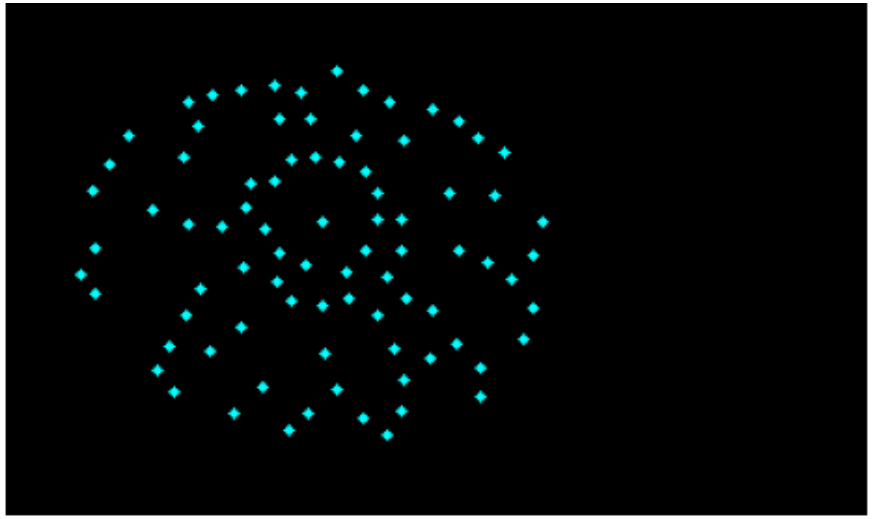
```
import cv2
import numpy as np
from google.colab.patches import cv2_imshow
import matplotlib.pyplot as plt  # Import matplotlib

def shi_tomasi(image):
    gray_img = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)

    # Shi-Tomasi corner detection parameters
    corners_img = cv2.goodFeaturesToTrack(gray_img, 1200, 0.01, 10)

    # Create a blank image to draw the corners
    blank_img = np.zeros((image.shape[0], image.shape[1], 3), np.uint8)

    # Mark the corners on the original image and the blank image
    for corners in corners_img:
        x, y = corners.ravel()
        # Convert x and y to integers before drawing the circle
        x = int(x)
        y = int(y)
        cv2.circle(image, (x, y), 3, [255, 255, 0], -1)  # Mark on original image
        cv2.circle(blank_img, (x, y), 2, [255, 255, 0], -1)  # Mark on blank image

    return image, blank_img


# Load the image
image = cv2.imread('daisy.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()
```

```
# Call the Shi-Tomasi corner detection function
image_with_corners, blank_img = shi_tomasi(image)

# Display the result with corners detected on the original image
plt.figure(figsize=(6, 4))
plt.imshow(cv2.cvtColor(image_with_corners, cv2.COLOR_BGR2RGB))  # Convert BGR to RGB for matplotlib
plt.title("Image with Shi-Tomasi Corners Detected")
plt.axis('off')
plt.show()

# Display the blank image with just the corners
plt.figure(figsize=(6, 4))
plt.imshow(cv2.cvtColor(blank_img, cv2.COLOR_BGR2RGB))  # Convert BGR to RGB for matplotlib
plt.title("Blank Image with Shi-Tomasi Corners")
plt.axis('off')
plt.show()
```

Original Image

Image with Shi-Tomasi Corners Detected

Blank Image with Shi-Tomasi Corners



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

def shi_tomasi(image):
    gray_img = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)

    # Shi-Tomasi corner detection parameters
    corners_img = cv2.goodFeaturesToTrack(gray_img, 1200, 0.01, 10)

    # Create a blank image to draw the corners
    blank_img = np.zeros((image.shape[0], image.shape[1], 3), np.uint8)

    # Mark the corners on the original image and the blank image
    for corners in corners_img:
        x, y = corners.ravel()
        # Convert x and y to integers before drawing the circle
        x = int(x)
        y = int(y)
        cv2.circle(image, (x, y), 3, [255, 255, 0], -1)  # Mark on original image
        cv2.circle(blank_img, (x, y), 2, [255, 255, 0], -1)  # Mark on blank image
```

