Detecting Breast Cancer using Neural Nets

What is the Project all about?

In India and over the world, Cancer has become a deadly disease and more and more people are suffering from Cancer and a survey says one in every 30 women suffer from this disease in their lifetime and so basically the project was first thought of because of the increase in cases of breast cancer and one thing which is very important that if we can detect the Cancer at an early stage then there is an increased chances of it getting cured.So this project lays a foundation in making the detection of the cancer automated so that more and more people can get it diagonised early so as get cured.

How it is implemented?

The signs of detection are Masses and micro calcification clusters which are important in early detection of breast cancer.

Micro calcification are nothing but tiny mineral deposits within the breast tissue. They look similar to small white colored spots. They may or may not be caused by cancer.

Masses can be many things, including cysts (fluid-filled sacs) and non-cancerous solid tumors, but they could also be cancerous.

The difficulty in cancer detection is that the abnormalities from normal breast tissues are hard to read because of their subtle appearance and ambiguous margins.Automated tools which can help radiologist in early detection of breast cancer.

Further we have classified the cancer into three categories after its detection- Normal,Malignant,Benign.

Methodology

We have used adaptive mean filter to remove noise from image. since it is better among all the spatial filters and distinguish fine details from noise.The Adaptive Median Filter performs spatial processing to determine which pixels in an image have been affected by impulse noise. **The Adaptive Median Filter** classifies pixels as noise by comparing each pixel in the image to its surrounding neighbor pixels.

```
function Output=adaptivemedian(GrayScaleImage);

        NoisyImage=GrayScaleImage;
        NoisyImage=double(GrayScaleImage);
        [R C P]=size(NoisyImage);
        OutImage=zeros(R,C);
        Zmin=[];
        Zmax=[];
        Zmed=[];
```

```
    for i=1:R

        for j=1:C
        if (i==1 & j==1)
        % for right top corner[8,7,6]
        elseif (i==1 & j==C)


        % for bottom left corner[2,3,4]
        elseif (i==R & j==1)


         % for bottom right corner[8,1,2]

        elseif (i==R & j==C)

        %for top edge[8,7,6,5,4]
        elseif (i==1)

        % for right edge[2,1,8,7,6]
        elseif (i==R)

        % // for bottom edge[8,1,2,3,4]
        elseif (j==C)


         %// for left edge[2,3,4,5,6]
        elseif (j==1)


       else

             SR1 = NoisyImage((i-1),(j-1));
             SR2 = NoisyImage((i-1),(j));
             SR3 = NoisyImage((i-1),(j+1));
             SR4 = NoisyImage((i),(j-1));
             SR5 = NoisyImage(i,j);
             SR6 = NoisyImage((i),(j+1));
             SR7 = NoisyImage((i+1),(j-1));
             SR8 = NoisyImage((i+1),(j));
             SR9 = NoisyImage((i+1)),((j+1));
             TempPixel=[SR1,SR2,SR3,SR4,SR5,SR6,SR7,SR8,SR9];
             Zxy=NoisyImage(i,j);
             Zmin=min(TempPixel);
             Zmax=max(TempPixel);
             Zmed=median(TempPixel);
             A1 = Zmed - Zmin;
             A2 = Zmed - Zmax;

             if A1 > 0 && A2 < 0

                  %   go to level B
                  B1 = Zxy - Zmin;
```

```
                   B2 = Zxy - Zmax;
                   if B1 > 0 && B2 < 0
                       PreProcessedImage(i,j)= Zxy;
                   else
                       PreProcessedImage(i,j)= Zmed;

                   end
              else

                  if ((R > 4 && R < R-5) && (C > 4 && C < C-5))

                  S1 = NoisyImage((i-1),(j-1));
                  S2 = NoisyImage((i-2),(j-2));
                  S3 = NoisyImage((i-1),(j));
                  S4 = NoisyImage((i-2),(j));
                  S5 = NoisyImage((i-1),(j+1));
                  S6 = NoisyImage((i-2),(j+2));
                  S7 = NoisyImage((i),(j-1));
                  S8 = NoisyImage((i),(j-2));

                  S9 = NoisyImage(i,j);
                  S10 = NoisyImage((i),(j+1));
                  S11 = NoisyImage((i),(j+2));
                  S12 = NoisyImage((i+1),(j-1));
                  S13 = NoisyImage((i+2),(j-2));
                  S14 = NoisyImage((i+1),(j));
                  S15 = NoisyImage((i+2),(j));
                  S16 = NoisyImage((i+1)),((j+1));
                  S17 = NoisyImage((i+2)),((j+2));
                  TempPixel2=[S1,S2,S3,S4,S5,S6,S7,S8,S9,S10,S11,S12,S13,S14,S15,S16,S17
                  Zmed2=median(TempPixel2);
                  PreProcessedImage(i,j)= Zmed2;
                  else

                  PreProcessedImage(i,j)= Zmed;

                  end

              end

           end

       end
 end
```

