CHAPTER 01

1. Introduction

The doctor often utilizes the hand approach to find the stone in the X-ray picture, but our method is totally automatic, which saves time and reduces the risks of making a mistake. This project describes a method for detecting kidney stones that includes image processing processes. The basic function of the kidney is to maintain electrolyte balance in the blood. A bean-shaped organ, the kidney can be found on either side of the spine. Kidney stones are becoming more widespread around the world, and most people with concretion illness aren't aware of it because it destroys organs gradually before symptoms appear. There were 16000 papers on detecting kidney stones using various types of filters in Google Scholar and IEEE Xplore in the previous 5 years. To locate photos with poor contrast and speckle noise, digital image processing is performed. An automated kidney stone detection system is built using digital imaging and data processing approaches. There is a great deal of noise on CT scans and MRIs, which beneficial to low accuracy. Artificial intelligence methods based on neural networks have shown significant results. Artificial intelligence, such as neural networks, has shown to be quite helpful in this area. As a result, in this project, the Digital Image Processing is being used on ultrasound sound images. Image - processing applications include image sharpening and restoration, colour processing, and pattern recognition. Low contrast and speckle noise in ultrasound images are drawbacks. The detection of kidney stones is a difficult task. Speckle noise is a natural characteristic of medical ultrasound imaging, and it reduces image resolution and contrast, lowering the diagnostic usefulness of the imaging modality. The kidney stone is found using ultrasound images and then removed with a surgical process that includes breaking up the stone into smaller bits that can be passed down the urinary tract. Because kidney stones have low contrast and speckle noise, detecting them is a difficult task. This problem is solved by employing the appropriate imaging techniques and filters. Speckle noise is common in ultrasound images, which cannot be eliminated with standard filters. As a result, the median filtering algorithm is proposed, which reduces speckle noise. A median filter is used to eliminate noise and detect the stone region in the pre-processed image.

2. Objective of the Major Project

Kidney stones is difficult to track early because of hidden symptoms. There is a huge burden on patients and the resilience system due to delays in diagnosis. The problem of kidney stone detection has encouraged researchers to expand their research in order to track and detect this problem at an early stage for the people with kidney stones and medical treatment should be given on time to avoid any further complications. We have used Ultrasound images to detect the kidney stones and applied techniques to predict accurate results. The reason why we collected the dataset of ultrasound images rather than CT scan images(which is widely used in the healthcare domain to predict results) is that we want to create an user friendly project which can be further implemented and can be used in the daily life by the healthcare system to generate and track the accuracy of the results within the short spare of time and further creating an ease to both healthcare system and to the common people with the disease.

3. Technical Requirements (Software) (1) MATLAB (a) For Implementation (2) MS-EXCEL (a) For Comparing Accuracy and Senstivity 4. Technical Requirements (Hardware)

As we will be working with

MATLAB

The Minimal System Specifications are:

- Memory: 4 GB
- Free Storage: 2 GB
- Screen Resolution: 1200 x 800
- OS: Windows 7/8/8.1/10 (64-bit)

The suggested system requirements are

- Memory: 8 GB RAM
- Free Storage: 4 GB (SSD Recommended)
- Screen Resolution: 1920 x 800
- OS: Windows 10 (64-bit)
- CPU: Intel Core i5-8400 3.0 GHz or better.

4. Deliverables of the Major Project

The project will predict whether the person is or is not having Kidney stones inside his/her kidney. Using the given data set we will analyze data using Image Processing Techniques (Using of Negative Image Filter, High-Low Contrast Filter, Edge Detecting filters and so on). Our goal is to attain Accuracy and sensitivity atleast more than the current methodologies existing in the healthcare industry so that this model can further be used and can be implemented in the real world and which makes ease to both the patient(s) and medical practitioner(s). We would also be looking forward to compare our approaches taken to finish this project and will try to obtain and predict which approach gives us the better results output based on numeric data

CHAPTER 02:

LITERATURE SURVEY

1. Existing approaches and recent research on the similar kind of project:

No.	References' titles	Algorithm/Techniques	Significance	Measuring
1	A method for segmenting kidney stones in CT scan pictures using image preprocessing	Threshold Technique: In this we convert a picture from colour or grayscale to binary by changing pixel of image to make analysis easy.	CT scan image segmentation preprocessing	Precision Accuracy Sensitivity
2	SVM and KNN are used to analyse and identify kidney stones.	K Nearest Neighbour & Support Vector Machine	Image pixel count and stone detection in kidney pictures	Recall Precision
3	A study of a systemic strategy to utilise ultrasound images to detect renal calculi using several analysis methods.	K Nearest Neighbor: It finds the most common class among the k-closest instances to classify unknown data items.	Various processing techniques are used to segment data.	Accuracy
4	Introduction using GLCM and pixel intensity matrix parameters to differentiate staghorn and struvite kidney stones	Gray Level Co-occurrence Matrix Future Extraction	The image is enhanced and the low parts are filtered.	Precision Accuracy Sensitivity
5	Detection of kidney stones in computed tomography images	Segmentation	The quantity and placement of the stones can be determined with the use of segmentation.	Precision Accuracy Sensitivity

6	Kidney Stone Detection in Ultrasound Images Using the Median Filter vs. the Rank Filter	Image Processing Techniques	Differentiation of filters using ultrasound images	Accuracy & Sensitivity

Table 1 Existing approaches and recent research

We have researched the above techniques/approaches to implement our project after analyzing existing techniques/approach in the healthcare and technology domain. On the basis of our research, we have created our own new approach taking references from our research part and successfully implemented our project and later we have compared it with the existing technology working on similar kind of projects.

2. Feasibility Study on Major Project:

This project predicts the presence of kidney stones inside an individual's kidney using ultrasound pictures. One of the imaging approaches for identifying kidney problems is ultrasound imaging, which can include changes in terms of size and placement as well as swelling of the limb. Other kidney abnormalities include the formation of stones, cysts, urine blockage, congenital anomalies, and cancerous cells. It is critical to detect the real and accurate position of a kidney stone during surgical procedures because the contrast is low and speckle noise is present , detecting them using ultrasound imaging is a difficult process. This problem may be solved by using the right image processing techniques.

3. Requirements on Major Project:

1. **<u>Functional Requirements</u>**

   MATLAB

   - ➢ Microsoft Windows 10
   - ➢ Minimal: Any Intel or AMD x86-64 processor.
   - ➢ Suggested: Any Intel or AMD x86-64 processor with four logical cores and AVX2 instruction set support

- Minimal: 3 GB of HDD space for MATLAB only, 5-8 GB for a typical installation.
- Suggested: An SSD is recommended.

MS-Excel

- Microsoft Windows 10
- The configuration required for the MATLAB would be enough to run and process MS-office.

2. Non-Functional Requirements

ATLEAST: 4 GB memory

- Intel 8th generation higher CPU's
- Nvidia 1650 min GPU

4. Objective of Major Project:

The goal of this project is to detect kidney stones using a dataset of ultrasound pictures (which includes 50% of images with kidney stones and 50% of images without kidney stones). This research will be undertaken using digital image processing techniques such as applying various filters to the dataset, and it will also be based on research acquired from other sources. This project has been separated into two sections, with Part 1 being the Median Filter and Part 2 being the Edge Detection Filter. To simulate and develop the code, we utilized MATLAB. We will compare the produced pictures on MATLAB to track the accuracy and sensitivity obtained after successfully acquiring the results from both filters on the dataset image. In addition, we will compare the findings in MS-Excel.

Once the code has been processed, the output picture will be seen in the command prompt. To acquire the result, we will carry out the experiment again with distinct kidney ultrasound pictures and use the algorithm to compute the rate of identification.

The input photos, which are independent variables, are ultrasound scans of kidney stones. Accuracy and sensitivity will be the output variables.

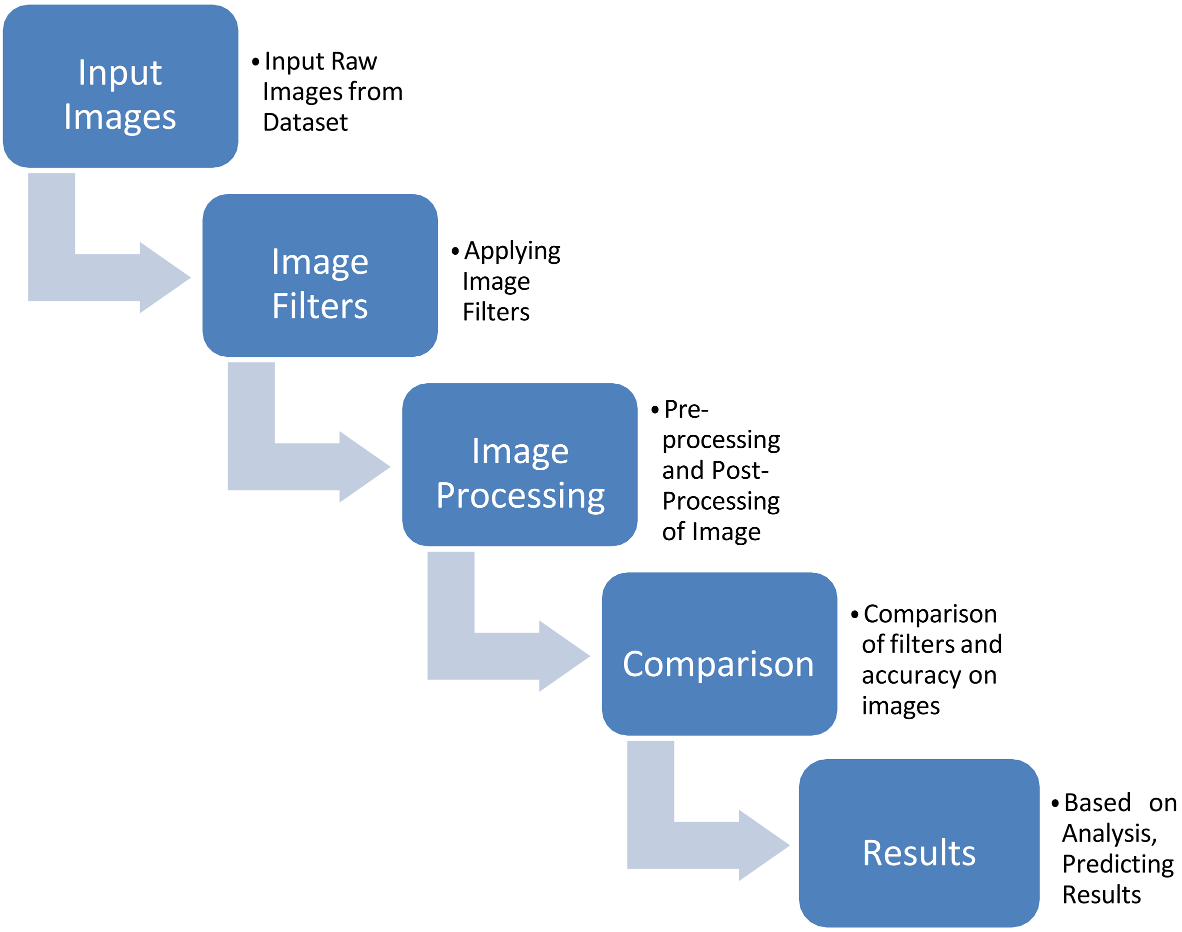


Fig. 2.1 proposed method

CHAPTER 03

System Development

3. Implementation of the Major Project

3.1 Dataset Used in the Major Project

This database contains a large number of biomedical ultrasound images from various hospitals of USA. There are total 228 Ultrasound images with 50% images that can have a possibility of kidney stones and rest 50% normal images without kidney stones. Following are the few images from the dataset which we will use.

Fig 3.1. Kidney stone detected image

3.2 Date Set Features

3.2.1 Types of Data Set

There is total 228 raw Ultrasound images taken from Kaggle as well as few from some Hospital's website to do this project. Every raw image before processi'ng has nil signal to noise ratio until some filter has been applied on it. All of the dataset is somewhat greyscale images and all of them have properties of ultrasound images. The dataset used in this project have been collected from different sources both available online and offline from hospitals/clinics.

3.2 Design of Problem Statement

More than half a million patients visit emergency departments each year with kidney stone issues. A kidney stone affects one out of every 10 people at some point in their lives. In the United States, kidney stone prevalence increased from 3.8 percent in the late 1970s to 8.8 percent in the late 2000s. In the years 2013–2014, 10% of people had kidney stones. Men have an 11 percent chance of developing kidney stones, while women have a 9 percent chance. High blood pressure, diabetes, and obesity are all conditions that increase the risk of kidney stones. Kidney stone disease is a serious health problem that affects people all over the world. Stone disorders go undiagnosed in the early stages, causing harm to the kidneys as they progress. Kidney failure affects the majority of individuals owing to diabetes, hypertension, glomerulonephritis, and other factors. Because kidney disease can be dangerous, it's best to get a diagnosis as soon as possible. Ultrasound (US) is one of the most often utilized noninvasive, low-cost imaging modalities for evaluating renal diseases. Kidney failure is a potentially fatal condition. Therefore, early diagnosis of kidney stones is crucial. To assure the accuracy of medical procedures, it is necessary to precisely detect kidney stones. The ultrasound images of the kidney contain speckle noise and weak contrast, making kidney abnormalities difficult to recognize. As a result, doctors may find it tough and confusing to detect small kidney stones and their nature.

3.3 Algorithm / Pseudocode of the Project Problem with the screenshots of various stages of the project

1. Creating A Graphical User Interface (GUI Model) on MATLAB

App Designer is an interactive environment that combines the two core aspects of app development: visual component layout and behaviour scripting. It enables you to switch between visual design on the canvas and coding in the MATLAB editor easily. With the help of App Designer we have created an app environment to run our project. Following images will show the step by step procedure how we have created an app environment for our project.

