A Novel Approach For Detection Of Fetal Head Parameters Identification And Classification With Ultra Sound Images Using Multi-Scale Convolutional Neural Network

S. Fathimuthu Joharah¹, Dr.S. Kother Mohideen²

¹Reg.No:18221252162004, Research Scholar, Department of IT, Sri Ram Nallamani Yadava College of Arts & Science, Tenkasi, Tamilnadu-627804, India, Affiliation of Manonmaniam Sundaranar University, Abishekapatti, Tirunelveli 627012, Tamilnadu, India.

²Assosciate Professor & Head, Department of IT, Sri Ram Nallamani Yadava College of Arts & Science, Tenkasi, Tamilnadu-627804, India.

Abstract
Ensuring accurate diagnosis and prognosis of fetal conditions is a critical objective, particularly when it comes to evaluating fetal head formation for obtaining crucial information. One of the prevailing challenges lies in mitigating the impact of low signal-to-noise ratio, especially in the context of the intricate measurements of small fetal head ultrasound images. This study focuses on the development of a fully automated system for detecting subsequent fetal head composition based on ultrasound images. To address speckle noise, the preprocessing phase employs the Recursive Least Square Mean Adaptive Filter. The detection of fetal head structure is achieved through the utilization of the Hadamard Transform-based Hough Transform method, yielding a segmentation accuracy of 97%. Ultimately, a feed-forward Artificial Neural Network (ANN) classifier is employed to assess the accuracy of the machine learning Link-Net with multi-scale images. The experimental outcomes, drawn from five ultrasound sequences, substantiate the effectiveness and precision of the proposed approach for genuine fetal head diagnostics.

Keywords: Fetal Head, ANN Feed Forward (NFFE) Classifier, Link-Net with Multi-Scale Image, ROI based Hadamard Transform (HT) and, Recursive Least Square Mean Adaptive Filter.

INTRODUCTION
The proposed algorithm demonstrates a commendable organization with regards to sensitivity, specificity, and overall accuracy [10]. In the realm of academic literature, numerous scholars have put forth a range of techniques for the automated identification of pulmonary nodules [6]. However, all existing methodologies must go through four distinct stages in order to effectively detect pulmonary nodules [1]. These stages encompass preprocessing, segmentation, feature extraction, and classification. Notably, Computed Tomography (CT) stands out as a more
efficient diagnostic tool for pulmonary nodules in comparison to conventional X-ray methods. In specific cells, the DNA does not undergo repair following damage. Instead, these cells proliferate and give rise to new atypical cells, which are referred to as cancerous cells [5]. Lung cancer is a hostile and diverse ailment, responsible for a significant portion of cancer-related fatalities [4]. The development of these atypical cells is a precursor to the onset of lung cancer, culminating in the formation of a tumor or nodule. It is important to note that not every tumor is classified as cancerous [2]. Non-malignant tumors are referred to as benign nodules. On the other hand, malignant nodules are cancerous growths that proliferate without a specific pattern, resist control, and damage the surrounding healthy lung tissues [9] & [7].

**SYSTEM MODEL**

Figure 1: Flow Diagram of the Proposed System

Our proposed system is an end-to-end multi-Task network utilizing the Link-Net architecture (MTLN) with multi-scale inputs. As depicted in Figure 1, the system comprises two primary modules: a segmentation network and a classification Tuner [3]. Each of these modules is associated with distinct loss functions, enhancing the training process for the entire network and ultimately resulting in improved overall performance. Throughout the training phase, the network parameters undergo refinement via the back-propagation of combined loss gradients originating from various entry points:

**RLS Mean Adaptive Filter**

Noise reduction is a critical aspect of the pre-processing technique as various factors such as the scanning process, patient movement, instrument vibrations, camera settings, and environmental interferences can introduce distortions in medical imaging inputs like ultrasound and CT/MRI scans. To address this, a Recursive Least Squares Adaptive Filter is employed for noise removal. The filter takes into account the initial conditions obtained from prior data and updates the old data estimates with new information. The length of the data varies depending on the data variable. The primary objective of this filter is to minimize the mean square error, thereby enhancing the quality of the medical images.

