Ultrasound Images Enhancement using UNet - Deep Learning according to Resolution and Speckle Noise

Khadeja M. Mohamed[1], Dr. Mohammed H. Ali[2]

Electronic & Communications Engineering, Al-Nahrain University, Baghdad-Iraq.[1]

Electronic & Communications Engineering, Al-Nahrain University, Baghdad-Iraq.[2]

ABSTRACT

Medical Ultrasound images are mostly corrupted by a multiplicative granular noise called speckle noise which degrades the quality of the images captured by using medical imaging techniques. It causes difficulties in image interpretation and this is mainly due to back scattered signals from the multiple targets. The main focus of this study is to reduce Speckle Noise and resolution enhancement of the medical ultrasound images based on the Artificial Intelligent (AI), Deep Learning (DL) especially Convolutional Neural Networks (CNN) algorithm. The speckle noise is multiplicative and nonwhite process which corrupts a low-level luminance images like ultrasound images. Various filtering techniques for speckle noise reduction have been proposed in the past; however, their performances are still limited as a comparison between speckle-noise reduction and image features preservation. Meanwhile, CNN have been proved effective and results for various computer vision tasks, including image classification, segmentation and detection.

In this study, a model has been designed based on CNN to reduce speckle noise in ultrasound images and enhancement the resolution. This trainable model is based on U-Net architecture and has been trained using the dataset include real-Ultrasound, grayscale, noisy/clear images with size of 256×256 pixels. To learn mapping between the noisy and clear images, eighteen layers have been used connected to each other. The simulation is done using Google Colaboratory (or Colab), Python language program. SSIM (Structural Similarity Index Measure) and PSNR (Peak Signal to Noise Ratio) which used to evaluate the final reconstructed image, and with different levels of speckle noise variance 0.05, 0.1, 0.25, 0.5 and 1, the SSIM are 99.028%, 98.692%, 97.4549%, 96.835% and 93.491%, and the PSNR are 31.636dB, 30.314dB, 28.7011dB, 27.9635dB and 24.2891dB, respectively.

Keywords: Ultrasound, Speckle Noise, Image denoising, CNN, UNET, Peak signal-to-noise ratio (PSNR), Structural Similarity Index Measure (SSIM), Enhancement.

1. INTRODUCTION

Image denoising is a classic-inverse problem in computer vision that tries to recover a clean image from a noisy image. Image-denoising algorithms are widely used in various sectors, such as remote sensing image restoration, since they can effectively recover original images and restore details [1] and medical image [2]. The image denoising problem can be expressed by y = x + σ, for a noisy image y, where x is the original image (also referred to as the clean image) and σ represents Additive White Gaussian Noise (AWGN) with standard deviation σ.

Medical ultrasound imaging has grown in popularity because it is more accessible, less expensive, safe, and easy to use than other medical imaging procedures. It also produces images in real time [3]. Ultrasound signifies an imaging process that constructs cross sectional images, it uses sound waves to produce images of the inside of the body.

Ultrasound waves have frequencies that are higher than the human ear's audible range (greater than 20 kHz). Clinical Ultrasound imaging generates images of structural characteristics in biological tissue using sound pressure waves with frequencies ranging from 1 to 12 MHz. A noise that occurs during image acquisition as a result of the effect of environmental condition on the imaging sensor is known as Speckle Noise.

The speckle texture is created through diffuse reflection, with the resulting waves interacting both constructively and destructively.

Speckle Noise can be modelled by random values multiplied by pixel values,

$$I^{`} = I\,(1+N)$$

$$I^{`} = I + IN \qquad (1.1)$$

where $I^{`}$ is a noisy image, I an image matrix, N is speckle noise, N: consists of normally distributed values with mean 0 and its default value is 0.04. [4]

For ultrasound imaging, speckle noise is multiplicative. In most ultrasound images, we have a combination of additive and multiplicative noises [5]. Speckle noise images reduce the image contrast and make it hard to perform image processing operations such as border detection, segmentation et, as shown in Figure 1.

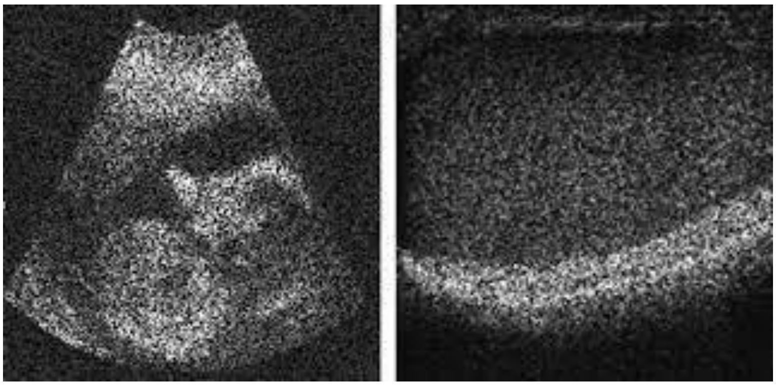


Figure 1: Noisy Ultrasound images corrupted by random noise(a) Fetal (b) Testicle.

2. Enhancement Principal

Image enhancement's major goal is to improve an image's visual look, or to provide a "better transform representation of the image." Various sorts of photos, such as medical images, satellite images, aerial images, and real-life photographs, are plagued by issues such as poor contrast and noise. To improve image quality, it's critical to boost contrast and reduce noise [6].

In recent years, DL which is a branch of Machine Learning is new approach in enhancement field and achieved good compared with techniques mentioned above. It is thought to be a representation learning strategy that can process and learn mid-level and high-level abstract characteristics collected from raw data directly and automatically. [7] (e.g., Ultrasound images).

3. LITERATURE REVIEW

In this section, we illustrate Ultrasound enhancement techniques review according to Speckle noise reducing and resolution enhancing across about 15 years ago.

In [8] an adaptive image enhancement method through selected dynamic filtering based on speckle detection for ultrasound images to enhance the tissue structure and smooth the speckle regions. The goal is to differentiate speckle and structure areas and apply different filtering to each for improved quality. As a result, image quality has improved, with the muscle boundary and tissue structure strengthened while the speckle areas have been smoothed without blurring.

Also, SMU (Srad Median Unsharp) algorithm for noise suppression in ultrasound breast images in order to realize a computer aided diagnosis (CAD) for breast cancer. it reduces significantly the speckle while preserving the resolution and the structure of the original ultrasound images and this is suitable to get a precise extraction of the ROI, by adding speckle noise with square standard deviation 0.5, the result was PSNR = 19dB, MSE= 0.02 [9].

Some other technique for denoising was to use a combined logarithmic transformation and a non-linear diffusion tensor [10]. The logarithmic transformation is a good choice for converting signal dependent or pure multiplicative noise to additive noise, by applying anisotropic diffusion along the coherent structure direction of interesting features in the image and the result was PSNR= 16.10 dB, MSE = 0.023 and SSIM = 0.82.

At 2015 Deep learning, CNN algorithm started to use in denoising problems, [11] proposed a single image super-resolution (SR) method used CNN. This method learns low/high-resolution images end-to-end mapping in order to train the network to perform super-resolution tasks, consists of large training set 395,909 colored images and different numbers of layers, the best result is PSNR = 30.29 and SSIM = 0.8977. As clear from these studies PSNR and SSIM are better in CNN algorithm than other classical techniques.

A residual learning framework, which is one of deep learning networks, aids in the preservation of good results in a network with several layers. One issue that professionals frequently raise is that with deep networks with dozens of layers, accuracy can become saturated, and some degradation can occur [12].

A Residual Learning was presented in [13] the use of discriminative model learning through CNN for SAR (Synthetic Aperture Radar) image de-speckling. A residual learning technique recovers the speckle component of the image rather than the filtered image, which is subsequently subtracted from the noisy image to obtain a clear image. There are 17 entire convolutional layers in a network, with no pooling getting PSNR = 25.95 dB, SSIM = 0.764.

Another enhancement method that combines CNN algorithms with classical filters. Filters that preserve the ROI (Region of Interest) edges like Anisotropic diffusion, Canny and others. These filters be the first step in enhancement operation, that make a data set of CNN model. These steps applied in [14]; at the training step, true edges were detected from noisy-free images by medical ultrasound images. The edge map created from the well-known Canny technique is used to train the network. The proposed method was utilized to denoise medical ultrasound images as well as conventional grayscale images, the results have been gotten PSNR = 30.74 dB at noise level = 0.5.

Autoencoder algorithm is the most algorithms that used for denoising the images, it is encode the noise image and built the clear image at decoder part. An autoencoder is an unsupervised learning strategy for neural networks that trains the network to disregard signal "noise" in order to develop efficient data representations (encoding). Image denoising, image compression, and, in some situations, image data synthesis can all be done with autoencoders [15]. In [16] a fully convolutional network that consists in an encoder-decoder with skip connections has been proposed for image denoising. The results obtained on grayscale images show that the network can remove AWGN (Additive white Gaussian Noise) and multiplicative speckle noise, provided that it is suitably trained for the targeted noise, network consist of 20 layers, with adding speckle noise level = 1, the PSNR = 24.54 dB.

As we saw in this section, CNN algorithm techniques have good effect in enhancement field, good result in image denoising problem. At our study we used CNN algorithm, U-Net technique.

4. THEORETICAL BACKGROUND

4.1 DEEP LEARNING IN IMAGE PROCESSING

Deep learning used in image processing in two aspects which are: image filtering and image classification [17]. One of the biggest issues in using DL in image processing is how to input the data of the images neural network. A simple way is that each pixel of the image being an input to the neural network. For example, if an image with a size of 5×7 and resolution of 630×450 pixels, and assumes that the image is in RGB scale, this value gets multiplied by 3. This needs neural network with 850,500 weights per neuron (processing unit), which can raise the size of the network if many hidden layers are applied, dramatically increasing the needed calculation time for the neural network to be trained. So that, CNN are created to overcome this issue.

