Edge detection

Abdullah
180374

21 December, 2019

Abstract

Abstract. Edge detection is a type of image segmentation techniques which determines the presence of an edge . The main purpose of edge detection is to find the edges in the Picture. Generally, an edge is defined as the boundary pixels that connect two separate regions with changing image amplitude attributes . In this paper, we present methods for edge segmentation of images; we used three techniques i.e Sobel ,Prewitt and Canny technique and they are compared with one another so as to choose the best technique for edge detection segment image. These techniques applied on one image to find edges on edge detection image.we use PYTHON and MATLAB software to find edges in an Image then we will use these edges to find congruence between objects

1 Introduction

Edge detection is a fundamental tool used in most image processing applications to obtain information from the frames. This process detects outlines of an object and boundaries between objects and the background in the image. An edge-detection filter can also be used to improve the appearance of blurred image. Edge is a boundary between two homogeneous regions. Edge detection refers to the process of identifying and locating sharp discontinuities in an image. K-means clustering is a type of unsupervised learning, which is used when you have unlabeled data (i.e., data without defined categories or groups). The goal of this algorithm is to find groups in the data(cluster), with the number of groups represented by the variable K. The algorithm works iteratively to assign each data point to one of K groups based on the features that are provided.

2 Related Work

Kumar et al. presented fuzzy relation method in image processing technique for edge detection of various gray scale and color image. This method to analyze the edges connected with an image and it displays the accuracy of edge detection using laplacian method. The laplacian constructor method detects the edges of images with advanced divergence in actual manner which deliver improved result .

Anjum et al. implemented canny algorithm for efficient edge detection algorithm. The canny edge detection method operates the entire image and that proportional to the size of the image . This method removes the inherent dependency between various blocks can be processed in parallel and it is very fast edge detection algorithm shows clear videos and images.

3 Model

3.1 Pseudo Code

- Read the image and store it in 2d matrix.
- Store kernel Matrix (Sobel, Canny, Prewitt).
- Adjust Kernel according to Image Dimension.
- Multiply Kernel with image element wise and

store the Output on the Desired location.

- Add row and column and store the output in centre location.
- Print that 2d matrix (in form of image).

3.2 Code snippets

import numpy as np

import matplotlib.image as img

import matplotlib.pyplot as plt

from skimage import io

from operator import truediv

import numpy.matlib

img = io.imread('house.jpg', as$_gray = True$)

img = img*255

plt.imshow(img, cmap='gray', vmin=0, vmax=255)

plt.show()

imgplot = plt.imshow(img)

Sobelx = [[1, 0, 1],[1, 0, -1], [1, 0, -1]]

Sobely = [[1, 2, 1],[0, 0, 0], [-1, -2, -1]] sobel

Sobelx = [[-1, 0, 1],[-2, 0, 2], [-1, 0, 1]] canny

Sobely = [[-1, -2, -1],[0, 0, 0], [1, 2, 1]]

Sobelx = [[1, 0, -1],[1, 0, -1], [1, 0, -1]] prewitt
Sobely = [[1, 1, 1],[0, 0, 0], [-1, -1, -1]]

Mul$_{11}$ = $np.multiply(Image1, Sobel_kernel_X)$

Mul$_{21}$ = $np.multiply(Image1, Sobel_kernel_Y)$

for i in range(0,N$_Dimension[1] - 2, 3)$:

tempXX1[:,i+1]=Mul$_{11}[:, i]$ + $Mul_{11}[:, i + 1]$ + $Mul_{11}[:, i + 2]$;

for i in range(0,N$_Dimension[0] - 2, 3)$:

tempXX2[i+1,:]=tempXX1[i,:]+tempXX1[i+1,:]+tempXX

for i in range(0,N$_Dimension[1] - 2, 3)$:

tempYY1[:,i+1]=Mul$_{21}[:, i]$ + $Mul_{21}[:, i + 1]$ + $Mul_{21}[:, i + 2]$;

for i in range(0,N$_Dimension[0] - 2, 3)$:

tempYY2[i+1,:]=tempYY1[i,:]+tempYY1[i+1,:]+tempYY

ImageX[0:N$_Dimension[0], 0 : N_Dimension[1]] = tempXX2 + ImageX[0 : N_Dimension[0], 0 : N_Dimension[1]]$

ImageY[0:N$_Dimension[0], 0 : N_Dimension[1]] = tempYY2 + ImageY[0 : N_Dimension[0], 0 : N_Dimension[1]]$

ImageA = ImageX + ImageY

plt.imshow(ImageA, cmap='gray', vmin=0, vmax=255)

plt.show()

4 Experiment

We give our code an Image, Our code read that image and give the resultant image which highlight the edges of in the given image. Our Algorithm uses three kernel i.e Sobel, Canny and Prewitt. and according to kernel matrix, multiply it with Actual image and give the resultant image.

4.1 Result

This program will generate following Image after compiling, their are 3 output images for each kernel.

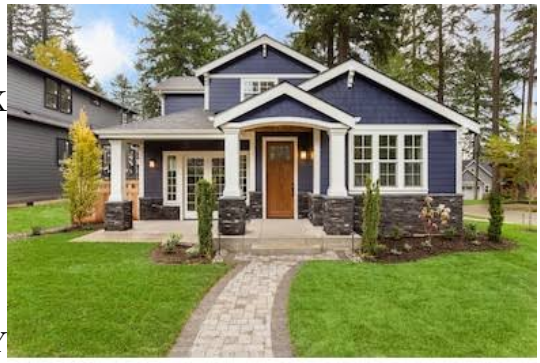

Figure 1: Actual Image.

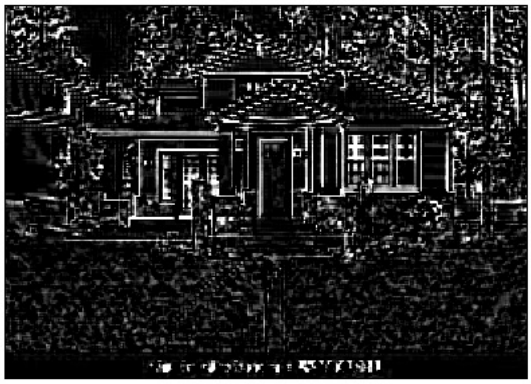

Figure 2: Edge Detection using Sobel kernel

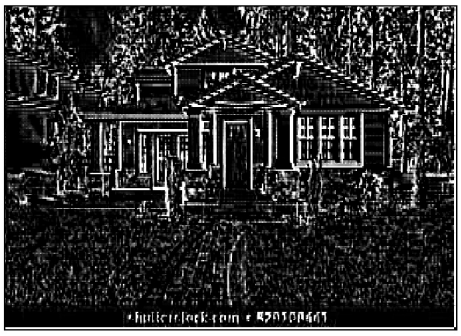

Figure 3: Edge Detection using Prewitt kernel

5 Conclusion

Edge detection is the important step in object extraction. we find Edges(objects) in a picture very easily using edge detection Algorithms.Therefore it becomes important to know about various edge detection techniques. In this paper we studied some edge detection techniques that are Sobel, Canny and Prewitt edge detection methods and are implemented as per the need of segmentation of image.

References

[1] Canny John, "A computational approach to edge detection", IEEE Transactions on Pattern Analysis and Machine Intelligence, PAMI-8(6), pp. 679-698, 1989

[2] X.L Xu, Application of matlab in digital image processing, Modern Computer, 43(5), pp. 35-37, 2008

[3] F Zhang, MATLAB digital image processing, [Beijing, Mechanical Industry, 2009].

[4] Ghassan Mahmoud Husien Amer, Edge Detection Method, [March 2015]