

22649011-ngohongthong-xla-lab2

August 12, 2024

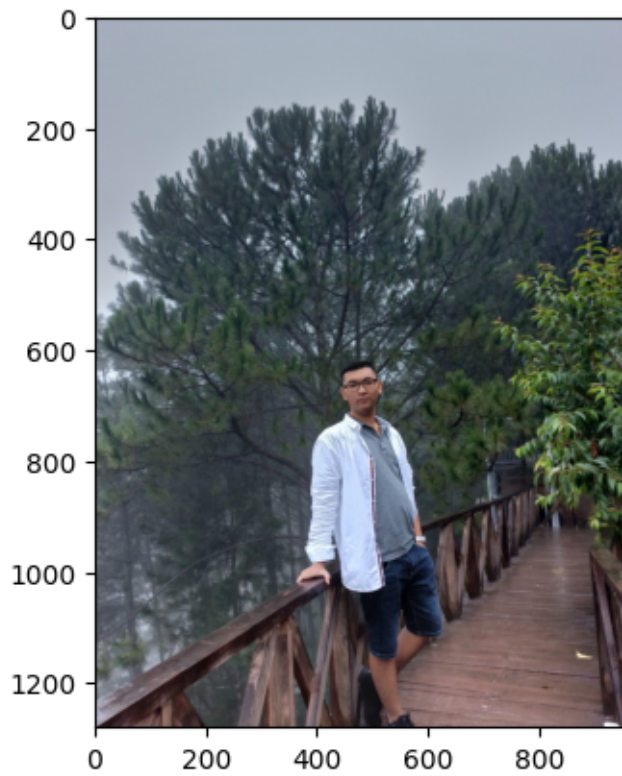
Ngô Hồng Thông

```
[1]: import matplotlib.pyplot as plt  
import numpy as np  
import pandas as pd  
import cv2
```

```
[2]: # read img  
img = plt.imread('pine.jpg')
```

```
[3]: # show img  
plt.imshow(img)
```

```
[3]: <matplotlib.image.AxesImage at 0x244bee5d8d0>
```



```
[5]: print(img.shape)
```

```
(1280, 960, 3)
```

0.1 Dữ liệu ảnh

```
[6]: # Biểu diễn dưới dạng tensor 3 chiều  
img
```

```
[6]: array([[[167, 174, 190],  
          [168, 175, 191],  
          [168, 175, 191],  
          ...,  
          [169, 176, 192],  
          [170, 177, 193],  
          [171, 178, 194]],  
         [[166, 173, 189],  
          [167, 174, 190],  
          [167, 174, 190],  
          ...,  
          [170, 177, 193],  
          [171, 178, 194],  
          [172, 179, 195]],  
         [[166, 173, 189],  
          [167, 174, 190],  
          [166, 173, 189],  
          ...,  
          [170, 177, 193],  
          [170, 177, 193],  
          [171, 178, 194]],  
         ...,  
         [[180, 172, 193],  
          [166, 159, 177],  
          [189, 180, 199],  
          ...,  
          [120, 96, 94],  
          [120, 96, 96],  
          [115, 91, 91]],  
         [[181, 172, 191],  
          [177, 168, 187],
```

```

[175, 163, 183],
...,
[123, 99, 97],
[116, 92, 92],
[111, 87, 87]],

[[172, 160, 180],
[180, 168, 188],
[172, 159, 177],
...,
[129, 105, 103],
[118, 94, 94],
[113, 89, 89]]], dtype=uint8)

```

```

[7]: # Biểu diễn ma trận ảnh thành ma trận 1 chiều
img_gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
img_gray

```

```

[7]: array([[178, 179, 179, ..., 180, 181, 182],
[177, 178, 178, ..., 181, 182, 183],
[177, 178, 177, ..., 181, 181, 182],
...,
[179, 165, 187, ..., 98, 99, 94],
[179, 175, 170, ..., 101, 95, 90],
[167, 175, 166, ..., 107, 97, 92]], dtype=uint8)

```

```

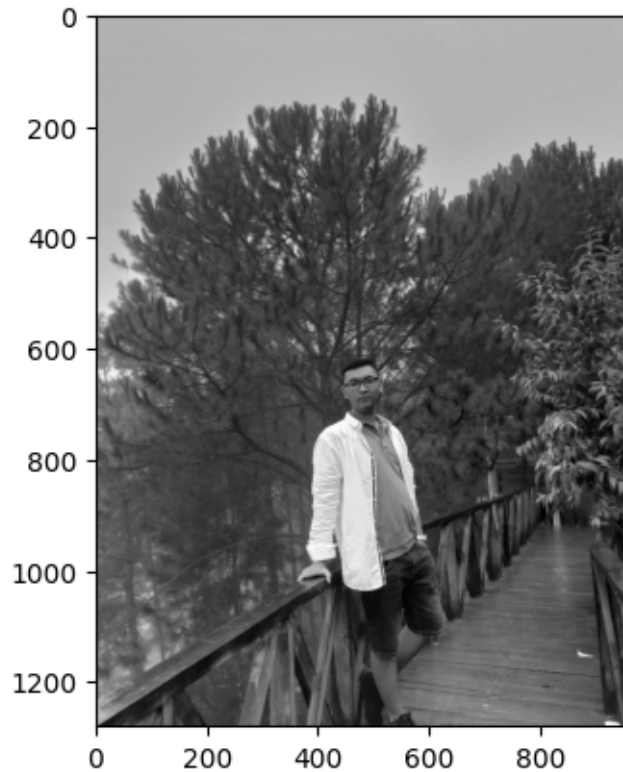
[9]: # Biểu diễn ma trận ảnh thành ma trận 2 chiều
plt.imshow(img_gray, cmap='gray')
img_gray

```

```

[9]: array([[178, 179, 179, ..., 180, 181, 182],
[177, 178, 178, ..., 181, 182, 183],
[177, 178, 177, ..., 181, 181, 182],
...,
[179, 165, 187, ..., 98, 99, 94],
[179, 175, 170, ..., 101, 95, 90],
[167, 175, 166, ..., 107, 97, 92]], dtype=uint8)

```



```
[10]: # Lấy mẫu và lượng tử hóa của ảnh
print(img_gray.shape)
print(img_gray.size)
```

```
(1280, 960)
1228800
```

```
[17]: # Lượng tử hóa của ảnh 256 gray levels (8-bit)
print(img_gray.dtype)
```

```
[17]: dtype('float32')
```

```
[16]: # Lượng tử hóa của ảnh 4 bytes
img_gray = img_gray.astype(np.float32)
img_gray
```

```
[16]: array([[178., 179., 179., ..., 180., 181., 182.],
        [177., 178., 178., ..., 181., 182., 183.],
        [177., 178., 177., ..., 181., 181., 182.],
        ...,
        [179., 165., 187., ..., 98., 99., 94.],
        [179., 175., 170., ..., 101., 95., 90.]])
```

```
[167., 175., 166., ..., 107., 97., 92.]], dtype=float32)
```

```
[18]: # Lượng tử hóa của ảnh 3 bytes  
img = img.astype(np.uint8)  
img
```

```
[18]: array([[[167, 174, 190],  
          [168, 175, 191],  
          [168, 175, 191],  
          ...,  
          [169, 176, 192],  
          [170, 177, 193],  
          [171, 178, 194]],  
  
          [[166, 173, 189],  
          [167, 174, 190],  
          [167, 174, 190],  
          ...,  
          [170, 177, 193],  
          [171, 178, 194],  
          [172, 179, 195]],  
  
          [[166, 173, 189],  
          [167, 174, 190],  
          [166, 173, 189],  
          ...,  
          [170, 177, 193],  
          [170, 177, 193],  
          [171, 178, 194]],  
  
          ...,  
  
          [[180, 172, 193],  
          [166, 159, 177],  
          [189, 180, 199],  
          ...,  
          [120, 96, 94],  
          [120, 96, 96],  
          [115, 91, 91]],  
  
          [[181, 172, 191],  
          [177, 168, 187],  
          [175, 163, 183],  
          ...,  
          [123, 99, 97],  
          [116, 92, 92],  
          [111, 87, 87]],
```

```
[[172, 160, 180],
 [180, 168, 188],
 [172, 159, 177],
 ...,
 [129, 105, 103],
 [118, 94, 94],
 [113, 89, 89]]], dtype=uint8)
```

