iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 2 **Issue:** IV **Month of publication:** April 2014

DOI:

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

ISSN: 2321-9653

INTERNATIONAL JOURNAL FOR RESEARCH IN APPLIED SCIENCE AND ENGINEERING TECHNOLOGY (IJRASET)

Determination of Atrial Diastole and Systole from 2-D Echocardiographic Images Using Artificial Neural Network

Asha Vincent[#1], Reshmi Reji Jacob[*2]

Dept.EIE,Karunya University

Abstract-This project proposes a new approach to estimate the cardiac cycle phases in 2-D echocardiographic images as a first step in cardiac volume estimation. We focused on analyzing the atrial systole and diastole events. The proposed metohd is based on a tandem of image processing methods and artificial neural networks as a classifier to robustly extract anatomical information. The aforementioned approach is performed in two denoising scenarios. In the first scenario, the images are corrupted with Gaussian noise, and in the second one with Rayleigh noise distribution. A dataset of 20 images that include both normal and infarct cardiac pathologies were used. The results of the employed methods are qualitatively and quantitatively compared in terms of efficiency for both scenarios. This method allows improving the time efficiency. In this method, feature extraction was assessed for both the analyzed cardiac phases, and then the images belonging to our database were classified by the instrumentality of an ANN. The cardiac cycle phase estimation is performed in apical two-chamber long-axis 0◦ view (LAX0) of 2-D echocardiographic images. The experimental images were divided into two sets corresponding to two analyzed cardiac cycles (systole and diastole). The experimental echocardiographic images came from a blend of healthy and cardiac patients that suffer from myocardial infarction. Once artificial neural network is trained, the detection becomes very fast.

Key words-Artificial neural network, cardiac cycle, Guassian noise, Rayleigh noise distribution, echocardiographic images

I. INTRODUCTION

The incidence of chronic heart failure in humans is very high and as a result, the morbidity and mortality are also very high in all countries, regardless of the economic development level [10]. The data on mortality caused by the chronic heart failure reveal that the five-year mortality rate remains at an alarmingly high level of 50% [3]. Heart failure is, therefore, a major clinical problem. Echocardiography represents a non invasive procedure to examine the heart and the surrounding blood vessels. By echocardiography, the physicians visually inspect the four cardiac cavities (left and right atria, left and right ventricle), the inferior vena cava, the aorta, the mitral valve, the aortic valve, the tricuspid valve, and the pulmonary valve. The image of the left ventricle (LV) and the analysis of the cardiac cycle are of great importance to cardiac research and represent a valuable tool to clinically assess cardiac health.

The automatic detection of cardiac phases is a first step in the diastolic and systolic volume calculation task. Moreover, the echocardiography also allows for automatic or semiautomatic analysis based on image features. Based on this general finding, this paper proposes a new approach able to accurately estimate the cardiac cycle phases in echocardiographic images without involving any manual tracing of the boundaries for segmentation process. The purpose is to increase the recognition ability of the algorithm, so that it correctly discerns between atrial systole and atrial diastole.

The aforementioned approach is performed in two denoising scenarios. In the first scenario, the images are corrupted with Gaussian noise, and in the second one with Rayleigh noise distribution. The results obtained in both scenarios were compared and redings were tabulated. To characterize the cardiac cycle phases, the feature extraction is done with region segmentation. Finally, an artificial neural network (ANN) has been trained as a classifier for effective detection of the cardiac phases in the echocardiographic images and [4].

ISSN: 2321-9653

II. METHODOLOGY

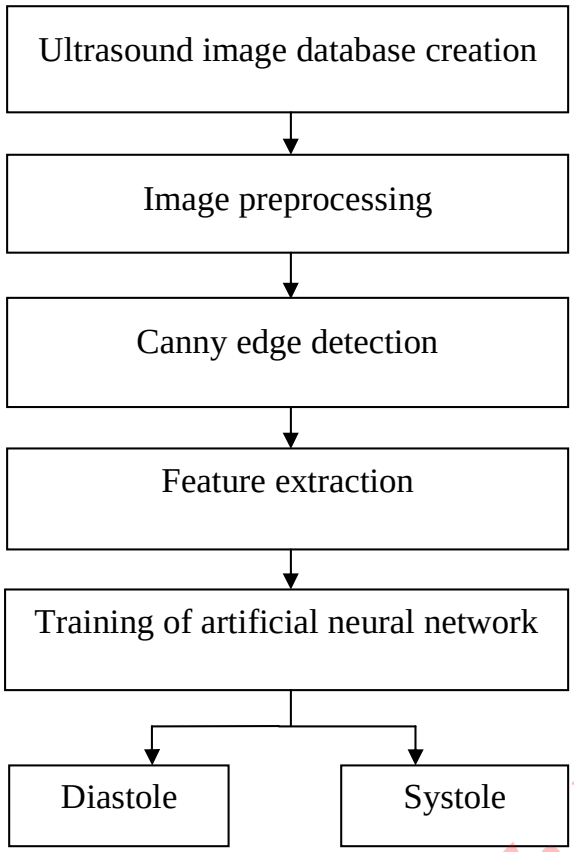


Fig 1: Flow Chart

A. *Ultrasound image database creation*

The image database was created by collecting images from hospital. Twenty echocardiographic images were collected which included normal heart and infarct cardiac pathology. The cardiac cycle phase estimation is performed in apical two-chamber long-axis 0° view (LAX0) of 2-D echocardiographic images.The image format was converted to png (portable network graphics) format.

B. *Image preprocessing*

Image processing refers to processing of a 2D picture by a computer. Image pre processing includes resizing the image, grayscale conversion filtering image enhancement, and binarization. Images are resized to reduce the size of the raw image. This makes the subsequent processing steps easier. A grayscale digital image is an image in which the value of each pixel is a single sample, that is, it carries only intensity information. Images of this sort, also known as black-and-white, are composed exclusively of shades of gray, varying from black at the weakest intensity to white at the strongest. Grayscale images have many shades of gray in between. Grayscale images are also called monochromatic.

The median filter is a nonlinear digital filtering technique, often used to remove noise. Such noise reduction is a typical pre-processing step to improve the results of later processing. Median filtering is used because; it preserves edges while removing noise. Image enhancement processes consist of a collection of techniques that seek to improve the visual appearance of an image. The image enhancement processor would emphasize salient features of the original image and simplify the processing task. Instead of simply replacing the pixel value with the mean of neighbouring pixel values, it replaces it with the MEDIAN of those values. The median is calculated by first sorting all the pixel values from the surrounding neighborhood into numerical order and then replacing the pixel being considered with the middle pixel value.

