**DESIGN**

**4.1 METHODOLOGY**

**Pre-Processing of Lung Cancer**

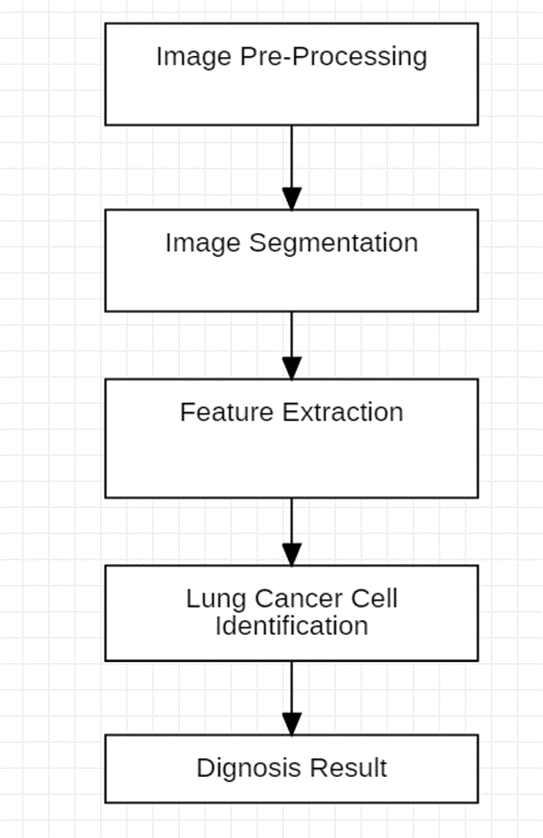
Pre-processing is an improvement of the image data that suppresses unwanted distortions or enhances some image features for further processing. It is needed to minimizing the effects of distortion found in imaging device such as light fluctuation, to remove blueness and in the same

time pre-processing is required to remove unwanted areas from the images and some time, it is used for enhancing the image features like lines, boundaries and textures of image so that we can easily divide the contents of images in two parts, wanted and un-wanted contents of image.

For removing noise from the image, many researchers use different filtering techniques which depends on type of noise. In medical imaging all types of filtering techniques may be used depending on noise present in image.

(a) **Gaussian Noise**: Outside the Normal distribution

values, usually we cannot see in the image.



**Fig 4.1.1 Flowchart of Lung Cancer Detection**

**Canny Edge Detection**

Canny Edge Detection is a popular edge detection algorithm. It is a multi-stage algorithm and we will go through each stage.

1.Noise Reduction

Since edge detection is susceptible to noise in the image, first step is to remove the noise in the image with a 5x5 Gaussian filter. We have already seen this in previous chapters.

2. Finding Intensity Gradient of the Image

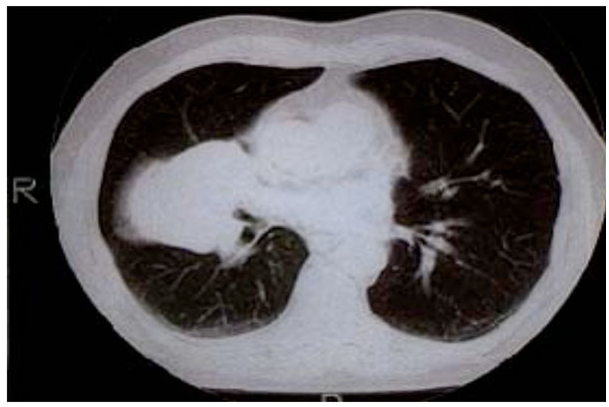
Smoothened image is then filtered with a **Sobel kernel** in both horizontal and vertical direction to get first derivative in horizontal direction (Gx) and vertical direction (Gy). From these two images, we can find edge gradient and direction for each pixel.

3.Non-maximum Suppression

After getting gradient magnitude and direction, a full scan of image is done to remove any unwanted pixels which may not constitute the edge. For this, at every pixel, pixel is checked if it is a local maximum in its neighborhood in the direction of gradient.

