KIDNEY STONE DETECTION

USING IMAGE PROCESSING TECHNIQUES

A Project Report Submitted
in Partial Fulfillment of the Requirements for the Degree of

MASTER OF COMPUTER APPLICATIONS

By

PODILA SAI
(L19CA140)

DECEMBER -2020

R.V.R. & J.C. COLLEGE OF ENGINEERING
(AUTONOMOUS)
(Approved by A.I.C.T.E.)
Chandramoulipuram :: Chowdavaram
GUNTUR -19

KIDNEY STONE DETECTION

USING IMAGE PROCESSING TECHNIQUES

A Project Report Submitted
in Partial Fulfillment of the Requirements for the Degree of

MASTER OF COMPUTER APPLICATIONS

By

PODILA SAI
(L19CA140)

Under the guidance of
Dr. M.SRIDHAR

Associate Professor

DECEMBER-2020

DEPARTMENT OF COMPUTER APPLICATIONS

R.V.R& J.C COLLEGE OF ENGINEERING
(AUTONOMOUS)
(GUNTUR-1

R.V.R & J.C COLLEGE OF ENGINEERING
(AUTONOMOUS)

DEPARTMENT OF COMPUTER APPLICATIONS

BONAFIDE CERTIFICATE

This is to certify that the project work entitled "**KIDNEY STONE DETECTION USING IMAGE PROCESSING TECHNIQUES**" is the bona fied work of Mr. **PODILA SAI (L19CA140)** who carried out the work under my supervision, in partial fulfillment of the requirements for the award of the degree, **MASTER OF COMPUTER APPLICATIONS**, during the year 2020-2021.

(**Dr. M.Sridhar**)
Associate Professor

(**Dr. K.Karteeka Pavan**)
Prof., & H.O.D

Abstract

Nowadays, kidney stone has become a major problem and if not detected at an early stage then it may cause complications and sometimes surgery is also needed to remove the stone. Kidney stone disease is when a solid piece of material occurs in the urinary tract. The small piece of stone may pass without causing symptoms. If a stone grows to more than 5 millimeters it can cause blockage of the ureter resulting in severe pain in the lower back or Abdomen. Hence it's necessary to have an approach to detect the stone in the kidney to avoid further health issues. Generally we use ultrasound technique to find the stones in kidney. But, the image produced by the ultrasound techniques is not suitable for further processing due to low contrast and the presence of speckle noise. Hence, the study also examined the effectiveness of various de-noising techniques on the ultrasound image to enhance the quality of the image. Further, the enhanced ultrasound image is used to locate the exact position of the stone.

ACKNOWLEDGEMENT

From the idea to the art, from the conception to the reality, from the emotion to the response, from the desire to the spasm, I led by those, about whom to write all words seem meek.

I express my sincere thanks to **Dr. M.Sridhar,** Asso.Prof, mini project guide, who was inspired me to select this topic and also for his valuable advices to complete this project.

I am very glad to express my special thanks to **Smt T.Hymavathi,** Asst.Prof, mini project-in-charge, who was inspired me to select this topic and also for her valuable advices in preparing this project.

I express my sincere thanks to **Dr. K.Karteeka Pavan**, Head of the Department of Computer Applications for her encouragement and support to carry out this major project successfully.

I am very much thankful to **Dr. A.Sudhakar**, Principal of **RVR & JC COLLEGE OF ENGINEERING**, Guntur for having allowed to delivering this mini project.

Finally I submit my reserves thanks to Lab staff in **Department of Computer Applications** and to all my **friends** for their cooperation during the preparation and developing the mini project.

PODILA SAI

(L19CA140)

Page Index

LIST OF TABLES

LIST OF FIGURES

LIST OF SCREENS

Chapter-1

INTRODUCTION

1.1. INTRODUCTION

Nowadays, kidney stone has become a major problem and if not detected at an early stage then it may cause complications and sometimes surgery is also needed to remove the stone. Kidney stone disease is when a solid piece of material occurs in the urinary tract. The small piece of stone may pass without causing symptoms. If a stone grows to more than 5 millimeters it can cause blockage of the ureter resulting in severe pain in the lower back or Abdomen. Hence it's necessary to have an approach to detect the stone in the kidney to avoid further health issues. Generally we use ultrasound technique to find the stones in kidney. But, the image produced by the ultrasound techniques is not suitable for further processing due to low contrast and the presence of speckle noise. Rahman and Uddin have proposed diminution of speckle noise and segmentation from US image. It not only detects problem in the kidney region but also provides image quality enhancement. Hence, the study also examined the effectiveness of various de-noising techniques on the ultrasound image to enhance the quality of the image. Further, the enhanced ultrasound image is used to locate the exact position of the stone.

1.2. OBJECTIVE

The main objective of this project is to detect the kidney stone from a digital ultrasound image of the kidney by performing various image processing techniques. But, the image produced by the ultrasound techniques is not suitable for further processing due to low contrast and the presence of speckle noise. Hence, the study also examined the effectiveness of various de-noising techniques on the ultrasound image to enhance the quality of the image. Further, the enhanced ultrasound image is used to locate the exact position of the stone.

1.3 EXISTING SYSTEM

CT scan:

- A computed tomography (CT or CAT) scan allows doctors to see inside your body. It uses a combination of X-rays and a computer to create pictures of your organs, bones, and other tissues. It shows more detail than a regular X-ray.
- You can get a CT scan on any part of your body. The procedure doesn't take very long, and it's painless. They use a narrow X-ray beam that circles around one part of your body. This provides a series of images from many different angles. A computer uses this information to create a cross-sectional picture. Like one piece in a loaf of bread, this two-dimensional (2D) scan shows a “slice” of the inside of your body.

Ultrasound:

- A type of imaging test to examine the internal organs using very high frequency sound waves. Ultrasound (US) images have recently become more widespread because of their better quality and portability for medical applications. Ultrasound is mostly preferred for initial medical monitoring, diagnosis, and follow-up in comparison to other medical image modalities for its low cost. Because of its benefits, it is broadly utilized for the diagnosis of various diseases in the liver, breast, prostate, and thyroid.
- Ultrasound is safe and effective when used to detect and remove the_stones without exposing patients and operating room staff to radiation.

DISADVANTAGES OF EXISTING SYSTEM

CT scan:

- Cost effective and not feasible.
- High doses of radiation are involved in CT scanning - chest CT scan is equivalent to 350 chest X-rays; CT abdomen to 400 chest X-ray.

Ultrasound:

- It has poor penetration through bone or air. Moreover it has limited penetration in obese patients. These are the disadvantages in imaging domain
- The image produced by the ultrasound techniques is not suitable for identifying small stones due to low contrast and the presence of speckle noise.

1.4. PROPOSED SYSTEM

Performing various de-noising techniques on the ultrasound image to enhance the quality of the image. Further, the enhanced ultrasound image is used to locate the exact position of the stones and also marking the stones. This detection can be done in any available PC's and hence any normal being can check an ultrasound for a kidney stone and dissolve it in the starting stage. These techniques mainly help the doctor to further treat the patient based on the size and location of the stone.

ADVANTAGES OF PROPOSED SYSTEM

- The system provides a realizable and efficient.
- We can find out the exact position of the stones and also marking the stones.
- This detection can be done in any available PC's and hence any normal being can check an ultrasound for a kidney stone

Chapter-2

SYSTEM ANALYSIS

2.1 INPUT AND OUTPUT

Inputs:

- Importing the all required packages like numpy, pandas, matplotlib, scikit – learn and required machine learning algorithms packages.
- Setting the dimensions of visualization graph.

Outputs:

- Preprocessing the importing data frame for imputing nulls with the related information.
- All are displaying cleaned outputs.
- After applying machine learning algorithms it will give good results and visualization plots.

2.2 FEASIBILITY REPORT

Preliminary investigation examine project feasibility, the likelihood the system will be useful to the organization. The main objective of the feasibility study is to test the Technical, Operational and Economical feasibility for adding new modules and debugging old running system. All system is feasible if they are unlimited resources and infinite time. There are aspects in the feasibility study portion of the preliminary investigation:

1. Technical Feasibility
2. Economical Feasibility
3. Operation Feasibility

A. TECHINICAL FEASIBILITY:

In the feasibility study first step is that the organization or company has to decide that what technologies are suitable to develop by considering existing system.

Here in this application used the technologies like Visual Studio 2008 and SQL Server 2005. These are free software that would be downloaded from web.

B. OPERATIONAL FEASIBILITY:

Not only must an application make economic and technical sense, it must also make operational sense.
Issues to consider when determining the operational feasibility of a project.

Operations Issues	Support Issues
What tools are needed to support operations? What skills will operators need to be trained in? What processes need to be created and/or updated? What documentation does operations need?	What documentation will users be given? What training will users be given? How will change requests be managed?

Very often you will need to improve the existing operations, maintenance, and support infrastructure to support the operation of the new application that you intend to develop. To determine what the impact will be you will need to understand both the current operations and support infrastructure of your organization and the operations and support characteristics of your new application.

To operate this application this system that the user no needs to require any technical knowledge that we are used to develop this project is. Asp.net, C#.net. that the application providing rich user interface by user can do the operation in flexible manner.

C. ECONOMIC FEASIBILITY:

It refers to the benefits or Outcomes we are deriving from the product as compared to the total cost we are spending for developing the product. If the benefits are more or less the same as the older system, then it is not feasible to develop the product. In the present system, the development of new product greatly enhances the accuracy of the system and cuts short the delay in the processing this application. The errors can be greatly reduced and at the same time providing a great level of security. Here we don’t need any additional equipment except memory of required capacity. No need for spending money on client for maintenance because the database used is web enabled database.

Chapter-3

EXPERIMENTAL ANALYSIS

3.1 INTRODUCTION TO MECHINE LEARNING

Machine Learning (ML) is basically that field of computer science with the help of which computer systems can provide sense to data in much the same way as human beings do. In simple words, ML is a type of artificial intelligence that extract patterns out of raw data by using an algorithm or method. The key focus of ML is to allow computer systems to learn from experience without being explicitly programmed or human intervention.

