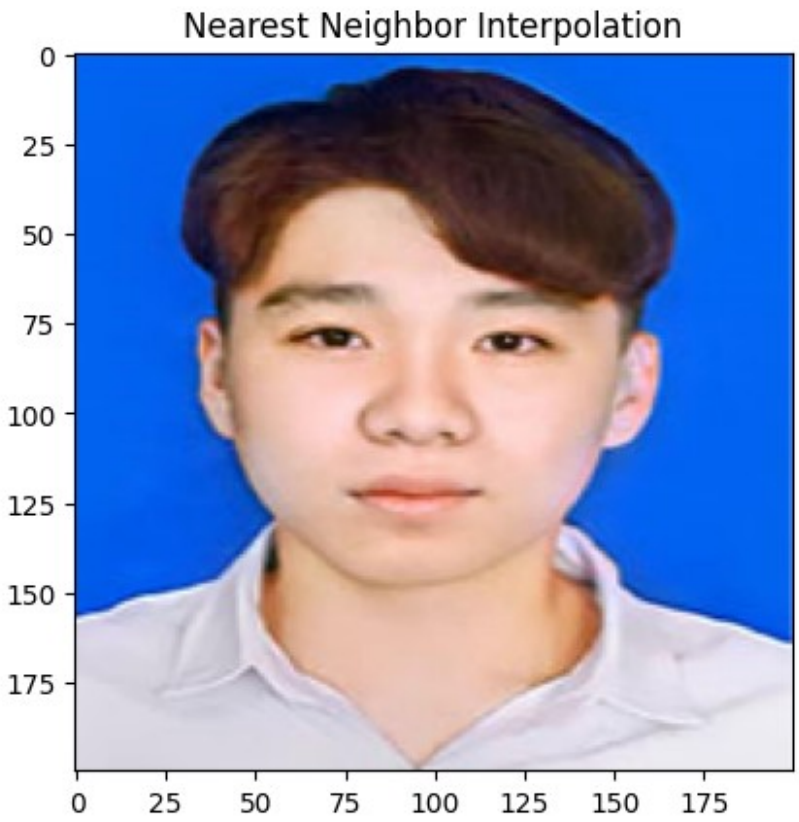



```
# resize ảnh với tỉ lệ scale_percent
def resize(img, scale_percent):
    width = int(img.shape[1] * scale_percent / 100)
    height = int(img.shape[0] * scale_percent / 100)
    dim = (width, height)
    return cv2.resize(img, dim, interpolation = cv2.INTER_AREA)
img_resized = resize(img_rgb, 1000)
print('Kích thước ảnh sau khi resize:', img_resized.shape)
plt.imshow(img_resized)
plt.show()
```

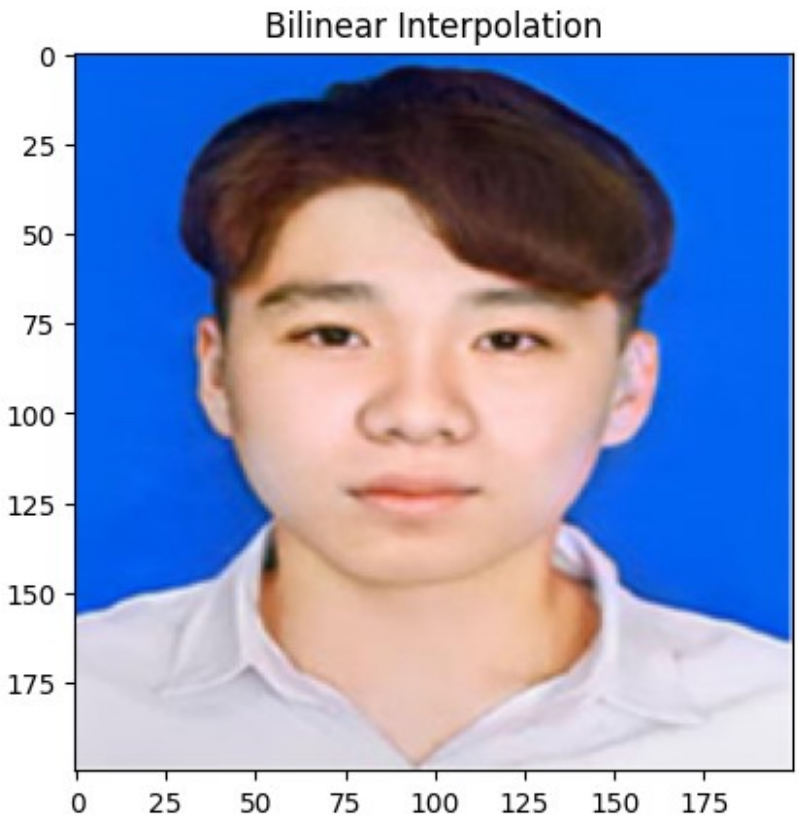
```
Kích thước ảnh sau khi resize: (10000, 8000, 3)
```




```
# Thay đổi kích thước ảnh với Nearest Neighbor Interpolation
img_resized_nn = cv2.resize(img_rgb, (200, 200),
interpolation=cv2.INTER_NEAREST)

plt.imshow(img_resized_nn)
plt.title("Nearest Neighbor Interpolation")
plt.show()
```

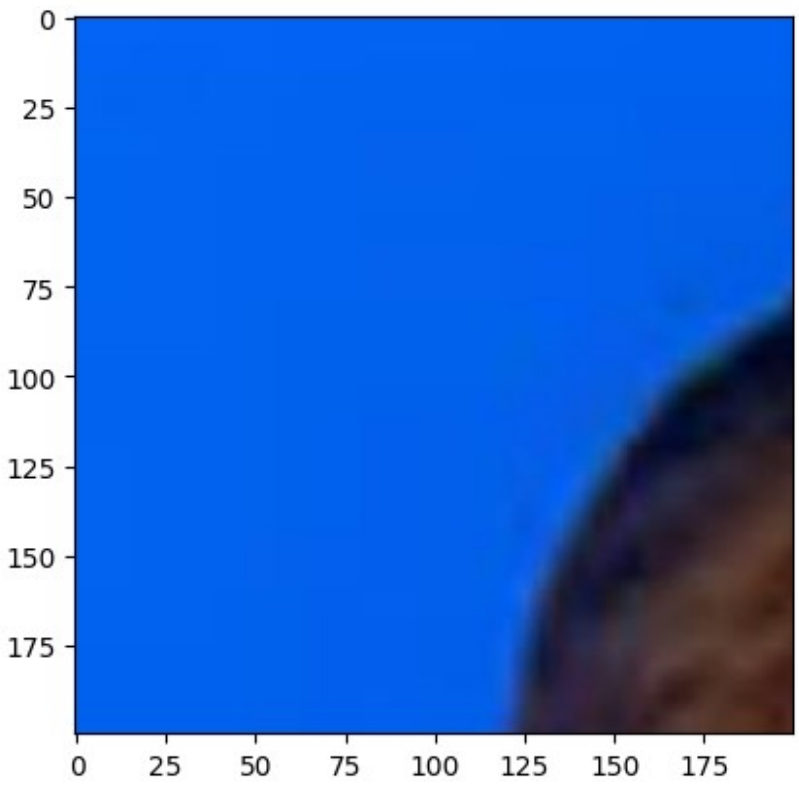



```
# Thay đổi kích thước ảnh với Bilinear Interpolation
img_resized_bilinear = cv2.resize(img_rgb, (200, 200),
interpolation=cv2.INTER_LINEAR)

plt.imshow(img_resized_bilinear)
plt.title("Bilinear Interpolation")
plt.show()
```

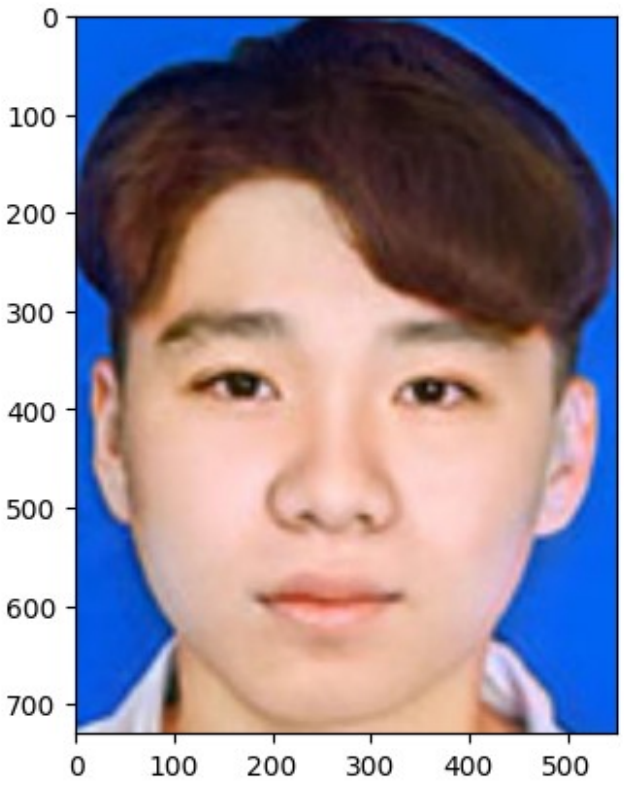


Crop image

```
img_crop = img[:200, :200] # crop image from (0, 0) to (200, 200)

plt.imshow(img_crop)
plt.show()
```

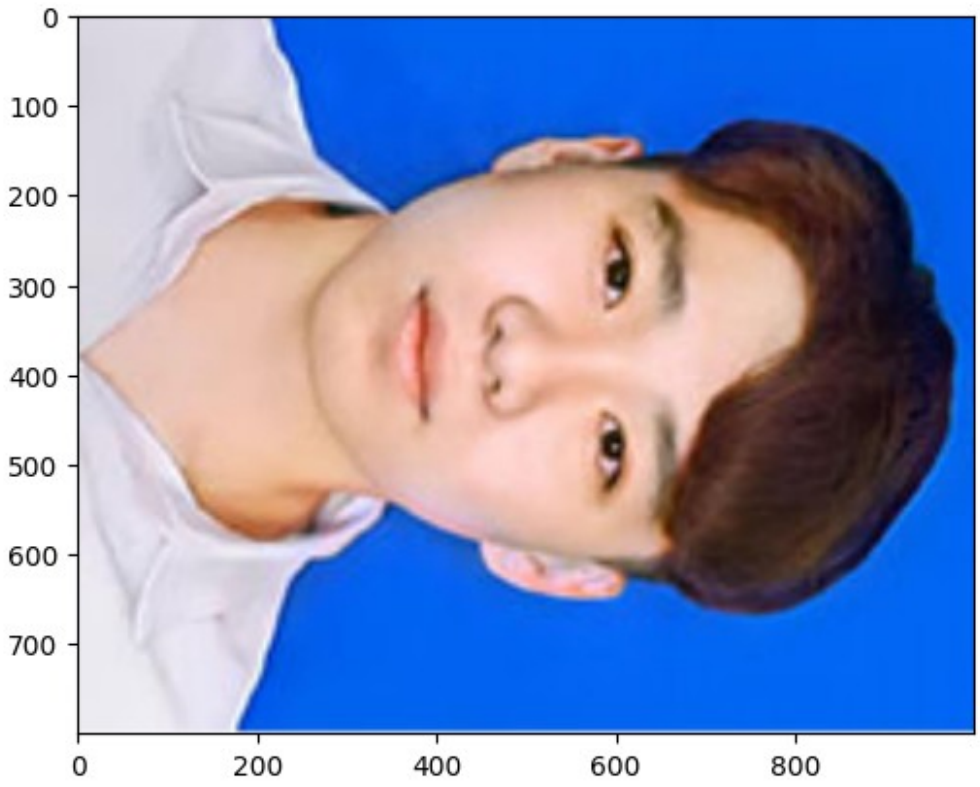



```
img_crop1 = img[20:750, 120:670] # crop image from (20, 120) to (750, 670)

plt.imshow(img_crop1)
plt.show()
```