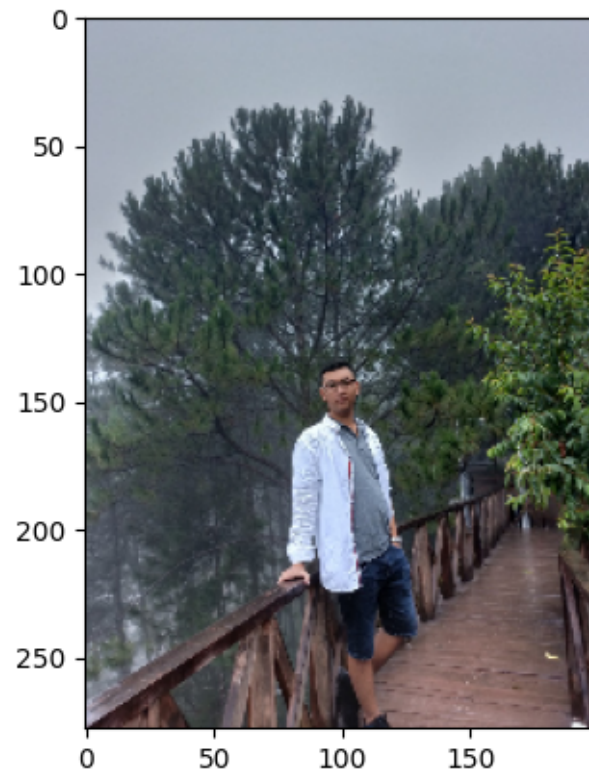
```
[20]: # Lượng tử hóa của ảnh 1 byte
img_gray = img_gray.astype(np.uint8)
img_gray
```

```
[20]: array([[178, 179, 179, ..., 180, 181, 182],
 [177, 178, 178, ..., 181, 182, 183],
 [177, 178, 177, ..., 181, 181, 182],
 ...,
 [179, 165, 187, ..., 98, 99, 94],
 [179, 175, 170, ..., 101, 95, 90],
 [167, 175, 166, ..., 107, 97, 92]], dtype=uint8)
```

0.2 Độ phân giải

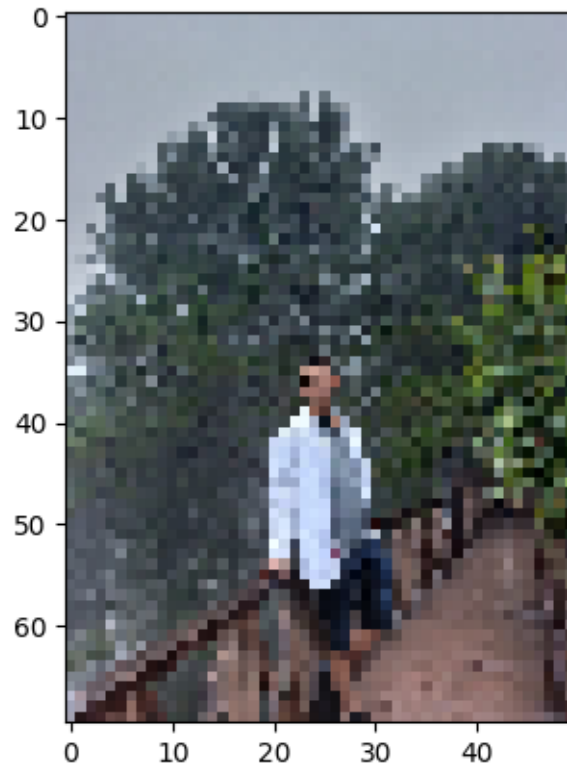
```
[23]: # Thay đổi kích thước ảnh
img_resized = cv2.resize(img, (200, 278))
plt.imshow(img_resized)
```

```
[23]: <matplotlib.image.AxesImage at 0x244c414dfc0>
```