The size of the neighborhood is adjustable, as well as the threshold for the comparison. A pixel that is different from a majority of its neighbors, as well as being not structurally aligned with those pixels to which it is similar, is labeled as impulse noise.

These noise pixels are then replaced by the median pixel value of the pixels in the neighborhood that have passed the noise labeling test.we are initially converting the image into grayscale image using rgb2gray()

function then applying adaptive mean filtering to the resulting image and then converted the image into unsigned integer 8 using unit8() function.

```
function  b=getbinary(a,X);
Centeroid=180;
a=uint8(a);
ag=rgb2gray(a);
[r c p]=size(ag);
for i=1:r
    for j=1:c
        data=ag(i,j);
        if data >Centeroid;
           X(i,j,1)=255;
           X(i,j,2)=255;
           X(i,j,3)=255;
        end

    end
end
```

In this way we preprocessed image.then we performed GMM segmentation(Gaussian Mixture Model) on the preprocessed image with number of regions 2 and number of GMM components 2 and maximum number iterations 10. we performed k-means segmentation with k=2. then we Implemented HMRF-EM (Hidden Markov Random Field Model) and its Expectation-Maximization Algorithm.

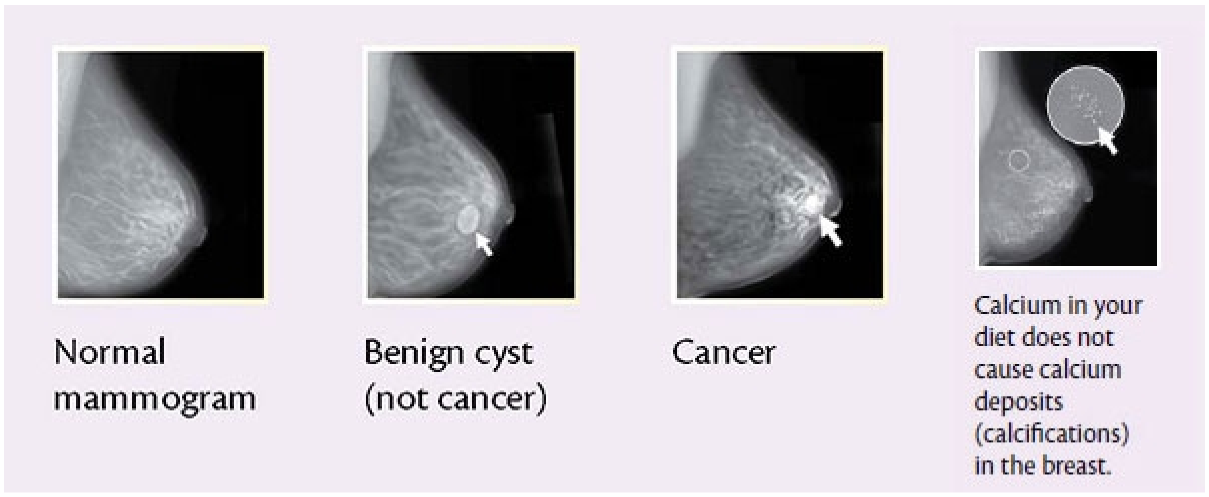
Gaussian Mixture Model:

```
function GMM=get_GMM(X,Y,g)

k=max(X(:));
GMM=cell(k,1);

for i=1:k
    index=(X==i);
    Y1=Y(:,:,1);
    Y2=Y(:,:,2);
    Y3=Y(:,:,3);
    XX=[Y1(index) Y2(index) Y3(index)];
    GMM{i} = gmdistribution.fit(XX,g,'Regularize',1);
end
```

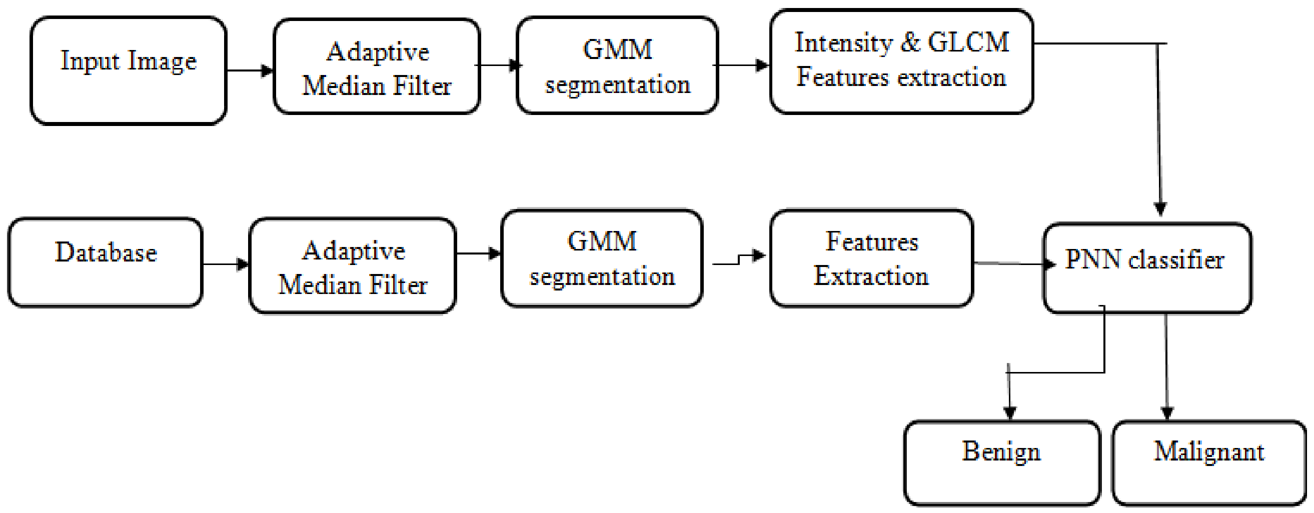
K Means:

```
function [X GMM ShapeTexture]=image_kmeans(Y,k,g)
[m n temp]=size(Y);

if temp==3
   b=rgb2gray(Y);
   ShapeTexture=wlt4(b);

elseif temp==1

      ShapeTexture=wlt4(Y);
        Y1(:,:,1)=Y;
        Y1(:,:,2)=Y;
        Y1(:,:,3)=Y;
Y=Y1;
end


y=reshape(Y,[m*n 3]);
x=kmeans(y,k);
X=reshape(x,[m n]);

GMM=get_GMM(X,Y,g);
```

The picture decribes the difference between Malignant and Benign tissues in Breast


Normal mammogram
Benign cyst (not cancer)
Cancer
Calcium in your diet does not cause calcium deposits (calcifications) in the breast.