```
        return image, blank_img


    # Load the image
    image = cv2.imread('penguin.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()

        # Call the Shi-Tomasi corner detection function
        image_with_corners, blank_img = shi_tomasi(image)

        # Display the result with corners detected on the original image
        plt.figure(figsize=(6, 4))
        plt.imshow(cv2.cvtColor(image_with_corners, cv2.COLOR_BGR2RGB))  # Convert BGR to RGB for matplotlib
        plt.title("Image with Shi-Tomasi Corners Detected")
        plt.axis('off')
        plt.show()

        # Display the blank image with just the corners
        plt.figure(figsize=(6, 4))
        plt.imshow(cv2.cvtColor(blank_img, cv2.COLOR_BGR2RGB))  # Convert BGR to RGB for matplotlib
        plt.title("Blank Image with Shi-Tomasi Corners")
        plt.axis('off')
        plt.show()
```

Original Image

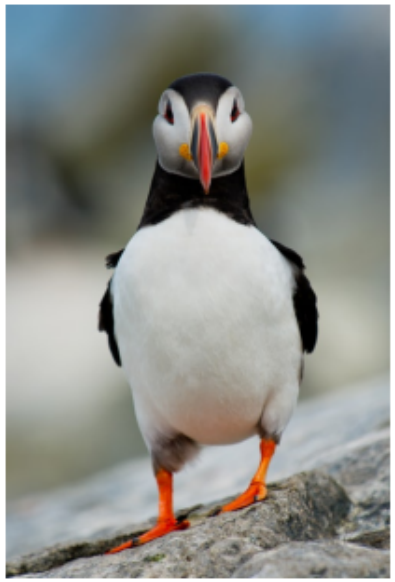
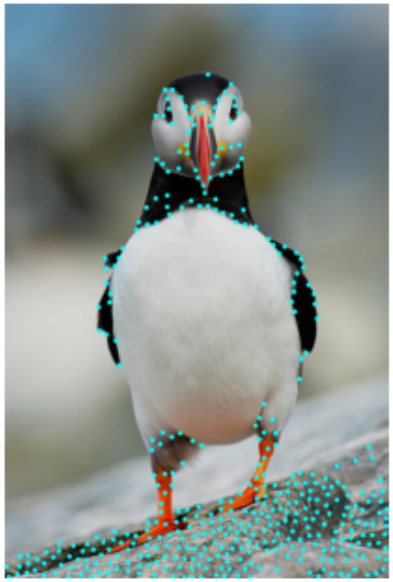
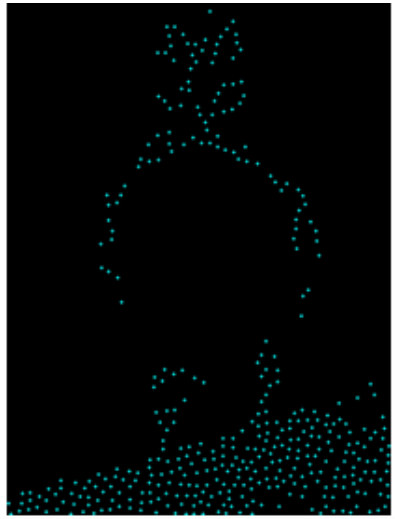


Image with Shi-Tomasi Corners Detected



Blank Image with Shi-Tomasi Corners



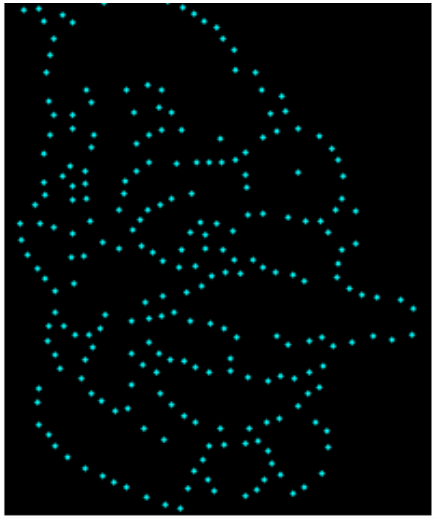
```
import cv2
import numpy as np
from google.colab.patches import cv2_imshow
import matplotlib.pyplot as plt  # Import matplotlib

def shi_tomasi(image):
    gray_img = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)