A binary image is a digital image that has only two possible values for each pixel. Typically the two colours used for a binary image are black and white though any two colours can be used. The colour used for the object in the image is the foreground colour while the rest of the image is the background colour. Binary images are also called bi-level or two-level. This means that each pixel is stored as a single bit (0 or 1). Otsu's method is used to automatically perform clustering-based image thresholding, or, the reduction of a graylevel image to a binary image. The algorithm assumes that the image to be thresholded contains two classes of pixels, then calculates the optimum threshold separating those two classes so that their intra-classvariance is minimal. In Otsu's method it exhaustively searches for the threshold that minimizes the intra-class variance i.e, the variance within the class, defined as a weighted sum of variances of the two classes:

$$\sigma_w^2(t) = w(t)\,\sigma_1^2(t) + w(t)\,\sigma_2^2(t) \qquad 4.1$$

Weights w_i are the probabilities of the two classes separated by a threshold t and σ_i^2 variances of these classes. Otsu shows that minimizing the intra-class variance is the same as maximizing inter-class variance.

$$\sigma_b^2(t) = \sigma^2 - \sigma_w^2(t) = w_1(t)\,w_2(t)[\mu_1(t) - \mu_2(t)]^2 \qquad 4.2$$

ISSN: 2321-9653

INTERNATIONAL JOURNAL FOR RESEARCH IN APPLIED SCIENCE AND ENGINEERING TECHNOLOGY (IJRASET)

which is expressed in terms of class probabilities w_i and class means μ_i. The class probability $w_1(t)$ is computed from the histogram as t:

$$w_1(t) = {}_0\sum^t p(i) \qquad 4.3$$

while the class mean $\mu_1(t)$ is:

$$\mu_1(t) = [{}_0\sum^t p(i)x(i)]/w_i \qquad 4.4$$

where $x(i)$ is the value at the centre of the t [th] histogram bin.

Algorithm:

1. Compute histogram and probabilities of each intensity level
2. Set up initial $w_i(0)$ and $\mu_i(0)$
3. Step through all possible thresholds $t=1$ maximum intensity
    1. Update w_i and μ_i
    2. Compute $\sigma_b^2(t)$
4. Desired threshold corresponds to the maximum $\sigma_b^2(t)$
5. Compute two maxima (and two corresponding thresholds). $\sigma_{b1}^2(t)$ is the greater max and $\sigma_{b2}^2(t)$ is the greater or equal maximum
6. Desired threshold = $(\text{threshold}_1+\text{threshold}_2)/2$

C. Canny edge detection

The Canny edge detector is an edge detection operator that uses a multi-stage algorithm to detect a wide range of edges in images. It is also known to many as the optimal detector, Canny algorithm aims to satisfy three main criteria: good detection good localization and minimal response .The Canny Edge Detection Algorithm runs in five separate steps:

1. Smoothing: Blurring of the image to remove noise.

2. Finding gradients: The edges should be marked where the gradients of the image has large magnitudes.

3. Non-maximum suppression: Only local maxima should be marked as edges.

4. Double thresholding: Potential edges are determined by thresholding.

5. Edge tracking by hysteresis: Final edges are determined by suppressing all edges that are not connected to a very certain edge.

1) Finding gradients

The Canny algorithm basically finds edges where the grayscale intensity of the image changes the most. These areas are found by determining gradients of the image. Gradients at each pixel in the smoothed image are determined by applying what is known as the Sobel-operator. First step is to approximate the gradient in the x- and y-direction respectively by applying the kernels as shown below.

$$K_{GX} = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$$

$$K_{GY} = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} \qquad 4.5$$

The gradien0t magnitudes, also known as the edge strengths can then be determined as a Euclidean distance measure by applying the law of Pythagoras as shown in equation 1. It is sometimes simplified by applying Manhattan distance is measured to reduce the computational complexity as shown in equation 2. The Euclidean distance measure has been applied to the input image.

$$|G| = \sqrt{G_x^2 + G_y^2} \qquad 4.6$$

$$|G| = |Gx| + |Gy| \qquad 4.7$$

where Gx and Gy are the gradients in the x- and y-directions respectively. An image of the gradient magnitudes often indicates the edges quite clearly using equation 3. However, the edges are typically broad and thus do not indicate exactly where the edges are. To make it possible to determine this, the direction of the edges must be determined and stored.

$$\theta = \arctan(|Gy|/|Gx|) \qquad 4.8$$

2) Non-maximum suppression

The purpose of this step is to convert the blurred edges in the image of the gradient magnitudes to sharp edges. Basically this is done by preserving all local maxima in the gradient image, and deleting everything else. The algorithm is for each pixel in the gradient image:

ISSN: 2321-9653

1. Round the gradient direction θ to nearest 45◦, corresponding to the use of an 8-connected neighbourhood.

2. Compare the edge strength of the current pixel with the edge strength of the pixel in the positive and negative gradient direction, i.e. if the gradient direction is north (theta = 90◦), compare with the pixels to the north and south.

3. If the edge strength of the current pixel is largest; preserve the value of the edge strength. If not, suppress the value.

3) Double thresholding

The edge-pixels remaining after the non-maximum suppression step are marked with their strength pixel-by-pixel. Many of these will probably be true edges in the image, but some may be caused by noise or colour variations for instance due to rough surfaces. The simplest way to discern between these would be to use a threshold, so that only edges stronger that a certain value would be preserved. The Canny edge detection algorithm uses double thresholding. Edge pixels stronger than the high threshold are marked as strong; edge pixels weaker than the low threshold are suppressed and edge pixels between the two thresholds are marked as weak.

4) Edge tracking by hysteresis

Strong edges are interpreted as certain edges, and can immediately be included in the final edge image. Weak edges are included if and only if they are connected to strong edges. The 5 logic is of course that noise and other small variations are unlikely to result in a strong edge. Thus strong edges will only be due to true edges in the original image. The weak edges can either be due to true edges or noise/colour variations. The latter type will probably be distributed independently of edges on the entire image, and thus only a small amount will be located adjacent to strong edges. Weak edges due to true edges are much more likely to be connected directly to strong edges. Edge tracking can be implemented by BLOB-analysis (Binary Large Object). The edge pixels are divided into connected BLOB's using 8-connected neighbourhood. BLOB's containing at least one strong edge pixel is then preserved, while other BLOB's are suppressed.