4.2 CONVOLUTIONAL NEURAL NETWORKS (CNN)

Convolutional Neural Networks (CNN), also known as ConvNet, is a deep CNN model that consists of a finite set of processing layers that can learn various properties of input data (e.g., a picture) at multiple levels of abstraction. They are consisting of many neurons with learnable weights and biases. The neurons receive inputs, carry out a dot product and then followed by non-linearity function.

The weight sharing feature of CNN, which reduces the number of trainable parameters in the network and helps the model avoid overfitting and increase generalization, is one of the key reasons for making CNN more significant than other traditional neural networks in the area of computer vision. [18][19]

The basic conceptual model of CNN was shown in Figure 2.

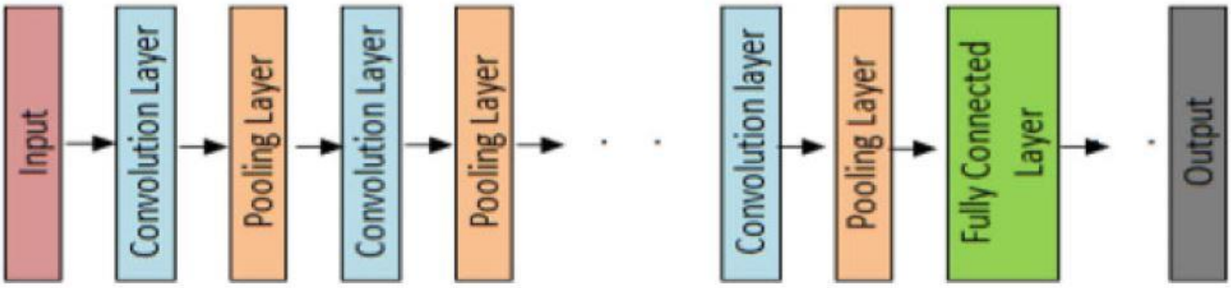


Figure 2: Conceptual model of CNN [20].

The following subsections explain many of concepts and terminologies associated with CNN that are:

4.2.1 Input of CNN

Each image in a computer system is represented by a matrix of pixel values. The gray image's width and height can be used to estimate the size of the input image. The depth of a colorful image, on the other hand, is related to the number of channels in the RGB (Red, Green, and Blue) input image; as a result, the image can be represented by three matrices [21]. Figure 3 illustrates RGB images.

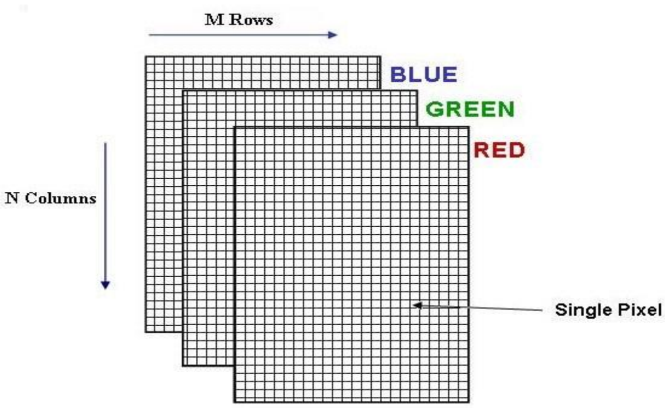


Figure 3: RGB images [22].

4.2.2 Image Features

Image features, such as edges, lines and interest points, give a lot of information about the image content. They identify local regions in the image and are fundamental in a lot of applications in image analysis such as recognition, matching, reconstruction …etc. [23].

4.2.3 Filter and Convolution Operation in Images

Kernel or filter is a small size matrix which consists of real entries. These filters correspond to the weights of the neural network; during the training process these filters are tuned.

In the image, the convolution operation extracts various features of the input image. For more details, suppose the input image is of size (width x height and depth) and is convolved with H kernels, each is of size (width x height and depth) separately. Convolution of an input image with only one kernel generates one output feature, and with H kernels independently generates H features [24]. Figure 4 illustrates convolutional operation.

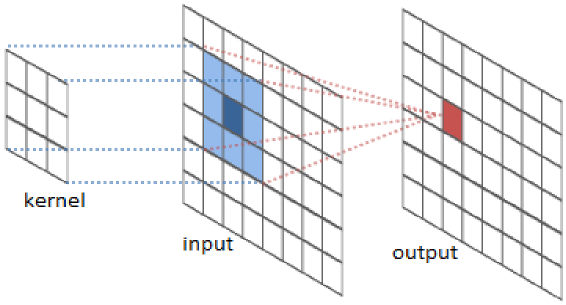


Figure 4: Convolutional operation [25].

Figure 5 illustrates which is applied filter to an image to create a feature map [26]. A value called stride determines how far the filter moves over the image. For example, if the stride is equal to 1, so the filter will move over the image only by one pixel [26].

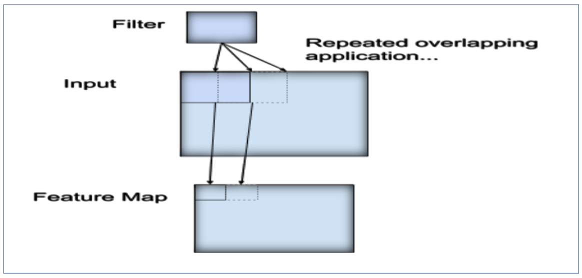


Figure 5: feature map creation [27].

4.2.4 Receptive Field

Instead of connecting each pixel in the input image to every single hidden neuron, a small region of the input image which is called (receptive field) will be connect to a hidden neuron. For example, if the size of the receptive field equals 5×5 corresponding to 25 pixel of input image it is connected to a single hidden neuron [28], as shown in Figure 6.

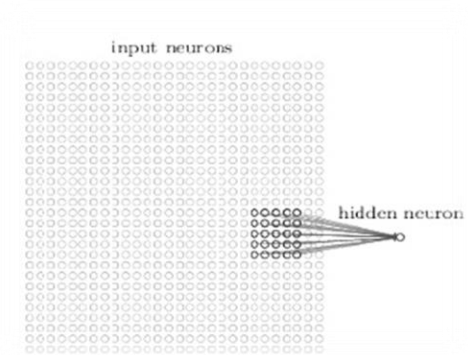


Figure 6: Receptive field [26].

4.2.5 Zero Padding

The technique of adding zeros to the input image's matrix in order to preserve the original input size is known as zero padding. Padding comes in two varieties. The first is known as valid, which indicates that there is no padding and that the convolutional layer is never pad, causing the input size to be lost. The second form, known as same, denotes that the original input image is padded before being convolved. As a result, the output matches the original input exactly. Zero-padding is seen in Figure 7 [29].

Figure 7: Zero- padding [30].

4.2.6 Epoch and Batch Size

An epoch is a unit of time that indicates how long the network will analyze data for. One epoch refers to the use of the entire dataset to train the network for a single run through the network. Every epoch, the batch size determines how many pairs of input/output the network is presented with [18].

4.2.7 Cost Function

The cost function is used to provide feedback on how poorly the network is doing. This parameter which the network is try to minimize and it resulted from the "learning" process of deep learning. To minimize the cost function of the network, an optimizer is needed; Adam Optimizer stands for (Adaptive Moment Estimation Optimizer) a popular choice for this matter [18].

4.3 CNN ARCHITECTURE

There are many layers in convolutional neural network; Figure 8 illustrates CNN architecture:

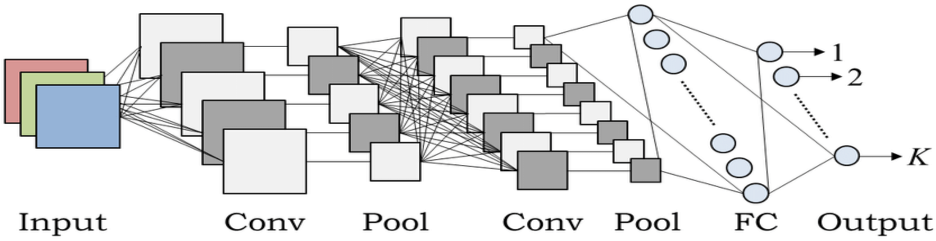


Figure 8: CNN architecture [31].

4.3.1 Convolution Layer

Convolution layer is the building block of the convolution neural network. It performs the operation of convolution over the input image. It is used to extract the features of the input image. Low-level features such as lines, corners, and edges, are extracted by the first convolution layer. Next-level layers extract higher-level of input image features. The equation of this layer is written sometimes as follow:

$$a^1 = \sigma\,(b + w\,l * a^0) \qquad (4.1)$$

Where a^1 denotes the set activations output from feature map, a^0 denotes the input activations, σ denotes for activation function, *w* denotes for weight, B denotes for Bias and * is the convolution operation [26].

4.3.2 Pooling Layer

Pooling layer reduces the resolution of the features. There are two types to do pooling: *max* and *average* pooling. In both types, the input image is separated into non-overlapping two dimensions spaces. For example, if the input feature is 4×4 and is split into 2×2 regions. By using average pooling, it will calculate the four values' average in the 2×2 region. In the situation of max pooling, the maximum number of the four values in the 2×2 region is the output [32]. Figure 9 illustrates max and average pooling.

4.3.3 Fully Connected Layer

Typically, it is utilized as the final layer of a convolutional neural network. This layer is a mathematically collection of weighting of the previous layer's features [33].

4.3.4 Activation Function Layer

When an image has been fully convolved with a filter/kernel from a previous layer, it is passed via an activation function. The activation function is utilized to give the network non-linear qualities, allowing it to tackle more complex non-linear problems than it can solve with linear problems. The most frequent activation function used in CNN is the Rectified Linear Unit (ReLU), which is defined as [18]:

$$F(x) = \max(x, 0) \qquad (4.2)$$

Because it is faster than the sigmoid and tanh activation functions, it is a common candidate for deep learning issues [18].