3.1.1 NEED FOR MACHINE LEARNING

Human beings, at this moment, are the most intelligent and advanced species on earth because they can think, evaluate and solve complex problems. On the other side, AI is still in its initial stage and hasn't surpassed human intelligence in many aspects. Then the question is that what is the need to make machine learn? The most suitable reason for doing this is, "to make decisions, based on data, with efficiency and scale". Lately, organizations are investing heavily in newer technologies like Artificial Intelligence, Machine Learning and Deep Learning to get the key information from data to perform several real-world tasks and solve problems. We can call it data-driven decisions taken by machines, particularly to automate the process. These data-driven decisions can be used, instead of using programming logic, in the problems that cannot be programmed inherently. The fact is that we can't do without human intelligence, but other aspect is that we all need to solve real-world problems with efficiency at a huge scale. That is why the need for machine learning arises.

3.1.2 CHALLENGES IN MACHINES LEARNING

While Machine Learning is rapidly evolving, making significant strides with cyber security and autonomous cars, this segment of AI as whole still has a long way to go. The reason behind is that ML has not been able to overcome number of challenges. The challenges that ML is facing currently are

- **Quality of data** − Having good-quality data for ML algorithms is one of the biggest challenges. Use of low-quality data leads to the problems related to data preprocessing and feature extraction.

- **Time-Consuming task** − Another challenge faced by ML models is the consumption of time especially for data acquisition, feature extraction and retrieval.
- **Lack of specialist persons** − As ML technology is still in its infancy stage, availability of expert resources is a tough job.
- **No clear objective for formulating business problems** − Having no clear objective and well-defined goal for business problems is another key challenge for ML because this technology is not that mature yet.
- **Issue of over fitting & under fitting** − If the model is over fitting or under fitting, it cannot be represented well for the problem.
- **Curse of dimensionality** − Another challenge ML model faces is too many features of data points. This can be a real hindrance.
- **Difficulty in deployment** − Complexity of the ML model makes it quite difficult to be deployed in real life.

3.1.3 APPLICATIONS OF MACHINES LEARNING

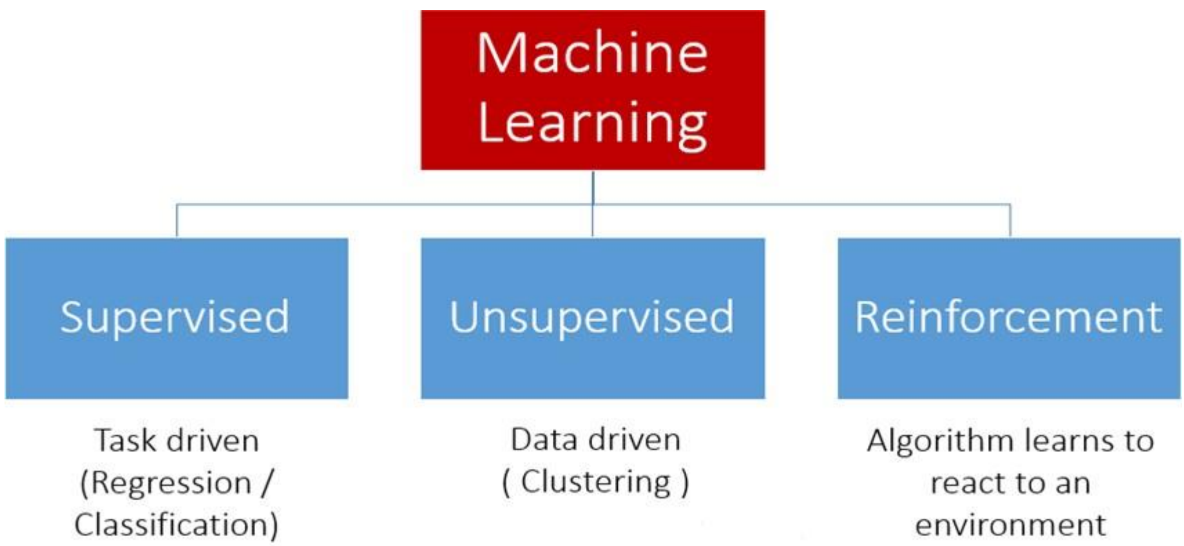
Machine Learning is the most rapidly growing technology and according to researchers we are in the golden year of AI and ML. It is used to solve many real-world complex problems which cannot be solved with traditional approach. Following are some real-world applications of ML

- Emotion analysis
- Sentiment analysis
- Error detection and prevention
- Weather forecasting and prediction
- Stock market analysis and forecasting
- Speech synthesis
- Speech recognition
- Customer segmentation
- Object recognition
- Fraud detection

- Fraud prevention
- Recommendation of products to customer in online shopping.

3.2 TYPES OF MACHINE LEARNING

At a high-level, machine learning is simply the study of teaching a computer program or algorithm how to progressively improve upon a set task that it is given. On the research-side of things, machine learning can be viewed through the lens of theoretical and mathematical modeling of how this process works. However, more practically it is the study of how to build applications that exhibit this iterative improvement. There are many ways to frame this idea, but largely there are three major recognized categories: supervised learning, unsupervised learning, and reinforcement learning.



3.2.1 Machine Learning

SUPERVISED LEARNING

Supervised learning algorithms build a mathematical model of a set of data that contains both the inputs and the desired outputs.[] The data is known as training data, and consists of a set of training examples. Each training example has one or more inputs and the desired output, also known as a

Supervisory signal. In the mathematical model, each training example is represented by an array or vector, sometimes called a feature vector, and the training data is represented by a matrix. Through iterative optimization of an objective function, supervised learning algorithms learn a function that can be used to predict the output associated with new inputs. An optimal function will allow the algorithm to correctly determine the output for inputs that were not a part of the training data. An algorithm that improves the accuracy of its outputs or predictions over time is said to have learned to perform that task.

UNSUPERVISED LEARNING

Unsupervised learning algorithms take a set of data that contains only inputs, and find structure in the data, like grouping or clustering of data points. The algorithms, therefore, learn from test data that has not been labeled, classified or categorized. Instead of responding to feedback, unsupervised learning algorithms identify commonalities in the data and react based on the presence or absence of such commonalities in each new piece of data. A central application of unsupervised learning is in the field of density estimation in statistics, such as finding the probability density function. Though unsupervised learning encompasses other domains involving summarizing and explaining data features.

Cluster analysis is the assignment of a set of observations into subsets (called clusters) so that observations within the same cluster are similar according to one or more predesignated criteria, while observations drawn from different clusters are dissimilar. Different clustering techniques make different assumptions on the structure of the data, often defined by some similarity metric and evaluated, for example, by internal compactness, or the similarity between members of the same cluster, and separation, the difference between clusters. Other methods are based on estimated density and graph connectivity.

REINFORCEMENT LEARNING

Reinforcement learning is an area of machine learning concerned with how software_agents ought to take actions in an environment so as to maximize some notion of cumulative reward. Due to its generality, the field is studied in many other disciplines, such as game theory, control theory, operations research, information theory, simulation-based optimization, multi-agent systems, swarm intelligence, statistics and genetic algorithms. In machine learning, the environment is typically represented as a Markov Decision Process (MDP). Many reinforcement learning algorithms use dynamic programming techniques. Reinforcement learning algorithms do not assume knowledge of an exact mathematical model of the MDP, and are used when exact models are infeasible. Reinforcement learning algorithms are used in autonomous vehicles or in learning to play a game against a human opponent.

3.3 EXPERIMENTAL ANALYSIS

To detect the kidney stone, we used the following approach:

The kidney stone produces an acoustic shadow in the ultrasound image. An acoustic shadow or sound shadow is an area through which the sound waves fail to propagate due to topographical obstructions. In this case, the kidney stone is the barrier that disrupts the sound waves. Once we detect the shadow, the stone can be located just above it in the image.

The ultrasound images contain speckle noise. To process the image and detect the location of the stone in the image, we need to remove the noise. Thus, the first step is to enhance the image by using various sharpening and smoothing filters.

After the image is enhanced, image segmentation is used to differentiate between the shadow and the stone by separating the foreground and the background

We will use the following various sharpening and smoothing filters:

Gaussian Filter:

- ✧ A Gaussian filter is a linear filter. It's usually used to blur the image or to reduce noise. If you use two of them and subtract, you can use them for "un-sharp masking" (edge detection). The Gaussian filter alone will blur edges and reduce contrast.

Median Blur:

- In general median filter may be used for filtering different types of noise but it is the best choice for images corrupted by 'salt and pepper' noise. This is not common type of noise. It appears when a CCD matrix pixel is corrupted or when the data transmission is unstable. A situation of that kind occurs on the satellite imagery (data transmission errors) or MRI scans. Median filter works very well with the salt and pepper noise.

Gaussian Blur:

- Gaussian blur (also known as Gaussian smoothing) is the result of blurring an image by a Gaussian function (named after mathematician and scientist Carl Friedrich Gauss). It is a widely used effect in graphics software, typically to reduce image noise and reduce detail. The visual effect of this blurring technique is a smooth blur resembling that of viewing the image through a translucent screen, distinctly different from the bokeh effect produced by an out-of-focus lens or the shadow of an object under usual illumination.

Laplacian filter:

- The Laplacian filter is an edge-sharpening filter, which sharpens the edges of the image.

Gabor filters:

- In image processing, a Gabor filter, named after Dennis Gabor, is a linear filter used for texture analysis, which essentially means that it analyzes whether there is any specific frequency content in the image in specific directions in a localized region around the point or region of analysis.

The first step was to get a clear and better quality of the data so we worked on the **IMAGE ENHANCEMENT**:

1. The first one we applied was Frequency Domain + Gaussian Filter. Here we had to divide the image into two plains real plain and complex plain and then apply the Gaussian Filter in both.

2. The second attempt was with Gaussian Blur + Laplacian Filter. We had to apply Gaussian Blur to make Laplacian filter less sensitive to noise.

3. On further studies, we found in some research papers (link given below) that Ultrasounds give speckle noise. So, our third attempt was to replace Gaussian Blur with Median Blur as the latter is more effective for low-level noises like speckle noise, salt-and-pepper noise etc. followed by Laplacian filter.

