

National University of Computer and Emerging Sciences, Islamabad Campus



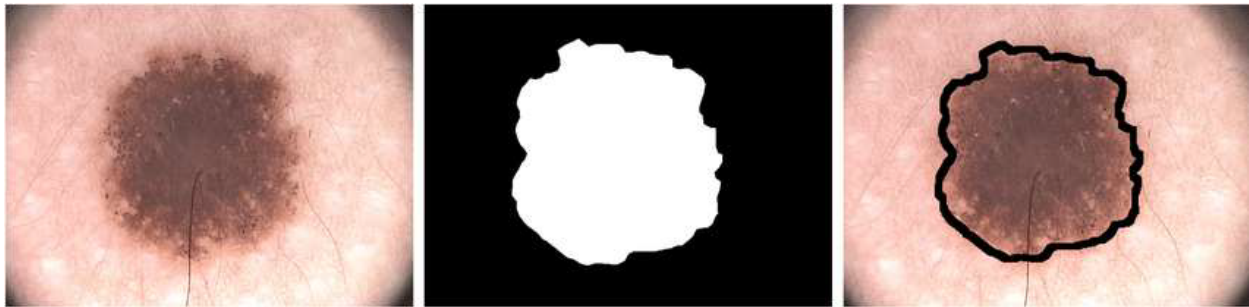
Course:	Data Structures	Course Code:	CS 2001
Program:	BS(Computer Science)	Semester:	Fall 2022
Due Date	1 Oct 2022	Total Marks:	100
Type:	Assignment 1	Page(s):	4

Important Instructions:

1. Submit your source files in a zipped file named as your roll number, i.e., 20I-1111.zip.
2. You are not allowed to copy solutions from other students. We will check your code for plagiarism using plagiarism checkers. If any sort of cheating is found, negative marks will be given to all students involved.
3. Late submission of your solution is not allowed.

Introduction:

Dermoscopy or dermatoscopy refers to the examination of the skin using skin surface microscopy and is also called 'epiluminoscopy' and 'epiluminescent microscopy'. Derm(at)oscopy is mainly used to evaluate pigmented skin lesions. In experienced hands it can make it easier to diagnose melanoma.



With latest advancements, the methods of image processing are used to perform dermoscopy. One way of performing the analysis is through **Connected Components Labeling**.

Connected Components Labeling:

Connected components labeling scans an image and groups its pixels into components based on pixel connectivity, i.e. all pixels in a connected component share similar pixel intensity values and are in some way connected with each other. Once all groups have been determined, each pixel is labeled with a gray level or a color (color labeling) according to the component it was assigned to.

How It Works

Connected component labeling works by scanning an image, pixel-by-pixel (from top to bottom and left to right) in order to identify connected pixel regions, *i.e.* regions of adjacent pixels which share the same set of intensity values V . (For a binary image $V=\{1\}$; however, in a graylevel image V will take on a range of values, for example: $V=\{51, 52, 53, \dots, 77, 78, 79, 80\}$.)

Connected component labeling works on binary or graylevel images and different measures of connectivity are possible. However, for the following we assume binary input images and 8-*connectivity*. The connected components labeling operator scans the image by moving along a row

until it comes to a point p (where p denotes the pixel to be labeled at any stage in the scanning process) for which $V=\{1\}$. When this is true, it examines the four neighbors of p which have already been encountered in the scan (*i.e.* the neighbors (i) to the left of p , (ii) above it, and (iii and iv) the two upper diagonal terms). Based on this information, the labeling of p occurs as follows:

- If all four neighbors are 0, assign a new label to p , else
- if only one neighbor has $V=\{1\}$, assign its label to p , else
- if more than one of the neighbors have $V=\{1\}$, assign one of the labels to p and make a note of the equivalences.

After completing the scan, the equivalent label pairs are sorted into equivalence classes and a unique label is assigned to each class. As a final step, a second scan is made through the image, during which each label is replaced by the label assigned to its equivalence classes. For display, the labels might be different gray levels or colors.

Task 1:

You have to design an algorithm to implement connected component labelling to detect the lesion region on the skin. You are provided with images in the folder. You will use Segmented outputs folder for this task. Where lesion has been narrowed down already. And is represented as 1. ***If multiple independent components are detected in an image. The biggest element will be considered as the lesion. Largest component will be the one that is occupying maximum pixels.***

K-Means Clustering

Another way to detect lesions is using K mean Clustering. K-Means clustering algorithm is an unsupervised algorithm and it is used to segment the interest area from the background. It clusters or partitions the given data into K-clusters or parts based on the K-centroids. Here you will have only two centroids, one centroid for lesion region. And one for non-lesion region.

Steps:

1. Choose the number of clusters K.
2. Select at random K points, the centroids (not necessarily from your dataset).
3. Assign each data point to the closest centroid → that forms K clusters.
4. Compute and place the new centroid of each cluster.
5. Reassign each data point to the new closest centroid. If any reassignment took place, go to step 4, otherwise, the model is ready.

Task 2:

In this task, use the original-colored images and apply K means clustering on them to detect the lesion region. Here centroid values will be pixel values. (At no point in colored image, pixel value goes above 255, choose centroids accordingly. The Lesion region is darker spot indicating lower pixel value. Rest of the skin is lighter tone indicating higher pixel values.). Once the lesion and non-lesion has been segmented out of the image, mark lesion region cluster using 1 and rest of the image as 0 for the next task.

Task 3:

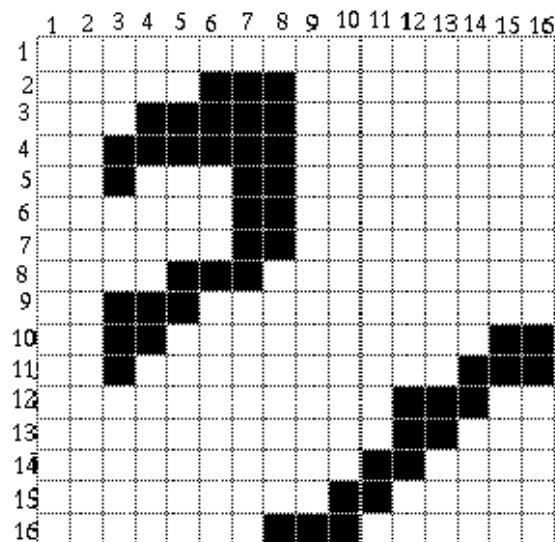
You are also provided with the ground truth images of already correctly identified images. Once you have detected your lesions, you will use the ground truth images and your detected lesion images to test both algorithms on all images and then find performance parameter DICE Coefficient as given in equation below.

Here, True positive (TP) is the number of true positive (pixels that actually belong to lesion according to ground truth and you have also extracted it as lesion) and false positive (FP) is false positives (pixels that don't belong to lesion according to ground truth but detected wrongly as lesions by algorithm you have implemented).

$$\text{Dice Coefficient} = \frac{2 * TP}{FN + (2 * TP) + FP}$$

Run Length Code:

Run Length Code (RLC) is a compact way of encoding images for storage and transmission. The idea is to use the redundancy in pixel value information among neighboring pixels in order to reduce amount of information to be stored. The algorithm proceeds row wise and stores indexes of columns of contagious segments of same pixel value. E.g. in given image below has RLC as follows:



```

16 16
-1
6 8 -1
4 8 -1
3 8 -1
3 3 7 8 -1
7 8 -1
7 8 -1
5 7 -1
3 5 -1
3 4 15 16 -1
3 3 14 16 -1
12 14 -1
12 13 -1
11 12 -1
10 11 -1
8 10 -1

```

Here, first line is width and height of image. Second line onwards, each line corresponds to a row of image. In each line we store the indices of contagious segments of required pixel value. E.g. in line 3, there are black pixels from column 4 to 8. The -1s demarcate the row. If there are more than once contagious blocks, we store all the starting and ending column indexes of that row e.g. row 5. If row does not have any pixels with required pixel value, -1 is alone in that row e.g. row 1.

Task 4:

You will implement RLC for given images. From given folder, you will use Segmented output images for this task. You will use LinkedList to store the image in RLC computed images. The image in example lists pixels containing black i.e. 0 value. You will focus on 1 as in segmented output folder, lesions are indicated in white.

NOTE:

You are not allowed to use any built-in functions provided by image processing libraries except image reading functionality. As images are 2-D arrays, you will process them yourself without any ready-made functions.