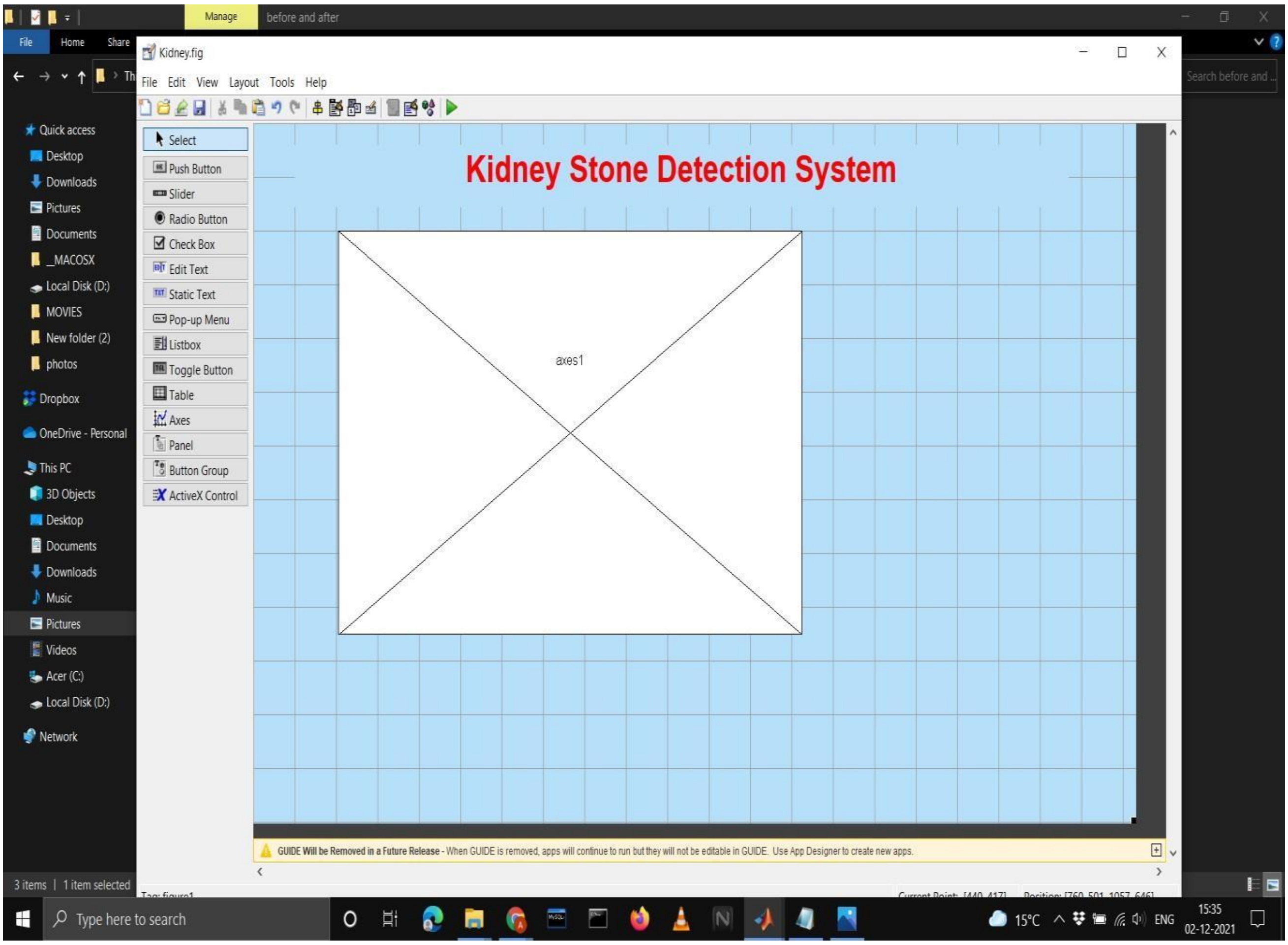


Fig 3.2 GUI Model

STEP 1

Created a GUI where we can implement our project

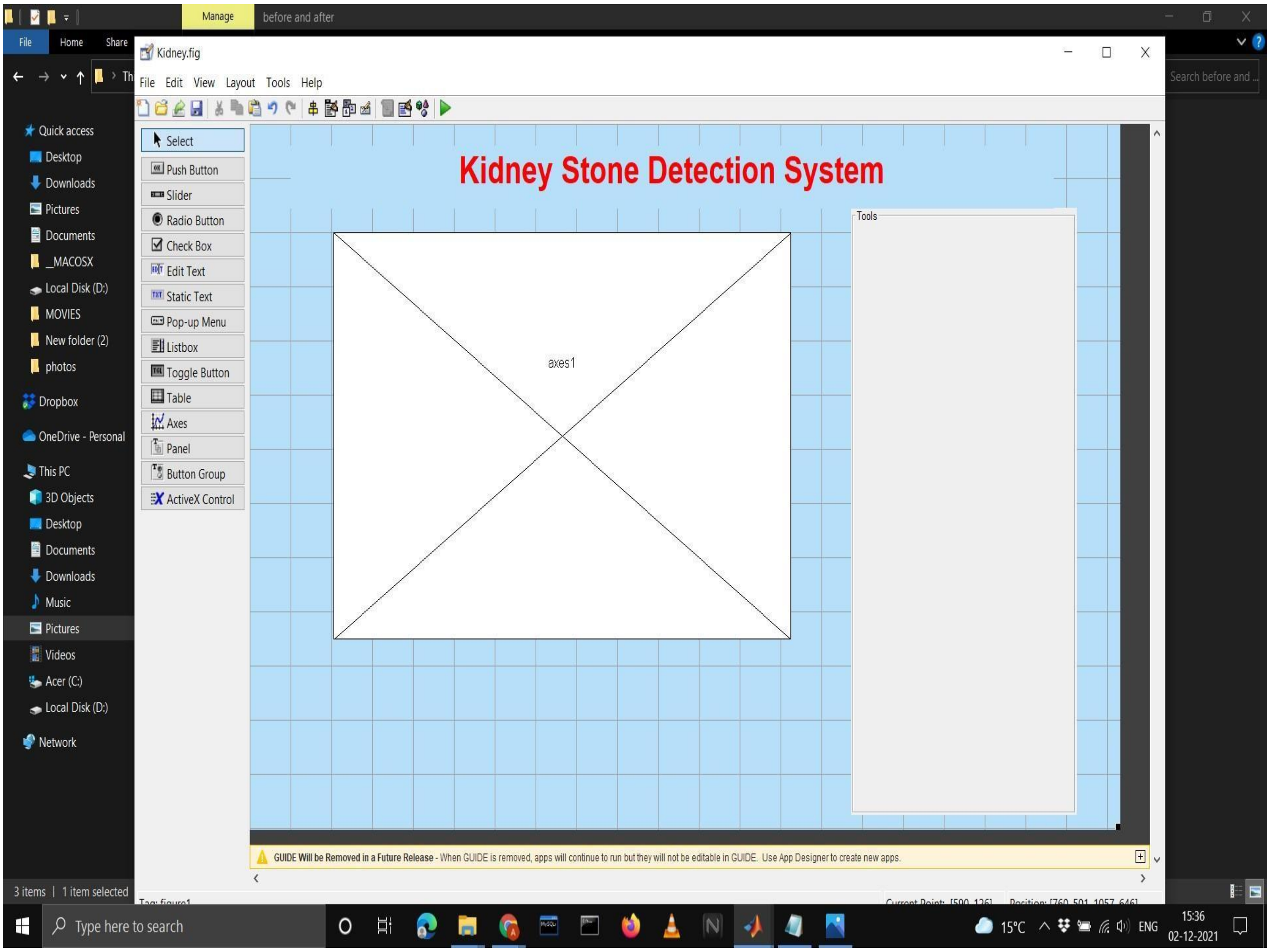


Fig 3.3.GUI model implementation

STEP 2: Created a separate toolbox where we can draw filter buttons and other Required buttons

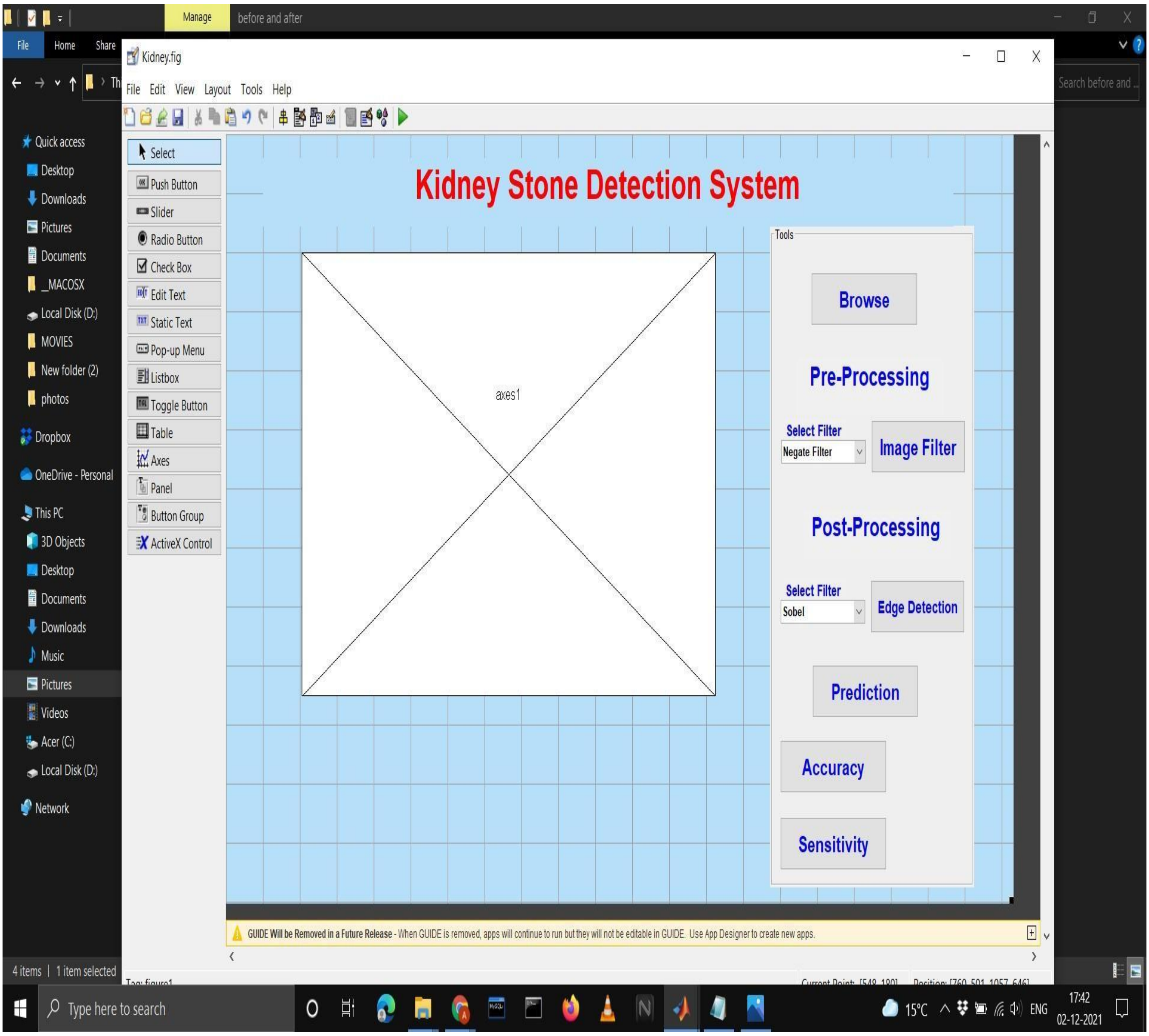


Fig 3.4.Button Creation

STEP 3: Created buttons using designer toolkit in the tools section that we have created in the previous Step

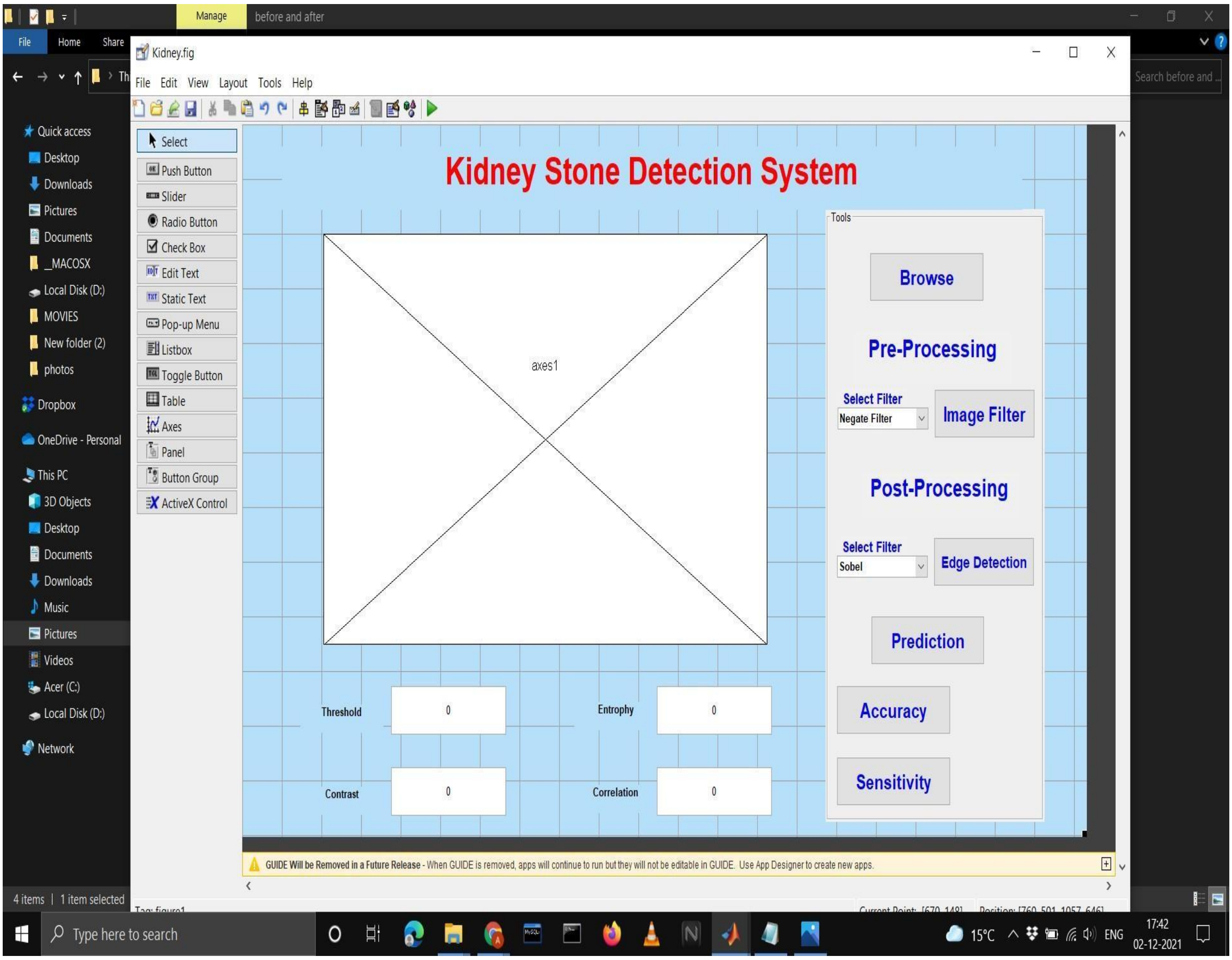


Fig 3.5 Impementing 4 button on GUI

STEP 4: To create an user friendly environment we have further created 4 buttons where we can also take input from the user and accordingly the prediction will be made

2. Explanations of filters used in the project

A. MEDIAN FILTER

Median filtering is a nonlinear method for minimising impulsive noise, often known as salt-and-pepper noise. It may also be used to keep an image's edges while decreasing random noise. A random bit mistake in a communication line can cause impulsive or salt-and-pepper noise. The median intensity value of the pixels within the window becomes the output intensity of the pixel being processed in a median filter, which slides along the picture. Assume that the pixel values within a window are 5,6, 55, 10, and 15, and that the pixel being processed is 55. The median filter produces a value of 10 at the current pixel position, which is the median of the five values. The median filter is a non-linear digital noise reduction approach for images and data. Noise reduction is a common pre-processing procedure used to improve the outcomes of subsequent processing (for example, edge detection on an image). Median filtering is commonly employed in digital image processing because it retains edges while reducing noise under certain conditions (though see the explanation below), and it also has uses in signal processing. The median filter's fundamental principle is to go over the signal entry by entry, replacing each one with the median of the entries next to it. The "window" is a pattern of neighbours that glides over the entire signal, entry by entry. The simplest evident window for one-dimensional signals is merely the first few preceding and following entries, but the window for two-dimensional (or higher-dimensional) data must encompass all entries within a certain radius or ellipsoidal region (i.e. the median filter is not a separable filter).

