Image Segmentation through Edge Detection using Contrast Boosting

Pattern Recognition

Project Report

विश्वजीवनामृतं ज्ञानम्

ABV-INDIAN INSTITUTE OF INFORMATION TECHNOLOGY AND MANAGEMENT
GWALIOR-474 015

December 2020

Submitted To:

Dr. Vinal Patel

Submitted By:

Saksham Tewari (2017IMT-076)
Utkarsh Agrawal (2017IMT-084)
Siddharth Verma (2017IMT-106)

ABSTRACT

The method of partitioning a digital image into several segments (sets of pixels, also known as image objects) is known as Image Segmentation. The aim of segmentation is to simplify and/or alter the representation of an image into something that is easier to analyse and more meaningful. Segmentation of images is usually done to locate objects and boundaries (lines, curves, etc.). It is the process of assigning a label to every pixel in an image such that pixels with the same label share certain characteristics. There are various deep learning based method available to perform this task. In this project we use edge detection for segmenting the objects out of the images in the dataset without using any neural network. We first grayscale the image for ease in application of enhancement methods after which to increase the accuracy of the segmentation process we use thresholding for enhancing the image by increasing the contrast for better edge detection. After finding the edges of the object the output is used to create a segmentation mask for the image. The mask is then stacked onto the original image and the original pixel values are kept in the segmented region. There are various applications of Image Segmentation, like in case of medical image processing, the area of interest is extracted from the X-rays which leads to better visualization and better diagnosis of disease efficiently.

Keywords: Image Segmentation, Thresholding, Edge Detection, Contrast Boosting, Deep Learning.

Contents

1 Introduction

The theory and algorithms of placing abstract objects, e.g. measurements made on physical objects, into groups are concerned with pattern recognition techniques. Usually, the categories are believed to be identified in advance, whereas the categories are learned through methods (clustering). Pattern recognition methods are useful in many applications, such as data extraction, data mining, document image analysis and recognition, computer linguistics, forensics, biometrics, and bioinformatics.

Image segmentation, is a widely used technique in digital image processing and analysis to divide an image into several sections or regions. Image segmentation might involve separating the foreground from the background, or clustering pixel regions based on color or shape similarities. Image segmentation is used for clustering pixels into salient image regions i.e. regions corresponding to individual surfaces, objects, or pieces of objects, for object recognition, occlusion boundary estimation within motion or stereo systems, image compression, image editing, or image database look-up.

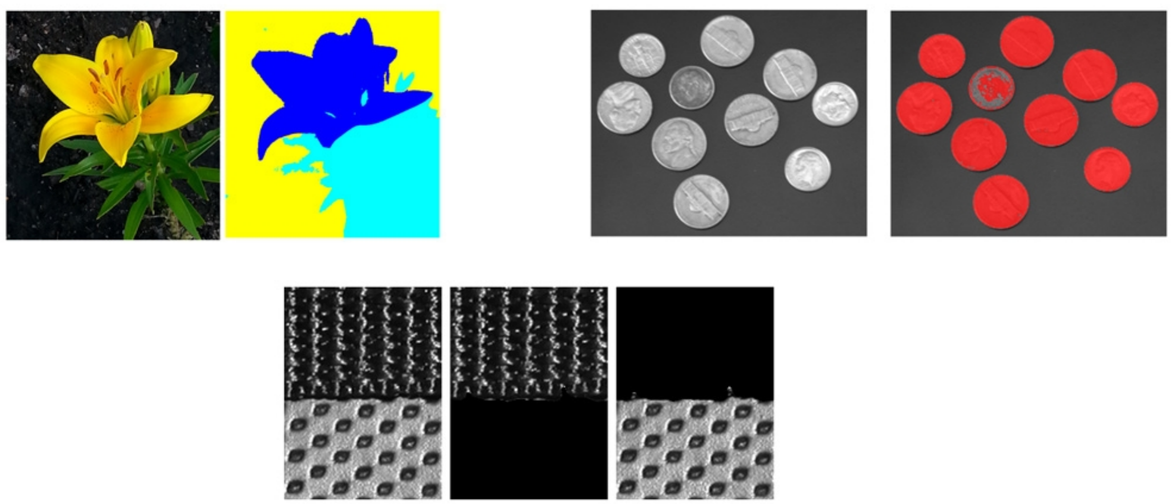

Figure 1: Segmenting regions based on color values, shapes, or texture.

Edge detection involves a number of mathematical methods aimed at detecting points in a digital image where the brightness of the image varies sharply or discontinuities are more formally present. Typically, the points at which image brightness sharply shifts are grouped into a collection of curved line segments called edges. Ideally, applying an edge detector to an image may result in a collection of linked curves indicating the boundaries of objects, the boundaries of surface markings, as well as curves corresponding to surface orientation discontinuities. The application of an edge detection algorithm to an image will therefore significantly decrease the amount of data to be processed and can therefore filter out information that can be considered less important, while preserving the important structural properties of an image.

1.1 Dataset Description

We have selected a large dataset of Canadian Maple leaves and augmented it to add variability. We have selected the aforementioned dataset as the foreground object has a large number of jagged edges which will help us to fully understand the extent of prowess of this method in image segmentation.

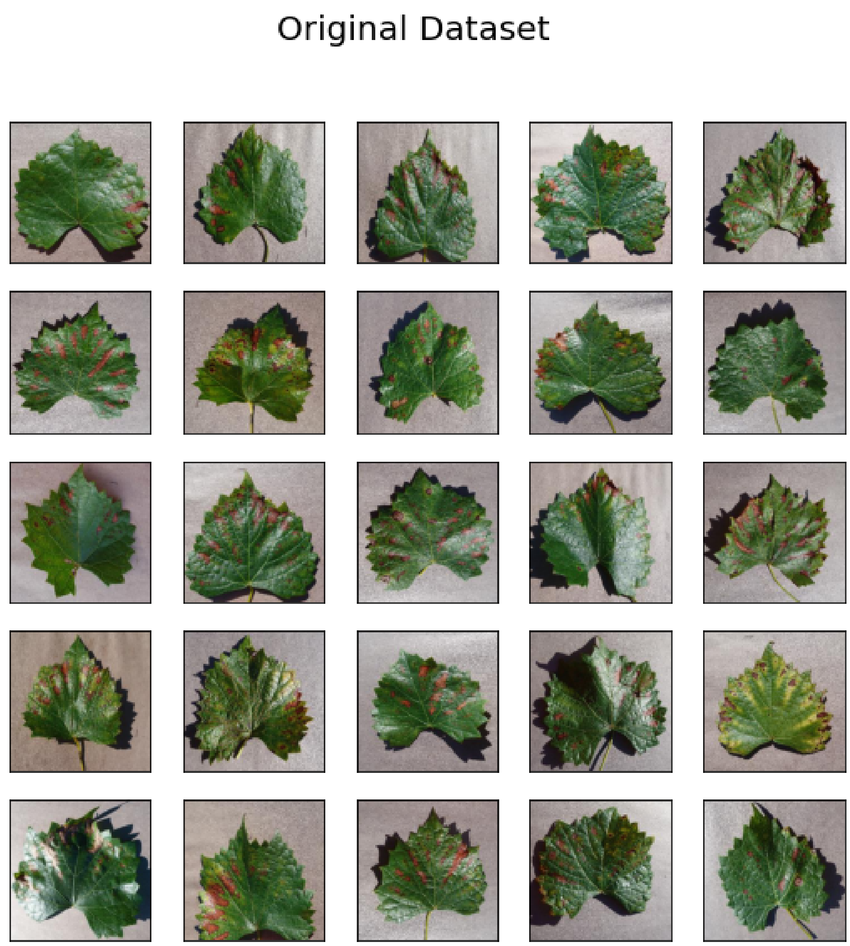

Figure 2: A sample of the original dataset

2 Expected research outcome

- The proposed method will result in a less complex, faster and lightweight method for segmenting simple objects from images.
- Proposed method will have a very short calculation time and can be useful in processes, with segmentation as a intermediate step, which do not require an accurate mask but rely on faster outputs.

3 Proposed Approach

The proposed method consists of multiple steps such as dataset Augmentation, Contrast boosting, dilation, edge detection and formation of segmentation masks.

3.1 Augmenting the dataset

The images present in the dataset have little to no variablity to the object position, lighting and orientation. Therefore to extensively test the proposed method we have randomly added variations to each image.

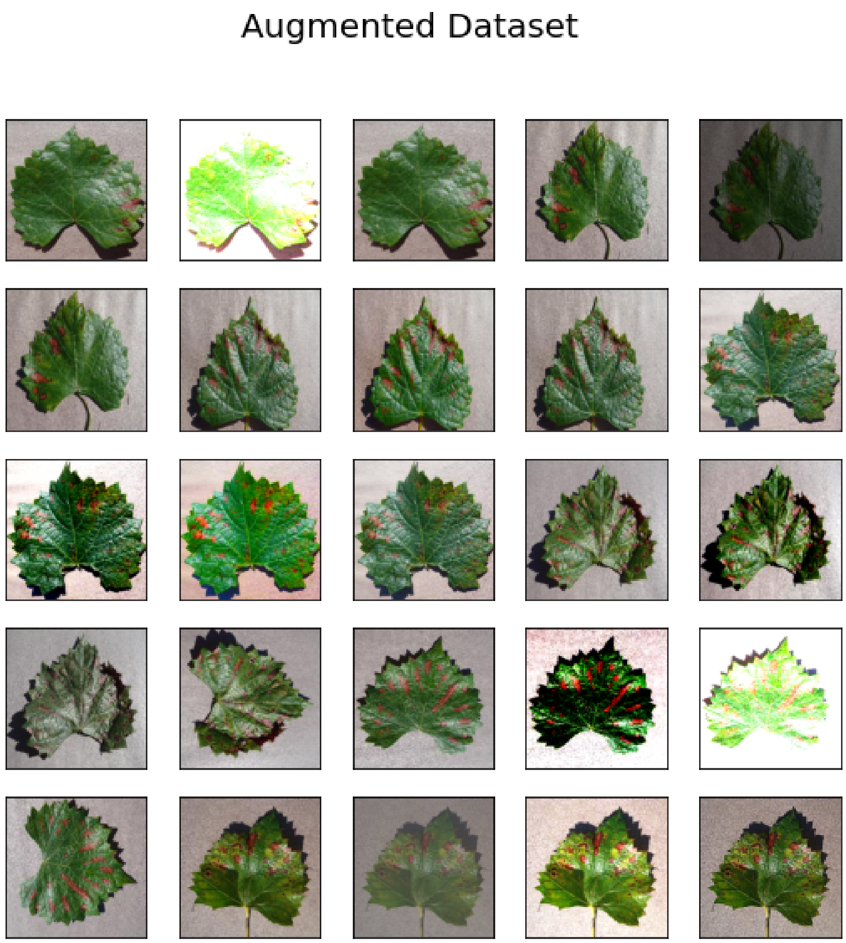

Figure 3: A sample of the augmented dataset

To the original images we have added variations as ***Rotation***, ***Sharpness***, ***Colour Intensity***, ***Brightness*** and ***Contrast***. Furthermore, we have added a random variable which decides whether an augmentation is to be applied to the image or not, which prevents unnecessary increase of dataset all the while introducing sufficeint amount of variability in the dataset.

3.2 Boosting Contrast through Thresholding

We will be using the simplest case of thresholding i.e. converting grascale images to binary image. In thresholding we define a boundary pixel value, all the pixels with value below this value are turned to pure black (pixel value = 0) and all the pixels with values above it are turned to pure white pixels (pixel valur = 255).

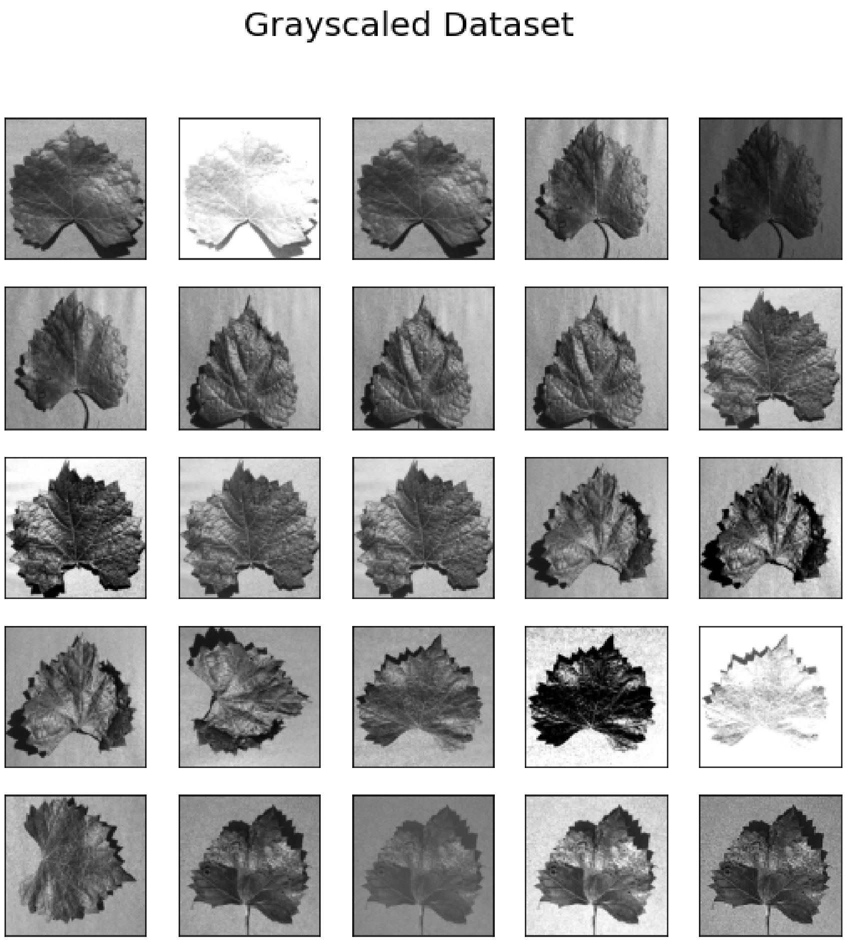

Figure 4: A sample of the grayscaled dataset

There are many techniques to find the optimal boundary pixel values but here we will be using Otsu's Binarization formula for calculating it. The Otsu algorithm aims at finding a threshold value (t) that minimises the weighted variance within the class provided by the relationship.

$$\sigma_\omega^2(t) = q_1(t)\sigma_1^2(t) + q_2(t)\sigma_2^2(t) \tag{1}$$

where,

$$q_1(t) = \Sigma_{i=1}^{t} P(i) \qquad \& \qquad q_2(t) = \Sigma_{t+1}^{I} P(i) \tag{2}$$

$$\mu_1(t) = \Sigma_{i=1}^{t} \frac{iP(i)}{q_1(t)} \qquad \& \qquad \mu_2(t) = \Sigma_{t+1}^{I} \frac{iP(i)}{q_2(t)} \tag{3}$$

$$\sigma_1^2(t) = \Sigma_{i=1}^{t} [i - \mu_1(t)]^2 \frac{P(i)}{q_1(t)} \qquad \& \qquad \sigma_2^2(t) = \Sigma_{t+1}^{I} [i - \mu_2(t)]^2 \frac{P(i)}{q_2(t)} \tag{4}$$

It simply considers a value of $\boldsymbol{t}$ that lies between two peaks, such that there are small variances between all groups.

3.3 Edge Detection and Mask Formation

We initially dialate the input image. Dialation is a process in which a pixel element is marked '1'(pixel value = 255) if atleast one pixel under the kernel(chosen grid) is already marked '1'. So it increases the white region in the image or size of foreground object increases. Another use is to join the broken pieces of the foreground object which is the reason we are using it. After this we use the canny algorithm for edge detection.

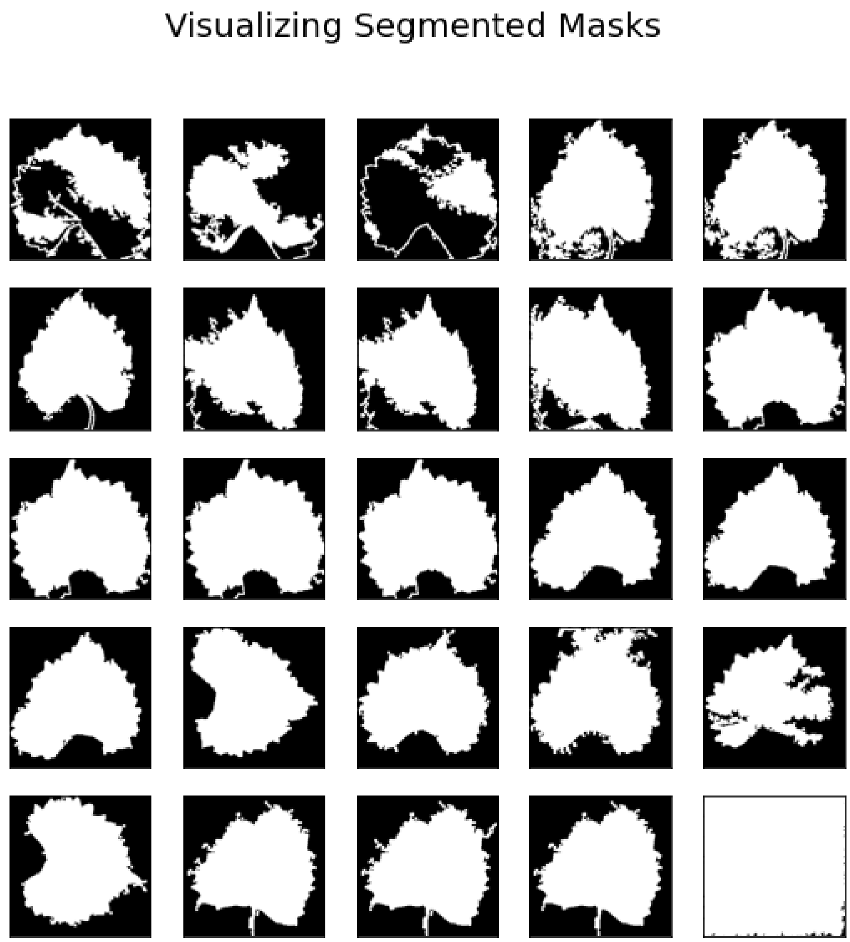

Figure 5: A sample of the Segmented Masks dataset

Now assuming that we have successfully obtained the foreground image edges we apply contours to our image to dissect the object from the image. Contours are simply a curve joining all the continuous points (along the boundary), having same color or intensity. They are specially useful for shape analysis and foreground object detection. After the foreground object is dissected it is filled with white color and stacked over a black canvas (2D array with all pixel value = 0) to obtain the final segmented mask.

4 Results

Finally, we multiply each pixel value of the original input image and the generated segmented masks and put the obtained value on another black canvas. So every pixel overlapping with a black pixel of the segmented mask is neglected and others are kept as it is.

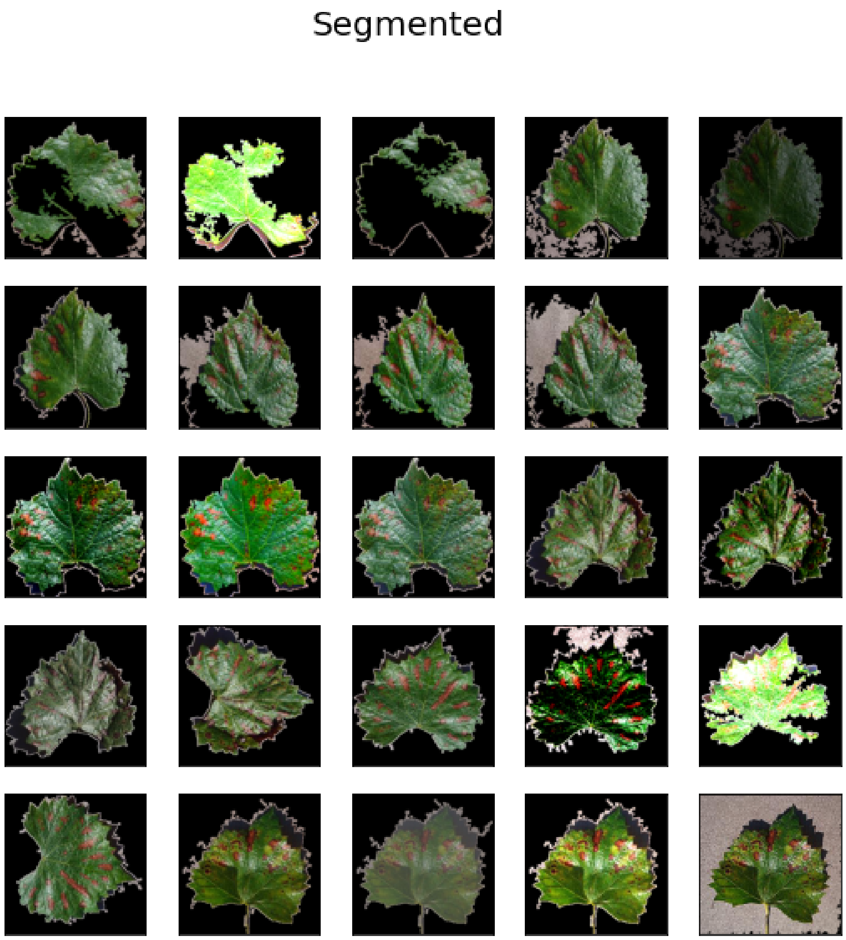

Figure 6: A sample of the Final Segmented dataset

This method is fast and low in complexity leading to faster but crude outcomes. Such methods can be used in real time applications which require faster outcomes rather than more accurate ones.

5 Conclusions

In summary, this approach for image segmentation is crude but fast and less computationally expensive leading to fast approximate results which is specially required in Real-Time applications. In comparison to CNN this approach is faster but lacks accuracy and can be tweaked specifically for a task unlike CNN networks that is a black box approach.

References

[1] R. M. Haralick and L. G. Shapiro, "Image segmentation techniques," *Computer vision, graphics, and image processing*, vol. 29, no. 1, pp. 100–132, 1985.

[2] M. H. Chowdhury and W. D. Little, "Image thresholding techniques," in *IEEE pacific Rim conference on communications, computers, and signal processing. Proceedings*. IEEE, 1995, pp. 585–589.

[3] S. D. Yanowitz and A. M. Bruckstein, "A new method for image segmentation," *Computer Vision, Graphics, and Image Processing*, vol. 46, no. 1, pp. 82–95, 1989.

[4] D. Bradley and G. Roth, "Adaptive thresholding using the integral image," *Journal of graphics tools*, vol. 12, no. 2, pp. 13–21, 2007.

[5] R. Maini and H. Aggarwal, "Study and comparison of various image edge detection techniques," *International journal of image processing (IJIP)*, vol. 3, no. 1, pp. 1–11, 2009.

[6] W. Rong, Z. Li, W. Zhang, and L. Sun, "An improved canny edge detection algorithm," in *2014 IEEE International Conference on Mechatronics and Automation*. IEEE, 2014, pp. 577–582.

[7] R. Biswas and J. Sil, "An improved canny edge detection algorithm based on type-2 fuzzy sets," *Procedia Technology*, vol. 4, pp. 820–824, 2012.

[8] D. Ziou, S. Tabbone *et al.*, "Edge detection techniques-an overview," *Pattern Recognition and Image Analysis C/C of Raspoznavaniye Obrazov I Analiz Izobrazhenii*, vol. 8, pp. 537–559, 1998.

[9] K. Ogawa, Y. Ito, and K. Nakano, "Efficient canny edge detection using a gpu," in *2010 First International Conference on Networking and Computing*. IEEE, 2010, pp. 279–280.