D. *Feature extraction*

Region growing segmentation is an approach to examine the neighboring pixels of the initial "seed points" and determine if the pixels are added to the seed point or not.

Step1. Selecting a set of one or more starting point (seed) often can be based on the nature of the problem.

Step2. The region is grown from these seed points to adjacent point depending on a threshold or criteria we make.

Step3. Region growth should stop when no more pixels satisfy the criteria for inclusion in that region

Boundary tracking technique starts with scanning the image for the pixel with the highest gradient. This pixel is for sure on the object's boundary. Then a 3-by-3 pixel segment (with the original pixel in its center) is used as tracking probe for the next pixel with the highest gradient, in the neighbor of the original one. This pixel is considered part of the object's boundary and the probe is moved to the new pixel to search for another boundary pixel. The process is repeated until a closed contour is formed. If, at any time, three adjacent pixels in the probe have the same highest gray level (i.e., gradient), the one in the middle is selected. If two have the same highest gradient, the choice is arbitrary.

E. *Artificial neural network training*

Artificial neural networks are based on approximate models of the brain. The basic building block of an artificial neural network is the neuron. The brain in made up of about 100 billion neurons, with an average of 1,000 to 100,000 input connections per neuron. When many of these neurons are combined they have the properties of a massively parallel super-computer.

In multi-layer neural nets, each neuron is connected to other neurons via a weighted communication line. The weights of the connections are adjusted in training to represent the knowledge of the neural network. One method for adjusting these weights is with a training algorithm.

Neural networks are good when dealing with abstract problems, like those based on features and patterns. Artificial neural networks are mainly used in two areas: feature detection and pattern mapping. Feature detection is done by classifying an unknown input pattern, by comparison to previously learned patterns. Neural networks are a new way of performing complex tasks. They have several properties which are advantageous.

• Feed forward neural networks have a fixed computation time,

ISSN: 2321-9653

• Computation speed is very high, as a result of the parallel structure,

• Fault tolerant, because of distributed nature of network knowledge,

• Learns general solutions of presented training data,

• Neural networks eliminate the need to develop an explicit model of a process.

•Neural networks can model parts of a process that cannot be modelled or are unidentified.

• If an explicit mathematical model is not required, then the network can be 'programmed' in a fraction of the time required for traditional development.

• A neural network can learn from noisy and incomplete data, the solution will just be less precise.

• Ability to generalize to situations not taught to network previously.

• Can be taught to compensate for system changes from initial training model.

The backpropagation learning algorithm was made widely popular by Rummelhart. Learning with backpropagation consists of determining the proper set of connection weights to estimate a given training set. A training set consists of expected outputs for specific inputs. The learning process involves i) solving the network for a set of inputs, ii) comparing the outputs to the expected values, and then iii) using the errors to estimate a correction to each weight value in the network.

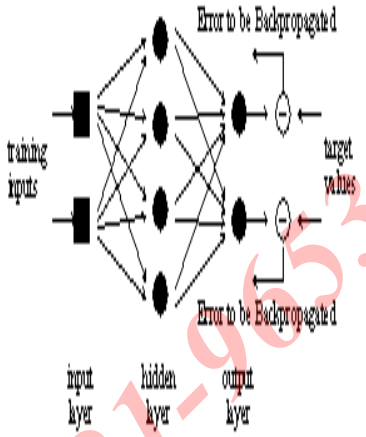


Fig 2 Backpropagation of errors

The training process is repeated iteratively until the network has closely matched its outputs with the training set. This is known as convergence. A trained network will have the property of generalization. This property may be evaluated by testing the network with a data set which is similar, but non-intersecting with the training set. If the results for the test set are the same as with the training set, then the network may be said to have generalized. If the network has converged, but has not generalized, then the network may be said to have memorized the training set. If the network generalizes, then it should be able to handle any problem that is similar to the training set.

The back propagation learning rule is simply a gradient descent algorithm. It minimizes the squares of the differences between the actual and desired outputs, summed over the output neurons for all training examples. The initial state in the network has a random set of connection weights. This is because, when a system starts with all connection weights being equal, the network begins at a type of local optimum, and will not converge.

The rule for modifying the connection weights for a single neuron is called the delta rule. The weights (wij) on each input should change by an amount (Δpwij) which is proportional to the error signal, δpj, and the input signal of the neuron (opi).

ISSN: 2321-9653

$$\Delta_p w_{ij} = \eta \, \delta_{pj} \, o_{pi} \qquad 4.9$$

'Δpwij' represents the change that should be made to the connection weight for the link between neuron 'ui' and neuron 'uj' for the input pattern 'p'. 'η' is the constant of proportionality usually called the learning rate, 'δpj' is the delta of neuron 'uj', this would just be the output error for a single neuron. 'opi' is output of preceding neuron 'ui' (or the input to neuron 'uj').

For a one-layered network, the change in connection weights can easily be calculated since the difference, or 'δ' of the output neurons is readily available (i.e., output error). With the introduction of hidden layers, the desired outputs of these hidden neurons are difficult to estimate. In order to compute the delta of a hidden neuron, the error signal from the output layer must be propagated backwards to all preceding layers. This is for influencing the modification of the connection weights leading into a hidden neuron.The delta value for any output neuron is computed as,

$$\delta_{pj} = (t_{pj} \, . \, o_{pj}) \, f'(net_{pj}) \qquad 4.10$$

where, 'tpj' is the desired, or target output of output neuron 'uj'. 'opj' is the actual output of output neuron 'uj', and 'f ′(netpj)' is the first derivative of the activation function for the given input pattern 'p' evaluated at neuron 'uj'.

For the hidden neurons, the deltas are calculated from the previously calculated deltas, found in subsequent layers. That is, the deltas found in the output layer must be propagated backwards through the connection weights so that an appropriate error signal can be estimated for each hidden neuron. The error signal, or delta, for the hidden neurons is,

$$\delta_{pj} = f'(net_{pj}) \sum \delta_{pk} w_{jk} \qquad 4.11$$

where 'f ′(netpj)' is the first derivative of the activation function with respect to the total output (netpj) evaluated at hidden neuron 'uj'. 'δpk' is the delta value for the subsequent neuron 'uk'. 'wjk' is the connection weight for the link between hidden neuron 'uj' and subsequent neuron 'uk'. 'δpj' is the delta for hidden neuron 'uj'.