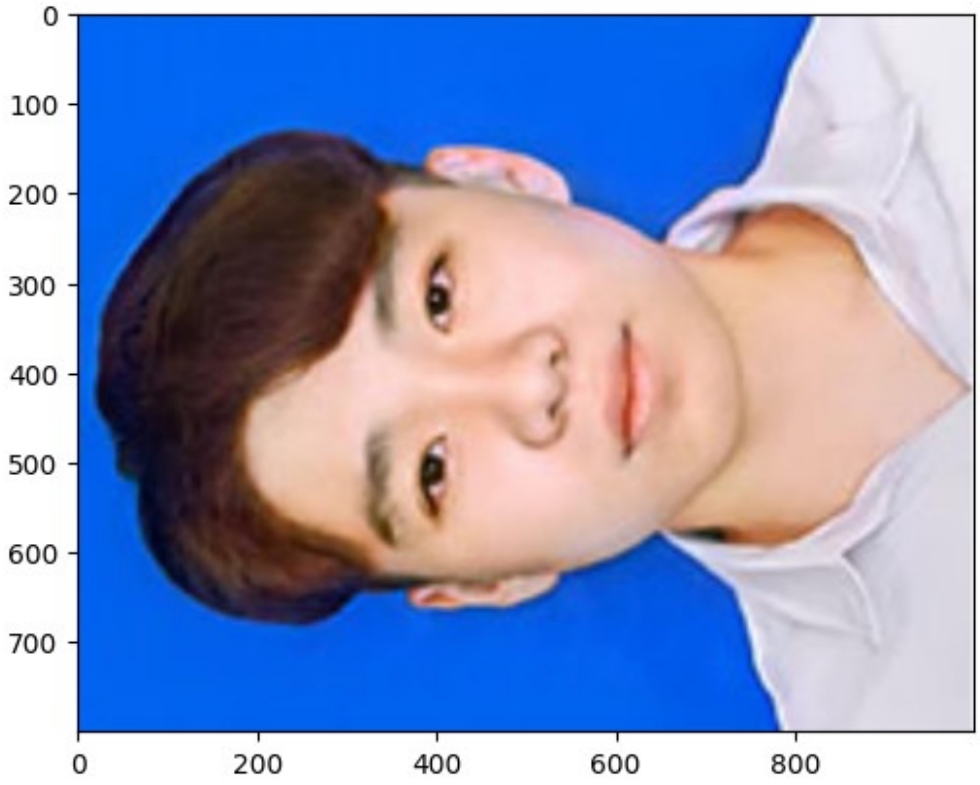


Xoay image

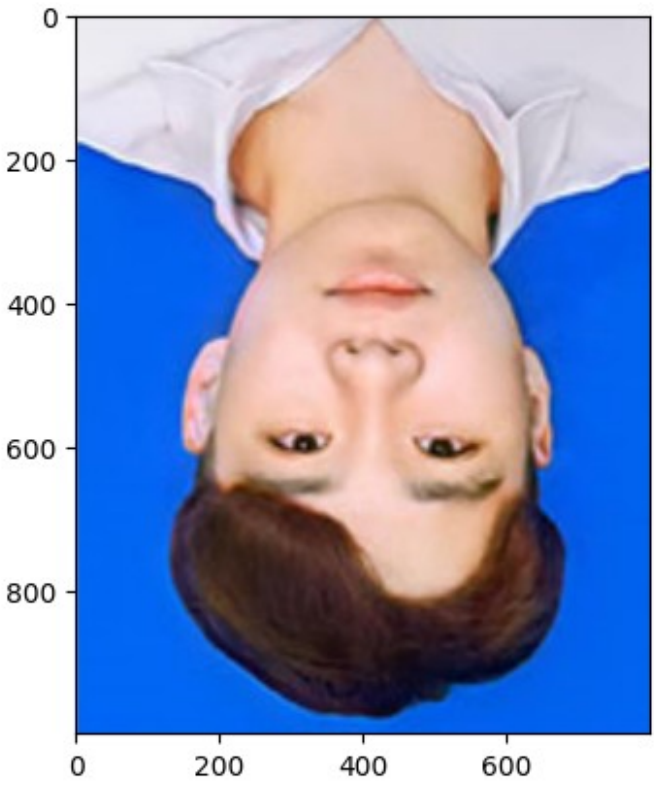
```
rotate_img = cv2.rotate(img, cv2.ROTATE_90_CLOCKWISE) # xoay 90 độ
theo chiê`u kim đô`ng hô`

plt.imshow(rotate_img)
plt.show()
```




```
rotate_img1 = cv2.rotate(img, cv2.ROTATE_90_COUNTERCLOCKWISE) # xoay
90 độ ngược chiê`u kim đô`ng hô`

plt.imshow(rotate_img1)
plt.show()
```