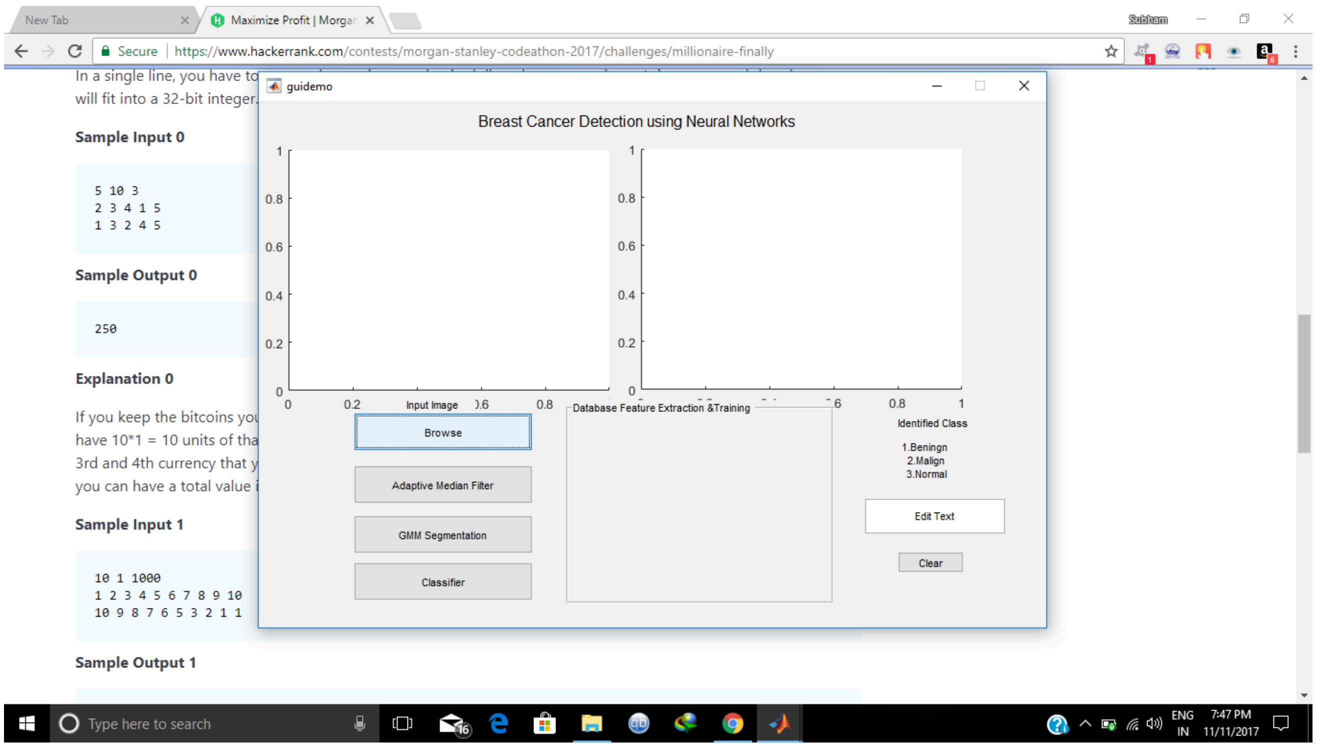
Block Diagram of the Project



How to make the project work?

Open the project in matlab and then run guidemo and then a gui mode window will open and then just follow the steps there.For further information check the screenshots

Now you have to browse the image of the mammograms and give it as an input:



In a single line, you have to
will fit into a 32-bit integer.

Sample Input 0

```
5 10 3
2 3 4 1 5
1 3 2 4 5
```

Sample Output 0

```
250
```

Explanation 0

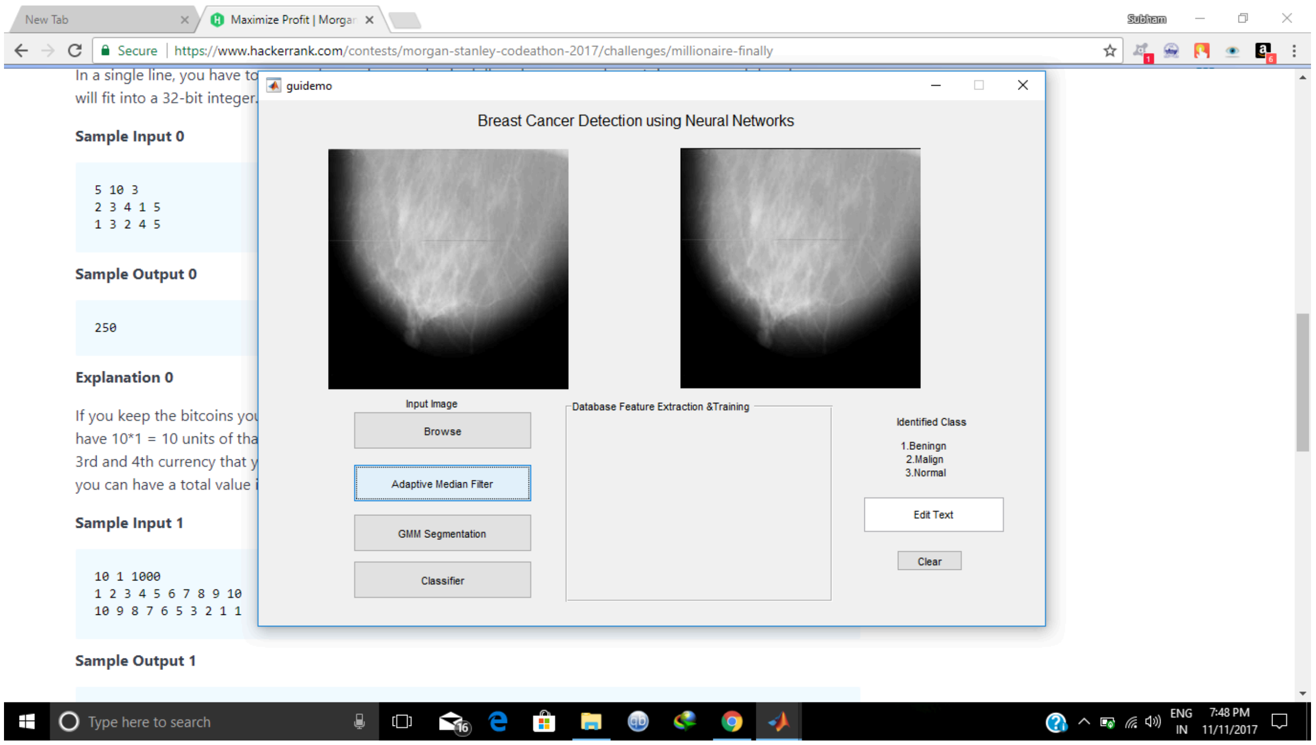
If you keep the bitcoins you
have 10*1 = 10 units of tha
3rd and 4th currency that y
you can have a total value i

Sample Input 1

```
10 1 1000
1 2 3 4 5 6 7 8 9 10
10 9 8 7 6 5 3 2 1 1
```

Sample Output 1

In this step adaptive mean filtering is done:



In a single line, you have to
will fit into a 32-bit integer.

