Assignment 2

Submitted by EE19S042

September 25,2020

1 Task 1 : Denoising

1.1 Noise Generation in image

In this section generation of various noise found in ultrasound images has been described. Basically noise present in ultra-sound images are Speckle noise and gaussian noise. So these two noises are described below.

1.1.1 Introducing Gaussian noise

It is also known as Amplifier noise and generated as a result of thermal vibration of atoms and radiation of warm objects.This noise like Gaussian distribution in structure.It is generated using Gaussian function given as,

$$W(n) = \frac{1}{\sqrt{2 * \pi * \sigma^2}} * \exp\left(\frac{(N-m)^2}{\sigma^2}\right) \tag{1}$$

1.1.2 Introducing Speckle Noise

This is a random granular pattern produced mainly by multiplicative disturbances. The speckle noise strongly impedes the visual evaluation of ultrasound images, affects edges and fine details and decreases the diagnostic value of ultrasound images. Speckle noise can also mask small, but diagnostically significant image features, and it reduces the efficiency of detection and recognition of the anatomical structures in medical images.

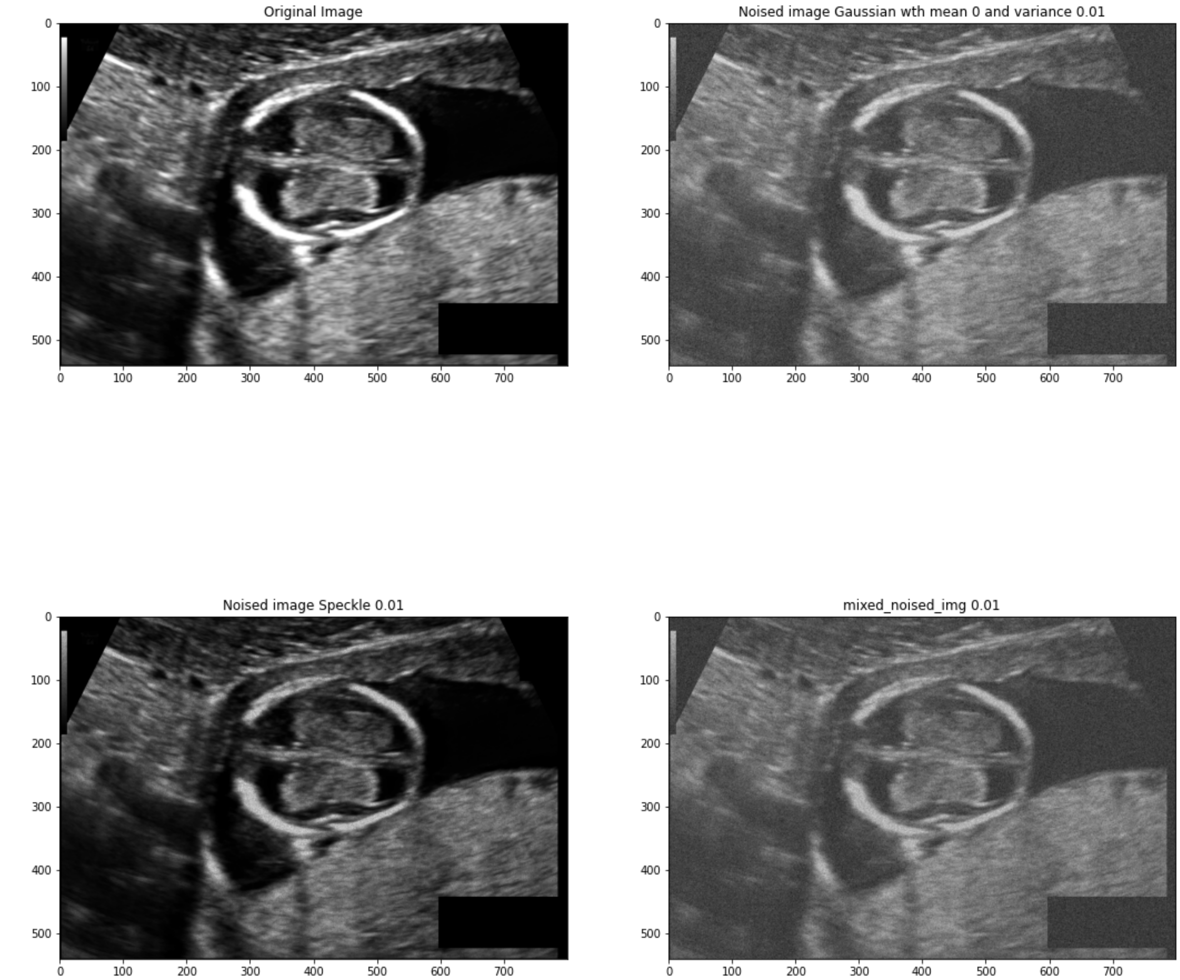

Figure 1: Gaussian and Speckle generated noised image (a)Original Image, (b)Gaussian Noised image (c)Speckle Noise (d)A mixture of gaussian and speckle noise

1.2 Filters used to denoise ultrasound imaging Noises

Various filters for removing above speckle and Gaussian noise are as described below.

1.2.1 Mean Filter

It is a spatial (linear) filtering technique that replaces the value of pixels in the window with the mean of the pixels value in that window. It is usually used for the purpose of de-noising and smoothening of the image. Poor in preservation of useful details in image after noise removal. Convolution of a kernel of ones with the image and normalizing it results in mean filtered image. The result of Mean filter for filtering out Gaussian and speckle noise is as shown in figure 2,

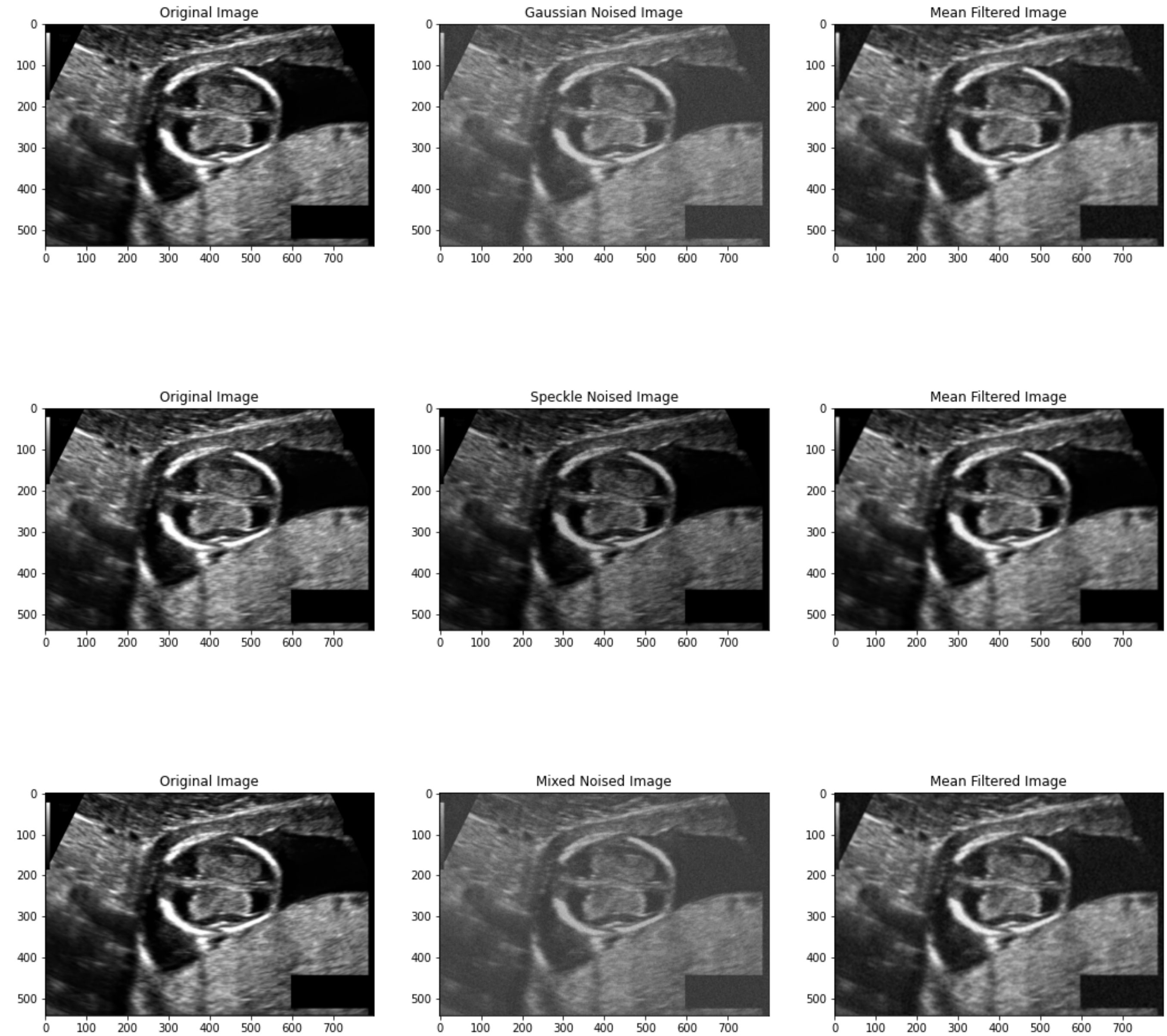

Figure 2: row 1 - (a)Original Image, (b)Gaussian Noised image (c)mean filtered image row 2-(a)Original Image, (b)Speckle Noised image (c)mean filtered image row 2-(a)Original Image, (b)mixed Noised image (c)mean filtered image

1.2.2 Lee Filter

Lee filter is used specifically for speckle noise removal.The Lee filter kernel function is defined as,

$$Y_{ij} = \bar{K} + W * (C - \bar{K}) \tag{2}$$

Where Y_{ij} is the filtered image
$\bar{K}$ is the mean of the kernel / window

W is the weighing function
C is the center element in the kernel / window
To calculate W : $W = \frac{\sigma_k^2}{\sigma_k^2 + \sigma^2}$

Where σ_k^2 is the variance of uniformly filtered image
σ^2 is the variance of noised image

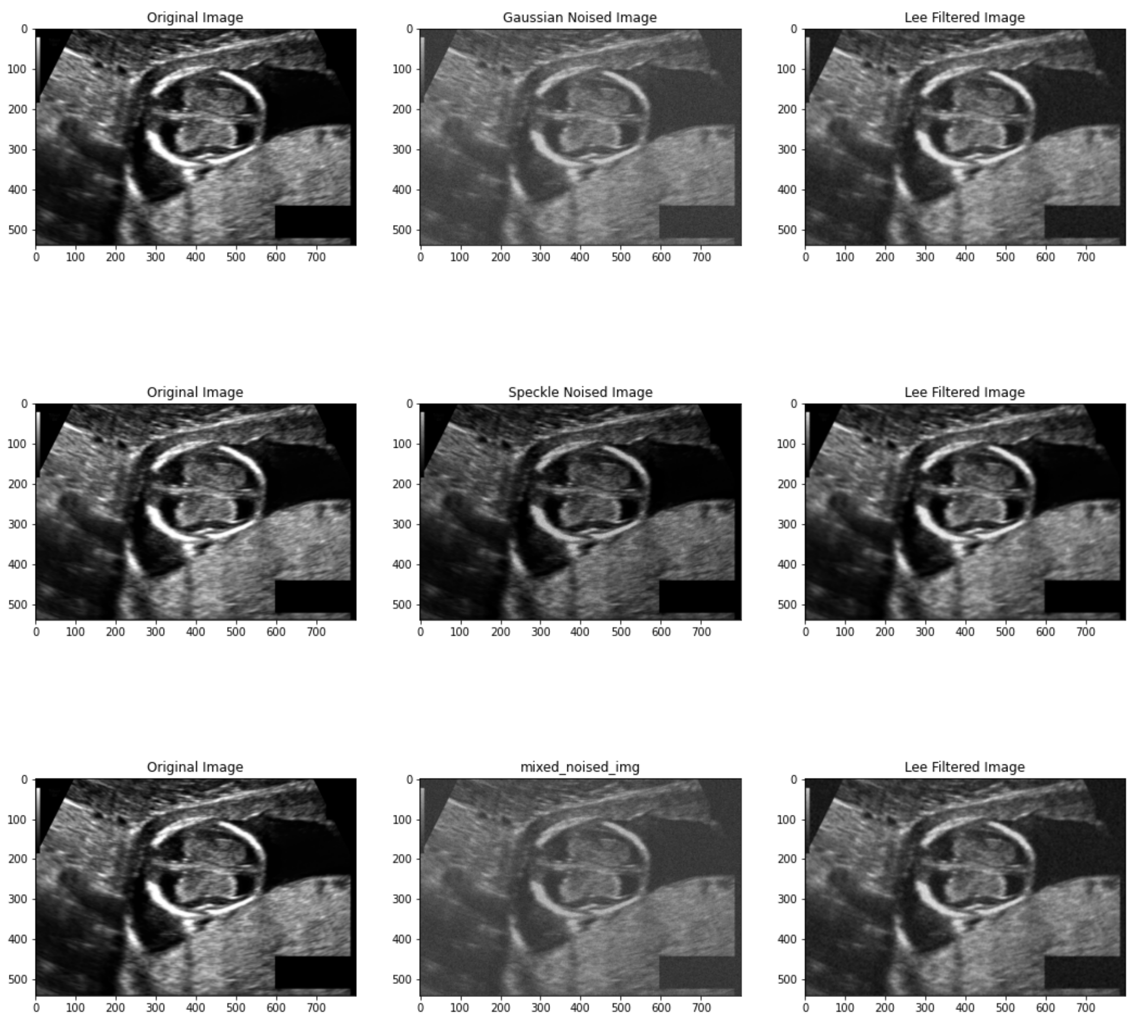

Figure 3: row 1 - (a)Original Image, (b)Gaussian Noised image (c)lee filtered image row 2-(a)Original Image, (b)Speckle Noised image (c)lee filtered image row 2-(a)Original Image, (b)mixed Noised image (c)lee filtered image

Algorithm of Lee filter is,
1. Uniform filter is used to calculate the variance of the given noised image.Uniform filter is applied patch by patch using a kernel.Corresponding variance is calculated.
2.Variance of the given image is calculated.
3.The mean of the uniformly filtered image is calculated on batch basis depending on kernel size.
4.Weight W is calculated using the formula given above.
5.The noisy image is processed patch by patch using above formula.
The result of Lee filter for filtering out Gaussian and speckle noise is as shown in figure3,

1.2.3 Gaussian Filter

It is a linear filter that is used to remove noise from the image along with the blurring of image similar to average filter. It differs from average filter in the aspect that it uses different kernel from mean filter which is in the shape of bell curve (Gaussian PDF). It gives more weightage to center pixel unlike meaan filter which gives equal wightage to all the pixels in the kernel. Convolution of gaussian kernel with the noised image result in filtered image.

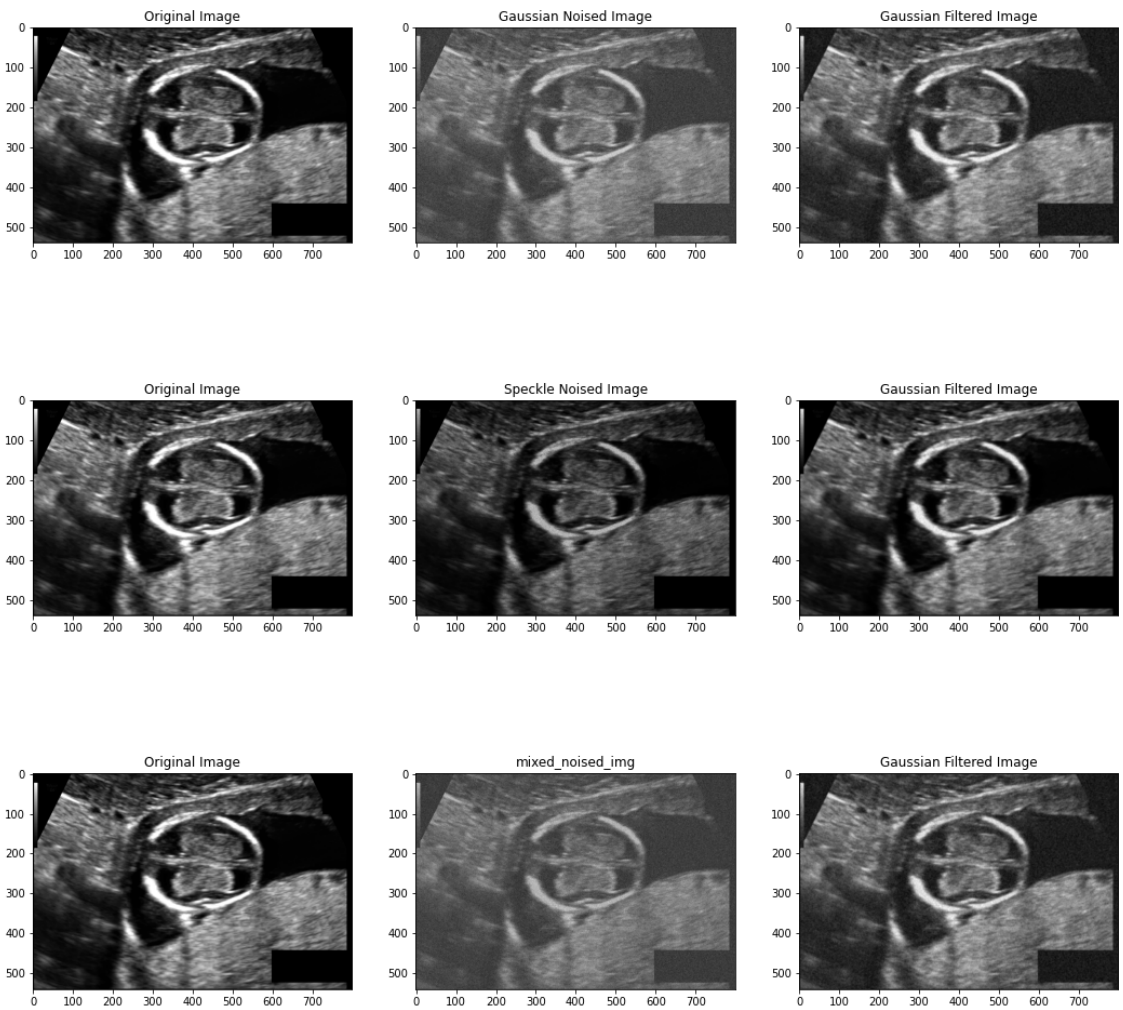

Figure 4: row 1 - (a)Original Image, (b)Gaussian Noised image (c)Gaussian filtered image row 2-(a)Original Image, (b)Speckle Noised image (c)Gaussian filtered image row 2-(a)Original Image, (b)mixed Noised image (c)Gaussian filtered image

In 2-Dimensional, Gaussian has the equation:

$$G(x, y) = \frac{1}{2 * \pi * \sigma^2} \exp -(\frac{x^2 + y^2}{2\sigma^2}) \tag{3}$$

where mean is (0,0) and σ^2 is the variance between 0 to 1. The result of Gaussian filter for filtering out Gaussian and speckle noise is as shown in figure 4.

1.2.4 Bilateral Filter

In medical ultrasound image processing, the suppression of speckle noise, while preserving edges and image details, plays a crucial role for the diagnosis. Bilateral filter, filter the image based on distance value as well as intensity value and hence helps in preserving the edges.Convolution of gaussian kernel with spatial kernel result in kernel for bilateral filtering which again convolved with the noised image result in filtered image.

$$BF[I_p] = \frac{1}{W_p} * \sum_{q\epsilon S} (G_{\sigma_S}(||\,(p-q)\,|| * (G_{\sigma_S}(||\,(I_p - I_q)\,||) * I_q \tag{4}$$

The result of Bilateral filter for filtering out Gaussian and speckle noise is as shown in figure5,

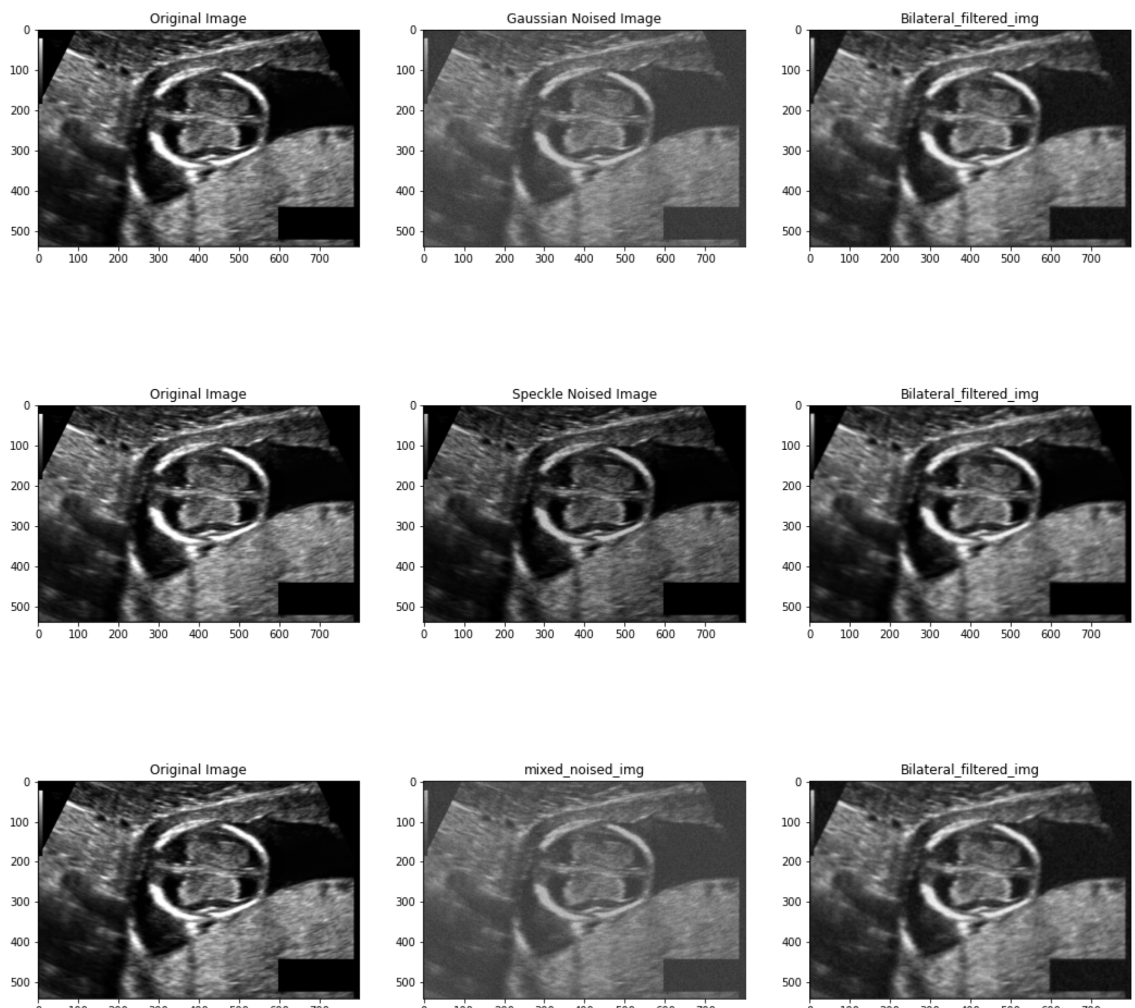

Figure 5: row 1 - (a)Original Image, (b)Gaussian Noised image (c)bilateral filtered image row 2-(a)Original Image, (b)Speckle Noised image (c)bilateral filtered image row 2-(a)Original Image, (b)mixed Noised image (c)bilateral filtered image

1.2.5 Wiener Filter

The Wiener filtering executes an optimal trade off between inverse filtering and noise smoothing. It removes the additive noise and inverts the blurring simultaneously.It minimizes the overall mean square error in the process of inverse filtering and noise smoothing. The Wiener filtering is a linear estimation of the original image. The approach is based on a stochastic framework. The orthogonality principle implies that the Wiener filter in Fourier domain can be expressed as follows:

$$F(i,j) = \frac{H^{\star}(i,j) * S_{xx}(i,j)}{\mid H(i,j) \mid^{2} * S_{xx}(i,j) + *S_{nn}(i,j)} \tag{5}$$

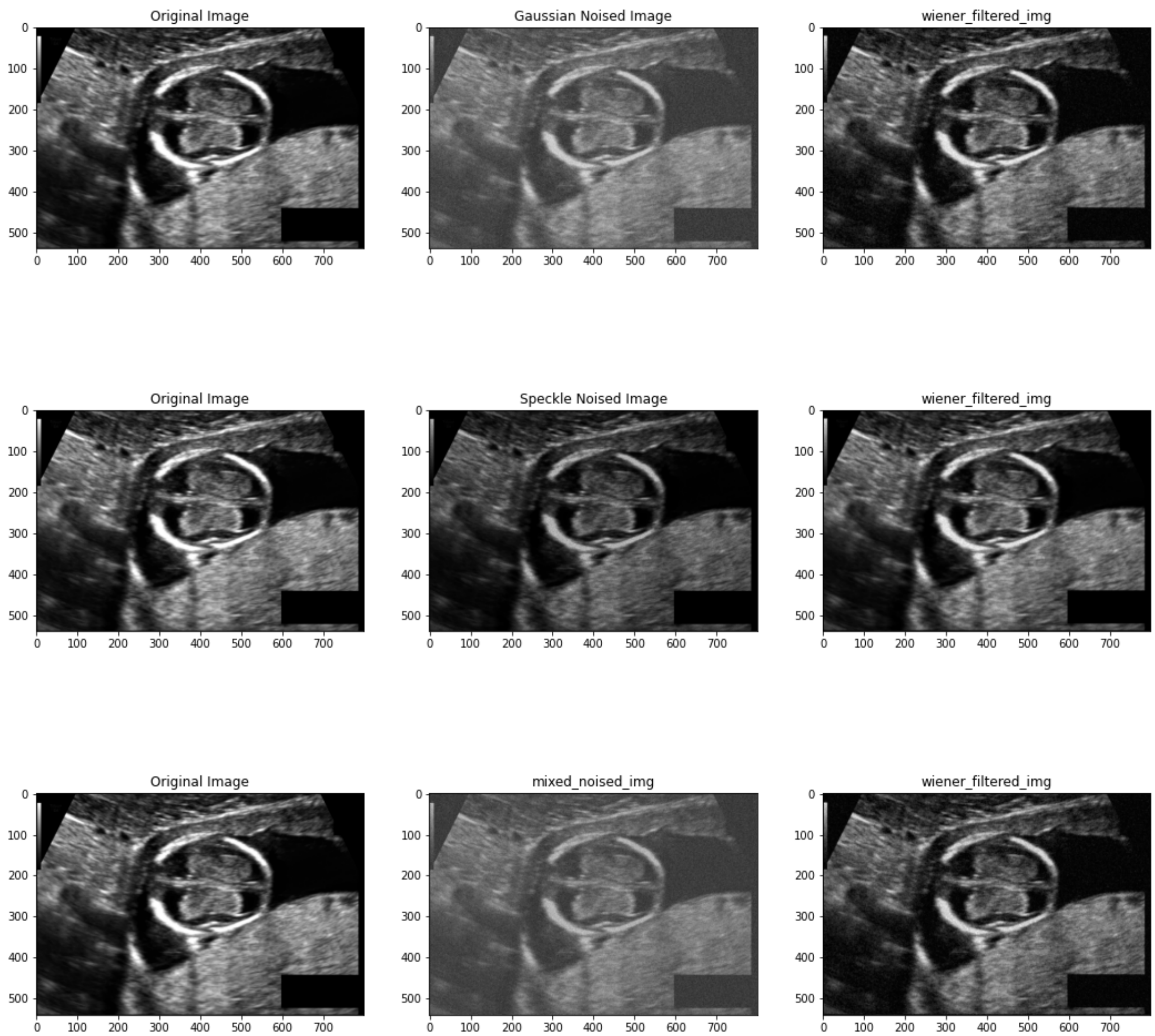

Figure 6: row 1 - (a)Original Image, (b)Gaussian Noised image (c)wiener filtered image row 2-(a)Original Image, (b)Speckle Noised image (c)wiener filtered image row 2-(a)Original Image, (b)mixed Noised image (c)wiener filtered image

In this case the point spread function is considered as Gaussian kernel.The wiener filter is computed using above formula in fourier domain. The multiplication of image with the wiener filter in fourier domain result in an fourier denoised image in fourier domain and converted back using inverse fourier to form the denoised image.

1.3 Plotting of residuals and analysing the performance using PSNR

PSNR As the name explains, it is the ratio of the maximum/peak value of the signal to the noisy signal value. PSNR formally describes the quality of the reconstructed image after the application of any technique on it. Higher the PSNR, better the quality of reconstructed image. PSNR is expressed as:

$$PSNR = 10 \log_{10} \frac{(peakvalue)^2}{MSE} \tag{6}$$

where Peakvalue is the maximum difference in the input image value and MSE is Mean Square Error and is computed as

$$MSE = \frac{1}{m \times n} \sum_{i=1}^{m \times n} (\hat{y}(i,j) - y(i,j))^2 \tag{7}$$

where $m \times n$ specifies the size of the image, $\hat{y}(i,j)$ is the recovered image and y(i,j) is the Original image. Table7 below shows the PSNR values for different filtering techniques with respect to different noises. σ_n^2 in db.

Filters	Gaussian Noise	Speckle Noise	Gaussian Noise + Speckle Noise
Mean	67.9372661969044	75.17484181608569	67.45742330921648
Lee	69.73267373019284	76.84186379566901	69.39683592453845
Gaussian	69.16930864913883	77.42059896897581	68.66611356899631
Bilateral	68.08594909624459	75.04509918540302	67.63309670046327
Bilateral Filter Inbuilt	68.29426818692546	76.38097679201772	74.2183416216738
Wiener	57.897212513632326	58.27765309232136	57.8443835548873
Wiener Filter Inbuilt	68.90545613252029	78.55206390987539	68.55708493518316

Figure 7: PSNR table showing PSNR of different noised image with respect to different filters

Residual plots: Residual plot is estimated as the difference of noised image and filtered denoised image.The residual plots of different noisy images and their filtered denoised image is as shown in figures below.

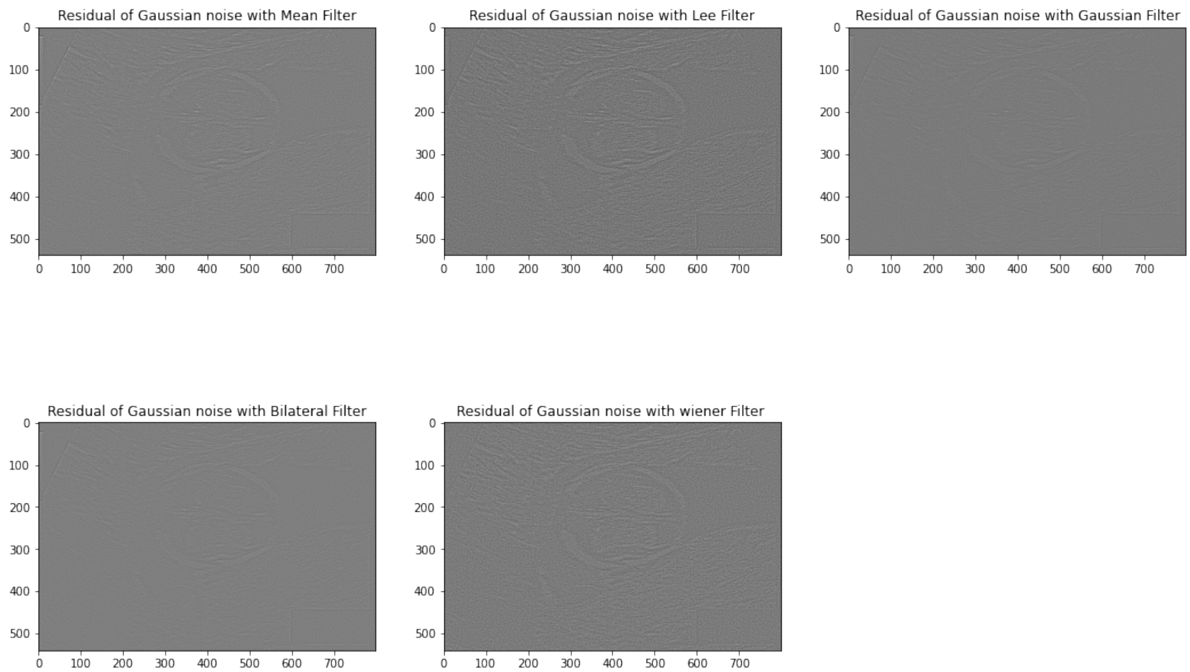

Figure 8: Residual plot for Gaussian Noise with respect to different filters.(a)Residual of Gaussian noise with mean filter (b)Residual of Gaussian noise with Lee filter(c)Residual of Gaussian noise with Gaussian filter(d)Residual of Gaussian noise with Bilateral filter(e)Residual of Gaussian noise with Wiener filter

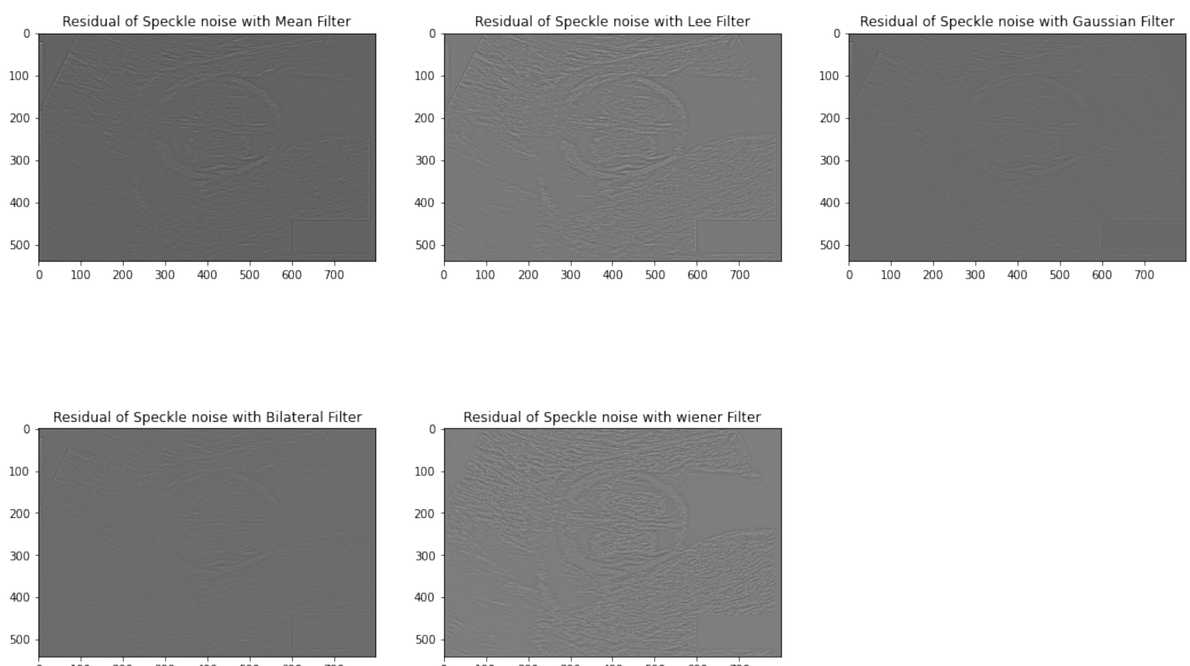

Figure 9: Residual plot for Gaussian Noise with respect to different filters.(a)Residual of Speckle noise with mean filter (b)Residual of Speckle noise with Lee filter(c)Residual of Speckle noise with Gaussian filter(d)Residual of Speckle noise with Bilateral filter(e)Residual of Speckle noise with Wiener filter

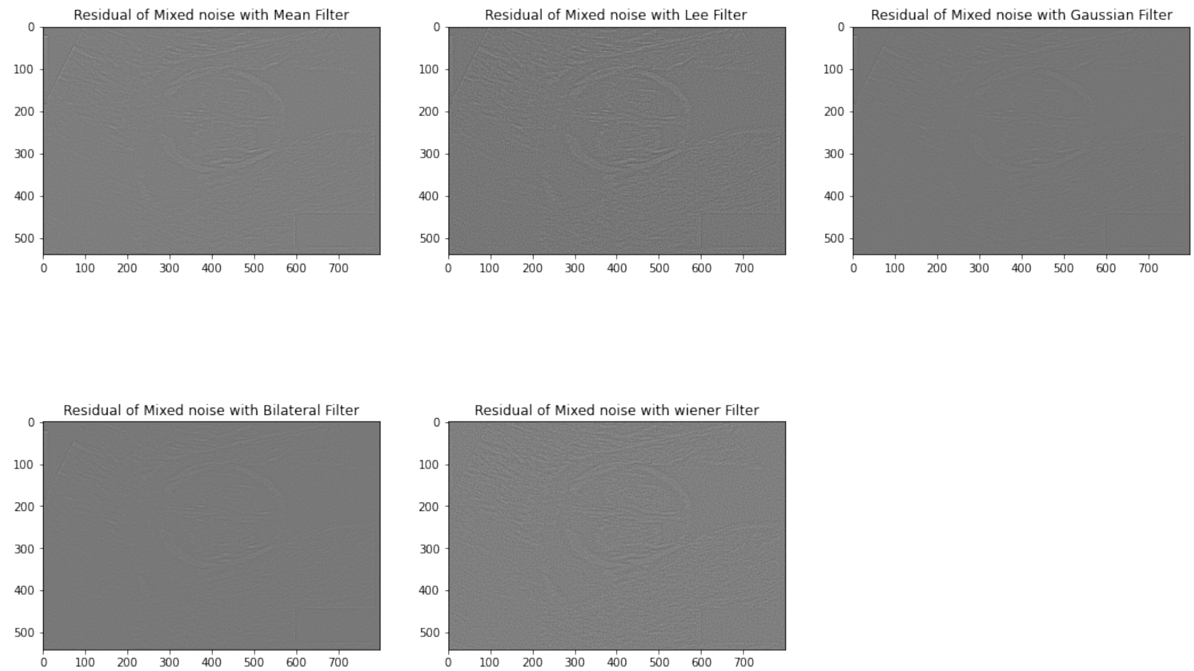

Figure 10: Residual plot for Mixed Noise with respect to different filters.(a)Residual of Mixed noise with mean filter (b)Residual of Mixed noise with Lee filter(c)Residual of Mixed noise with Gaussian filter(d)Residual of Mixed noise with Bilateral filter(e)Residual of Mixed noise with Wiener filter

Observations: Implementation of various filters on noise generated images and their performance efficiency are described above. From the figures of noisy and filtered image the effect of filters on image for improving image quality can be observed visually. PSNR table shows the statistical representation the performance of various filters. From PSNR table it is clear that for Gaussian noise, Lee filter gives better performance than any other filter. Wiener, Gaussian, Bilateral filter gives comparable results. For speckle noise Wiener filter gives better performance and along with this Gaussian, Lee and bilateral filter gives relatively comparable performance.For a combination of speckle and Gaussian noise Bilateral filter gives better performance than other. Lee, Gaussian and wiener filters give comparable results.The filtering operation also depends on the value of sigma. Large sigma value will remove the noise and smoothen the image but increasing it more result in removing the edge information. The residual plots of noisy image with respect to different filters are also been shown.

2 Task :2 Contrast Enhancement and Edge Detection

2.1 Contrast Enhancement

In this part contrast of the given image is enhanced using histogram equalisation and Contrast Adaptive Histogram Equalisation (CLAHE) methods. The results and corresponding histograms are as shown in figure11and 12.

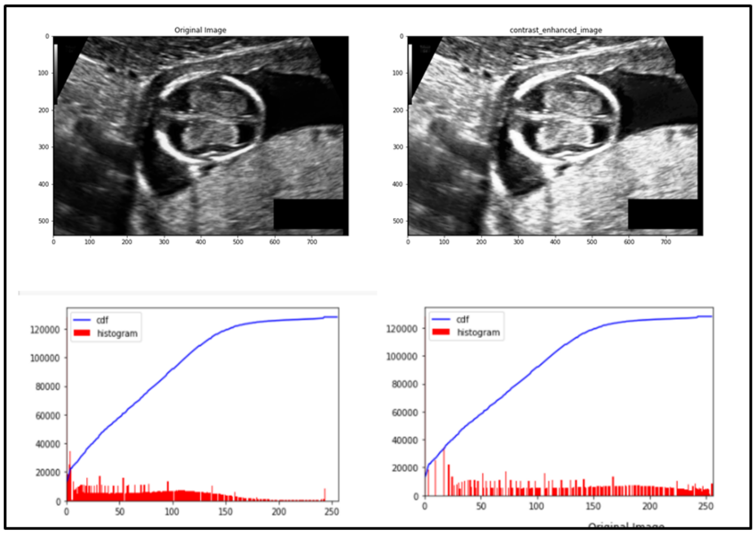

Figure 11: 1st row (a) original image (b)Histogram contrast enhanced image 2nd row (a) original image hitogram(b)Histogram contrast enhanced image histogram

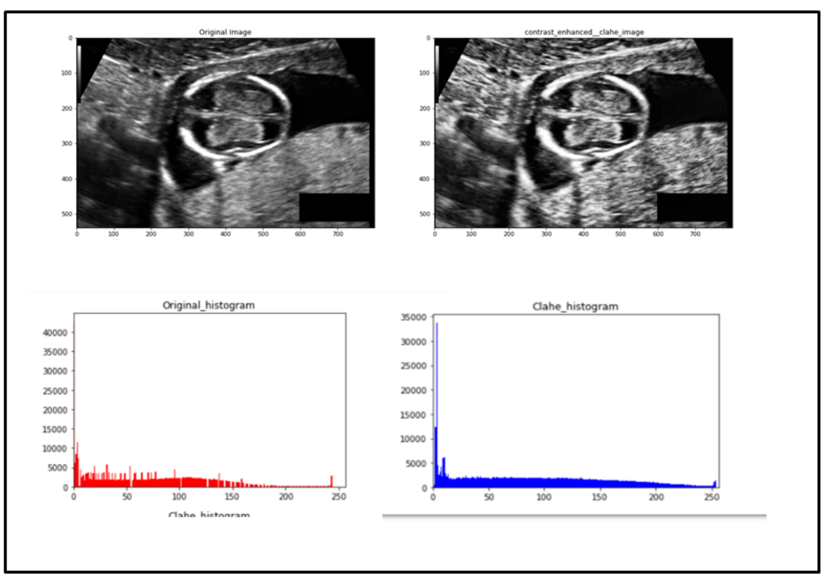

Figure 12: 1st row (a) original image (b)contrast enhanced image using CLAHE 2nd row (a) original image histogram (b)Histogram of contrast enhanced image histogram using CLAHE

Observations: From the figure histogram equalisation and contrast adaptive histogram equalisation it can be interpreted that Histogram equalisation brightens the overall image. As a result distinction of only head circumference from overall image will be difficult. While in CLAHE the contrast is more improved for foreground edgy regions than others. So CLAHE method is more efficient for head detection.

2.2 Edge Detection

In this section different edge estimation techniques are implemented on the contrast enhanced image obtained using CLAHE method.
Principle Behind Edge Detection-

1. The first thing to be done is finding the gradient of the gray scale image, allowing finding edge-like regions in the x and y direction. The gradient is a multi-variable generalization of the derivative. While a derivative can be defined on functions of a single variable, for functions of several variables, the gradient takes its place.

2. The gradient is a vector-valued function, as opposed to a derivative, which is scalar-valued. Like the derivative, the gradient represents the slope of the tangent of the graph of the function. More precisely, the gradient points in the direction of the greatest rate of increase of the function, and its magnitude is the slope of the graph in that direction.

The various edge detection techniques implemented for detecting fatal head [1]are as described below.

2.2.1 Sobel Filter

The Sobel operator measures a 2-Dimensioanal spatial gradient on an image and gives more attention on regions of high spatial gradient corresponding to edges. It is used for finding gradient magnitude at each point in a gray scale image. One way to keep away from having the gradient computed about an interpolated point between the pixels is to use 3 x 3 neighbourhood for the gradient computation. There are various pairs of Sobel operator such as 3x3, 5x5 convolution kernels. Sobel mask is defined as shown in 13. The pseudo code algorithm is discussed as follows:

-1	0	1
-2	0	2
-1	0	1

g_x

-1	-2	-1
0	0	0
1	2	1

g_y

Figure 13: Sobel kernel

Step1: Accept the sample input image.
Step2: Perform masking on given image.
Step3: Apply algorithm and the gradient.
Step4: On the input image perform mask manipulation in both the directions.
Step5: Find the absolute magnitude of the gradient. The absolute magnitude is the output edge.
The result of implementing sobel filter on contrast enhanced image is as shown in figure 14

[1]Various edge detection techniques https://www.researchgate.net/profile/Radhika_Chandwadkar/publication/297736749_Comparison_of_Edge_Detection_Techniques/links/56e2435308aebc9edb19d2fb.pdf.

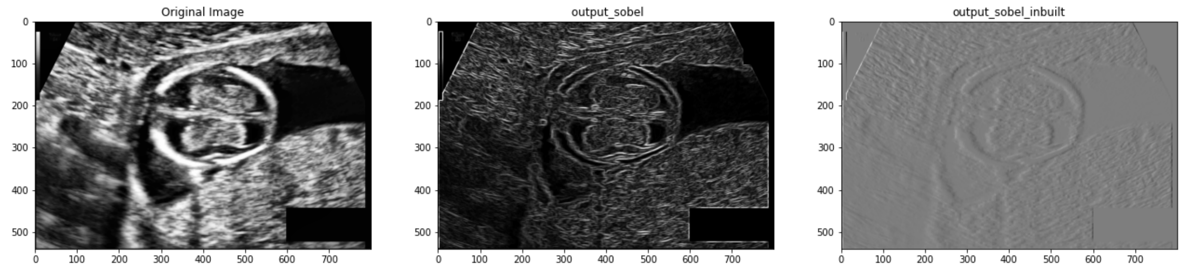

Figure 14: (a)Original Image, (b)Sobel Filtered image(c)Sobel filter using inbuilt

2.2.2 Prewitt Filter

The Prewitt operator measures a 2-Dimensioanal spatial gradient on an image and gives more attention on regions of high spatial gradient corresponding to edges. Unlike sobel filter it does not emphasize the middle pixel. There are various pairs of Prewitt operator such as 33, 5x5 convolution kernels.Prewitt mask is defined as shown in 15.

$$g\_x = \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix} \qquad g\_y = \begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix}$$

Figure 15: Prewitt Kernel

The pseudo code algorithm is discussed as follows:
Step1: Accept the sample input image.
Step2: Perform masking on given image.
Step3: Apply algorithm and the gradient.
Step4: On the input image perform mask manipulation in both the directions.
Step5: Find the absolute magnitude of the gradient. The absolute magnitude is the output edge.
The result of implementing Prewitt filter on contrast enhanced mage is as shown in figure 16

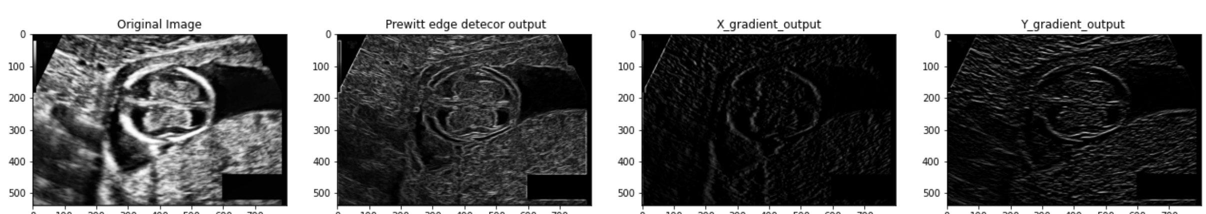

Figure 16: (a)Original Image, (b)Prewitt edge detector output (c)$X_{g}radientoutput(d)Y_{g}radientoutput$

2.2.3 Laplacian Filter

Unlike the Sobel edge detector, the Laplacian edge detector uses only one kernel. It calculates second order derivatives in a single pass.It is second order differential filter so it is extremely sensitive to noise.The mask of Laplacian operator is as shown in figure 17

0	-1	0
-1	4	-1
0	-1	0

Figure 17: Laplacian kernel

The pseudo code algorithm is discussed as follows:
Step1: Accept the sample input image.
Step2: Perform Gaussian smoothing for noise removal.
Step3: Perform masking on given image.

The result of implementing Laplacian filter on contrast enhanced mage is as shown in figure 18

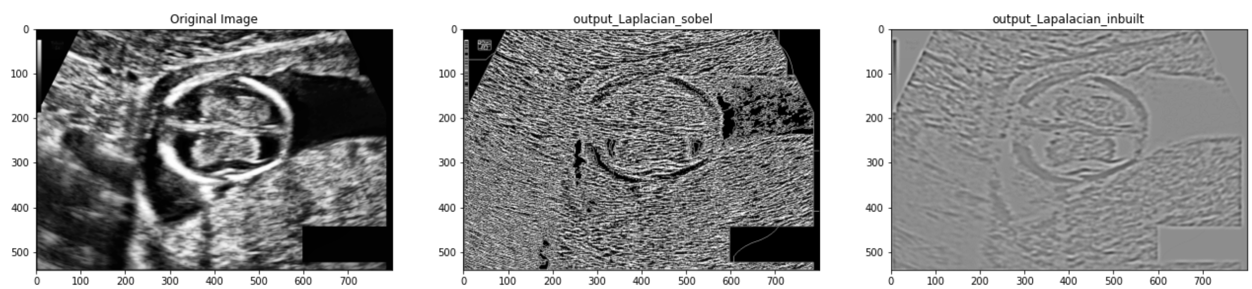

Figure 18: (a)Original Image, (b)Laplacian filtered image (c)Laplacian filtered image inbuit

2.2.4 Laplacian Of Gaussian Filter

The Laplacian is a 2-D isotropic measure of the 2nd spatial derivative of an image. The Laplacian of an image highlights regions of rapid intensity change and is therefore often used for edge detection. The Laplacian is often applied to an image that has first been smoothed with something approximating a Gaussian smoothing filter in order to reduce its sensitivity to noise, and hence the two variants will be described together.

The pseudo code algorithm is discussed as follows:
Step1: Accept the sample input image.
Step2: Perform Gaussian smoothing for noise removal.
Step3: Perform masking on given image.
step4: Find the zero crossing of smoothed image to obtain the edge detected image.
The result of implementing Laplacian of Gaussian filter on contrast enhanced mage is as shown in figure 19

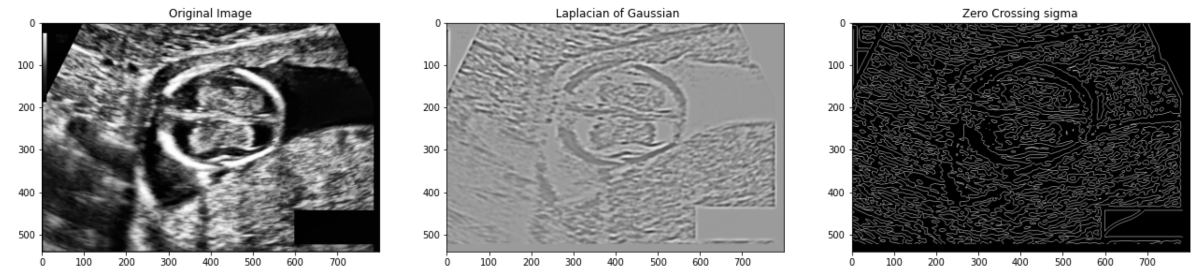

Figure 19: (a)Original Image, (b)LOG filtered image (c)Zero crossing image

2.2.5 Canny Filter

The canny edge detection algorithm was proposed to enhance the edge detection process. Three important criteria were taken into consideration for this purpose. The first and most important criterion was to detect all the important edges in the source image. This means the goal was to lower the error rate. The second criterion was that the edge points to be detected as close as possible to the true edge, also called as localization. A third criterion was not to have more than one response to a single edge. The first two were not significant enough to remove the possibility of more than one response to an edge due to which the third one was implemented.

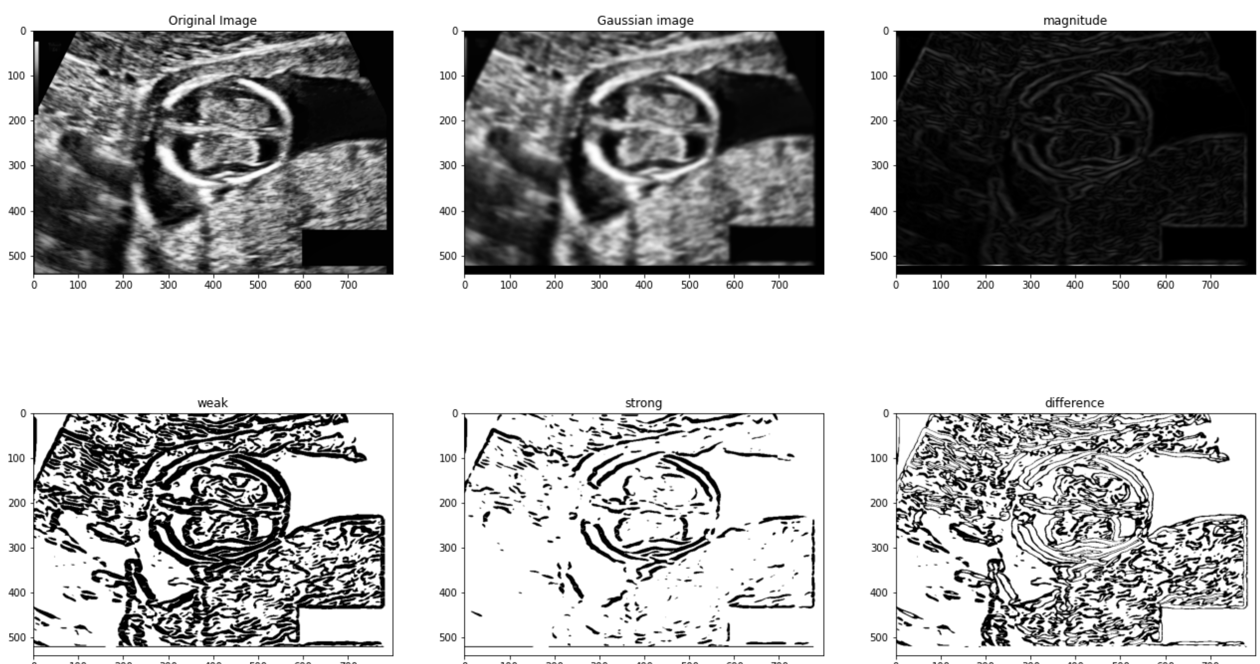

Figure 20: (a)Original Image, (b)Gaussian image (c)Magnitude (d)weak edged image (e)Strong edged image (f)Difference of weak edged and strong edged image

The pseudo code algorithm is discussed as follows:

1. Smoothing: The input image must be converted into gray scale by adjusting contrast and brightness, so that the image is blurred to remove noise. Thus, the first step is to filter out the noise in the original image to make the location and detection edges efficient. Generally a Gaussian filter is

used for noise removal .

2. Finding gradients: Edge pixels are those where there is a sharp change in gray level values, these are identified by computing the gradient of the image. The gradient is a unit vector which points in the direction of maximum intensity change. In this step first the vertical and horizontal components of the gradient are computed and then the magnitude and direction of the gradient is computed .

3. Non-maxima suppression: In this step the detector converts the thick edges in the image, based on the gradient magnitudes, to approximately thin and sharp edges which can be further used for recognition purpose. Mainly edge thinning is performed in non-maxima suppression. In this process the image is scanned along the edge direction and discards any pixel value that is not considered to be an edge which will result in thin line in the output image.

4. Double Thresholding: The threshold value consists of 2 characters, T1 = High Threshold, T2 = Low Threshold. The pixels having values of gray scale level higher than T1 are strong edge pixels, and the result is edge region. The pixels having values of gray scale level less than T2 are weak edge pixels, and the result is non-edge region. If the pixels have values of gray scale level between T1 and T2, the result is depending on the neighbouring pixels.

5. Edge tracking by hysteresis: Edges that do not connect to a very certain (strong) edge are discarded in the final output image. Strong edges are interpreted as "certain edges" and are included in the final edge image. Weak edges that are linked with strong edges are included in the output image.

The result of canny edge detected image is as shown below20 and a comparison of thinned image with inbuilt function is as shown in figure21.

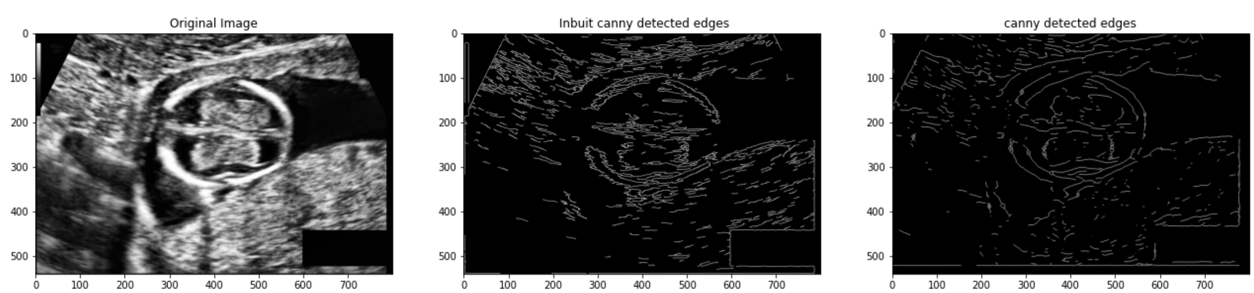

Figure 21: (a)Original Image, (b)Canny edge detector using inbuilt (c)Thinned image of strong thresholded canny edge detector

Observations: From the different edge detection technique shown above it can be interpreted that out of all the filters canny edge detector outperforms than any of the other filters and detect the fatal head circumference efficiently. Strong edge image from high threshold value detect the strongly connected edges and efficient for detecting fatal head, where as weak edged image with low threshold value results in weakly connected edges. The difference between these two results in only very weakly connected edges which should be removed for efficient edge detection. Given ground truth edge information the performance of above edge detectors can be evaluated using Pratt's Figure Of Merit (FOM), Bowyer's Closest Distance Metric (CDM),and Prieto and Allen's Pixel Correspondence Metric(PCM), Grey-scale Figure of Merit(GFOM) as described in [2].

[2]Various metrics for evaluating performance efficiency of different edge detection algorithms https://pdfs.semanticscholar.org/0e5c/4b2fd5f872e9caa36f13aee8b1ecf13b7e37.pdf.

3 Application Of Hough Transform

3.1 Estimating the head

Using iterative randomized Hough transform,the heads in an ultrasound image can be detected. Head is assumed to be elliptical shape with parameters estimated by randomized Hough transform. Preprocessing- For extracting the head ellipse from the image, the image need to be preprocessed, so that the points extracted from the ellipse would have enough points to detect the edges. Flow chart of the image preprocessing as shown in figure
22

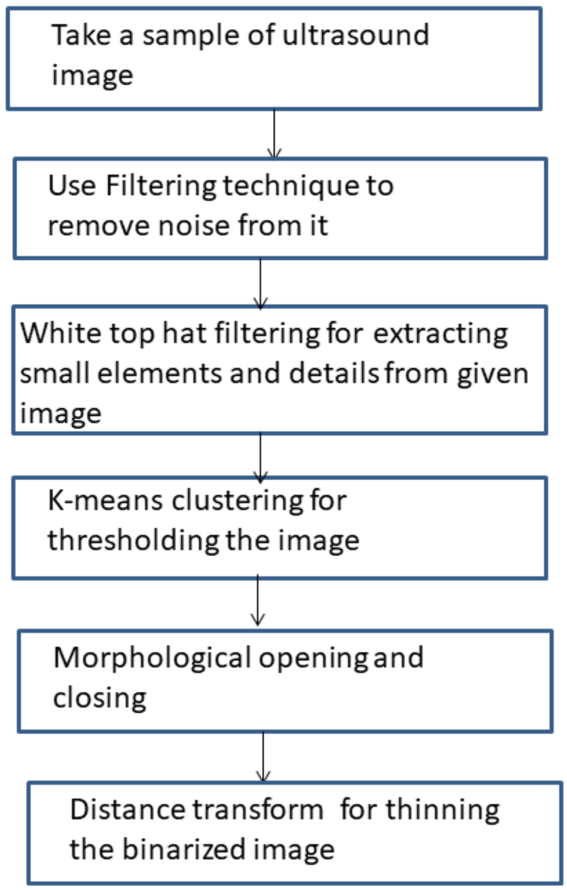

Figure 22: Flow chart for preprocessing

Algorithm for preprocessing-

1. First the image is filtered using bilateral filtering to remove noise.
2. Then the filtered image is processed using white top hat filtering to extract small elements and details from the image.
3. K-means is computed for finding out the threshold value depending on cluster centers.
4. Morphological opening and closing is done to maintain the continuity of edges and remove the small hole parts in edges.
5. Distance transform is performed for thinning and medial technique for skeletonize it.

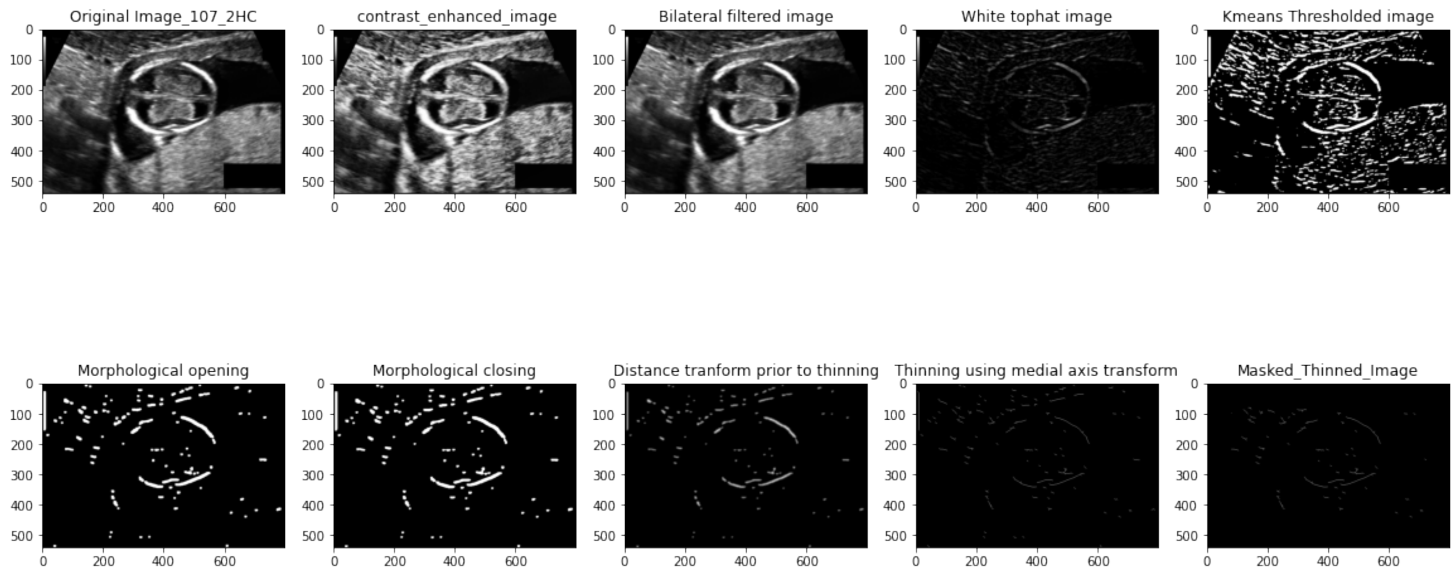

Figure 23: row 1 (a)Original Image, (b)Contrast enhanced image, (c)Bilateral filtered image, (d)White top head image, (e)K means threshold ed image,
row 2 (a)Morphological opening, (b)Morphological closing (c)Distance transform prior to thinning, (d)Thinning using medial axis transform (e)Masked thinned image

In preprocessing, the bilateral filter image, Morphological opening and closing kernels, Threshold value after k means regulate the head detection efficiency.

Algorithm for Fetal head(ellipse detection)-

1. First the all pixels of pre-processed image is converted to an 1-D vector. 3 pixels are sampled randomly from these extracted pixels.

2. .From the points choosen centre,axis and angles are calculated using ellipse equation and validity of these parameters are checked.If the ellipse detected is valid then its stored in the accumulator else go to step 1 to choose another random 3 points.

3. The steps are repeated for a specific number of iterations so that most of the points sampled defined by user.An the peaks of accumulator values are choosen depending on score value.

4. The detected curve is verified with intersection with original image.For this another loop is executed depending on score value and axis constraints.The most fitted curve is choosen for head detection.

Flow chart for head circumference detection The flow chart of fatal head detection in ultrasound images is referred from [3]

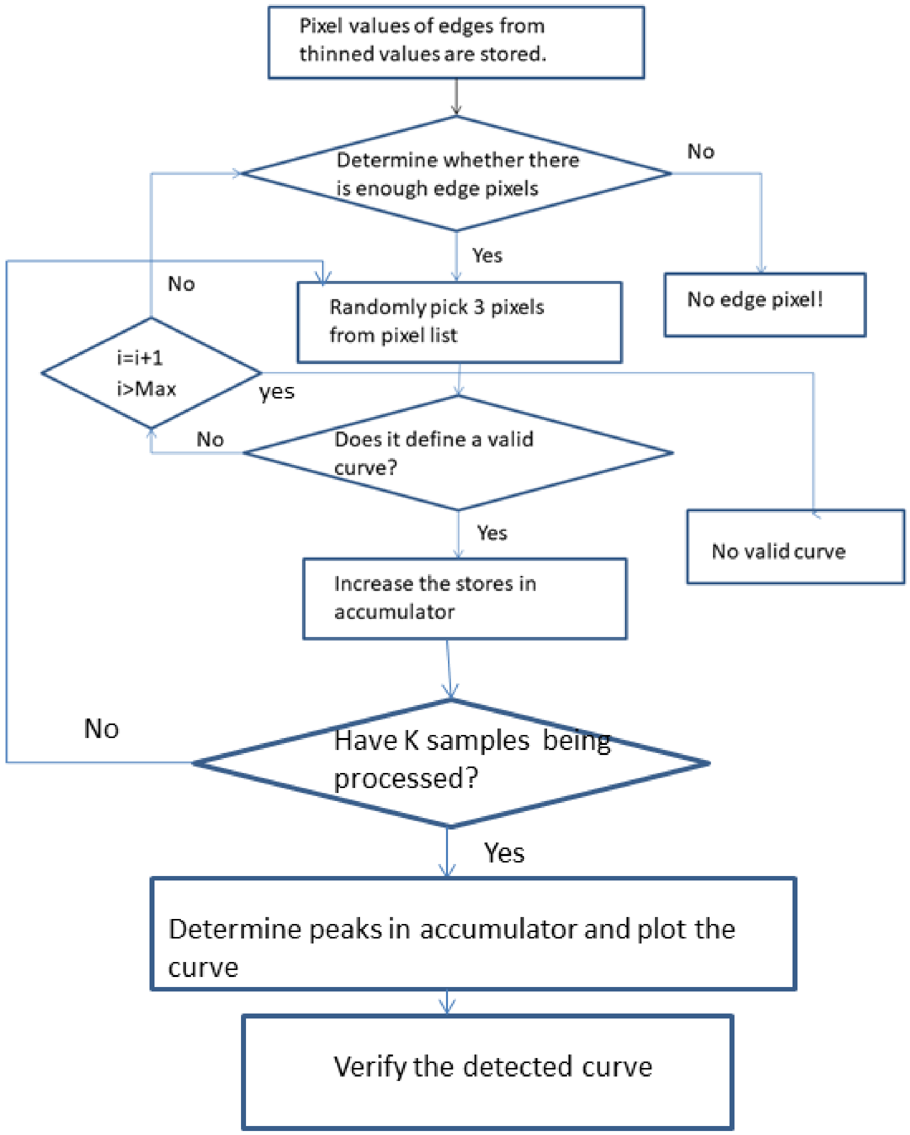

Figure 24: Flow chart for head circumference detection

[3]AUTOMATED FETAL HEAD DETECTION AND MEASUREMENT IN ULTRASOUND IMAGES BY ITERATIVE RANDOMIZED HOUGH TRANSFORM https://www.sciencedirect.com/science/article/abs/pii/S0301562905001742

Different Ultrasound images and their fatal head detection-:

1. 107- Head detection of $107-HC$ and $107-2HC$ ultrasound images given is as shown in 25

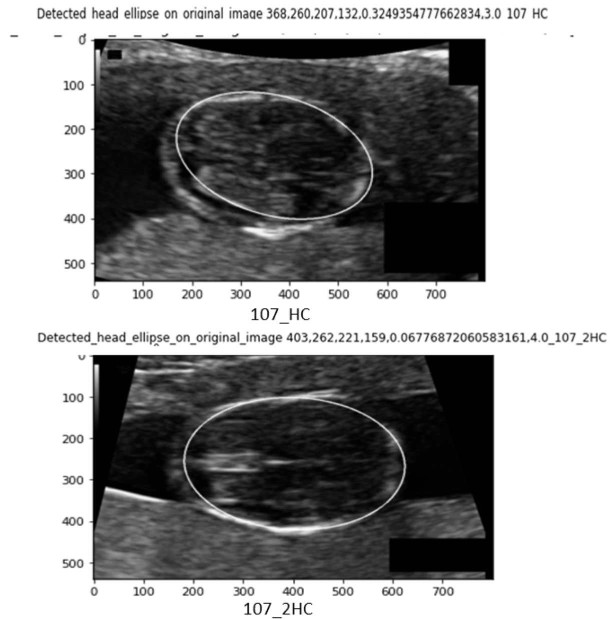

Figure 25: Head detection of $107-HC$ and $107-2HC$ ultrasound images

2. 108-Head detection of $108-HC$, $108-2HC$ and $108-3HC$ ultrasound images given is as shown in 26

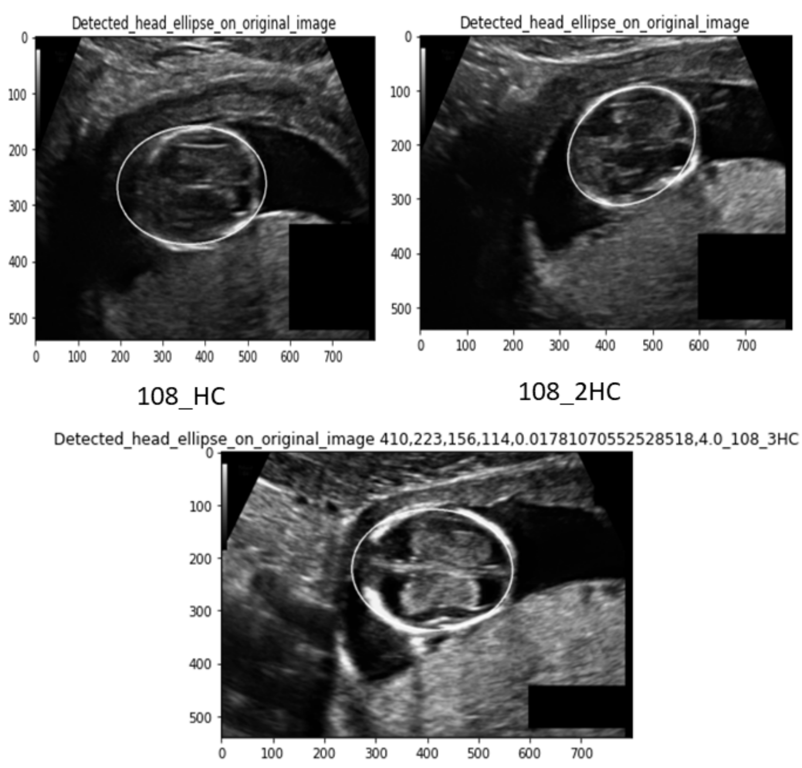

Figure 26: Head detection of $108-HC$, $108-2HC$ and $108-3HC$ ultrasound images

3.2 Comparison of measured head circumference with ground truth

The ellipse detected for given 5 different images are as follows The performance efficiency of the detected head circumference is evaluated using following metrics.
Difference - It is basically the difference of detected ellipse and the given ground truth.Figure 27 shows the comparison of detected head circumference ellipse with ground truth. Ellipse with white colour is the detected one and black one is the given ground truth.

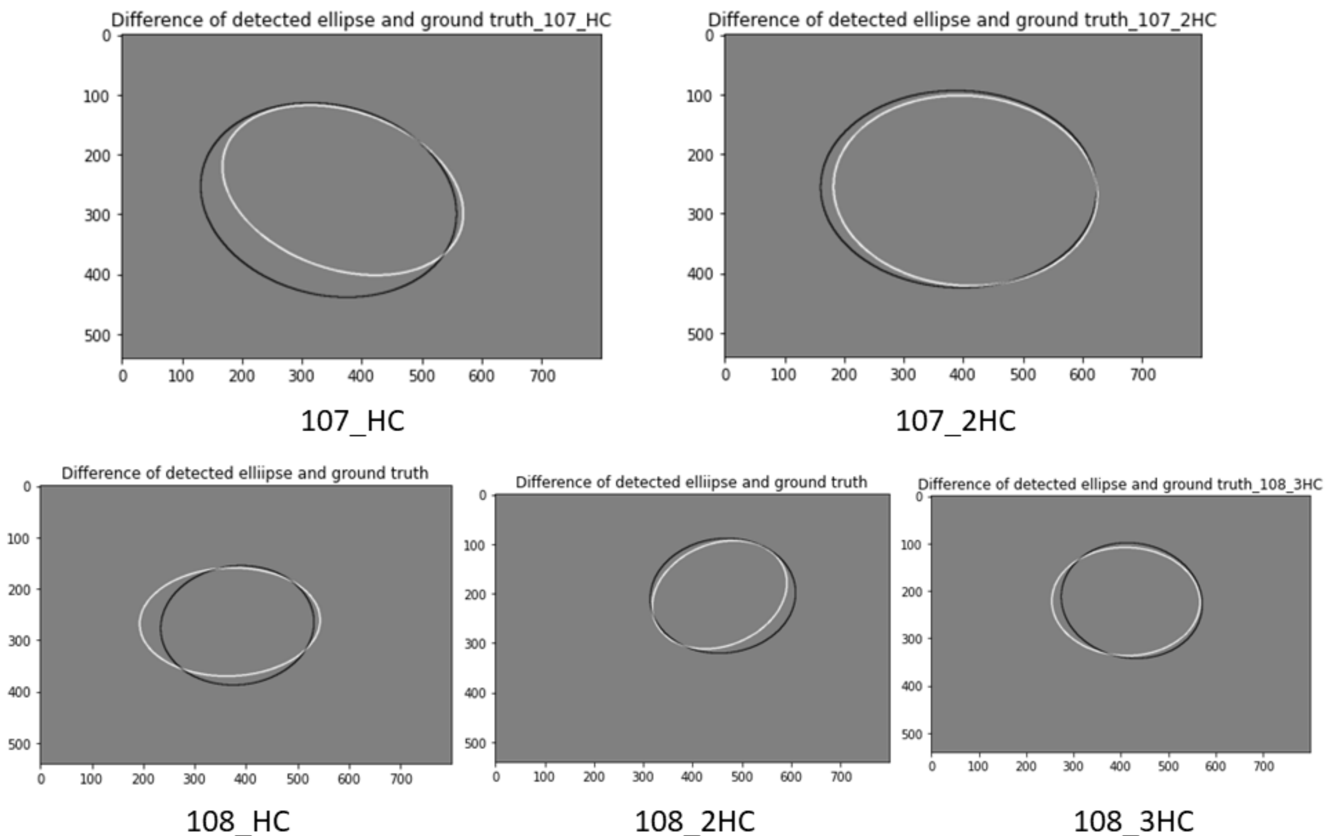

Figure 27: comparison of detected head circumference ellipse with ground truth by estimating the difference.White ellipse-detected head circumference, Black ellipse- Ground truth

Absolute Difference -It shows the difference between detected ellipse and given ground truth without any negative values.Figure 28 shows the comparison of detected head circumference ellipse with ground truth. Ellipse with white colour is the detected one and black one is the given ground truth.

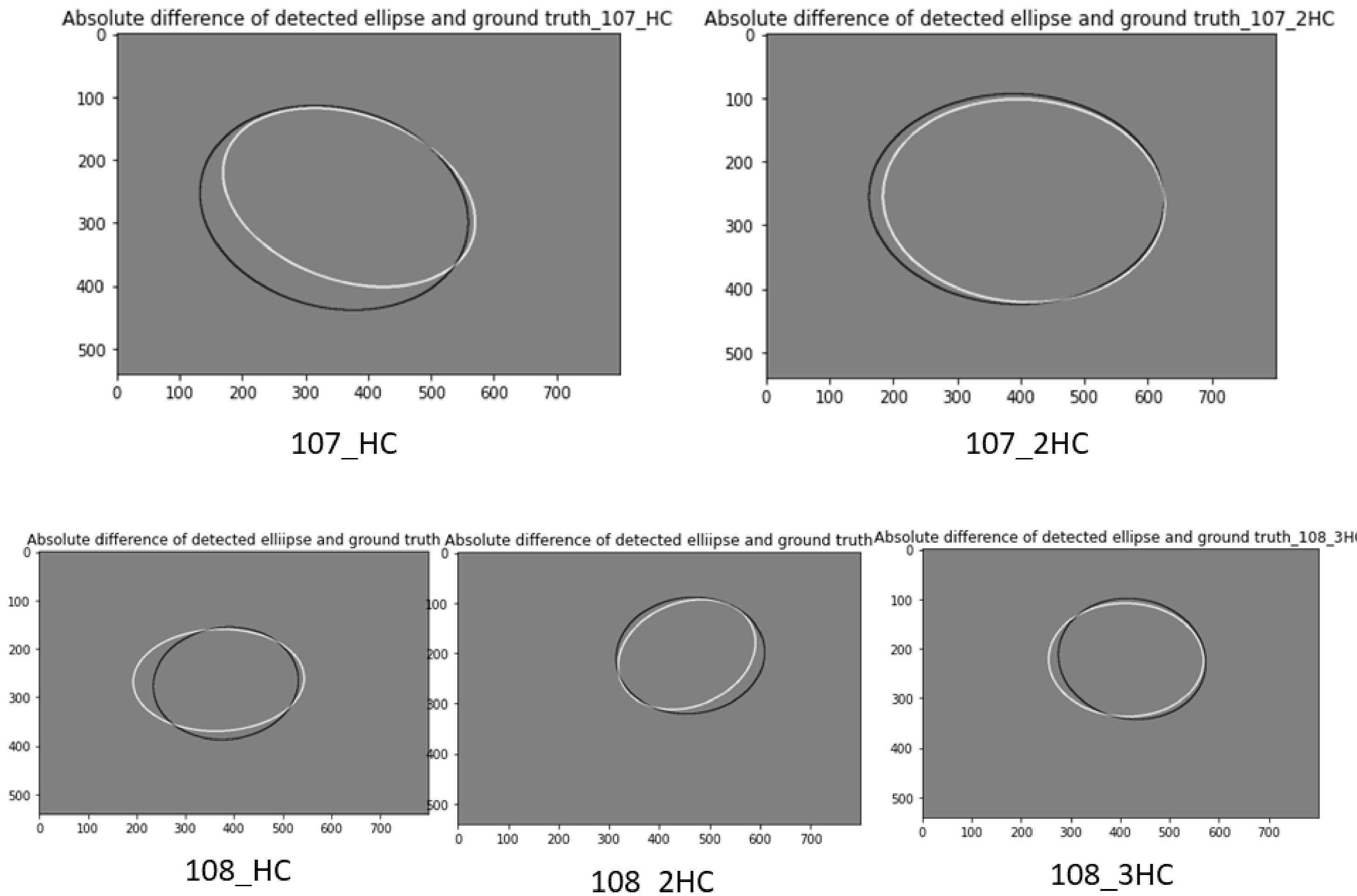

Figure 28: Comparison of detected head circumference ellipse with ground truth by estimating the absolute difference.White ellipse-detected head circumference, Black ellipse- Ground truth

Hausdorff distance - Hausdorff distance measures how far two subsets of a metric space are from each other. Informally, it is the greatest of all distances from a point in one set to the closest point in the other set. Two sets are "close" if for any one point on either set, the nearest point in the other set is "not too far".It Computes the directed Hausdorff distance between two N-D arrays.Distances between pairs are calculated using a Euclidean metric.
Dice similarity coefficient-The Dice similarity coefficient, also known as the Sorensen–Dice index or simply Dice coefficient, is a statistical tool which measures the similarity between two sets of data.The equation for the concept is,

$$Dicesimilarity = 2 * \frac{(\mid X \mid \cap \mid Y \mid)}{(\mid X \mid + \mid Y \mid)} \tag{8}$$

Calculation of Hausdorff distance and dice similarity for ultrasound images given is as shown in figure 29

Images	Parameters	Hausdorff distance	Dice similarity coefficient
107.HC	Centre = (368,260) Axis = (207,132) Angle = 0.03249354777662834 Accumulator Rank=3.0	(8.12403840463595,401,149)	0.05460750853242321
107.2HC	Centre = (403, 262) Axis = (221, 159) Angle = 0.0677687206058316 Accumulator Rank=4.0	(6.48074069840786, 419, 94)	0.08063076465337697
108.HC	Centre = (368,265) Axis = (175,105) Angle =-0.03850767 Accumulator Rank=6	(6.40312,367,169)	0.07129
108.2HC	Centre = (454,203) Axis = (141,103) Angle =-0.39002181759 Accumulator Rank = 5	(7.483314773547,311,320)	0.0949977
108.3HC	Centre =(410,223) Axis = (156,114) Angle =0.01781070552528518 Accumulator Rank= 4	(4.58257569495884,112,102)	0.0427441761060055

Figure 29: Metrics Calculation for head detection using iterative hough transform

Observations: The head detection algorithm highly dependent the pre processing step. The sigma values for bilateral filtering, kernel window size for morphological operation and the threshold values for binarization affect the head detection a lot.The highest score value in accumulator may not always detect the right ellipse as shown in figure 30. For this$(108 - 2Hc)$ the center is (346,26) and major-axis 120 and minor-axis 21 with angle of rotation -0.0068783099, score 5. So cross check with the given image or ground truth should be done for efficient detection.

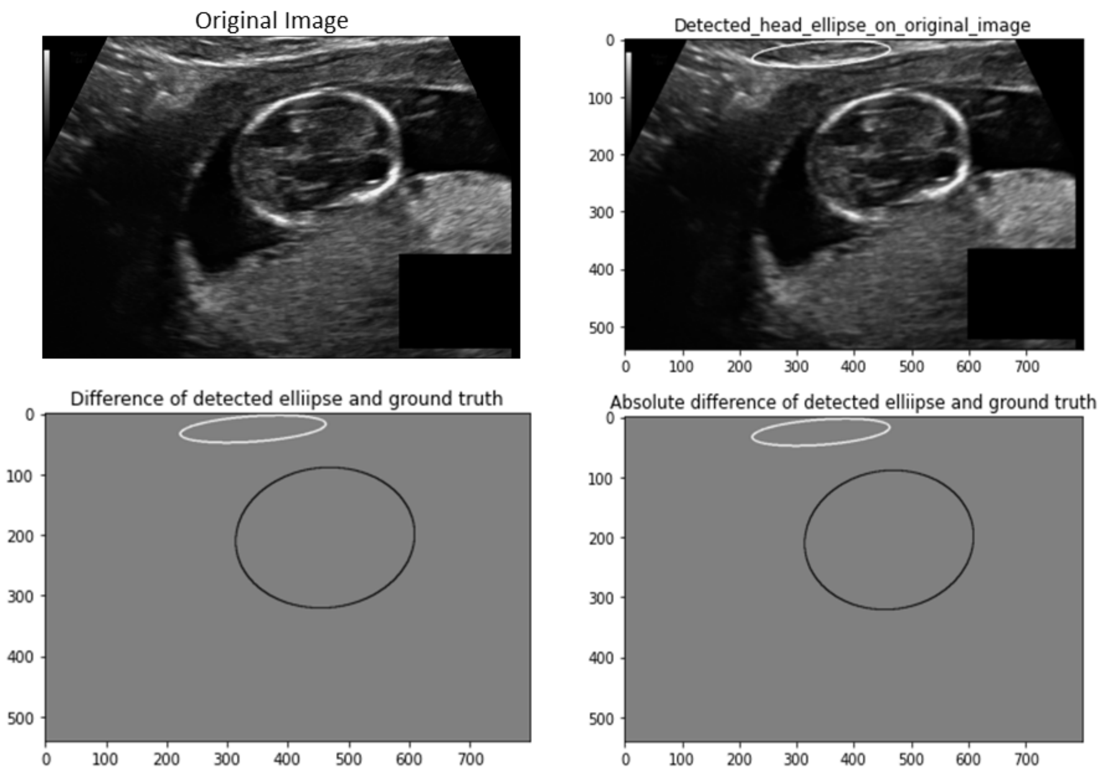

Figure 30: wrong ellipse detection in 108-2HC ultrasound image.

More score value always does not imply detection of correct ellipse as shown in figure 31. So introducing the number (N) of pixels on the detected ellipse, and the detected ellipse with the maximal number of pixels on the ellipse, which is selected from the top-M peaks in the accumulators of the whole detected ellipse samples, is accepted as the result on each iteration for updating the region of interest as proposed Iterative random hough transform in [4]in may solve this preoblem.

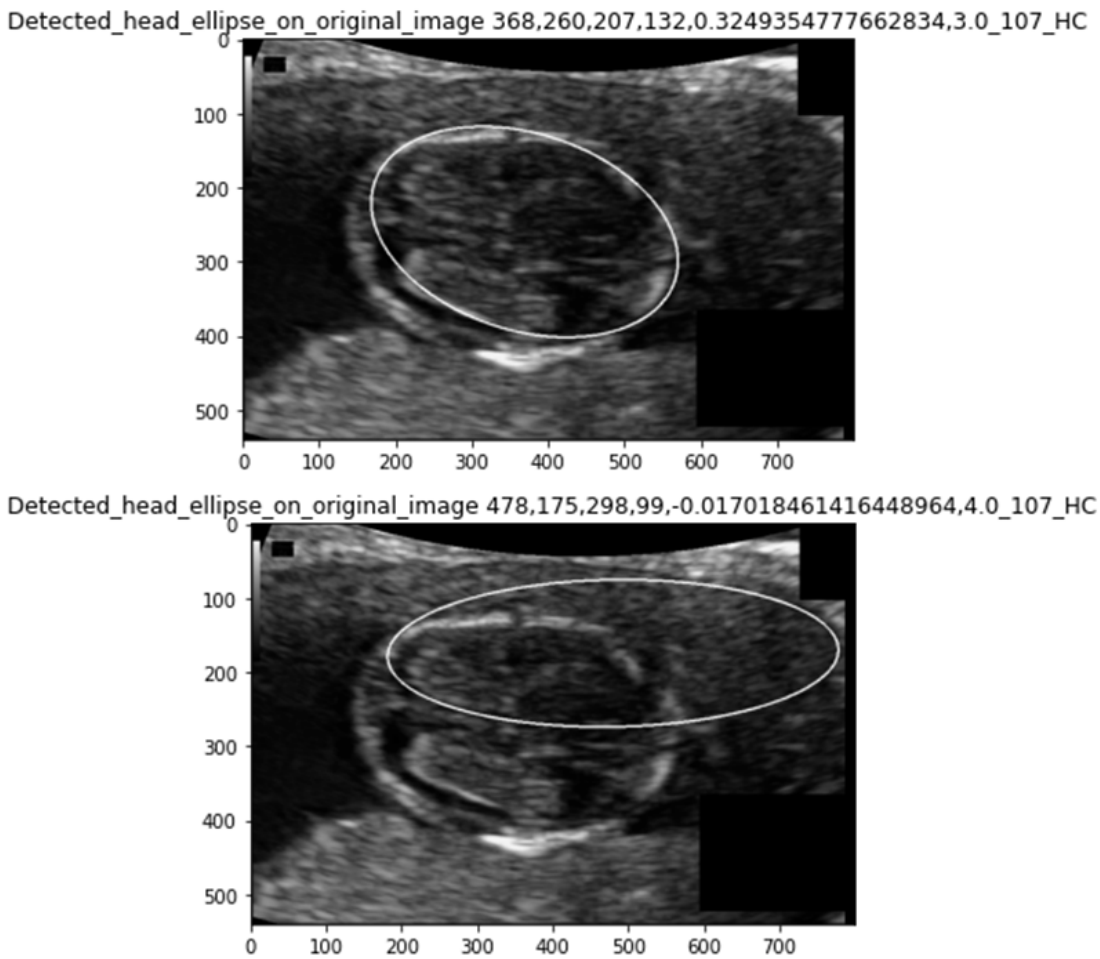

Figure 31: Comparison of ellipse detection based on score value showing higher score value does not always result in better head detection

After finding out ellipse parameters, head circumference (HC) can be calculated using the formula ,

$$HC = 1.62 * (BPD + OFD)^3 \tag{9}$$

where Biparietal diameter (BPD) is the minor axis, and Occipital frontal diameter() is the major axis as referred in [5]

[4]IRHT method https://www.researchgate.net/profile/Rong_Xu19/publication/303723203_Automatic_Fetal_Head_Detection_on_Ultrasound_Images_by_An_Improved_Iterative_Randomized_Hough_Transform/links/574fb8df08aef199238efc9b/Automatic-Fetal-Head-Detection-on-Ultrasound-Images-by-An-Improved-Iterative-Randomized-Houg pdf

[5]IRHT method https://radiopaedia.org/articles/head-circumference?lang=us