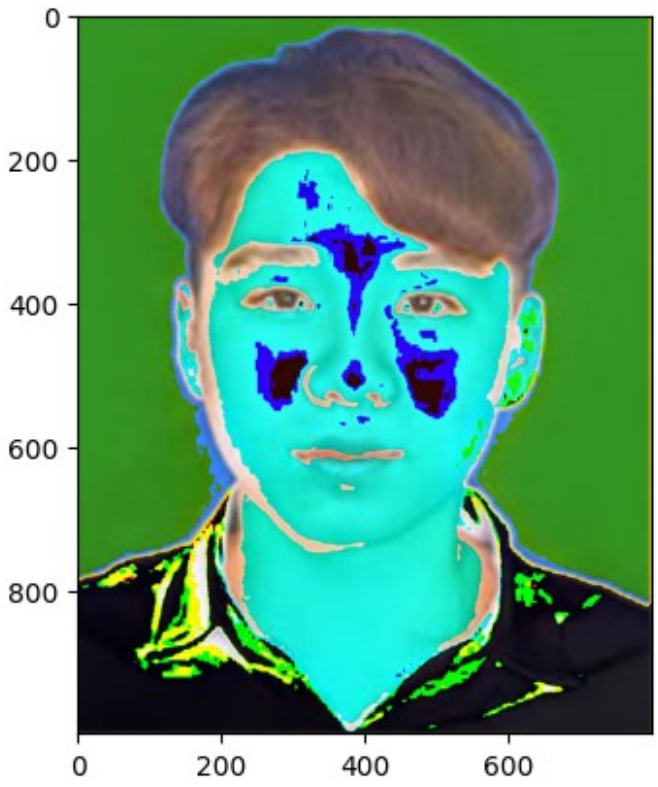



```
rotate_img2 = cv2.rotate(img, cv2.ROTATE_180) # xoay 180 độ

plt.imshow(rotate_img2)
plt.show()
```



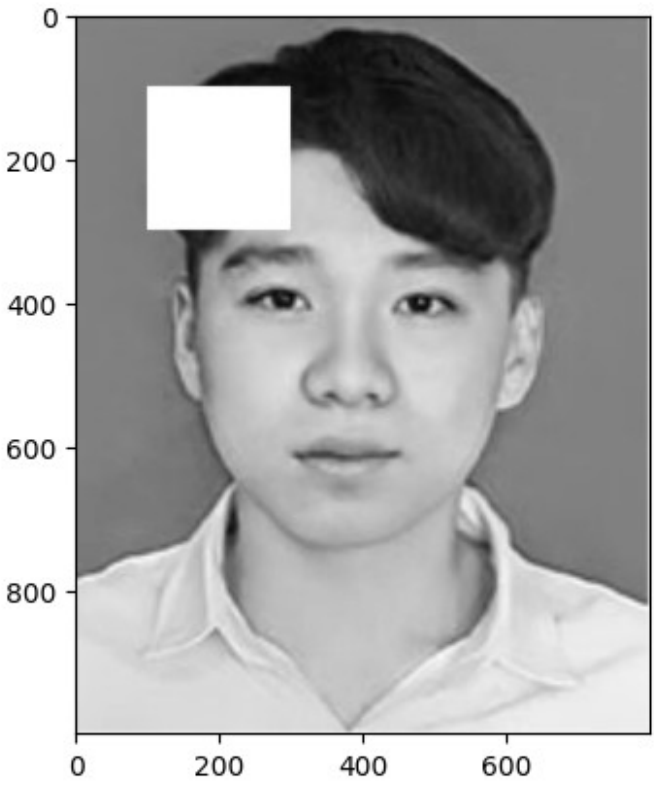
Điều chỉnh độ sáng tối

```
# Điều chỉnh độ sáng và độ tương phản
result = img.copy()
result = result + 50
plt.imshow(result,cmap='gray')
plt.show()
```



Xóa vùng ảnh

```
# Xóa vùng ảnh
result1 = img_gray.copy()
result1[100:300, 100:300] = 255 # gán giá trị 255 cho vùng ảnh có tọa độ từ (100, 100) đến (300, 300)
plt.imshow(result1, cmap='gray')

plt.show()
```



END