CS 254: Algorithms Project Report

Connected Components Labelling in Images

By Vraj Shah(1200131) and Vikas Raunak(1200137)

# Project Objective

To count the number of objects in a given PGM image.

Input: A PGM Image, Number of Counts for Image Dilation

Output: Display the Number of Objects in the Image

Intermediate Stages: Output a binary Image (transformed from the Grayscale Input , Output a dilated image to reduce boundary noise

# Definitions and Terminology

PGM Image: Portable Gray Map Images (Basic form of Images with each pixel having a value between 0 and 255, 0 standing for Black and 255 for White)

Image Thresholding : Process of transforming a grayscale image (pixel intensity ranging from 0 to 255) into a binary image (pixel intensity either 0 or 255 i.e. each pixel is either Black or White).This step is fundamental to the recognition of Objects in an Image as it separates images into two clusters – a foreground (pixels having intensity value 255 i.e White) and a background (pixels having intensity value 0 i.e. Black).We have implemented a standard method for image Thresholding called [Otsu Method](http://www.otsu-method.com) which is based on minimizing the intra class variance between the two clusters.

Objects in an Image: Objects are distinct i.e. not connected entities in the foreground of the binary image. This will be our definition of objects. It highlights two points: 2 distinct objects must be disconnected and all the pixels of a unique object must be connected. This definition will allow us to use a connected components labelling algorithm to mark the objects.

Image Dilation: A standard morphological operation where image is slightly increased in size. We included image dilation to reduce the number of erratic pixels (too disconnected to be called an object) around object boundaries in the foreground as we observed that often in binary images the boundaries are not smoothly connected. This operation will make the boundaries smoothly connected

Connectivity of Pixels: Two pixels in the binary image will be called connected if they are both white (pixel intensity value 255) and are adjacent to each other in the left-right or top-down or diagonal sense i.e. our description involves 8-connectivity (8 pixels around a central pixel have to be tested for connectivity).

# Algorithm Design

Main Tasks:

1. Processing the input Image

This step is essential to convert the image into a form where we can apply the connected components labeling algorithm. The task is to separate foreground from background of an image and reduce the number of erratic pixels (too disconnected to be called an object) around the boundaries of objects in the foreground.

1. Connected Components Labeling Algorithm

We have implemented a 8-connectivity based labeling algorithm and the task is to run Connected Components Labeling algorithm based on depth first search to mark the Objects and then, the number of such labels should correspond to the number of objects based on the definition of object.

Our Entire Program Implements this main idea by building the algorithms for each sub-task:

Step 1: Image Thresholding algorithm ([Otsu Method](http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=4310076))

* Objective is to choose a value t such that pixels with intensity <= t are classified as background and rest is foreground (the objects)
* Find the no. of pixels of each intensity value ( in the range 0 – 255 )
* Find probability of occurrence of each intensity value ( w.r.t area)
* Find t such that variance of each class’s probability is minimized (by checking for each t between 0 to 255)
* t obtained is the required threshold.

Step 2: Image Dilation Algorithm ([Link](http://www.mathworks.in/help/images/morphology-fundamentals-dilation-and-erosion.html))

* Check the pixel value in a 3 by 3 matrix around each point.
* If any of those pixels are set to 255 i.e. White the output is to set the pixel being checked to 255.
* Implemented using simple for loops that check neighborhood (3 by 3) pixel values around each pixel.

Step 3: Depth First Based based 8-connectivity labeling algorithm

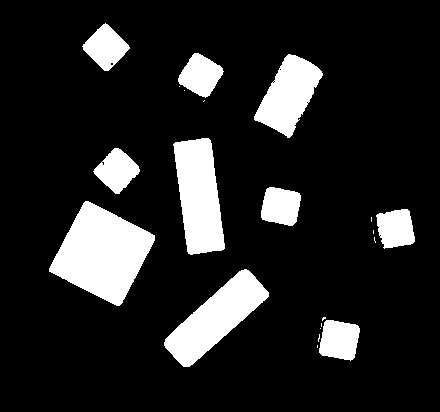
* Scan pixels left to right, top to bottom
* Apply depth first search recursively if pixel value is set ( = 255 ) and pixel is not labeled and label the pixels visited accordingly
* Repeat for all pixels
* Connected components are now labeled and no. of connected components = objects

# Sample Run

This section shows how the algorithm works.

Step 1: Image Thresholding

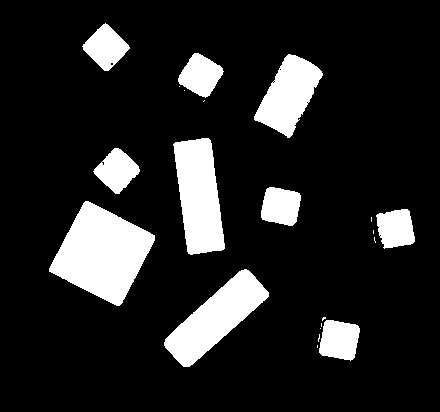
* Input : A PGM Image
* Output : A binary image after image thresholding



Input Output

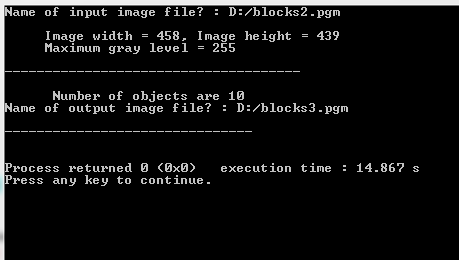
Step 2: Image Dilation

* Input : The binary image generated from the previous step and Number of Counts for dilation
* Output : A dilated image with much smoother boundaries



Step 3: Connected Components Labelling

* Input : The dilated image generated from the previous step
* Output : Prints Image height and width and the number of objects



# Accuracy Analysis And Results

We have run the program taking 5 inputs in 2 categories as mentioned below :

|  |  |  |
| --- | --- | --- |
| Image Type (w.r.t. Objects) | Number of Dilations | Accuracy ( w.r.t. % of Objects Detected ) |
| Sparse Images | 3 | 90-100 % |
| Dense Images | 0 or 1 | Unreliable (As we do not have any reliable counter of actual number of objects in the original image) |

Hence our program works well if the image is sufficiently sparse (Objects are well separated and not too many).

# References

1. [Otsu Method](http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=4310076)