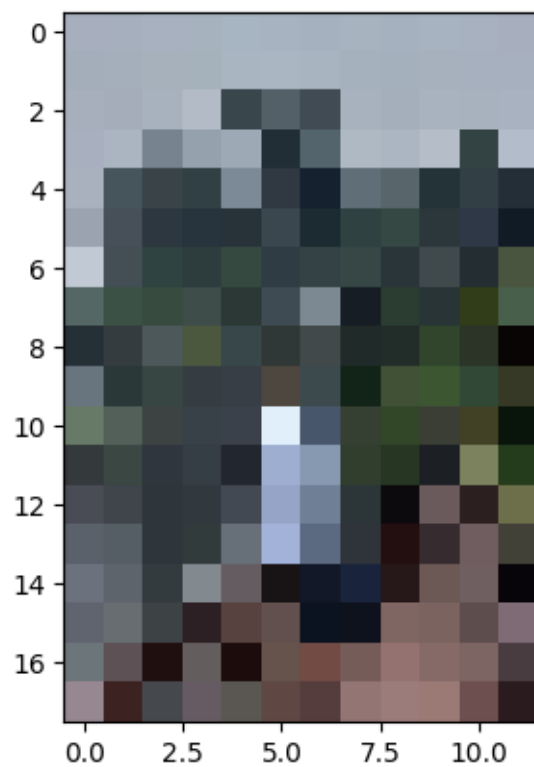
```
[24]: # Thay đổi kích thước ảnh  
img_resized = cv2.resize(img, (50, 70))  
plt.imshow(img_resized)
```

```
[24]: <matplotlib.image.AxesImage at 0x244c41b5750>
```



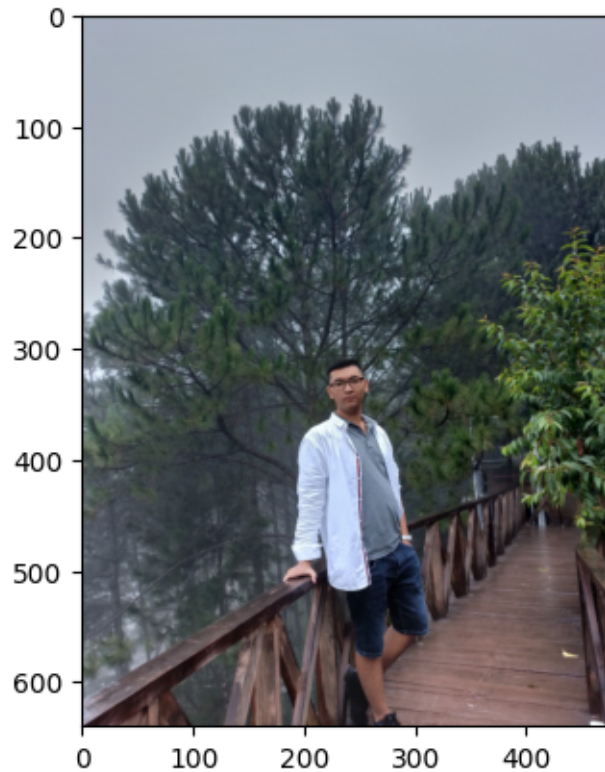
```
[25]: # Thay đổi kích thước ảnh  
img_resized = cv2.resize(img, (12, 18))  
plt.imshow(img_resized)
```

```
[25]: <matplotlib.image.AxesImage at 0x244bf7324a0>
```

```
[27]: # Giảm kích thước ảnh đi một nửa  
img_resized = cv2.resize(img, (int(img.shape[1]/2), int(img.shape[0]/2)))  
plt.imshow(img_resized)
```

```
[27]: <matplotlib.image.AxesImage at 0x244d93b5b40>
```



```
[34]: # in 3 hình HSV trên một hàng
# Chuyển đổi hình ảnh từ RGB sang HSV
image_hsv = cv2.cvtColor(img, cv2.COLOR_RGB2HSV)

# Tách các kênh Hue, Saturation, và Value
hue, saturation, value = cv2.split(image_hsv)

# Hiển thị các kênh trên cùng một hàng
fig, axs = plt.subplots(1, 3, figsize=(15, 5))

axs[0].imshow(hue, cmap='hsv')
axs[0].set_title('Hue')
axs[0].axis('off')

axs[1].imshow(saturation)
axs[1].set_title('Saturation')
axs[1].axis('off')

axs[2].imshow(value, cmap='gray')
axs[2].set_title('Value')
axs[2].axis('off')
```

```
plt.show()
```



```
[35]: # in 3 ảnh RGB trên một hàng
image_rgb = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)

# Tách các kênh màu Red, Green, Blue
red, green, blue = cv2.split(image_rgb)

# Tạo các ảnh chỉ chứa một kênh màu, các kênh còn lại đặt bằng 0
red_image = np.zeros_like(image_rgb)
green_image = np.zeros_like(image_rgb)
blue_image = np.zeros_like(image_rgb)

red_image[:, :, 0] = red # Chỉ giữ lại kênh Red
green_image[:, :, 1] = green # Chỉ giữ lại kênh Green
blue_image[:, :, 2] = blue # Chỉ giữ lại kênh Blue

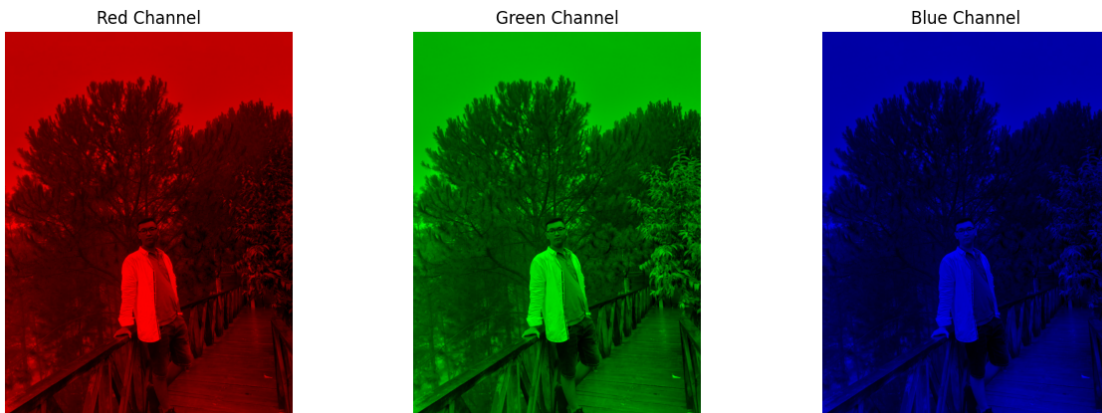
# Hiển thị các ảnh trên cùng một hàng
fig, axs = plt.subplots(1, 3, figsize=(15, 5))

axs[0].imshow(red_image)
axs[0].set_title('Red Channel')
axs[0].axis('off')

axs[1].imshow(green_image)
axs[1].set_title('Green Channel')
axs[1].axis('off')

axs[2].imshow(blue_image)
axs[2].set_title('Blue Channel')
axs[2].axis('off')
```

```
plt.show()
```



