Vidyavardhini's College of Engineering &Technology

Department of Computer Engineering

To study Edge detection with Canny
Date of Performance: 31/07/23
Date of Submission:7/08/23

Aim: To study Edge detection with Canny

Objective: Perform Canny Edge detector using Noise reduction using Gaussian filter ,Gradient calculation along the horizontal and vertical axis
Non-Maximum suppression of false edges ,Double thresholding for segregating strong and weak edges ,Edge tracking by hysteresis

Theory: The Canny edge detector is an edge detection operator that uses a multistage algorithm to detect a wide range of edges in images. It was developed by John F. Canny in 1986. Canny also produced a computational theory of edge detection explaining why the technique works.

What are the three stages of the Canny edge detector
To fulfill these objectives, the edge detection process included the following stages.

Stage One - Image Smoothing.
Stage Two - Differentiation.
Stage Three - Non-maximum Suppression.

The basic steps involved in this algorithm are:

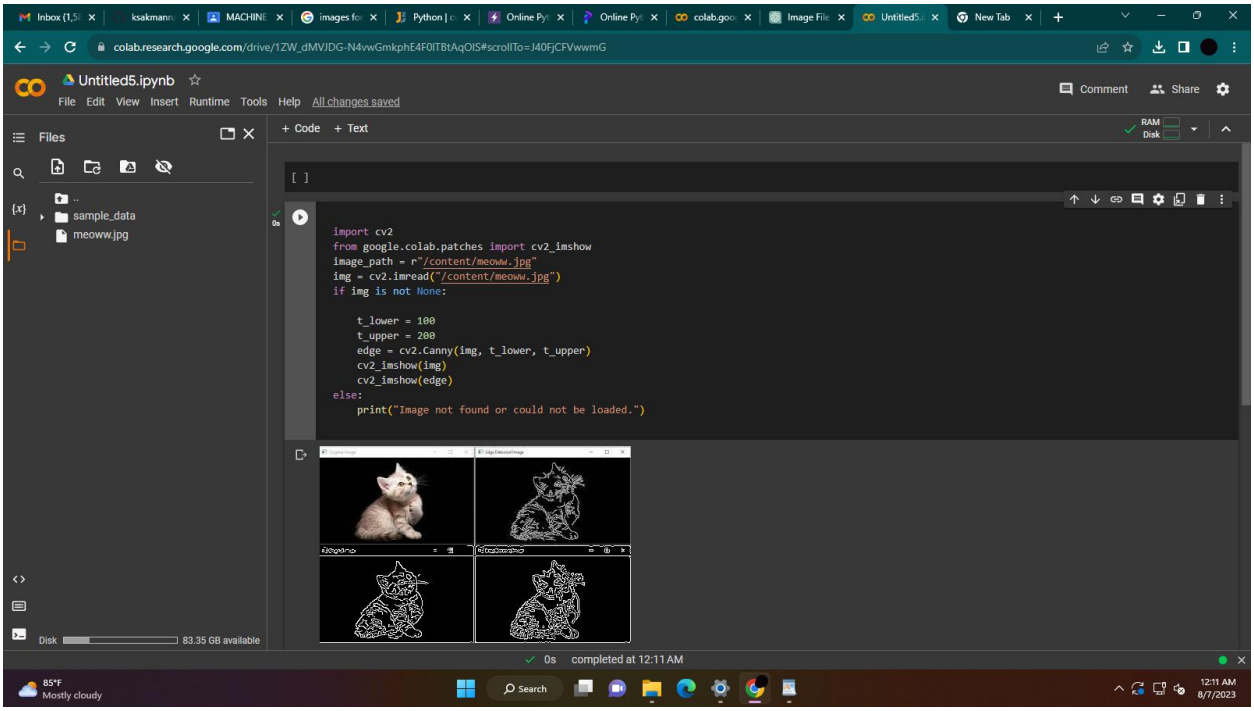
- Noise reduction using Gaussian filter
- Gradient calculation along the horizontal and vertical axis
- Non-Maximum suppression of false edges
- Double thresholding for segregating strong and weak edges
- Edge tracking by hysteresis

Since a weak edge pixel caused by true edges will be connected to a strong edge pixel, pixel W with weak gradient is marked as edge and included in the final edge map if and only if it is involved in the same connected component as some pixel S with strong gradient. In other words, there should be a chain of neighbor weak pixels connecting W and S (the neighbors are 8 pixels around the considered one). We will make up and implement an algorithm that finds all the connected components of the gradient map considering each pixel only once. After that, you can decide which pixels will be included in the final edge map.
Below is the implementation.

```
import cv2
from google.colab.patches import cv2_imshow
image_path = r"/content/meoww.jpg"
img = cv2.imread("/content/meoww.jpg")
if img is not None:

    t_lower = 100
    t_upper = 200
    edge = cv2.Canny(img, t_lower, t_upper)
    cv2_imshow(img)
    cv2_imshow(edge)
else:
    print("Image not found or could not be loaded.")
```

Output:



Conclusion: In this I have learned the Canny edge detection algorithm and its implementation using the OpenCV library. I have performed the following steps: Importing Libraries , Reading the Image , Setting Parameters , Applying Canny Edge Detection , Displaying Results .