

# Reclassification of low Intensity pixels using seed growing

V.Saran Raj<sup>1</sup>, S.Santhosh Rewanth<sup>2</sup>, G.Kaliyugavaratharaj<sup>3</sup>, M.Poonkodi,B.E., M.Tech<sup>4</sup>

<sup>1</sup>Student, Dept. of Computer Science Engineering SRM University, Tamilnadu, India

<sup>2</sup>Student, Dept. of Computer Science Engineering SRM University, Tamilnadu, India

<sup>3</sup>Student, Dept. of Computer Science Engineering SRM University, Tamilnadu, India

<sup>4</sup>Assistant Professor, Dept. of Computer Science Engineering SRM University, Tamilnadu, India

\*\*\*

**Abstract** - Our objective is to reclassify low intensity pixels from seed chosen by random walker and seed growing methods to grow seed. The proposed system, which is an enhanced algorithm used with Local Binary Pattern, is to process the same image so that it can be proved that the enhanced method gives more clarity to the image than the random walker and seed growing.

**Key Words:** Image segmentation, Radom Walker, Seed Growing, Local Binary Pattern

## 1. INTRODUCTION

The image processing can take many forms: filtering, compression, feature extraction and enhancement. Image segmentation is the first step in image processing to distinguish the object and background. Through image segmentation, the image is transformed into various phases, however the process tracks the important features of each phase. Among various segmentation technologies available, choosing an appropriate technology is decided by the particular type of image and the characteristics of the problem being considered. There is no universally accepted method for image segmentations. The techniques are combined with the domain knowledge to solve an image segmentation problem effectively. The image segmentation techniques are classified into two categories: Region based image segmentation and Edge based image segmentation.

Random walk, graph cut and grab cut are few of the image segmentation techniques. Each technique has its merits and demerits. In this paper, we analyze how the local binary pattern technique is used to enhance the intensity of blurred image pixels and compare its performance with other methods.

## 1.1 Blurring

Blurring is present in all imaging processes including vision, photography, and medical imaging methods. It is important to understand the smallest details of the image which depends on the amount of blur produced by the imaging procedure. The amount of blurring can be measured as the dimension of the blurred image of a very small object . The blur also has different shape that depends on the source of blur. For example, some x-ray system produce round blur patterns.

The intensity distribution in the blur area is also an important factor in addition to blur size and shape. One of the distribution patterns is high intensity near the center with a gradual reduction of intensity toward the periphery. The blur limits the amount of details that can be extracted from the objects. This will result in spreads of small objects into surrounding background area.

The visibility of an object depends on the relationship between object size and blur value. If the blur value is comparatively very less than the dimension of the object, the visibility will not be affected much.

The image with much details and distinct boundaries is described as sharp. The presence of blur produces un-sharpness. Un-sharpness is especially notable at the boundaries and edges. It is very important to the technique to separate objects that are close together. To resolve the segmentation of the objects, their separation distance must be increased in proportion to the amount of blur present.

It is important to identify the high intensity region and low intensity region. High intensity region will tend to lose more signal to their surrounding pixels than they gain, and

the region with low intensity pixels will tend to accumulate extra signals from their neighbors.

## 1.2 Challenges in Image Processing

A number of challenges present in recognizing the objects in an image. Some of the challenges are

- This is not feasible to obtain the lightning throughout the day, it may differ during the course of the day. The weather condition, cloudy or sunny, also affects the lightning. Apart from this, indoor and outdoor images for same object can have different lightning conditions. The shadow also affect the clearness of the object. The Image segmentation algorithm should recognize the object whatever the lightning may be.
- Different positions of the object also impacts the ability of identifying the image.
- It is a great challenge to recognize the mirrored image.
- The occultation is defined as the object in an image is not completely visible. The segmentation algorithm must handle that kind of image effectively.

## 2. IMAGE SEGMENTATION METHODOLOGIES

### 2.1 RANDOM WALKER

It is a semi supervised algorithm, in which the user initially identifies a number of seeds and labels them before applying the algorithm. The random walker algorithm releases a random walker from an unlabeled seed and it reaches the boundaries of the labeled seeds. The probability is computed that each pixel arrives at the edge of labeled seeds.

Based on the computed probabilities, the pixel in the unlabeled area is assigned with appropriate label. This process continuous until all the pixels in unlabeled seeds are assigned with proper labels.

#### 2.1.1 Pseudocode

- Read the given image

- Convert the intensity image to double precision.
- Get the size of the image and assign the rows count in X and columns count in Y.
- Set two seeds by assigning values for X1 and Y1 & X2 and Y2.
- Call the random walker function with parameters of image, two seeds, labels and beta.
- If beta value is not supplied assign beta with value 90.
- Get the size of the image and assign the values in variable X,Y and Z.
- If Z value is not 1 or 3 throw the error as 'ERROR: Image must have one (grayscale) or three (color) channels.', and assign ExitFlag as 1.
- If image contains NaN or Inf image values throw the error as 'ERROR: Image contains NaN or Inf values - Do not know how to handle.', and assign ExitFlag as 1.
- If seed locations are not within the image throw the error as 'ERROR: All seed locations must be within image. The location is the index of the seed, as if the image is a matrix.', and assign ExitFlag as 1.
- If given seeds are duplicates throw the error as 'ERROR: Duplicate seeds detected. Include only one entry per seed in the "seeds" and "labels" inputs.', and assign ExitFlag as 1.
- If the given labels are not integers throw the error as 'ERROR: Labels must be integer valued.', and assign ExitFlag as 1.
- If length of the beta value is not one throw the error as 'ERROR: The "beta" argument should contain only one value.', and assign ExitFlag as 1.

- If ExitFlag contains value as 1 return the function with displaying 'Exiting...!'
- Build graph with X and Y values.
- Call the lattice function with X and Y and assign the returned value to the variable edges.
- If image is a color image assign the ImgVals variable with values of RGB or assign the ImgVals with intensity value.
- Call the makeweight function with parameters edges, imgVal and beta and assign the returned value to the variable weights.
- Call the laplacian function with parameters edges and weights and assign the returned value to the variable L.
- Initialize the variable Label\_adjust by assigning the minimum value of labels.
- labels is assigned with the value returned by the expression labels-label\_adjust+1.
- The vector labels\_record is initiated and whose label index value is assigned with 1.
- Call the find function with parameter labels\_record and assign the returned value to the variable labels\_present.
- The variable number\_labels is assigned with the length of the vector labels\_present.
- Initialize the matrix boundary, whose size is number of seeds and length of the vector number\_labels with zeros.
- for k in {1.. number\_labels}
  - do
  - assign the boundary matrix with values as  
boundary(:,k)=(labels(:)==labels\_present(k))
- done
- call the dirichletboundary function with parameters L and seeds(:), boundary and assign the return value to the variable probabilities.
- Find the maximum of the second column of the matrix probability and assign that value to dummy mask.
- Assign the value of the expression labels\_present(mask)+label\_adjust-1 to the variable mask.
- Call the functions reshape with parameters mask and [X Y] and assign the returned value to mask.
- Call the functions reshape with parameters probabilities and [X Y number\_labels] and assign the returned value to probabilities
- Print the original image.
- Print the mask image.
- Print the Image Markup.
- Print the result of probabilities.

## 2.2 SEED GROWING

The first step in seed growing is to select a set of seed points. The initial region is formed at the exact location of these seeds. The regions then grows from these seed points to adjacent points based on the homogeneity characteristics of the seeds. The homogeneity is derived from the properties of the image such as pixel intensity, grey level texture or color.

### 2.2.1 Pseudocode

- Read the given image
- Convert the intensity image to double precision.
- Set one seed by assigning values for X and Y.
- Initialize the variable max\_dist with threshold value.

- Call the functions SeedGrowing with parameters image, seed max\_dist.
- If max\_dist value is 0 assign max\_dist with value 0.2
- If y value of the seed is 0 call the getpts function and assign the returned values to x and y.
- Initialize an image J of size of given image and assign the matrix values as 0.
- Assign the size values of the image to the variable Isizes.
- Calculate the mean of the region and assign that value to reg\_mean variable.
- Initialize the reg\_size value with 1.
- Free the memory to store the values of neighbours.
- Initialize the variable pixdist with 0
- Initialize the neighbour location variable neighb.
- WHILE  
(pixdist<reg\_maxdist&&reg\_size<numel(I))
  - for j in {1.. 4}
    - do
    - Assign the value of the expression  $x + \text{neighb}(j,1); y = y + \text{neighb}(j,2)$  to the variable xn.
    - If neighbour is inside the image
    - Add 1 to the neg\_pos vriable
    - Calculate the value of expression  $[xn \ yn \ I(xn,yn)]; J(xn,yn)=1$  and assign he value to  $\text{neg\_list}(\text{neg\_pos};)$
  - Calculate the value of expression  $\text{abs}(\text{neg\_list}(1:\text{neg\_pos},3)-\text{reg\_mean})$  and assign he value to dist.
  - Find the minimum of dist vector and assign the value to [pixdist, index]
  - Assign value 2 to J(x,y) and increment the variable reg\_size with 1.
  - Calculate the value of expression  $(\text{reg\_mean} * \text{reg\_size} + \text{neg\_list}(\text{index},3))/(\text{reg\_size}+1)$  and assign the value to reg\_mean.
  - Assign x with value of  $\text{neg\_list}(\text{index},1)$  and y with  $\text{neg\_list}(\text{index},2)$ .
  - Assign  $\text{neg\_list}(\text{index};)$  with value of  $\text{neg\_list}(\text{neg\_pos};)$  and decrease the value of neg\_pos with 1.
- END WHILE
- Print the original image.
- Print the masked image.

### 2.3 LOCAL BINARY PATTERN

- The local binary pattern algorithm uses the threshold of the neighboring images to label the pixel. This iber.algorithm labels the pixel by using the threshold of the neighborhood of each pixel and forms the result as a binary number.
- Initially the LBP algorithm forms the label of the image pixels by thresholding the 3 X 3 neighborhood of the each pixel with the center value and assume the result as a binary number. The histogram formed by this algorithm has 28 = 256 different labels and that will be used as a texture descriptor.

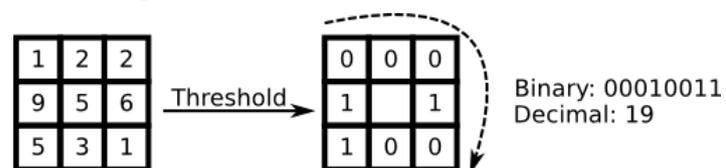


Fig -1: LBP 3 X 3 Matrix

The next extension of LBP uses neighborhoods of different shapes. The circular neighborhood method employs any radius and number of pixels in the neighborhood.

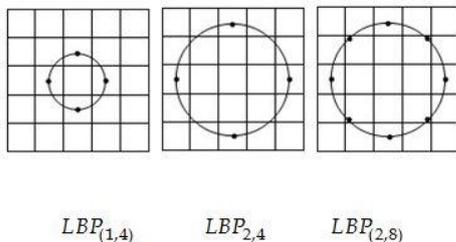


Fig -2: LBP Circular Shape

Another extension to the original operator is the concept of uniform patterns. The advantage of this algorithm is that it reduces the length of feature vector and implement a simple rotation-invariant describer. The objective of this method is that some binary patterns occur more commonly in texture images than others.

If the local binary pattern formed by the algorithm for a pixel contains at most two bitwise transition from 0 to 1 or vice versa, then it is called uniform.

For Example

- 11111111 (0 transition)
- 11111000 ( 1 transition)
- 00001000 (2 transition).

If the local binary pattern contains more than two transition then that is not considered as uniform.

Example:

- 11101010 ( 5 transition)
- 10000101 ( 4 transition)

While computing the LBP labels, the uniform patterns are used in order to have separate label for each uniform pattern and all the non-uniform patterns are labeled with a single label.

### 2.3.1 Enhanced Algorithms

In the enhanced algorithm, the basic Local Binary Pattern algorithm is enhanced with applying threshold weighted vector and median concept to derive the features of the images effectively. The algorithm removes noise or blur if it presents in the image.

The system first analyze the image by splitting the image into Red, Blue and Green images.

The image is converted into a HSI image where H means Hue, S means Saturation and I means Intensity. The HSI image is an important conversion to analyze the image because he colors are represented similarly how the human eyes senses colors. The Hue component represents the color in angle in the range of 0 to 360. Where 0 represents red, 120 represents green and 240 represents blue. The in between numbers represent the combination of these colors. The saturation component describes how much the signal is polluted with white color. It ranges from 0 to 1. The Intensity component ranges from 0 to 1 where 0 means black and 1 means white.

A threshold vector is prepared by using the vectors of geometrical features of the image. The matrix values of the HSI image are manipulated with the threshold vector to produce the output image which contains the feature enhancement. In the manipulation of matrix of image, every pixel is replaced with the median of the vector formed by the algorithm for that particular pixel.

### 3. ARCHITECTURE DIAGRAM

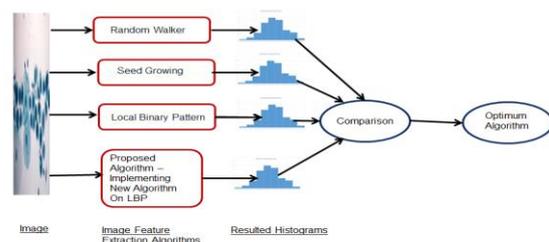


Fig -3: Architecture Diagram

#### 4. FUTURE ENHANCEMENTS

The algorithm can be improved by preparing the threshold vector in different optimal ways so that the feature extracted by the method gives more information to the appropriate classification method such as KNN, Neural Network etc. The classification method analyses the image in deep.

The feature extraction by this algorithm is given to an appropriate classification method to analyse the image in deep.

#### 5. CONCLUSIONS

The algorithm converts the given image into HSI image and the information provided by the HIS image is given to the LPB method. The HSI image is again enhanced by the manipulation of its pixel with threshold vector. This is the way the given image is processed in a sophisticated way and it produces a histogram with more information about the image. It also removes noise and bur if they present. The features of the image are enhanced for the further processing. The histogram produced by the algorithm provides more information about the image.

#### REFERENCES

- [1] Boykov, Y., and Funka-Lea, G.: 'Graph cuts and efficient N-D image segmentation', *Int. J. Comput. Vis.*, 2006, 70, (2), pp. 109–131.
- [2] Grady, L.: 'Random walks for image segmentation', *IEEE Trans. Patt. Anal. Mach. Intell.*, 2006, 28, (11), pp. 1768–1783.
- [3] Rother, C., Kolmogorov, V., and Blake, A.: 'Grabcut: interactive object extraction using iterated graph cuts', *ACM Trans. Graph.*, 2004, 23, (3), pp. 309–314.
- [4] Vanika Singhal, Preeti Singh: 'Local Binary Pattern for Automatic Detection of Acute Lymphoblastic Leukemia', *IEEE* 2014.