Practical – 2

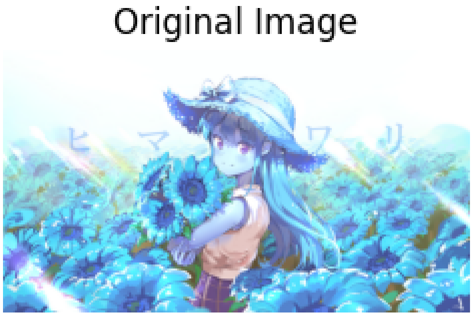
Practical: Implementing feature extraction techniques.

Topics: Implementing Harris corner detection algorithm.
Using OpenCV functions to extract SIFT, SURF, and ORB features.
Extracting Histogram of Oriented Gradients(HOG) features for object detection.
Visualizing extracted features and descriptors.

Code: (With output)

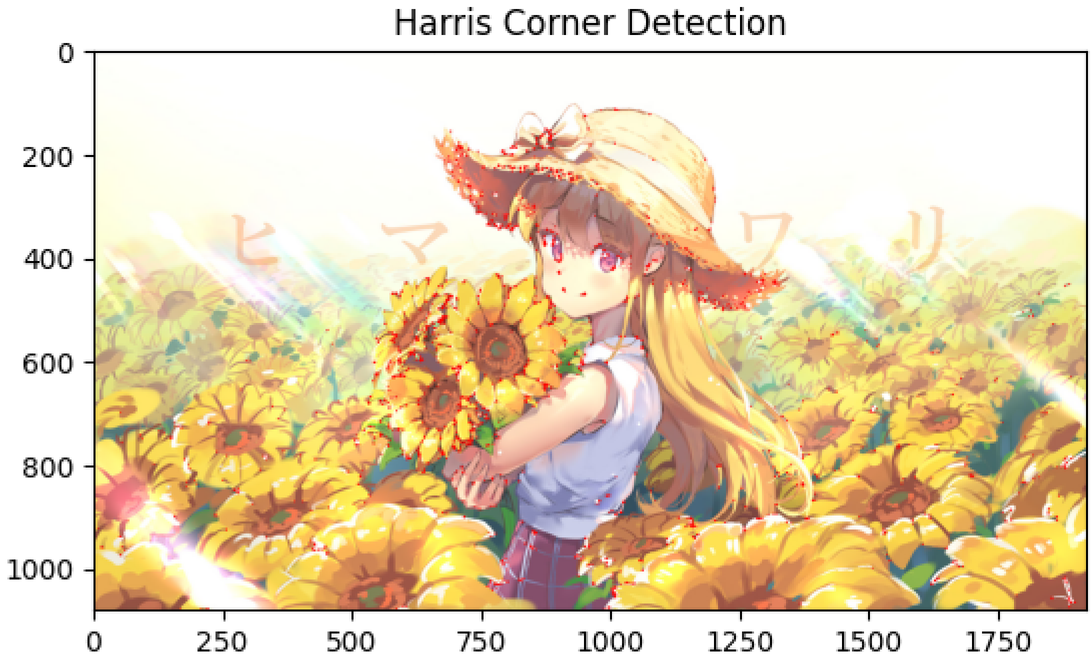
```
import cv2
from PIL import Image, ImageOps
import numpy as np
import matplotlib.pyplot as plt
from google.colab.patches import cv2_imshow
from google.colab import files
uploaded = files.upload()

#load an image
image=cv2.imread('nn.jpg')
gray_image = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
plt.figure(figsize=(3, 3))
plt.imshow(image)
plt.title('Original Image')
plt.axis('off')
plt.show()
```



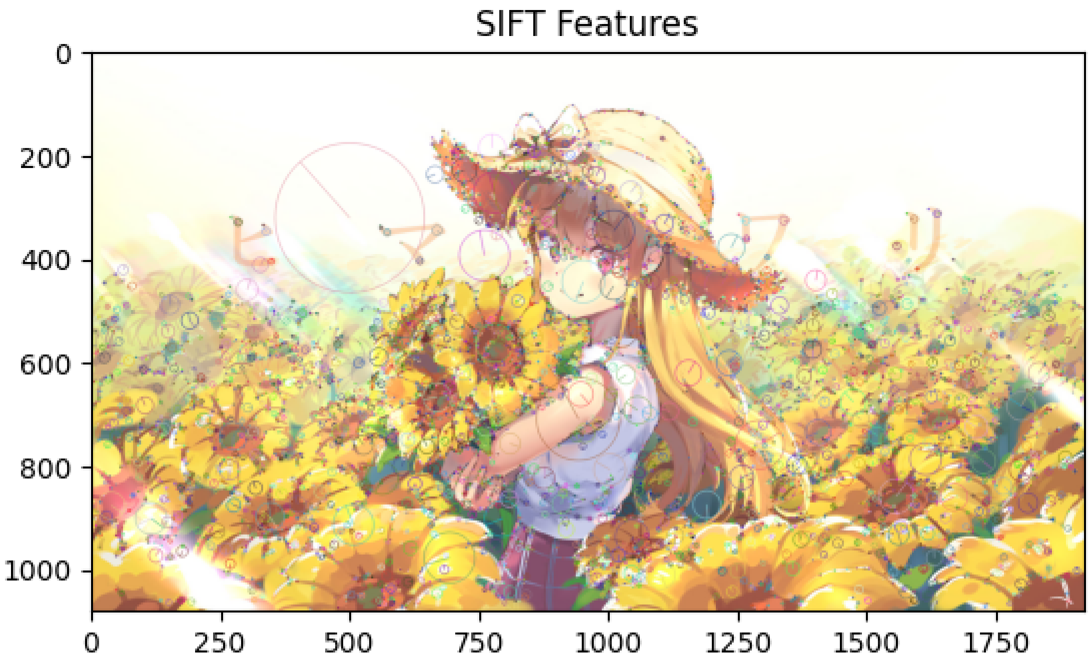
```
#Harris Corner Detection Algorithm
# Convert to floating-point for calculations
gray_image = np.float32(gray_image)
# Apply cornerHarris function
dst = cv2.cornerHarris(gray_image, blockSize=2, ksize=3, k=0.04)
# Dilate to mark the corners
dst = cv2.dilate(dst, None)
# Threshold for an optimal value, it may vary depending on the image.
image[dst > 0.01 * dst.max()] = [0, 0, 255]  # Mark corners in red
```

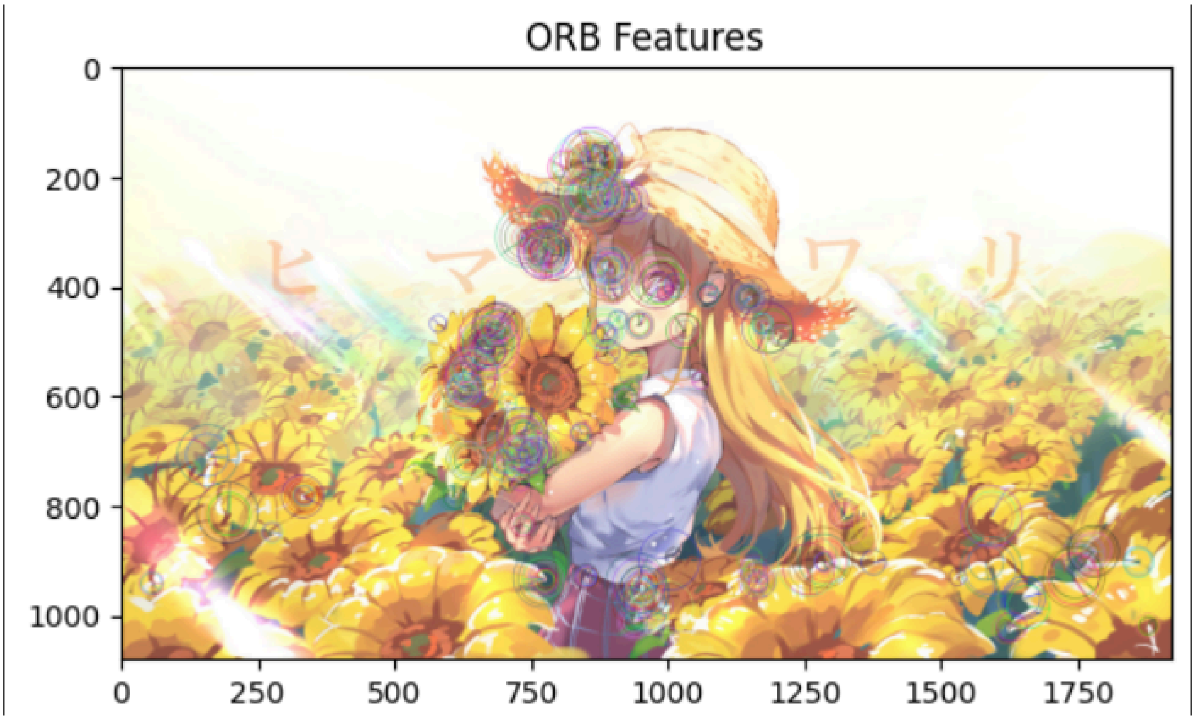
```
plt.imshow(cv2.cvtColor(image, cv2.COLOR_BGR2RGB))
plt.title('Harris Corner Detection')
plt.show()
```



```
!pip install opencv-contrib-python==4.7.0.72
```

```
#Using OpenCV functions to extract SIFT, SURF, and ORB features.
import cv2
import matplotlib.pyplot as plt
image = cv2.imread('nn.jpg')  # Replace with your image path
gray_image = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
sift = cv2.SIFT_create()
keypoints_sift, descriptors_sift = sift.detectAndCompute(gray_image, None)
orb = cv2.ORB_create()
keypoints_orb, descriptors_orb = orb.detectAndCompute(gray_image, None)
# SIFT
image_sift = cv2.drawKeypoints(image.copy(), keypoints_sift, None,
flags=cv2.DRAW_MATCHES_FLAGS_DRAW_RICH_KEYPOINTS)
plt.imshow(cv2.cvtColor(image_sift, cv2.COLOR_BGR2RGB))
plt.title('SIFT Features')
plt.show()
# ORB
image_orb = cv2.drawKeypoints(image.copy(), keypoints_orb, None,
flags=cv2.DRAW_MATCHES_FLAGS_DRAW_RICH_KEYPOINTS)
plt.imshow(cv2.cvtColor(image_orb, cv2.COLOR_BGR2RGB))
plt.title('ORB Features')
plt.show()
```

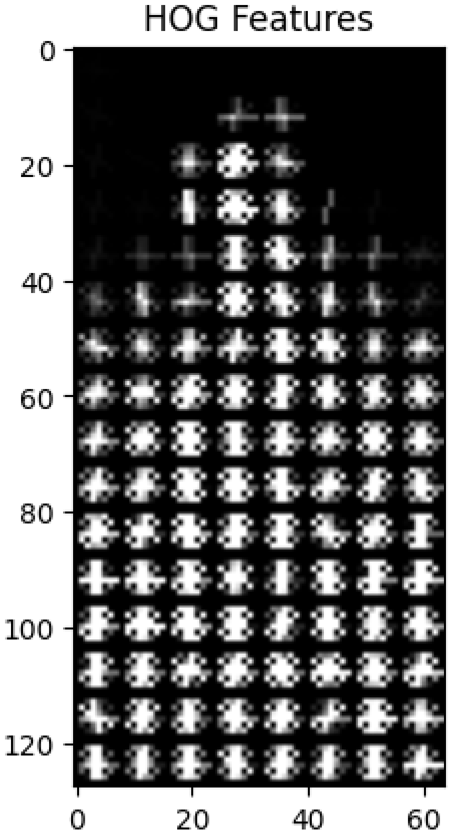




```
# Histogram of Oriented Gradients(HOG) features for object detection
from skimage.feature import hog
from skimage import exposure
# Resize if needed
image_resized = cv2.resize(image, (64, 128))
# Convert to grayscale if not already
gray_image = cv2.cvtColor(image_resized, cv2.COLOR_BGR2GRAY)
fd, hog_image = hog(gray_image, orientations=9, pixels_per_cell=(8, 8),
                cells_per_block=(2, 2), visualize=True)
```

```
hog_image_rescaled = exposure.rescale_intensity(hog_image, in_range=(0, 10))

plt.imshow(hog_image_rescaled, cmap=plt.cm.gray)
plt.title('HOG Features')
plt.show()
```


HOG Features