III. RESULTS AND DISCUSSION

A. *INPUT IMAGE*

Twenty echocardiographic images were collected which included normal heart and infarct cardiac pathology. The images were acquired using the echomachine of Philips. They were obtained in DICOM format. For our purpose it was converted to PNG format. Image preprocessing step includes resizing the image to 256*256 images, grascale conversion and filtering.Image resizing is done using imresize .imresize computes the number of rows or columns automatically to preserve the image aspect ratio.

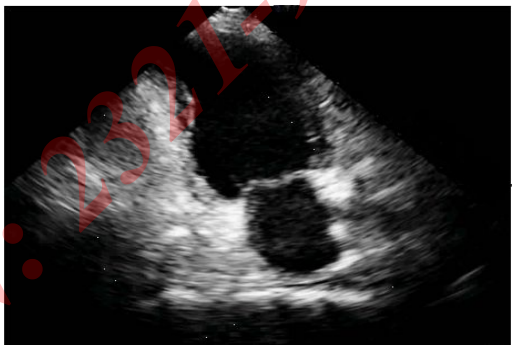


Fig 3 Input image

Grayscale conversion is done using rgb2gray. It converts the true colour image to the grayscale intensity image. Filtering is done using median filter. Median filtering is a nonlinear operation often used in imageprocessing to reduce salt and pepper noise. A median filter is more effective than convolution when the goal is to simultaneously reduce noise and preserves edges. medfilt2 is used here.

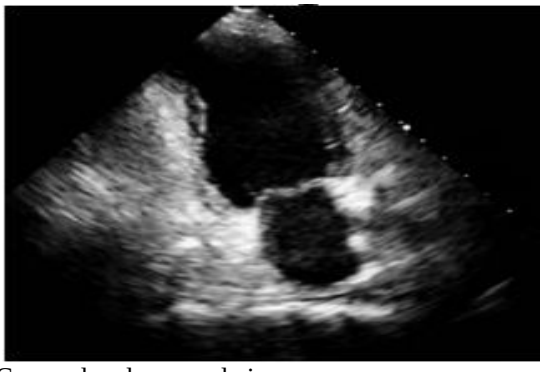


Fig 4 Cropped and grayscale image

ISSN: 2321-9653

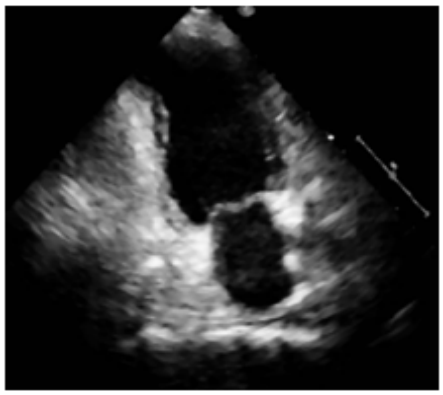


Fig 5 Median filtered image

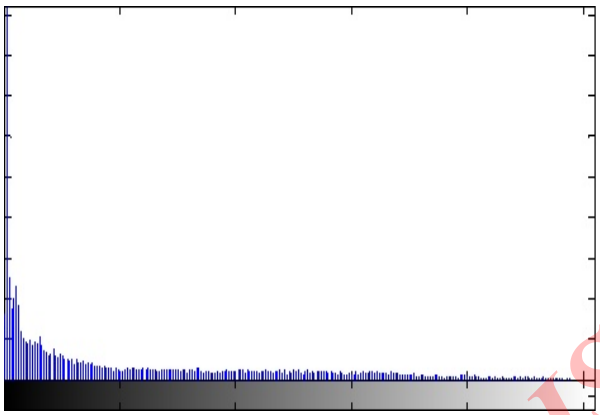


Fig 6 Histogram of filtered image

B. ENHANCED IMAGE

For image enhancement, imadjust maps the intensity values in grayscale image such that 1% of data is saturated at low and high intensities. This increases the contrast of the output image.

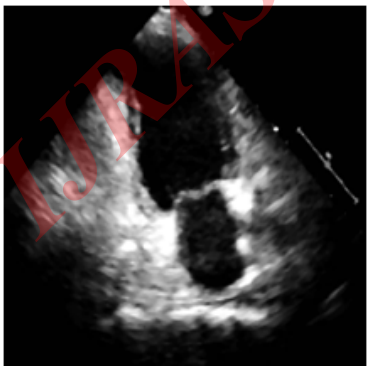


Fig 7 Enhanced image

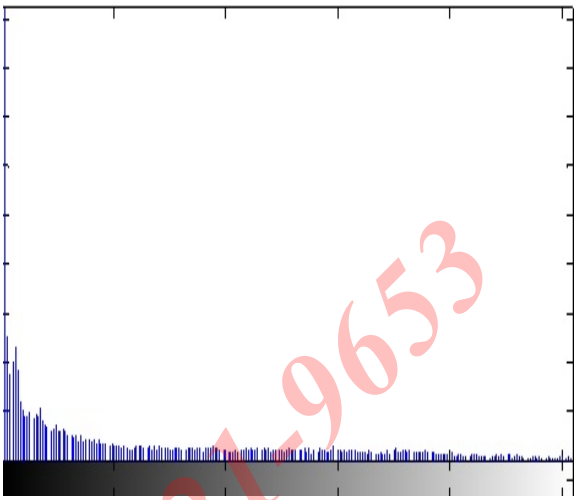


Fig 8 Histogram of enhanced image

C. BINARIZED IMAGE

level = graythresh(I) computes a global threshold (level) that can be used to convert an intensity image to a binary image with im2bw. level isa normalized intensity value that lies in the range [0, 1]. The graythresh function uses Otsu's method,which chooses the threshold to minimize the intraclass variance of the black and white pixels.

Fig 9 Otsu binarized image

D. EDGE DETECTED IMAGE

The edge detection is done using canny edge detector and show in figure below.

ISSN: 2321-9653

Fig 10 Edge detected in binary image

E. GRADIENT IMAGE

[FX,FY] = gradient(F) where F is a matrix returns the x and y components of the two-dimensional numerical gradient. FX corresponds to ∂F/∂x, the differences in x (horizontal) direction. FY corresponds to ∂F/∂y, the differences in the y (vertical) direction. The spacing between points in each direction is assumed to be one.Gradient calculated images are shown in figure below.