4. In the application of the Gabor filter, the restored image is enriched with optimal resolution in both spatial and frequency domains (as stated in one of the research papers, whose link is given below). 2-D Gabor filter is easier to tune the direction and radial frequency bandwidth, and easier to tune centre frequency, so they can simultaneously get the best resolution in the spatial domain and frequency domain.

Now we moved to our next step **HISTOGRAM EQUALIZATION**.

It is a technique for adjusting image intensities to enhance contrast. It assigns the intensity values of pixels in the input image such that the output image contains a uniform distribution of intensities. It improves contrast and obtains a uniform histogram. This technique can be used on a whole image or just on a part of an image. This process leads to an increase in contrast of the shadow of the stone and the stone itself. Thus, it became more visible.

3.3.1 Histogram Equalization Algorithm

Histogram Equalization aims to enhance the contrast of an image by stretching out the most frequently used intensity values. It's objective is to increase contrast in areas where it's low resulting in an image that displays an increased number of darker and lighter areas.

The histogram equalization algorithm can be broken up into four main parts:

- Creating the histogram
- Calculating the histograms cumulative sum
- Making a mapping from old colours to new ones
- Applying the mapping to create our new image

Creating the Histogram

- The first step is to create a histogram that records the number of pixels at each intensity level, we do this for two reasons:

- It allows us to see how good/bad the colour range of an image is

- We use this histogram to calculate our mapping in the next couple of stages

```
defmake_histogram(img):
""" Take a flattened greyscale image and create a historgram from it """
 histogram=np.zeros(256,dtype=int)
for i in range(img.size):
histogram[img[i]] += 1 return histogram
```

- Where the img parameter is a flattened (1-dimensional) array containing pixel values for our image. The function outputs an array where each index represents the number of pixels at that grey level e.g. if histogram[5] = 99 it means 99 pixels in the image have a grey level of 5.

Calculating Cumulative Sum

The next step is to calculate the cumulative sum (a.k.a running total) of our histogram. This is important for the next stage as it's used in our mapping calculation; luckily for us, it's easy to calculate! Let's look at the Python code:

```
def make_cumsum(histogram):

""" Create an array that represents the cumulative sum of the histogram """

 cumsum = np.zeros(256, dtype=int)

 cumsum[0] = histogram[0]

 for i in range(1, histogram.size):

 cumsum[i] = cumsum[i-1] + histogram[i]

return cumsum
```

- The histogram parameter is the output from our make_histogram(img) function. This function returns an array where each index is the sum of everything before it + itself. For example, if the histogram started with [5, 2, 9, 1, ...] the start of cumsum would be [5, 7, 16, 17, ...]

Create a Mapping

- Now for the fun part, let's make our mapping! The mapping is created from the following formula:

- $M(i)=max(0,round((grey\_levels*cumsum(i))/(h*w))-1)$

where:

i = 0 … grey_levels

grey_levels = number of grey levels in our image (usually 256)

cumsum(i) = i'th element of the array produced by our make_cumsum(histogram) function

h, w = the height and width of our image

Apply the Mapping

- Time to create our new image, to do this we simply apply the mapping we made in the last step:

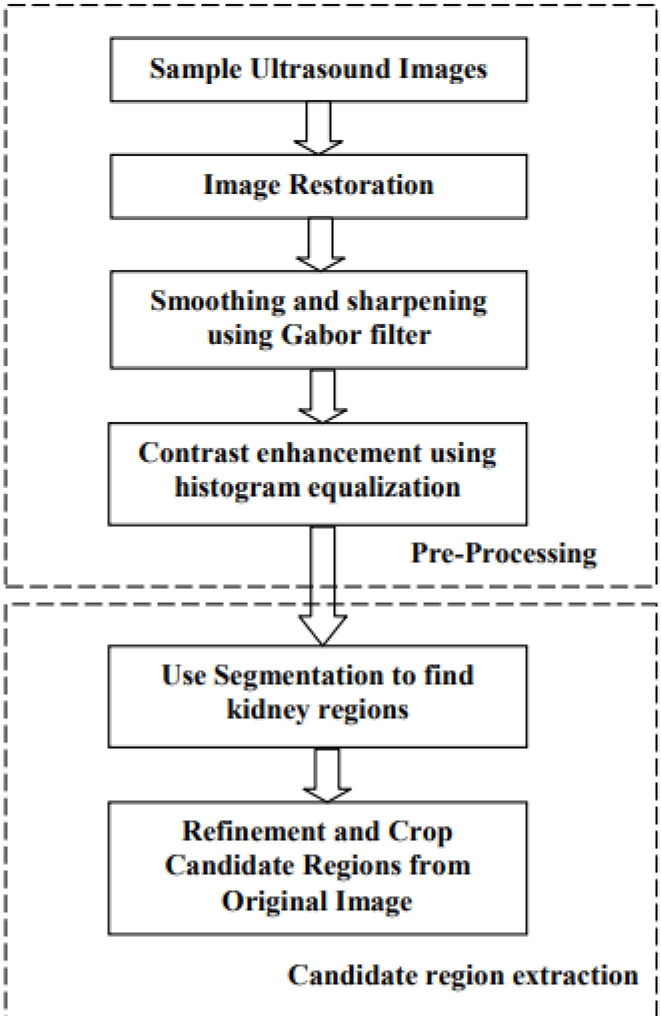
- new_image[i] = mapping[img[i]]

- The output of this function is an array containing the pixel values of the new, histogram equalized image

After this, we proceeded towards image segmentation.