    # Shi-Tomasi corner detection parameters
    corners_img = cv2.goodFeaturesToTrack(gray_img, 1200, 0.01, 10)

    # Create a blank image to draw the corners
    blank_img = np.zeros((image.shape[0], image.shape[1], 3), np.uint8)

    # Mark the corners on the original image and the blank image
    for corners in corners_img:
        x, y = corners.ravel()
        # Convert x and y to integers before drawing the circle
        x = int(x)
        y = int(y)
        cv2.circle(image, (x, y), 3, [255, 255, 0], -1)  # Mark on original image
        cv2.circle(blank_img, (x, y), 2, [255, 255, 0], -1)  # Mark on blank image
```

```
        return image, blank_img


    # Load the image
    image = cv2.imread('simpsonTest.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()

        # Call the Shi-Tomasi corner detection function
        image_with_corners, blank_img = shi_tomasi(image)

        # Display the result with corners detected on the original image
        plt.figure(figsize=(6, 4))
        plt.imshow(cv2.cvtColor(image_with_corners, cv2.COLOR_BGR2RGB))  # Convert BGR to RGB for matplotlib
        plt.title("Image with Shi-Tomasi Corners Detected")
        plt.axis('off')
        plt.show()

        # Display the blank image with just the corners
        plt.figure(figsize=(6, 4))
        plt.imshow(cv2.cvtColor(blank_img, cv2.COLOR_BGR2RGB))  # Convert BGR to RGB for matplotlib
        plt.title("Blank Image with Shi-Tomasi Corners")
        plt.axis('off')
        plt.show()
```

Original Image

Image with Shi-Tomasi Corners Detected

Blank Image with Shi-Tomasi Corners



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

def shi_tomasi(image):
    gray_img = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)

    # Shi-Tomasi corner detection parameters
    corners_img = cv2.goodFeaturesToTrack(gray_img, 1200, 0.01, 10)

    # Create a blank image to draw the corners
    blank_img = np.zeros((image.shape[0], image.shape[1], 3), np.uint8)

    # Mark the corners on the original image and the blank image
    for corners in corners_img:
        x, y = corners.ravel()
        # Convert x and y to integers before drawing the circle
        x = int(x)
        y = int(y)
        cv2.circle(image, (x, y), 3, [255, 255, 0], -1)  # Mark on original image
        cv2.circle(blank_img, (x, y), 2, [255, 255, 0], -1)  # Mark on blank image
```

```
        return image, blank_img


    # Load the image
    image = cv2.imread('giraffe.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()

        # Call the Shi-Tomasi corner detection function
        image_with_corners, blank_img = shi_tomasi(image)

        # Display the result with corners detected on the original image
        plt.figure(figsize=(6, 4))
        plt.imshow(cv2.cvtColor(image_with_corners, cv2.COLOR_BGR2RGB))  # Convert BGR to RGB for matplotlib
        plt.title("Image with Shi-Tomasi Corners Detected")
        plt.axis('off')
        plt.show()

        # Display the blank image with just the corners
        plt.figure(figsize=(6, 4))
        plt.imshow(cv2.cvtColor(blank_img, cv2.COLOR_BGR2RGB))  # Convert BGR to RGB for matplotlib
        plt.title("Blank Image with Shi-Tomasi Corners")
        plt.axis('off')
        plt.show()
```

Original Image

Image with Shi-Tomasi Corners Detected