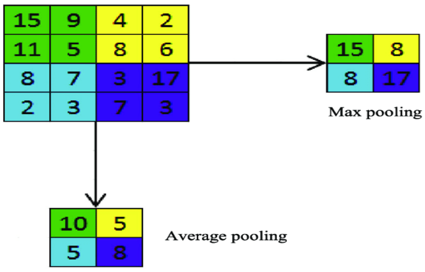


Figure 9: Max and average pooling [34].

4.4 DECONVOLUTION NETWORK

A deconvolution is a mathematical operation that reverses the effect of convolution. Imagine throwing an input through a convolutional layer, and collecting the output. Figure 10 illustrate the deconvolution operation. Now throw the output through the deconvolutional layer, and you get back the exact same input. [35]

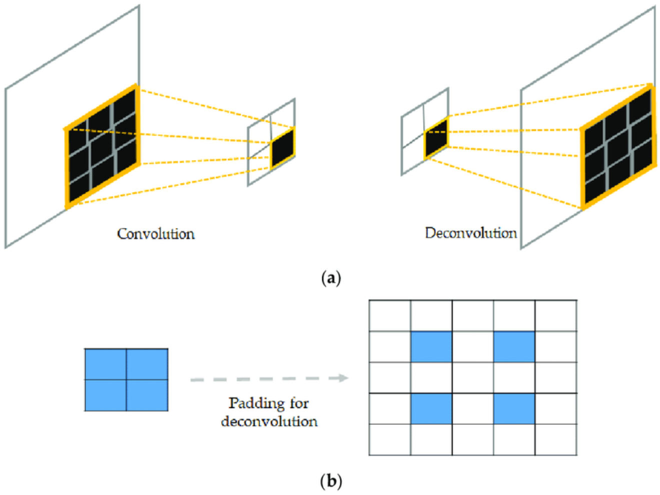


Figure 10: (a) Illustration of convolution and deconvolution; and, (b) illustration of padding for backwards convolution (deconvolution).[36]

The convolutional layer employs convolution and pooling layers to extract feature maps, whereas the deconvolutional network uses deconvolution to restore the original activation size. The convolution network followed by deconvolution network (joining the necessary layers, parameters and functions) have been made U-shaped model [19].

U-shaped architecture [37] consists of contracting and expansive path. Each step of the contracting path consists of convolutions, ReLU and max-pooling. In contrast, expansive path consists of deconvolution, and ReLU. In the expansive path, the feature map is concatenated with the cropped feature map from contracting path from corresponding layer, this operation called skip connections; skips some of the layers in the neural network and feeds the output of one layer as the input to the next layers. Figure 11 shows U-shaped, also called U-Net model CNN.

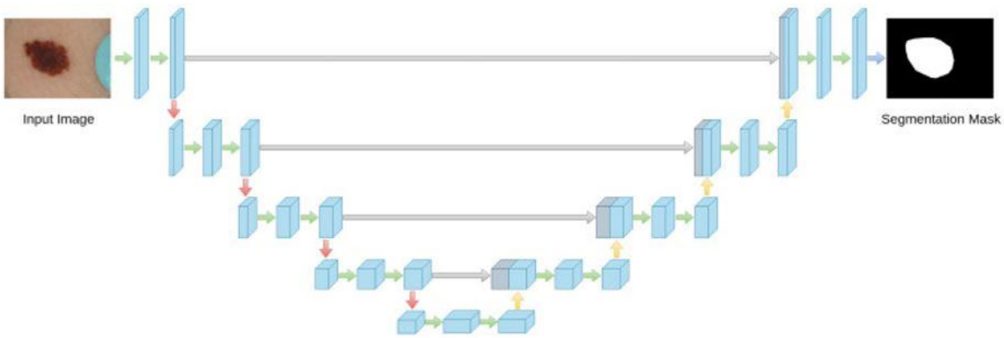


Figure 11: The classic U-Net architecture. [38]

At the end, it is worth mentioning that, beside of these the layer design, the improvement of CNN depends on several different aspects such as activation function, normalization method, loss function, regularization, optimization and processing speed, etc.

4.5 EVALUATION FUNCTION

The evaluation function is simply a parameter that allows us to determine how well the network has performed over time. Two familiar evaluation functions that are used in image filtering applications listed below:

4.5.1 Peak Signal to Noise Ratio (PSNR): is simple to determine the cumulative squared error between the actual image and desired output image in decibels. While it's a valid cheap-and-cheerful evaluation function, it works purely pixel-by-pixel and disregard the characteristics of the image [39].

$$\text{PNSR}=10log_{10}\frac{(l-1)^2}{MSE} \text{ in (dB)} \qquad (4.3)$$

Where, *l* is the image gray levels, and MSE is the mean square error. When MSE can be expressed by following equation [40]:

$$\text{MSE} =\frac{1}{mn}\sum_{i=0}^{m-1}\sum_{j=0}^{n-1}[I(i,j)-K(i,j)]^2 \qquad (4.4)$$

4.5.2 Structural Similarity Index (SSIM): is a complicated function that compares the desired images and actual images. It is focuses on images structural information, The value of SSIM is in range [0, 1]. When SSIM Value equal to 1 means the desired output and real output images are totally the same. [41].

$$\text{SSIM} = \frac{2\mu_x\mu_y+C_1}{\mu_x+\mu_y+C_1}\cdot\frac{2\sigma_{xy}+C_2}{\sigma_x^2+\sigma_y^2+C_2} \qquad (4.5)$$

Where, μ_x, μ_y are the local means, σ_x, σ_y standard deviations, σ_{xy} is cross-covariance for images, and C_1, C_2 are constants introduced to avoid instabilities.

5. METHODOLOGY

This section presents the procedure steps of proposed model in details, which is based on Machine Learning Algorithms, deep learning, convolution neural network (CNN).

The goal of this model is to enhance noisy ultrasound images to get better image quality. A U-Net learning structure is used in this model, which has two parts: a contraction path and an expansion path. Image size is lowered on the network's left side, known as the contraction path, through max-pooling and convolution layers. In the expansion path, on the network's right side, the reduced picture is transformed to the original image. The network's expansion and contraction sections form short-circuit connections with one another. There is a large number of feature channels in the contraction part, which allow the network to propagate context information to expansion layers. As a consequence, the expansive path is mostly symmetric to the contraction path, this yields a U-shaped architecture. In our study, a U-Net model has 18 layers deep learning network.

5.1 The Dataset

This model is designed to deal with speckle noise in Ultrasound images which is gray-level images. So, the dataset which used is available to download at [42], [43].

The main characteristics of this dataset are as follows:

- Ultrasound Images dataset consists of 1720 Clear Ultrasound Image with size of (256 × 256) pixels, portable network graphics format (PNG) and Joint Photographic Experts Group (JPG).
- It is consisting of two classes which have taken from real scenes for breast diagnosis and Fetal abdominal images

5.2 The Proposed Model

The workflow process of the proposed model is made up of five main phases are shown in Figure 12.

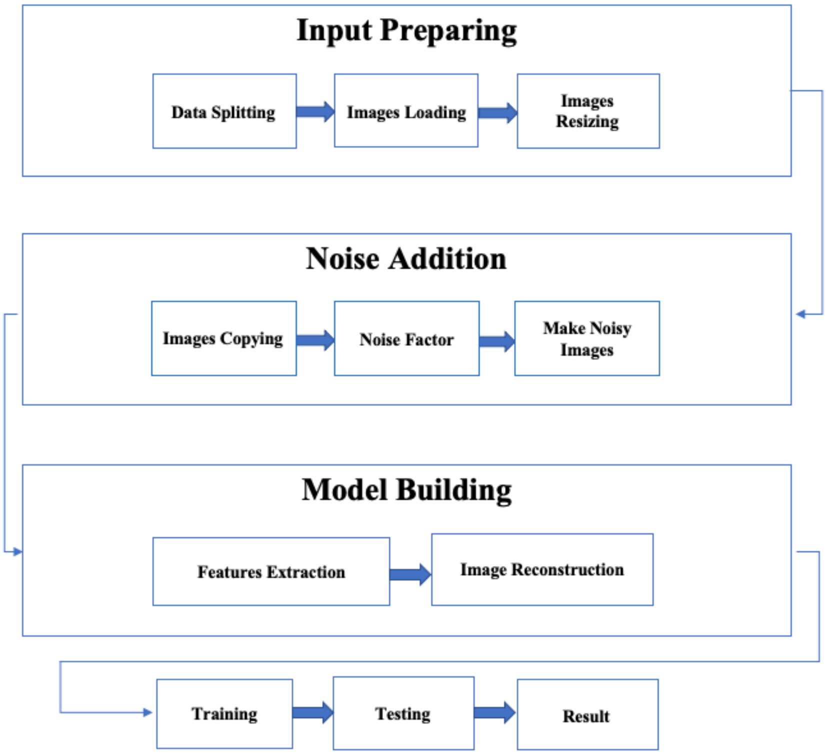


Figure 12: The proposed model phases

5.2.1 Input Preparing

The model starts with preparing the input images that will be processed by the model. This phase consists of three sub- phases as follows:

5.2.1.1 Dataset Splitting

The dataset is partitioned into two sets: (training and testing) sets. The training dataset is consisted of 1376 images of the US dataset for training the network that means it is with percentage of 80% of the total dataset.

The testing dataset consisted of 344 images of the US dataset for testing the trained network that means it is with percentage of 20% of the total dataset.

5.2.1.2 Loading of Images

The Data set has been uploaded to the google drive about 73.7 MB. Google Colab called a dataset by specifying the path location in drive.

5.2.1.3 Images Resizing

At this sub phase the two Dimensions of images have been marked as 256 × 256 as Hight_Img, Width_Img.

5.2.2 Noise Addition

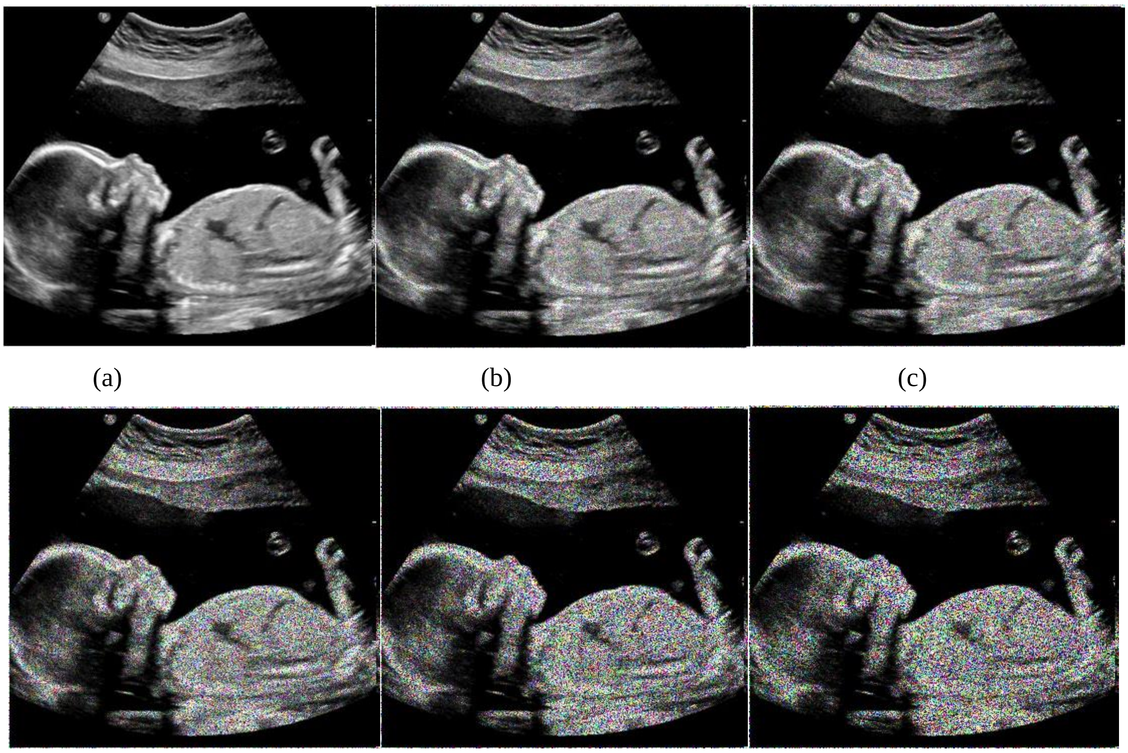
The second model phase concerns with all about noise. This phase consists of three sub- phases as follows:

5.2.2.1 Images copying

At this sub phase, the train and test images have been copied to create x_train, y_train, x_test and y_test is 1376, 1376, 344 and 344, respectively. Therefore, the new dataset consists of 3440 images.

5.2.2.2 Noise Factor

The noise that added to ultrasound images is speckle noise, the range that mostly used is 0.05, 0.1, 0.25, 0.5 and 1. These ranges shown in Figure 13.



(a) (b) (c)

(d) (e) (f)

Figure 13: Different Noise Factors (a) original image (b) 0.05 (c) 0.1. (d) 0.25 (e) 0.5 (f) 1.

5.2.2.3 Making Noisy Images

Speckle Noise is a multiplicative noise. So, x_train and x_test images have been multiplied by the noise factor to get Noisy Images, as shown in Figure 14.

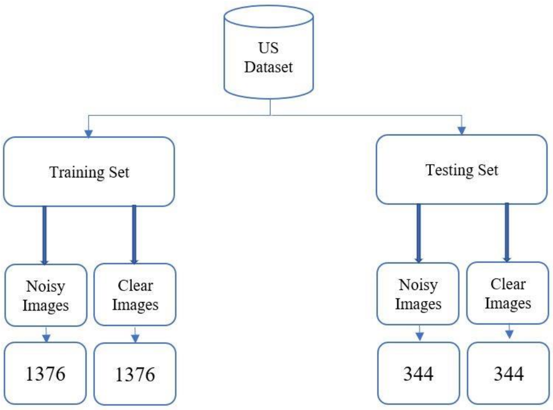


Figure 14: Noisy/ Clear Dataset

5.2.3 Model Building

The model consists of several different layers connected to each other. The number of layers is Eighteen layer, each layer of them has its own function in enhancement processing which will be described in the next sections. More depth size of the network means more perceiving of edges and texture of the input image which results in better output image.

U-Net model was introduced a skip connected between contracting layers and expanding layers which allows some data to flow and help in better image generation.

A Network consists four parts:

1- Input layer: specify the input shape of network, has no parameters, parameters = 0.

2- Contracting path: consist of repeated application of three convolution layers with 3x3 filter, same padding, each convolution layer followed by ReLU as activation function layer, and a 2x2 Max-Pooling operation layer for down-sampling. At each down-sampling step, number of filters have been halved.

3- Expansive path: consist of repeated application of three deconvolutions layers (Transposed convolution layers) with 3x3 filter, same padding, two stride, each deconvolution layer followed by ReLU as activation function layer. At each up-sampling step, number of filters have been doubled.

4- At the final convolution layer of 1x1 filter is used to map each 128-component feature vector to the desired number of channels with Sigmoid layer as activation function. In total the network has 18 layers.

Figure 15 illustrates the idea of proposed model U-Net structure.

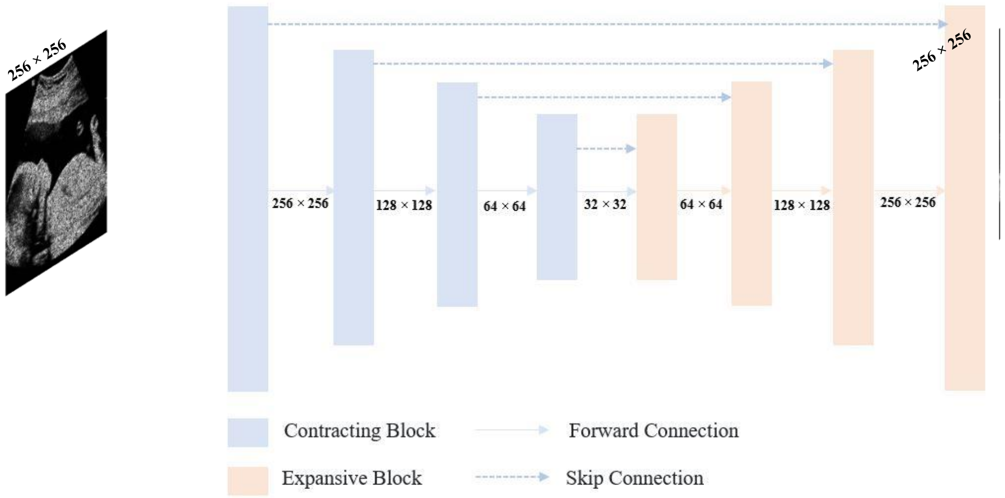


Figure 15: Proposed model U-Net structure.

5.2.4 Training Phase

Adam optimizer was used in the training as one of the best optimization algorithms. It is widely used in such image enhancement problems. The epoch number of trainings, that is, how many times the network training will repeat, is set to 2000. After saving the tainted network, the network is being used as a model and it is ready to be tested.

5.2.5 Testing Phase

The testing image employs the trained model architecture and traverses all of its layers, utilizing saved parameters such as the network's weights and filter size and number.

Peak Signal to Noise Ratio (PSNR) and Structural Similarity Index (SSIM) are the most used measurements that is used in order to evaluate similar problems.

6. SIMULATION RESULTS

Using Apple Laptop with Intel® core_i5, CPU 1.8 GHz, RAM 8 GB, Mac OS Big Sur system, Python, Google Colaboratory (Google Colab) and Google Drive. The proposed architecture U-Net Model plot shown in Figure 16.

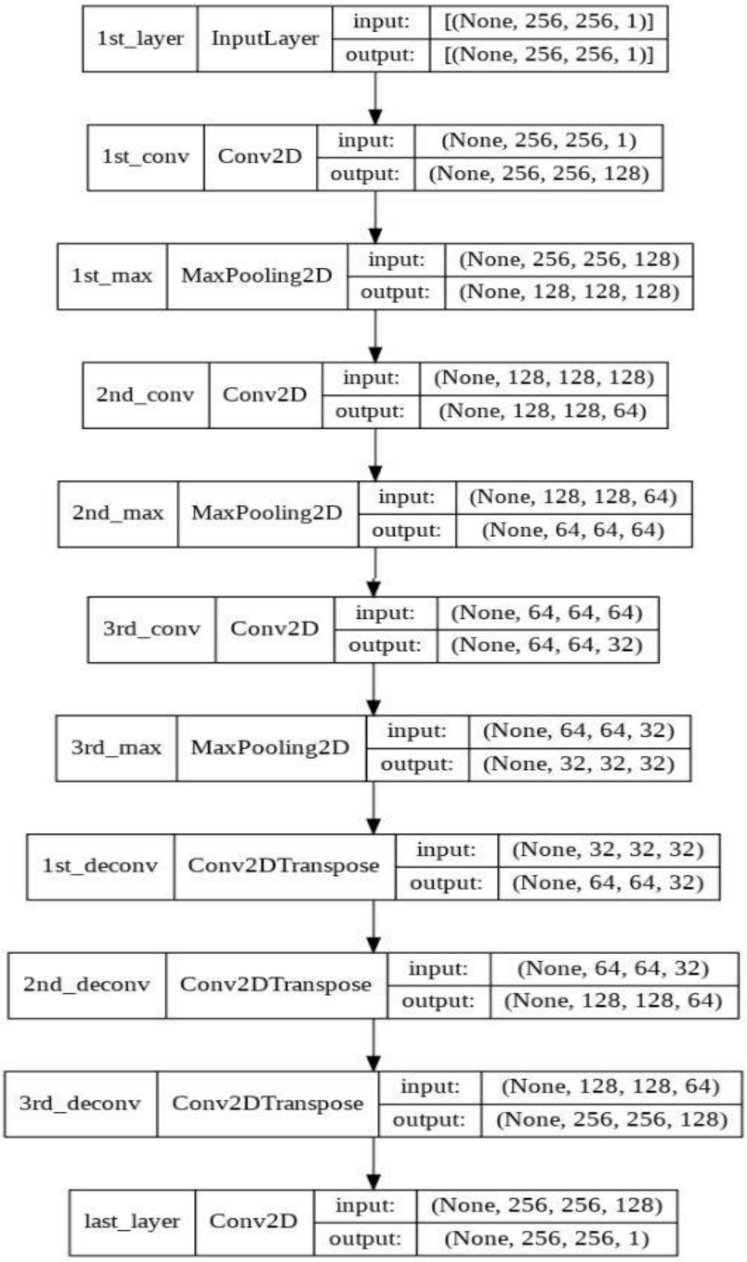


Figure 16: architecture U-Net Model plot

After 2000 epochs the training loss and validation loss show how the learning curve become down for each epoch. The training loss indicates how well the model is fitting the training data, while the validation loss indicates how well the model fits new data. Table 1 illustrate the training phase results performance that including losses, spending time and No. of epochs, at each speckle noise level.

Table 1: training phase results

A First Case with Variance = 0.05			
Epoch No.	Spending time hh:mm:ss	Training_loss	Validation_loss
10	00:00:10	0.0815	0.0813
50	00:00:50	0.0136	0.0124
100	00:01:40	0.0028	0.0030
200	00:03:20	0.0018	0.0020
300	00:05:00	0.0016	0.0017

400	00:09:40	0.0014	0.0015
500	00:08:20	0.0013	0.0014
1000	00:16:40	0.0011	0.0012
1500	00:25:00	9.9139e-04	0.0011
2000	00:37:20	9.4609e-04	0.0011
		Figure 17	Figure 18
A Second Case with Variance = 0.1			
Epoch No.	Spending time hh:mm:ss	Training_loss	Validation_loss
10	00:00:15	0.1059	0.1124
50	00:01:15	0.0127	0.0126
100	00:02:30	0.0048	0.0049
200	00:05:00	0.0017	0.0017
300	00:07:30	0.0014	0.0014
400	00:10:00	0.0013	0.0013
500	00:12:30	0.0012	0.0012
1000	00:25:00	0.0013	0.0012
1500	00:37:30	9.8383e-04	0.0010
2000	00:55:00	0.0011	0.0012
		Figure 19	Figure 20
A Third Case with Variance = 0.25			
Epoch No.	Spending time hh:mm:ss	Training_loss	Validation_loss
10	00:01:10	0.0951	0.1061

Table 1: training phase results (cont.)

50	00:05:50	0.0154	0.0134
100	00:11:40	0.0060	0.0053
200	00:23:20	0.0026	0.0025
300	00:35:00	0.0023	0.0022
400	00:46:40	0.0023	0.0021
500	00:58:20	0.0019	0.0021
1000	01:56:40	0.0017	0.0021
1500	03:05:00	0.0016	0.0021
2000	03:53:20	0.0016	0.0021
		Figure 21	Figure 22
A Four Case with Variance = 0.5			
Epoch No.	Spending time hh:mm:ss	Training_loss	Validation_loss
10	00:01:50	0.0712	0.0992
50	00:09:10	0.0706	0.0968
100	00:18:20	0.0142	0.0180
200	00:36:40	0.0029	0.0040

300	00:55:00	0.0024	0.0035
400	01:13:20	0.0022	0.0032
500	01:31:40	0.0020	0.0031
1000	03:03:20	0.0017	0.0029
1500	04:00:35	0.0014	0.0029
2000	06:06:40	0.0013	0.0027
		Figure 23	Figure 24
A Fifth Case with Variance = 1.0			
Epoch No.	Spending time hh:mm:ss	Training_loss	Validation_loss
10	00:02:10	0.1031	0.0817
50	00:10:50	0.0150	0.0179
100	00:21:40	0.0073	0.0086
200	00:43:20	0.0032	0.0038
300	01:05:00	0.0027	0.0035

Table 1: training phase results (cont.)

400	01:26:40	0.0026	0.0034
500	01:48:20	0.0025	0.0033
1000	03:36:40	0.0022	0.0031
1500	05:25:00	0.0020	0.0031
2000	07:13:20	0.0020	0.0021
		Figure 25	Figure 26

The training phase curves during 2000 epoch: in Figures 17, 19, 21, 23 and 25 illustrate the training loss of the training progress. The x-axis of the figure is representing the number of epochs, The y-axis representing the training loss at each epoch, While Figures 18, 20, 22, 24 and 26 illustrate the validation loss of the training progress. The x-axis of the figure is representing the number of epochs, The y-axis representing the validation loss at each epoch.

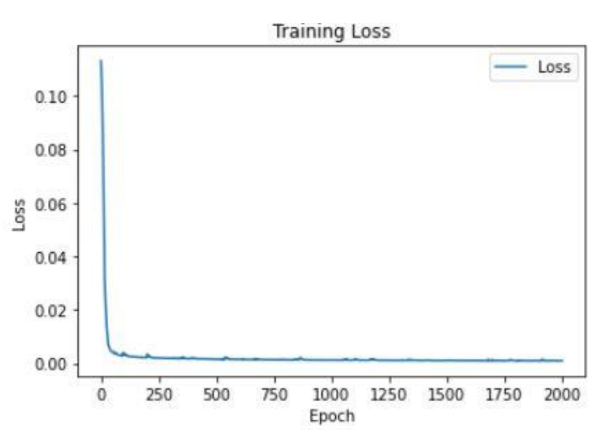


Figure 17: Training loss at First case

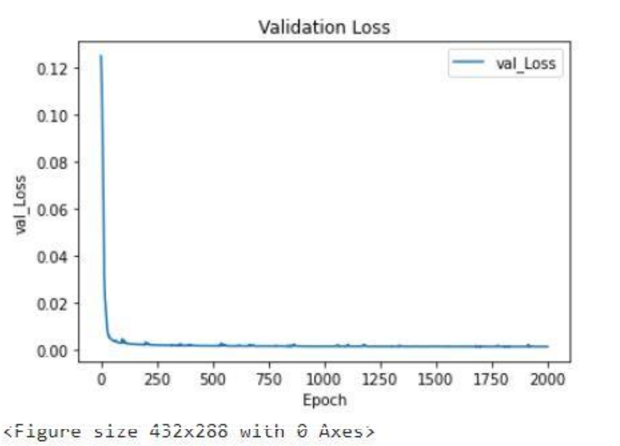


Figure 18: Validation loss at First case

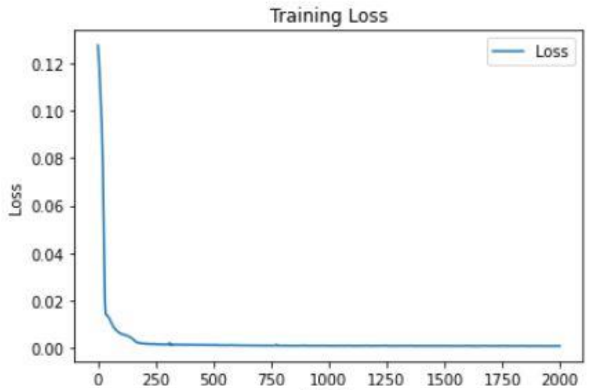


Figure 19: Training loss at Second case

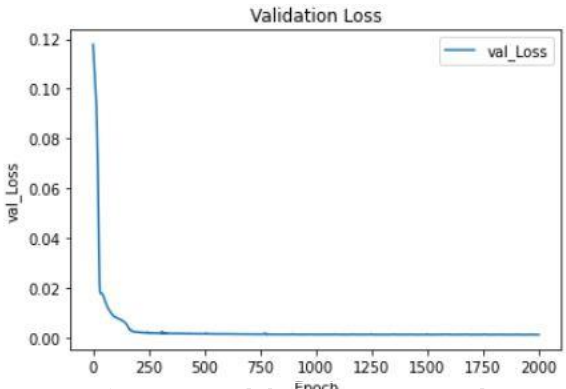


Figure 20: Validation loss at Second case

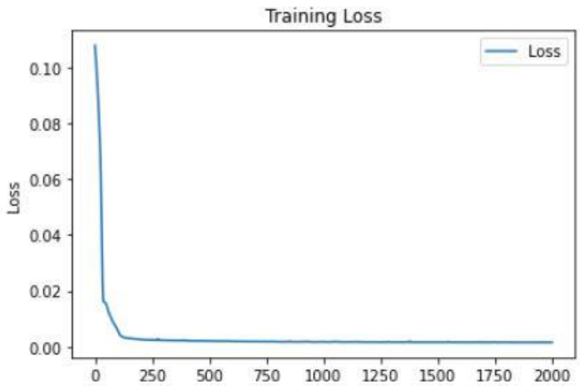


Figure 21: Training loss at Third case

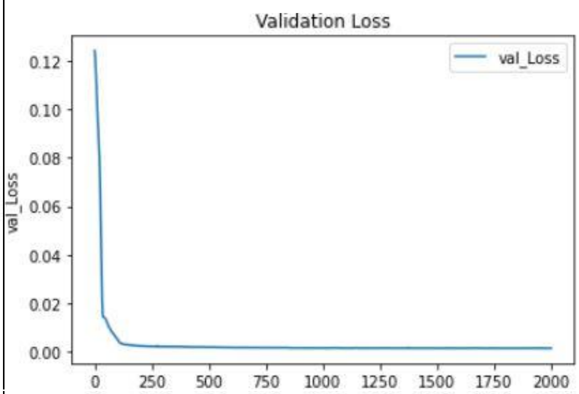


Figure 22: Validation loss at Third case

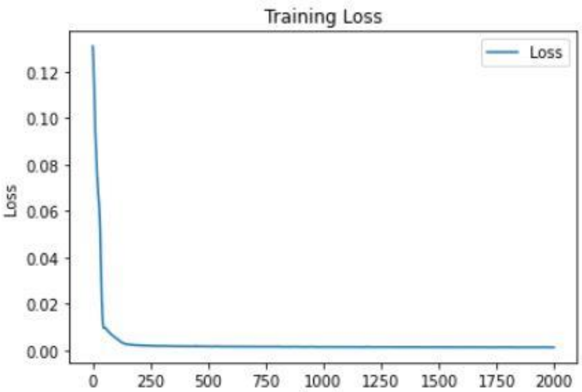


Figure 23: Training loss at Fourth case

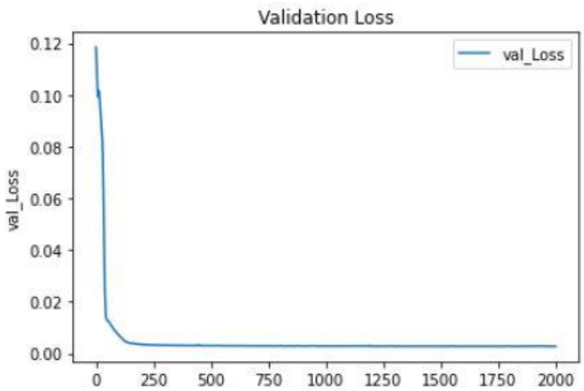


Figure 24: Validation loss at Fourth case

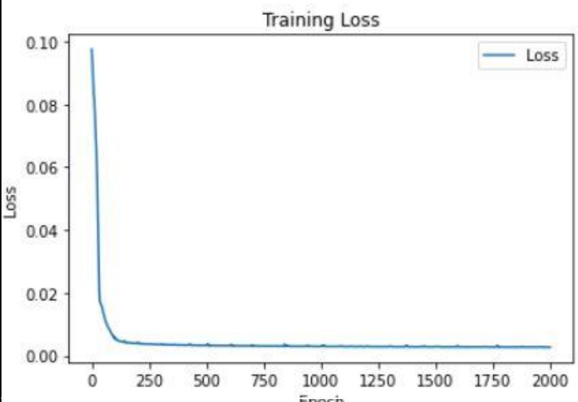


Figure 25: Training loss at Fifth case

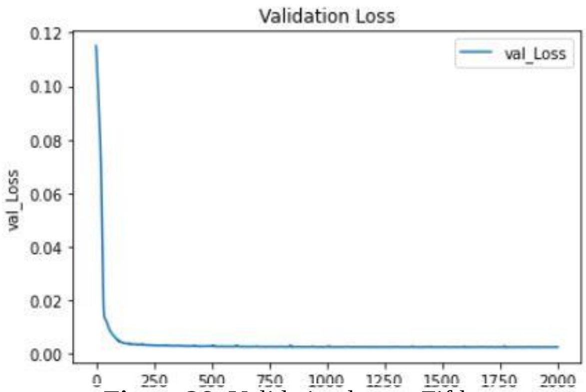
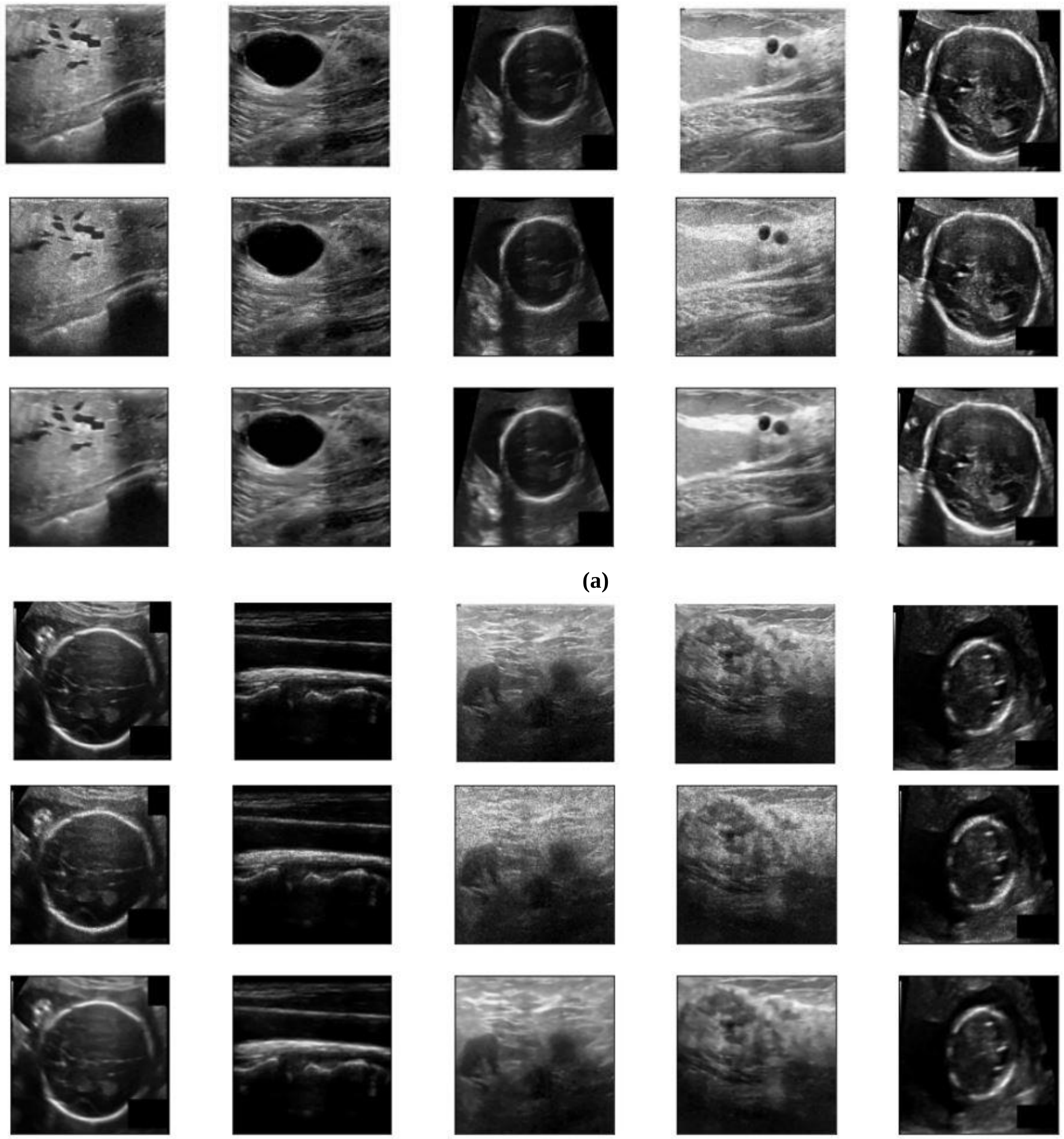


Figure 26: Validation loss at Fifth case

After training stage has been completed, the model has been saved in google drive as model.png file. It is ready to be tested and evaluated.

First, Noisy image selected from the testing dataset then processed by the trained network which result the enhanced image. Then, the ground truth of the enhanced image selected to compare with the enhanced image. This comparison is done in order to evaluate and estimate the trained network and determining the SSIM and PSNR values. Figure 27 illustrates testing phase results included the original image, noisy image and finally the reconstructed image, from top to bottom, and their numerical measurements in table 2.

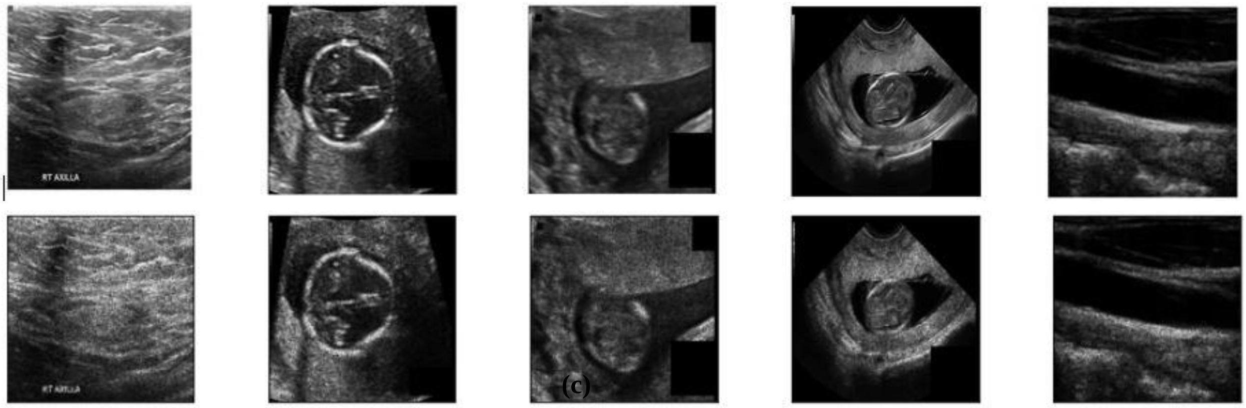


(a)

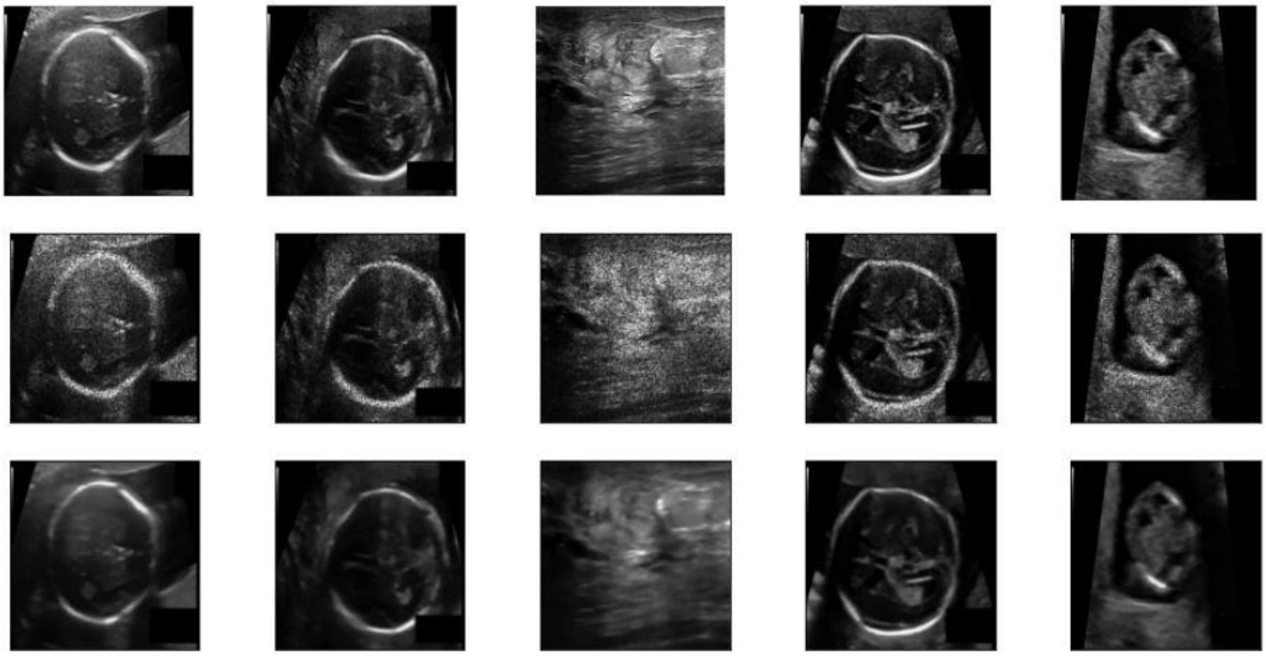
(b)

Figure 27 output images in testing phase, original image, noisy image and reconstructed image, from top to bottom.

(a) At Var = 0.05, (b) At Var = 0.1.



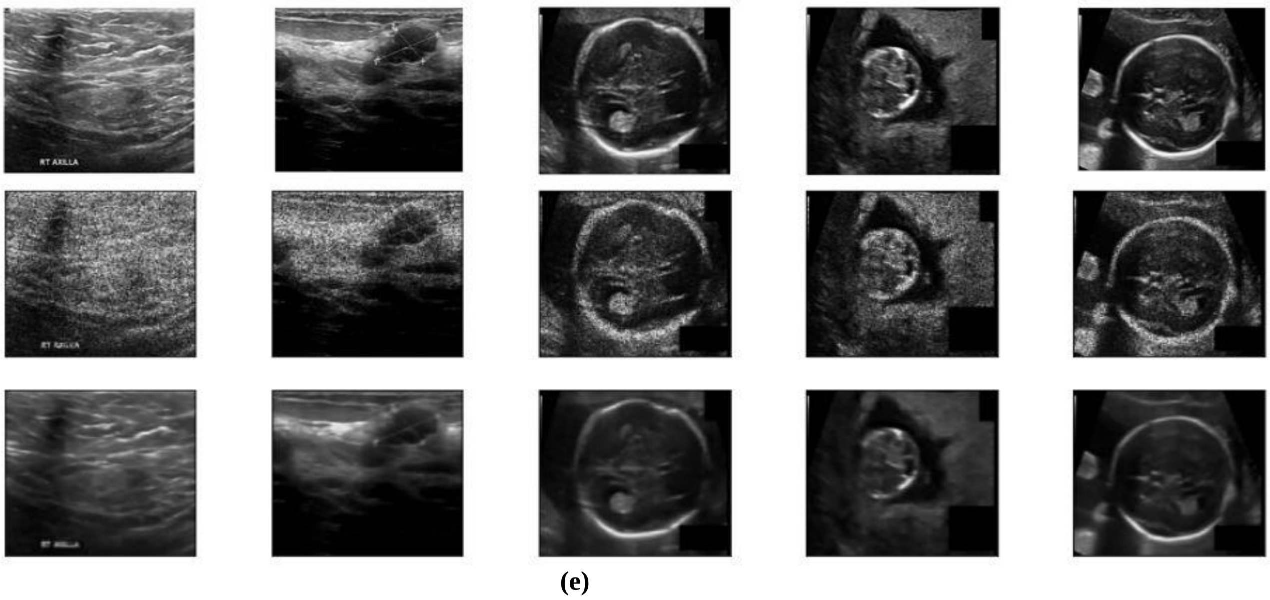
(c)



(d)

Figure 27 output images in testing phase, original image, noisy image and reconstructed image, from top to bottom.

(c) At Var = 0.25, (d) At Var = 0.5



(e)

Figure 27 output images in testing phase, original image, noisy image and reconstructed image, from top to bottom.

(e) At Var = 1.0

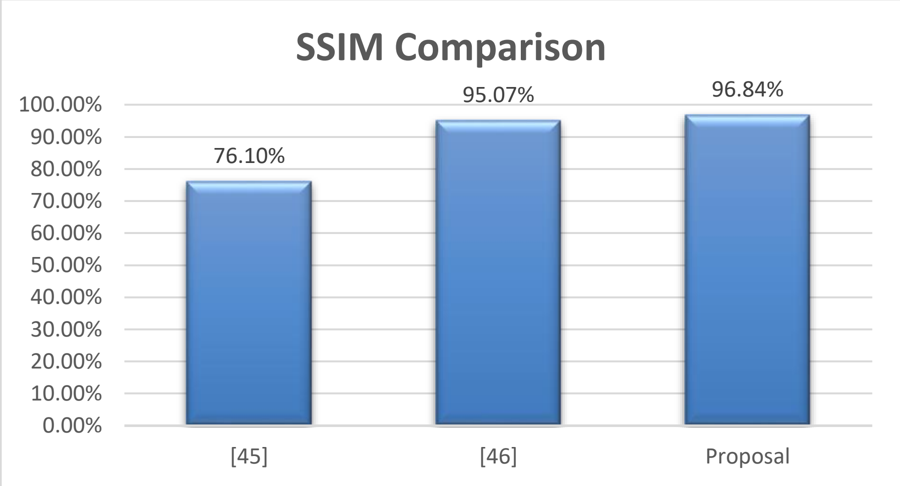
Table 2: PSNR and SSIM values for the Testing phase.

Figure 27	SSIM (%)	PSNR (dB)
(a) Var =0.05	99.028	31.636
(b) Var =0.1	98.692	30.314
(c) Var =0.25	97.4549	28.7011
(d) Var =0.5	96.835	27.9635
(e) Var =1.0	93.491	24.2891

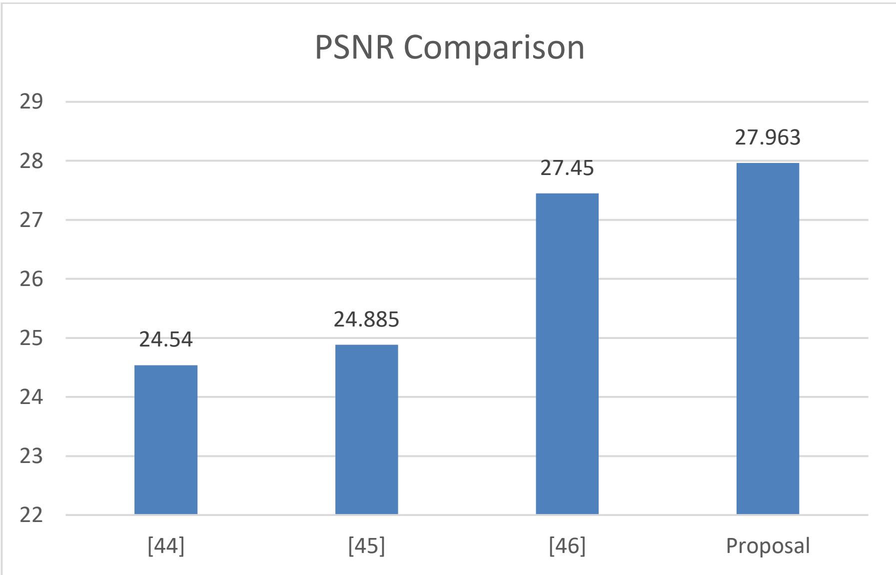
Several methods have been proposed to enhance medical ultrasound images by using convolution neural network, each one of these methods used different datasets in order to testing and training their model. Table 3 illustrates the average SSIM/PSNR values results from different method and using different datasets.

Table 3: Comparison with Previous Studies.

Method Name	No. of layers	Noise level	SSIM	PSNR
Autoencoder 20 layer	20	1.0	-	24.54
U-shaped CNN- 30 layer	30	50 AWGN	76.1	24.88
U-shaped CNN- 49 layer	49	0.5	95.07	27.45
The proposed model	18	0.5	96.835	27.9635



(a)



(b)

Figure 6.13 Comparison image enhancement models during last years (a) SSIM (b) PSNR.

7. Conclusions

After implementing of CNN model based on U-Net architecture computing and comparing the results of all situations used in this thesis, the following were concluded:

According to the obtained results, the study has concluded the following:

- CNN is like all deep learning techniques; it is very relying on the size and quality of the training data. Using good dataset (good resolution images, large number and convenient size of images) to train the model result in better performance.
- Several experiments were performed while training the model by using different parameters and epoch number found that decreased the complexity of the model, mean decrease in training time and required storage.
- According to enhance general and medical images, it is not required that the number of layers be many in order to obtain clear images, and this is illustrated with our model and the aforementioned experiments.
- U-Shaped (U-Net) of CNN technique is a better technique for Denoising images, resolution enhancement than other like autoencoder etc.
- By eighteen layers, U-Net model, custom dataset of Ultrasound images and python language, the best SSIM has been gotten is 99.028% and 98.692% at Noise level 0.05 and 0.1 respectively.

8. REFERENCES

[1] Du, B., Wei, Q., Liu, R., 2019. “An improved quantum-behaved particle swarm optimization for endmember extraction” IEEE Transactions on Geoscience and Remote Sensing.

[2] Li, S., Yin, H., Fang, L., “Group-sparse representation with dictionary learning for medical image denoising and fusion”. IEEE Transactions on biomedical engineering 59 (12), 3450–3459, 2012.

[3] Khadeja M. Mohamed, Dr. Mohammed H. Ali, "Medical B-Mode Ultrasound Imaging Reconstruction Algorithms: Evaluation and Simulation”. Asian Journal of Convergence in Technology, 2021.

[4] Dr J. Shanbehzadeh, M.Gholizadeh, “Digital Image Processing” 2015.

[5] P.S. Hiremath, Prema T. Akkasaligar and Sharan Badiger, “Speckle Noise Reduction in Medical Ultrasound Images” Advancements and Breakthroughs in Ultrasound Imaging, 2013.

[6] Image processing 101, dynamsoft, Available: https://www.dynamsoft.com/Documents/image%20processing_compressed.pdf

[7] Shengfeng Liu, Yi Wang, Xin Yang, Shawn Xiang Li, “Deep Learning in Medical Ultrasound Analysis: A Review” Elsevier LTD, 2019.

[8] Ines N., Olfa S., Khalil CH., Ahmed H., “Speckle Noise Reduction in Breast Ultrasound Images SMU (Srad Median Unsharp) Approach” Eighth International Multi-Conference on Systems, Signals & Devices, 2011.

[9] Faouzi Benzarti, Hamid Amiri, “Speckle Noise Reduction in Medical Ultrasound Images” International Journal of Computer Science Issues, 2013.

[10] Shyh-Kuang Ueng. Guan-Zhi Chen, “Ultrasound Image Enhancement Using Structure-Based Filtering” Computational and Mathematical Methods in Medicine, 2014.

[11] Chao Dong, Chen Change Loy, Xiaoou Tang, “Image Super-Resolution Using Deep Convolutional Networks” Computer Vision and Pattern Recognition, 2015.

[12] 2018, https://www.techopedia.com/definition/33295/deep-residual-network-deep-resnet

[13] G. Chierchia, D. Cozzolino, G. Poggi, L. Verdoliva, “SAR Image Despeckling through Convolutional Neural Networks” Computer Vision and Pattern Recognition, 2017.

[14] Hamid Reza Shahdoosti, Zahra Rahemi, “Edge-preserving image denoising using a deep convolutional neural network” Elsevier Signal Processing, 2019.

[15] Hmrishav Bandyopadhyay. “An Introduction to Autoencoders: Everything You Need to Know” 2022. Available: https://www.v7labs.com/blog/autoencoders-guide

[16] Raphael Couturier, Gilles Perrot, Michel Salomon, “Image Denoising Using a Deep Encoder-Decoder Network with Skip Connections” International Conference on Neural Information Processing, 2019.

[17] Toqa A. Sadoon, Dr. Mohammed H. Ali, "An Overview of Medical Images Classification based on CNN", International Journal of Current Engineering and Technology,2020.

[18] Y. Jones, “Deep Learning for Image Enhancement and Visibility Improvement,” Advanced Reading, 2018.

[19] Anirudha Ghosh, Abu Sufian, Farhana Sultana, Amlan Chakrabarti and Debashis De, “Fundamental Concepts of Convolutional Neural Network,” Recent Trends and Advances in Artificial Intelligence and Internet of Things, 2020.

[20] Sultana, F., Sufian, A., Dutta, P., “Advancements in image classification using convolutional neural network” International Conference on Research in Computational Intelligence and Communication Networks, 2018

[21] L. Spreeuwers, "Image Filtering With Neural Neural Netwoks Application and Performance Evaluation". 1992.

[22] Carlos Hung, Aldred De Leon, Jhun Sampaga, Carlos Candano, Neol Calangi, “Color Detection Software using the program matlab 6.5” Available: http://lbycp13project.yolasite.com/project.php

[23] Angel Das, "Convolution Neural Network for Image Processing — Using Keras" 2020. Available: https://towardsdatascience.com/convolution-neural-network-for-image-processing-using-keras-dc3429056306

[24] L. Spreeuwers, "Image Filtering With Neural Neural Netwoks Application and Performance Evaluation". 1992.

[25] Sumanth Chennupati, “Hierarchical Decomposition of Large Deep Networks”, 2016. Available: https://www.researchgate.net/publication/312935261_Hierarchical_Decomposition_of_Large_Deep_Networks

[26] M. Nielsen, “Neural Networks and Deep Learning,” Artif. Intell., 2018.

[27] “A Gentle Introduction to Convolutional Neural Networks” Available:

https://chatbots.marketing/a-gentle-introduction-to-convolutional-neural-networks/

[28] H. Le and A. Borji, “What are the Receptive, Effective Receptive, and Projective Fields of Neurons in Convolutional Neural Networks,” 2017.

[29] J. Murphy, “An Overview of Convolutional Neural Network Architectures for Deep Learning,” Microway, Inc., 2016.

[30] Arun Prakash, "Machine Learning - Convolution for image processing” 2018. Available: https://blog.francium.tech/machine-learning-convolution-for-image-processing-42623c8dbec0

[31] Arun Prakash, Machine Learning - Convolution for image processing, 2018. Available:

https://www.researchgate.net/figure/An-example-of-CNN-architecture_fig1_320407067

[32] Y. Lecun, Y. Bengio, and G. Hinton, “Deep learning,” Nature, 2015.

[33] A. Kaur, G. Kaur, “A Review on Image Enhancement With Deep Learning Approach,” Accent. Trans. Image Process. Comput. Vis., vol. 4, no. 11, pp. 16–20, 2018.

[34] Melina Tourni, “Texture Analysis of Histopathology Slides for the prediction of EGFR Gene Mutation”, National Technical University of Athens, 2018.

[35] Mars Xiang, “Convolutions: Transposed and Deconvolution”, 2020. Available: https://medium.com/@marsxiang/convolutions-transposed-and-deconvolution-6430c358a5b6

[36] Caleb Vununu, Kwang-Seok Moon, Suk-Hwan Lee, Ki-Ryong Kwon, “A Deep Feature Learning Method for Drill Bits Monitoring Using the Spectral Analysis of the Acoustic Signals”,Sensors, 2018.

[37] Ronneberger, O., Fischer, P., Brox, T., “U-net: Convolutional networks for biomedical image segmentation” Medical Image Computing and Computer-Assisted Intervention (MICCAI), Springer, 2015

[38] Nabil Ibtehaz, M. Sohel Rahman, “MultiResUNet : Rethinking the U-Net architecture for multimodal biomedical image segmentation,” 2020.

[39] Deshpande, R.G., Ragha, L.L. and Sharma, S.K. “Video Quality Assessment through PSNR Estimation for Different Compression Standards” Indonesian Journal of Electrical Engineering and Computer Science, 2018.

[40] Wang, Z. and Sheikh, H.R., “Image Quality Assessment: From Error Visibility to Structural Similarity. IEEE Transactions on Image Processing,” 2004.

[41] Wang, Z., Simoncelli, E.P. and Bovik, A.C., “Multiscale Structural Similarity for Image Quality Assessment” Conference Record of the Thirty-Seventh Asilomar Conference on Signals, Systems and Computers, 2014.

[42] Arya Shah, “Breast Ultrasound Images Dataset”, Available: https://www.kaggle.com/aryashah2k/breast-ultrasound-images-dataset

[43] Thomas L. A. van den Heuvel, Dagmar de Bruijn, Chris L. de Korte, Bram van Ginneken, “Automated measurement of fetal head circumference using 2D ultrasound images”, Available: https://zenodo.org/record/1327317#.Ye7nQ-pBzIU

[44] Rina Komatsu, Tad Gonsalves, “Comparing U-Net Based Models for Denoising Color Images” MDPI, 2020.

[45] Sandeep N. Menon, V. B. Vineeth Reddy, A. Yeshwanth, B. N. Anoop, Jeny Rajan, “A Novel Deep Learning Approach for the Removal of Speckle Noise from Optical

Coherence Tomography Images Using Gated Convolution–Deconvolution Structure” Proceedings of 3rd International Conference on Computer Vision and Image Processing, 2020.

[46] Alexander L. Klibanov, Ph.D. and John A. Hossack, Ph.D. “Ultrasound in Radiology: from Anatomic, Functional, Molecular Imaging to Drug Delivery and Image-Guided Therapy”, 2015. Available: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4580624/