3.3.2 IMAGE SEGMENTATION

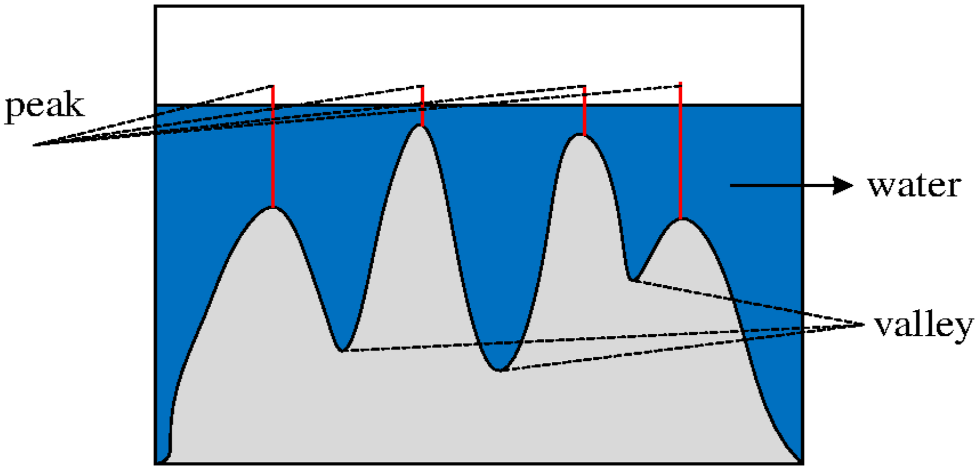
It partitions an image into distinct regions containing each pixel with similar attributes. To be meaningful and useful for image analysis and interpretation, the regions should strongly relate to depicted objects or features of interest. Meaningful segmentation is the first step from low-level image processing transforming a grayscale or color image into one or more other images to high-level image description in terms of features, objects, and scenes. Hereby performing this, we needed to partition our stone from the rest of the image. The type of Image Segmentation used here is Watersheds.



3.3.2.1: Block diagram of kidney segmentation system

3.3.3 Watersheds:

Here we divide the image into peaks (high intensity) and valleys (low intensity). We fill the valleys (points of minima or background) with water of different colors (labels). As the water rises, depending on the peaks(gradients) nearby, water from different valleys, obviously with different colors will start to merge. To avoid this, we need to build barriers in the area where the water is mixing. Also, we continue the process of filling water until all the peaks are under water. The barriers, thus, created are the result of image segmentation. Now, the shadow gets separated from the stone giving a clear image of the stone.



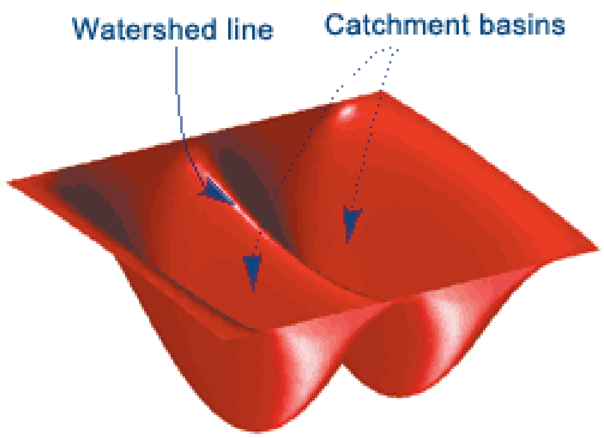
3.3.3.1 Watershed representation

Basic concept of watershed algorithm is to find the watershed lines.

•We flood this surface from its minima

•And we prevent the merging of the waters coming from different sources.

• We partition the image into two different sets: the catchment basins and the watershed lines.



3.3.3.2 Watershed lines

Steps in Watershed Algorithm:

- Read in an Image and covert it in grayscale
- Use the gradient magnitude as the segmentation function
- Mark the foreground objects
- Compute the Background markers
- Compute the watershed transform of the segmentation function
- Visualize the result.

3.3.4 MARKING:

It provides you with a way to identify or label a spot within the processing pipeline. Adding a marker into the pipeline will allow you to refer to the image currently being processed at that point.

It can also be used to specify a location in the pipeline for other modules.

RESULTS

Total of 4 approaches have been performed to enhance the quality of the image, namely: Gabor Filter, Median blur + Laplacian Filter, Gaussian blur + Laplacian Filter, Gaussian Filtering in Frequency domain. Out of the 4, we have noticed that Gabor Filter was effective on a larger number of images though not all. Nextly Median blur + Laplacian Filter and Gaussian blur + Laplacian Filter worked perfectly with other of the few images which are lesser compared to Gabor Filter. Lastly, Gaussian Filtering in Frequency domain, which didn’t work as expected and hence this technique was not employed to enhance any of the other images. We also could generalize the effectiveness of the techniques due to lack of data samples.