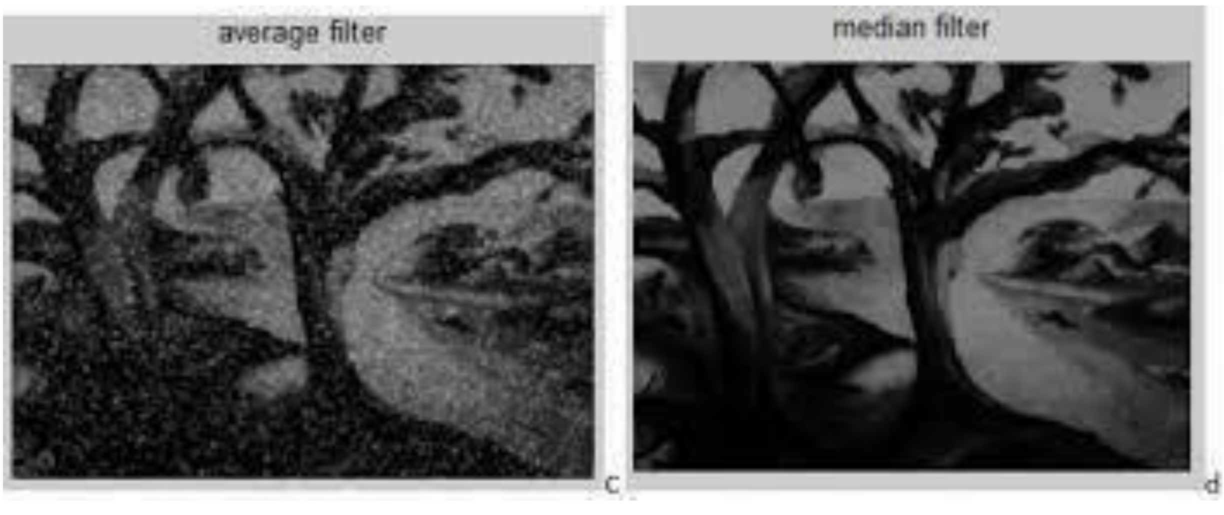


Fig 3.6 Median Filtered Images

Median Filtered Images

B. Negative Filter

The original picture's intensity I is replaced with 'i-1,' resulting in the darkest pixels becoming the brightest and the brightest pixels becoming the darkest. By removing each pixel from the greatest intensity value, an image negative is created.

An 8-bit grayscale picture's highest intensity value, for instance, is 255, so each pixel is subtracted from that value to get the final/result image. An image's negative is one in which the lightest portions appear to be the darkest and the darkest areas appear to be the lightest.

In a grey scale picture, the change in pixel intensity values from highest to lowest and lowest to highest causes the appearance shift from lightest to darkest and darkest to lightest..

Different colours are represented as negatives of different colours in a colour picture based on their intensity values.

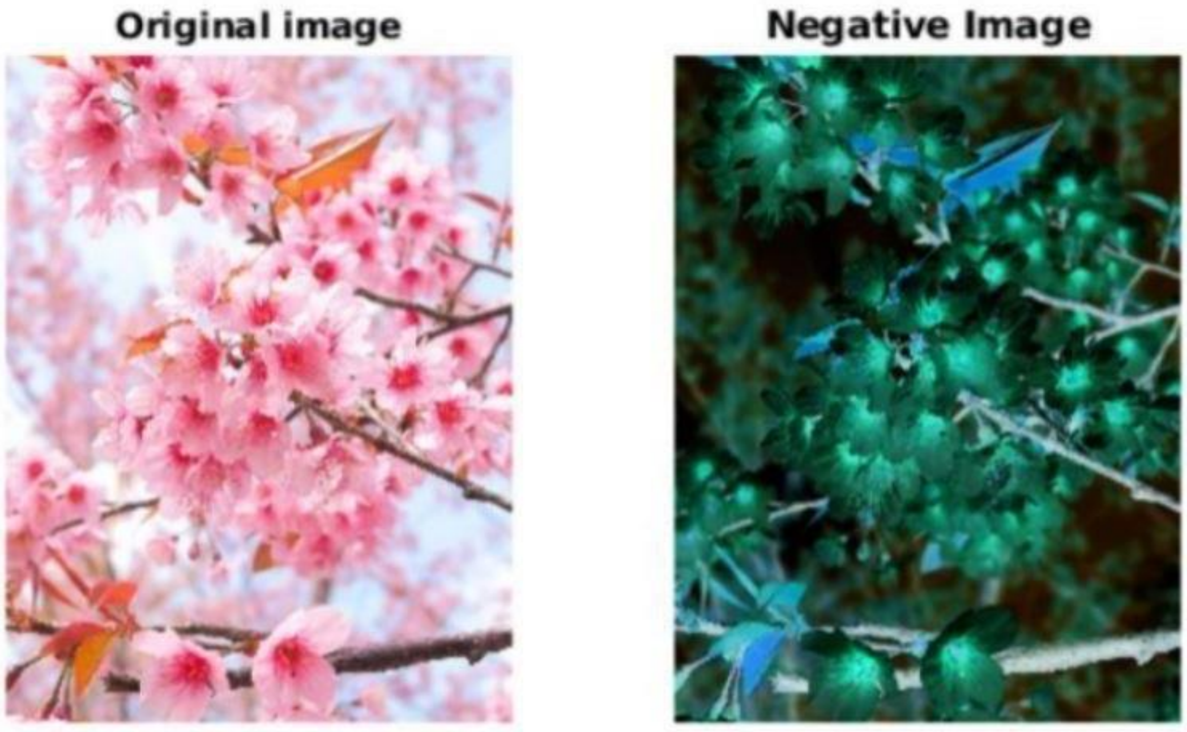


Fig 3.7 Negate filter images

C. Prewitt Edge Detection Filter

The Prewitt operator extensively used in edge detection approaches in processing of images. It is a separate differentiation filter that calculates a rough estimate of the gradient of the picture intensity function. At each point in the image, the filter returns either the relevant gradient vector or the norm of this vector. It is a gradient-based operator, as the name implies. It is one of the most effective methods for determining an image's orientation and magnitude. It computes the gradient approximation of the image intensity function for image edge detection. The Prewitt operator provides either the normal to a vector or the associated gradient vector at the pixels of an image. It calculates approximations of the derivatives using two 3 x 3 kernels or masks convolved with the input picture — one for horizontal changes and one for vertical changes. When we apply this mask to an image, the vertical edges become more visible. It simply calculates the difference of pixel intensities in an edge area, similar to a first order derivate. Because the centre column is 0, it does not contain the image's original values, instead calculating the difference between the right and left pixel values around that edge. The edge intensity is increased, and the picture is boosted in comparison to the original.

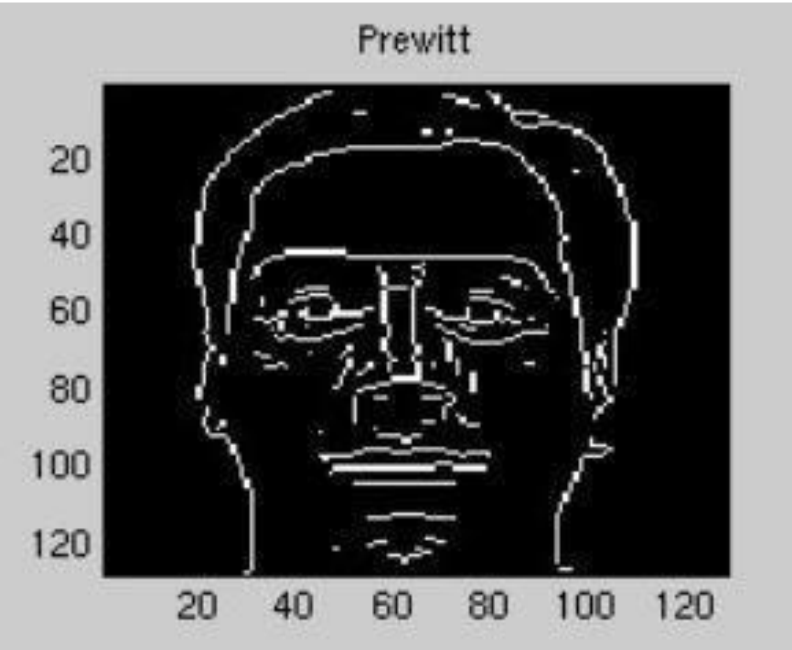


Fig 3.8. Prewitt filter image

D. Canny Edge Detection Filter

The Canny edge detector is a multi-stage edge detection operator that detects a wide variety of edges in images. It was created in 1986 by John F. Canny. Canny edge detection is a method for obtaining structural information from a variety of visual objects while drastically lowering the amount of data to be analyzed. It's been used in a variety of computer vision systems. Canny discovered that the criteria for using edge detection on a variety of vision systems are quite similar. As a result, an edge detection system that meets these requirements can be used in a variety of scenarios. The Canny filter is a multi-stage edge detector. It uses a filter based on the derivative of a Gaussian to determine the gradient intensity. The Gaussian filter reduces the influence of image noise. After that, potential edges are reduced to 1-pixel curves by eliminating non-maximum pixels of the gradient magnitude. Finally, edge pixels are retained or deleted using hysteresis thresholding on the gradient magnitude. The Gaussian width is the width of the Gaussian curve (the noisier the picture, the wider the width), as well as the low and high thresholds for hysteresis thresholding, are all customizable in the Canny.

Fig 3.9. Canny filter Image

E. Sobel Edge Detection Filter

The Sobel operator, also known as the Sobel–Feldman operator or Sobel filter, is a tool used in image processing and computer vision, particularly for edge identification. It is named after Stanford Artificial Intelligence Laboratory colleagues Irwin Sobel and Gary Feldman (SAIL). This method carry-out a 2-D spatial gradient measurement on a picture that helps in emphasizing high-frequency regions that correspond to edges. It is used to calculate the approximate absolute gradient magnitude at each point in a grayscale image input. For Edge Detection, the Sobel Filter is utilized. It calculates the picture intensity gradient at each pixel inside the image. It determines the direction and pace of change in the most significant rise from light to dark. The Prewitt operator and the Sobel operator are extremely similar. It's also a derivate mask that's utilized to identify edges. The main distinction is that with the Sobel operator, the mask coefficients are not set and may be altered according to our needs unless they break some derivative mask condition.

When we apply this mask to an image, the vertical edges become more visible. It simply calculates the difference of pixel intensities in an edge area, similar to a first order derivate. Because the center column is 0, it does not contain the image's original values, instead calculating the difference between the right and left pixel values around that edge. In addition, the first and third column's center values are 2 and -2, respectively. This gives the pixel values at the edge of the display more weight. The edge intensity is increased, and the picture is boosted in comparison to the original.

Fig 3.10 Sobel Filter Image

F.. Pre-Processing

In the pre-processing part we have used median, contrast and negate filter which will process our ultrasound images through these filters.