The adaptive filter offers notable benefits, including efficient computational speed, optimized
system performance, and the minimization of cost functions. The Recursive Least Squares (RLS) algorithm plays a pivotal role in tackling intricate challenges through the adaptive filter. This technique is characterized by

\[ x(k) = \sum_{n=0}^{q} b_k(n) d(k - n) + v(k) \quad (1) \]

Let \( v(k) \) represent the additive noise, and \( b_k(n) \) denote the binary representation. The desired signal \( d(k) \) is expressed as follows:

\[ d(k) \approx \sum_{n=0}^{p} w(n)x(k - n) = W^T X_k \quad (2) \]

Where \( X_k \) be the column vector

\[ d(k) \approx \sum_{n=0}^{p} w_k(n)x(k - n) = W_k^T X_k \quad (3) \]

For good estimate \( \hat{d}(k) - d(k) \) are small in magnitude and least square sense. It also reduces the cost function \( C \) by fixing the \( W_k \) coefficient of filter. The error signal \( e(k) \) can be expressed as,

\[ e(k) = d(k) - \hat{d}(k) \quad (4) \]

The cost function of weighted least square error can be estimated as,

\[ (w_k) = \sum_{i=0}^{k} \lambda^k e^2 \quad (5) \]

Where for getting factor be \( 0<\lambda\leq1 \)

The cost function is decreased by applying the derivative of \( n \) entries and the coefficient vector \( w_k \) and the result of settling time is zero.

\[ \frac{\partial w_k}{\partial w_{k(n)}} - \sum_{i=0}^{k} 2\lambda^{k-i}(i) \frac{\delta e(i)}{\delta w_{k(n)}} = - \sum_{i=0}^{k} 2\lambda^{k-i}(l)x(l - n) = 0 \quad (6) \]

Where \( n=0,1,\ldots,p \)

Apply \( e(k) \) error signal to Equation (4)

\[ \sum_{i=0}^{k} \lambda^{k-i}[(i) - \sum_{l=0}^{p} w(l)x(i - l)]x(i - n) = 0 \quad (7) \]

Where \( n=0,1,\ldots,p \)

After rearranging the equation is,

\[ \sum_{l=0}^{p} w(l)\left| \sum_{i=0}^{K} \lambda^{K-i}x(i-l)x(i-n) \right| = \sum_{i=0}^{K} \lambda^{K-i}(l)x(l - K) \quad (8) \]

The above equation can be rewritten as matrix form,
\[ R_x(K)w_k = r_{dx}(K) \]  \hspace{1cm} (9)

Where \( R(k) \) be the weighted sample covariance matrix

\( r_d(k) \) be the cross-covariance between \( d(k) \) and \( x(k) \)

\[ w_k = R^{-1}(K)r \]  \hspace{1cm} (10)

Equation (10) is to decrease the cost function and this is the main result.

**Algorithm for RLS Mean Adaptive Filter**

The algorithm for Recursive Least square mean adaptive filter for \( p \)th order is as follow.

Parameter are, \( p \) be the filter order, \( \lambda \) be the forgetting factor, \( \partial \)be the initialize value\( P(0) \)

**Step 1:** Initialize all the parameter

\[ (k) = 0 \]

\[ (n) = 0, \text{where} \ n = -p, ..., -1 \]

\[ (n) = 0, \text{where} \ n = -p, ..., -1 \]

\[ P(0) = \delta I, \text{where Identity matrix be I of rank } p + 1 \]

**Step 2:** After initialization of all parameter which is pre-defined, then the computation process takes place for \( k=1,2,\ldots,p \)

\[ |x(k)| \]

\[ x(n) = |x(k - 1)| \]

\[ |\ldots| \]

\[ |x(k - p)| \]  \hspace{1cm} (11)

\[ a(k) = (k) - x^T(k)w(k - 1) \]  \hspace{1cm} (12)

\[ g(k) = P(k - 1)x(k)(\lambda + x^T(k)P(k - 1)x(k)) - x^T(k)w(k - 1) \]  \hspace{1cm} (13)

\[ (k) = \lambda^{-1}P(k - 1) - g(k)x^T(K)\lambda^{-1}P(k - 1) \]  \hspace{1cm} (14)

\[ (k) = w(k - 1) + a(k)g(k) \]  \hspace{1cm} (15)

The primary benefit of this filter is its capacity to eliminate noise signals from images, leading to a reduction in the least mean square value of the image. Additionally, it enhances computational speed and overall system performance.

**A. Segmentation Network**

The suggested segmentation process can be executed using Link-Net, and the sample data can be stored in cache files. The Region of Interest (ROI) selection is conducted in two stages to identify the specific area using the ROI method. Ultimately, the segmented images are obtained as a result.
Figure 3: Flowchart of the Proposed Segmentation Method

Region of Interest based HT

ROI based Hadamard Transform (HT) has high segmentation efficiency.

\[
(u, v) = \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} p_{xy} \cdot g(x, y, u, v) \tag{16}
\]

\[
= \frac{1}{N} \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} p_{xy} (-1) \sum_{i=0}^{N-1} [bi(x)pi(u) + bi(y)pi(v)] \tag{17}
\]

\[
P_{xy} = \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} H(u, v) \cdot h(x, y, u, v) \tag{18}
\]

\[
= \frac{1}{N} \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} (u, v) (-1) \sum_{i=0}^{N-1} [bi(x)pi(u) + bi(y)pi(v)] \tag{19}
\]

\[
P_0(u) = b_{n-1}(u)
\]

\[
P_1(u) = b_{n-1}(u) + b_{n-2}(u)
\]

\[
P_2(u) = b_{n-2}(u) + b_{n-3}(u)
\]

\[
P_{n-1}(u) = b_1(u) + b_0(u) \tag{20}
\]

In this suggested framework, the Hadamard transform is employed to minimize redundancy in the segmented image and eliminate blocking artifacts. The Hadamard transform is characterized by its symmetry, separable unitary transformation, and a kernel matrix containing only two elements: +1 and -1. The Hadamard transform is applicable for \((N = 2^n)\), where \(n\) is the variable representing a specific value.

Jeyathilake et al. (2013) presented the Hadamard Transform (HT).

The kernel matrix for these two cases is given below:

\[
\frac{1}{\sqrt{2}} H_2 = \frac{1}{\sqrt{2}} \begin{bmatrix}
1 & 1 \\
1 & -1
\end{bmatrix} \tag{21}
\]

For larger \(N\), it can be created from block matrix form is:

\[
\frac{1}{\sqrt{N}} H_2 = \frac{1}{\sqrt{N}} \begin{bmatrix}
H_{N/2} & H_{N/2} \\
-H_{N/2} & -H_{N/2}
\end{bmatrix} \tag{22}
\]

The matrix contains only element that are 1 for size \(N = 2^n\), it makes the transform very less expensive.
If N=8, the Hadamard kernel matrix is given as:

$$H_8 = \frac{1}{2\sqrt{2}} \begin{bmatrix}
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \\
1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 \\
1 & -1 & -1 & 1 & 1 & -1 & -1 & -1 \\
1 & 1 & 1 & 1 & -1 & -1 & 1 & 1 \\
1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \\
1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 \\
1 & -1 & -1 & 1 & 1 & -1 & 1 & 1
\end{bmatrix}$$ (23)

The matrix above illustrates the count of sign changes in each row, which is indicated in the corresponding column to the right. This count of sign changes is denoted as the row’s sequence.

**CLASSIFICATION NETWORK**

**Artificial Neural Network Feed Forward (NFFE) Classifier**

The Feed Forward Artificial Neural Network (FFANN) comprises two layers: a hidden layer with r nodes, and an output layer with a single node. For each node in the hidden layer, the input vector Z is augmented with its respective weight. The function $f_h(.)$ is a scalar function specific to the hidden layer. It operates on the sum of each node in the hidden layer, and the layer’s response is determined by Equation 22.

$$y_i = f_h(\sum_{l=1}^{n} w^h z), j = 1,2,...,m \quad (24)$$

Where, $w^h$ is the weight applied between input $z$ of $i^{th}$ hidden layer node.

In the same way, the outputs of the hidden layer $y_j$ are allocated to weights and their sum activates an output layer function $f_0(.)$, in order to generate the absolute ANN response $\hat{M}$ which is defined as in Equation 23.

$$\hat{M} = f_0(\sum_{j=1}^{m} w^o y_j) \quad (25)$$
Figure 4: Structure of ANN-NFFE

Figure 4 shows the ANN-NFFE structure. The weights vector can be defined as \( W = [W^T W^T]^T \), where \( W_h \) the set of weights is applied to input parameters to the hidden layer nodes and \( W_o \) the set of weights is applied to hidden layer nodes to the output layer nodes. To enhance its performance, the Artificial Neural Network (ANN) is trained using established input-output training data. During this training phase, the network’s weight values are fine-tuned to achieve the desired mapping.

**Training Mode**

A finite set of Ntr training pairs \( \{Z_k, M_k\}, k=1, 2… Ntr \) uses in the training mode. \( Z \) is the \( k^{th} \) input, which is defined as \( Z = \left[ Z_{-\frac{Ntr}{2}+k-1} Z_{-\frac{Ntr}{2}+k} \cdots Z_{-\frac{Ntr}{2}+k=n-1} \right]^T \) and \( M_k \) is the delayed version using time shift operator \( \tau \), which is given by, \( M_k = m(kT - \tau) \). The optimum weight is obtained using the equation,

\[
E_k = \dot{M}(W.Z_k) - M_k \quad (26)
\]

\[
W^{(i+1)} = W^{(i)} - (H^{(i)} + \mu I)^{-1} \nabla f^{(i)}, \ i = 0,1,2,\ldots. \quad (27)
\]

\(^{(i)} \) and \( H^{(i)} \) represented in equation 26 and 27

\[
\frac{\partial j}{\partial W} = 2 \sum_{k=1}^{N} \stackrel{\wedge}{M}(W.Z)E \quad (27)
\]

\( H = J^T J \quad (28) \)

here the Jacobian matrix \( J \) is given in equation 28.
\[ J = \left[ M_1, M_2, ..., M_N \right]^T \] (29)

\[ Z_k = \left[ Z_{-k+k}, Z_{-k+k+1} ..., Z_{k+k} \right]^T \]

Where, \( z_k \) is the transmitted signal and \( K \) is the predictable number of interfering symbol. For each input vector \( Z_k \), the ANN-NFFE generate signal which compensates the dispersion present in the system.

Based on the error signal, the system performance is measured. The error calculated using the Equation 31.

\[ E_k = \hat{M}(W, Z_k) - I_q \] (30)

\[ DSC = -\frac{2 \times (\text{Area}_S \cap \text{Area}_R)}{||\text{Area}_S|| \times ||\text{Area}_R||} \] (31)

\[ DF = H_{CP} - H_{CGT} \] (32)

\[ ADF = |H_{CP} - H_{CGT}| \] (34)

were \( \text{area}_S \) is the ground truth area and \( \text{Area}_R \)

\[ (S, R) = \max (h_s(S, R), h(R, S)) \] (35)

RESULT AND DISCUSSION

The experimental setup encompasses both software and hardware components. In terms of the software component's development, MATLAB2018a software package was used. For the hardware part, the system was run on an Intel(R)Core(TM) i7-8700 @3.2 GHz and Geforce GTX 1080ti 11GB GPU.

Figure 5: Sample Images of Fetal Ultrasound Dataset

Figure 6: Comparison of Segmentation Results. From left to right:

1) Sample Image, 2) FC segmentation, 3) MTLN Segmentation Result.
Figure 7 displays the segmented outputs generated by the proposed method. The third column showcases the results obtained from the MTLN. Ground truth images are outlined in red, while the segmented outputs are represented in regions of green and blue colors.

Figure 7: Ground Truth Segmentation Results of Fetal Head
Figure 8: Steps of quantification system C. From top to Bottom: (A) Sample of the Fetal Output Image

Figure 9: Fetal result system (1) Sample fetal image. (2) Pixel classifier output. (3) Dynamic programming pixels.
CONCLUSION
The preprocessed head fetal images are subjected to the calculation of three diverse performance metrics. The identification is achieved through a multi-Task deep network based on the Link-Net architecture, incorporating CNN. The dataset's effectiveness is assessed through classification accuracy, which is reflected in the highest recorded accuracy within the confusion matrices. The proposed approach demonstrates improved sensitivity while reducing computational time. This study and its findings contribute to the ongoing research in this field. Looking ahead, future work will focus on developing innovative techniques for precise localization of these fetal head diagnostics, addressing a persistent demand in the field.

REFERENCES