Chapter-4

SYSTEM IMPLEMENTATION

4.1 SOFTWARE REQUIREMENTS

- OS : Windows
- Python IDE : python 3.x and above
- Jupyter Notebook, Anaconda 3.5 Setup tools and pip to be installed for 3.6.x and above

4.2 HARDWARE REQUIREMENTS

- RAM : 4GB and Higher
- Processor : Intel i3 and above
- Hard Disk : 500GB: Minimum

4.3 TOOLS USED

1. Numpy
2. Matplotlib
3. OpenCV

1. Numpy:

Numpy is a general-purpose array-processing package. It provides a high-performance multidimensional array object, and tools for working with these arrays.

It is the fundamental package for scientific computing with Python. It contains various features including these important ones:

- A powerful N-dimensional array object
- Sophisticated (broadcasting) functions
- Tools for integrating C/C++ and Fortran code
- Useful linear algebra, Fourier transform, and random number capabilities

Besides its obvious scientific uses, Numpy can also be used as an efficient multi-dimensional container of generic data. Arbitrary data-types can be defined using Numpy which allows Numpy to seamlessly and speedily integrate with a wide variety of databases.

2. Matplotlib

Matplotlib is a Python 2D plotting library which produces publication quality figures in a variety of hardcopy formats and interactive environments across platforms. Matplotlib can be used in Python scripts, the Python and IPython shells, the Jupyter notebook, web application servers, and four graphical user interface toolkits. Matplotlib tries to make easy things easy and hard things possible. You can generate plots, histograms, power spectra, bar charts, error charts, scatter plots, etc., with just a few lines of code. For examples, see the sample plots and thumbnail gallery.

For simple plotting the pyplot module provides a MATLAB-like interface, particularly when combined with IPython. For the power user, you have full control of line styles, font properties, axes properties, etc, via an object oriented interface or via a set of functions familiar to MATLAB users.

3. OpenCV

OpenCV is a cross-platform library using which we can develop real-time computer vision applications. It mainly focuses on image processing, video capture and analysis including features like face detection and object detection. In this tutorial, we explain how you can use OpenCV in your applications.

Computer Vision overlaps significantly with the following fields −

- **Image Processing** − It focuses on image manipulation.
- **Pattern Recognition** − It explains various techniques to classify patterns.
- **Photogrammetric** − It is concerned with obtaining accurate measurements from images.

Features of OpenCV Library

Using OpenCV library, you can −

- Read and write images
- Capture and save videos
- Process images (filter, transform)
- Perform feature detection
- Detect specific objects such as faces, eyes, cars, in the videos or images.
- Analyze the video, i.e., estimate the motion in it, subtract the background, and track objects in it.

4.4 Coding

```
import cv2
import argparse
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline
s = r'C:\Users\Arnab Sinha\Documents\GitHub\Kidney-Stone-Detection-IP\images'
image_no = '\image1.jpg'
s = s + image_no
img = cv2.imread(s,0)
def build_filters():
    #returns a list of kernels in several orientations
    filters = []
    ksize = 31
    for theta in np.arange(0, np.pi, np.pi / 32):
        params = {'ksize': (ksize, ksize), 'sigma': 0.0225, 'theta': theta, 'lambd': 15.0,
                  'gamma': 0.01, 'psi': 0, 'ktype': cv2.CV_32F}

        kern = cv2.getGaborKernel(**params)
        kern /= 1.5*kern.sum()
        filters.append((kern, params))
    return filters
def process(img, filters):
    #returns the img filtered by the filter list
    accum = np.zeros_like(img)
    for kern, params in filters:
        fimg = cv2.filter2D(img, cv2.CV_8UC3, kern)
        np.maximum(accum, fimg, accum)
    return accum
def Histeq(img):
    equ = cv2.equalizeHist(img)
    return equ
def GaborFilter(img):
    filters = build_filters()
    p = process(img, filters)
    return p
def Laplacian(img,par):
    lap = cv2.Laplacian(img,cv2.CV_64F)
```

```
    sharp = img - par*lap
    sharp = np.uint8(cv2.normalize(sharp, None, 0 , 255, cv2.NORM_MINMAX))
    return sharp
def Watershed(img):
    ret, thresh = cv2.threshold(img,0,255,cv2.THRESH_BINARY+cv2.THRESH_OTSU)
    # noise removal
    kernel = np.ones((3,3),np.uint8)
    opening = cv2.morphologyEx(thresh,cv2.MORPH_OPEN,kernel, iterations = 2)
    # sure background area
    sure_bg = cv2.dilate(opening,kernel,iterations=3)
    # Finding sure foreground area
    dist_transform = cv2.distanceTransform(opening,cv2.DIST_L2,5)
    ret, sure_fg = cv2.threshold(dist_transform,0.23*dist_transform.max(),255,0)
    # Finding unknown region
    sure_fg = np.uint8(sure_fg)
    unknown = cv2.subtract(sure_bg,sure_fg)

    # Marker labelling
    ret, markers = cv2.connectedComponents(sure_fg)
    # Add one to all labels so that sure background is not 0, but 1
    markers = markers+1
    # Now, mark the region of unknown with zero
    markers[unknown==255] = 0
    img2 = cv2.imread(s,1)
    img2 = cv2.medianBlur(img2,5)
    markers = cv2.watershed(img2,markers)
    img2[markers == -1] = [255,0,0]
    return img2
if image_no=='\image1.jpg':
    img3 = Laplacian(img,0.239)

elif image_no=='\image2.jpg':
    img3 = GaborFilter(img)
    img3 = Histeq(img3)
elif image_no=='\image4.jpg':
    img3 = GaborFilter(img)
img3 = Watershed(img)
plt.imshow(img3,'gray')
plt.title('Marked')
plt.xticks([]),plt.yticks([])
```

