IMAGE SEGMENTATION

Seminar (IT290) Report

Submitted in partial fulfilment of the requirements for the degree of

BACHELOR OF TECHNOLOGY

In

INFORMATION TECHNOLOGY

by

ADARSH KISHORE (201IT103)

DURGA SUPRIYA HL (201IT121)

DEPARTMENT OF INFORMATION TECHNOLOGY

NATIONAL INSTITUTE OF TECHNOLGY KARNATAKA

SURATHKAL, MANGALORE - 575025

MAY - 2022

<u>DECLARATION</u>

I hereby *declare* that the *Seminar (IT290) Report* entitled “Image Segmentation” which is being submitted to the National Institute of Technology Karnataka Surathkal, in partial fulfilment of the requirements for the award of the Degree of Bachelor of Technology in the Department of Information Technology, is a ***bonafide report of the work carried out by me.*** The material contained in this seminar report has not been submitted to any University or Institution for the award of any degree.

Adarsh Kishore (201IT103)

Durga Supriya HL (201IT121)

Place: NITK, Surathkal

Date: May 5, 2022

CERTIFICATE

This is to certify that the Seminar entitled "Image Segmentation" has been presented by Adarsh Kishore (201IT103) and Durga Supriya HL (201IT121), students of IV semester B.Tech. (IT), Department of Information Technology, National Institute of Technology Karnataka, Surathkal, on May 5, during the even semester of the academic year 2022-2023, in partial fulfilment of the requirements for the award of the degree of Bachelor of Technology in Information Technology.

Dr. Bhawana Rudra
Project Guide

Place: NITK, Surathkal
Date: May 5, 2022

Abstract

In digital image processing and computer vision, image segmentation is the process of partitioning a digital image into multiple image segments, also known as image regions or image objects (sets of pixels). The goal of segmentation is to simplify and/or change the representation of an image into something that is more meaningful and easier to analyze. Image segmentation is typically used to locate objects and boundaries (lines, curves, etc.) in images. More precisely, image segmentation is the process of assigning a label to every pixel in an image such that pixels with the same label share certain characteristics.

The result of image segmentation is a set of segments that collectively cover the entire image or a set of contours extracted from the image (see edge detection). Each of the pixels in a region is similar with respect to some characteristic or computed property, such as colour, intensity, or texture. Adjacent regions are significantly different in colours with respect to the same characteristic(s). When applied to a stack of images, typical in medical imaging, the resulting contours after image segmentation can be used to create 3D reconstructions with the help of interpolation algorithms like marching cubes.

We have tried implementing image segmentation by converting the image into an equivalent graph and solving the max-flow/min-cut problem. In this seminar, we look into the theory of image segmentation, as well as an implementation of it in Python.

Table of contents

List of Figures

1 INTRODUCTION

1.1 GENERAL IMAGE SEGMENTATION

Image segmentation is an important problem in image theory. It is the problem of identifying individual components in an image. In easy words, it is the task of assigning labels to pixels. All picture elements or pixels belonging to the same category have a common label assigned to them. For example: Let's take a problem where the picture has to be provided as input for object detection. Rather than processing the whole image, the detector can be inputted with a region selected by a segmentation algorithm. This will prevent the detector from processing the whole image thereby reducing inference time.

Approaches to Image Segmentation

1) Similarity approach: This approach is based on detecting similarities between image pixels to form a segment, based on a threshold. ML algorithms like clustering are based on this type of approach to segment an image.

2) Discontinuity approach: This approach relies on the discontinuity of pixel intensity values of the image. Line, Point, and Edge Detection techniques use this type of approach for obtaining intermediate segmentation results which can be later processed to obtain the final segmented image.

There are mainly two classes of segmentation:

1) Classical computer vision approaches
2) AI-based methods

Due to the advent of computer technology image-processing techniques have become increasingly important in a wide variety of applications. Image segmentation is a classic subject in the field of image processing and also is a hotspot and focus of image processing techniques. Several general-purpose algorithms and techniques have been developed for image segmentation. Since there is no general solution to the image segmentation problem, these techniques often have to be combined with domain knowledge in order to effectively solve an image segmentation problem for a problem domain. Some of the groups of image segmentation are as follows -

1. **Semantic segmentation** is an approach to detecting, for every pixel, the belonging class of the object. For example, when all people in a figure are segmented as one object and the background as one object.

2. **Instance segmentation** is an approach that identifies, for every pixel, a belonging instance of the object. It detects each distinct object of interest in the image. For example, when each person in a figure is segmented as an individual object.

3. **Panoptic segmentation** combines semantic and instance segmentation. Like semantic segmentation, panoptic segmentation is an approach that identifies, for every pixel, the belonging class. Unlike semantic segmentation, panoptic segmentation distinguishes different instances of the same class.

In this project, we focus mainly on classical computer vision-based image segmentation, specifically graph-based image segmentation techniques. It is a type of semantic image segmentation. Edge-based segmentation is one of the most popular implementations of segmentation in image processing. It focuses on identifying the edges of different objects in an image.

Edge detection is widely popular because it removes unwanted and unnecessary information from the image. It reduces the image's size considerably, making it easier to analyse the same.

1.2 MOTIVATION

Image segmentation is an important image processing, and it seems everywhere if we want to analyze what is inside the image. For example, if we seek to find if there is a chair or person inside an indoor image, we may need image segmentation to separate objects and analyze each object individually to check what it is. Image segmentation usually serves as the pre-processing before image pattern recognition, image feature extraction and image compression. Research of it started around 1970, while there is still no robust solution, so we want to find the reason and see what we can do to improve it.

We have max flow/ Ford-Fulkerson theorem in Data Structures and Algorithms course. Given a graph which represents a flow network where every edge has a capacity and also given two vertices source ‘s’ and sink ‘t’ in the graph, the algorithm helps to find the maximum possible flow from s to t with the following conditions:

a) Flow on an edge doesn’t exceed the given capacity of the edge.

b) Incoming flow is equal to outgoing flow for every vertex except s and t.

The algorithm seemed interesting and on further research on the topic, we got to know that it can be used in image processing. More specifically speaking image segmentation.

2 LITERATURE REVIEW

For this project, we have mainly consulted on [1] to get the concept and formulas behind graph based image segmentation. It explains the calculation of edge weights and the working of the max-flow algorithm/min-cut algorithm to find a segmentation. It is the primary paper to which we have referred for our project.

[2] explains the general theory of image segmentation. It delves into the development of the formula for the general case, then shows how the special case for grid based images (i.e. those that can be represented by a 2D matrix) is exactly what we need to use for our project.

[3] explains an application of image segmentation in the field of medical imaging. It explains how image segmentation can be used to find the different organs which are visible in an MRI or CT scan. This paper helps us in understanding some of the real-world applications of segmentation.

[4] deals with general image segmentation theory. Although we approach only a specific approach in this project, image segmentation is a vast field with many techniques as discussed above. It is good to know for future work that such general classifications of the concept exist.

[5] deals with the differential image segmentation method, in which we use properties of PDEs to find segments. The methods based on statistic theory can work well on images with no noise or little noise. But the procedure of segmentation is difficult to be obtained and the accuracy of the result often depends on some artificial parameters. A lot of physical phenomena can be described by Partial Differential Equations (PDEs) and the related procedure is easy to be displayed. This is why PDE based methods are among the most powerful methods in image segmentation.

Early attempts to use combinatorial graph cut algorithms in vision were restricted to image clustering. In the late 90's a large number of new computer vision techniques appeared that figured out how to use min-cut/max-flow algorithms on graphs for solving more interesting non-binary problems. Sebastien Roy and Ingemar Cox were the first to use these algorithms to compute multi-camera stereo.

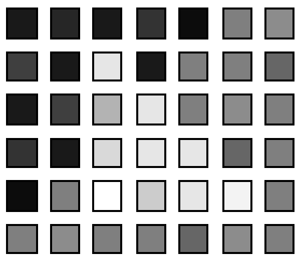


Fig 2.1 Image

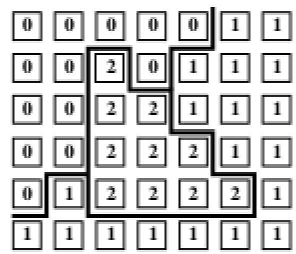


Fig 2.2 After labeling

An example of image labeling. An image in Fig 2.1 is a set of pixels P with observed intensities I p for each p ∈ P. A labeling L shown in Fig 2.2 assigns some label L p ∈ {0, 1, 2} to each pixel p ∈ P. Such labels can represent depth (in stereo), object index (in segmentation), original intensity (in image restoration), or other pixel properties. Normally, graph-based methods assume that a set of feasible labels at each pixel is finite. Thick lines in (b) show labeling discontinuities between neighbouring pixels.

2.1 REAL-WORLD APPLICATIONS

Image segmentation has uses in many areas. One of the prime areas which come to mind is the military. Enemy units often camouflage with the background, hiding from detection. It is undoubtedly useful to identify them from the background and take pre-emptive action. For this, image segmentation is important.

Another use is Photoshop. Current technologies allow us to cut images out of their backgrounds and paste them into another image, or visual effects applied to a particular section of the image. For this, we first need to identify the sections of images. Using image segmentation, we can achieve this.

Other applications are in medical imaging, object detection, and edge detection among others. Image Segmentation is also used in medical imaging, that is shape-based segmentation for this example. This is a method widely used in the medical industry, this method can be used on MRI images of the brain to segment different areas of the brain or segment areas of importance such as a tumour. It is also used to access a patient's cardiac function and diagnose cardiac disease, the left ventricle of the heart is segmented to gain data such as ejection fraction and stroke volume.

Object detection is a computer technology related to computer vision and image processing that deals with detecting instances of semantic objects of a certain class (such as humans, buildings, or cars) in digital images and videos. Well-researched domains of object detection include face detection and pedestrian detection. Object detection has applications in many areas of computer
vision, including image retrieval and video surveillance. Face detection is used in authentication, and pedestrian detection prevents accidents.

Since this is a clearly useful technology, we decided to choose this topic for the seminar.

3 TECHNICAL DISCUSSION

3.1 METHODOLOGY

This project focuses on using graph cuts to divide an image into background and foreground segments. The framework consists of two parts. First, a network flow graph is built based on the input image. Then a max-flow algorithm is run on the graph in order to find the min-cut, which produces the optimal segmentation.

3.1.1 Graph

A graph $G = (V, E)$ is a 2-tuple consisting of a set V of vertices and a set E of edges. The vertices are connected to each other by the edges. The edges can be directed, undirected, or a mix of both.

A weighted graph is one in which each edge has a weight associated with it. This can be interpreted as the cost of travelling from one end point to the other of the edge.

3.1.2 Max Flow/Min-Cut Problem

Consider a weighted graph $G = (V, E)$ where edges have a weight function $w: V \times V \rightarrow R$ associated with it. Now consider two special vertices s and t in the graph, which are called source and sink respectively. As the name suggests, the source vertex has all edges leading out of it, while the sink vertex has all edges leading into it.

Consider a different interpretation of the weight function. Consider water flowing from the source to sink, and the edge weights represent the maximum capacity of water that can flow through the edge. The max flow problem is concerned with finding the maximum possible flow from the source to the sink.

A cut in a graph is a partitioning of its vertices into two disjoint sets. A cut which separates the source and sink in the graph is called an $s - t$ cut. The capacity of a cut is simply the sum of the capacities of edges which cross through the cut. The max flow possible in a graph is equivalent to the capacity of the minimum cut in the graph.
Efficient algorithms exist for calculating the max flow/min-cut of a graph given a source and sink, like Ford-Fulkerson, Push-Relabel Algorithm etc.

3.1.3 Image to Graph

For converting the given image to a graph, we first resize the image to 30x30 and grayscale it. This is done so that there are no more than 900 vertices in the graph, which can be handled in a reasonable amount of time. Grayscaling is done so that we have to deal with one colour channel only, instead of three for R, G, B.

The intensity of a pixel is numerically equal to its grayscale value. A graph is constructed corresponding to the image by considering a vertex corresponding to each pixel. Adjacent pixels are connected to each other via edges. These edge weights are calculated by the following formula,

$$B(I_p, I_q) = 100 \cdot \exp\left(\frac{-(I_p - I_q)^2}{2\sigma^2}\right)$$

Equation - 3.1.1 The Boundary Penalty Function

For an edge (p, q), the boundary penalty is calculated as this where

1) I_p is the intensity of pixel p
2) I_q is the intensity of pixel q

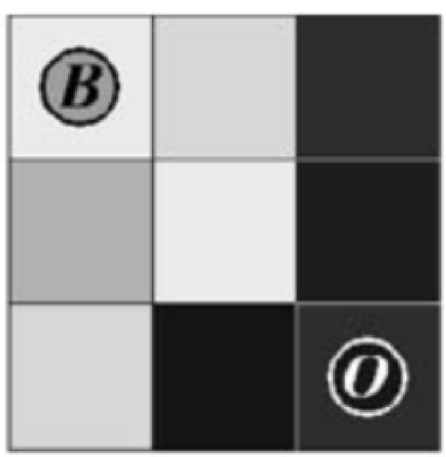
3) σ is an experimentally determined constant. It is a scaling factor and is usually taken as 30

This result is cast to an integer as flow algorithms work on integers.

The user labels some of these vertices explicitly as foreground or background vertices. These 'seeds' planted into the graph help in connecting two extra vertices called the SOURCE and SINK to the graph.

For example, all background seeds are connected to the SOURCE by edges of the same edge weight. For every background vertex, this edge weight K is calculated as

$$\mathcal{K} = \max(\{B(I_p, I_q) | (p,q) \in E\})$$



$\Rightarrow$

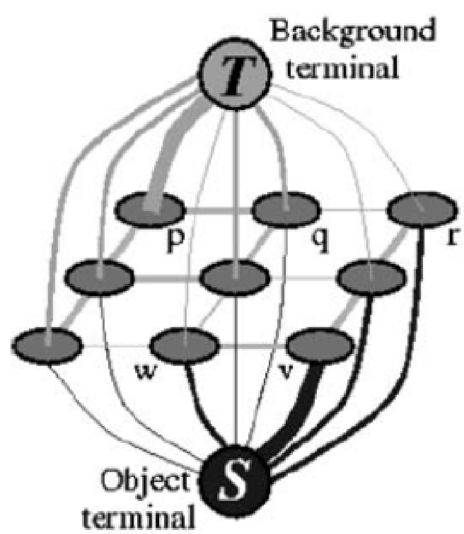


Fig-3.1 Image with seeds

Fig-3.2 Network defined on a simple 3x3 image with a background seed and a foreground seed

3.2 EXPERIMENTAL RESULTS

Picture 3.3

Fig-3.3(a): Picture before image segmentation

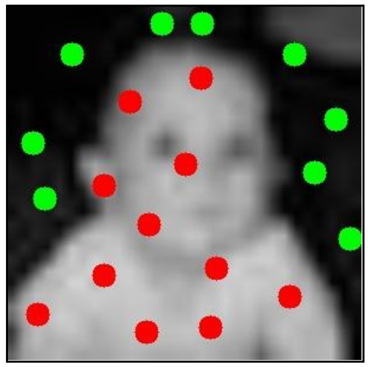


Fig-3.3(b): Picture after marking seeds

Fig-3.3(c): Picture after image segmentation

Picture 3.4

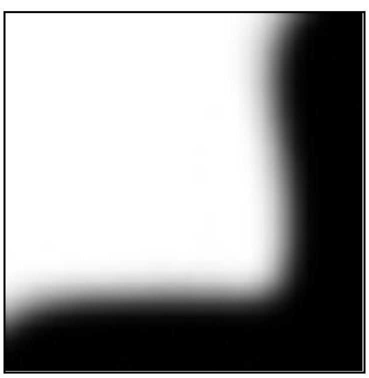


Fig-3.4(a): Picture before image segmentation

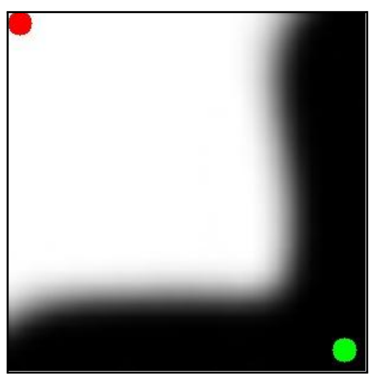


Fig-3.4(b): Picture after marking seeds

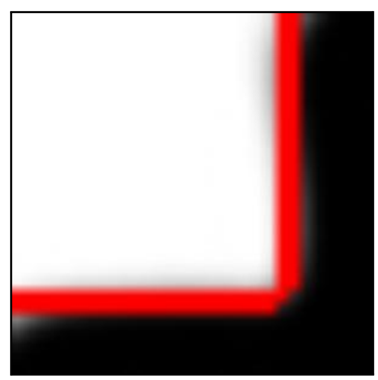


Fig-3.4(c): Picture after image segmentation

Picture 3.5

Fig-3.5(a): Picture before image segmentation

Fig-3.5(b): Picture after marking seeds

Fig-3.5(c): Picture after image segmentation

4 CONCLUSION AND FUTURE WORK

4.1 CONCLUSION

Although considerable research exists in the image segmentation literature, a majority of the methods rely on huge training datasets or require manual intervention to set the parameters. In contrast, only a handful of unsupervised approaches have been reported in the literature, many of which are computationally complex (e.g., normalized cuts) or require partial supervision (e.g., continuous maxflow). In the present work, we developed an approach to consistently estimate the flow capacity parameters leading to a fully unsupervised image segmentation approach. Our framework is based on iteratively estimating the image labels using a continuous max-flow approach followed by the MAP estimation of the flow capacities. The fact that it can do this much without using AI/ML shows how powerful graph-based algorithms can be.

4.2 FUTURE WORK

For future work, we can:

1. Image segmentation on coloured image:

   As of now, we are converting the given image first into its greyscale and then performing the image segmentation process. So, one of the future works could be to find the intensity differences in the coloured image itself and perform image segmentation.

2. Improve resolution from 30x30:

   In our present work, we are resizing the given image to a 30 x 30 scale first and then so that the maximum number of vertices would be 900. But this is decreasing the clarity of the image. So, one of the future works could be to increase the clarity.

REFERENCES

[1] Y. Boykov and V. Kolmogorov, "An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision," in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 26, no. 9, pp. 1124-1137, Sept. 2004, DOI: 10.1109/TPAMI.2004.60.

[2] Felzenszwalb, P.F., Huttenlocher, D.P. Efficient Graph-Based Image Segmentation. *International Journal of Computer Vision* 59, 167–181 (2004). https://doi.org/10.1023/B:VISI.0000022288.19776.77

[3] G. Wang, M. A. Zuluaga, W. Li, R. Pratt, P. A. Patel, M. Aertsen, T. Doel, A. L. David, J. Deprest, S. Ourselin et al., “DeepIGeoS: a deep interactive geodesic framework for medical image segmentation,” IEEE Trans. Pattern Anal. Mach. Intell., 2018.

[4] Nida M. Zaitoun, Musbah J. Aqel, Survey on Image Segmentation Techniques, Procedia Computer Science, Volume 65, 2015, Pages 797-806, ISSN 1877-0509.

[5] Bin Zhou, Xiao-Lin Yang, Rui Liu and Wei Wei, 2010. Image Segmentation with Partial Differential Equations. Information Technology Journal, 9: 1049-1052