CHAPTER 1

INTRODUCTION

1.1 Introduction

Pancreatic Tumor is one of the leading causes of cancer. Pancreatic cancer is the 10th most commonly diagnosed cancer in men and 9th in women, but the 4th cause of cancer death for both men and women. A major cause of this, there are no effective detection methods available. Also, most of the symptoms of pancreatic tumor are vague and could be contributed to many other abdominal conditions. These symptoms include back pain, weight loss, jaundice, loss of appetite, and diabetes.

The pancreas is a pear-shaped gland located in the abdomen between the stomach and the spine. It is surrounded by several large and important organs and blood vessels. The pancreas has two main functions. It makes pancreatic juices containing enzymes. These enzymes help to break down food so the body can absorb it. The digestive juices flow down a tube called the pancreatic duct, which runs the length of the pancreas and empties into the duodenum. It also makes hormones, including insulin, which control sugar levels in the blood.

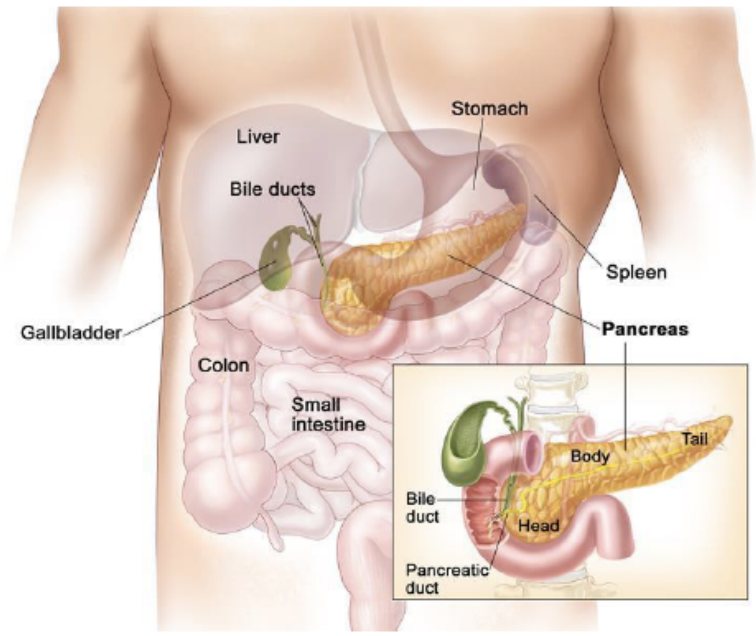

Fig. 1.1(a) Pancreas

Normal healthy cells grow in a carefully controlled way. Pancreatic cancer develops when cells in the pancreas grow out of control, forming a tumor. This can happen in the head, body or tail of the pancreas. Pancreatic cancers are divided into two main groups. Exocrine tumors start in the exocrine cells. These cells make enzymes. About ninety-five out of a hundred pancreatic cancers (95%) are exocrine tumors. The most common type is pancreatic ductal adenocarcinoma – about eighty out of a hundred of all pancreatic cancers (80%). Endocrine tumors start in the cells that produce hormones. Less than five in a hundred (5%) of all pancreatic cancers are endocrine tumors. There are four stages of pancreatic cancers.

- **Stage 1**

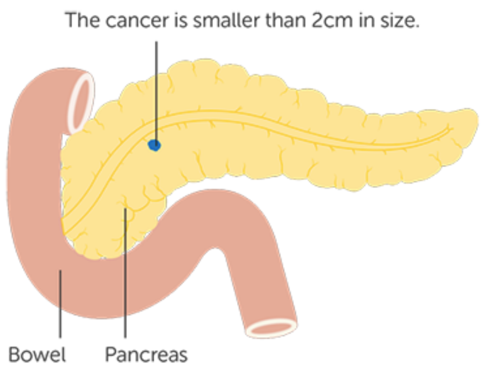

Fig 1.1(b) Stage 1 of Pancreatic Cancer

In Stage 1, the cancer is contained inside the pancreas. This is known as early, localized or resectable pancreatic cancer. It may be possible to operate to remove the cancer. The cancer is completely inside the pancreas and is smaller than 2 cm in size. There is no cancer in the lymph nodes or other areas of the body.

- **Stage 2**

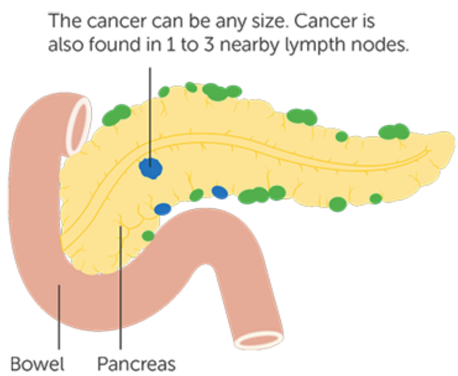

Fig 1.1(c) Stage 2 of Pancreatic Cancer

In stage 2, the cancer has started to grow into the duodenum, bile duct or tissues around the pancreas, or there may be cancer in the lymph nodes near the pancreas. Lymph nodes are small glands found around the body, which are part of the immune system. This may be resectable pancreatic cancer – it may be possible to operate to remove the cancer, depending on how far the cancer has grown.

- **Stage 3**

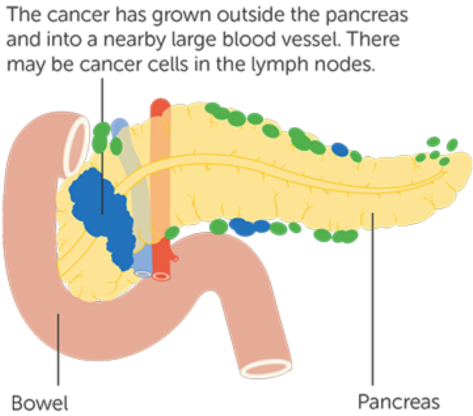

Fig 1.1(d) Stage 3 of Pancreatic Cancer

In stage 3, the cancer has spread into the stomach, spleen, large bowel or into large blood vessels near the pancreas. This is usually locally advanced or unrespectable pancreatic cancer, which means it is not possible to remove the cancer with surgery. However, it may very occasionally be borderline resectable cancer which means it may be possible to remove the cancer, but it depends which blood vessels are affected.

- **Stage 4**

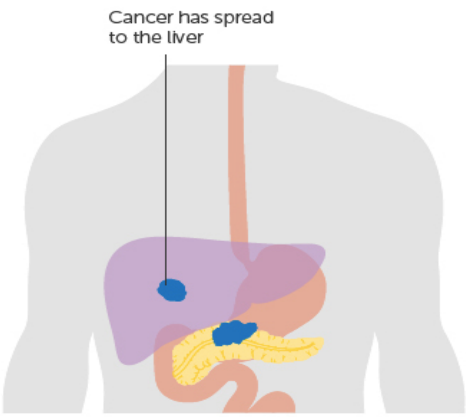

Fig 1.1(e) Stage 4 of Pancreatic Cancer

In stage 4, the cancer has spread to other parts of the body such as the lungs or liver. It's not possible to remove the cancer with surgery, as surgery can't remove all the cancer cells once they have spread to other parts of the body.

1.2 Introduction to Image processing

Image processing was first used in the early 1920s in a paper industry where images were coded for a submarine cable transfer & reconstructed by a telegraph printer at the receiving point. In the mid to late 1920s, there had been improvements in the system. In 1964, image processing was used to improve the images of the moon taken by the Ranger 7 space probe. Such techniques were used in the other space missions as well. In the 1970s, image processing began to be used in the fields of medical science. In 1979, Allan M. Cormack and Godfrey N. Hounsfield jointly received The Nobel Prize for the invention of computer assisted tomography. Nowadays digital image processing is getting more and more attention because of the focus on two principal areas:

- Improvements in the image information for human interpretation.
- Processing of image for autonomous perception.

1.2.1 Importance of Image processing in Cancer Detection

People are looking for techniques that helps in diagnosis of cancer, because cancer detection is critical to increase survival and cost effectiveness of treatment, and as a result decrease mortality rate. Medical images are the most important tools to provide assistance. However, medical images have some limitations for optimal detection of some neoplasias, originating either from the imaging techniques themselves, or from human visual or intellectual capacity. Image processing techniques are allowing detection of abnormalities and treatment monitoring. Because the time is a very important factor in cancer treatment, especially in cancers such as the pancreas, imaging techniques are used to accelerate diagnosis more than with other cancers. In our project, we outline experience in use of image processing technique for pancreatic cancer diagnosis.

A lot of research has been done on the detection of brain cancer, breast cancer and skin cancer, etc. Many algorithms have been successfully implemented for diagnosis of these tumors. There is not much work done on pancreatic cancer detection. It is found from the literature survey that pancreatic cancer detection was done using clinical data, symptoms and patient history by not using image processing. The results acquired were more false-positive and accuracy was less than 80%. In our project, we are using CT scan images to predict the pancreatic cancer. These images are preprocessed using image processing techniques, segmentation, feature extraction and then a basic classifier is used to classify the tumor area in the image.

1.3 Objectives

- To apply GLCM feature extraction on CT images.
- To develop the generic framework for the prediction of pancreatic cancer disease using image processing techniques.
- To implement Support Vector Machine classifier to detect the pancreatic cancer to use in public healthcare environment.

1.4 Problem Statement

- To design and develop a framework for a prediction of Pancreatic cancer by using Image Processing.

CHAPTER 2

LITERATURE SURVEY

1) Real-time contrast-enhanced ultrasound imaging of focal pancreas lesions in fatty pancreas

AUTHORS: G.-J. Liu et al .

Real-time contrast-enhanced ultrasound (CEUS) using second-generation ultrasound contrast agents (UCAs) has been widely used as a noninvasive modality for the detection and characterization of focal pancreas lesions (FLLs) in clinical practice. The enhancement patterns of different kinds of FLLs on CEUS have been well described in normal pancreas parenchyma, and the CEUS diagnostic performance of FLLs in normal pancreas parenchyma has greatly improved compared with conventional gray-scale ultrasound, with 85–96% overall accuracy in differentiating malignant FLLs from benign ones and 81–88.5% overall accuracy in characterizing specific FLL.

2) Contrast-enhanced ultrasound for the characterization of focal pancreas lesions-diagnostic accuracy in clinical practice

AUTHORS: D. Strobel et al

Presented the practice of two experienced centre's concerning the use of contrast enhanced ultrasound (CEUS) in the characterization of focal pancreas lesions (FLL). Material and method: A prospective, bicentric study, between 09.2009- 09.2010 was undertaken and 729 FLL (506-Center A, 223-Center B) were evaluated. A CEUS examination was considered conclusive, if the FLL had a typical enhancement pattern according to EFSUMB Guidelines. Results: From the 729 cases with FLL, 389 (53.4%) were patients without known and 340 (46.6%) with known chronic pancreas disease. CEUS was conclusive for the diagnosis in 597/729 cases (82%) and allowed the positive diagnosis of benign vs. malignant lesion in 662/729 (90.8%) FLL. For each center, independently (Center A vs. Center B) the situation was as follows: conclusive for the diagnosis 390/506 (77.1%) vs 207/223 (92.8%) conclusive for the differentiation benign/malignant 449/506 (88.7%) vs. 213/223 (95.5%) (p=0.0032). Conclusion: In their bicentric study, CEUS was conclusive for diagnosis in 82% of FLL and the benign or malignant character of a lesion was demonstrated in 90.8% of cases. Thus, when faced with an uncharacteristic FLL on standard ultrasound, our local strategy in both centers was to perform CEUS as a first-line investigation thus avoiding other expensive examinations.

3) Computer-aided lesion diagnosis in automated 3-D breast ultrasound using coronal speculation

AUTHORS: T. Tan et al

A computer-aided diagnosis (CAD) system for the classification of lesions as malignant or benign in automated 3-D breast ultrasound (ABUS) images, is presented. Lesions are automatically segmented when a seed point is provided, using dynamic programming in combination with a spiral scanning technique. A novel aspect of ABUS imaging is the presence of speculation patterns in coronal planes perpendicular to the transducer. Speculation patterns are characteristic for malignant lesions. Therefore, we compute speculation features and combine them with features related to echotexture, echogenicity, shape, posterior acoustic behavior and margins. Classification experiments were performed using a support vector machine classifier and evaluation was done with leave-one-patient-out cross-validation. Receiver operator characteristic (ROC) analysis was used to determine performance of the system on a dataset of 201 lesions. We found that speculation was among the most discriminative features. Using all features, the area under the ROC curve was 0.93, which was significantly higher than the performance without speculation features. On a subset of 88 cases, classification performance of CAD was comparable to the average performance of 10 readers.

4) Feature-based image patch approximation for lung tissue classification

AUTHORS: Y. Song, W. Cai, Y. Zhou, and D. Feng

In this paper, they proposed a new classification method for five categories of lung tissues in high-resolution computed tomography (HRCT) images, with feature-based image patch approximation. We design two new feature descriptors for higher feature descriptiveness, namely the rotation-invariant Gabor-local binary patterns (RGLBP) texture descriptor and multi-coordinate histogram of oriented gradients (MCHOG) gradient descriptor. Together with intensity features, each image patch is then labeled based on its feature approximation from reference image patches. And a new patch-adaptive sparse approximation (PASA) method is designed with the following main components: minimum discrepancy criteria for sparse-based classification, patch-specific adaptation for discriminative approximation, and feature-space weighting for distance computation. The patch-wise labeling are then accumulated as probabilistic estimations for region-level classification. The proposed method is evaluated on a publicly available ILD database, showing encouraging performance improvements over the state-of-the-arts.

5) Ultrasound image segmentation and tissue characterization

AUTHORS: J. A. Noble

Ultrasound image segmentation deals with delineating the boundaries of structures, as a step towards semi-automated or fully automated measurement of dimensions or for characterizing tissue regions. Ultrasound tissue characterization (UTC) is driven by knowledge of the physics of ultrasound and its interactions with biological tissue, and has traditionally used signal modelling and analysis to characterize and differentiate between healthy and diseased tissue. Thus, both aim to enhance the capabilities of ultrasound as a quantitative tool in clinical medicine, and the two end goals can be the same, namely to characterize the health of tissue. This article reviews both research topics, and finds that the two fields are becoming more tightly coupled, even though there are key challenges to overcome in each area, influenced by factors such as more open software-based ultrasound system architectures, increased computational power, and advances in imaging transducer design.

2.1 Existing System

- Cancer is one of the major problem today, diagnosing cancer in earlier stage is still challenging for doctors.
- Prediction models proposed so far were developed for clinical data set, accuracy was less than 80% & result is more false positive.
- The textural features were more often used in the existing methods.
- In medical images the statistical parameters and the intensity based features were also needed to get the best features from the images.

2.2 Proposed System

- The main aim of this model is to provide the warning to the patients.
- Using the Image processing techniques, we will extract various features to predict the cancer.
- CT scan and MRI images can be used as input rather than textual data.
- Images processing techniques is used as it provides maximum accuracy.

2.3 Advantages of Proposed System

- The comparison of the classifiers helps in the study of the different types of classifiers.
- The feature selection process employed is effective for selecting the best features from the dataset.
- The performance measures used proves that the proposed feature extraction method is more efficient while applying the features using SVM classifier.

CHAPTER 3

SYSTEM REQUIREMENTS SPECIFICATIONS

3.1 Functional Requirements

It defines the basic functionalities that are carried out at the working for the task. This project as well as about developing the system which includes the functionality like,

1. Pre-processing
2. Segmentation
3. Feature Extraction
4. Classification

Label the Image: For the obtained input image, Preprocessing has to be done along with that region of interest that has to be segmented.
Segmentation: In this project, we use watershed segmentation
Feature Extraction: Here the cluster will be extracted and GLCM features will be obtained
Classification: SVM Classifier will be used for classification.

3.2 Non-Functional Requirements

- **Usability**
  Basic & additionally key here. This framework will basic where individuals like for utilize it. Yet, not then complex which individuals abstain as of utilizing it. These clients might be acquainted along these client interfaces & ought not have issue at relocating to new framework with another environment. The menus, utilitarian components at application ought to be at such a way, to the point that they give clear thought or comprehension for usefulness. A few clients are going to utilize the framework at the same time, then that ease for use for the framework ought not get influenced as for individual clients.
- **Reliability**
  The framework ought to be dependable & solid at giving the functionalities. Once a client has rolled out a few improvements at the framework, the progressions must be made unmistakable by the framework.

The adjustments at framework application made by developer ought to be noticeable both to the task pioneer &at addition the analyzer.

- **Security**

  Aside from imperfection following the framework must give fundamental security & must secure the entire procedure from smashing or harm. As innovation developed at quick rate the security turn into the exceptionally real worry for an association. A huge number for dollars (cash) are put resources into giving security to framework & to applications. Bug following conveys the greatest security accessible at the most noteworthy execution rate conceivable, guaranteeing that unapproved clients can't get to fundamental issue data without authorization. Bug or abandon following framework issues distinctive confirmed clients their mystery passwords then that there are confined functionalities for every one for the clients.

- **Performance**

  The framework & also going to be utilized by numerous representatives all the while. Since these framework is facilitated at solitary web server along the solitary database server out for sight, execution turns into a noteworthy concern. The framework ought not succumb when numerous clients would utilize it all the while. It ought to permit quick openness to the majority for its clients. At the event that two analyzers are all the while attempting for reporting these nearness for the bug, there ought not be any irregularity while then.

- **Scalability**

  These framework ought to sufficiently adaptable for include new functionalities in later stage. There ought to regular channel, that can oblige these new functionalities. Include for new usefulness ought not influence different capacities at the applications.

- **Maintainability**

  The framework observing & support ought to be straightforward & objective at its methodology. There ought not be an excessive number for employments or capacities running at various machines as if it gets hard for screen whether these occupations will be running with no mistakes. Capacities must run straightforward & mistake free.

- **Portability**

  The framework ought to be effortlessly compact to another framework & keep running at various stages. This & additionally required when the web server, which & facilitating framework, gets adhered because for a few issues, which requires the framework to be taken to another framework with no mistake.

- **Reusability**

  The framework ought to be separated onto such modules which could be utilized the part for other framework with no requiring a lot for work & it ought to be client helpful.

3.3 Software & Hardware Requirement Specification

3.3.1 Hardware Specification

- Processor : Intel Dual Core and Above
- RAM : 2 GB and Above
- Hard Disk : 150 GB and Above
- Monitor : VGA Colour.
- Mouse : Optical mouse

3.3.2 Software Specification

- Operating System : Windows 7 and Above
- MATLAB Version : MATLAB R2015a

3.4 Software Description

3.4.1 MATLAB

MATLAB (matrix laboratory) is a multi paradigm numerical computing environment and fourth-generation programming language. Developed by MathWorks, MATLAB allows matrix manipulations, plotting of functions and data, implementation of algorithms, creation of user interfaces, and interfacing with programs written in other languages, including C, C++, Java, Fortran and Python.

Although MATLAB is intended primarily for numerical computing, an optional toolbox uses the MuPAD symbolic engine, allowing access to symbolic computing capabilities. An additional package, Simulink, adds graphical multi-domain simulation and Model-Based Design for dynamic and embedded systems.

In 2004, MATLAB had around one million users across industry and academia. MATLAB users come from various backgrounds of engineering, science, and economics. MATLAB is widely used in academic and research institutions as well as industrial enterprises.

It is a high-performance language for technical computing. It integrates computation, visualization, and programming in an easy-to-use environment where problems and solutions are expressed in familiar mathematical notation. Typical uses include:

- Math and computation
- Algorithm development
- Modeling, simulation, and prototyping
- Data analysis, exploration, and visualization
- Scientific and engineering graphics
- Application development, including Graphical User Interface building

MATLAB is an interactive system whose basic data element is an array that does not require dimensioning. This allows you to solve many technical computing problems, especially those with matrix and vector formulations, in a fraction of the time it would take to write a program in a scalar noninteractive language such as C or Fortran.
The name MATLAB stands for matrix laboratory. MATLAB was originally written to provide easy access to matrix software developed by the LINPACK and EISPACK projects, which together represent the state-of-the-art in software for matrix computation.

MATLAB features a family of application-specific solutions called toolboxes. Very important to most users of MATLAB, toolboxes allow you to learn and apply specialized technology. Toolboxes are comprehensive collections of MATLAB functions (M-files) that extend the MATLAB environment to solve particular classes of problems. Areas in which toolboxes are available include signal processing, control systems, neural networks, fuzzy logic, wavelets, simulation, and many others.

MATLAB has evolved over a period of years with input from many users. In university environments, it is the standard instructional tool for introductory and advanced courses in mathematics, engineering, and science. In industry, MATLAB is the tool of choice for high-productivity research, development, and analysis.

3.4.2 Importance of MATLAB

MATLAB may not be as user friendly as an application like Photoshop, however, being a general purpose programming language it provides many important advantages for forensic image processing.

- It ensures the image processing steps used are completely documented, and hence can be replicated.
- In general, the source code for all image processing functions are accessible for scrutiny and test.
- It allows one to ensure numerical precision is maintained all the way through the enhancement process.
- Image processing algorithms available under MATLAB are likely to be more advanced than those available from other image processing applications.

- MATLAB allows you to test algorithms immediately without recompilation. You can type something at the command line or execute a section in the editor and immediately see the results, greatly facilitating algorithm development.

- The MATLAB Desktop environment, which allows you to work interactively with your data, helps you to keep track of files and variables, and simplifies common programming/debugging tasks

- The ability to read in a wide variety of both common and domain-specific image formats.

- The ability to call external libraries, such as OpenCV

- Clearly written documentation with many examples, as well as online resources such as web seminars ("webinars").

- Bi-annual updates with new algorithms, features, and performance enhancements

- If you are already using MATLAB for other purposes, such as simulation, optimation, statistics, or data analysis, then there is a very quick learning curve for using it in image processing.

- The ability to process both still images and video.
- A large user community with lots of free code and knowledge sharing

The ability to auto-generate C code, using MATLAB Coder, for a large (and growing) subset of image processing and mathematical functions, which you could then use in other environments, such as embedded systems or as a component in other software.

3.4.3 MATLAB System

The MATLAB system consists of five main parts:

3.4.3.1 The MATLAB language

This is a high-level matrix/array language with control flow statements, functions, data structures, input/output, and object-oriented programming features. It allows both "programming in the small" to rapidly create quick and dirty throw-away programs, and "programming in the large" to create complete large and complex application programs.

3.4.3.2 The MATLAB working environment

This is the set of tools and facilities that you work with as the MATLAB user or programmer. It includes facilities for managing the variables in your workspace and importing and exporting data. It also includes tools for developing, managing, debugging, and profiling M-files, MATLAB's applications.

3.4.3.3 Handle Graphics

This is the MATLAB graphics system. It includes high-level commands for two-dimensional and three-dimensional data visualization, image processing, animation, and presentation graphics. It also includes low-level commands that allow you to fully customize the appearance of graphics as well as to build complete Graphical User Interfaces on your MATLAB applications.

3.4.3.4 The MATLAB mathematical function library

This is a vast collection of computational algorithms ranging from elementary functions like sum, sine, cosine, and complex arithmetic, to more sophisticated functions like matrix inverse, matrix eigenvalues, Bessel functions, and fast Fourier transforms.

3.4.3.5 The MATLAB Application Program Interface (API)

This is a library that allows you to write C and Fortran programs that interact with MATLAB. It include facilities for calling routines from MATLAB (dynamic linking), calling MATLAB as a computational engine, and for reading and writing MAT-files.

CHAPTER 4

SYSTEM DESIGN

System design is the process of defining the elements of a system such as the architecture, modules and components, the different interfaces of those components and the data that goes through that system. It is meant to satisfy specific needs and requirements of a business or organization through the engineering of a coherent and well-running system.

4.1 SYSTEM ARCHITECTURE

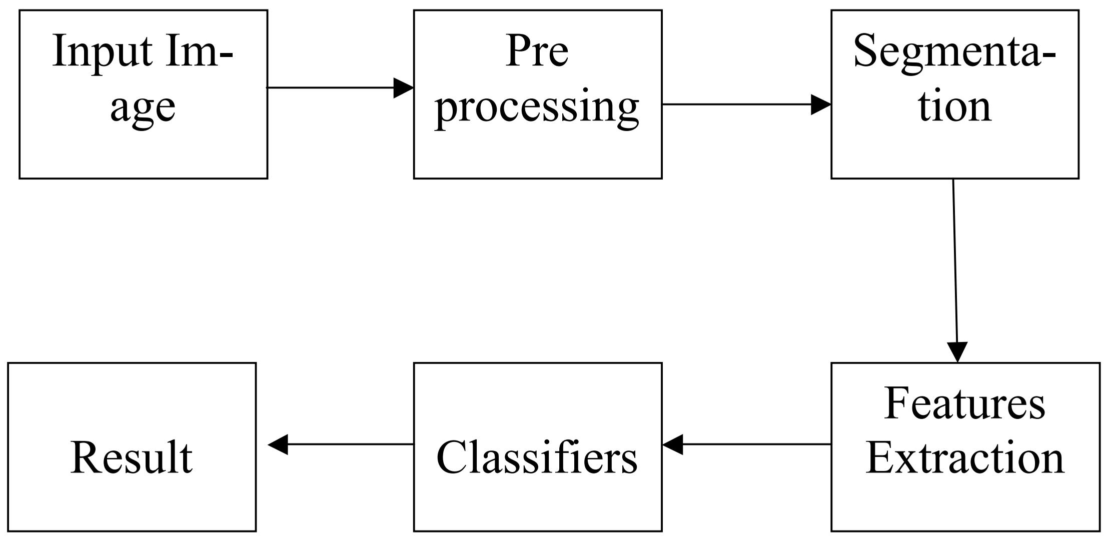

Fig 4.1 System architecture

The System architecture consist of following modules

- **Input image** : Read in the CT image slice of a particular patient.
- **Pre-processing** : Pre-process the CT image slice by using Gaussian filtering.
- **Segmentation** : Segment the pre-processed image using marker controlled Watershed Segmentation
- **Feature extraction :** Extract the features from the binary image by using GLCM algorithm.
- **Classifiers** : With extracted features, identify the stage of cancer using SVM classifier.

4.2 FLOW CHART

A flowchart is a type of diagram that represents a workflow or process. A flowchart can also be defined as a diagrammatic representation of an algorithm, a step-by-step approach to solving a task.

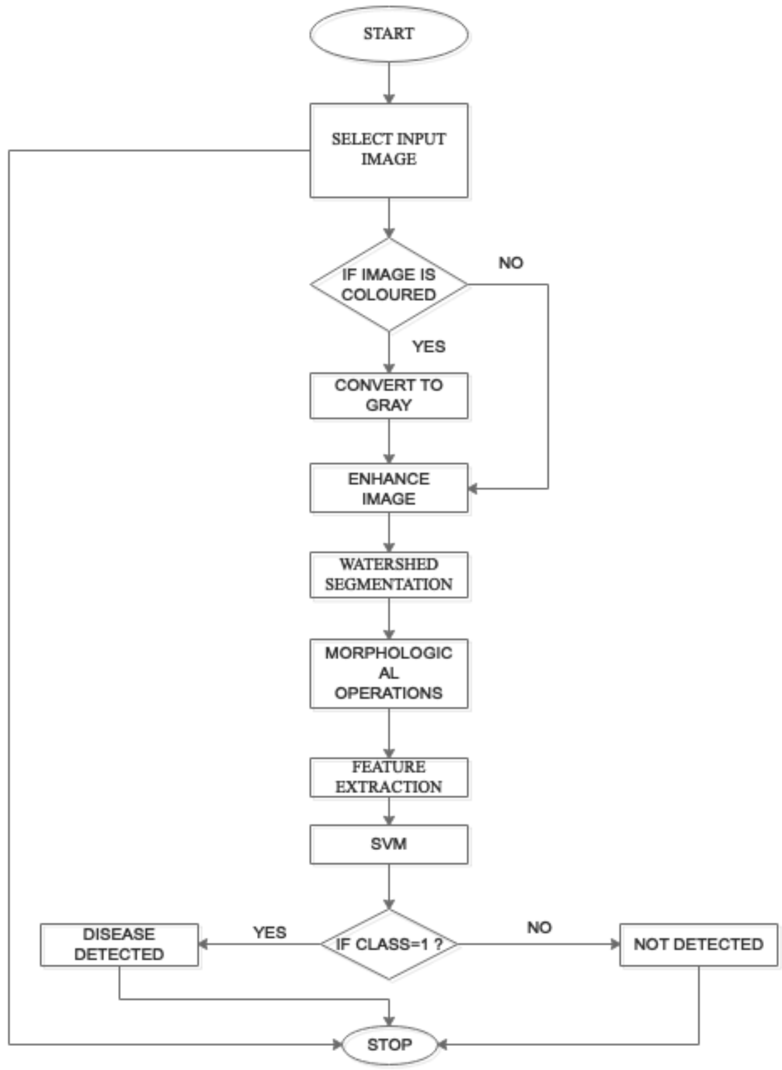

Fig 4.2 Flow Chart

The Fig 4.2 shows the Flow Chart of our system. Algorithm workflow is represented in steps using the flowchart. The steps are as follows

- At first, the input CT scan image is taken as input and no input is taken operation ends.
- It will check if the image is colored or not. If it is colored then it is converted to gray scale or else it will be taken as same input.

- Next, we will enhance the image to remove noises using Gaussian filter and perform marker controlled Watershed Segmentation to segment our region of interest.
- After segmentation, we will perform morphological operations on our ROI to increase density of pixels.
- Feature is Extracted using GLCM feature extraction.
- At last SVM classifier is used to classify based on extracted features. If class=1 then disease is detected or else not detected.

4.3 USE CASE DIAGRAM

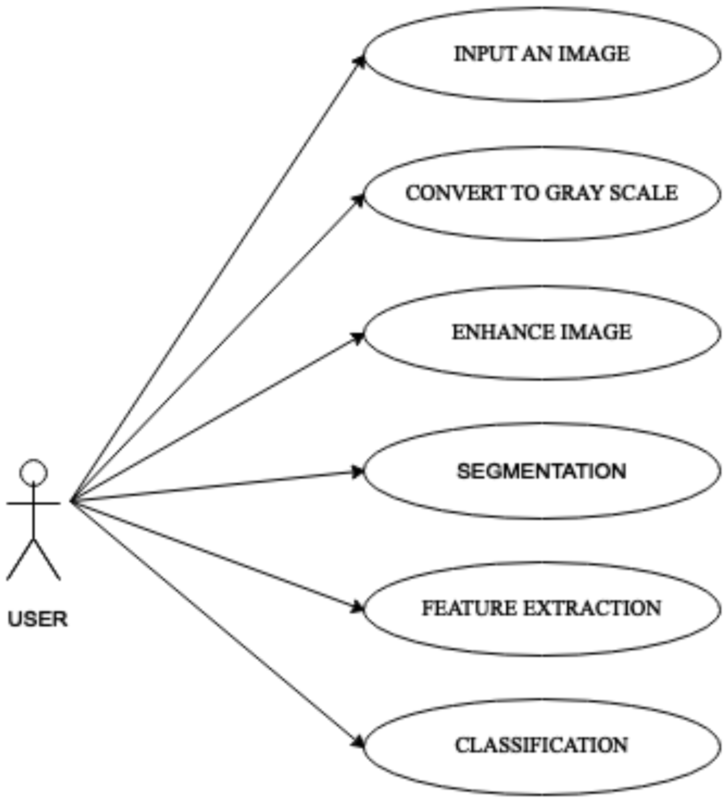

Fig 4.3 Use Case Diagram

A use case diagram in the Unified Modelling Language (UML) is a type of behavioural diagram defined by and created from a Use-case analysis. Its purpose is to present a graphical overview of the functionality provided by a system in terms of actors, their goals (represented as use cases), and any dependencies between those use cases. The main purpose of a use case diagram is to show what system functions are performed for which actor. Roles of the actors in the system can be depicted.

The Primary goals in the design are as follows:

- Provide users a ready-to-use, expressive visual modelling Language so that they can develop and exchange meaningful models.
- Provide extendibility and specialization mechanisms to extend the core concepts.
- Be independent of particular programming languages and development process.
- Provide a formal basis for understanding the modelling language.
- Support higher level development concepts such as collaborations, frameworks, patterns and components.
- Integrate best practices.

CHAPTER 5

IMPLEMENTATION

5.1 Image Pre-processing

Picture improvement procedures in Picture Preparing Tool kit empower you to expand the classification calculations let you distinguish question limits in a picture. These calculations incorporate the Sobel, Prewitt, Roberts, Watchful, and Laplacian of Gaussian strategies. The Watchful strategy can identify genuine feeble edges without being tricked by commotion.

A portion of the improvements Strategies are

- Differentiate Extending
- Commotion separating
- Histogram change

5.1.1 Gaussian Filtering

The Gaussian filter is a type of image-blurring filter that uses a Gaussian function for calculating the transformation to apply to each pixel in the image. The formula of a Gaussian function in one dimension is

$$g(x) = \sqrt{\frac{a}{\pi}} \cdot e^{-a \cdot x^2}$$

In two dimensions, it is the product of two such Gaussian functions, one in each dimension: where x is the distance from the origin in the horizontal axis, y is the distance from the origin in the vertical axis, and σ is the standard deviation of the Gaussian distribution. When applied in two dimensions, this formula produces a surface whose contours are concentric circles with a Gaussian distribution from the center point. Values from this distribution are used to build a convolution matrix which is applied to the original image. This convolution process is illustrated visually in the figure on the right. Each pixel's new value is set to a weighted average of that pixel's neighborhood. The original pixel's value receives the heaviest weight (having the highest Gaussian value) and neighboring pixels receive smaller weights as their distance to the original pixel increases. This results in a blur that preserves boundaries and edges better than other, more uniform blurring filters. Gaussian blurs have nice properties, such as having no sharp edges, and thus do not introduce ringing into the filtered image.

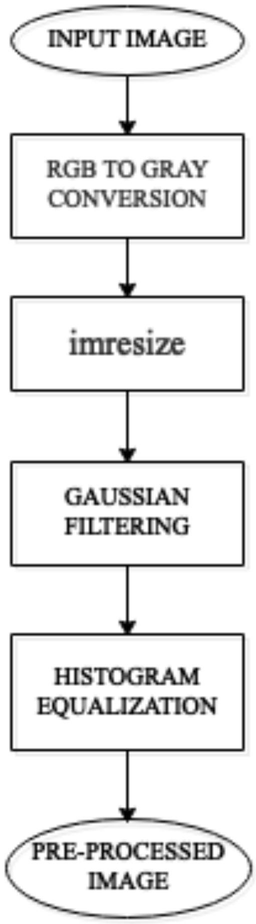

Fig 5.1.1 Flow chart for Image Pre-processing

The above figure 5.1.1 shows the flow chart for Preprocessing of an image. At the start, the original image is converted from RGB To Gray and then it is resized to required size. Later for which Gaussian Filtering and Histogram Equalization Techniques are applied. Finally, we get the preprocessed Output.

5.2 Image Segmentation

Image segmentation is the way toward separating a picture into different parts. This is commonly used to recognize objects or other important data in computerized pictures. The objective of division is to improve or potentially change the portrayal of a picture into something that is more significant and simpler to examine.

5.2.1. Marker Controlled Watershed Segmentation

This count considers the information picture as a topographic surface (where higher pixel regards mean higher height) and reenacts its flooding from specific seed centers or markers. A typical decision for the markers are the nearby minima of the slope of the picture, yet the technique takes a shot at a particular marker, either chose physically by the client or decided naturally by another calculation. The yield of marker control watershed division yield is as appeared in fig 5.2.1

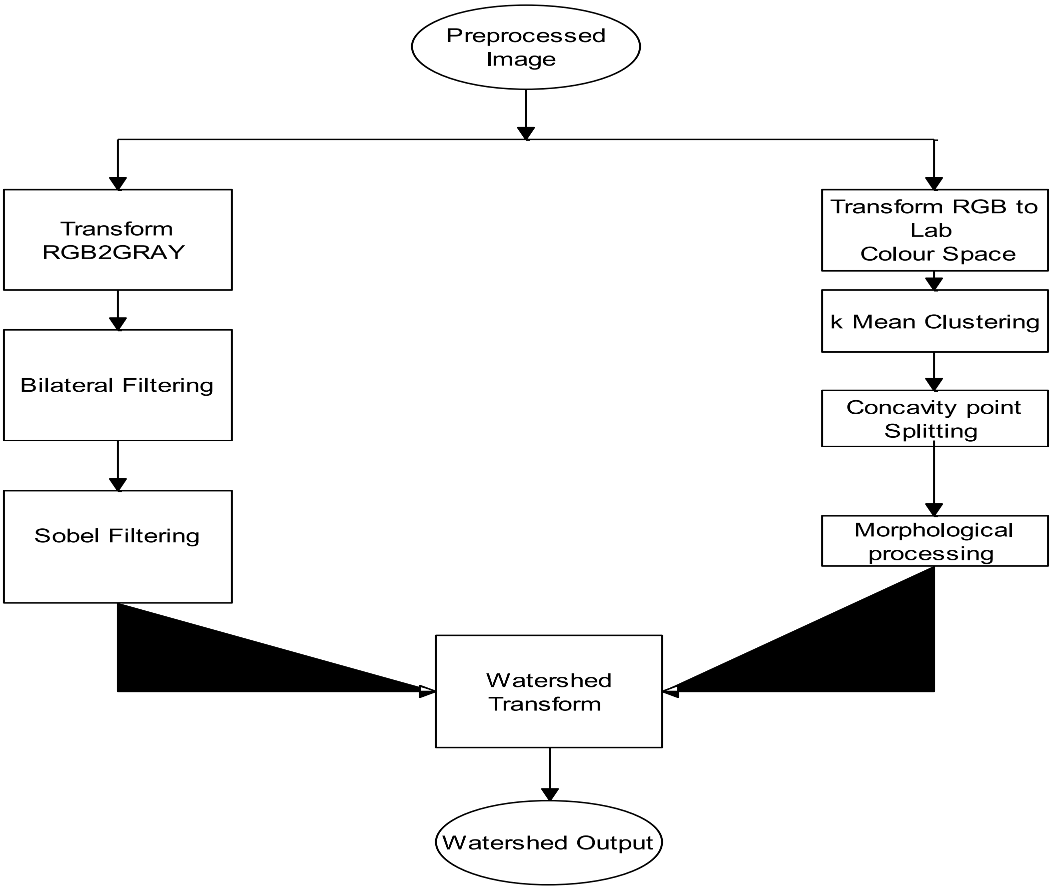

Fig 5.2.1 Flow chart of Watershed transform

Fig. 5.2.1. Shows the detailed Step by Step flow of watershed segmentation. At the beginning the preprocessed image is Given as input to the watershed segmentation. At beginning the flow starts with two parts. In first part the preprocessing of image is Done as shown in figure.

In second Stage the image is transformed into L*a*b Color space from RGB. For which k-mean Clustering is obtained to find Concavity point splitting. Which is ended with Morphological processing. At the end output of both preprocessing and marker extraction is used to find watershed transform of an image.

5.3 Feature Extraction

In Image processing, plan affirmation and in picture get ready, highlight extraction starts from a basic course of action of measured data and amasses induced esteems (highlights) proposed to be valuable and non-dull, empowering the following learning and theory steps, and once in a while provoking better human understandings. Incorporate extraction is related to dimensionality diminishment. Exactly when the data to a count is excessively broad, making it impossible to be in any capacity arranged and it is suspected to be abundance at that point it can be changed into a diminished plan of parts (moreover named a component vector). Choosing a subset of the fundamental segments is called incorporate assurance. The picked parts are required to contain the essential information from the data, so that the desired task can be performed by using this lessened depiction as opposed to the whole beginning data.

5.3.1. Gray Level Co-Occurrence Matrix (GLCM)

One of the most commonly used technique to extract textual data of Images is Gray Level Co-occurrence Matrix (GLCM). The GLCM technique gives sensible surface data of a picture that can be acquired just from two pixels. Dark level co-event frameworks acquainted by Haralick endeavor with portray surface by measurably inspecting how certain dim levels happen in connection to other dim levels. Assume a picture to be broke down is rectangular and has Nx lines and Ny Levels. Accept that the dim level showing up at every pixel is quantized to Ng levels.

Let $L_x=\{1,2,3\ldots N_x\}$ be the flat spatial space, $L_{y=}\{1,2,3\ldots.Ny\}$ be the vertical spatial area, and $G=\{0,1,2,\ldots.N_g\text{-}1\}$ be the arrangement of Ng quantized dim levels. The set Lx×Ly is the arrangement of pixels of the picture requested by their row column assignments. At that point, the picture I can be spoken to as an element of co-event framework that doles out some dark level in Lx×Ly; I: Lx×Ly→G. The dim level moves are ascertained in light of the parameters, dislodging (d) and rakish introduction (θ). By utilizing a separation of one pixel and edges quantized to 450 interims, four lattices of flat, first corner to corner, vertical, and second slanting (0, 45, 90 and 135 degrees) are utilized.

$$P(i,j,d,\theta) = \#$$

$$([k.l],(m,n) \ \in |\left(L_x \times L_y\right) \ \times \left(L_x \times L_y\right))$$

$$k-m=0, |l-n|=d)\ or (k-m=d, l-n=-d)$$

$$or \ \ (k-m=-d, l-n=d) \ \ or(|l-m|=d, l-n=0)$$

$$or(k-m=d, l-n=d)\ \ or(k-m=-d, l-n=-d)$$

$$I(k,l)\ \ = i, I(m,n) = j$$

where # is the quantity of components in the set, (k, l) the directions with dark level i, (m, n) the directions with dim level j.

3	3	3
1	3	3
1	3	2

(a)

0°	1	2	3
1	0	0	2
2	0	0	0
3	0	1	3

(b)

45°	1	2	3
1	0	0	2
2	0	0	0
3	0	0	2

(c)

90°	1	2	3
1	1	0	1
2	0	0	1
3	0	0	3

(d)

135°	1	2	3
1	0	0	0
2	0	0	1
3	0	0	2

(e)

Fig 5.3.1(a) An example of GLCM

Despite the fact that Haralick extricated 24 parameters from co-event network, just seven are generally utilized, for example, vitality, entropy, differentiate, neighborhood homogeneity, relationship, group shade what's more, group noticeable quality as given in Conditions and is put away in highlight database. In expansion, the principal arranges measurable elements (i.e., mean and standard deviation (StdDev) are utilized to portray the qualities of picture as appeared in Conditions individually. The first what's more, second request factual elements are demonstrated as follows:

Vitality measures the quantity of rehashed sets and furthermore measures consistency of the standardized Network.

$$\text{Entrophy} = \sum_{i,j=0}^{N-1} -\ln\,(P_{ij})P_{ij}$$

The differentiation highlight is a distinction snapshot of the P lattice and is a standard estimation of the measure of nearby varieties show in a picture. The higher the estimation of differentiation are, the more honed the auxiliary varieties in the picture.

$$\text{Energy} = \sum_{i,j=0}^{N-1} -\ln P_{ij}{}^{2}$$

The contrast feature is a distinction snapshot of the P lattice and is a standard estimation of the measure of neighborhood varieties show in a picture. The higher the estimation of complexity are, the auxiliary varieties in the picture.

$$\text{Contrast} = \sum_{i,j=0}^{N-1} P_{ij}\,(i-j)^{\,2}$$

It quantifies the sameness of the dispersion of components in the GLCM to the GLCM corner to corner. The opposite of homogeneity results in the announcement of differentiation.

$$\text{Local Homogenity} = \sum_{i,j=0}^{N-1} \frac{P_{ij}}{1+(i-j)^2}$$

where Pij is the pixel esteem in location (i,j) of the surface picture, N is the quantity of dark levels in the picture, μ=∑_ (i,j=0)^(N-1) iP〗 _ij is mean of the surface picture and σ=∑_(i,j=0)^(N-1)〖Pij 〖(i-μ)〗 ^2 〗 is difference of the surface picture.

Correlation is the measure of similarity between two images in likeness. The measures mean (m), which represents the average intensity.

$$\text{Correlation} = \sum_{i,j=0}^{N-1} P_{ij} \frac{(i-\mu)\ (j-\mu)}{\sigma^2}$$

$$\text{Clusre Shade} = \sum_{i,j=0}^{N-1} P_{ij}(i - M_x + j - M_y)^2$$

$$\text{Cluster prominence} = \sum_{i,j=0}^{N-1} P_{ij}(i - M_x + j - M_y)^4$$

Where $M_x = \sum_{i,j=0}^{N-1} iP_{ij}$ and $M_y = \sum_{i,j=0}^{N-1} P_{ij}(i-\mu)^2$

$$\text{mean}(m) = \sum_{i=0}^{L-1} z_i P(Z_i)$$

The measures mean (m), which represents the average intensity

$$\text{Standard Deviation}(\sigma^2) = \sum_{i=0}^{L-1} (x_i - m)^2 P(x_i)$$

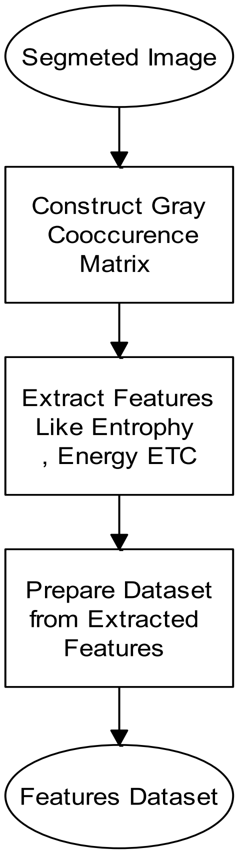

Fig 5.3.1(b) Flowchart for GLCM

5.4 Classification

Classification is gathering of pixels in light of its dark esteem. Grouping is a standout amongst the frequently utilized techniques for data extraction. In Order, generally numerous elements are utilized for an arrangement of pixels i.e., many pictures of a specific question are required. In Remote Detecting region, this system expect that the symbolism of a particular geographic territory is gathered in numerous districts of the electromagnetic range and that the pictures are in great enlistment.

5.4.1. Support Vector Machines (SVM)

Support-vector machine constructs a hyperplane or set of hyperplanes in a high- or infinite-dimensional space, which can be used for classification, regression, or other tasks like outliers detection. Intuitively, a good separation is achieved by the hyperplane that has the largest distance to the nearest training-data point of any class (so-called functional margin), since in general the larger the margin, the lower the generalization error of the classifier.

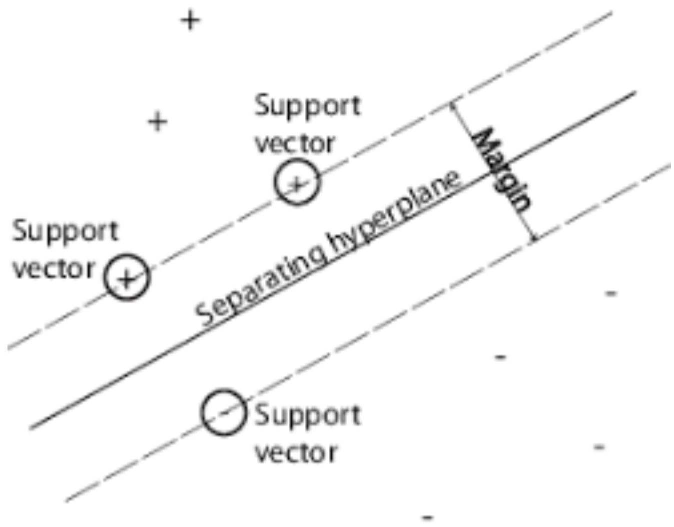

Fig 5.4.1(a) SVM hyperplane

Definitions of SVM and Margin

Find $f(x) = (W^T X + b)$with maximum margin, such that for points closer to the separating hyperplane, $|W^T X_i + b|$

$$|W^T X_i + b| > 1$$

that w is a vector perpendicular to the hyper lane, so we have:

$$f(\mathbf{x}) = f(\mathbf{x}_p + \frac{\mathbf{w}}{\|\mathbf{w}\|} \cdot r) = \mathbf{w}^T \mathbf{x}_p + \mathbf{w}^T \frac{\mathbf{w}}{\|\mathbf{w}\|} r + \mathbf{b} = \|\mathbf{w}\| \cdot r \text{ (since } \mathbf{w}^T \mathbf{x}_p + \mathbf{b} = 0\text{)}$$

Therefore: $r = \dfrac{f(\mathbf{x})}{\|\mathbf{w}\|}$

Now, solve for margin length ρ:

$$\boldsymbol{\rho} = \frac{f(\mathbf{x}_\oplus)}{\|\mathbf{w}\|} - \frac{f(\mathbf{x}_\ominus)}{\|\mathbf{w}\|} = \frac{2}{\|\mathbf{w}\|}$$

Hypothetical Defense:

The accompanying imbalance could be inferred:

$$h \leq \frac{R^2}{\rho^2} + 1$$

margin h speaks to the VC measurement that measures how effective the learning calculation is. It is desirable over utilize the easiest conceivable calculation that gains adequately accurately from the given information.

Accepting a straightly detachable dataset, the errand of learning coefficients w and b of bolster vector machine $f(\mathbf{x}) = (\mathbf{w}^T \mathbf{x}_i + b)$ decreases to taking care of the accompanying obliged streamlining issue: $\frac{1}{2}\|\mathbf{w}\|^2$

subject to imperatives: $y_i(\mathbf{w}^T \mathbf{x}_i + b) \geq 1, \quad \forall i$

This streamlining issue can be comprehended by utilizing the Lagrangian work characterized as:

$$L(\mathbf{w}, b, \boldsymbol{\alpha}) = \frac{1}{2}\mathbf{w}^{\mathrm{T}}\mathbf{w} - \sum_{i=1}^{N} \alpha_i [y_i(\mathbf{w}^{\mathrm{T}}\mathbf{x}_i + b) - 1], \text{ such that } \alpha_i \geq 0, \forall i$$

where $\alpha_1, \alpha_2, \ldots \alpha_N$ are Lagrange multipliers and $\boldsymbol{\alpha} = [\alpha_1, \alpha_2, \ldots \alpha_N]^T$.

The arrangement of the first obliged streamlining issue is controlled by the seat purpose of L(w,b, α) which must be limited concerning w and b and amplified regarding α.

Comments about Lagrange multipliers:

The arrangement of the first obliged improvement issue is dictated by the seat purpose of L(w,b,α) L(w,b, α) which must be limited concerning w and b and expanded as for α.. If $y_i(\mathbf{w}^T \mathbf{x}_i + b) > 1$, the value of α_i that maximizes L(w,b,α) is $\alpha_i = 0$.

If $y_i(\mathbf{w}^T \mathbf{x}_i + b) < 1$, the value of α_i that increases L(w,b,α) is $\alpha_i = +\infty$. However, since w and b are trying to decreases L(w,b,α), they will be altered in such a way to make $y_i(\mathbf{w}^T \mathbf{x}_i + b)$ at least equal to +1.

From this brief discussion, the so-called Kuhn Tucker Conditions follow:

$$\alpha_i\{y_{i(}W^T X_{i+b)-1}\} = 0, \forall i$$

Notation:

Data points x_i with $\alpha_i > 0$ are called the support vectors

Optimality conditions:

The essential conditions for the saddle point of L(w,b,α) are

$$\frac{\partial \mathrm{L}}{\partial \mathrm{W_j}} = 0, \forall \mathrm{j}$$

$$\frac{\partial \mathrm{L}}{\partial \alpha \mathrm{W_i}} = 0, \forall \mathrm{i}$$

or, stated a different way, $\nabla_{\mathrm{w}} L = 0, \quad \nabla_{\alpha} L = 0$

Solving for the essential conditions results in

$$\mathrm{w} = \sum_{\mathrm{i}=1}^{\mathrm{N}} \alpha_{\mathrm{i}}\, \mathrm{y_i x_i}$$

$$\sum_{\mathrm{i}=1}^{\mathrm{N}} \mathrm{a_i}\, \mathrm{y_i} = 0$$

By restoring $\mathrm{w} = \sum_{\mathrm{i}=1}^{\mathrm{N}} \alpha_{\mathrm{i}}\, \mathrm{y_i x_i}$ nto the Lagrangian function and by using $\sum_{\mathrm{i}=1}^{\mathrm{N}} \mathrm{a_i}\, \mathrm{y_i} = 0$

as a new constrain the dual optimization problem can be constructed as

Find α that increases $\sum_{\mathrm{i}} \alpha_{\mathrm{i}} - \frac{1}{2} \sum_{\mathrm{i}} \sum_{\mathrm{j}} \alpha_{\mathrm{i}} \; \alpha_{\mathrm{j}} \mathrm{y_i x_j}$ subject to $\sum_{\mathrm{i}=1}^{\mathrm{N}} \mathrm{a_i} \; \mathrm{y_i} = 0_{, \alpha_{\mathrm{i}} \geq 0}$,

This is a curved quadratic programming issue, so therfis a worldwide least. There are various improvement schedules fit for taking care of this enhancement issue. The streamlining can be illuminated in O(N3) time (cubic with the extent of preparing information) and in direct time in the quantity of characteristics. (Contrast this with neural systems that are prepared in O(N) time)

Bolster Vector Machine: Last Indicator

Given the qualities α1, α2, … αN acquired by arrangement of the double issue, the last SVM indicator can be communicated from as

$$\boldsymbol{f(x) = W^T\ X_i + b = \sum_{i=1}^{N} \alpha_i\ y_i X_i^T\ X + b}$$

where

$$\boldsymbol{b = \frac{1}{|I_{support}|} \quad i.e \quad \sum_{I_{support}} \left(y_i - \sum_{i=1}^{N} \alpha_i\ y_i X_i^T\ X + b \right)}$$

I support is the arrangement of bolster vectors.

Vital remarks:

To acquire the expectation, all information focuses from the preparation information are counseled Since αi α 0 just for the bolster vectors, just bolster vectors are utilized as a part of giving a forecast Take note of that is a scalar.

Bolster Vector Machine: Directly Non-Separable Case Up until this point, we have examined the development of bolster vector machines on straightly divisible preparing information. This is an extremely solid suspicion that is farfetched in most genuine applications.

Solution: Introducing the slack variables ξ_i, i=1, 2, …, N, to relax the constraint

$$y_i(W^T X_i + b) \geq 1 \text{ to } y_i(W^T X_i + b) \geq 1 - \varepsilon_{1,} \varepsilon_1 \geq 0$$

In a perfect world, one would lean toward every slack variable to be zero and this would compare to the directly detachable case. Subsequently, the enhancement issue for development of SVM on directly Nonseparable information is characterized as:

Find w and b that minimize: $\frac{1}{2}\|w^2\| + c\sum_i \in^2$

subject to: $y_{i(w^T x_i + b) \geq 1 - \in_i, \in_i} \geq 0$

Caveat: Cover's Hypothesis just demonstrates the presence of the changed property space that could take care of the nonlinear issue. It doesn't give the rule tote development of the quality change.

SVM answer for grouping

Denote Φ: $\Re^M \rightarrow F$ as a mapping from the first M-dimensional credit space to the exceptionally dimensional characteristic space F.

By taking care of the accompanying double issue

where C > 0 is an appropriately picked parameter. The additional term maintains each slack variable to be as close to zero as could be normal in light of the current situation.

Double issue: As in the directly detachable issue, this improvement issue can be changed over to its doubled issue.

find α that increases subject to

$$\sum_{i} \alpha_i - \frac{1}{2}\sum_{i}\sum_{j} \alpha_i\alpha_j y_i x^t{}_j x_j \qquad \sum_{i-1}^{N} \alpha_i y_j = 0$$

Take note of: The result of presenting parameter C is in obliging the scope of adequate estimations of Lagrange multipliers α_i. The most proper decision for C will rely on upon the particular informational collection accessible.

Issue: Bolster vector machines spoken to with a straight capacity f(x) (i.e. an isolating hyperplane) have extremely constrained depictional power. In that capacity, they couldn't be extremely helpful in useful characterization issues.

SVM answer for order:

Indicate Φ: $\Re^M \rightarrow F$ as a mapping from the first M-dimensional credit space to the very dimensional characteristic space F.

By taking care of the accompanying double issue

find α that maximizes $\quad \sum_{i}\alpha_i - \frac{1}{2}\sum_{i}\sum_{j}\alpha_i\alpha_j y_i y_j \Phi(\mathbf{x}_i)^T \Phi(\mathbf{x}_j)$

subject to $\quad \sum_{i-1}^{N} \alpha_i y_j = 0$

the resulting SVM is of the form

$$f(x) = w^T \emptyset(x_i) + b = \sum_{i=1}^{N} \alpha_i y_i \emptyset(x_i)^T \emptyset(x) + b$$

Viable Problem: Although SVM are effective in managing profoundly dimensional characteristic spaces, the way that the SVM preparing scales directly with the quantity of properties, and considering restricted memory space could to a great extent confine the decision of mapping Φ. Arrangement: Kernel Trick. It permits registering scalar items in the first property space. It takes after from Mercer's Theorem:

There is a class of mappingsΦ that has the following property:

$$\emptyset(\boldsymbol{x_i})^{\boldsymbol{T}} \emptyset(\boldsymbol{y}) = \boldsymbol{K}(\boldsymbol{x}, \boldsymbol{y})$$

where *K* is a corresponding kernel function.

By introducing the kernel trick:

The dual problem: find α that maximizes

$$\sum_{i} \alpha_i - \frac{1}{2}\sum_{i}\sum_{j} \alpha_i\alpha_j y_i x^t{}_j x_j \, K(\; x_i, y_i \,)$$

subject to $\sum_{i-1}^{N} \alpha_i y_j = 0$

The resulting SVM is:

$$f(\mathbf{x}) = \mathbf{w}^T \Phi(\mathbf{x}_i) + b = \sum_{i=1}^{N} \alpha_i y_i K(\mathbf{x}_i, \mathbf{x}) + b$$

$$f(x) = w^T \emptyset(x_i) + b = \sum_{i=1}^{N} \alpha \;\; i\, y_i K(x_i, x) + b$$

The figure 5.4.1(b) show the flowchart for the SVM Classifier. From the Results of GLCM Features Prepare the test and train data for SVM Classifier. Later train the pre-pared trained data and test data into the classifier for classification as shown in flowchart. At the end plot and analyze the result.

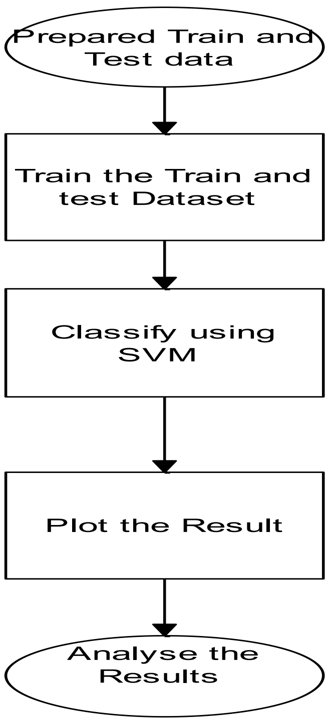

Fig 5.4.1(b) Flowchart for SVM Classifier

5.5 Source Code

5.5.1 Code for User Login GUI

```
function varargout = main1(varargin)
% Begin initialization code %
gui_Singleton = 1;
gui_State = struct('gui_Name',mfilename, ...
                'gui_Singleton',  gui_Singleton, ...
                'gui_OpeningFcn', @main1_OpeningFcn, ...
                'gui_OutputFcn',  @main1_OutputFcn, ...
                'gui_LayoutFcn',  [] , ...
                'gui_Callback',   []);
if nargin && ischar(varargin{1})
   gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
   [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
   gui_mainfcn(gui_State, varargin{:});
end
% End initialization code %
% Choose default command line output for main1%
handles.output = hObject;

% Changing background image of GUI%
guidata(hObject, handles);
ah=axes('unit','normalized','position',[0 0 1 1]);
bg=imread('b3.jpg');
imagesc(bg);
set(ah,'handlevisibility','off','visible','off')
% Outputs from this function are returned to the command line%
function varargout = main1_OutputFcn(hObject, eventdata, handles)
varargout{1} = handles.output;
% Executes on button press in pushbutton1 %
```

```
function pushbutton1_Callback(hObject, eventdata, handles)
global c
global d
 a = get(handles.edit1,'String');
b = get(handles.edit2,'String');
e = get(handles.edit5,'String');
save a a
save b b
save c c
save d d
save e e
e=a;
disp(c)
a=fopen('PatientData.txt','w')
fprintf(a,'\n %s',e)
fprintf(a,'\n %s',a)
fprintf(a,'\n %s',b)
fprintf(a,'\n %s',c)
fprintf(a,'\n  %s',d)
fclose(a)
maingui
% Executes on selection change in popupmenu1%
function popupmenu1_Callback(hObject, eventdata, handles)
global c
val1=get(handles.popupmenu1,'value');
 if val1==1
    c='Male';
else
    c='Female';
end
% Executes on selection change in popupmenu2 %
function popupmenu2_Callback(hObject, eventdata, handles)
global d
val1=get(handles.popupmenu2,'value');
 if val1==1
```

```
    d='Weight Loss';
elseif val1==2
    d='Diabetes';
    elseif val1==3
    d='Diarrhea';
    elseif val1==4
    d='Abdominal Pain';
    elseif val1==5
    d='Acidity';
else
    d='Others';
end
```

5.5.2 Code for Training the data

```
for iiii = 1:27
     I = imread(['Dataset\','IMG (',num2str(iiii),')','.jpg']);
% -- Image Noise Filtering -- %
    flevel = 0.5;
    fmat = 3;
    Im = fspecial('gaussian',[fmat fmat],flevel);
    Filt = imfilter(I,Im);
% -- SEgmentation -- %
if size(Filt,3) == 3
    I = rgb2gray(Filt);
else
    I = Filt;
end
    se = strel('disk', 20);
    Io = imopen(I, se);
    Ie = imerode(I, se);
    Iobr = imreconstruct(Ie, I);
    Ioc = imclose(Io, se);
     Iobrd = imdilate(Iobr, se);
    Iobrcbr = imreconstruct(imcomplement(Iobrd), imcomplement(Iobr));
    Iobrcbr = imcomplement(Iobrcbr);
```

```
    fgm = imregionalmax(Iobrcbr);
     I2 = I;
    I2(fgm) = 255;
     se2 = strel(ones(5,5));
    fgm2 = imclose(fgm, se2);
    fgm3 = imerode(fgm2, se2);
    fgm4 = bwareaopen(fgm3, 20);
    I3 = I;
    I3(fgm4) = 200;
    for ii = 1:size(fgm4,1)
       for jj = 1:size(fgm4,2)
          if fgm4(ii,jj) == 1
             outp(ii,jj) = I(ii,jj);
          else
             outp(ii,jj) = 0;
          end
       end
    end
    bw1 = double(fgm);
    bw1 = im2bw(bw1);
    figure(11),
    imshow(I), title('Objects in cluster 1');
     hold on;
    boundaries = bwboundaries(bw1);
    numberOfBoundaries = size(boundaries);
    for k = 1 : numberOfBoundaries
       thisBoundary= boundaries{k};
       figure(11),plot(thisBoundary(:,2), thisBoundary(:,1), 'r', 'LineWidth',1.5);
     end
    hold off;
% -- GLCM Feature -- %
GLCM2 = graycomatrix((outp),'Offset',[2 0;0 2]);
stats = glcm(GLCM2,0);
v1 = stats.autoc(1);
v2 = stats.contr(1);
```

```
v3 = stats.corrm(1);
v4 = stats.corrp(1);
v5 = stats.cprom(1);
v6 = stats.cshad(1);
v7 = stats.dissi(1);
v8 = stats.energ(1);
v9 = stats.entro(1);
v10 = stats.homom(1);
v11 = stats.homop(1);
v12 = stats.maxpr(1);
v13 = stats.idmnc(1);
Glcm_fea = [v1 v2 v3 v4 v5 v6 v7 v8 v9 v10 v11 v12 v13];
Trainfea(iiii,:) = Glcm_fea;
end
save Trainfea Trainfea
```

5.5.3 Code for Main GUI

```
function varargout = maingui(varargin)
% Begin initialization code %
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                   'gui_Singleton',  gui_Singleton, ...
                   'gui_OpeningFcn', @maingui_OpeningFcn, ...
                   'gui_OutputFcn',  @maingui_OutputFcn, ...
                   'gui_LayoutFcn',  [] , ...
                   'gui_Callback',   []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end
if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code %
```

```
% Outputs from this function are returned to the command line %
function varargout = maingui_OutputFcn(hObject, eventdata, handles)
varargout{1} = handles.output;
% Executes on button press in pushbutton1 %
function pushbutton1_Callback(hObject, eventdata, handles)
load a
load b
load c
load d
set(handles.edit16,'string',a)
set(handles.edit17,'string',b)
set(handles.edit18,'string',c)
set(handles.edit19,'string',d)
global I
[file,p] = uigetfile('*.*');
if file == 0
   warndlg('You Have Cancelled...');
else
   I = imread([p file]);
       imshow(I,'parent',handles.axes1);
   title('Input Image','parent',handles.axes1);
end
% Executes Gaussian filtering %
function pushbutton2_Callback(hObject, eventdata, handles)
global I
global Filt
   flevel = 0.5;
   fmat = 3;
   Im = fspecial('gaussian',[fmat fmat],flevel);
   Filt = imfilter(I,Im);
   imshow(Filt,'parent',handles.axes2);
   title('Filtered Image','parent',handles.axes2);
% Executes Watershed Segmentation %
function pushbutton3_Callback(hObject, eventdata, handles)
global Filt
```

```
global I
if size(Filt,3) == 3
   I = rgb2gray(Filt);
else
   I = Filt;
end
    hy = fspecial('sobel');
   hx = hy';
   Iy = imfilter(double(I), hy, 'replicate');
   Ix = imfilter(double(I), hx, 'replicate');
   gradmag = sqrt(Ix.^2 + Iy.^2);
       L = watershed(gradmag);
   Lrgb = label2rgb(L);
   figure, imshow(I),colormap('jet')
   imshow(Lrgb,'parent',handles.axes2)
   title('Watershed transform','parent',handles.axes2);
% --- Executes Morphological operations %
function pushbutton4_Callback(hObject, eventdata, handles)
global I
global outp
se = strel('disk', 20);
   Io = imopen(I, se);
    imshow(Io,'parent',handles.axes3);
    pause(2)
   Ie = imerode(I, se);
   Iobr = imreconstruct(Ie, I);
   imshow(Iobr,'parent',handles.axes3)
   title('Opening-by-reconstruction (Iobr)','parent',handles.axes3);
   pause(2)
  Ioc = imclose(Io, se);
       imshow(Ioc,'parent',handles.axes3);
       title('Opening-closing (Ioc)','parent',handles.axes3);
   Iobrd = imdilate(Iobr, se);
   Iobrcbr = imreconstruct(imcomplement(Iobrd), imcomplement(Iobr));
   Iobrcbr = imcomplement(Iobrcbr);
```

```
    imshow(Iobrcbr,'parent',handles.axes3);
    title('Opening-closing by reconstruction (Iobrcbr)','parent',handles.axes3);
    pause(2)
 fgm = imregionalmax(Iobrcbr);
  imshow(fgm,'parent',handles.axes3);
  title('Regional maxima','parent',handles.axes3);
  I2 = I;
I2(fgm) = 255;
imshow(I2,'parent',handles.axes3)
title('Regional maxima','parent',handles.axes3);
 se2 = strel(ones(5,5));
fgm2 = imclose(fgm, se2);
fgm3 = imerode(fgm2, se2);
fgm4 = bwareaopen(fgm3, 20);
I3 = I;
I3(fgm4) = 200;
    imshow(I3,'parent',handles.axes3);
title('Modified regional maxima ','parent',handles.axes3);
for ii = 1:size(fgm4,1)
    for jj = 1:size(fgm4,2)
        if fgm4(ii,jj) == 1
            outp(ii,jj) = I(ii,jj);
        else
            outp(ii,jj) = 0;
        end
    end
end
  imshow(outp,[],'parent',handles.axes3);
title('Modified regional maxima','parent',handles.axes3);
bw1 = double(fgm);
bw1 = im2bw(bw1);
    %imshow(I), title('Objects in cluster 1');
        boundaries = bwboundaries(bw1);
numberOfBoundaries = size(boundaries);
for k = 1 : numberOfBoundaries
```

```
        thisBoundary= boundaries{k};
        end
% --- Executes GLCM feature extraction %
function pushbutton5_Callback(hObject, eventdata, handles)
global outp
global Testfea
GLCM2 = graycomatrix((outp),'Offset',[2 0;0 2]);
 stats = glcm(GLCM2,0);
 v1 = stats.autoc(1);
v2 = stats.contr(1);
v3 = stats.corrm(1);
v4 = stats.corrp(1);
v5 = stats.cprom(1);
v6 = stats.cshad(1);
v7 = stats.dissi(1);
v8 = stats.energ(1);
v9 = stats.entro(1);
v10 = stats.homom(1);
v11 = stats.homop(1);
v12 = stats.maxpr(1);
v13 = stats.idmnc(1);
Glcm_fea = [v1 v2 v3 v4 v5 v6 v7 v8 v9 v10 v11 v12 v13];
 A = Glcm_fea;
set(handles.edit1,'string',v1);
set(handles.edit2,'string',v2);
set(handles.edit3,'string',v3);
set(handles.edit4,'string',v4);
set(handles.edit5,'string',v5);
set(handles.edit6,'string',v6);
set(handles.edit7,'string',v7);
set(handles.edit8,'string',v8);
set(handles.edit9,'string',v9);
set(handles.edit10,'string',v10);
set(handles.edit11,'string',v11);
set(handles.edit12,'string',v12);
```

```
set(handles.edit13,'string',v13);
Testfea  = Glcm_fea;

% --- Executes SVM classification %
function pushbutton7_Callback(hObject, eventdata, handles)
global outp
global Testfea
load Trainfea
kk=Trainfea;
BW=outp;
A=Testfea
Target(1:9)=1;
Target(10:27)=2;
Svm_Class = multisvm(Trainfea,Target,Testfea);
if Svm_Class == 1
   msgbox('Cancer Detected');
   nbm='Cancer Detected';
   a=fopen('Result.txt','w')
fprintf(a,'\n \n \n \n \n \n %s',nbm)
fclose(a)
   [r,c]=size(BW);
bc=1;
wc=1;
for i=1:r
   for j=1:c
      if BW(i,j)==0
         bc=bc+1;
      else
         wc=wc+1;
      end
   end
end
Area=wc*0.264;
k1=Area
if k1>10000 && k1<12000
   ff='25% Affected 1st  Stage'
```

```
    set(handles.edit15,'string',ff)
elseif k1>12001 && k1<13000
      ff='50% Affected 2nd  Stage'
    set(handles.edit15,'string',ff)
    elseif k1>13000 && k1<14000
      ff='75% Affected 3rd Stage'
    set(handles.edit15,'string',ff)
else
      ff='More than 75% Affected Final Stage'
    set(handles.edit15,'string',ff)
end
    else
    msgbox('Cancer not-affected');
    nbm='Cancer not-affected';
    a=fopen('Result.txt','w')
fprintf(a,'\n \n \n \n \n \n %s',nbm)
fclose(a)
end
load e
mno=e;
matmal1(mno,'Result','Patient Data')
matmal(mno,'Result','Tested Report')
msgbox('Email Sent to Patient')
```

CHAPTER 6

TESTING

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub-assemblies, assemblies and/or a finished product It is the process of exercising software with the intent of ensuring that the software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of test. Each test type addresses a specific testing requirement.

6.1 Types of Tests

6.1.1 Unit Testing

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application .it is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

6.1.2 Integration testing

Integration tests are designed to test integrated software components to determine if they actually run as one program. Testing is event driven and is more concerned with the basic outcome of screens or fields. Integration tests demonstrate that although the components were individually satisfaction, as shown by successfully unit testing, the combination of components is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components.

6.1.3 Functional test

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals.

Functional testing is centered on the following items:

Valid Input : identified classes of valid input must be accepted.

Invalid Input : identified classes of invalid input must be rejected.

Functions : identified functions must be exercised.

Output : identified classes of application outputs must be exercised.

Systems/Procedures: interfacing systems or procedures must be invoked.

Organization and preparation of functional tests is focused on requirements, key functions, or special test cases. In addition, systematic coverage pertaining to identify Business process flows; data fields, predefined processes, and successive processes must be considered for testing. Before functional testing is complete, additional tests are identified and the effective value of current tests is determined.

6.1.4 System testing

System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. An example of system testing is the configuration oriented system integration test. System testing is based on process descriptions and flows, emphasizing pre-driven process links and integration points.

6.1.5 White box testing

White Box Testing is a testing in which in which the software tester has knowledge of the inner workings, structure and language of the software, or at least its purpose. It is purpose. It is used to test areas that cannot be reached from a black box level.

6.1.6 Black box testing

Black Box Testing is testing the software without any knowledge of the inner workings, structure or language of the module being tested. Black box tests, as most other kinds of tests, must be written from a definitive source document, such as specification or requirements document, such as specification or requirements document. It is a testing in which the software under test is treated, as a black box, you cannot “see” into it. The test provides inputs and responds to outputs without considering how the software works.

6.1.7 Acceptance Testing

User Acceptance Testing is a critical phase of any project and requires significant participation by the end user. It also ensures that the system meets the functional requirements. The acceptance test suite may need to be performed multiple times, as all of the test cases may not be executed within single test iteration.

The acceptance test suite is run using predefined acceptance test procedures to direct the testers which data to use, the step-by-step processes to follow and the expected result following execution. The actual results are retained for comparison with the expected results.] If the actual results match the expected results for each test case, the test case is said to pass. If the quantity of non-passing test cases does not breach the project's predetermined threshold, the test suite is said to pass.

The anticipated result of a successful test execution:

- Test cases are executed, using training data
- Actual results are recorded, and
- Test results are determined.

CHAPTER 7

RESULTS AND SNAPSHOTS

7.1 Snapshots with description

7.1.1 User Login window

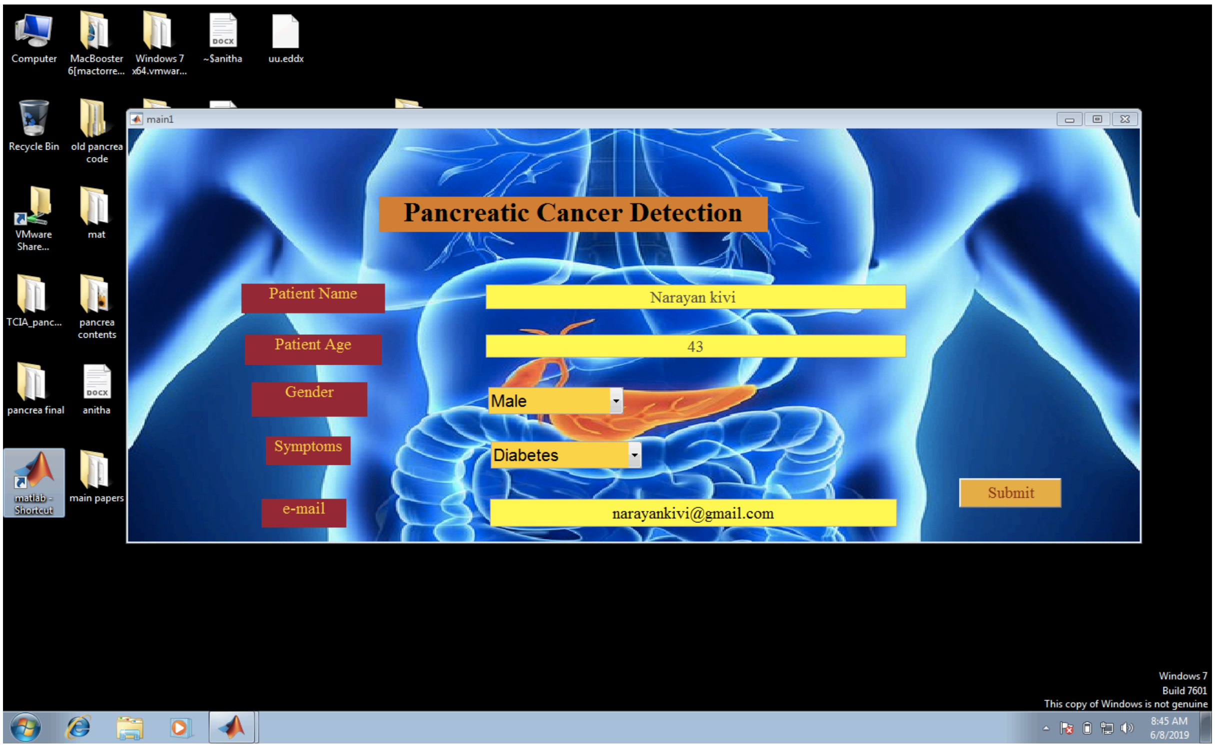

Fig 7.1.1 User Login window

Fig 7.1.1 shows the User Login window. Here the user has to enter his name, age, and select gender, symptoms and enter email address. Then the user can login successfully.

7.1.2 Main Graphical User Interface (GUI)

Fig 7.1.2 shows the Main Graphical User Interface (GUI). Once the user has been logged in successfully it redirects to the main GUI where actual operations are done. All the user information is being displayed. Here the user can interact with the GUI very easily.

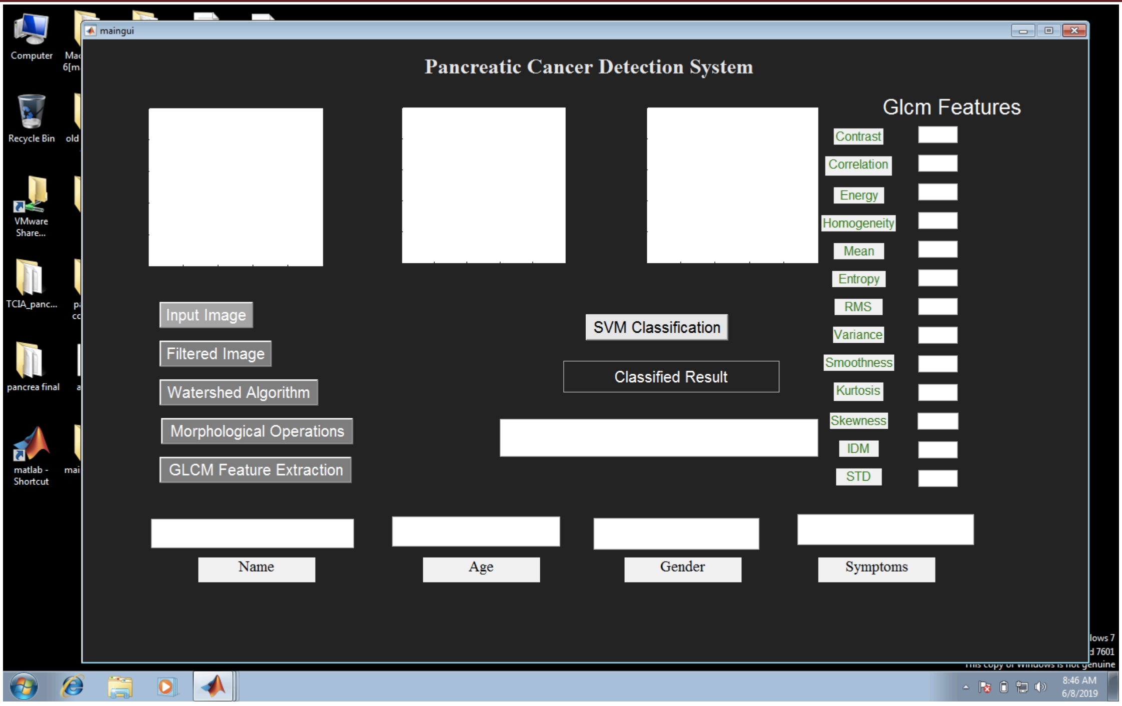

Fig 7.1.2 Main Graphical User Interface (GUI)

7.1.3 Selecting input image

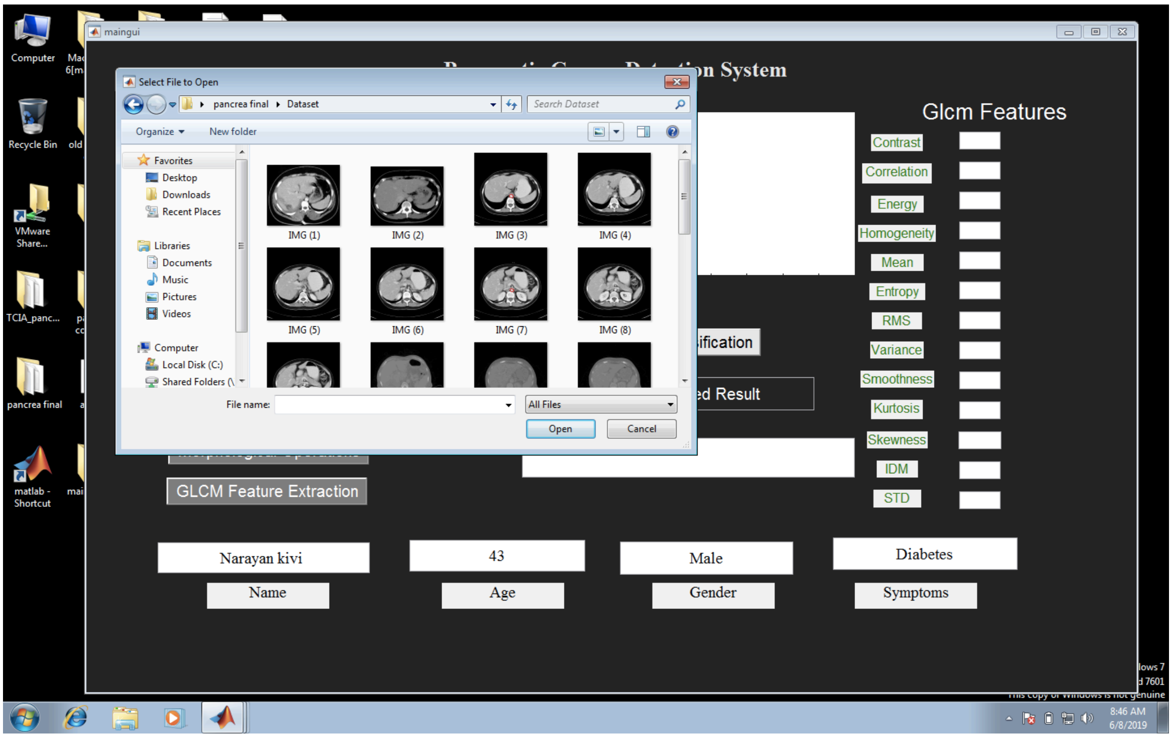

Fig 7.1.3 Selecting input image

Fig 7.1.3 shows the Selection of input image. The user selects the input of his/her CT scan images. CT scan images are fed into the system manually.

7.1.4 Image Filtering

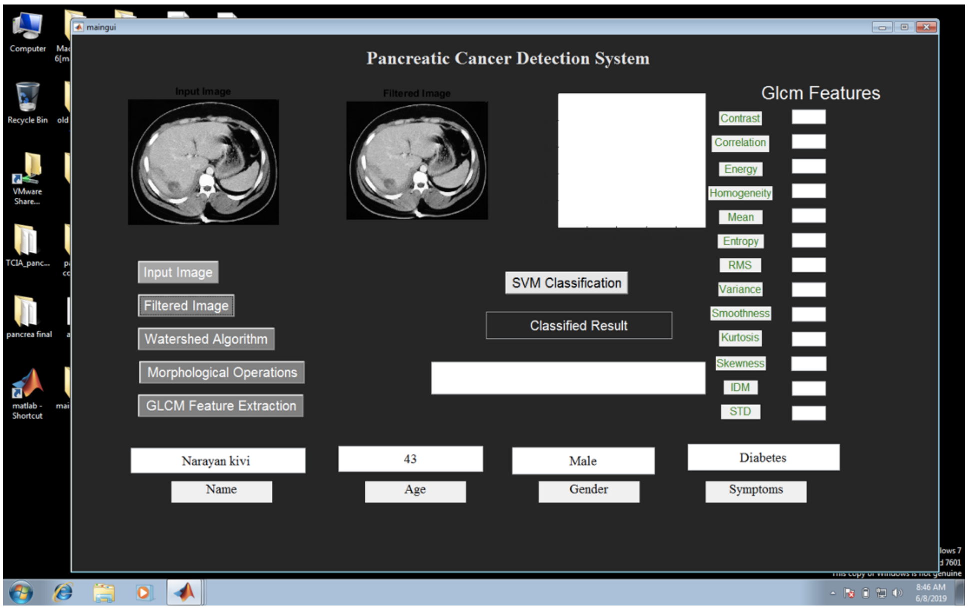

Fig 7.1.4 Image Filtering

Fig 7.1.4 shows the Image Filtering. The input image may contain noises so we have to filter it accordingly to get clear visuals. Gaussian filtering is used to filter images.

7.1.5 Segmentation

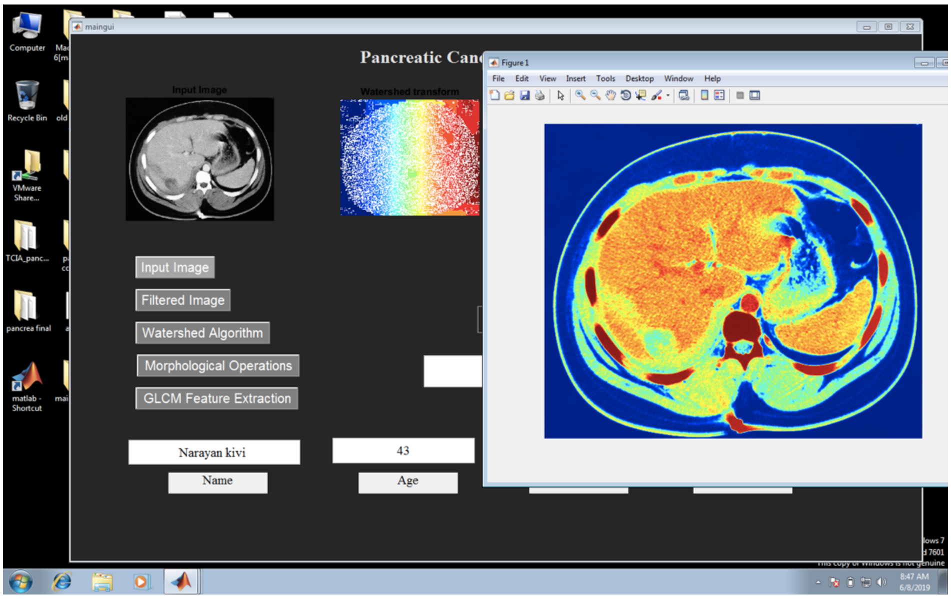

Fig 7.1.5 Segmentation

Fig 7.1.5 shows the Segmentation of the image. After filtering the image is segmented using marker controlled Watershed Segmentation.

7.1.6 Morphological operations

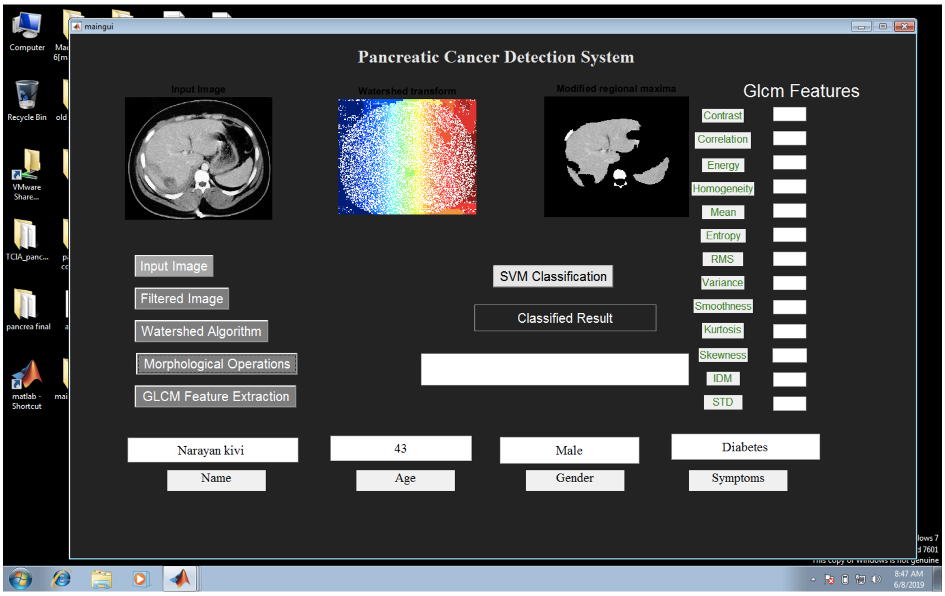

Fig 7.1.6 Morphological operations

Fig 7.1.6 shows the Morphological operations. After segmentation, the morphological operations are done to get clearer ROI.

7.1.7 Feature extraction

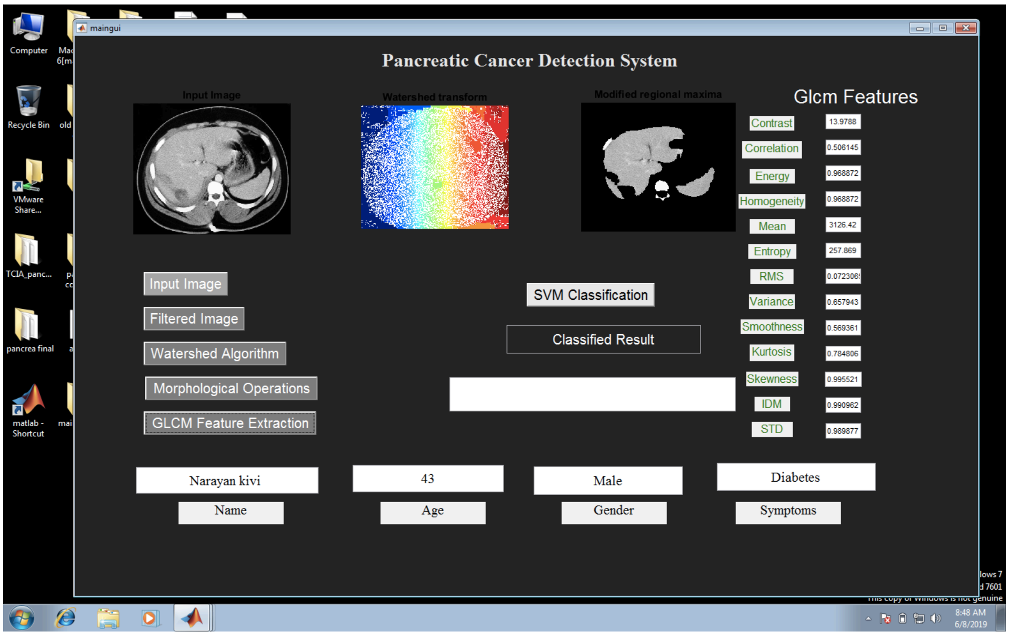

Fig 7.1.7 Feature extraction

Fig 7.1.7 shows Feature extraction. All the features are extracted using GLCM feature extraction.

7.1.8 Classification

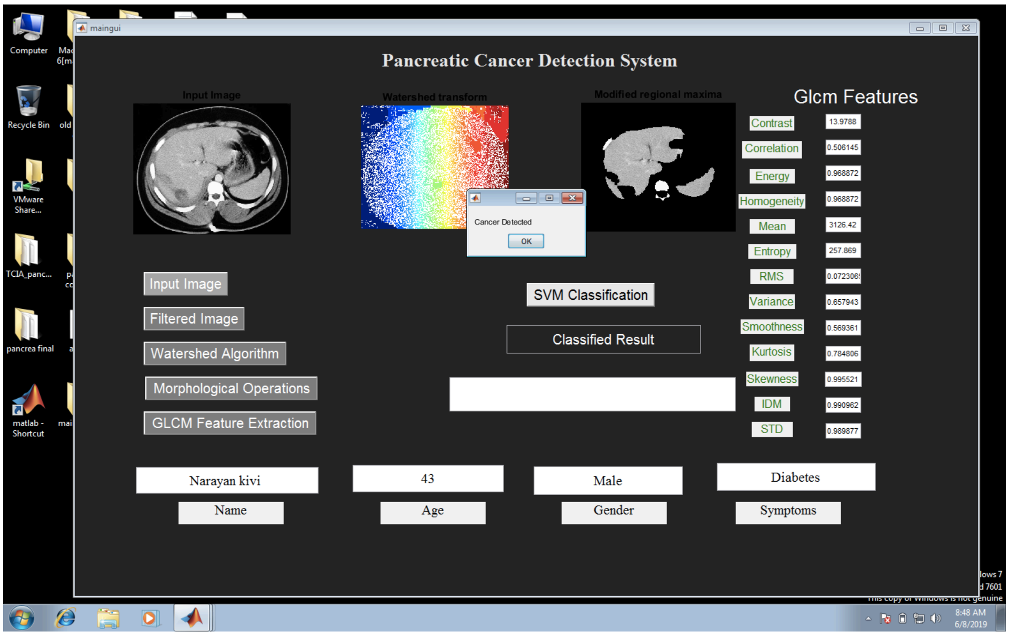

Fig 7.1.8 Classification

Fig 7.1.8 shows the Classification. By using the obtained features SVM classifier is used to detect whether the cancer is affected or not.

7.1.9 Result

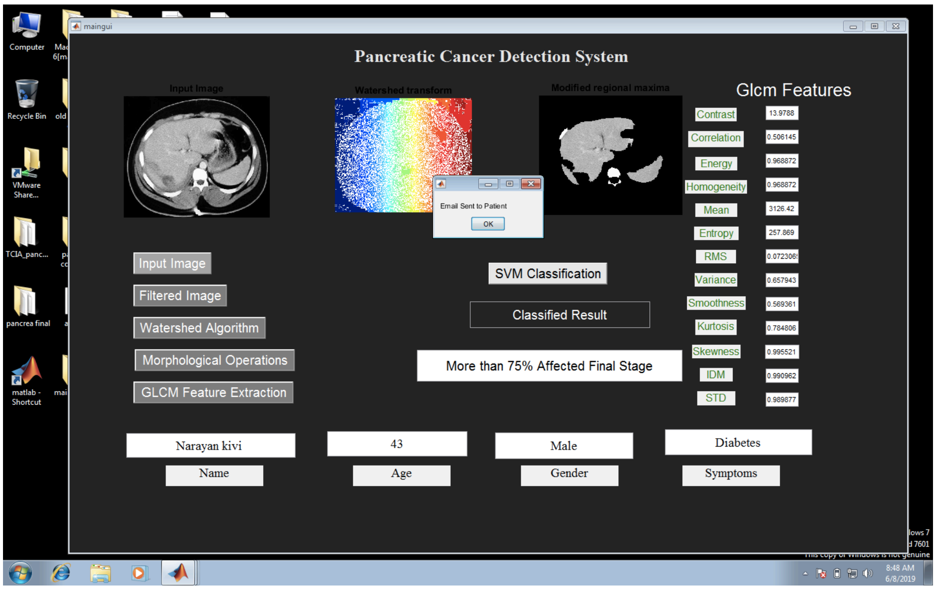

Fig 7.1.9(a) Cancer is affected

Fig 7.1.9(a) shows the final Result. Cancer is affected here and stage of cancer and how much it is affected is displayed on the screen to user. An email is sent to the user.

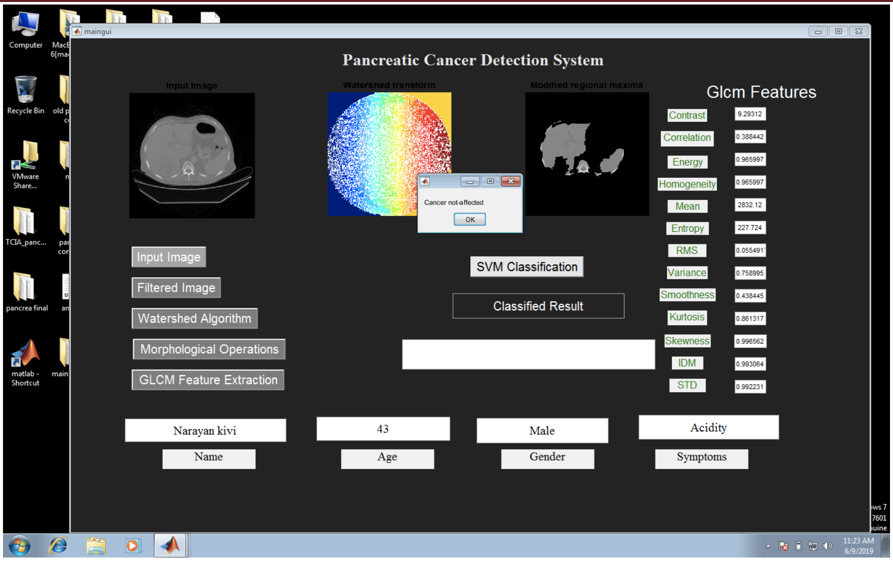

Fig 7.1.9(b) Cancer is not affected

Fig 7.1.9(b) shows result when Cancer is not affected and email is sent to user.

7.2 ADVANTAGES AND DISADVANTAGES

7.2.1 Advantages

- The main aim of this model is to provide the earlier warning to the users.
- It is Low cost. and is also less time consuming.
- Image processing techniques used provides maximum efficiency.
- Digital image processing has many advantages that allows a much wider range of algorithms to be applied to the input data and can avoid problems such as the build-up of noise and signal distortion during processing.

7.2.2 Disadvantages

- The system is not fully automated, it needs data from user for full diagnosis.
- User authentication is not provided.

7.3 Applications

- This model is used in the hospitals and clinical laboratory.
- It is used in healthcare environment for prediction of Pancreatic cancer.
- It can also be used in Government healthcare centres.

CHAPTER 8

FUTURE WORK

Future work will include the modification of the system to recognize images that have multiple scattered tumor on the Pancrea. Tthe proposed work can be extended by additional range of algorithms on additional range of dataset from huge medical database.the perormance analysis of different classification algorithms can be done to provide better result.

CONCLUSION

Pancreatic cancer is one of the most dangerous diseases in the world. Correct Diagnosis and detection of pancreatic cancer can increase the survival rate. CT scan images are acquired from various hospitals. These images include less noise as compared to X-ray and MRI images. The CT captured images are processed. The System consist of Pre-processing, Segmentation ,Feature extraction and final classification. The proposed marker controlled Watershed Segmentation technique separates the touching objects in the image. It provides best identification of the main edge of the image and also avoids over segmentation. The SVM classifier increases accuracy of detection and reduces false detection. The proposed technique gives very promising results comparing with other used techniques.