The following images will show us the implementation of the project-

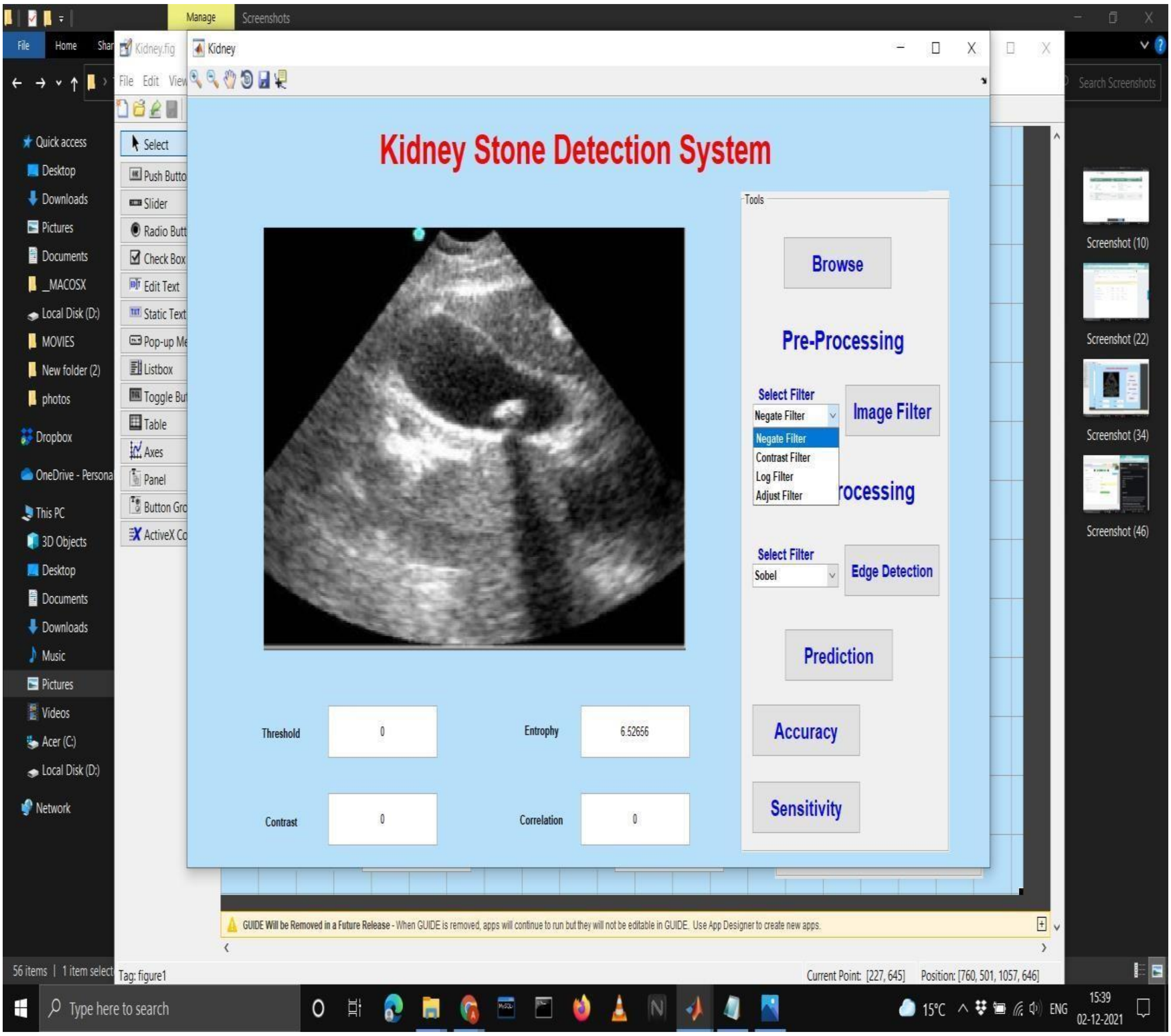


Fig 3.11 Negative Filter in Pre-processing

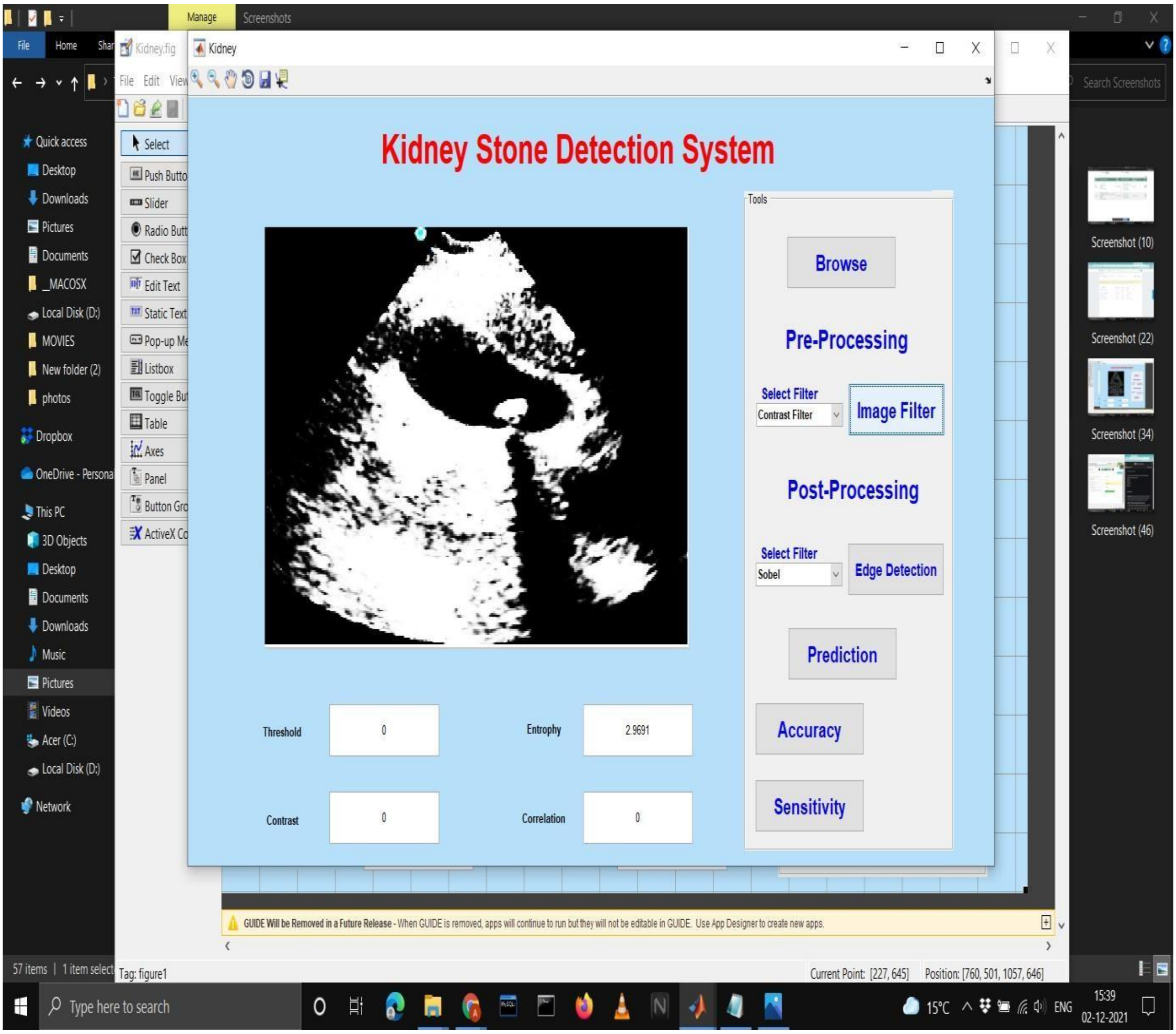


Fig 3.12 Contrast Filter in Pre-processing

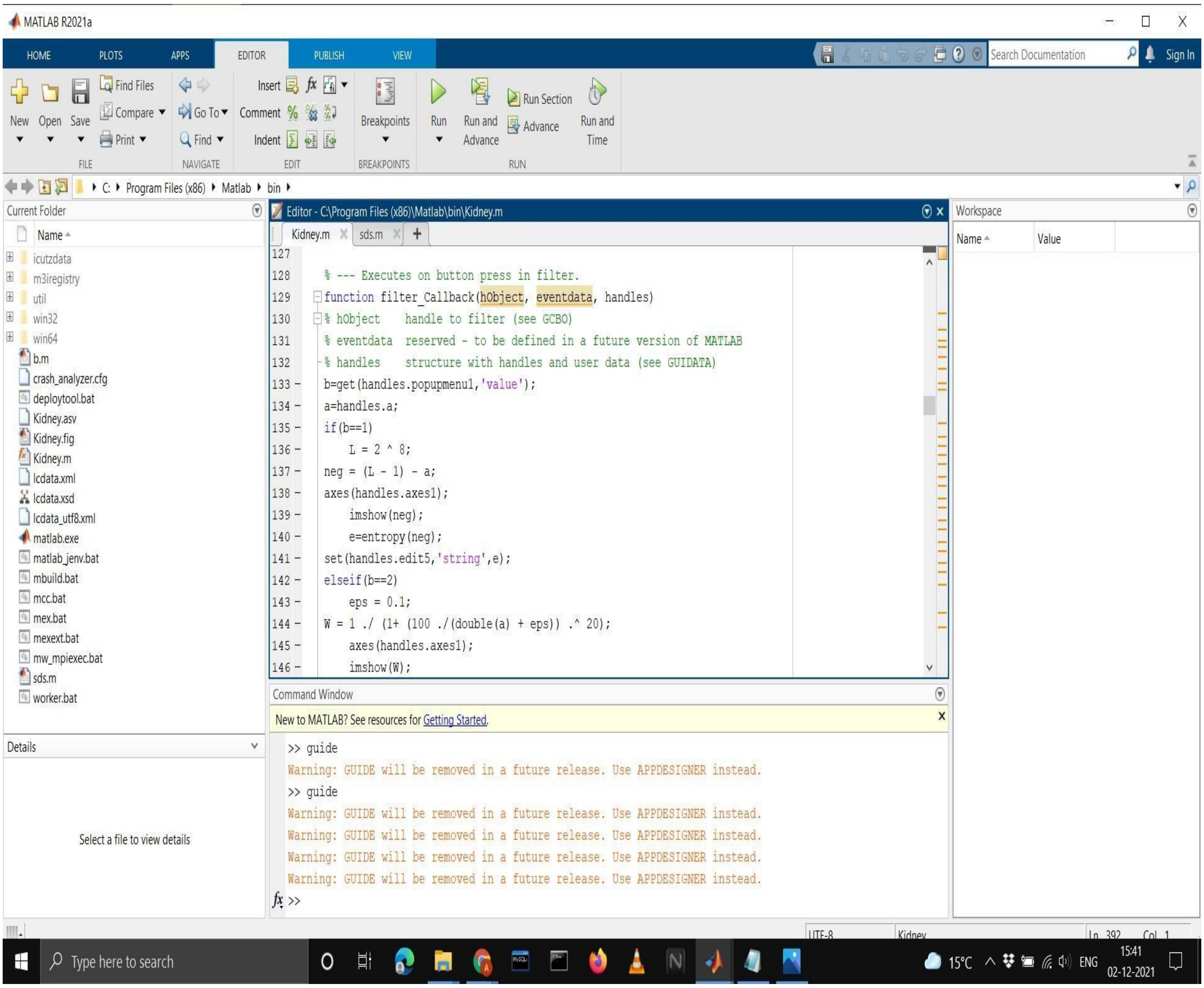


Fig 3.13. MATLAB code design in Pre-processing

Code used in MATLAB to apply filters in the Pre-Processing step

```
b=get(handles.popupmenu1,'value');
a=handles.a; if(b==1) L = 2 ^ 8; neg
= (L - 1) - a; axes(handles.axes1);
imshow(neg); e=entropy(neg);
set(handles.edit5,'string',e);
elseif(b==2) eps = 0.1;
W = 1 ./ (1+ (100 ./(double(a) + eps)) .^ 20);
axes(handles.axes1); imshow(W);
e=entropy(W);
set(handles.edit5,'string',e);
elseif(b==3) c =0.1;
V = c * log (1 + double(a));
axes(handles.axes1);
imshow(V); e=entropy(V);
set(handles.edit5,'string',e);
elseif(b==4)
Y = imadjust (a, [0 1], [1 0], .5);
axes(handles.axes1);
imshow(Y); e=entropy(Y);
set(handles.edit5,'string',e); end
```

ALGORITHM

- Load the input images from the available database.
- Applying pre-processing filters on images such as negate, contrast, median filter that are widely used in the field of image processing.
- We have then chosen/customized our filter values and their sensitivity according to our requirement.
- We have then loaded and processed our image on console and obtain the values of each image that we have generated followed by further processing of image in the post-processing step.

G. Post Processing

In the Post-Processing part we have used Edge detection filters like Sobel, Canny and Prewitt filter which will process our ultrasound images from the dataset through these edge detection filters.

The following images will show us the implementation of the project-

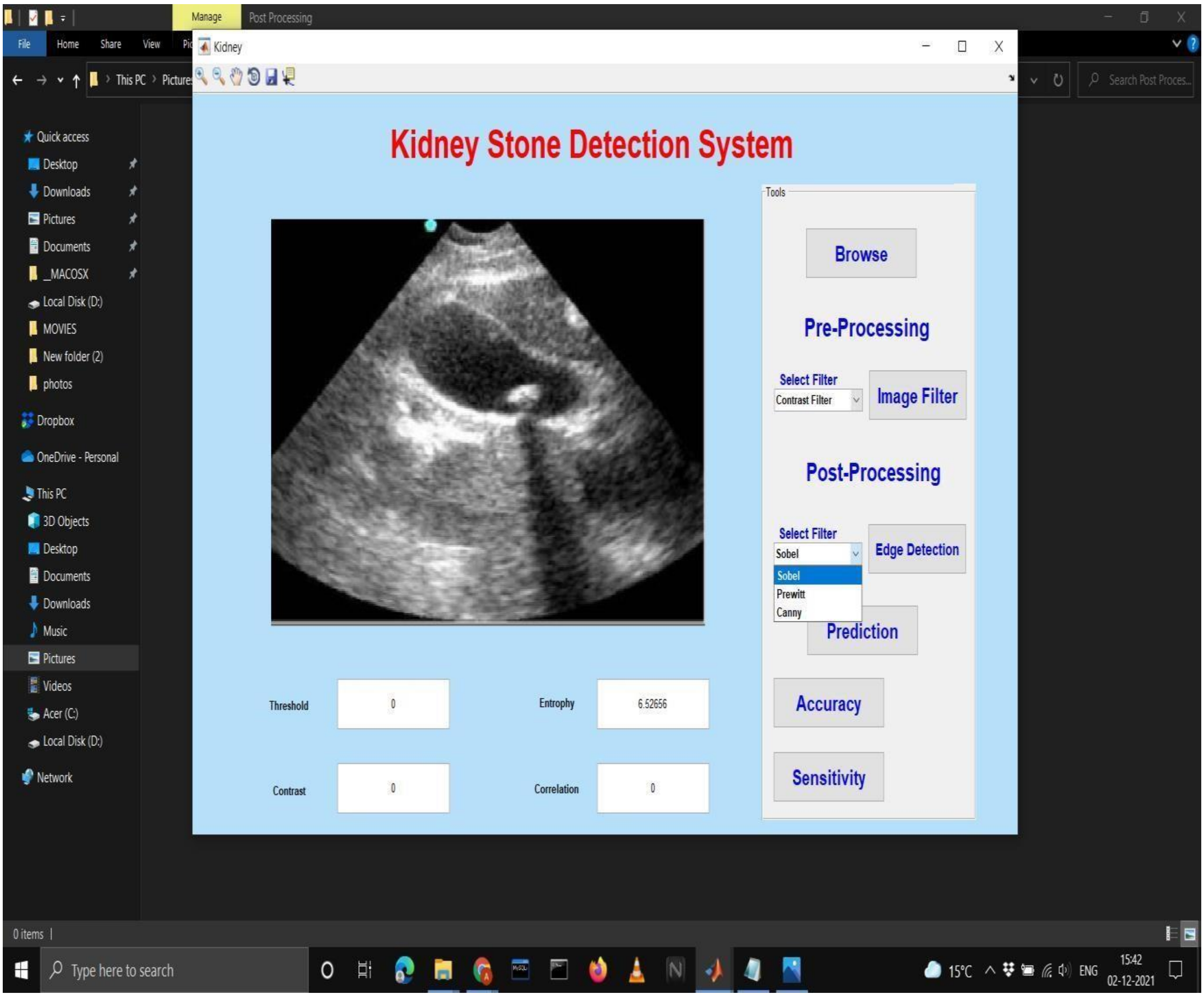


Fig 3.14 Sobel Filter in Post-processing

.

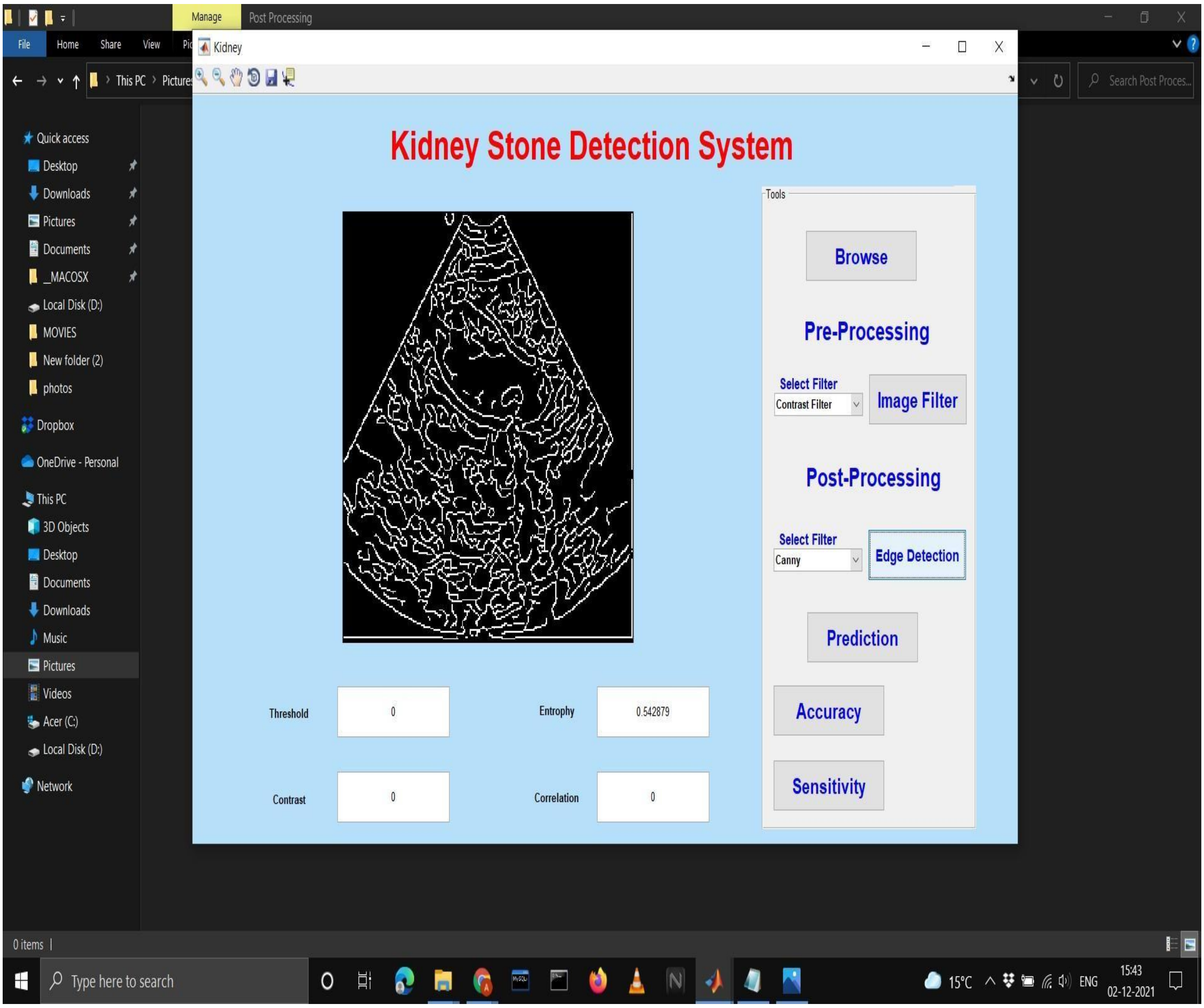


Fig 3.15. Canny Filter in Post-processing

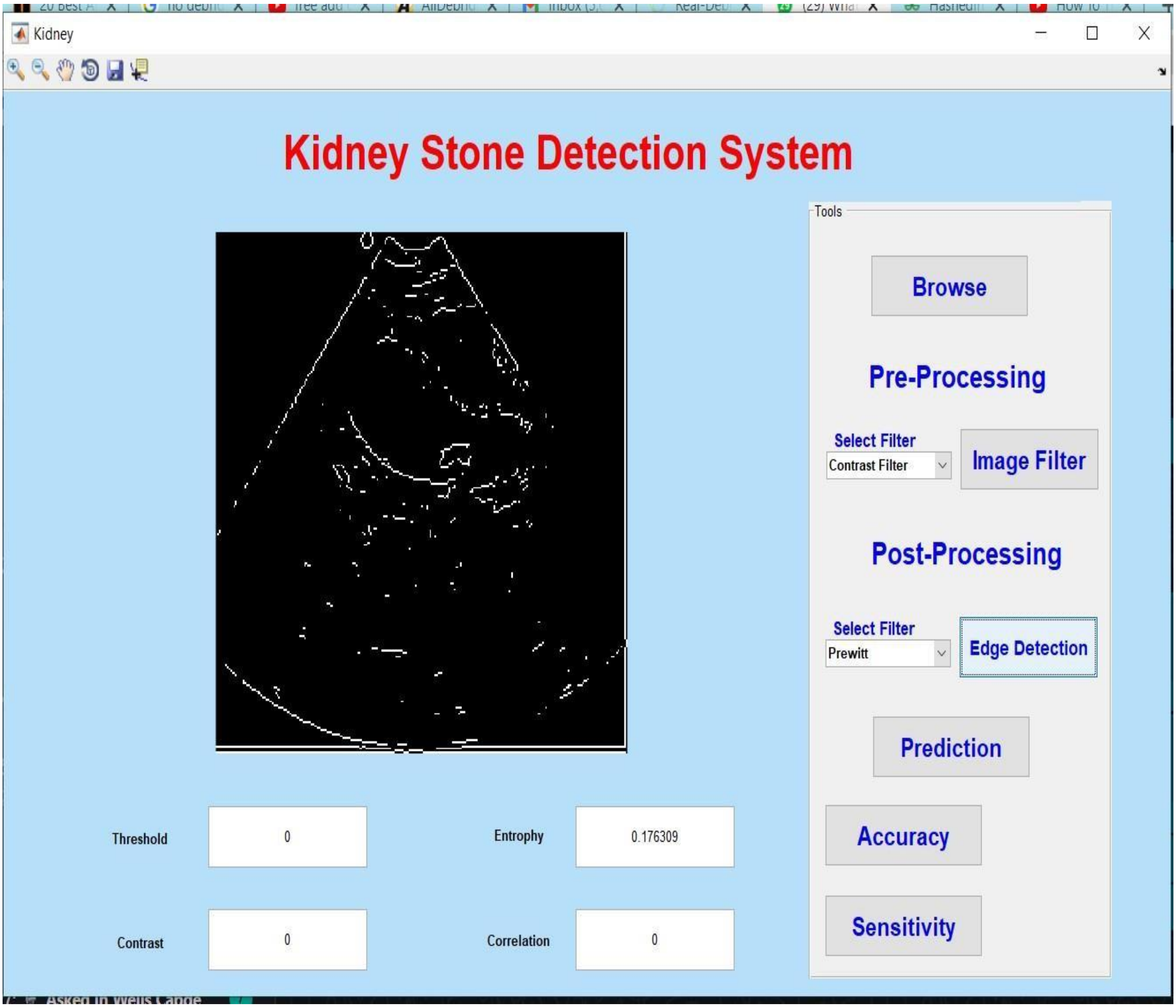


Fig 3.16. Prewitt Filter in Post-processing

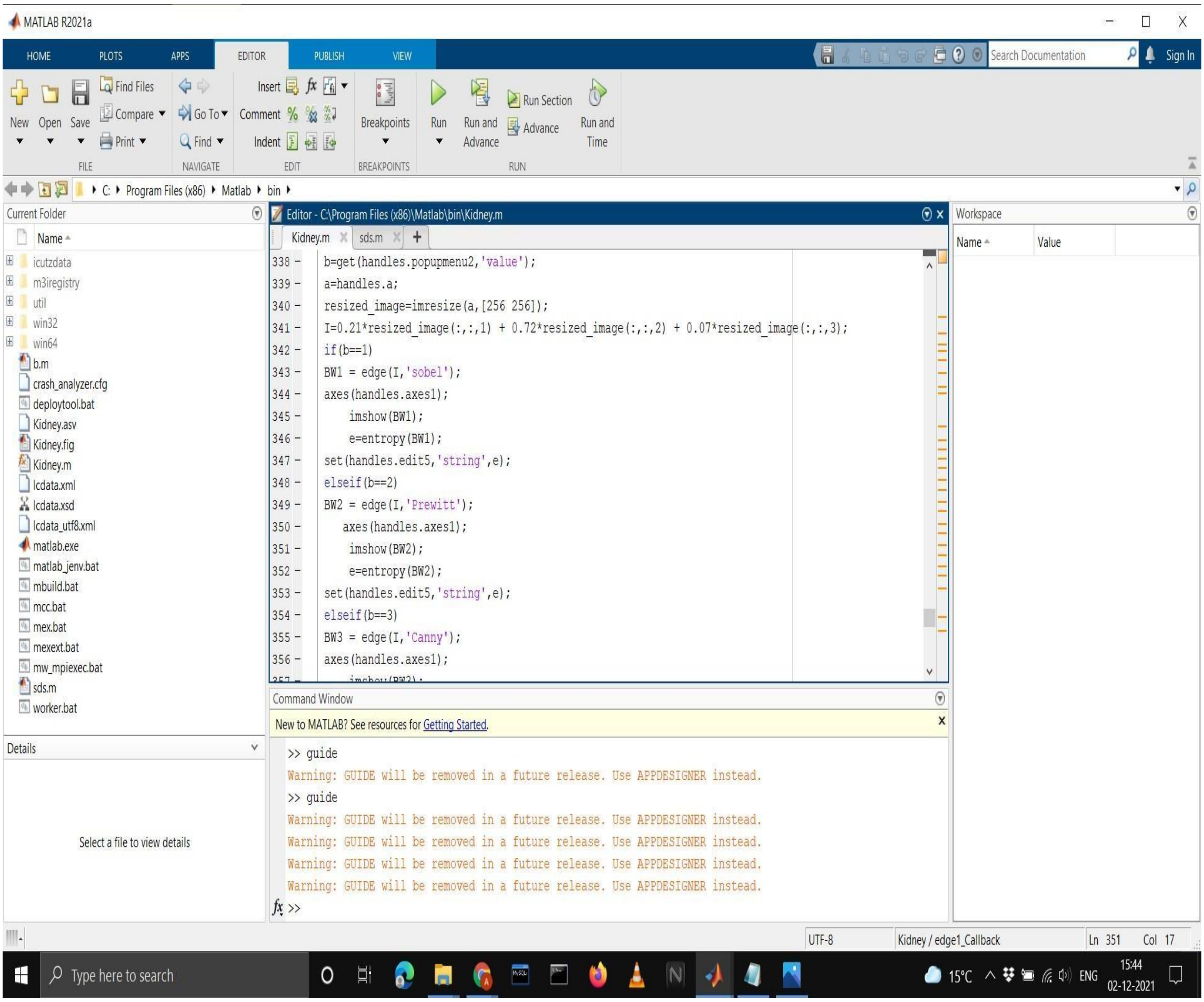


Fig 3.17 Filter applied code in Post-Processing

Code used in MATLAB to apply filters in the Post-Processing step

b=get(handles.popupmenu2,'value'); a=handles.a;

resized_image=imresize(a,[256 256]);

I=0.21*resized_image(:,:,1) + 0.72*resized_image(:,:,2) + 0.07*resized_image(:,:,3); if(b==1)

BW1 = edge(I,'sobel');

axes(handles.axes1);

```
imshow(BW1);
e=entropy(BW1);
set(handles.edit5,'string',e);
elseif(b==2)
BW2 = edge(I,'Prewitt');
axes(handles.axes1);
imshow(BW2);
e=entropy(BW2);
set(handles.edit5,'string',e);
elseif(b==3)
BW3 = edge(I,'Canny');
axes(handles.axes1);
imshow(BW3);
e=entropy(BW3);
set(handles.edit5,'string',e);
end
```

ALGORITHM

- Load the input images from the pre-processed image database.
- Applying post-processing filters on images such as sobel, canny and other edge detection customised filters that are widely used in the field of image processing.
- We have then chosen/customized our filter values and their sensitivity according to our requirement.
- We have then loaded and processed our image on console and obtain the values of each image that we have generated followed by further processing of image and then pixel values has been recorded for further analysis.

H. Prediction

Based on the pre-processing and post processing of data, we have obtained the prediction on the presence of kidney stones using ultrasound images and the prediction button will analyse our applied filters and their values passed on every pixel of the selected image from the database and if any changes in the value occurs, the system will predict the possibility of unidentified objects in the kidney using ultrasound images

The following images will show us the implementation of the project-

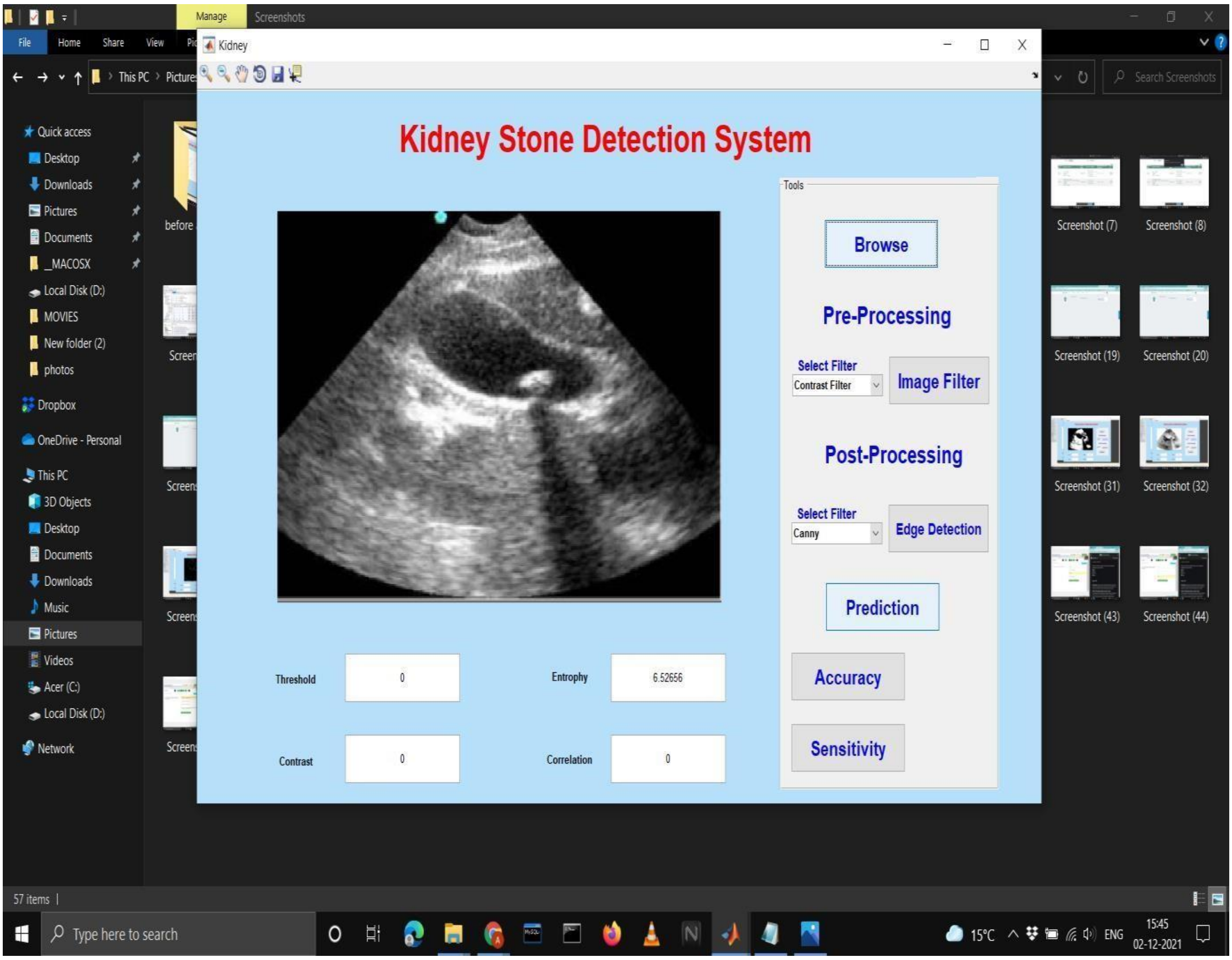
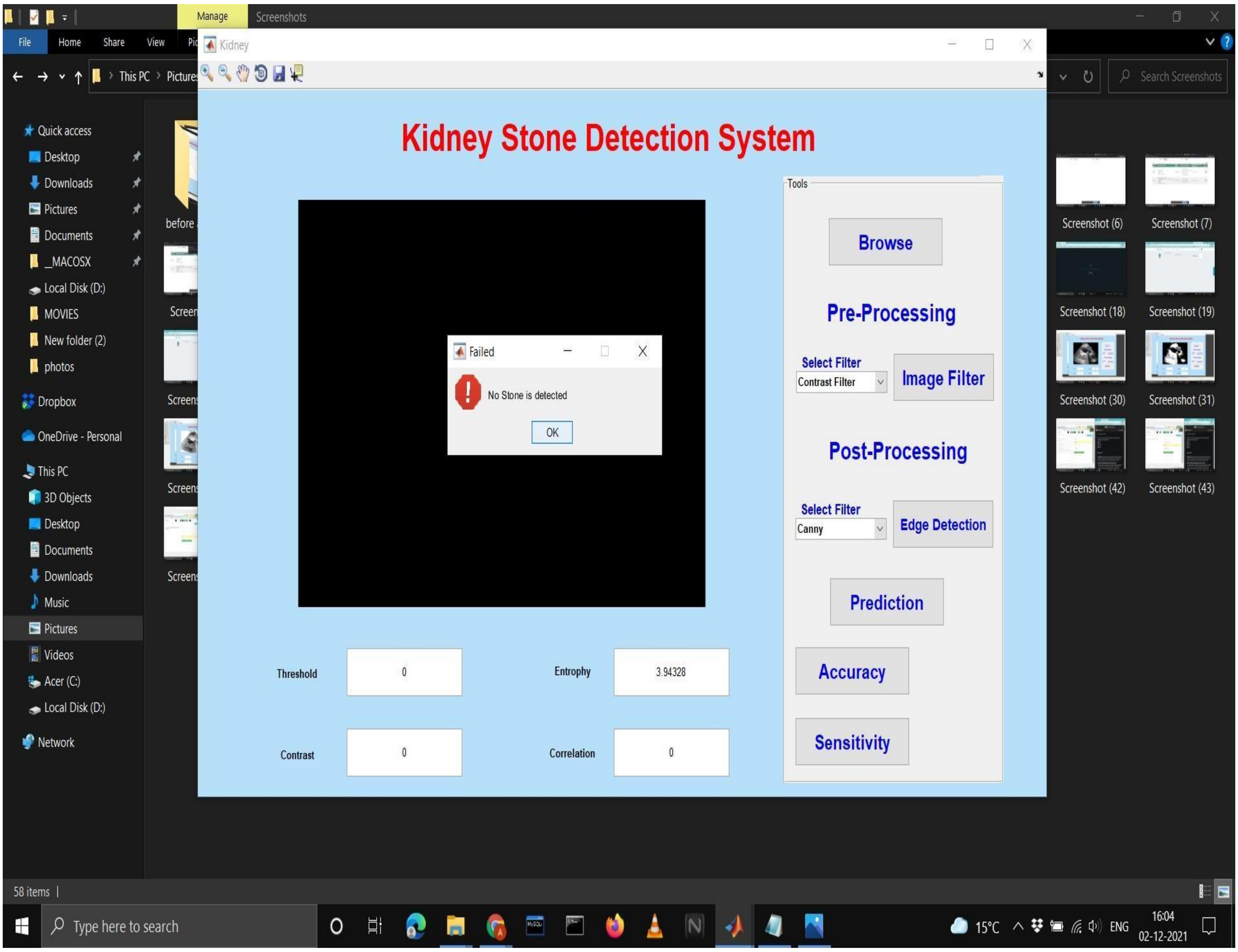


Fig 3.18 After Pre-processing and Post-Processing of the image



.

Fig 3.19 **No Stone is detected** After Pre-processing and Post-Processing of the image

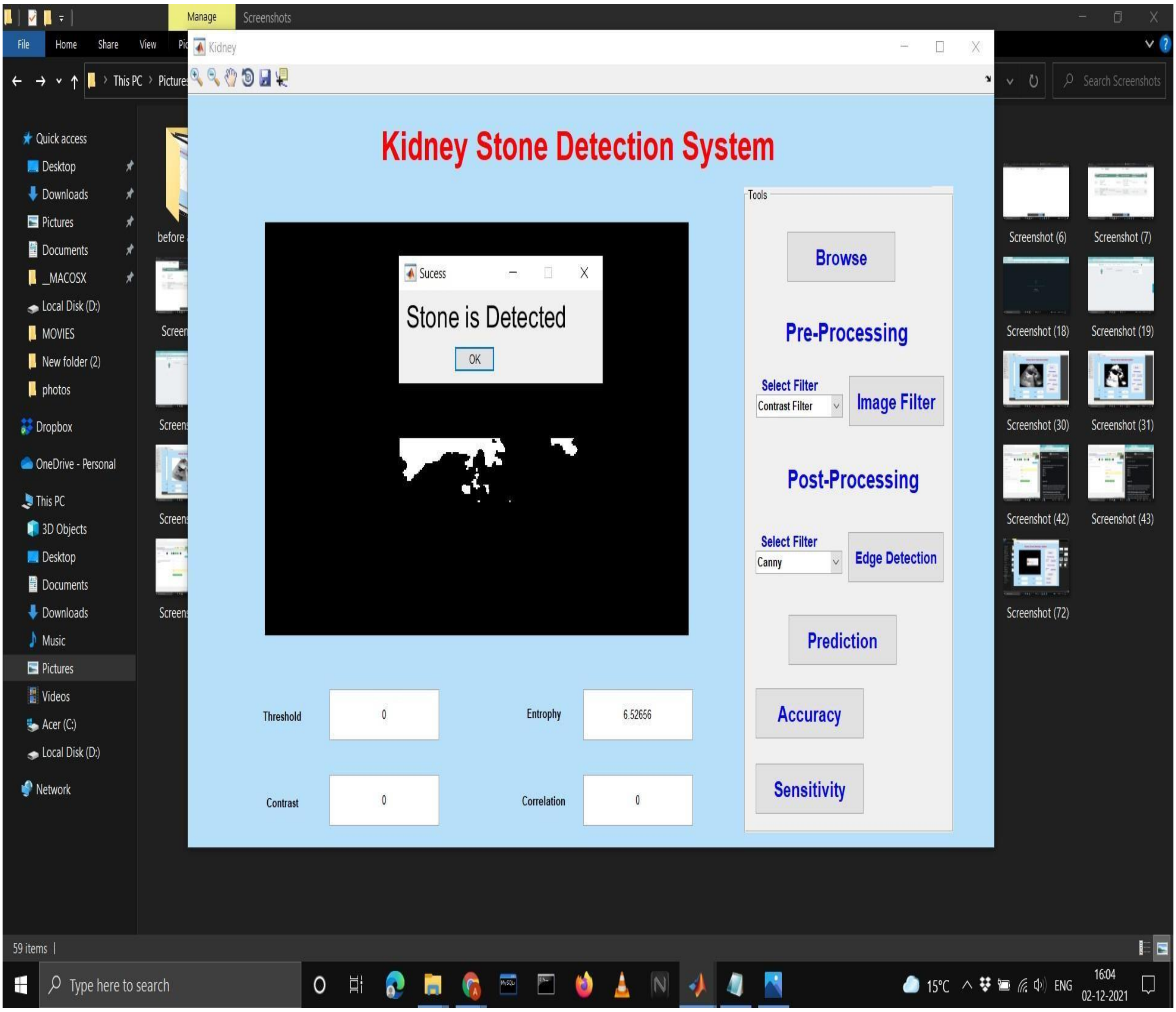


Fig 3.20 **Stone is detected** in the different image After Pre-processing and Post-Processing of the input image from database

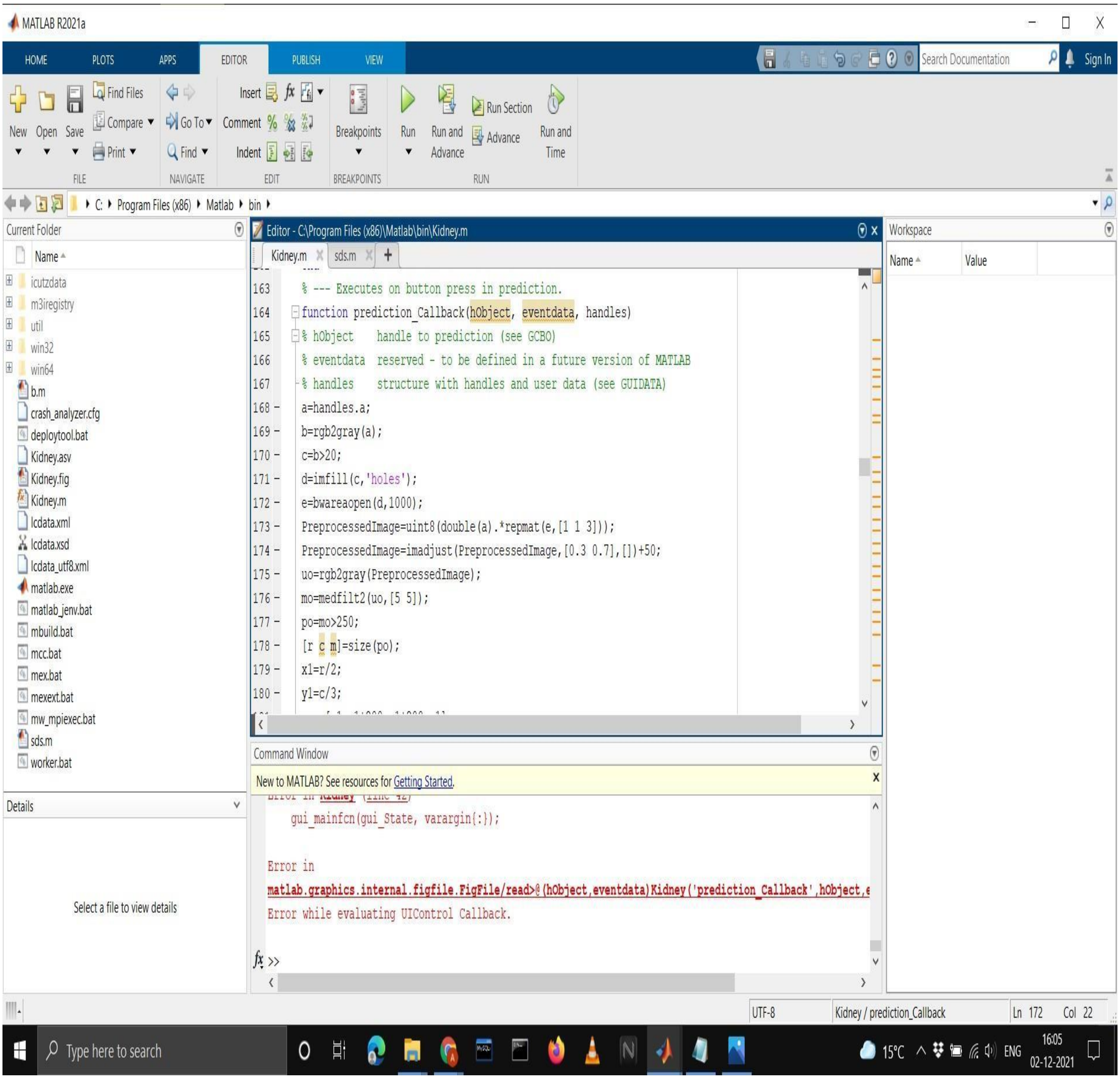


Fig. 3.21 Prediction

Code used in MATLAB for making Prediction after processing

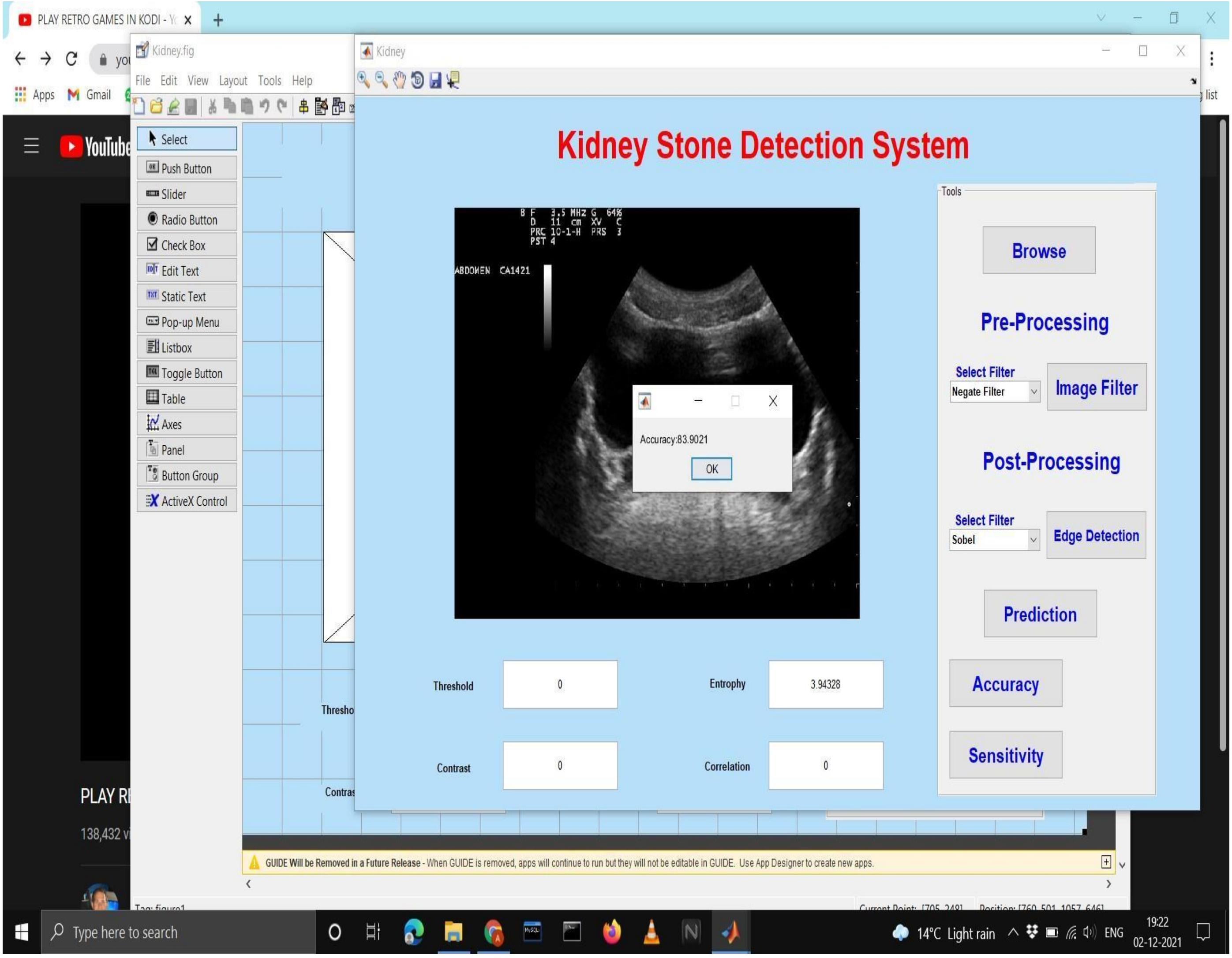
```
a=handles.a;
b=rgb2gray(a); c=b>20;
d=imfill(c,'holes'); e=bwareaopen(d,1000);
```

```
PreprocessedImage=uint8(double(a).*repmat(e,[1 1 3]));
PreprocessedImage=imadjust(PreprocessedImage,[0.3 0.7],[])+50;
uo=rgb2gray(PreprocessedImage); mo=medfilt2(uo,[5 5]);
po=mo>250;
[r c m]=size(po); x1=r/2; y1=c/3; row=[x1 x1+200 x1+200 x1]; col=[y1 y1 y1+40 y1+40];
BW=roipoly(po,row,col); k=po.*double(BW); axes(handles.axes1); imshow(k);
M=bwareaopen(k,4); [ya number]=bwlabel(M); if(number>=1)    h= msgbox('Stone is
Detected','Sucess');    object_handles = findall(h);    set(h, 'position', [200 60 200 60]);    set(
object_handles(6),'FontSize', 20, 'HorizontalAlignment', 'left','VerticalAlignment', 'bottom'); else
h=msgbox('No Stone is detected','Failed','error');     object_handles = findall(h);    set(h,
'position', [200 60 200 60]);    set( object_handles(6),'FontSize', 20, 'HorizontalAlignment',
'left','VerticalAlignment', 'bottom'); end
```

I. Accuracy

Based on the Prediction we need to find the accuracy of the system so that we could compare our results with the existing approach and find the efficiency of our project which can help us to implement this project in the real world.

The following are the images we obtained for accuracy in our project –



.

Fig 3.22 Accuracy of the particular image after prediction is 83.9021

Code used in MATLAB for Calculating and reflecting Accuracy on the Predicted Image

```
a=handles.a; b=rgb2gray(a); img=b;
img=double(img(:));
ima=max(img(:)); imi=min(img(:));
mse=std(img(:)); snr=120*log10((ima-
imi)./mse);
msgbox(sprintf('Accuracy:%0.4f',snr))
;
```

J.Sensitivity

Based on the Prediction, we need to find the sensitivity of the system so that we could compare our results with the existing approach and find the efficiency of our project which can help us to implement this project in the real world.

The following are the images we obtained for accuracy in our project –

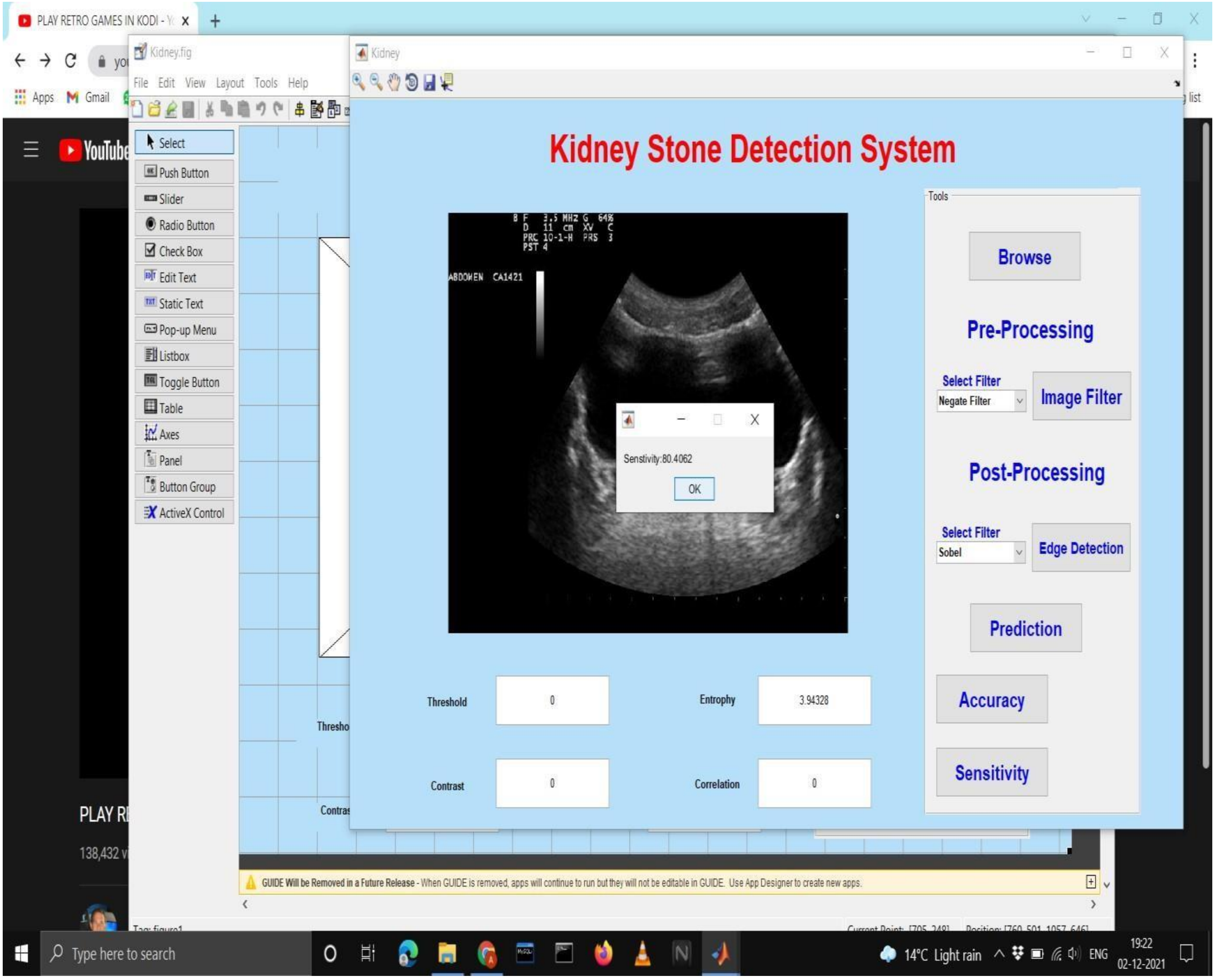


Fig 3.23. Sensitivity of the particular image after prediction is 80.4062

Code used in MATLAB for Calculating and reflecting Accuracy on the Predicted image

```
a=handles.a; b=rgb2gray(a); img=b;
img=double(img(:)); ima=max(img(:));
imi=min(img(:)); mse=std(img(:));
snr=115*log10((ima-imi)./mse);
msgbox(sprintf('Senstivity:%0.4f',snr))
;
```

CHAPTER 04

PERFORMANCE ANALYSIS

4.1 Discussion on the Results Achieved

Based on the Prediction we made in the project we also obtained the accuracy and sensitivity values for each image from the database and we have recorded randomly 10 iterations from the database to generate mathematical analysis of our project .

The following table is created using MS-Excel when we recorded data for 10 iterations that shows kidney stones in ultrasound images-

S.No	Pre-Processing		Post-processing	
Parameter	Accuracy	Senstivity	Accuracy	Senstivity
1	82.5	80.3	80.2	80
2	89.4	84.3	86.2	82.3
3	87	84.1	84	83.7
4	87.2	86.4	82.4	84.6
5	91.2	87.2	84.1	85
6	87.2	88.5	84.2	82.1
7	89.6	84	86.5	85.2
8	85.8	81.5	81.8	79.3
9	86.3	84.6	82.1	83.8

10	84.5	82.1	80.5	80.1

Table 2.

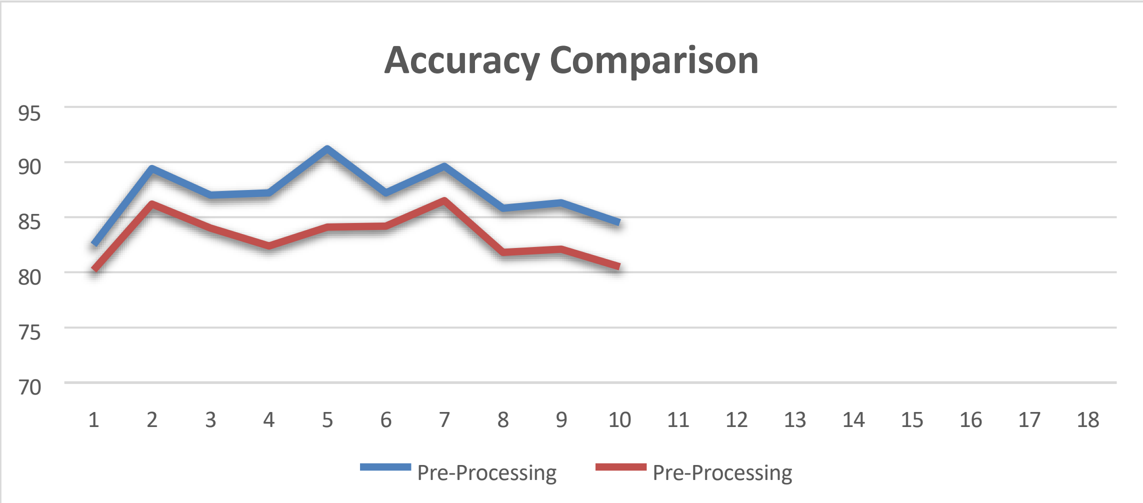
Based on the data recorded from images on MS-Excel, We further calculated the accuracy and sensitivity of our project and also a comparison between the Pre-Processing Filters and PostProcessing Filters .

The following output is received for ultrasound where kidney stone is detected –

GROUP		**No.of Samples**	**Mean**	**Std. Deviation**	**Std. Mean Error**
Accuracy	Pre-Processing	228	87.07	2.552145067	0.807059133
	Post-Processing	228	83.2	2.171532997	0.686699028
Sensititvity	Pre-Processing	228	84.3	2.568181371	0.812130258
	Post-Processing	228	82.61	2.196183558	0.69449422

Table 3.

The following graphs compares the accuracy and sensitivity of the Pre-Processing and Post-Processing based on 10 iterations recorded of ultrasound images where kidney stones are detected



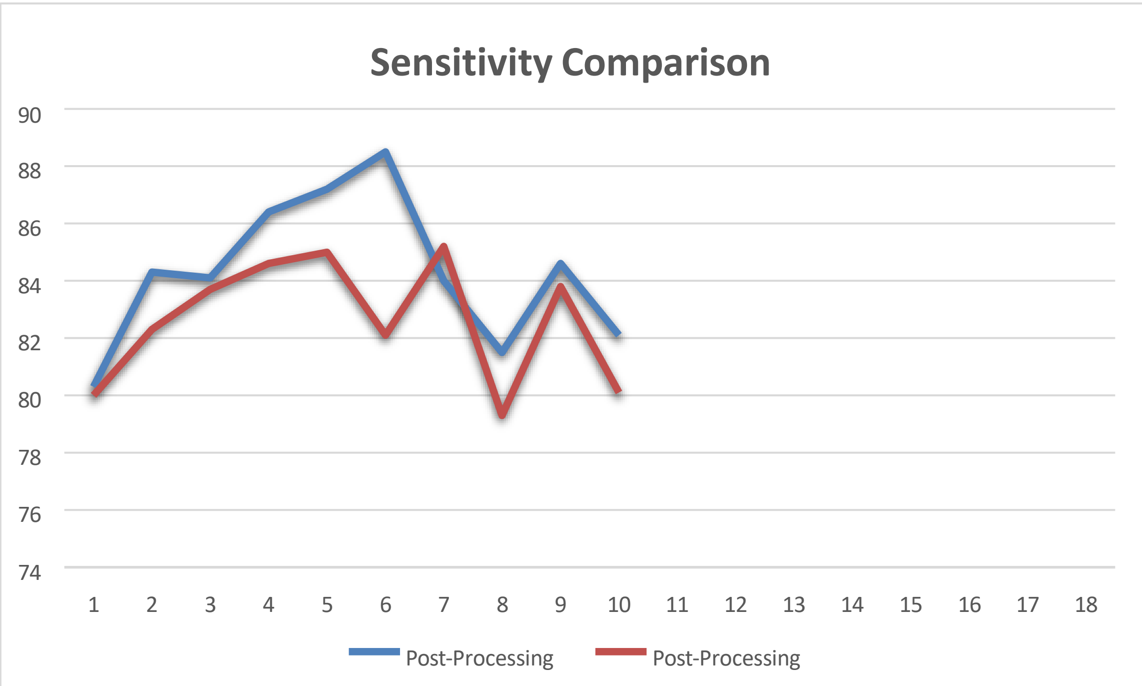


Fig 4.1.
Graphs representing Accuracy and sensitivity

The following table is created using MS-Excel when we recorded data for 10 iterations that does not detected kidney stones in ultrasound images-

S.No	Pre-Processing		Post-processing	
Parameter	Accuracy	Senstivity	Accuracy	Senstivity
1	84.5	84.7	80.2	85.2
2	88.4	84.3	85	88.6
3	82	84.1	84.2	84.3
4	86.4	86.4	84.2	84.6
5	81.1	83.1	85.3	85
6	87.2	84.6	88	82.1
7	89.6	86.2	84.9	85.2
8	85.7	88.9	86.5	79.3
9	88.1	84.6	87.8	76.2
10	89.1	83	84.7	88.1

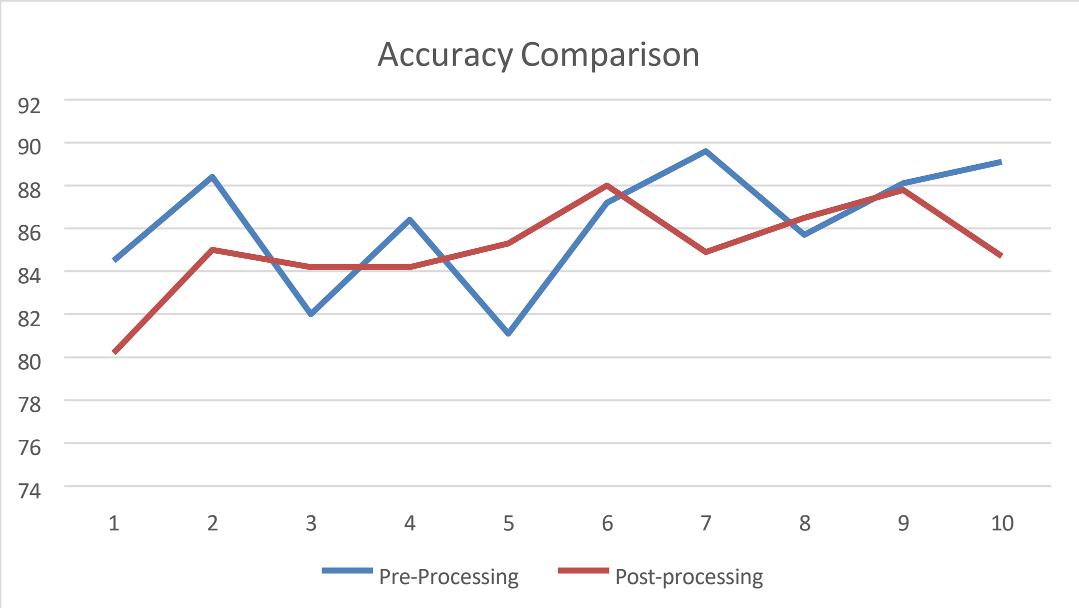
Table 4.

Based on the data recorded from images on MS-Excel, We further calculated the accuracy and sensitivity of our project and also a comparison between the Pre-Processing Filters and PostProcessing Filters .

The following output is received for ultrasound where kidney stone is not detected –

GROUP		No. of Samples	Mean	Std. Deviation	Std. Mean Error
Accuracy	Pre-Processing	228	86.21	2.910689	0.807059133
	Post-Processing	228	85.08	2.199394	0.686699028
Sensititvity	Pre-Processing	228	84.99	1.761596	0.812130258
	Post-Processing	228	83.86	3.780711	0.69449422

The following graphs compares the accuracy and sensitivity of the Pre-Processing and Post-Processing based on 10 iterations recorded of ultrasound images where kidney stones are detected



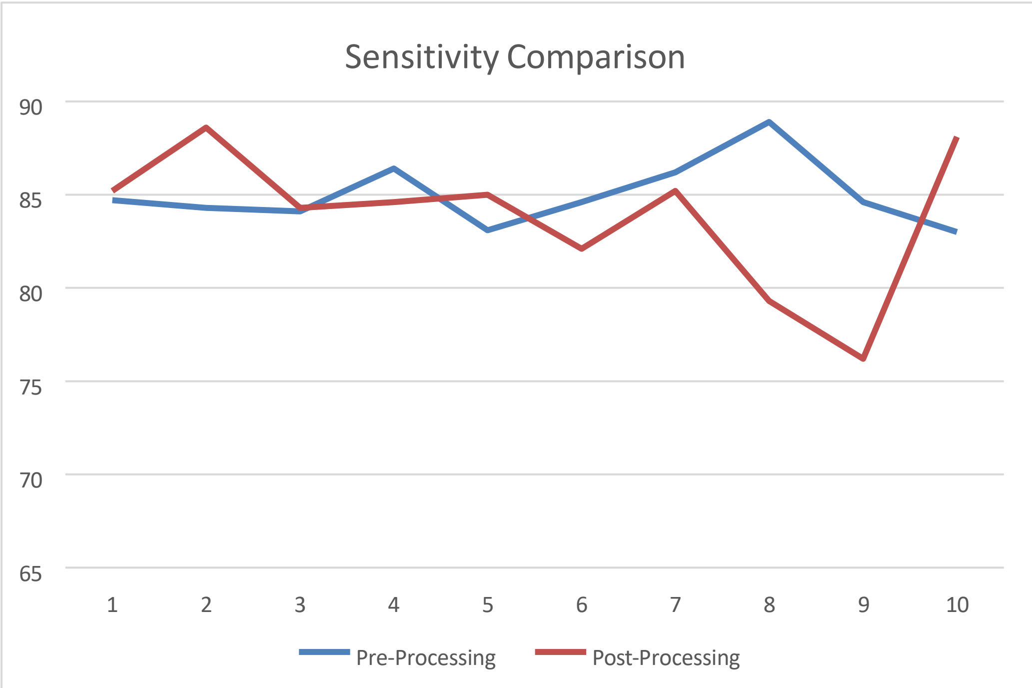


Fig 4.2
Graphs representing Accuracy and Sensitivity

4.1 Project Outcome:

- Project Accuracy: 87%.
- Project Sensitivity: 84%
- Project Report: Completed
- Research Paper: Submitted
- Website: N/A
- Patent: N/A

CHAPTER 5

5.1 Conclusion:

We, in this Major Project, Developed a Kidney stone Detection system using Image Processing Techniques. We Initially started with a database of 114 ultrasound images and further interest to the project we obtain another 114 images from a hospital's website. we Initially wanted to develop a model with the highest accuracy and compete with the existing AI and Deep learning models present in the industry with similar kind of project. We used several Image Processing Techniques including (Log filter, Negate filter, Contrast Filter, Edge detection techniques such as Prewitt filter, Sobel Filter and Canny filter. From our calculations and recorded data, we were able to:

- Achieved 87% Accuracy using Basic Digital Image Processing Techniques.
- Create a user friendly model which can be implemented by anyone
- Able to utilize ultrasound images and make prediction based on that which is itself beneficial in terms of cost as well as easy to get to any patient or doctor.
- 5.2 Applications of the Major Project
-
- This project can be widely used in the field of healthcare.
- This project can also be widely used by any person who is having any doubt regarding kidney problem or is suffering from some kidney disease.
- This project can be used by researchers as well to detect new object inside any hollow or blurred image using basic image processing techniques.
- This project can be used by doctors / medical practitioner for operating kidney stones related operations.
- This project can be used by researchers for studying about kidney and kidney stones related diseases for improving their research in the field of medicine.

5.3 Limitations of the Major Project:

Note that although in our Project we have got a 87 % prediction accuracy, we are still researching on deep learning and neural network methods corresponding to this project as they can work better and more efficiently if the data set in hand was more complex and bigger. However, the study's fundamental flaw is the healthcare system's complexities. These limitations will be overcome over time, but in the meanwhile, we will be able to apply more complex combinations of ways to achieve more significant results.

It takes more time to take load more complex dataset i.e images of high pixel with different properties to run on the system and extract features from the images.

5.3 **Contributions:**

- Nishkarsh Sharma:
  Research and Development for the Database, GUI Development, Image PrePreprocessing, Image Post-Processing Research and development of approach taken for Prediction, Calculating Accuracy, Tracking Sensitivity, Data Analysis, Comparison and visualization of data.

5.5 Future Work

- Note that although in our Project we have got a 87 % prediction accuracy, we are still researching on deep learning and neural network methods corresponding to this project as they can work better and more efficiently if the data set in hand was more complex and bigger.

- First, we are thinking of increasing the size of the current data set by a huge amount by gathering data off the internet or by putting some random manual data into it, then we will try to implement a Neural Network for Kidney stone Detection using deep learning and Artificial Intelligence Model, as of right now we are in the process of learning and researching about neural network and its different implementation on this project.

References

[1] "Automatic Detection and Classification of Kidney Stones Using Deep Learning Techniques" by Krishnamoorthy et al. (2019):

[2]"Deep Learning-Based Automated Kidney Stone Detection and Localization Using Non- Contrast Computed Tomography Images" by Al-Turaiki et al. (2020):

[3]"Automated Detection of Kidney Stones from CT Scans Using Convolutional Neural Networks" by Kader et al. (2020):

[4]"Automatic Detection of Kidney Stones on CT Images Using Deep Convolutional Neural Networks and a Transfer Learning Strategy" by Asadi et al. (2021):

[5]"Kidney Stone Detection Using Deep Learning Techniques: A Comparative Study" by Singh et al.(2021):

[6]Shabana, A., & Jaafar, F. (2019). Deep learning approach for kidney stone detection using X-ray images. International Journal of Advanced Computer Science and Applications,10(5),89-94.

[7]Huang, X., Li, L., Liu, B., Li, S., Li, S., & Xie, X. (2019). A deep learning approach to kidney stone detection using CT images. Journal of medical systems,43(7),178.

[8]Mohammed, S. S., & Mokhtar, N. (2019). Kidney stone detection using convolutional neural networks. Journal of engineering and applied sciences, 14(18), 6599-6603.

[9]Patel, D. K., Patel, K. S., & Patel, P. M. (2019). Kidney stone detection using deep learning approach. International Journal of Advanced Research in Computer and Communication Engineering,8(8),425-428.

[10]Li, C., & Li, Z. (2019). Research on kidney stone image recognition based on deep learning. Journal of healthcare engineering,2019.

[11]Song, C., Jia, X., Wang, M., Xu, W., & Wang, Y. (2019). Automatic detection of kidney stones using deep learning techniques. Biomedical engineering online,18(1),81.

[12]Rathore, S., & Dubey, S. R. (2020). Kidney stone detection using deep learning approach. Procedia Computer Science, 167, 1660-1669.

[13]Li, Y., Sun, L., Jiang, M., Wang, Y., & Zhang, H. (2020). Automatic detection of kidney stones on computed tomography images using a deep learning algorithm. Journal of X- ray science and technology, 28(1),99-109.

[14]Mohamed, S., & Almalki, S. (2020). A deep learning approach for the detection of kidney stones using ultrasound images. Journal of medical systems,44(6),119.

[15] Alkinani, M. H., & Al-Garadi, M. A. (2020). A deep learning approach for kidney stone detection using ultrasound images. Sensors,20(11),3092

[16] Sun, L., Li, Y., Wang, Y., & Zhang, H. (2020). Kidney stone detection on computed tomography images using deep learning techniques: a review. Journal of healthcare engineering 2020.

[17] Ghassemi, M. (2020). A deep learning approach for kidney stone detection and classification using ultrasound images. SN Applied Sciences,2(12), 1-8.

18] Alomari, R. S., Al-antari, M. A., & Abualsaud, M. A. (2020). Kidney stone detection in ultrasound images using deep learning techniques. Journal of digital imaging,33(6),1493-1502

[19] Xiao, Y., He, X., Gao, J., Zhang, Y., & Gao,J. (2020). A deep learning approach for kidney stone detection using CT images. Journal of X- ray science andtechnology,28(5),947-959.

[20] Zhao, X., Zhang, Y., Li, X., Li, H., Li, M.,Li, Y.,& Li, Z. (2020). A deep learning approach for kidney stone detection using CT