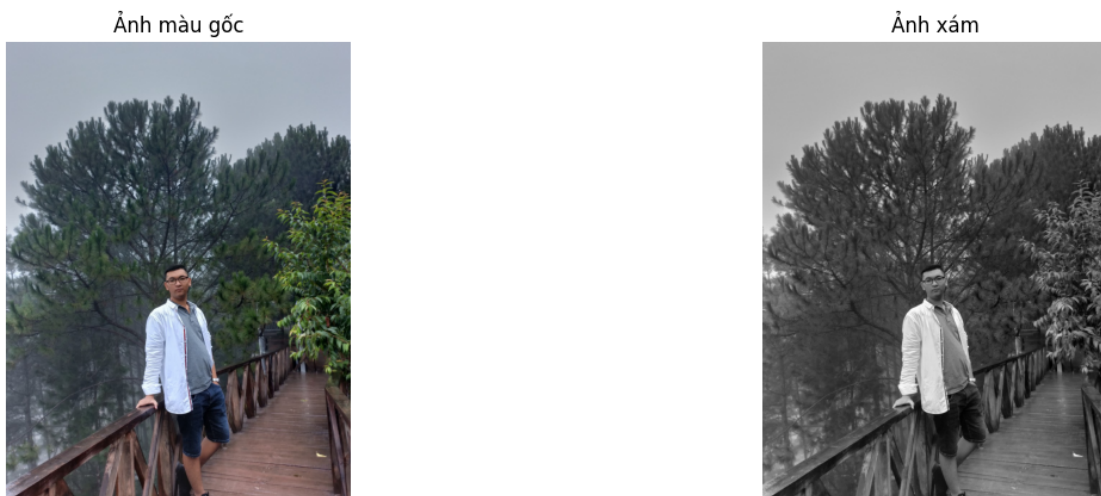
```
[40]: # Chuyển đổi hình ảnh từ RGB sang ảnh xám sử dụng hàm cv2.cvtColor
gray = cv2.cvtColor(img, cv2.COLOR_RGB2GRAY)

# Hiển thị ảnh màu và ảnh xám trên cùng một hàng
fig, axs = plt.subplots(1, 2, figsize=(15, 5))

axs[0].imshow(img)
axs[0].set_title('Ảnh màu gốc')
axs[0].axis('off')

axs[1].imshow(gray, cmap='gray')
axs[1].set_title('Ảnh xám')
axs[1].axis('off')

plt.show()
```



```
[41]: # Chuyển đổi ảnh RGB sang ảnh xám sử dụng cv2.imread() với flag=zero
gray = cv2.imread('pine.jpg', cv2.IMREAD_GRAYSCALE)

# Hiển thị ảnh màu và ảnh xám trên cùng một hàng
fig, axs = plt.subplots(1, 2, figsize=(15, 5))

axs[0].imshow(img)
axs[0].set_title('Ảnh màu gốc')
axs[0].axis('off')

axs[1].imshow(gray, cmap='gray')
axs[1].set_title('Ảnh xám')
axs[1].axis('off')

plt.show()
```

Ảnh màu gốc



Ảnh xám



```
[42]: # Chuyển đổi ảnh RGB sang ảnh xám sử dụng phương pháp trung bình
# Chuyển đổi hình ảnh từ BGR sang RGB (vì OpenCV đọc ảnh theo định dạng BGR)
image_rgb = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)

# Lấy kích thước của ảnh
rows, cols, _ = image_rgb.shape

# Tạo một mảng rỗng để chứa ảnh xám
gray_image = np.zeros((rows, cols), dtype=np.uint8)

# Sử dụng vòng lặp để tính toán giá trị grayscale cho từng pixel
```

```

for i in range(rows):
    for j in range(cols):
        gray_image[i, j] = int(sum(image_rgb[i, j]) / 3)

# Hiển thị ảnh gốc và ảnh xám
fig, axs = plt.subplots(1, 2, figsize=(10, 5))

axs[0].imshow(img)
axs[0].set_title('Original Image')
axs[0].axis('off')

axs[1].imshow(gray_image, cmap='gray')
axs[1].set_title('Grayscale Image (Mean with Loops)')
axs[1].axis('off')

plt.show()

```

Original Image



Grayscale Image (Mean with Loops)



```

[47]: # Xóa vùng ảnh khoảng 50*50 ảnh với 255
image_rgb = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)

# Lấy kích thước của ảnh
rows, cols, _ = img.shape

# Tạo một mảng rỗng để chứa ảnh xám
gray_image = np.zeros((rows, cols), dtype=np.uint8)

```



```

# Sử dụng vòng lặp để tính toán giá trị grayscale cho từng pixel
for i in range(rows):
    for j in range(cols):
        if 100 < i < 150 and 100 < j < 150:
            gray_image[i, j] = 255
        else:
            gray_image[i, j] = int(sum(image_rgb[i, j]) / 3)

# Hiển thị ảnh gốc và ảnh xám
fig, axs = plt.subplots(1, 2, figsize=(10, 5))

axs[0].imshow(img)
axs[0].set_title('Original Image')
axs[0].axis('off')

axs[1].imshow(gray_image, cmap='gray')
axs[1].set_title('Grayscale Image with Center Removed')
axs[1].axis('off')

plt.show()

```

Original Image



Grayscale Image with Center Removed



```

[49]: # crop ảnh

img_cropped = img[505:1505, 505:1505]
plt.imshow(img_cropped)

```

[49]: <matplotlib.image.AxesImage at 0x244db3cc1f0>



```
[44]: # Xoay vùng ảnh
# Tạo một ma trận xoay 2D
rotation_matrix = cv2.getRotationMatrix2D((cols/2, rows/2), 45, 1)

# Thực hiện phép biến đổi affine để xoay ảnh
rotated_image = cv2.warpAffine(img, rotation_matrix, (cols, rows))

# Hiển thị ảnh gốc và ảnh đã xoay
fig, axs = plt.subplots(1, 2, figsize=(10, 5))

axs[0].imshow(img)
axs[0].set_title('Original Image')
axs[0].axis('off')

axs[1].imshow(rotated_image)
axs[1].set_title('Rotated Image')
axs[1].axis('off')

plt.show()
```


Original Image

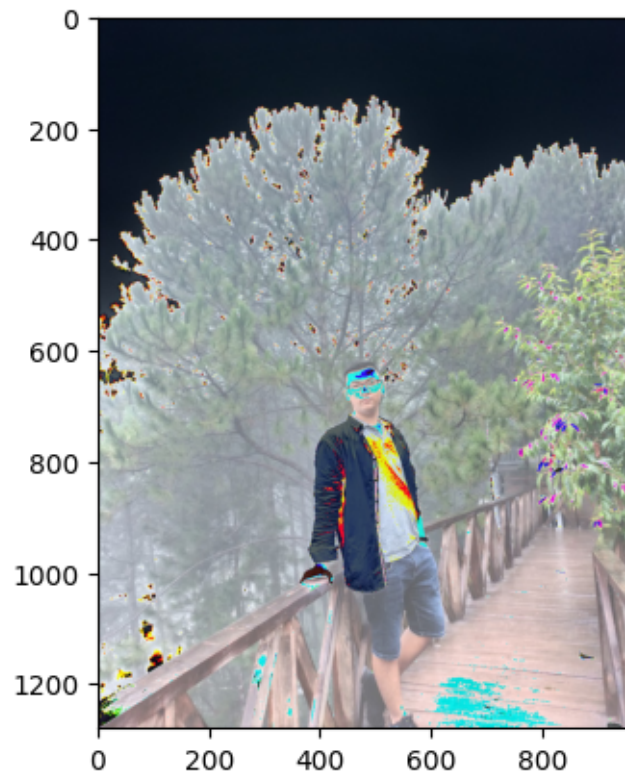


Rotated Image



```
[50]: result = img.copy()  
      result = result + 100  
      plt.imshow(result)
```

```
[50]: <matplotlib.image.AxesImage at 0x244daf07b80>
```



[]: