**Code Explanation for Edge Detection of an image using Canny**

**import cv2**

**import matplotlib.pyplot as plt**

**# Load the image in grayscale**

**image = cv2.imread('C:/Users/asus/Desktop/Tuwaiq Academy/Dr. Afshan/horse.jpg', cv2.IMREAD\_GRAYSCALE)**

**# Apply the Canny edge detection algorithm**

**edges = cv2.Canny(image, 100, 200)**

**# Display the edges using matplotlib**

**plt.figure(figsize=(8, 8))**

**plt.imshow(edges, cmap='gray')**

**plt.title('Edges Detected')**

**plt.axis('off') # Hide the axes**

**plt.show()**

**Output**



1. Importing Libraries:
2. import cv2
3. import matplotlib.pyplot as plt

cv2 (OpenCV) is used for reading and processing images.

matplotlib.pyplot (as plt) is used for displaying images.

1. Load the Image in Grayscale:
2. image = cv2.imread('C:/Users/asus/Desktop/Tuwaiq Academy/Dr. Afshan/horse.jpg', cv2.IMREAD\_GRAYSCALE)

cv2.imread() loads the image from the specified file path.

cv2.IMREAD\_GRAYSCALE ensures the image is loaded in grayscale, which is required for edge detection since color information isn't necessary for identifying edges.

1. Apply Canny Edge Detection:
2. edges = cv2.Canny(image, 100, 200)

cv2.Canny() applies the Canny edge detection algorithm to the grayscale image.

Thresholds 100 and 200:

* + - Pixels with gradient values above 200 are considered strong edges.
    - Pixels with gradient values below 100 are discarded as non-edges.
    - Pixels with gradient values between 100 and 200 are considered weak edges, and only retained if connected to strong edges.

1. Display the Edges Using Matplotlib:
2. plt.figure(figsize=(8, 8))
3. plt.imshow(edges, cmap='gray')
4. plt.title('Edges Detected')
5. plt.axis('off') # Hide the axes
6. plt.show()

plt.figure(figsize=(8, 8)): Sets the display figure size to 8x8 inches.

plt.imshow(edges, cmap='gray'): Displays the result of the edge detection using a grayscale colormap.

plt.title('Edges Detected'): Adds a title above the image.

plt.axis('off'): Hides the axis labels for a cleaner view of the image.

plt.show(): Displays the image with the detected edges.

Summary:

* This code loads an image, applies Canny edge detection to detect edges, and then displays the result using Matplotlib.
* Canny edge detection identifies areas with rapid intensity changes in the image, which often correspond to the edges of objects.