

Tutorial 2 - How to Save Figure and Images using Matplotlib in High Quality

September 27, 2024

1 How to Save Figure and Images using Matplotlib in High Quality

1.1 Import Required Libraries

```
[4]: import os
import cv2
import matplotlib.pyplot as plt
```

1.2 Sample Code for Processing Image to 1-bit Image or Binary Image

```
[ ]: path = os.path.join(os.getcwd(), '../assets/images/parrot.jpg')

# Read Image
image = cv2.imread(path)

# Convert Image to RGB
image = cv2.cvtColor(image, cv2.COLOR_BGR2RGB)

# Convert Image to Grayscale
gray = cv2.cvtColor(image, cv2.COLOR_RGB2GRAY)

# Convert Image to Binary
_, binary = cv2.threshold(gray, 127, 255, cv2.THRESH_BINARY)

# Display Original Image and Binary Image
plt.figure("Original Image and Binary Image")

plt.subplot(1, 2, 1)
plt.title("Original Image")
plt.imshow(image)
plt.axis('off')

plt.subplot(1, 2, 2)
plt.title("Binary Image")
plt.imshow(binary, cmap='gray')
```

```
plt.axis('off')
```

1.3 Save Figure in High Quality

To save the figure in high quality, we can use the `savefig()` function of the `pyplot` module. The `savefig()` function takes the following parameters:

- `fname`: The file name of the saved figure.
- `dpi`: The resolution of the saved figure in dots per inch (DPI). The default value is 100.
- `bbox_inches`: The bounding box in inches that will be saved. The default value is 'tight'.

```
[6]: plt.savefig('./outputs/t2-figure-save-figure.png', dpi=300, bbox_inches='tight')
```

<Figure size 640x480 with 0 Axes>

1.4 Save Image in High Quality

To save the image in high quality, we can use the `imsave()` function from the `matplotlib.pyplot` module. The `imsave()` function requires the following parameters:

- `path`: The path where the image will be saved.
- `binary`: The binary image that will be saved.
- `dpi`: The resolution of the image in dots per inch (DPI).

Since in this example we are saving a binary image, we can use the `cmap` parameter to specify the colormap of the image. In this case, we are using the `gray` colormap to save the binary image in grayscale.

```
[7]: plt.imsave('./outputs/t2-save-image.png', binary, dpi=300, cmap='gray')
```

1.5 Display Figure

To display the figure, we can use the `plt.show()` method. This method will display the figure on the screen.

```
[8]: plt.show()
```

1.6 Full Code

```
[10]: import os
import cv2
import matplotlib.pyplot as plt

path = os.path.join(os.getcwd(), '../assets/images/parrot.jpg')

# Read Image
image = cv2.imread(path)

# Convert Image to RGB
image = cv2.cvtColor(image, cv2.COLOR_BGR2RGB)
```

```

# Convert Image to Grayscale
gray = cv2.cvtColor(image, cv2.COLOR_RGB2GRAY)

# Convert Image to Binary
_, binary = cv2.threshold(gray, 127, 255, cv2.THRESH_BINARY)

# Display Original Image and Binary Image
plt.figure("Original Image and Binary Image")

plt.subplot(1, 2, 1)
plt.title("Original Image")
plt.imshow(image)
plt.axis('off')

plt.subplot(1, 2, 2)
plt.title("Binary Image")
plt.imshow(binary, cmap='gray')
plt.axis('off')

plt.savefig('./outputs/t2-figure-save-figure.png', dpi=300, bbox_inches='tight')
plt.imsave('./outputs/t2-save-image.png', binary, cmap='gray', dpi=300)

plt.show()

```

Original Image



Binary Image

