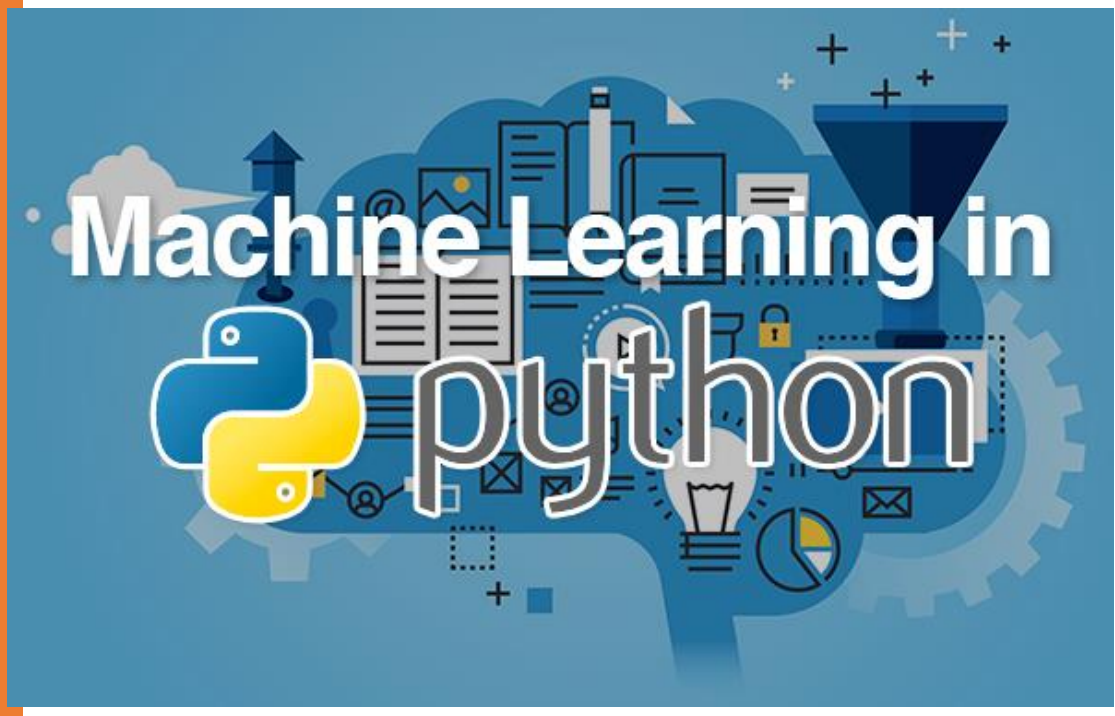
Computer Vision Problem - The Edge Of Laddoos



By-

SHIVAM KUMAR GIRI

python™ OpenCV

PROBLEM STATEMENT

This project is designed to test your mastery over Computer Vision through an interactive edge detection assignment. Write Python programs to detect the edge of laddoos. The image details are included in the assignment below. Put comment lines in the program to show your implementation steps. Make suitable assumptions if required. The programs should directly run in the python 3 environment without any modifications.

ABSTRACT

PIL is the Python Imaging Library which provides the python interpreter with image editing capabilities.

Here I am attaching a python code, which can read the designated Image and can detect the edges. And display the output

METHODOLOGY

Step 1: Load the libraries

Basic Libraries are loaded such as pillow(for image Processing), and cv2(for computer vision)

Step 2: Get the image paths and setup output directory

In this Project:

Input image = laddu.jpeg

Output image = output_laddu.jpeg

Optimized output image = optimized_laddu.jpeg

Step 3: Display the basic input file

Step 4: Read the image_ input file

Here I used cv2.imread() to read the file

Step 5: Find the Edge of the image_ input file

```
#Convert the image einto gray shades
gray = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
#Blur the gray Image to remove/whiten non significant part of image
blurred = cv2.GaussianBlur(gray, (11, 11), 0)
#Convert the image into outline to get edges
canny = cv2.Canny(blurred, 30, 150)
#Finding all contours in aboyve image
contours, hierarchy = cv2.findContours(canny.copy(), cv2.RETR_TREE, cv2.CHAIN_APP
ROX_SIMPLE)
#Inscribe contours on orignal image
image = cv2.drawContours(image, contours, -1, (0, 255, 0), 2)
```

I set the initial canny as 150 to check the output

Step 6: Display the output Image file

Step 7: Optimize the edges

As the canny is more, it's not detecting all Edges. This could be optimized to a larger number of edges as It can give much better edges, hence, code is optimized. Here I used the canny and Found 90 as a better edge compared to 150

Step 8: Display the output optimized Image file

Xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx**End**xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx