MEF UNIVERSITY

Department of Engineering
Programming Studio

DIGITAL IMAGE PROCESSING

Ertuğrul Yılmaz
Instructor: Muhittin Gökmen
Date: 05 March 2019

Abstract

In this project I used two different algorithms to learn the digital image processing. There were two different algorithms called ‘Levialdi’s Shrinking Algorithm’ and ‘Two SubFields Parallel Shrinking Algorithm’. I used Levialdi for my work. And made a Graphic User Interface for user. With that algorithm you will get the number of components of an image.

Problem Definition

People need to work with pictures and images. But human eye is not reliable for image analysis. Point people need to work a different work area except human eye. Drawing may be a way to work on an image, but it is completely un efficient for flexibility. In this point digital image processing is very useful to get significant data from an image. With binary image you will have concrete data to work on. Binary image consists of numbers. To work with numbers, you will need operations. This is the need to implement a parallel shrink algorithm to work on a digital image. For example, if you are working on a microscope and you want to know how many bacteria in an area. You can implement an algorithm to learn the number of components in this area.

Solutions

Levialdi’s Parallel Shrinking Algorithm:

Levialdi algorithm starts from any corner of the image and using 0 to 1 erasements for its iterations. Levialdi is checking conditions for erasement processes. And when a pixel whose 8 neighbours are 0, that means there was a component there.

Work

I used library of tkinter to make a graphic user Interface and library of PILLOW to process on raw images. I took the image as input from user and work on it. I started the iteration from bottom left corner. Levialdi algorithm has three basic conditions. First one is if your pixel has the value of 1 and the West, South and SouthWest neighbours of your pixel is 0 then erase your pixel 1 to 0. Second condition is if your pixel is 1 and the West, North, East, South, NorthWest, NorthEast, SouthEast and SouthWest neighbours of your pixel are 0 then erase the pixel 1 to 0 and that condition means you found a component in the image. Third condition is if your pixel is 0 and the West and South neighbours of your pixel are 1 then erase your pixel 0 to 1. You will store the image data on an array and all of this conditions for your image array.

In gui I used tkinter. My grid was 3x3 to show all buttons, image, binaryImage, live iterations and number of components and iterations.

Results

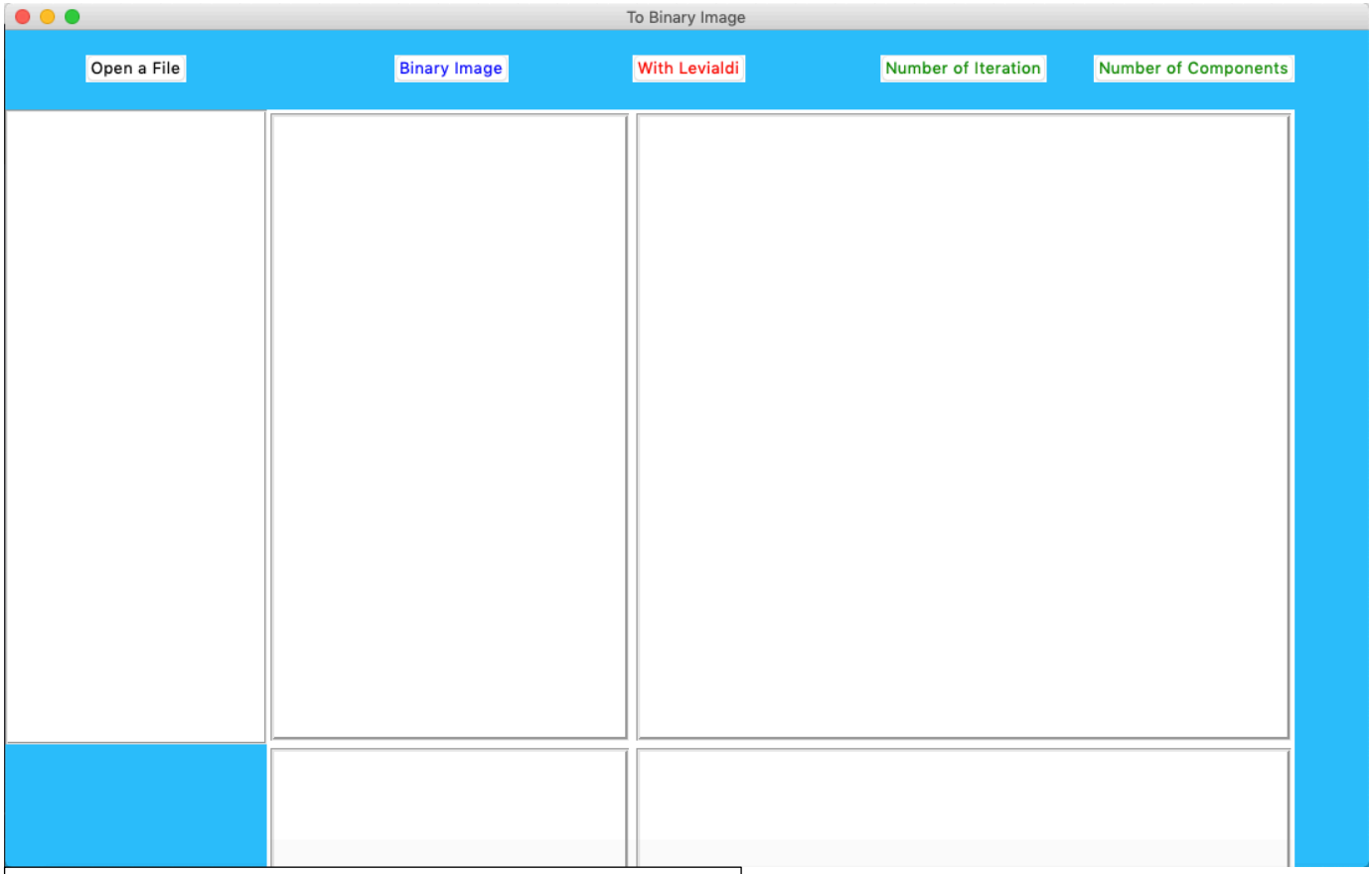


Figure 1. The Opening Screen in Program

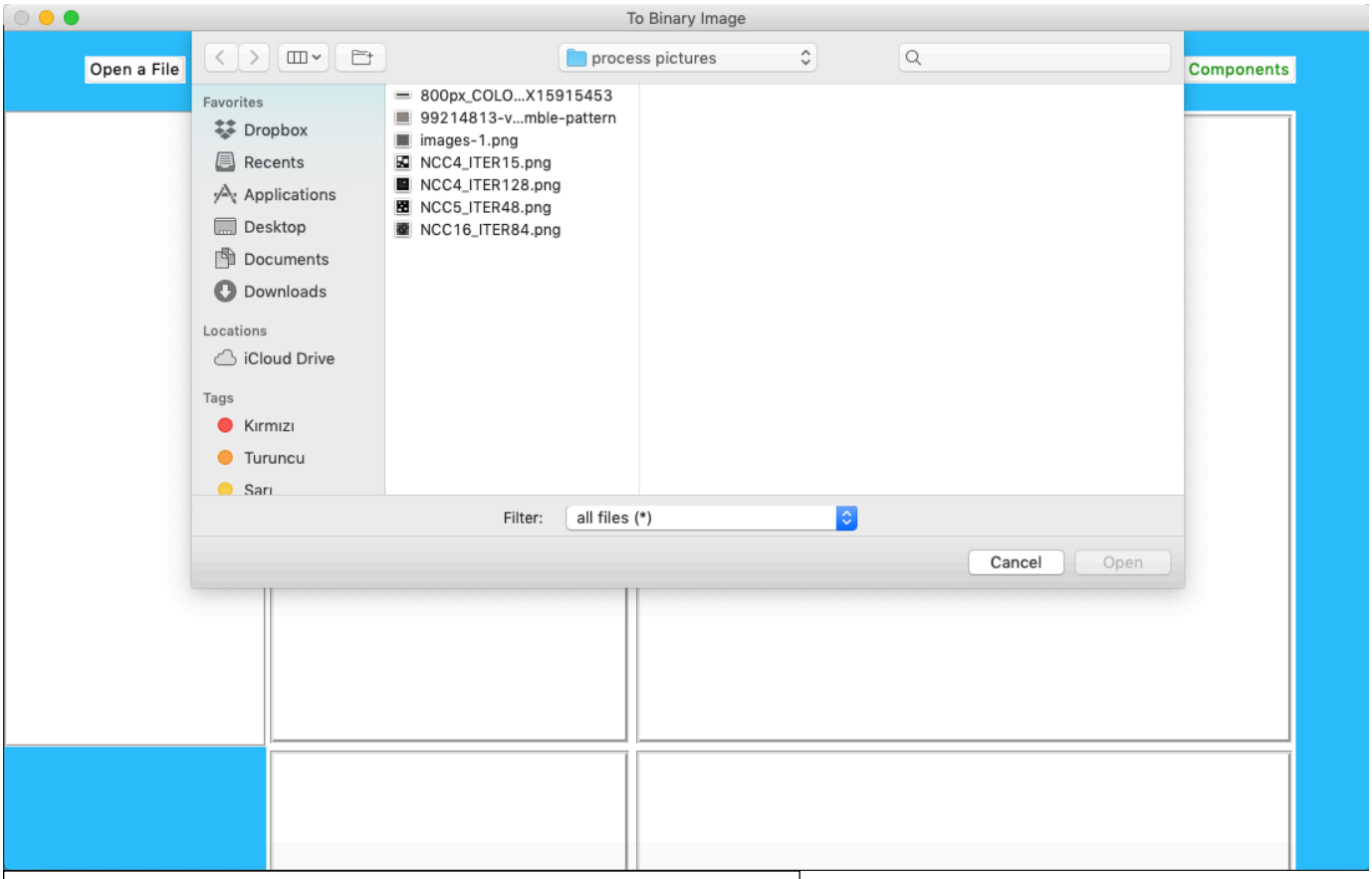


Figure 2. Taking Image from System Explorer

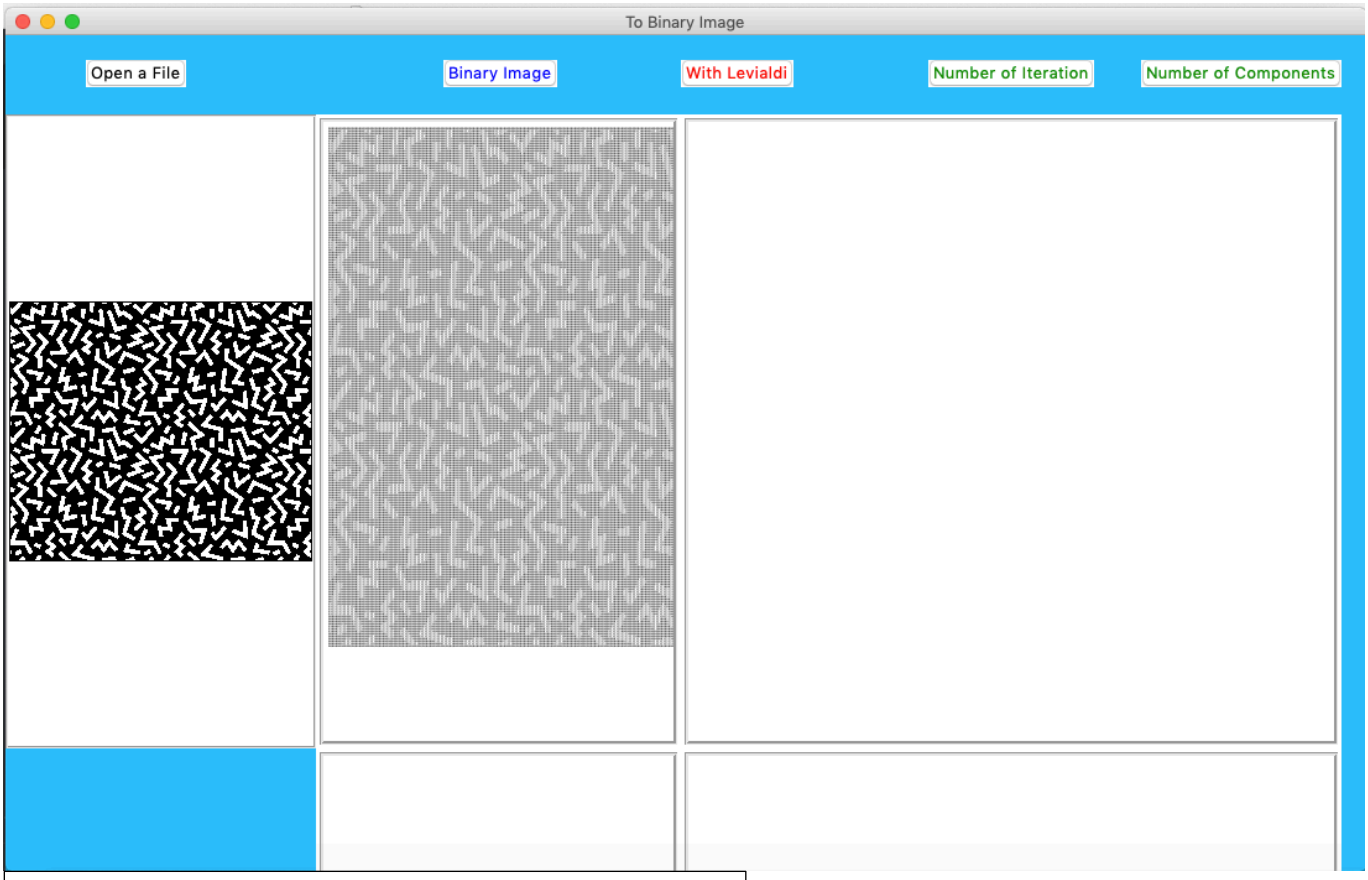


Figure 3. The Binary form of Input Image

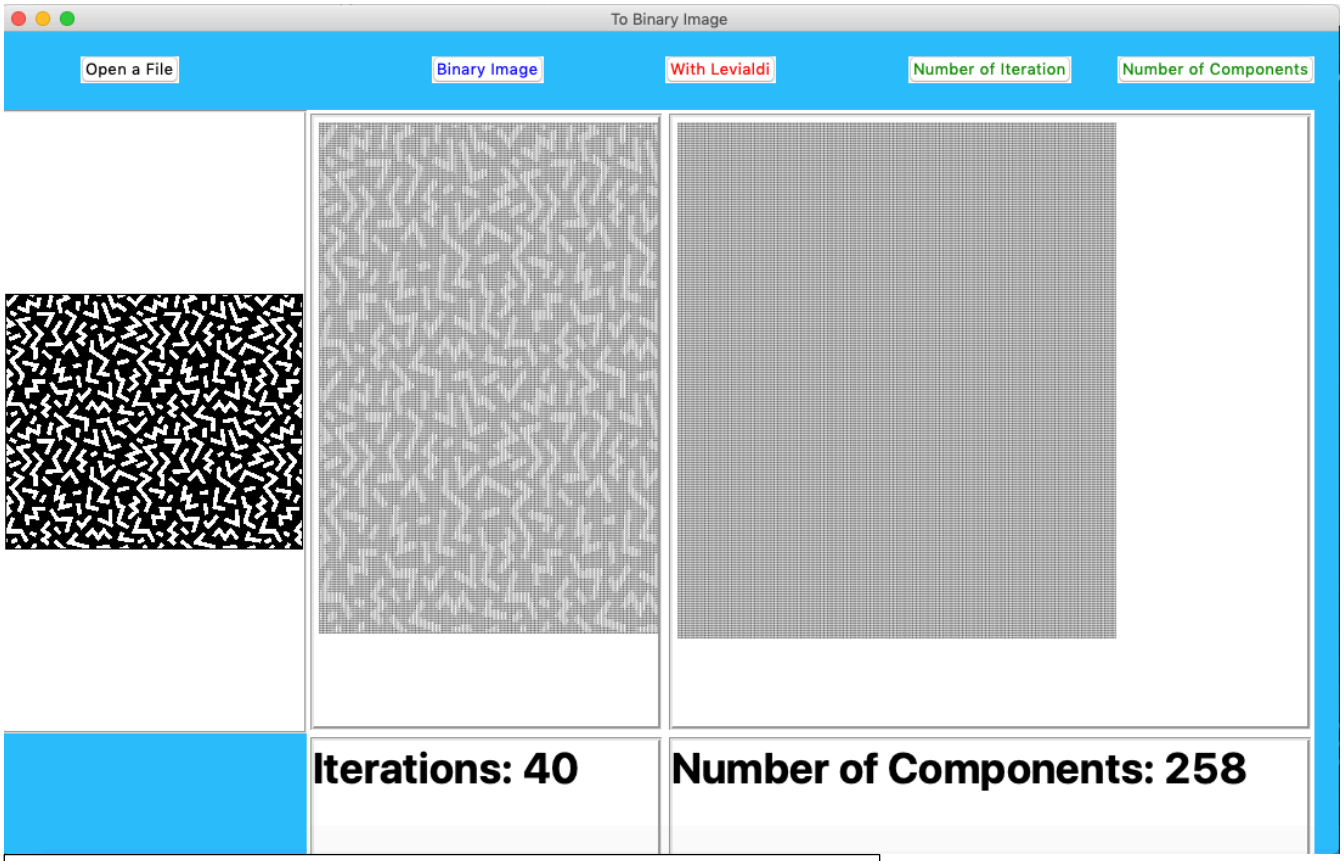
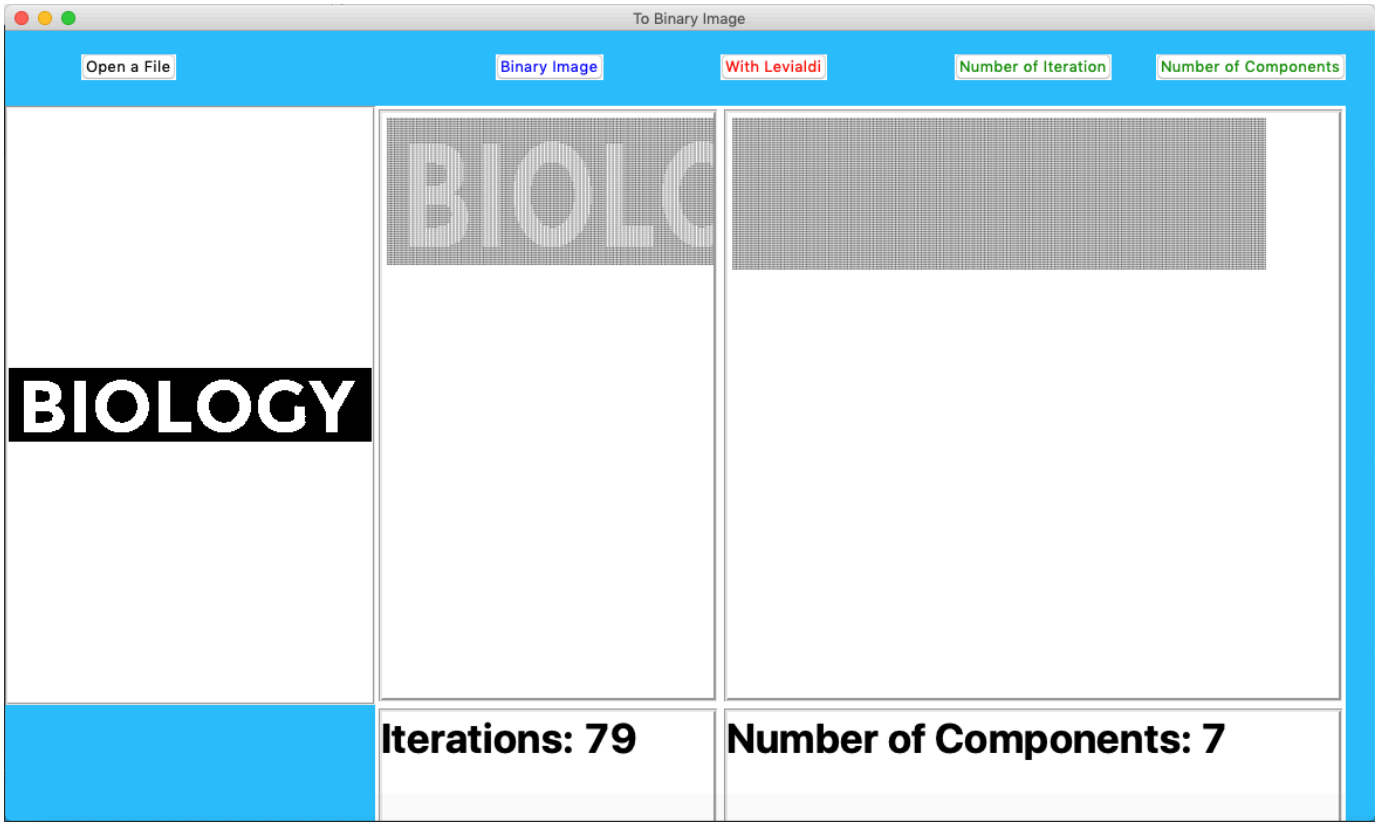
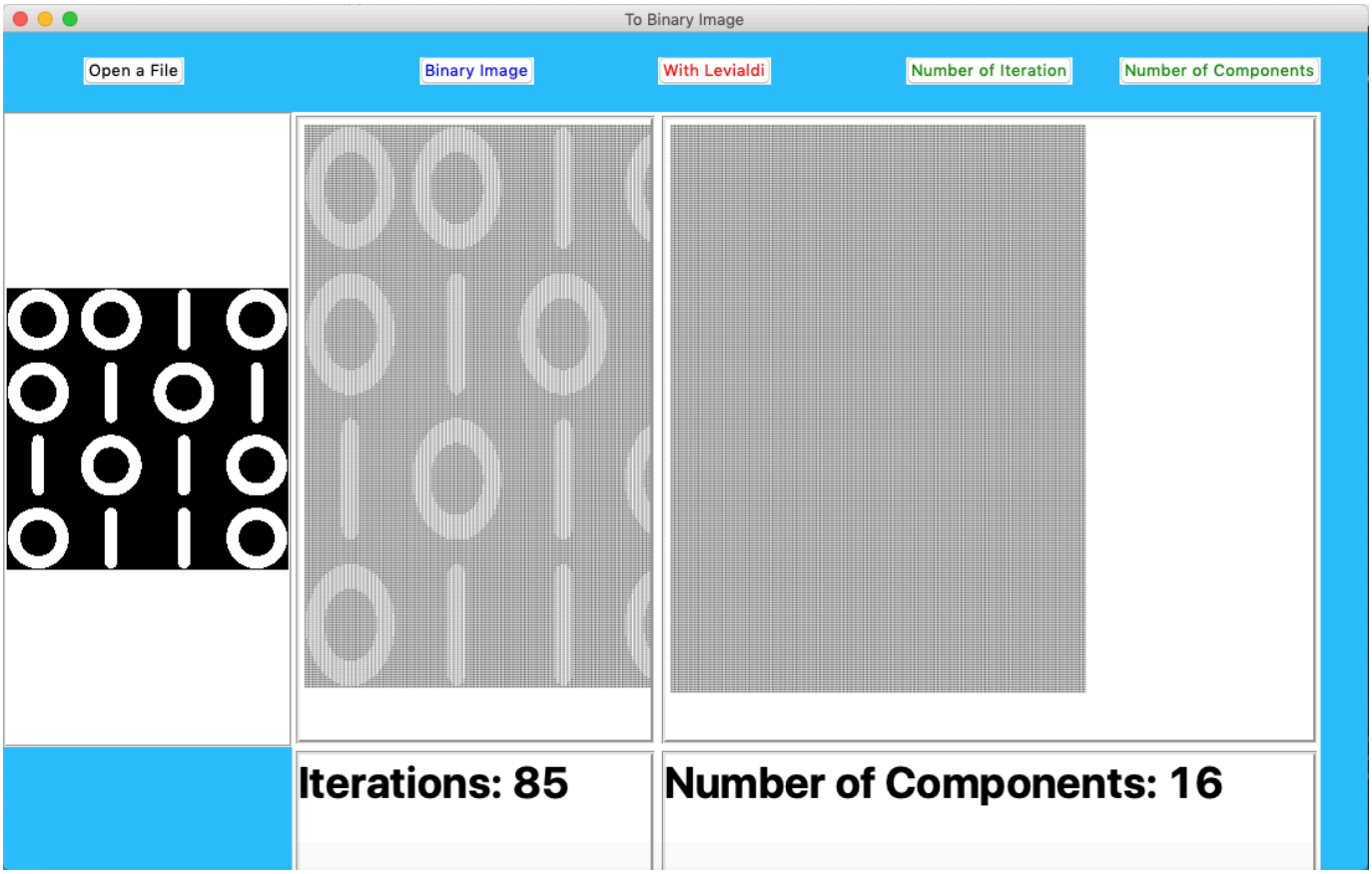


Figure 4. After the levialdi operation on program

Samples





Contribution of this project

In this project I had learned that how can I convert a raw image to binary image and how can I process on that binary image. Also, I learned that how to create graphic user interface which very useful and fun. With this I have learned many things about libraries. I think the biggest profit of this project is now I can see things around to make them digital and process on them. The project gave a new perspective about the relation between image and digital.

References

1. https://www.tutorialspoint.com/python/python_gui_programming.htm
2. https://pythontips.com/2013/07/30/20-python-libraries-you-cant-live-without/
3. https://www.geeksforgeeks.org/digital-image-processing-basics/
4. https://www.learneroo.com/modules/12/nodes/94