4.5 OUTPUT SCREENS

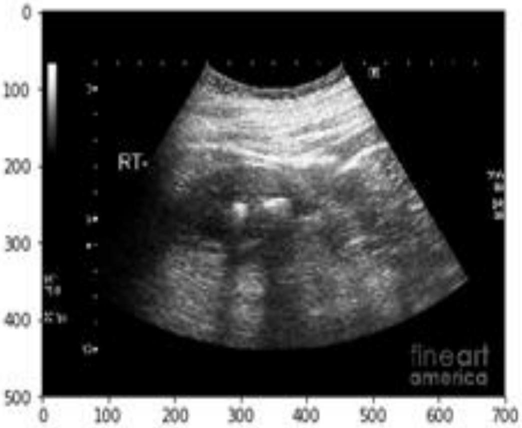


Fig 4.5.1 Original image

Fig 4.5.2 Image with marking

Fig 4.5.3 Filtered Image

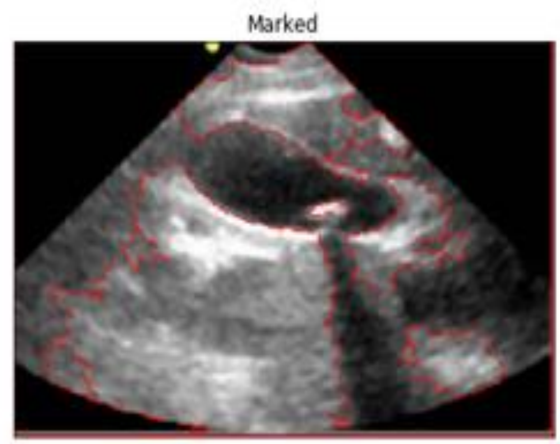


Fig 4.5.4 Marking on filtered image

Chapter-5

TESTING

A strategy for software testing integrates software test case design methods into a well-planned series of steps that result in the successful construction of software. Testing is the set of activities that can be planned in advance and conducted systematically. The underlying motivation of program testing is to affirm software quality with methods that can economically and effectively apply to both strategic to both large and small-scale systems.

5.1TEST CASES:

S.No	**Test Case Description**	**Actual Value**	**Excepted Value**	**Result**
1	Load image	Load only kidney stone image produced by ultrasound scanning	Kidney stone image is loaded successfully	True
2	Performing Histogram Equalization Algorithm	aims to enhance the contrast of an image	Image enhanced successfully	True
3	Performing watershed	find the watershed lines	Successfully shows watershed lines	True
4	Marking	Marking stones	Successfully marked the stones	True

Table 5.1: Test Cases

Chapter-6

CONCLUSION

We had an image of an ultrasound of a kidney containing a stone. We applied the Gabor Filter for the Image Enhancement followed by Histogram Equalization. The restored image went under Image Segmentation, namely, Watersheds after which Marking was done and the final image was produced. The final image showed a distinct location of the stone in the kidney. Hence the stone was detected. In the future work, the system will be designed for real time implementation by placing biomedical sensors in the abdomen region to capture kidney portion. The captured kidney image is subjected to the proposed algorithm to process and detect stone on FPGA using hardware description language (HDL). The identified kidney stone in the image is displayed with colour for easy identification and visibility of stone in monitor.

REFERENCES

1. http://imgsimon.blogspot.com/2016/05/python-image-filtering-in-python-and.html
2. https://github.com/tvganesh/weiner
3. https://www.sciencedirect.com/science/article/pii/S0169260717300639
4. http://matlabserver.cs.rug.nl/edgedetectionweb/web/edgedetection_params.html
5. http://tudr.thapar.edu:8080/jspui/bitstream/10266/969/1/Abhay_Thesis.pdf
6. https://www.ijser.org/researchpaper/Enhancement-of-Ultrasound-Images-Using-RADWT.pdf
7. https://www.hindawi.com/journals/cmmm/2014/758439/fig1/
8. https://www.hindawi.com/journals/cmmm/2014/758439/
9. https://wiseodd.github.io/techblog/2016/11/05/levelset-method/
10. https://medium.com/nanonets/how-to-use-deep-learning-when-you-have-limited-data-part-2-data-augmentation-c26971dc8ced
11. https://ieeexplore.ieee.org/document/6572601
12. https://onlinelibrary.wiley.com/doi/pdf/10.7863/jum.1984.3.3.123
13. https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=802756
14. https://www.youtube.com/watch?v=7Nxpa0cjOBE
15. https://www.hindawi.com/journals/vlsi/2015/581961/
16. https://opencv-python-tutroals.readthedocs.io/en/latest/py_tutorials/py_imgproc/py_watershed/py_watershed.html