In short, the result you get is a binary image with "thin edges".

4. Hysteresis Thresholding

This stage decides which are all edges are really edges and which are not. For this, we need two threshold values, minVal and maxVal. Any edges with intensity gradient more than maxVal are sure to be edges and those below minVal are sure to be non-edges, so discarded. Those who lie between these two thresholds are classified edges or non-edges based on their connectivity. If they are connected to "sure-edge" pixels, they are considered to be part of edges. Otherwise, they are also discarded.

**Fig 4.1.2a Original CT image Fig 4.1.2b Edged Image**

**Image Segmentation**

The process of subdividing the image into two parts one is wanted and other is unwanted parts. Wanted part is also called ROI. In the case of lung cancer our target is to identify the tumor present in the Lung image (Fig. 4.1.3(a-b)). So, tumor is our ROI and other part is unwanted area [11]. It is crucial task for machine to automatically detect the tumor because of variant texture properties of Lung tissue in abnormal region such as reûective of tumors or level of cancer malignancy [12]. For segment the image there are two strategies found in various literature such as edge-based segmentation and region-based segmentation.



**Fig 4.1.3a Original Lung CT Scan image Fig 4.1.3b Segmented CT Scan Image**

**FEATURE EXTRACTION**

In the process of features extraction from the image it is necessary that all properties of an object present in the image should be clear. So, for growing the digital objects we need image enhancement [3]. Image enhancement is divided into two categories. First one is spatial domain

and other is frequency domain. In spatial domain, operations are performed on the pixel values directly so it is easy to understand and analysis. While in frequency domain the methods is used to explain the analysis of signals and mathematical formulas with respect to frequency and function. Image enhancement is achieved when we are able to interpret and threshold the image into two parts, one is known as ROI (Region of Interest) and second is compliment of it.

**GRAY CONVERSION**

Medical image datasets are found in number of different formats. For the simplicity we divide images in three

following types:

(i) Color image

(ii) Gray image

(iii) Binary image

Color image has three channels each channel has 256variant intensity values ranging 0-255. In RGB if RGB have zero value then color is black while all channel values equal to 255 then color is white. Computer can easily distinguish each color shade while human vision is limited as compare to machine vision. In color image there are almost 16777216 different shade. In gray image there are 256 shades in the form of black and white where0 represents black color and 256 represents white color (Fig. 2).



**Fig.4.1.4. Gray Level of Gray Images**

**Thresholding**

A value which is able to segment the ROI is called threshold value. In lung cancer case we convert our image into gray image and then decide the threshold value on the basis of tumor properties. For example, we found that tumor has intensity values 115-255 then we can convert all values which are less than 115 into 0 and all other values between 115-255 is equal to 1 in binary image. In this way all unwanted area become black and wanted area become white region [13].



**Fig 4.1.5a Input Image Fig 4.1.5b Output of Gaussian Filtered Image**



**Fig 4.1.5c Segmented Lungs**

**4.2) Algorithm**

**STEPS INVOLVED –**

Start: Gathering required dataset in form of images.

Step-2: Processing and enhancing the dataset obtained.

Step-3: OpenCV for detecting an object present in an image.

Step-4: Canny edge detector algorithm used as a multi stage algorithm to detect a wide range of edges in images.

Step-5: By the use of image processing in computer vision for detection of tumor that may be present in any individual lung.

Step-6: Making an application for general use by the people for detecting the presence of lung cancer inside the human body.

End: The application will then give necessary information based upon the results obtained.

**Canny edge detection algorithm**

Step-1: Apply Gaussian Filter to smooth the image in order to remove the noise

Step-2: Find the intensity gradients of the image

Step-3: Apply gradient magnitude thresholding or lower bound cut-off suppression to get rid of spurious response to edge detection

Step-4: Apply double threshold to determine potential edges

Step-5: Track edge by hysteresis: Finalize the detection of edges by suppressing all the other edges that are weak and not connected to strong edges.

**Time complexity of canny edge detection algorithm is “O(mn log mn).”**

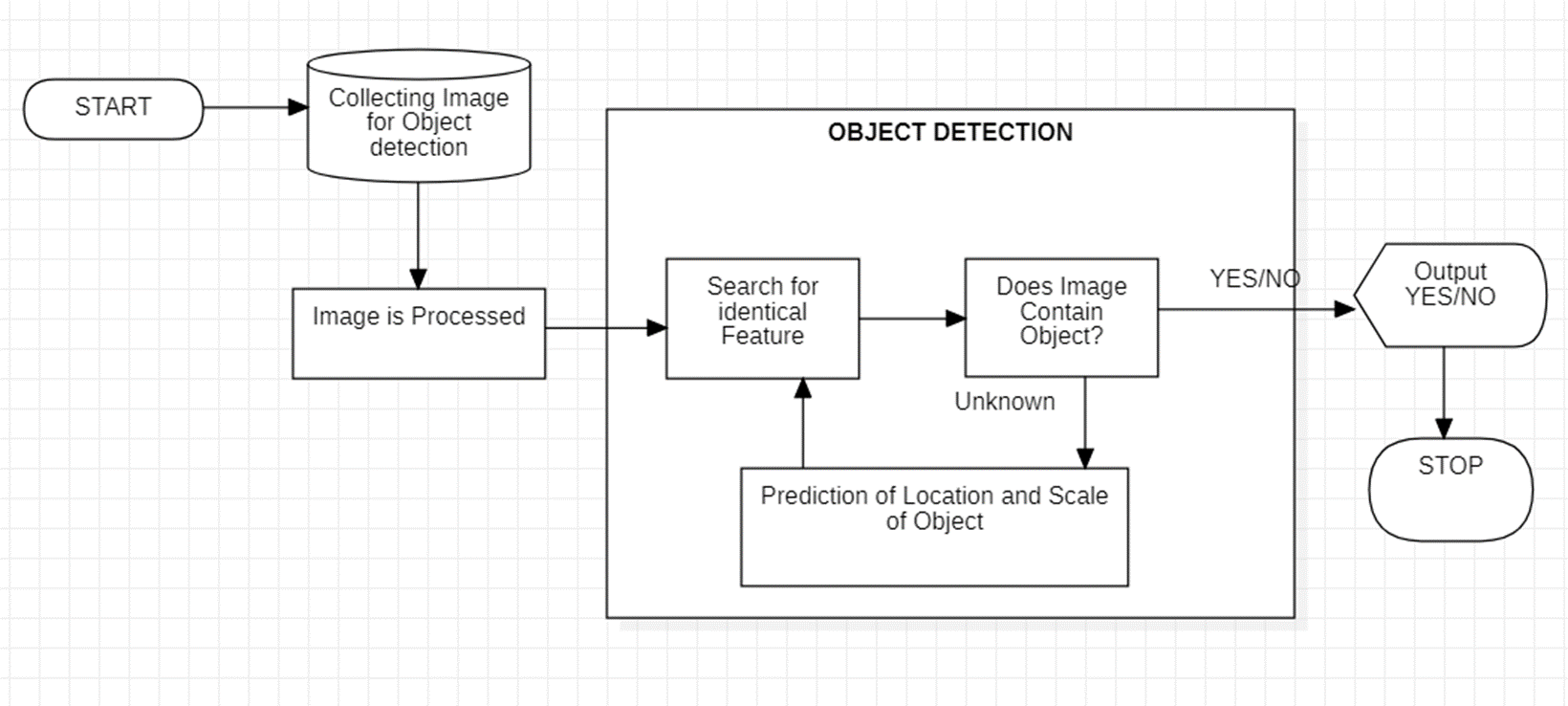
**Pseudo Code:**

1. START
2. Matrix to Store Image Pixels. Its datatype used to store fixed size array matrix
3. Mat src; Mat src\_gray; Mat image\_roi;
4. void thresh\_callback (int, void\*);
5. Main Function Starts Here
6. int main (int argc, char\*\* argv)
7. Load Image Pixels
8. src = imread("../Images/Normal\_01.jpg");
9. Convert Image into Gray
10. cvtColor (src, src\_gray, CV\_BGR2GRAY);
11. Applying Filter, it blurs the image using Normalized box filter
12. blur (src\_gray, src\_gray, Size (3,3));
13. Keeping our Focus on Lung Only
14. Rect rectangle (60,180, src\_gray. cols-150,src\_gray.rows-260);
15. image\_roi = src\_gray(rectangle);
16. named Window (“Segmentation Focus",CV\_WINDOW\_AUTOSIZE );
17. imshow (“Segmentation Focus", image\_roi );
18. Detects Edges
19. Canny (InputArray image, OutputArray edges, double threshold1, double threshold2, int apertureSize=3) , apertureSize – aperture size for the Sobel() operator.
20. Canny (image\_roi, canny\_output, thresh, thresh\*2, 3 );
21. Applying Canny
22. End of Canny
23. imshow("Segmented Lung",canny\_output);
24. Find Contours (Tumors in our case)
25. findContours(InputOutputArray image, OutputArrayOfArrays contours, OutputArray hierarchy, int mode, int method, Point offset=Point())
26. findContours( canny\_output, contours, hierarchy, CV\_RETR\_TREE, CV\_CHAIN\_APPROX\_SIMPLE, Point(0, 0) );
27. Mat drawing = Mat::zeros( canny\_output.size(), CV\_8UC3 );
28. Categorizing Tumors
29. double area;
30. double clength;
31. for (int i=0; i<contours. size();i++)
32. if(hierarchy[i][2]!=-1)
33. Tumor\_Total=Tumor\_Total+1;
34. area=contourArea(contours[i]);
35. if(area>Max\_Tumor\_Area)
36. Max\_Tumor\_Area=area ;
37. Total\_Tumor\_Area=Total\_Tumor\_Area+area;
38. if(area<=35)
39. Tumor\_Lessthan\_25=Tumor\_Lessthan\_25+1;
40. else if(area>35)
41. Tumor\_Greaterthen\_25=Tumor\_Greaterthen\_25+1;
42. clength = arcLength(contours [i],true);
43. Total\_Tumor\_Length=Total\_Tumor\_Length+clength;
44. Scalar color = Scalar (rng.uniform(0, 255), mrng.uniform(0,255), rng.uniform(0,255) );
45. drawContours(drawing, contours, i, color, 2, 8, hierarchy, 0, Point() );
46. Printing On CMD
47. cout "Total Tumor :
48. cout “Total Tumot Area : "
49. cout “Total Tumor Length : “
50. cout "Tumor Area less than 35 : “
51. cout “Tumor Area Greater than 35: “
52. cout <<"Maximum Tumor Area : "
53. Writing into Textfile
54. ofstream myfile;
55. myfile.open("Tumorfile.txt");
56. myfile
57. myfile.close();
58. Displaying Tumors
59. namedWindow( "Tumors", CV\_WINDOW\_AUTOSIZE );
60. imshow(“Tumors", drawing );
61. END

**CANNY EDGE DETECTION Pseudocode**

1. START
2. Input image to be processed.
3. Load the Opencv library for the canny process.
4. Reads the image from the file and saves it to the matrix object.
5. Read the image.
6. Create an empty matrix to save the processed image.
7. Apply the canny function to detect edges.
8. Imgproc.Canny (source, dest, 100, 200, 3);
9. Imgproc.Canny (an object representing the source (input image) for this operation, a Mat
10. object representing the destination (edge) for this operation, a double type variable
11. representing the first threshold for a hysteration procedure, a double type variable
12. representing a second threshold for a hysterization procedure) ;
13. Create and save image processing results.
14. END

**FLOWCHART**

**OBJECT DETECTION WORK FLOW MODEL**