Sample Input 0

```
5 10 3
2 3 4 1 5
1 3 2 4 5
```

Sample Output 0

```
250
```

Explanation 0

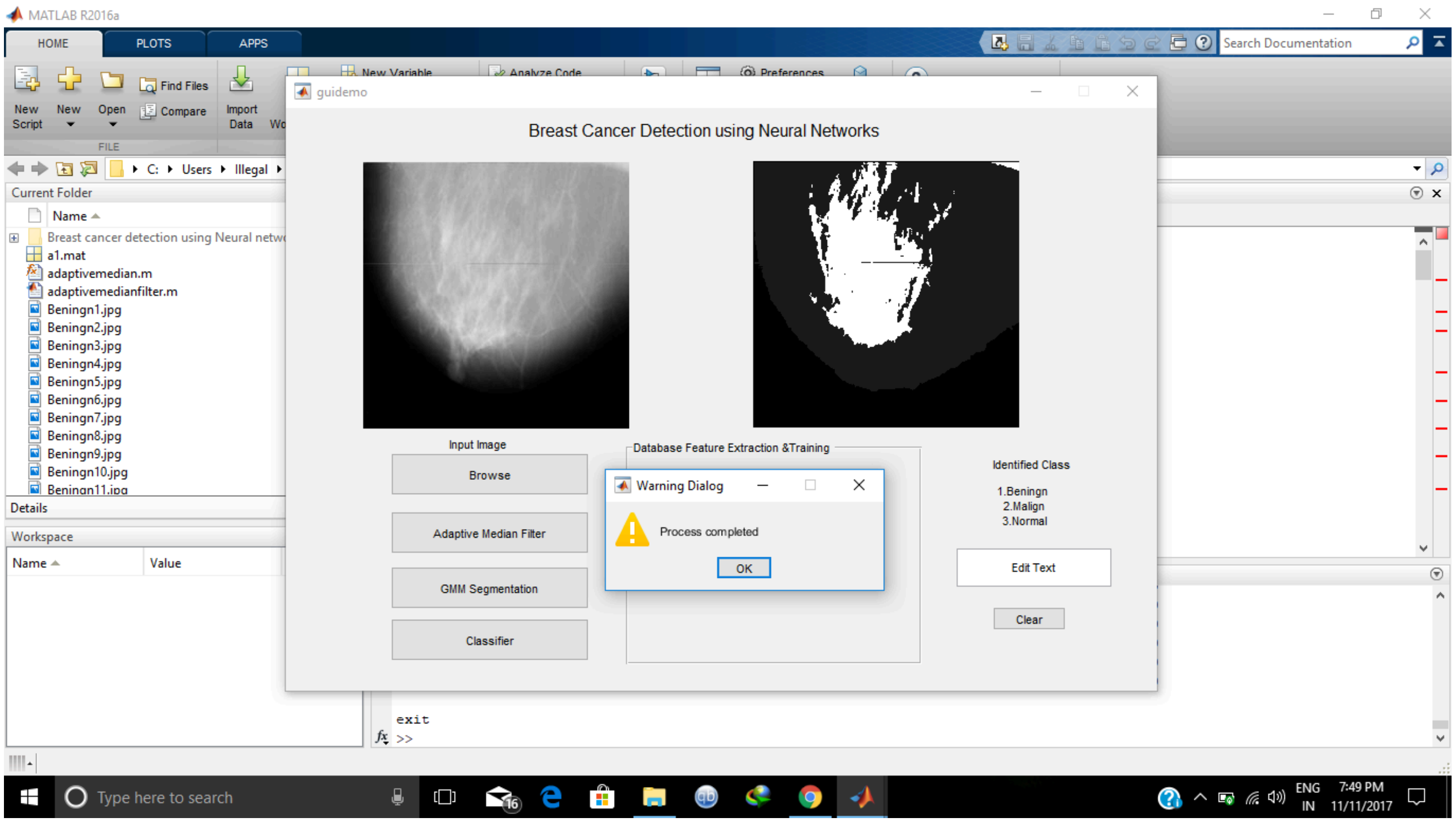
If you keep the bitcoins you
have 10*1 = 10 units of tha
3rd and 4th currency that y
you can have a total value i