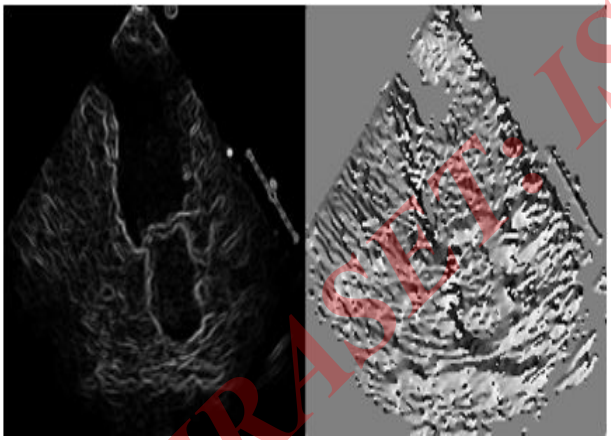


Fig 11 Gradient direction and magnitude

TABLE I
CALCULATION OF GRADIENT MAGNITUDE AND DIRECTION

Sl.no	Gradient calculations	
	G_{mag}	G_{dir}
Dab1	885.3406	179.8031
Dab2	894.7771	179.8488
Dab3	856.1227	179.8921
Dab4	825.4090	179.7846
Dab5	763.6177	179.8843
Dn1	954.9335	179.8949
Dn2	940.9665	179.7771
Dn3	966.9674	179.8258
Dn4	881.3081	179.7870
Dn5	883.3040	179.6610
Sab1	882.3208	179.7886
Sab2	862.0557	179.7454
Sab3	867.7200	179.8645
Sab4	912.0022	179.7431
Sab5	906.0199	179.8578
Sn1	872.0745	179.8084
Sn2	975.5788	179.6155
Sn3	862.5196	179.7205
Sn4	980.7028	179.6397
Sn5	747.1881	179.8439

F. NON MAXIMUM SUPPRESSED IMAGE

Given estimates of the image gradients, a search is carried out to determine if the gradient magnitude assumes a local maximum in the gradient direction. At every pixel, it suppresses the edge strength of the center pixel (by setting its value to 0) if its magnitude is not greater than the magnitude of the two neighbors in the gradient direction.

Fig 12 Non maximum suppressed image

G. DOUBLE THRESHOLDED IMAGE

Edge pixels stronger than the high threshold are marked as strong; edge pixels weaker than the low threshold are suppressed and edge pixels between the two thresholds are marked as weak.

ISSN: 2321-9653

Fig 13 Double thresholded images

H. HYSTERESIS RESULT

Strong edges are interpreted as "certain edges", and can immediately be included in the final edge image. Weak edges are included if and only if they are connected to strong edges. The logic is of course that noise and other small variations are unlikely to result in a strong edge. Thus strong edges will only be due to true edges in the original image. The weak edges can either be due to true edges or noise/colour variations. The latter type will probably be distributed independently of edges on the entire image, and thus only a small amount will be located adjacent to strong edges. Weak edges due to true edges are much more likely to be connected directly to strong edges.

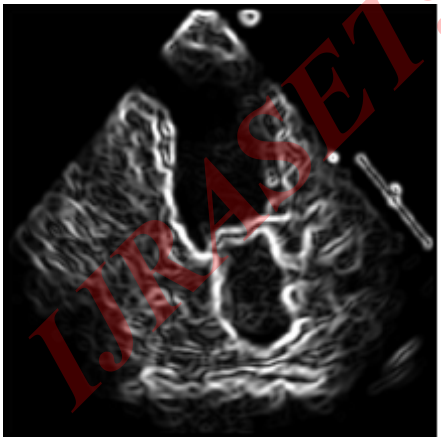


Fig 14 Hysteresis Result

I. FEATURE EXTRACTION

For feature extraction the countour segmentation was used.the region segmentation was done for 500 iterations.

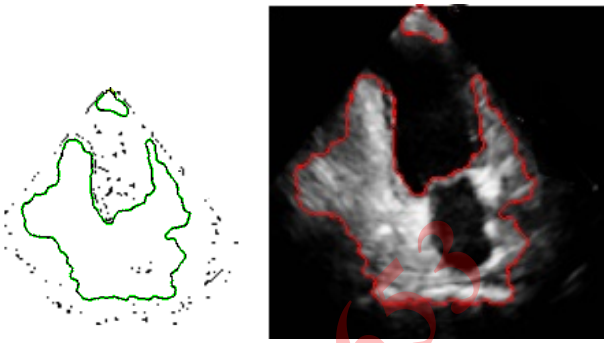


Fig 15 Feature extracted image

TABLE II
READINGS FOR FEATURE EXTRACTION

Sl.no	Height	Width
Dab1	180.7005	171.4801
Dab2	182.6589	170.8836
Dab3	182.1923	170.8999
Dab4	181.1089	170.6298
Dab5	153.0712	111.9176
Dn1	185.3866	166.1864
Dn2	186.6004	169.4167
Dn3	185.5817	166.4875
Dn4	184.7686	168.0325
Dn5	183.8778	168.8163
Sab1	187.7095	168.0731
Sab2	186.8528	161.6697
Sab3	184.1577	166.4822
Sab4	185.2621	164.3348
Sab5	148.0178	127.2772
Sn1	186.2671	166.5947
Sn2	186.4225	164.6229
Sn3	187.8104	163.5185
Sn4	186.9912	162.5272
Sn5	157.6316	89.2274

J. ARTIFICIAL NEURAL NETWORK TRAINING

"logsig" is a transfer function. Transfer functions calculate a layer's output from its net input. purelin is a neural transfer function. Transfer functions calculate a layer's output from its net input. trainrp is a network training function that updates weight and bias values according to the resilient backpropagationalgorithm (Rprop). trainrp can train any network as long asits weight, net input, and transfer functions have derivative functions.

