Chapter 5

An Augmented Deep Learning Network with Noise Suppression feature for Efficient Segmentation of Cardiac MR Images

Highlights of the Chapter

- *Deep Learning method for segmentation of noisy CMRI images.*

- *Incorporation of noise stifler block.*

Contribution of the chapter

The segmentation of cardiac MR images requires extensive attention as it needs a high level of care and analysis for the diagnosis of affected part. The advent of deep learning technology has paved the way for efficient, automatic and reliable segmentation of medical images for proper diagnosis. This chapter presents a deep learning based model for effective segmentation of noisy MR images. The model incorporates the depth wise separable convolution and group normalization as basic building blocks. Moreover, a noise stifler block is also induced between the encoder and decoder to counter the noise in the medical images. This path helps in delineating the precise boundary contours as the noise often reduces the boundary segmentation capability of the segmentation network. The network trained once produces exceedingly good results for the images of other datasets. An improvement of above $(5 \pm 0.03)\%$ and $(3.5 \pm 0.02)\%$ was observed in the Jaccard index and Dice score for cardiac MR images. The results are statistically validated as $p < 0.05$. The automatic computer investigated approach can help in reducing the burden on the medical system by producing accurate and reliable results. The algorithmic results were clinically verified by the senior radiologists by comparison with the manually segmented images. The training time of the network was about 30% less than U-net.

5.1 Introduction:

Image segmentation is a process that subdivides an image into its constituent parts or region of interest. The problem under analysis determines the level of subdivision in the image. It is a low-level operation as it operates pixel-wise. A corresponding category label is provided to each pixel in the image by the image segmentation operation. The basic requirement for medical image segmentation is to study the changes in pathological and anatomical structures for analysis of the problem. The accurate and precise extraction of features forms the area being segmented is reasonably necessary for the medical experts to provide the correct diagnosis. The primary medical image segmentation task includes segmentation of brain tumour, skin cancer tissues segregation, cardiac image segmentation, analysis of ventricle cavities by image segmentation, outlining of liver tumour, optical disc segmentation, pulmonary nodules, cell segmentation etc. The advancement in medical imaging technology has paved the way for efficient analysis of the disease affected regions. X-Ray [128], Computed Tomography (CT) [129], MRI [111] and Ultrasound [130] are the main modalities in the medical imaging domain, which are widely used for the detection and analysis of diseases. Thus extreme attention is given to the images of these modalities for segmentation and other imaging domain applications.

The primitive approaches for the medical image segmentation were dependent on active contour matching, statistical models, edge locations template matching. Basically, image segmentation is categorized into manual, semi-automatic, and fully automatic segmentation. Manual segmentation requires the experience and expertise of radiologists and doctors to draw a precise region of interest (ROI) which is time-consuming, often impractical for a large dataset, and involves significant variabilities as well. The semi-automatic methods require handcrafted features as well as manual interference for the processing steps [86]. The fully automatic methods which are gaining importance has to lead the way for the computer-based

approaches to fully take over the feature extraction as well as segmentation process. With the advent in the field of deep learning, fully automatic methods are producing reliable and accurate results. These methods are self-capable in feature extraction and learning. Deep learning-based architectures use supervised learning techniques for the processing of images. The revolutionary work in the field of medical image segmentation was presented by Olaf Ronneberger in the form of U-net [30]. The end to end trainable U-net architecture is capable of producing very accurate results while getting trained on very few images. The advantage of U-net is that input uses data patches which solves the problem of fewer data as the patches are equivalent to the data augmentation. Moreover, the target category can be located by the output results. Other end to end trainable image segmenting network includes Fully Convolutional Networks (FCN) [109] [131], Deeplab [132], SegNet [94] etc. The pixel-level segmentation was achieved using the SegNet model as it incorporates the symmetric structured encoder-decoder network, which is based on the semantic segmentation concept of FCN. The FCN deploys locally connected layers and avoids the use of dense layers. The down sampling path in FCN extracts the features while the up-sampling path localizes the object. The deeplab was developed by Google as an open source model for image segmentation. It upsamples the output of the last convolutional layer and evaluates pixel-wise loss. The FCN was used by Zhou et al. for the segmentation of nineteen organs in CT images [133]. A cascaded FCN was proposed by Christ et al. [36], which was incorporated using the superposition of a series of FCN. In this approach, context features were extracted from the prediction map. The adversarial network proposed by Goodfellow et al. [134] were used by Luc et al. [135] to segment the images by training a CNN based segmentation network and an adversarial network. Segmentation Adversarial Network (SegAN) [136] proposed by Xue et al. used the U-net model as the generator for GAN. This network is subdivided into segmentor network and critic network. These two are trained alternately to finally produce a good segmentation model. A high-level

3D semantics capturing network names as Projective Adversarial Network (PAN) [137] was proposed by Khosravan et al. PAN uses a segmentor and two adversarial networks as the basic constituents of the architecture. 2D projection is used to integrate high-level 3D information without increasing the complexity in the segmentation process. Deep contour network (DCAN) [110] proposed by Hao Chen et al. generates the predictions of both objects and contours for the segmentation process. Peter Naylor et al. [113] developed a method by replacing binary annotation with a distance map to train the network to predict pixel by pixel segmentation.

The presented research's motivation is to develop an efficient deep learning-based architecture to segment the noisy MR images for reliable analysis of the disease [127]. The primary source of noise in MR images is thermal in nature, which originates from the stochastic motion of free electrons. Thermal noise is a white additive and follows Gaussian distribution having variance (σ) and zero mean (μ). The raw data for MR images is K-space data which is corrupted by Gaussian noise. The noise distribution of data changes when the magnitude of data is computed as this process is non-linear. The Gaussian noise distribution of data is converted to Rician after this process. This noise creates problems in the proper segmentation of the boundary of the objects. The main objective of this work is to effectively segment the MR images corrupted with the Rician noise as the real-time MR images contain some percentage of unwanted Rician distributed noise. The proposed is integrates noise stifler block for noise suppression at the time of segmentation. The accurate diagnosis of the disease is dependent on the perfection in interpretability of the image. Thus precise and accurate segmentation is quite helpful for the medical practitioner to plan the diagnosis. The proposed model produces more and accurate results in comparison to its conventional counterpart.

5.2 Methodology:

This section presents the methodology of the proposed architecture, which is incorporated using U-net architecture as the backbone. The proposed network comprises depth wise

separable convolution, group normalization, Scaled exponential Linear Unit (SELU) and noise stifler block. The depth wise separable convolution [74] with bottleneck connections [96] is used in this end to end trainable segmentation network, as shown in figure 5.1. The in-between inputs are encoded by the bottleneck connections. The transformation from lower-level units such as pixels to higher-level illustration such as image is performed by the middle layers. It consists of three convolution layers in the block. The layers are named as 1) Expansion Layer, 2) Depth wise convolutional Layer and, 3) Projection Layer. For each input channel, a dedicated single filter is incorporated. A pointwise 1×1 convolution is used for the process of combining the outputs of depth wise separable convolution. Filtering and combining is collectively termed as depth wise separable convolution. This process is split into two parts, a different layer for filtering and a different layer for combining, as shown in figure 5.1. Figure 5.2 shows the overall architecture of the network.
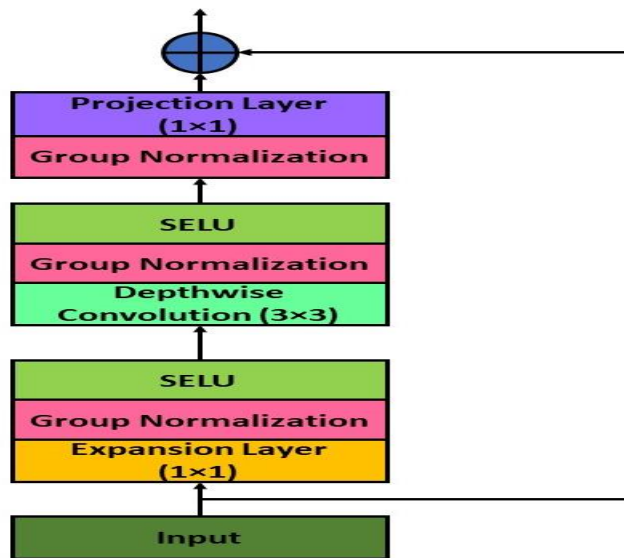

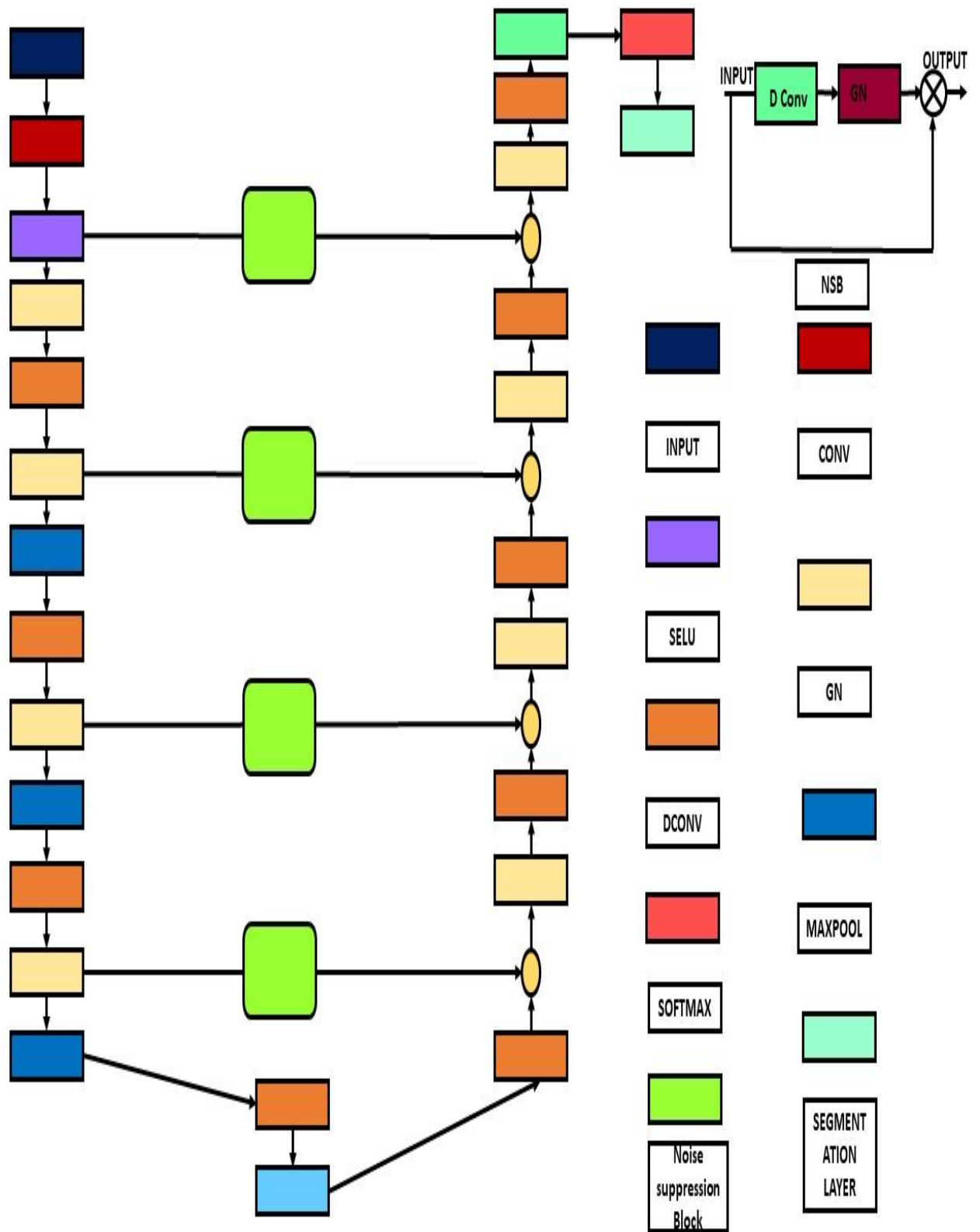***Figure 5.1: Modified Depth wise Separable block.***

Figure 5.2: Architecture of the proposed network.

Depth wise convolution with one filter per input channel (input depth) can be written as:

$$\hat{G}_{k.l,m} = \hat{K}_{i,j,m}.F_{k+i-1,l+j-1,m} \qquad (5.1)$$

Where is the depth wise convolutional kernel of size $D_K \times D_K \times M$ where the mth filter is applied to the mth channel to produce the mth channel of the filtered output feature map. The following equation gives the cost of convolution for depth wise separable convolution:

$$D_K.\ D_K.M.\ D_F.\ D_F + M.N.\ D_F.\ D_F \qquad (5.2)$$

Equation 2 is the sum of depth wise convolution and 1×1 pointwise convolution. This 1×1 pointwise convolution layer causes the number of channels to be lower and is also known as the Projection Layer and Bottleneck. The projection layer projects the data with higher dimensions into a tensor with a lower number of dimension. The first layer in figure 5.1 is named the expansion layer, which possesses the opposite characteristics of the projection layer. It has less number of input channels than the output channels. The hyperparameter expansion factor determines how much the data will be expanded. Each layer incorporates group normalization and Scaled exponential Linear Unit (SELU) as activation function. The use of non-linearity may destroy meaningful information, so the use of activation function is avoided in the projection layer. The overall cost of the computation is reduced by the incorporation of depth wise separable convolution because of the reduction in the number of multiplications. The training parameters are also less as compared to standard convolution. Thus overfitting issues are reduced to a large extent. The image is transformed once, so the computation complexity is reduced. The boundary and object-related information are precisely captured in the case of depth wise separable convolution. This is because, in the case of regular convolution, an image is transformed m times where m denotes the number of channels, while in the case of depth wise separable convolution, the image is transformed only once and elongated to m channels. Scaled exponential unit (SELU) is used instead of RELU as the former shows robust performance than the latter. The SELU cannot die because of the negative slope, and thus it removes the problem of vanishing gradient in deep neural networks. It possesses the

self-normalization property. SELU [138] is a trade-off between RELU and Leaky RELU as the constant slope of the leaky RELU may elevate the risk of vanishing gradient. The SELU is given by the following equation:

$$f(\alpha, x) = \tau \begin{cases} \alpha e^x & for \ x < 0 \\ x & for \ x \geq 0 \end{cases}$$ (5.3)

For normalizing the data, group normalization [115] is used, which normalizes the data across the groups. The channels is divided into groups before the normalization. The accuracy is stable over a wide batch size in case of group normalization as there is no dependence on batch size. Thus the main drawback of handling the large batch size of batch normalization is eliminated by the use of group normalization. The process of group normalization requires the computation of the mean and variance of the inputs over the specified groups of channels. The following equation describes group normalization:

$$\hat{a}_\iota = \frac{a_i - \mu_g}{\sqrt{\sigma_g^2 + \varepsilon}}$$ (5.4)

Where a_i is the input, μ_g is the mean and is the variance. ε is the property to improve numerical stability for the slight group variance.

The max-pooling operation is performed in the subsequent stage with window size 2×2 and stride 2. The related contents of the features are rejected by this layer, and the features with vital information are stored. Translational invariance is achieved over a small spatial shift using this max-pooling operator. The SoftMax operator is applied next to the input data, which turns the raw score, i.e. logits, into probability which sums to 1. It acts as a multiclass sigmoid function. The SoftMax function [101] is used for the determination of probability computed once of multiclass. The following equation represents the SoftMax function:

$$Y_r(x) = \frac{\exp(a_r(x))}{\sum_{j=1}^k \exp(a_j(x))}$$ (5.5)

Where r signifies the class $0 \leq Yr \leq 1$, and the inputs and outputs of the layer are denoted by x and Y, respectively. The segmented result generated by the softmax corresponds to the class with the maximum probability for each pixel in the image. Finally, a categorical label is provided to every pixel in the image by the pixel classification layer. For optimisation of network parameters, the cross-entropy loss function [139] is used in the network. The loss was calculated using the following equation:

$$loss = -\frac{1}{N}\sum_{i=1}^{M} T_i \log(X_i) \qquad (5.6)$$

Where Xi and Ti are the response of network and target value, the total number of responses in the image is denoted by M, and N is the total number of response in X. The noise stifler block is inserted between the connections from the encoder and decoder. The objective of this block is to suppress the noise from the image features. By reducing the noise, the spurious signal may be eliminated from the image feature, which is the unwanted property of the real-time medical images. The noise stifler block is shown in figure 5.3. This block performs pointwise multiplication of group normalized depth wise convolved features with the original features. The pointwise multiplication reduces the high frequency noise components. Thus the noise in the feature maps gets reduced which is quiet desirable for effective segmentation of MR images.

5.2.1 Experimental Setup

The following steps were followed for performing the experiment:

Step 1: Rician noise [49] was added manually in the MR images. The percentages of added noise were 1%, 3%, 5% and 7%. (Rician noise was added because the noise corruption follows Rician distribution in MR images.)

Step 2: The proposed network was implemented using the tensor flow framework on Python 3.7.

Step3: The network was supplied with the original images and the noisy images for segmentation.

Step 4: The results were obtained, and evaluation metrics were calculated.

Step 5: For comparison of results, step 2 to 4 were repeated for other networks.

Step 6: The comparative study of the results of various methods was performed.

5.2.2 Datasets and Evaluation Metrics

In this paper, two datasets were used for segmentation. The heart image segmentation datasets used in this paper are Sunny Brook Dataset (SCD) and Automatic Cardiac Diagnosis Challenge (ACDC). The Sunnybrook Cardiac MR Left Ventricle Segmentation Challenge (MICCAI 2009) dataset consists of 45 CMR images from patients having four pathological conditions like heart failure with ischemia, heart failure without ischemia, hypertrophic cardiomyopathy and normal subjects. Manually drawn contours for the endocardium and epicardium are provided in text format, which consists of contour points. The automatic Cardiac Diagnosis Challenge (ACDC) dataset contains MR images of the Heart in a Short-axis (SAX) view of 100 patients with manual segmentation of LV, Myocardium, and RV for End-Systole and End-Diastole phase in the NIfTI (Neuroimaging Informatics Technology Initiative) image format. The network was trained on the ACDC dataset for heart segmentation while tested on 20% part of the same dataset and full SCD dataset. The results obtained were evaluated using the metrics: accuracy, Jaccard Index [140] [105], Dice Score and Matthews correlation coefficient (MCC) [141]. Accuracy is given as the ratio of correctly classified pixels, irrespective of class, to the total number of pixels in the dataset. Jaccard is defined as the fraction of the overlapped area between the predicted segmentation and the ground truth data

and the union of the area of segmentation being predicted and ground truth data. The Dice score is defined as the harmonic mean (F1-measure) of the precision and recall values with a distance error tolerance to decide whether a point on the predicted boundary matches the ground truth boundary or not. The Matthews correlation coefficient (MCC) is used to firmly quantitative the analysis of classification as the mathematical properties of MCC handles both dataset imbalance and their invariants effectively. It gives correct predictions for both majorities of the negative cases and positive cases, independently of their ratios in the overall dataset. The following equations describe the above stated metrics:

$$Jaccard\ (U,V) = \frac{U \cap V}{U \cup V} \qquad (5.7)$$

Where U is the predicted segmentation data, and V is the ground truth data.

$$Dice\ Score = \frac{2 \times P \times R}{R + P} \qquad (5.8)$$

Where, P and R denotes the precision and recall values respectively.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \qquad (5.9)$$

$$MCC = \frac{TP * TN - FP * FN}{\sqrt{(TP+FP)(TP+FN)(TN+FP)(TN+FN)}} \qquad (5.10)$$

Where TP, TN, FP and FN are the four cardinalities of confusion metrics.

The proposed Convolutional network was implemented using Keras API in Python 3.6 configured with TensorFlow library as the backend. The python scripts were executed on Python 3 Google compute engine backend provided by Google Collaboratory. It provides a virtual machine that has 2 x Intel(R) Xeon(R) @ 2.3 GHz CPUs, 12 GB RAM and an NVIDIA K80 GPU with 16 GB GRAM. Training of each model is done for 100 epochs with ADAM optimizer [118], and the learning rate is kept at 0.0001.

5.3 Results:

5.4 The results obtained from the experiment are presented in this section. The results are

divided in two parts:

5.3.1 Results obtained from heart segmentation ACDC dataset:

The segmentation results obtained when the network was trained and tested on the ACDC
dataset are shown in figure 5.4. The 60% data was used for training, 20% data was used for
validation and rest 20% was used for testing. Table 5-1 shows the comparison of evaluation
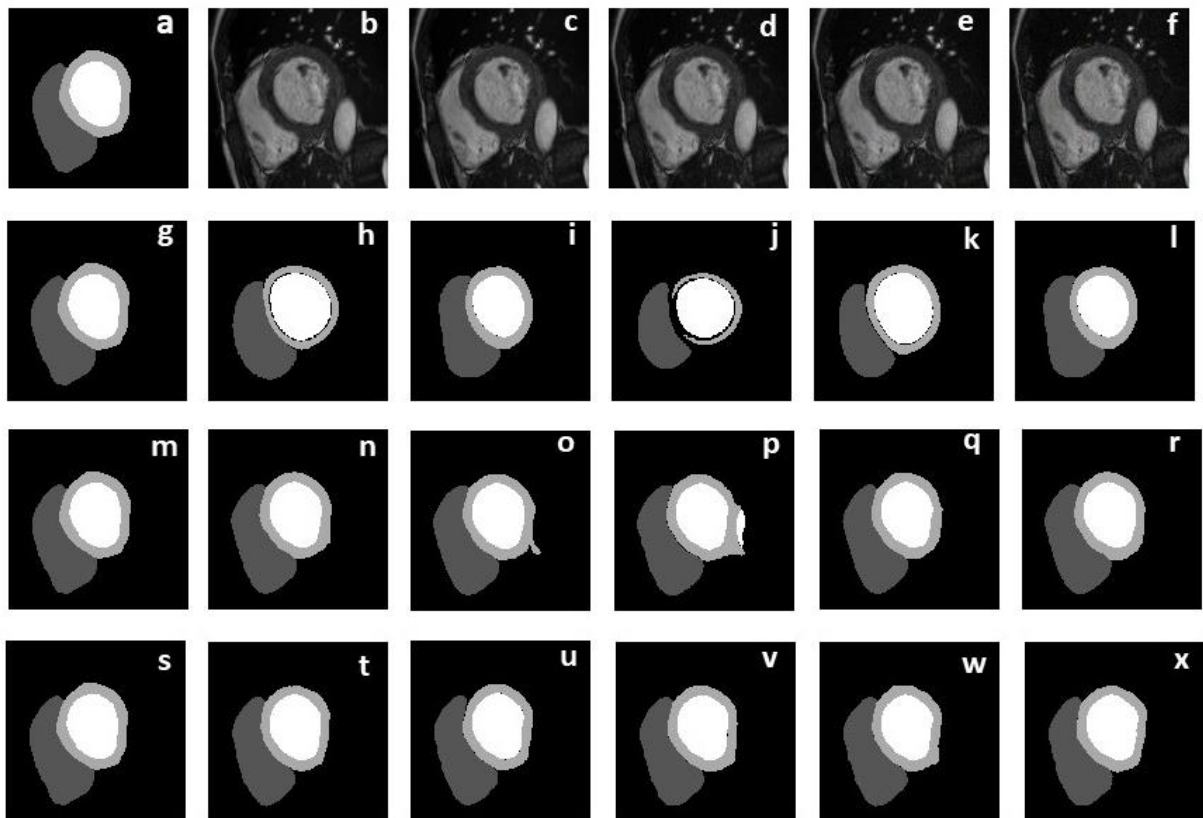metrics for the various networks in the study, along with the proposed network.

*Figure 5.3: Segmentation results of ACDC dataset. First column (a)-(s)presents the mask
of the image being segmented, first row (b)-(f) presents the original image and noisy version
(1%,3%,5% and 7%) of the original image. Second, third and fourth rows presents the
segmented results for Seg-Net (h)-(l), U-net (n)-(r) and proposed network (t)-(x) . Second,
third, fourth, fifth and sixth column presents the segmented results for original (h)-(t), 1%
(i)-(u), 3% (j)-(v), 5% (k)-(w) and 7%(l)-(x) noise corrupted images.*

Table 5-8: Evaluation Metrics Comparison for ACDC dataset.

Evaluation Metric		Original Image			1%			3%			5%			7%		
		LV	MYO	RV	LV	MYO	RV	LV	MYO	RV	LV	MYO	RV	LV	MYO	RV
Jaccard	Segnet	0.610837	0.390525	0.448898	0.668688	0.53187	0.491962	0.478324	0.19922	0.208784	0.620809	0.429762	0.386698	0.354944	0.459648	0.4362
	Unet	0.829327	0.77696	0.74518	0.857205	0.781452	0.765546	0.837983	0.747099	0.736681	0.84869	0.775356	0.742175	0.844648	0.769157	0.758487
	Proposed	**0.848739**	**0.822056**	0.815148	**0.886603**	**0.817429**	**0.827207**	**0.880581**	**0.784184**	**0.842265**	**0.872142**	**0.795444**	**0.806627**	**0.879148**	**0.793819**	**0.82564**
Dice score	Segnet	0.716276	0.525729	0.542069	0.763185	0.654547	0.574678	0.595965	0.305505	0.288754	0.72936	0.567226	0.473547	0.453706	0.638954	0.34958
	Unet	0.904913	0.870324	0.840163	0.903053	0.860088	0.824019	0.886519	0.836551	0.795343	0.895397	0.850741	0.797711	0.891176	0.853484	0.818647
	Proposed	**0.934188**	**0.897406**	**0.882229**	**0.935823**	**0.887128**	**0.895023**	**0.928857**	**0.872175**	**0.905266**	**0.924101**	**0.87634**	**0.881922**	**0.92311**	**0.881606**	**0.893077**
MCC	Segnet	0.760895	0.6094	0.713109	0.807558	0.743136	0.776683	0.658909	0.40216	0.520122	0.765388	0.643301	0.700705	0.643241	0.53903	0/684638
	Unet	0.93519	0.861342	0.850354	0.933591	0.871755	0.883905	0.922448	0.86196	0.87915	0.930767	0.876979	0.892227	0.923174	0.879801	0.894616
	Proposed	**0.955498**	**0.897416**	**0.883573**	**0.955837**	**0.904334**	**0.916292**	**0.948268**	**0.890859**	**0.905882**	**0.955853**	**0.893264**	**0.904389**	**0.949682**	**0.898999**	**0.915058**
Accuracy	Segnet	0.986579	0.971801	0.982969	0.989042	0.979366	0.985651	0.981115	0.963649	0.971592	0.986302	0.973186	0.981022	0.977316	0.97175	0.98465
	Unet	0.990899	0.992372	0.991955	0.996695	0.991359	0.994336	0.991319	0.99099	0.993454	0.991785	0.992051	0.990133	0.991591	0.991666	0.990084
	Proposed	**0.996185**	**0.998687**	0.996225	**0.996178**	**0.991881**	**0.996374**	**0.99627**	**0.991097**	**0.994006**	**0.995483**	**0.996748**	**0.99162**	**0.995946**	**0.995606**	**0.993131**

5.3.2 Results obtained from heart segmentation SCD dataset:

The trained network was tested on the SCD dataset, and the obtained results are shown in figure 5.4. The comparative study of evaluation metrics is presented in Table 5-2. The comparison of sensitivity, specificity and precision is shown in figure 5.5 for the SCD dataset.
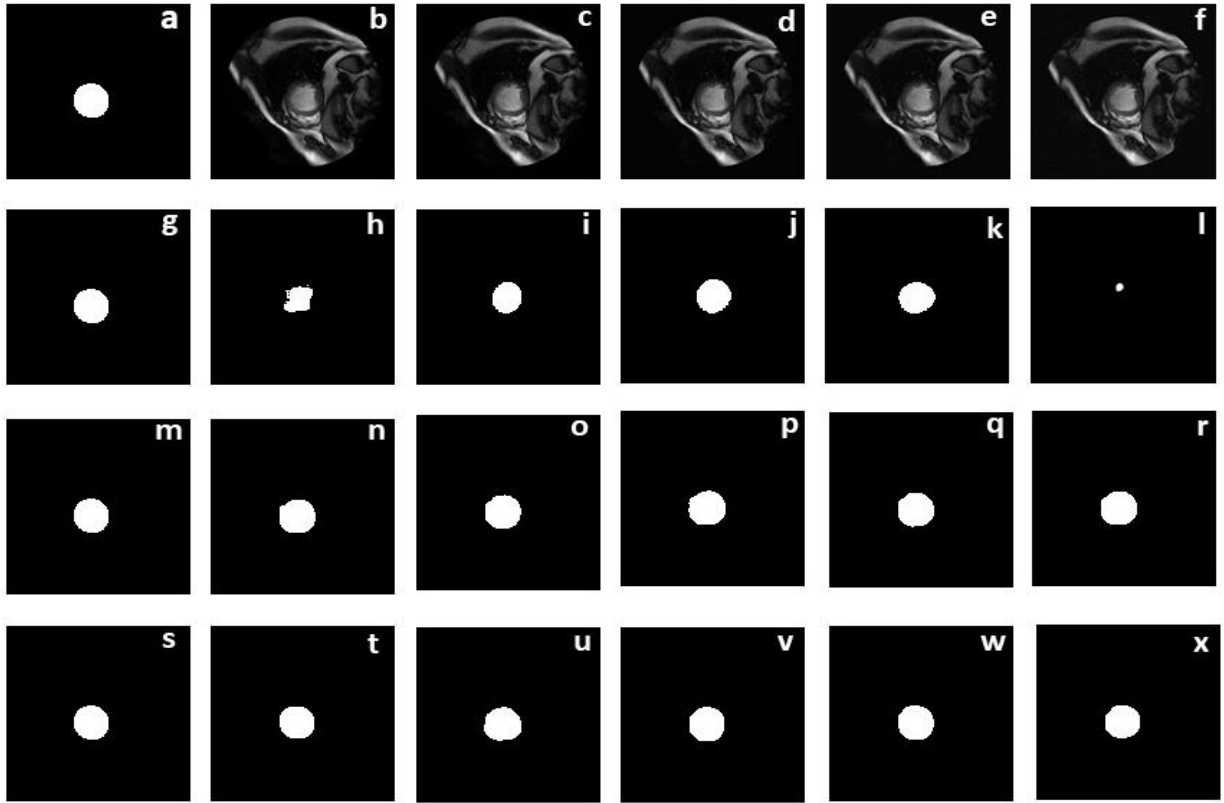
Figure 5.4: Segmentation results of SCD dataset. First column (a)-(s)presents the mask of the image being segmented, first row (b)-(f) presents the original image and noisy version (1%,3%,5% and 7%) of the original image. Second, third and fourth rows presents the segmented results for Seg-Net (h)-(l), U-net (n)-(r) and proposed network (t)-(x) . Second, third, fourth, fifth and sixth column presents the segmented results for original (h)-(t), 1% (i)-(u), 3% (j)-(v), 5% (k)-(w) and 7%(l)-(x) noise corrupted images.
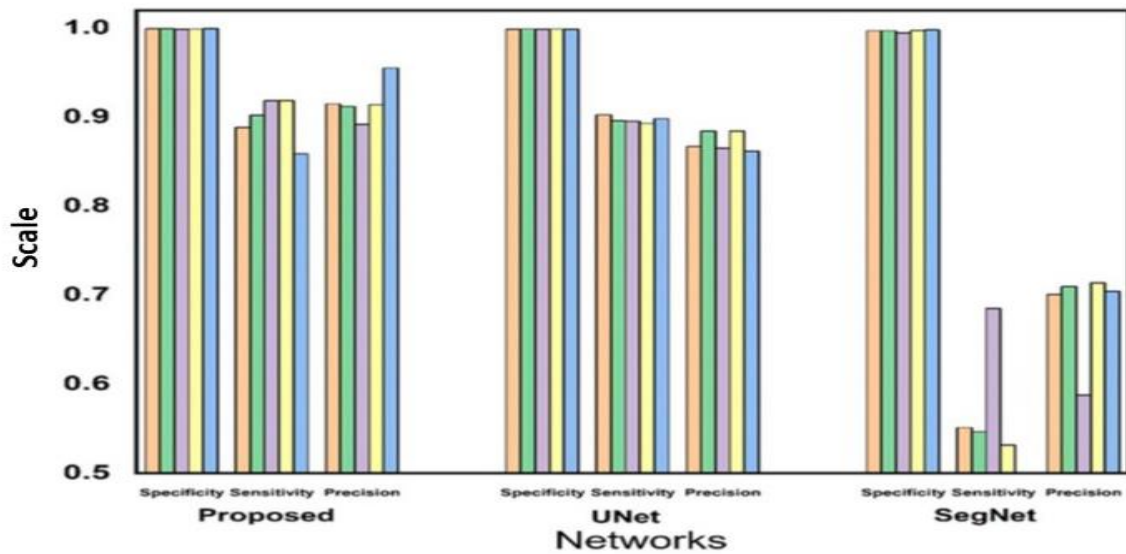

Figure 5.5: Comparison of Specificity, Sensitivity and Precision for the networks. First to fifth bars of each metric presents results for original image, 1%, 3%, 5% and 7% noise corrupted images respectively.

Table 5-9: Evaluation Metrics Comparison for SCD dataset.

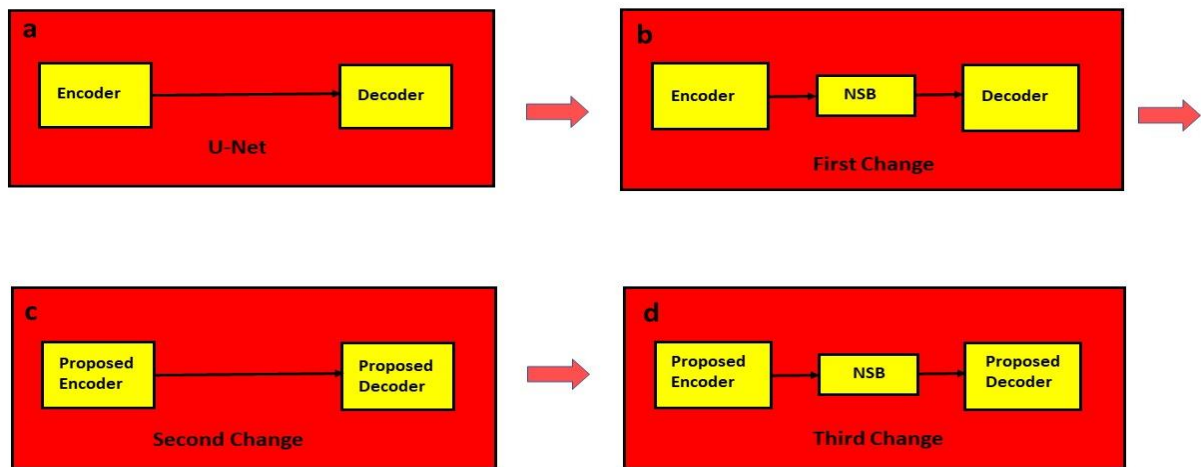| Evaluation Metric		Images with % noise				
		Original Image	1%	3%	5%	7%
Jaccard	**Segnet**	0.404762	0.39145	0.414578	0.399341	0.280637
	Unet	0.80289	0.792754	0.781819	0.776972	0.755668
	Proposed	**0.840893**	**0.828769**	**0.816192**	**0.81158**	**0.802293**
Dice score	**Segnet**	0.528635	0.521266	0.561425	0.527443	0.389564
	Unet	0.860453	0.85487	0.845826	0.83031	0.82461
	Proposed	**0.894782**	**0.880821**	**0.889817**	**0.875198**	**0.860907**
MCC	**Segnet**	0.619621	0.602494	0.590257	0.603583	0.546244
	Unet	0.873165	0.869793	0.860283	0.851614	0.832359
	Proposed	**0.910548**	**0.900022**	**0.89506**	**0.884658**	**0.867582**
Accuracy	**Segnet**	0.991418	0.990479	0.989066	0.991015	0.989112
	Unet	0.997342	0.998445	0.996254	0.997285	0.998141
	Proposed	**0.997789**	**0.998946**	**0.997865**	**0.998063**	**0.994**1

5.3.3 Ablation Study:

Figure 5.6: Layout plan of Ablation study.

Table 5-3: Ablation Study: Evaluation Metrics Comparison of SCD dataset for noise free original images.

Network Arrangement	Jaccard Index	Dice Score	MCC
(a)	0.80289	0.860453	0.873165
(b)	0.81983	0.86886	0.87946
(c)	0.82974	0.87602	0.88012
(d)	**0.840893**	**0.894782**	**0.910548**

Table 5-4: Ablation Study: Evaluation Metrics Comparison of SCD dataset for 7% noise corrupted images.

Network Arrangement	Jaccard Index	Dice Score	MCC
(a)	0.755668	0.82461	0.832359
(b)	0.77082	0.83198	0.84054
(c)	0.78264	0.85035	0.84925
(d)	**0.802293**	**0.860907**	**0.867582**

The ablation study [142] was performed to study the network performance. Figure 5.6 shows the layout plan for the ablation study. This study was performed on SDC dataset for original, and 7% noise corrupted images. Figure 5.6(a) shows the conventional U-net as segmentation network, (b) shows U-net with NS block as segmentation network, (c) shows Modified proposed encoder-decoder network without NS block and shows Modified proposed encoder-decoder network with NS block. The changes observed for segmentation of images without noise and images with 7% added noise are shown in Table 5-3 and Table 5-4. The rows of Table 5-3 and Table 5-4 includes the changes in the network, while columns denote the comparison of evaluation metrics. Starting from U-net (Figure 5.6 (a)) to the final proposed method (Figure 5.6 (d)), the improvement in the evaluation metrics can be clearly observed. With every change in the U-net structure, the metrics tend to improve, which indicates the significance of every change made to the structure. The statistical paired t-test [143] was performed to validate the statistical significance of the proposed method. The null hypothesis was rejected as the value of p was less than 0.05 (with $\alpha = 0.05$), and a confidence level of 95% was achieved [144] [145].

5.4 Discussion:

Real-time cardiac MR images are contaminated with spurious and unwanted Rician noise. These signals create issues for the segmentation networks in the segmentation of the boundary of the objects [146] [147]. The proposed network eliminates the issue of noise in the segmentation process by the incorporation of noise stifler block [24]. This block is inserted between the encoder and decoder on the path, which transfers the feature and indices information. The noise stifler block (NSB) shown in figure 5.2 performs pixel-wise multiplication on the depth wise convolution outputs for the reduction of noise. In NSB, depth wise separable convolution reduces the number of extra multiplication and produces strongly correlated features, which are group normalized and then the pixel-wise multiplication is performed with the original feature maps [114].

The comparative analysis of Figure 5.3 and figure 5.4 shows the proposed network's performance effectiveness. When the network was tested on a different dataset, the Jaccard and dice score showed a good improvement of above 5% and 3.5%, respectively for noise-free images. For 7% noise, the improvement of above 6% and 4.8% was observed in both the metrics, respectively. This performance is achieved because of the incorporation of noise stifler block, which is missing in the conventional counterpart of the proposed network. The depth wise separable convolution produces enhanced spatial information as compared to the standard convolution because of reduced convolution cost. The use of group normalization instead of batch normalization gives stable accuracy as the dependence on batch size is eliminated. This combination produces robust feature maps with reliable information contents. These maps are further applied to NS blocks for suppressing the noise. In the delineation of boundary and contour details, the noise is the main disturbing element that is suppressed using the NS blocks. The noise suppressed features, when received by the up sampling layers, produces the results with accurate boundary and contour details. The use of SELU activation instead of normal

69

RELU gives the added advantage to saving the network from the problem of dying RELU, which in turn eliminates the issue of vanishing gradient [125]. The self and internal normalization property of the SELU produces reliable normalized data, which produces more accurate results in comparison to the conventional counterparts. The prominent figure of merit for the proposed network is the ability to produce exceedingly good results without retraining the network. The increment in the Jaccard indicates that the amount of overlapping area is more between the segmented image and ground truth image for the proposed method as compared to the other methods in the study. The increased dice score signifies that the boundary pixels are segmented perfectly. As the noise level is increased, the proposed method produces more clinically pertinent images as compared to the other methods in the study. This shows the better segmentation capability of the proposed network at various noise levels. The statistical analysis [148] shows a significant difference in the results obtained. The proposed method can be applied to the medical images, even corrupted by noise, for efficient segmentation.

5.6 Conclusion:

The network presented in this chapter not only accurately segmented the standard images but also effectively segmented the noisy images. Thus it can be seen as the comprehensive capability of the network as the images acquired using any method contains noise that is unwanted. The region of interest in medical imaging is most important for planning the diagnosis of the disease. So, for medical analysis, the segments of the heart are clearly delineated by the proposed network, which is having nearly the same contour shape and boundary details. The network trained on one dataset produces accurate segmentation results for other datasets. This proves the ambient capability of the network, which is helpful in saving time and resources. The proposed automatic approach can help in reducing the time and burden of the medical system by producing accurate results in less time.