Sample Input 1

```
10 1 1000
1 2 3 4 5 6 7 8 9 10
10 9 8 7 6 5 3 2 1 1
```

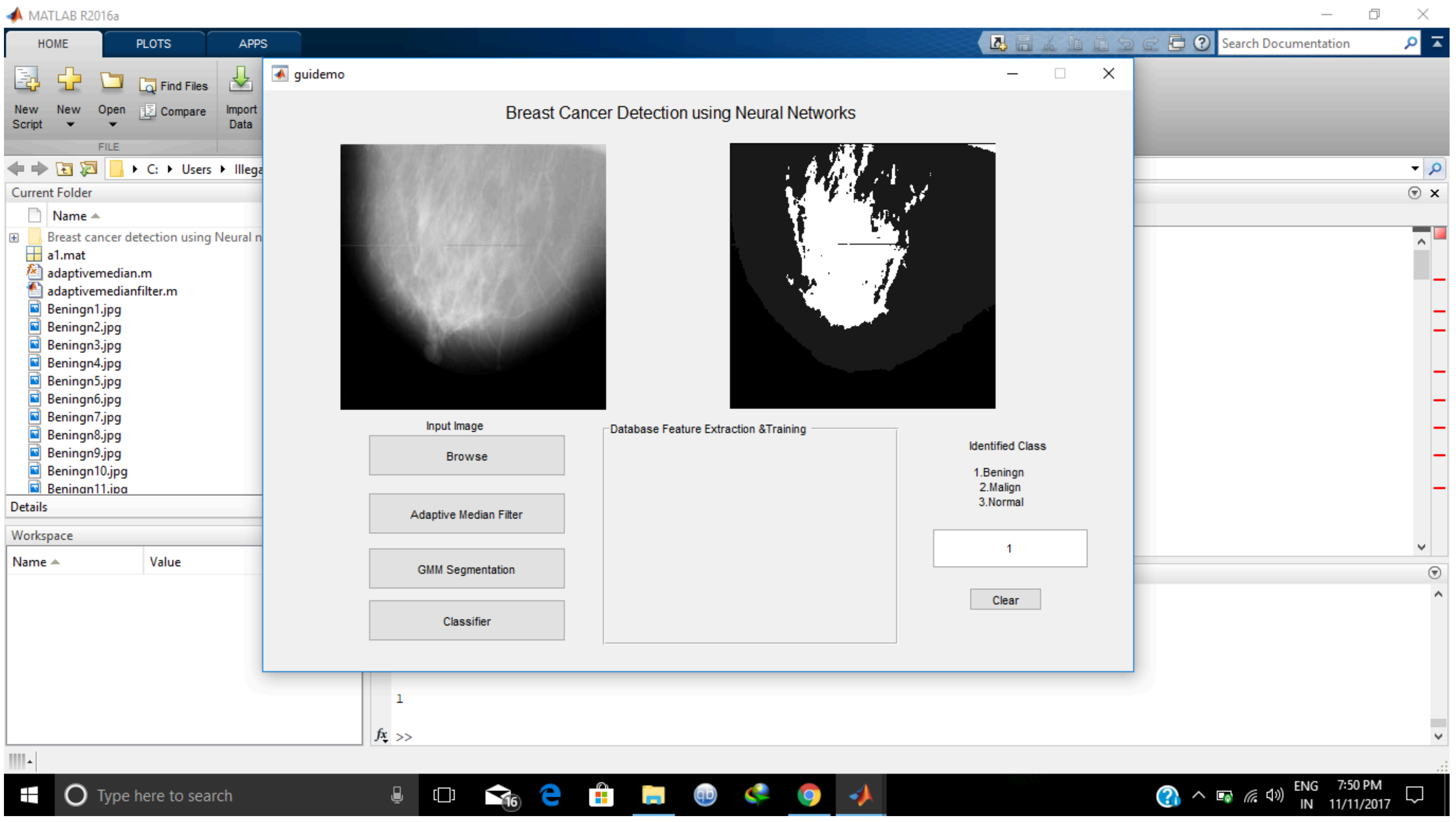
Sample Output 1

GMM Segmentation is done:



```
exit
>>
```

So you can see one as the output in the right side which depicts that the cancer is benign:



```
1
>>
```

References

Anuj Kumar Singh and Bhupendra Gupta “A novel approach for breast cancer detection and segmentation in mammography ” Expert System With Applications 42(2015)990-1002.

J. Dheeba, N.Albert Singh, S. Tamil Selvi “Computer-aided detection of breast cancer on mammograms: A swarm intelligence optimized wavelet neural network approach” Journal of Biomedical Informatics (2014).