ISSN: 2321-9653

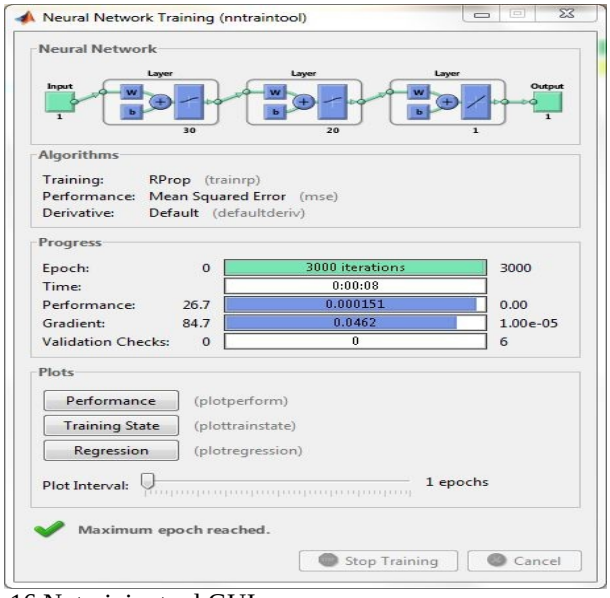


Fig 16 Nntrainingtool GUI

K. ANALYSIS IN NOISE SCENARIOS

In the first scenario, the images are corrupted with Gaussian noise, and in the second one with Rayleigh noise distribution.

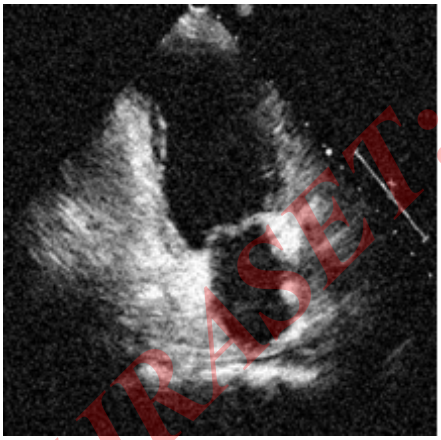


Fig 17 Gaussian noisy image

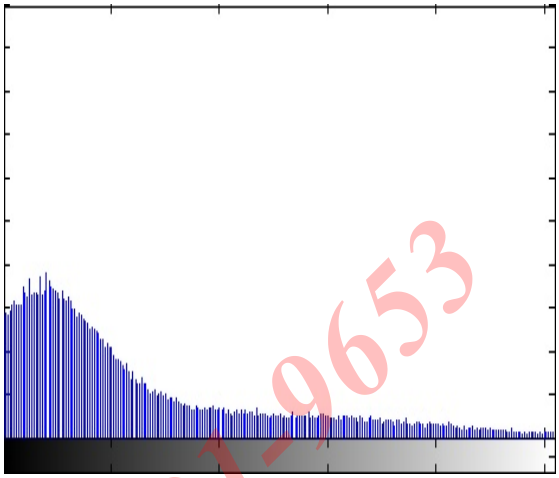


Fig 18 Histogram of Gaussian noisy image

Fig 19 Gaussian noise removed image

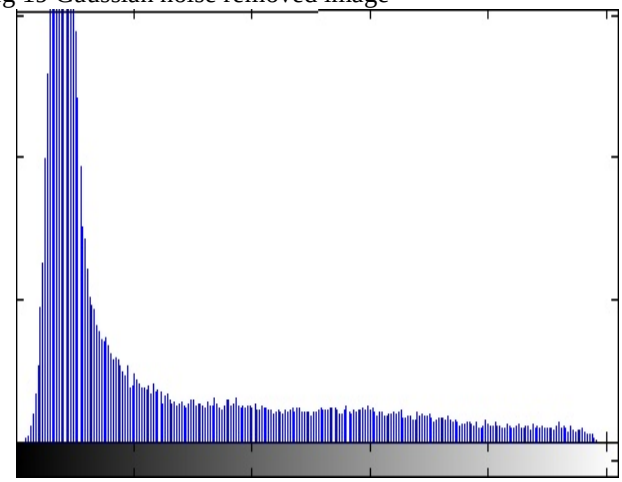


Fig 20 Histogram of Gaussian noise removed image

TABLE III

NOISE QUALITY PARAMETER CALCULATION IN GAUSSIAN NOISE SCENARIO

Sl. No	Gaussian Noise removed Image				
	SNR	PSNR	MSE	RMSE	MAE
Dab1	19.9055	47.7081	1.11089	1.05399	0.88948
Dab2	19.95803	47.70697	1.11118	1.05412	0.88998
Dab3	19.90496	47.72927	1.10548	1.05142	0.88750
Dab4	19.78850	47.71838	1.10826	1.05274	0.88762
Dab5	19.08434	47.73425	1.10422	1.05082	0.88641
Dn1	19.39926	47.71455	1.10924	1.05320	0.88954
Dn2	19.29457	47.70697	1.11118	1.05412	0.88998
Dn3	19.31236	47.75688	1.09848	1.04808	0.88434
Dn4	19.16161	47.71706	1.10860	1.05290	0.88884
Dn5	19.21422	47.77205	1.09465	1.04626	0.88258
Sab1	20.18944	47.74687	1.10101	1.04929	0.88303
Sab2	20.43706	47.72394	1.10684	1.05207	0.88708
Sab3	20.23899	47.71862	1.10820	1.05271	0.88847
Sab4	20.38549	47.72921	1.10550	1.05143	0.88751
Sab5	19.39308	47.72897	1.10556	1.05146	0.88675
Sn1	19.44574	47.72796	1.10582	1.05158	0.88573
Sn2	19.48036	47.74399	1.10175	1.04964	0.88428
Sn3	19.57300	47.72131	1.10751	1.05238	0.88669
Sn4	19.56360	47.70882	1.11070	1.05390	0.88939
Sn5	18.30441	47.71814	1.10832	1.05277	0.88768

The noise quality parameters, such as the signal-to-noise ratio (SNR), the peak signal-to-noise ratio (PSNR), the mean squared error (MSE), and mean absolute error (MAE), represent the quality indicators of the denoising process. A better denoising is characterized by higher values of SNR and PSNR parameters and lower values of the MAE and MSE parameters. The aforementioned charecterestics were satisfied in Gaussian noise scenario.

