

8/8/2024

Task



TIDKE ASHOK TATERAO
NSTI CALICUT

Task

Requirements:

- Pc/Laptop
- VS Code
- Chrome
- Python installed with required libraries (Cv2, matplotlib)

Procedure:

1. Task-01

Step-1: Load an image using OpenCV

```
# Load an image using OpenCV
image_path = "images.jpg"
image_cv2 = cv2.imread(image_path)

# Convert the image from BGR to RGB
image_cv2_rgb = cv2.cvtColor(image_cv2, cv2.COLOR_BGR2RGB)

# Display the image
plt.imshow(image_cv2)
plt.title('Image loaded with OpenCV')
plt.show()

from PIL import Image
```

Step-2: Load an image using PIL

```
from PIL import Image
# Load an image using PIL
image_pil = Image.open(image_path)

# Display the image
plt.imshow(image_pil)
plt.title('Image loaded with PIL')
plt.show()
```

Step-3: Load an image using imageio

```
import imageio
# Load an image using imageio
image_imageio = imageio.imread(image_path)
# Display the image
plt.imshow(image_imageio)
plt.title('Image loaded with imageio')
plt.show()
# PNG image path
image_path_png = "images.jpg"
image_path_jpg = "images.jpg"
```

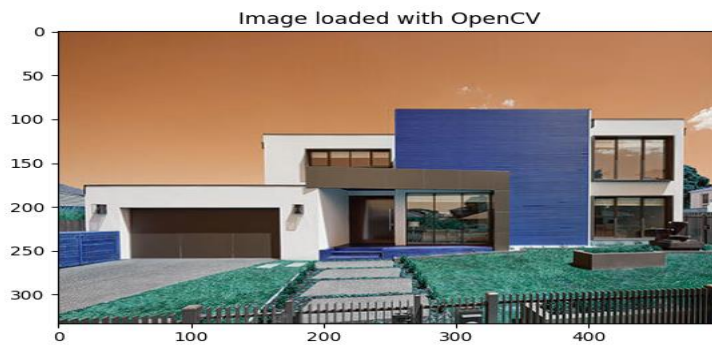
Step-4:

```
# OpenCV
image_cv2_png = cv2.imread(image_path_png)
image_cv2_png_rgb = cv2.cvtColor(image_cv2_png, cv2.COLOR_BGR2RGB)
plt.imshow(image_cv2_png_rgb)
plt.title('PNG loaded with OpenCV')
plt.show()

# PIL
image_pil_png = Image.open(image_path_png)
plt.imshow(image_cv2_png_rgb)
plt.title('PNG loaded with OpenCV')
plt.show()

# imageio
image_imageio_png = imageio.imread(image_path_png)
plt.imshow(image_cv2_png_rgb)
plt.title('PNG loaded with OpenCV')
plt.show()
```

Output:



2. Task-02

Step-1: load an image

```
# Load an image
image = cv2.imread('images.jpg')

# Convert the image from BGR (OpenCV format) to RGB (Matplotlib format)
image_rgb = cv2.cvtColor(image, cv2.COLOR_BGR2RGB)

# Resize image to 256x256 pixels
resized_image = cv2.resize(image_rgb, (125, 128))
```

Step-2: display the original and resized image

```
# Display the original and resized images
plt.figure(figsize=(10, 5))
plt.subplot(1, 2, 1)
plt.title('Original Image')
plt.imshow(image_rgb)
plt.axis('off')
plt.subplot(1, 2, 2)
plt.title('Resized Image (125x128)')
plt.imshow(resized_image)
plt.axis('off')
```

Step-3: rotate image by 45 degree

```
# Rotate image by 45 degrees
(h, w) = image_rgb.shape[:2]
center = (w // 2, h // 2)
M = cv2.getRotationMatrix2D(center, 45, 1.0)
rotated_image = cv2.warpAffine(image_rgb, M, (w, h))
```

Output:

Original Image



Resized Image (125x128)



Original Image



cropped_image



Original Image



rotated_image



1. Task-01

Step-1: load an image

```
# Load an image
image = cv2.imread('images.jpg')

# Convert the image from BGR (OpenCV format) to RGB (Matplotlib format)
image_rgb = cv2.cvtColor(image, cv2.COLOR_BGR2RGB)

# Apply Gaussian blur to denoise
denoised_image = cv2.GaussianBlur(image_rgb, (11, 11), 0)
```

Step-2: display the original and resized image

```
# Display the original and resized images
plt.figure(figsize=(10, 5))
plt.subplot(1, 2, 1)
plt.title('Original Image')
plt.imshow(image_rgb)
plt.axis('off')
plt.subplot(1, 2, 2)
plt.title('denoised_image')
plt.imshow(denoised_image)
plt.axis('off')
plt.show()
```

Step-3: convert to grayscale

```
# Convert to grayscale
gray_image = cv2.cvtColor(image_rgb, cv2.COLOR_BGR2GRAY)

# Apply histogram equalization
equalized_image = cv2.equalizeHist(gray_image)
```

Step-4: display the original image

```
# Display the original and resized images
plt.figure(figsize=(10, 5))
plt.subplot(1, 2, 1)
plt.title('Gray Image')
plt.imshow(gray_image, cmap="gray")
plt.axis('off')
plt.subplot(1, 2, 2)
plt.title('equalized_image')
plt.imshow(equalized_image, cmap="gray")
plt.axis('off')
plt.show()
```

Output:

Original Image



denoised_image



Gray Image



equalized_image

