

✓ Name : Saiyyed Kaif Aalam

Section & Roll no. : A-56

Topic : Harris Corner Detection

```
import cv2
import numpy as np
from google.colab.patches import cv2_imshow
import matplotlib.pyplot as plt # Import matplotlib

# Load the image
image = cv2.imread('bird1.jpeg')

if image is None:
    print("Error: Could not open or find the image.")
else:
    # Display the original image with title
    plt.figure(figsize=(6, 4)) # Adjust figure size if needed
    plt.imshow(cv2.cvtColor(image, cv2.COLOR_BGR2RGB)) # Convert BGR to RGB for matplotlib
    plt.title("Original Image")
    plt.axis('off') # Turn off axis ticks and labels
    plt.show()

    gray_img = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
    gray_img = np.float32(gray_img)

    # Display the grayscale image with title
    plt.figure(figsize=(6, 4))
    plt.imshow(gray_img, cmap='gray') # Use 'gray' colormap for grayscale
    plt.title("Grayscale Image")
    plt.axis('off')
    plt.show()

    # Harris corner detection parameters
    blocksize = 3
    ksize = 3
    k = 0.04

    # Detect corners
    corners_img = cv2.cornerHarris(gray_img, blocksize, ksize, k)

    # Mark the corners in yellow
    image[corners_img > 0.001 * corners_img.max()] = [255, 255, 0]

    # Display the result with title
    plt.figure(figsize=(6, 4))
    plt.imshow(cv2.cvtColor(image, cv2.COLOR_BGR2RGB))
    plt.title("Image with Corners Detected")
    plt.axis('off')
    plt.show()
```



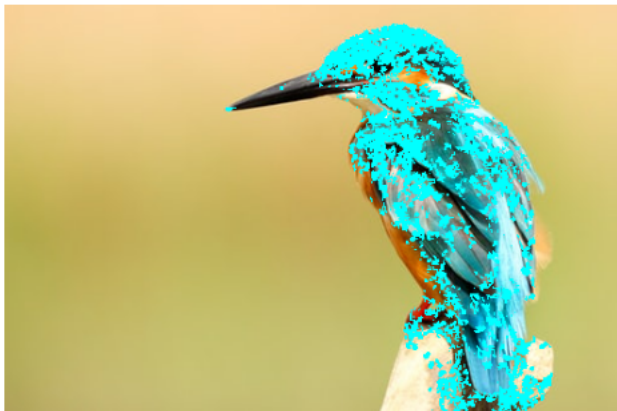
Original Image



Grayscale Image



Image with Corners Detected



```
import cv2
import numpy as np
from google.colab.patches import cv2_imshow
import matplotlib.pyplot as plt # Import matplotlib

# Load the image
image = cv2.imread('eagle.jpeg')

if image is None:
    print("Error: Could not open or find the image.")
else:
    # Display the original image with title
    plt.figure(figsize=(6, 4)) # Adjust figure size if needed
    plt.imshow(cv2.cvtColor(image, cv2.COLOR_BGR2RGB)) # Convert BGR to RGB for matplotlib
    plt.title("Original Image")
    plt.axis('off') # Turn off axis ticks and labels
    plt.show()

    gray_img = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
    gray_img = np.float32(gray_img)

    # Display the grayscale image with title
    plt.figure(figsize=(6, 4))
    plt.imshow(gray_img, cmap='gray') # Use 'gray' colormap for grayscale
    plt.title("Grayscale Image")
```

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

# Harris corner detection parameters
blocksize = 3
ksize = 3
k = 0.04

# Detect corners
corners_img = cv2.cornerHarris(gray_img, blocksize, ksize, k)

# Mark the corners in yellow
image[corners_img > 0.001 * corners_img.max()] = [255, 255, 0]

# Display the result with title
plt.figure(figsize=(6, 4))
plt.imshow(cv2.cvtColor(image, cv2.COLOR_BGR2RGB))
plt.title("Image with Corners Detected")
plt.axis('off')
plt.show()
```



Original Image



Grayscale Image



Image with Corners Detected



```
import cv2
import numpy as np
from google.colab.patches import cv2_imshow
import matplotlib.pyplot as plt # Import matplotlib

# Load the image
image = cv2.imread('ansel.jpg')

if image is None:
    print("Error: Could not open or find the image.")
else:
    # Display the original image with title
```

```
plt.figure(figsize=(6, 4)) # Adjust figure size if needed
plt.imshow(cv2.cvtColor(image, cv2.COLOR_BGR2RGB)) # Convert BGR to RGB for matplotlib
plt.title("Original Image")
plt.axis('off') # Turn off axis ticks and labels
plt.show()

gray_img = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
gray_img = np.float32(gray_img)

# Harris corner detection parameters
blocksize = 3
ksize = 3
k = 0.04

# Detect corners
corners_img = cv2.cornerHarris(gray_img, blocksize, ksize, k)

# Mark the corners in yellow
image[corners_img > 0.001 * corners_img.max()] = [255, 255, 0]

# Display the result with title
plt.figure(figsize=(6, 4))
plt.imshow(cv2.cvtColor(image, cv2.COLOR_BGR2RGB))
plt.title("Image with Corners Detected")
plt.axis('off')
plt.show()
```



Original Image



Image with Corners Detected