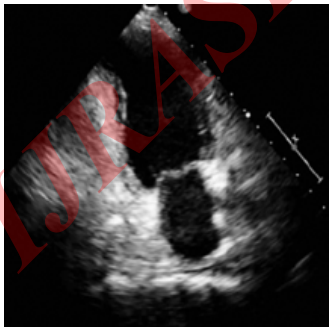


Fig 21 Rayleigh noisy image

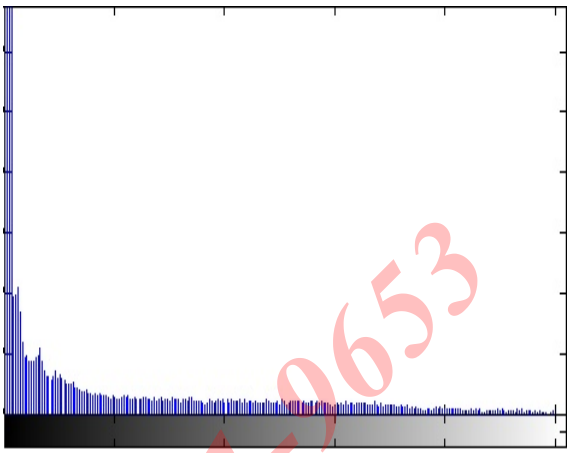


Fig 22 Histogram of Rayleigh noisy image

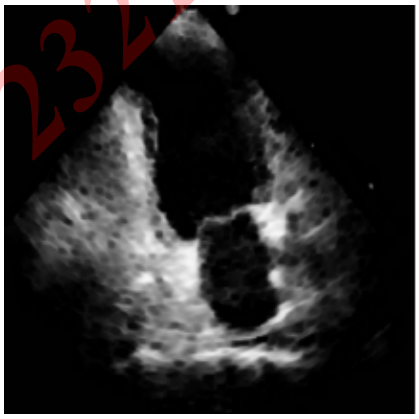
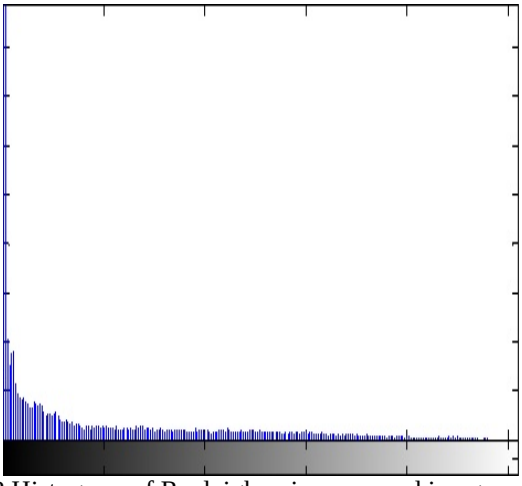


Figure 5.21: Rayleigh noise removed image



Figu 22 Histogram of Rayleigh noise removed image

TABLE IV

ISSN: 2321-9653

NOISE QUALITY PARAMETER CALCULATION IN RAYLEIGH NOISE SCENARIO

Sl. No	Multiplicative Rayleigh Noise removed Image				
	SNR	PSNR	MSE	RMSE	MAE
Dab 1	-8.15662	19.64592	711.02965	26.66514	19.85146
Dab 2	-8.14711	19.60184	718.28317	26.80081	19.97318
Dab 3	-8.23648	19.58783	720.60406	26.84407	19.98969
Dab 4	-8.30835	19.62153	715.03520	26.74014	19.91273
Dab 5	-9.00269	19.64722	710.81668	26.66115	19.51733
Dn1	-8.68954	19.62576	714.33878	26.72712	19.72232
Dn2	-8.79145	19.62095	715.13000	26.74191	19.76901
Dn3	-8.84252	19.60200	718.25787	26.80033	19.81375
Dn4	-8.93384	19.62161	715.02159	26.73989	19.69872
Dn5	-8.96010	19.59773	718.96436	26.81351	19.76202
Sab1	-8.01544	19.54200	728.24931	26.98609	20.19136
Sab2	-7.66254	19.62434	714.57143	26.73147	20.04240
Sab3	-7.83660	19.64303	711.50386	26.67403	19.89619
Sab4	-7.78738	19.55634	725.84784	26.94156	20.21225
Sab5	-8.66009	19.67580	706.15443	26.57357	19.50963
Sn1	-8.65578	19.62643	714.22740	26.72503	19.80144
Sn2	-8.64634	19.61728	715.73373	26.75320	19.76312
Sn3	-8.59344	19.55487	726.09407	26.94613	20.01781
Sn4	-8.50098	19.64425	711.30420	26.67029	19.75854
Sn5	-9.73216	19.68157	705.21733	26.55593	19.28587

L. DISPLAYED RESULT

The final matlab figure will be displayed as show in the figure 23. For diastolic image it will displayed "DIASTOLE" and so for systolic image "SYSTOLE".

Fig 23 Final figure displayed

IV. CONCLUSION

In this project the discerning of atrial diastole and systole was done successfully. From the region segmented image the major axis height and minor axis width were calculated. This helped to determine the phases. Salt and pepper noise was considerably reduced using median filter. It is noted that the preprocessing steps and edge detection steps are well applicable for apical two chamber long-axis, 0◦ view (LAX0) of 2-D echocardiographic images. The region segmentation highlights the feasible myocardium wall region in the grayscale image. The noise quality parameters such as

signal-to-noise ratio (SNR), peak signal-to-noise ratio (PSNR), mean squared error (MSE), mean absolute error (MAE) were calculated for Gaussian noise scenario and Multiplicative Rayleigh noise scenario. A better denoising is characterized by higher values of SNR and PSNR parameters and lower values of the MAE and MSE parameters. The aforementioned characteristics were obtained in Gaussian noise scenario. The backpropagation network works well for the classification as it update the weights in each step to attain the best result.

V. ACKNOWLEDGEMENT

I thank almighty, my parents, teachers and friends for their constant encouragement without which this work would not be possible.

VI. REFERENCES

[1]. Dorin Bibicu., Luminita Moraru., (2013) "Cardiac cycle phase estimation in 2-D echocardiographic images using an artificial neural network", *IEEE transactions on biomedical engineering, pp. 1273-1279.*

[2]. Dorin Bibicu, Luminita Moraru, Anjan Biswas. (2012) **"**Thyroid nodule recognition based on feature selection and pixel classification methods", Society for Imaging Informatics in Medicine, DOI 10.1007/s10278-012-9475-5.

[3]. Simona Moldovanu., Luminita Moraru., Dorin Bibicu.,(2012) "Computerized decision support in liver steatosis investigation", *International Journal of Biology and Biomedical Engineering, pp. 69–76.*

[4]. Matilda Landgren., Niels Christian Overgaard., Anders Heyden. (2011) "Segmentation Of The Left Heart Ventricle In Ultrasound Images Using A Region Based Snake," Centre *of Mathematical Sciences, Lund University, Sweden,504-510.*

[5] S. A. Aase., S. R. Snare., O. C. Mjlstad., H. Dalen., F. Orderud and H. Torp. (2009) *"QRS detection and cardiac cycle separation without ECG,"IEEE International Ultrasonics Symposium, 1399–1402.*

[6]. S Suryanarayana., Dr. B L Deekshatulu., Dr. K Lal Kishore., Y RakeshKumar., (2009) "Novel impulse detection technique for image denoising", Applications of Information Technology,pp. 102–106.

[7]. Eileen M. McMahon., Josef Korinek., ShiroYoshifuku. (2008) " Classification Of Acute Myocardial Ischemia By Artificial Neural Network Using Echocardiographic Strainwaveforms," *Computers in Biology and Medicine 38,416 – 424.*

[8] Wen Fang., Kap Luk Chan., Sheng Fu. (2008) "Incorporating Temporal Information Into Active Contour Method For Detecting Heart Wall Boundary From Echocardiographic Image Sequence," *Computerized Medical Imaging and Graphics 32,590–600.*

[9]. Ritwik Kumar1., Fei Wang., David Beymer. (2008) " Cardiac Disease Detection From Echocardiogram Using Edge Filtered Scale-Invariant Motion Features," *IBM Almaden Research Center, San Jose, CA, USA,205-209.*

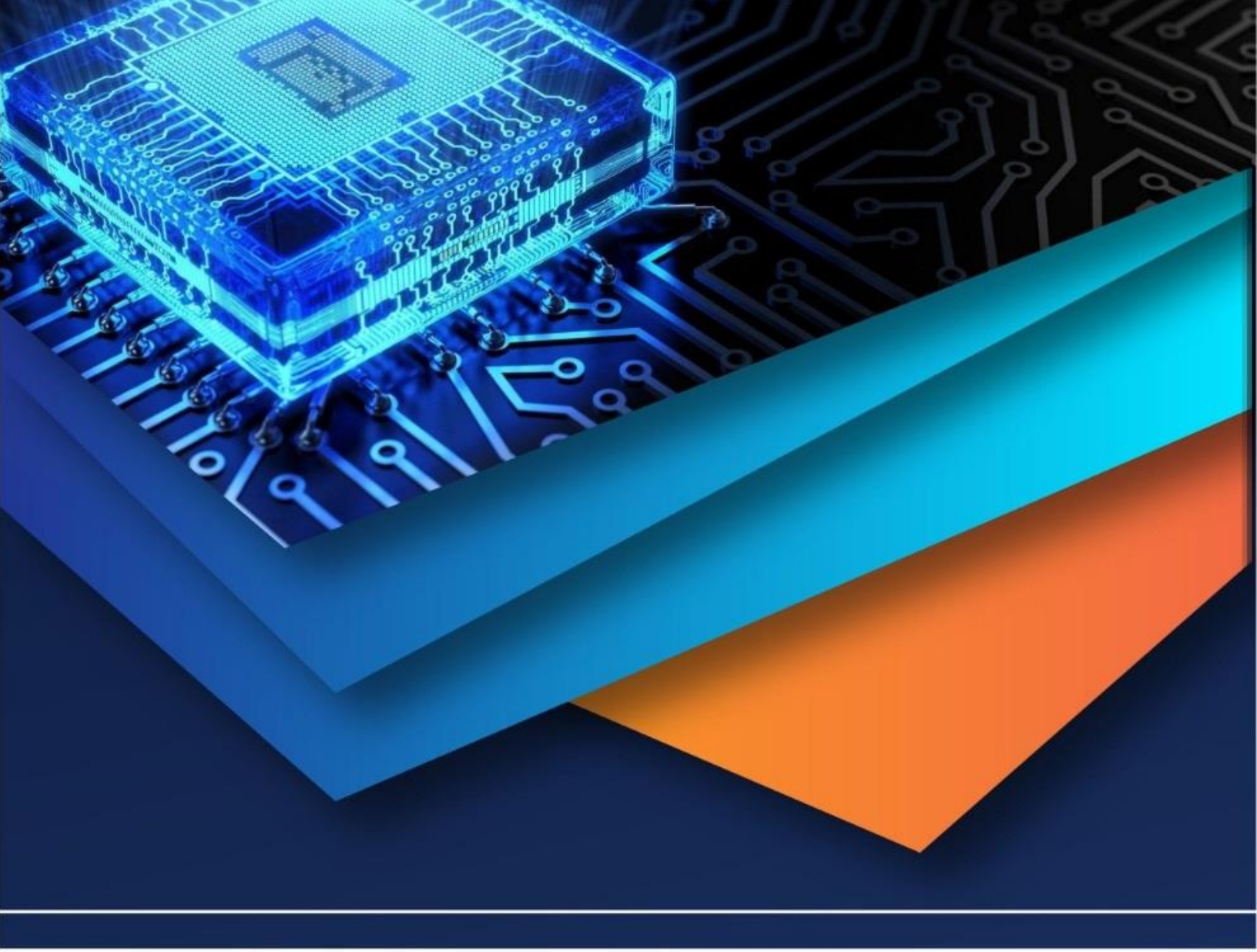
[10]. N. Kachenoura., A. Delouche., A. Herment., F. Frouin., and B. Diebold. (2007) "Automatic detection of end systole within a sequence of left ventricular echocardiographic images using autocorrelation and mitral valve motion detection", *Annual International Conference of the IEEE Engineering in Medicine and Biology Society, 4504–4507.*

[11]. S. Martin., V. Daanen., J. Troccaz., and O. Chavanon.(2006) "Tracking of the mitral valve leaflet in echocardiography images", *IEEE International Symposium Biomedical Imaging,181–184.*

[12]. P. Negrini., P. Tomassini., G. Cella., M. Magrin., M. Ercolani., A. Mazzarisi., V. Gemignani., A. Ciampa., P. Marcheschi., and P. Marraccini. (2002) "Automatic border detection through a cardiac cycle to analyze left ventricular function" *Computers in Cardiology, 181–183.*

[13]. J.B. Santos., D. Celorico., J. Varandas.(2000)"The Importance Of The Pre-Processing On The Echocardiographic Images For The Left Ventricular Contour Extraction," *Department of Electrical Engineering and Computers, Institute of Science and Materials Engineering, University of Coimbra, Portugal,303-309.*

[14]. G. Ramachandran., M. Singh., and K. Nityanandan.(1991) "Left ventricle wall motion analysis during various phases of cardiac cycle from 2D-echocardiographic images", *Annual International Conference of the IEEE Engineering in Medicine and Biology Society 13, 235–236.*



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY