1. Add any text on an image using PIL. Display the result image.
2. Draw three straight lines on an image using PIL. Display the result image.
3. Draw a rectangle on an image with Blue outline using PIL. Display the result image.
4. Draw a polygon with 10 sides using PIL. Display the result image.
5. Read and image and print the array representation of the image using Matplotlib.
6. Read an RGB Image and convert it to GrayScale Image using openCv and display the result image.
7. Read a GrayScale Image and convert it to Binary Image using openCv and display the result image.

Assignment 3

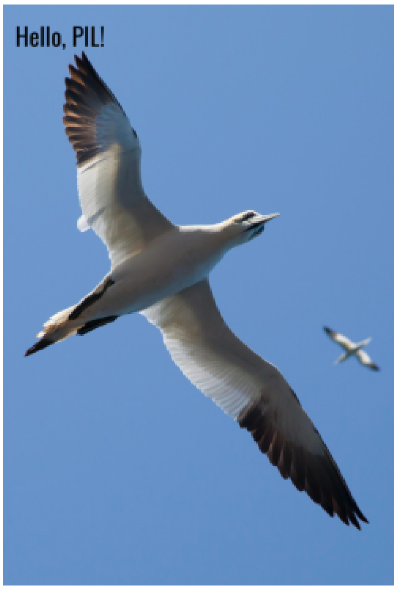
PIL and Matplotlib

1. Add any text on an image using PIL. Display the result image.

```
from PIL import Image, ImageDraw, ImageFont
import matplotlib.pyplot as plt

# Open an image
image = Image.open('example.jpg')
# Ensure the image is in RGB mode
image = image.convert("RGB")
# Initialize ImageDraw
draw = ImageDraw.Draw(image)
# Define the text
text = "Hello, PIL!"
# Use the uploaded font file
font_path = "/content/font.ttf"  # Path to the uploaded font file
font = ImageFont.truetype(font_path, size=100)
# Position of the text
position = (50, 50)
# Add text to the image (using black color)
draw.text(position, text, font=font, fill="black")
# Display the result image
plt.imshow(image)
plt.axis('off')  # Hide axis
plt.show()
```

Output:

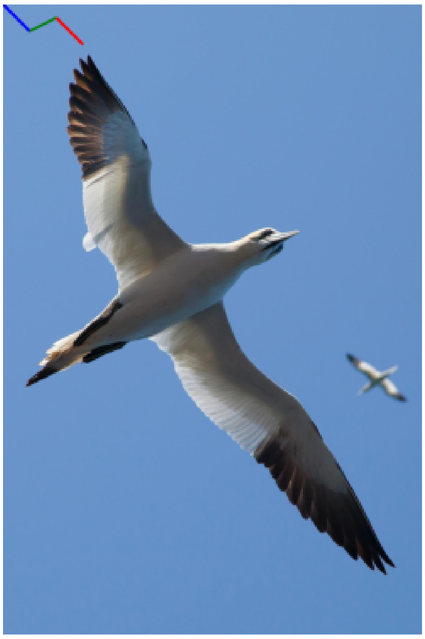


2. Draw three straight lines on an image using PIL. Display the result image.

```
from PIL import Image, ImageDraw
import matplotlib.pyplot as plt

# Open an image
image = Image.open('example.jpg')
# Initialize ImageDraw
draw = ImageDraw.Draw(image)
# Draw three lines (example coordinates)
draw.line((0, 0, 100, 100), fill="blue", width=10)
draw.line((100, 100, 200, 50), fill="green", width=10)
draw.line((200, 50, 300, 150), fill="red", width=10)
# Display the result image
plt.imshow(image)
plt.axis('off')  # Hide axis
plt.show()
```

Output:



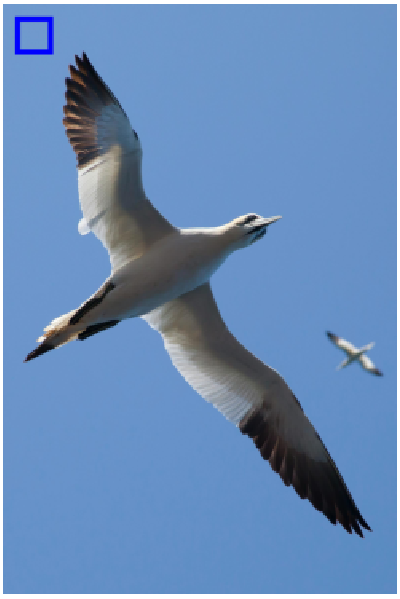
3. Draw a rectangle on an image with Blue outline using PIL. Display the result image.

```
from PIL import Image, ImageDraw
import matplotlib.pyplot as plt

# Open an image
image = Image.open('example.jpg')
# Initialize ImageDraw
```

```
draw = ImageDraw.Draw(image)
# Draw a blue rectangle (example coordinates)
draw.rectangle([50, 50, 200, 200], outline="blue", width=20)
# Display the result image
plt.imshow(image)
plt.axis('off')  # Hide axis
plt.show()
```

Output:



4. Draw a polygon with 10 sides using PIL. Display the result image.

```
from PIL import Image, ImageDraw
import matplotlib.pyplot as plt
import numpy as np

# Open an image
image = Image.open('example.jpg')
# Ensure the image is in RGB mode
image = image.convert("RGB")
# Initialize ImageDraw
draw = ImageDraw.Draw(image)
# Define coordinates for the polygon (10-sided) with a larger radius
radius = 150  # Increase the radius for a larger polygon
polygon = [(150 + radius * np.cos(np.pi * 2 * i / 10), 150 + radius *
np.sin(np.pi * 2 * i / 10)) for i in range(10)]
# Draw the polygon (Red fill)
draw.polygon(polygon, fill="red", outline="black")
# Display the result image
plt.imshow(image)
plt.axis('off')  # Hide axis
plt.show()
```

Output:

5. Read an image and print the array representation of the image using Matplotlib.

```
import matplotlib.pyplot as plt
import matplotlib.image as mpimg
# Read the image
image = mpimg.imread('example.jpg')
# Print the array representation of the image
print(image)
```

Output:

```
[[[112 155 206]
  [111 154 205]
  [111 154 205]
  ...
  [102 145 196]
  [102 145 196]
  [103 146 197]]

 [[112 155 206]
  [111 154 205]
  [111 154 205]
  ...
  [104 147 198]
  [105 148 199]
  [105 148 199]]

 [[112 155 206]
```

```
  [111 154 205]
  [111 154 205]
  ...
  [107 150 201]
  [107 150 201]
  [107 150 201]]

 ...

 [[102 143 199]
  [102 143 199]
  [101 144 199]
  ...
  [ 91 138 192]
  [ 91 138 192]
  [ 92 139 193]]

 [[102 143 199]
  [102 143 199]
  [101 144 199]
  ...
  [ 90 137 191]
  [ 92 139 193]
  [ 93 140 194]]

 [[103 144 200]
  [102 143 199]
  [100 143 198]
  ...
  [ 90 137 191]
  [ 92 139 193]
  [ 94 141 195]]]
```

6. Read an RGB Image and convert it to GrayScale Image using openCv and display the result image.

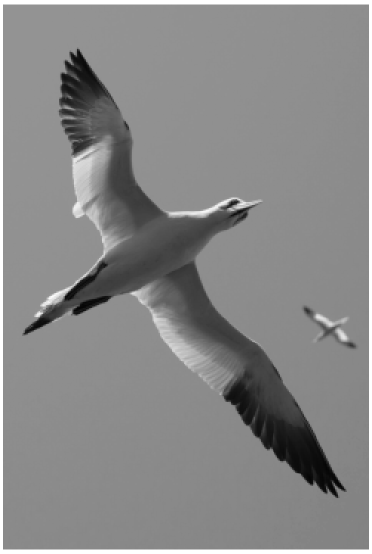
```python
import cv2
import matplotlib.pyplot as plt

# Read the image in color (RGB)
image = cv2.imread('example.jpg')

# Convert the image to grayscale
gray_image = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)

# Display the result image
plt.imshow(gray_image, cmap='gray')
plt.axis('off')  # Hide axis
plt.show()
```

Output:



7. Read a GrayScale Image and convert it to Binary Image using openCv and display the result image.

```
import cv2
import matplotlib.pyplot as plt

# Read the grayscale image
image = cv2.imread('example.jpg', cv2.IMREAD_GRAYSCALE)
# Convert the grayscale image to binary using Otsu's thresholding
_, binary_image = cv2.threshold(image, 0, 255, cv2.THRESH_BINARY +
cv2.THRESH_OTSU)
# Display the binary image
plt.imshow(binary_image, cmap='gray')
plt.axis('off')  # Hide axis
plt.show()
```

Output: