Chapter 6

Far Wall Plaque Segmentation and Area Measurement in Common and Internal Carotid Artery Ultrasound using U-Series Architectures: An Unseen Artificial Intelligence Paradigm for Stroke Risk Assessment

Summary

Stroke risk assessment using deep learning (DL) requires automated, accurate, and real-time risk assessment while ensuring compact model size. Previous DL paradigms suffered from challenges like memory size, low speed, and complex in nature lacking multi-ethnic, and multi-institution databases. This research segments and measures the area of the plaque far wall of the common carotid (CCA) and internal carotid arteries (ICA) in B-mode ultrasound using four types of solo, namely, UNet, UNet+, UNet++, and UNet+++, and three types of hybrids, namely, Inception-UNet, Fractal-UNet, and Squeeze-UNet, architectures. These seven models are benchmarked against autoencoder-based solution. Three kinds of databases, namely, CCA, ICA, and combined CCA+ICA were implemented using K5 cross-validation protocol. This was validated using unseen Hong Kong data. The CCA database consisted of 379 Japanese images from low-to medium-risk, while the ICA database consisted of 970 Japanese images taken from 97 medium- to high-risk patients.

Using the coefficient of correlation (CC) metric between automated measured area and manually delineated area, seven deep learning solo and hybrid models for CCA yielded 0.96, 0.96, 0.98, 0.95, 0.96, and 0.96 respectively, whereas ICA yielded 0.99, 0.99, 0.98, 0.99, 0.98, 0.98, and 0.98 respectively. Area under the receiver operating characteristics curve values for CCA images was 0.97, 0.969, 0.974, 0.969, 0.962, 0.969, and 0.960 respectively, whereas for ICA images were 0.99, 0.989, 0.988, 0.989, 0.986, 0.989, and 0.988, respectively ($p<0.001$). The percentage improvement in offline memory size, training time and training parameters for Squeeze-UNet compared to UNet++ were 569%, 122.46%, and 569%, respectively.

6.1 Introduction

Non-communicable diseases such as heart attack, cancer, and stroke are the main causes of death worldwide [1]. Among these diseases, stroke is one of the major causes of mortality and morbidity in developing countries like India [244]. Further, 70% of stroke cases occur in developing countries compared to developed countries [1]. The major risk factors for stroke include hypertension, hypercholesterolemia, diabetes, smoking, alcohol, obesity and physical inactivity [245]. Atherosclerosis is the common cause of plaque development in the walls of the arteries and, subsequently, its rupture and blockage of vessels [8]. Over time, this leads to thrombosis and eventually stroke. Several studies have shown an increase in carotid artery atherosclerosis with an increase in coronary SYNTAX score [246]; hence carotid artery disease signifies a surrogate marker

for coronary artery disease. Therefore, low-cost early screening and diagnosis of stroke are vital for saving lives [188]. Even after carotid endarterectomy, patients need careful monitoring over the subsequent years [170].

Ultrasound (US) is the preferred medical imaging modality [128] over MRI [24] and CT scan due to its non-invasive, economical, portable, and non-radiative nature. Modern ultrasound imaging provides high-resolution imaging (0.0625 mm/pixel) and enhanced noise cancellation during beam formation [59]. Thus, US offers two primary applications (i) segmentation of the plaque wall region and (ii) tissue characterization or classification of the segmented plaque region. Segmentation is the field of image processing where the object of interest needs to be isolated or detected and quantified [43]. Several methods have been applied for vascular plaque segmentation [216]. The far wall of the carotid artery has been shown to reflect both high brightness and clinically sound plaque capture [247]. The plaque segmentation and its measurement in the carotid artery are typically done manually or semi-automatically, which leads to variability. Thus, unbiased automated methods play a vital role in stroke/cardiovascular disease management.

Artificial Intelligence (AI) has recently started to penetrate in US, especially in non-cardiac vascular diseases [248]. Not many methods exist in machine learning (ML)-based on segmentation; however, several techniques have been tried using ML-based tissue characterization (TC) using classification paradigms. This includes cancer detection and classification based on US such as liver [131, 132, 165], prostate [249], thyroid [136], plaque tissue characterization (PTC) for coronary plaque [137, 162] and carotid plaque [121, 134, 139]. Recently, studies have been proposed that deploy machine learning (ML)-based PTC [36, 37, 43, 83, 98, 107, 134, 139, 140, 143] for risk stratification; however, they are ad hoc in nature for feature extraction and optimization. It has been recently found that deep learning (DL) or traditionally called solo DL (SDL), has started to penetrate medical imaging [97, 250], particularly US imaging, and more recently in the field of atherosclerosis imaging for CVD/stroke risk assessment [98]. The power of DL has been enhanced by adding two solo DL models, so-called hybrid deep learning (HDL) [97, 118, 250, 251]. Since HDL seems to have shown a more effective solution than SDL, we hypothesize that HDL is more effective for carotid plaque wall segmentation. Previously, UNet has shown robust DL infrastructure; thus, one can use SDL-based UNet architecture to design the HDL-based UNet, which can lead to superior performance. Further, since the configuration of convolution layers can be arranged, one can use the parallelization concept to improve the performance of the HDL designs.

Previous studies proposed different segmentation techniques using solo UNet models which were not suitable for web-based applications, android-based applications, and faster operations in a standalone desktop-based system. Therefore, in this study, we used novel concept of hybrid models which overcomes these gaps. This study proposes three novel hybrid deep learning segmentation

designs using parallel convolutions that are automated, fast, have less training parameters, and require small memory sizes. These HDL models, namely, Inception-UNet, Fractal-UNet, and Squeeze-UNet, are primarily designed to segment the far wall carotid plaque. All HDL models are compared against four SDL, namely, UNet, UNet+, UNet++, UNet3P (or UNet+++) models. The Inception-UNet model deploys various size filters and a parallel architecture to reduce the speed and enhance feature engineering. The Squeeze-UNet model is built upon a sequence of fire [252] modules which drastically reduces the model size. The Fractal-UNet architecture is also designed along with its complete mathematical expression. Thus, these seven models are our unique contributions and first time being used in the plaque measurement paradigm. Figure 6.1 shows the global system diagram of the stroke risk assessment system using SDL/HDL framework. Section 6.2 describes the demographics and baseline characteristics of the two cohorts. An architectural overview of SDL and HDL models is presented in section 6.3.

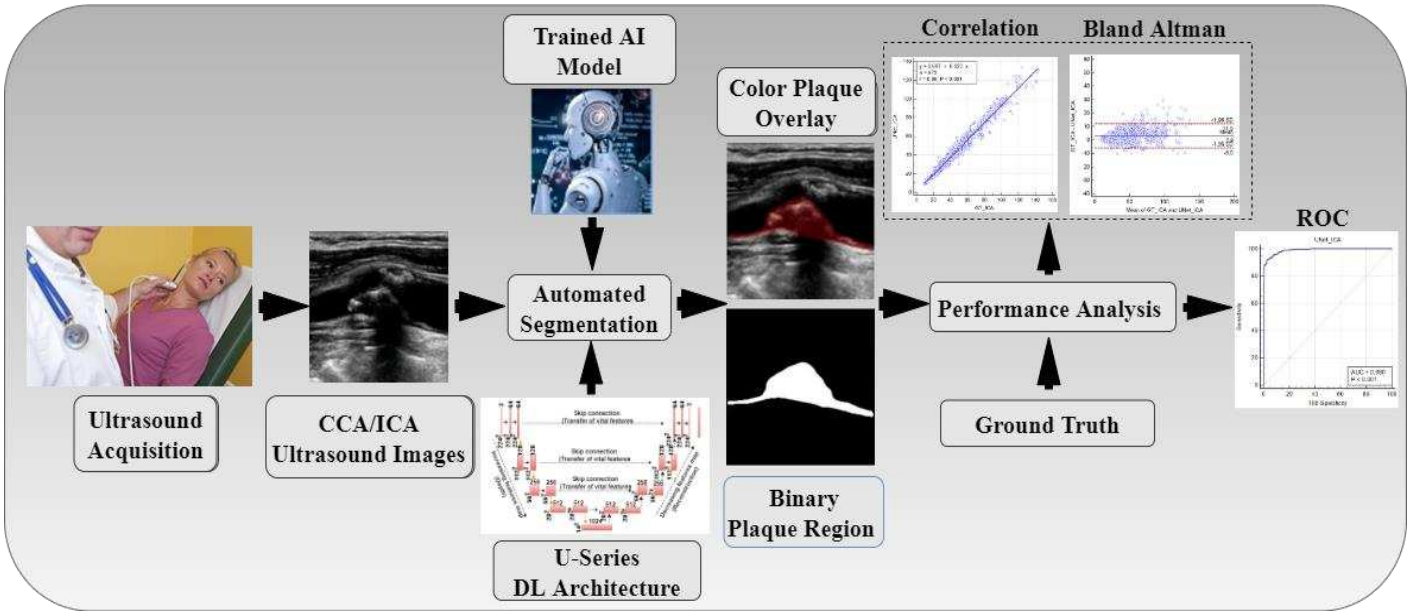

Figure 6.1 Global system for stroke risk assessment. The central engine is the automated segmentation using U-series architecture.

Further, section 6.4 shows experimental results on ICA and CCA. We conducted a detailed performance evaluation on all eight networks described in section 6.5. Finally, critical discussions and benchmarking are presented in section 6, followed by conclusions in section 6.7.

6.2 Demographics and data preparation

In this section we discus three kinds of databases we considered for our experiments. The databases include CCA, ICA ultrasound images. Further, we discuss an unseen database which we used to test our system.

6.2.1 Common carotid artery dataset collection

For patient demographics and baseline characteristics of Japanese database please refer to the section 4.2.1 from previous chapter.

6.2.2 Internal carotid artery dataset collection

For patient demographics and baseline characteristics of ICA database please refer to the section 3.2.1 from previous chapter.

6.2.3 Unseen Hong Kong Database

For patient demographics and baseline characteristics of Hong Kong database please refer to the section 5.2.1.2 from previous chapter.

6.2.4 Ground truth data preparation

For ground truth data preparation (for all databases) please refer to the section 5.2.3 from previous chapter. Figure 6.2 below shows two moderate-plaque and low-plaque CCA and two-moderate and high-plaque ICA images and their corresponding binary masks.

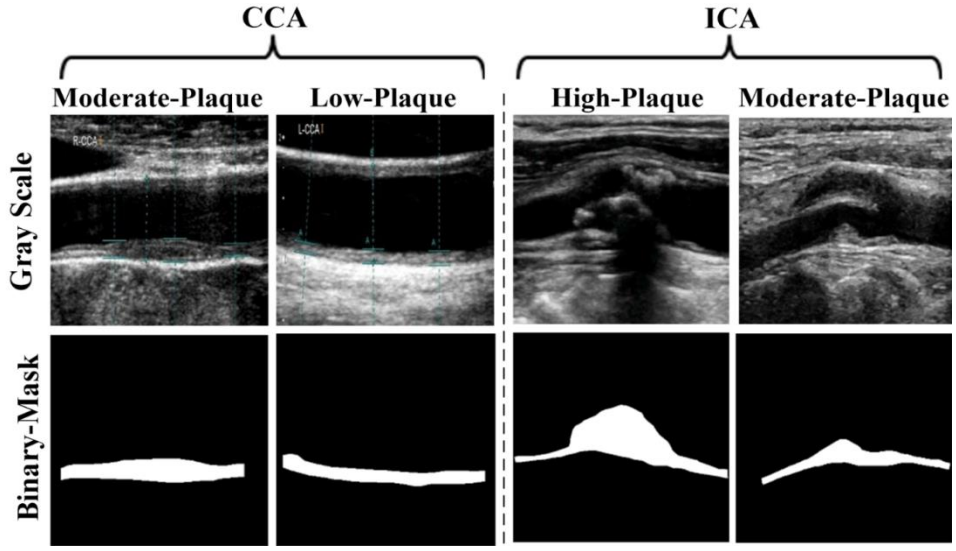

Figure 6.2 Column 1 and column 2 are CCA scans while column 3 and column 4 are ICA scans. Row 1 is raw scans and row 2 is binary (mask) images used as ground truth.

6.3 Methodology

The methodology consists of an architectural overview of the SDL and HDL models. In this section, we have described the layer-wise architectures of three SDL and three HDL models. The size of feature maps at the output of each convolutional layer is calculated using the following equation 6.1

$$S_{FM} = [(W-K+2P)/S +1] \qquad (6.1)$$

Where W is input shape, K is filter size, P is padding, and S is stride. Using any input shape, filter size 3, same padding (=1), and default stride of 1, the output image size after convolution operation remains the same as the input size. We used maxpooling operation with padding=0, filter size (K)=2, stride (S)=2, so the output shape turns into half of the input shape. The second phase or decoder phase (right side of the U shape) starts with up-convolutional (UpConv) layer, which doubles the input shape in each stage (increasing size). UpConv layer is followed by the convolutional layer, which retains the input shape but reduces the depth as the number of filters is half (decreasing feature map). Skip connections play a vital role in feature engineering. These vital features are added depth-wise to the corresponding decoder stage via a skip connection. A feature map is represented by height and width (in two dimensions) and depth (in the third dimension). Depth-wise concatenation of two feature maps is only possible when the remaining shape (height and width) is same.

6.3.1. UNet Architecture

The fundamental principle of UNet is the process of feature extraction using "convolution and maxpooling" during down-streaming (encoding process) and plaque segmentation using "skip connection and up-convolution" during up-streaming (decoding process). Such a combination turns into a U-shape network. Figure 6.3 exactly shows the block diagram of the basic UNet model. It is a four-stage encoder-decoder-based model for carotid plaque segmentation. Each encoder stage consists of two consecutive convolution layers (pink) with a ReLU activation function followed by a maxpooling layer (blue dotted). In each encoder stage, we see the feature map depth S_{FM} is doubled due to the number of convolutional filter depths. Also, the output image size is reduced to half after the maxpooling operation, using the following formula.

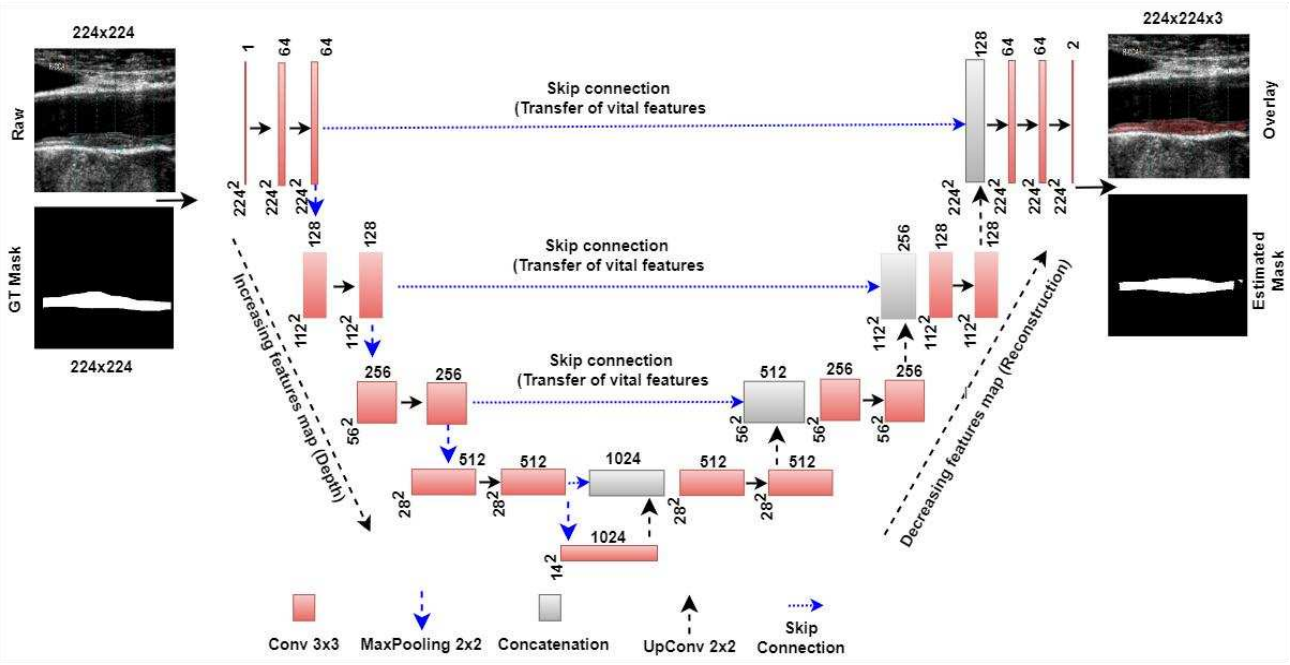

Figure 6.3 UNet architecture.

6.3.2 UNet+ and UNet++ Architectures

Figure 6.4 and 6.5 below shows the UNet+ and UNet++ architectures [98].

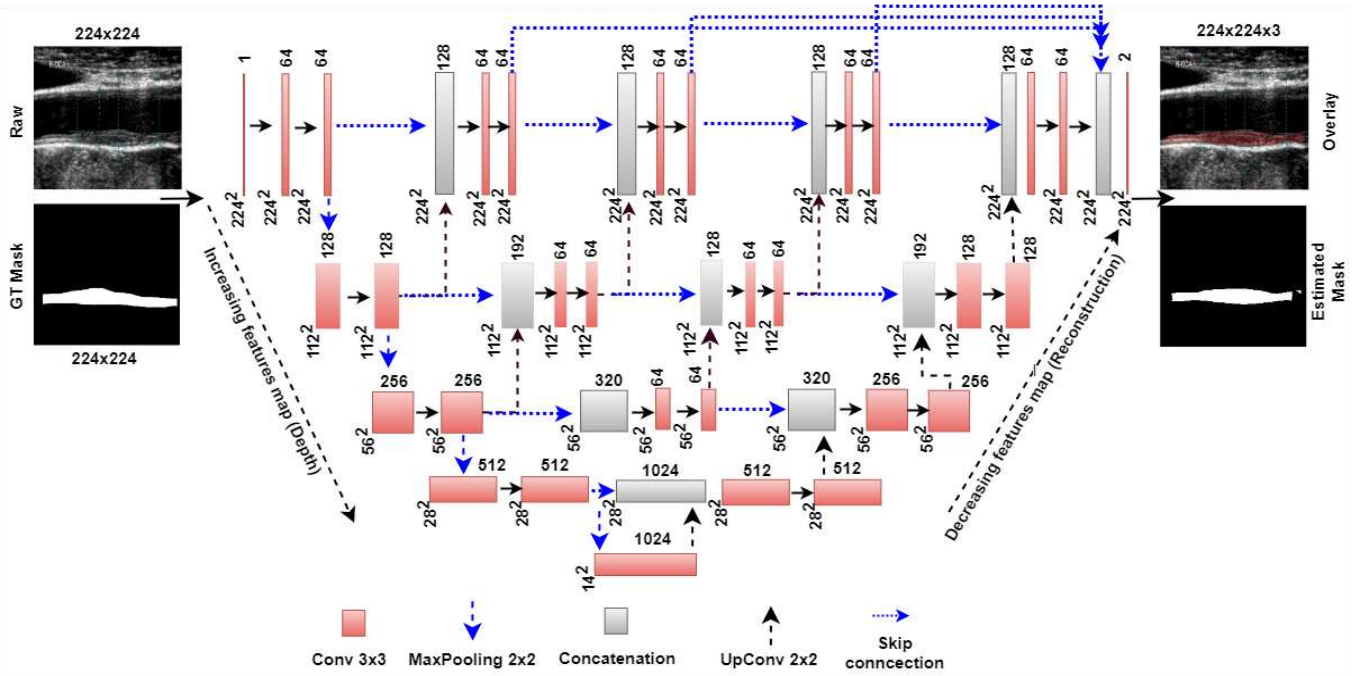

Figure 6.4 UNet+ architecture.

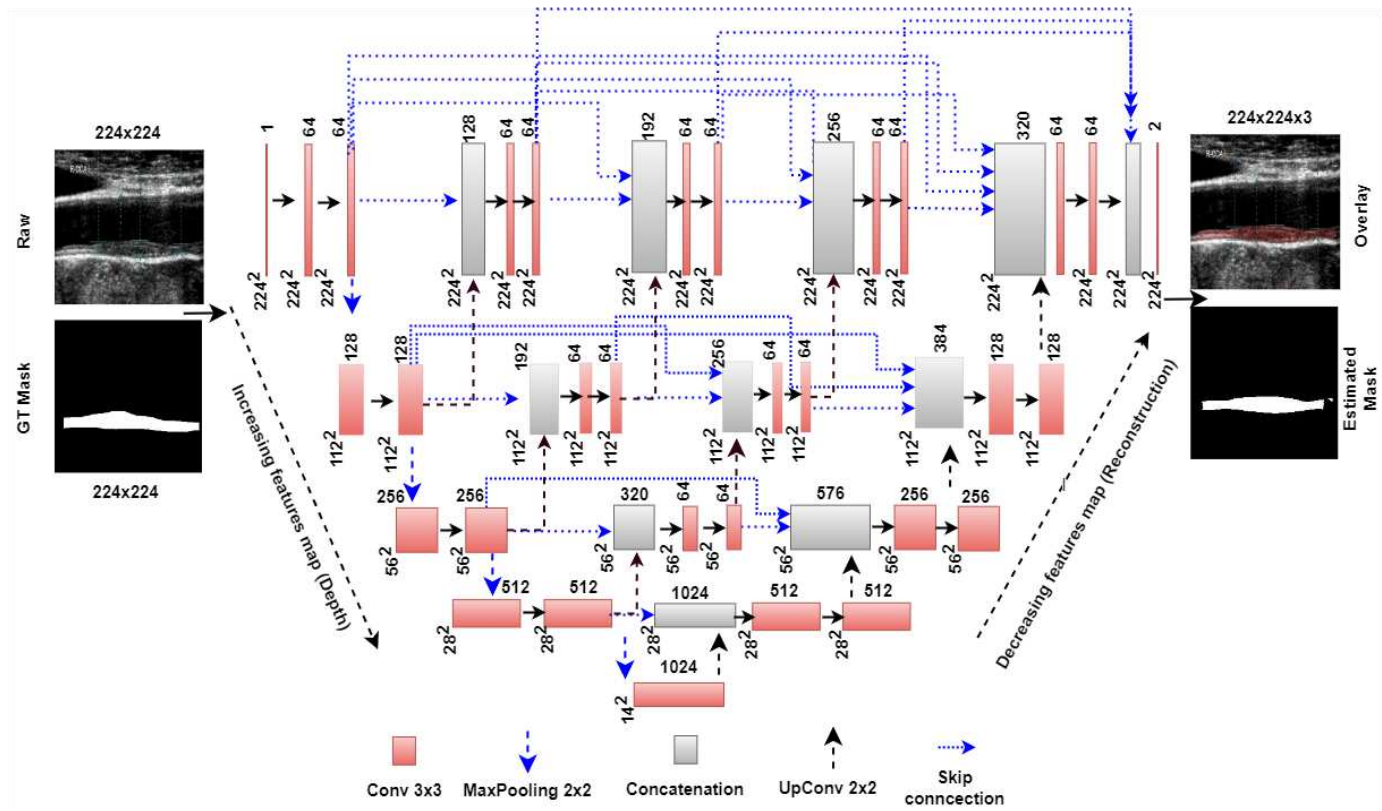

Figure 6.5 UNet++ Architecture.

are routed through a “dense skip network (DSN)”. DSN consists of the UpConv layer followed by concatenation and two convolutional layers. The UpConv layer drives the output of the next encoder Both networks are improved versions of UNet architecture. In both architectures, connections between encoder and decoder stages goes into the concatenation layer, where it is combined with the output of the same encoder level. UNet+ and UNet++ both have the same number of DSN at each level of encoder/decoder stages. Note that, in UNet+, each DSN is connected to only its preceding skip network output, as shown in Figure 6.4, while in the case of UNet++ architecture, each DSN is connected to all previous DSN at the same stage via skip network outputs, as shown in Figure 6.5.

6.3.3. UNet3P: Multiscale architecture

UNet3P network is another modification of UNet network. Figure 6.6 shows the block diagram representation of the UNet3P architecture. This model introduces a new method of full-scale skip connection, which enhances the use of multi-scale features. Using these full-scale skip connections of low-level details of region of interest are merged with high-level definitions of feature maps from multi-scale features. All three solo networks UNet, UNet+, and UNet++ suffer from lacking multi-scale feature connections. Thus, UNet3P takes advantage of multi-scale features by adding the same scale and lower-scale features from the encoder side and high-scale features from the decoder side. As seen from the UNet3P block diagram, Decoder stage 1 concatenates the features map from encoder stage 1 (same scale), decoder stage 2, 3, 4 and the bridge network (large-scale). Decoder stage 2 concatenates the features map from encoder stage 1 (lower scale), encoder-stage 2 (same scale), decoder stage 3, 4, and the bridge (large scale).

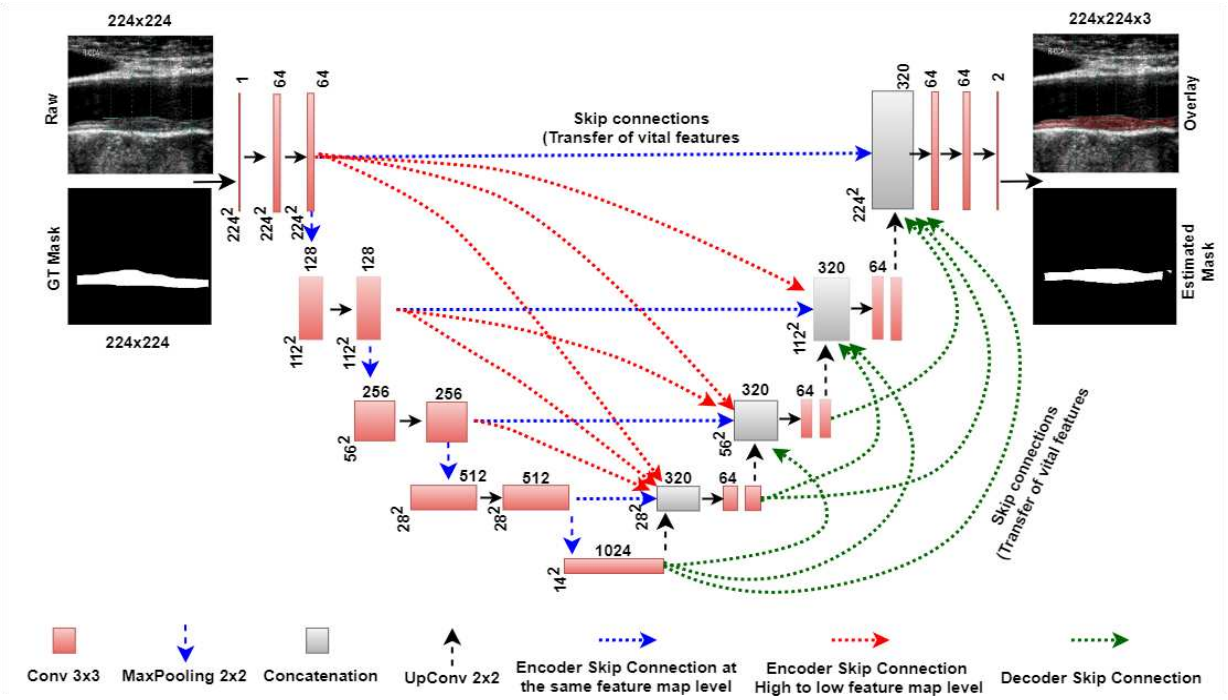

Figure 6.6 UNet3P architecture.

Decoder stage 3 concatenates the features from encoder stages 1 and 2 (lower scale), encoder stage 3 (same scale), decoder stage 4 and bridge (large scale). Finally, decoder stage 4 concatenates the features from encoder stages 1, 2, and 3 (lower scale), encoder stage 4 (same scale), and the bridge.

6.3.4 Autoencoder

An autoencoder is a deep neural network designed for unsupervised learning. However, here we used the same network with a binary mask for the supervised learning application. Autoencoder has three modules an encoder, code, and decoder. The encoder module converts higher dimension features of an image into lower dimension representation. The encoder or dimensionality reduction module learns important features from the images and transforms it into the bottleneck module of the network called as code. The bottleneck module contains only precise information about the image and transfers to the decoder module. Further, the decoder module up samples the encoded representation back to its original size. The encoder module has convolutional layers of 64,128, 256, and 512 filter depth. The decoder module is a replica of the encoder with 512, 256, 128 and 64 filter map depth. The bottleneck module has 512 filter depths. Figure 6.7 below shows the block diagram representation of the Autoencoder network.

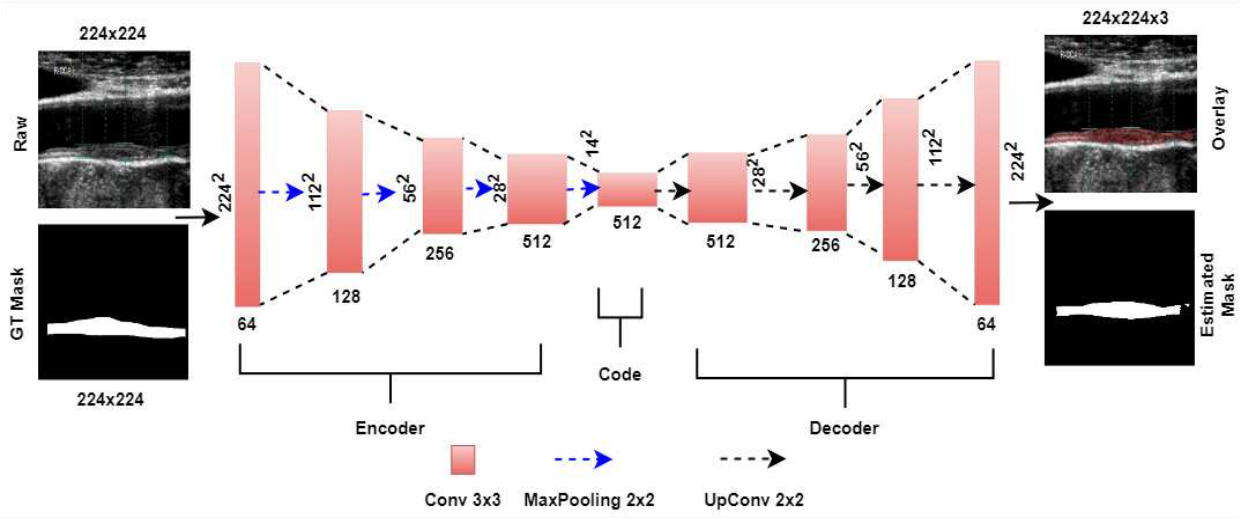

Figure 6.7 Autoencoder architecture.

6.3.5. Hybrid deep learning models

The SDL models have inherent limitations in speed and memory size. Therefore the hybrid models [96, 97, 110] come into existence which overcomes the problems of sluggishness and high memory requirements. Earlier HDL was developed by fusing two solo models. In the present work, we have presented three different novel hybrid architectures which exploit the uses of parallel convolutions. The parallel placement of convolutional layers boosts the speed of operation. We present the following three new hybrid models: Inception-UNet, Fractal-UNet, and Squeeze-UNet.

6.3.5.1 Hybrid Type 1: Inception-UNet architecture

Recently, inception architecture has been used in the HDL framework [132, 150, 250]. Some have been tried in the classification framework [118, 253]. None of the previous Inception architectures has been tested with solo UNet and, further, not with carotid plaque segmentation. Figure 6.8 shows the Inception-UNet architecture. Each encoder and decoder stage in the basic UNet model is replaced by "Inception blocks" shown in the same figure. Each inception module [254] comprises three 1x1 convolutional filters used to reduce the dimensionality of the network.

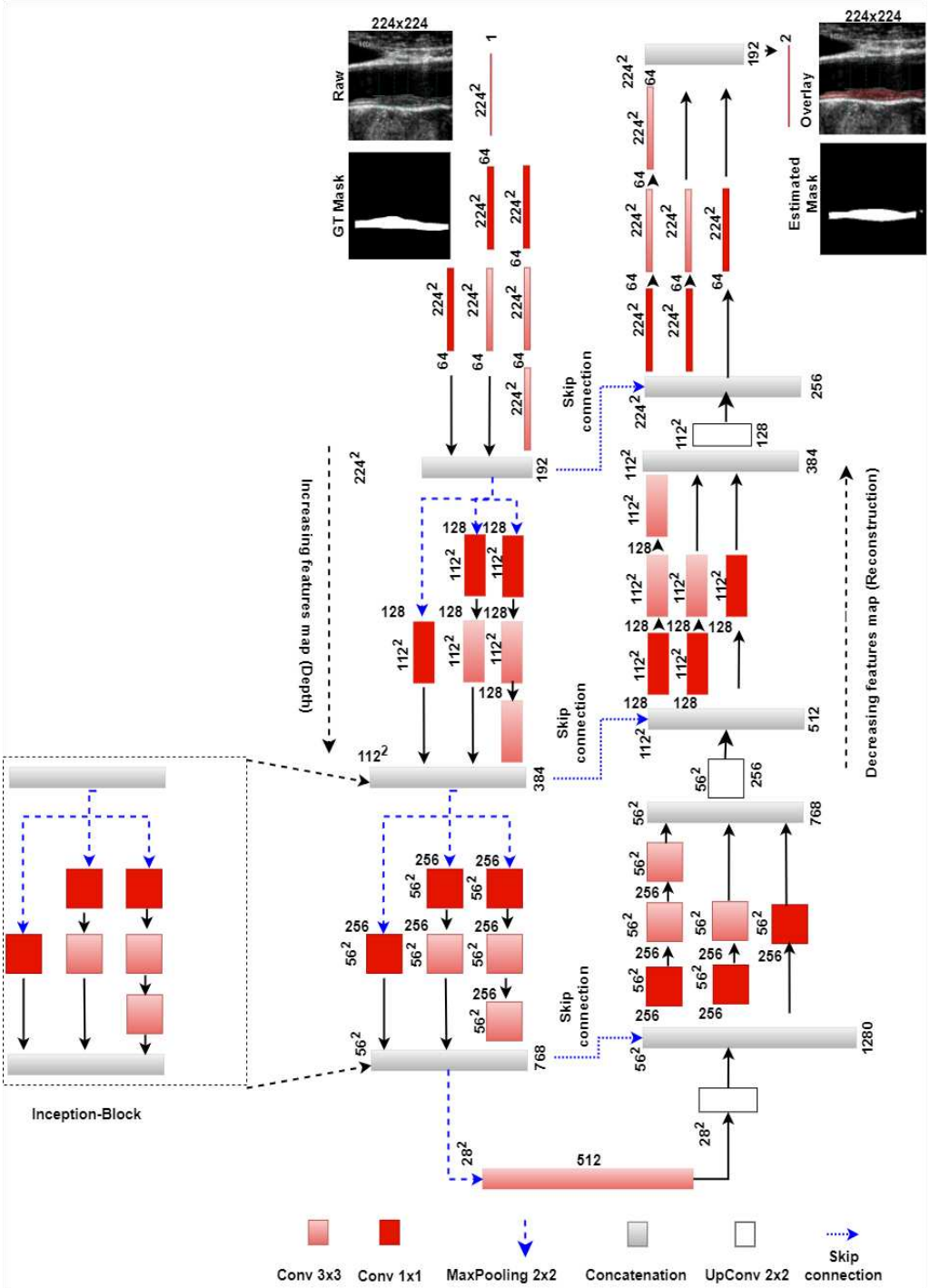

Figure 6.8 Inception-UNet HDL architecture showing encoder-decoder phases.

Two 1x1 convolutional layers feed two other 3x3 convolutional filters to expand the dimension. Thus, Inception blocks allow us to use multiple filter sizes instead of choosing fixed filter sizes. It has the inherent ability to extract varying features maps by using varying filter sizes.

6.3.5.2 Hybrid Type 2: Squeeze-UNet Architecture

The Squeeze-UNet architecture is shown in Figure 6.9 consists of a combination of Squeeze (S) and Expansion (E) modules [252, 255, 256] in place of encoder and decoder stages.

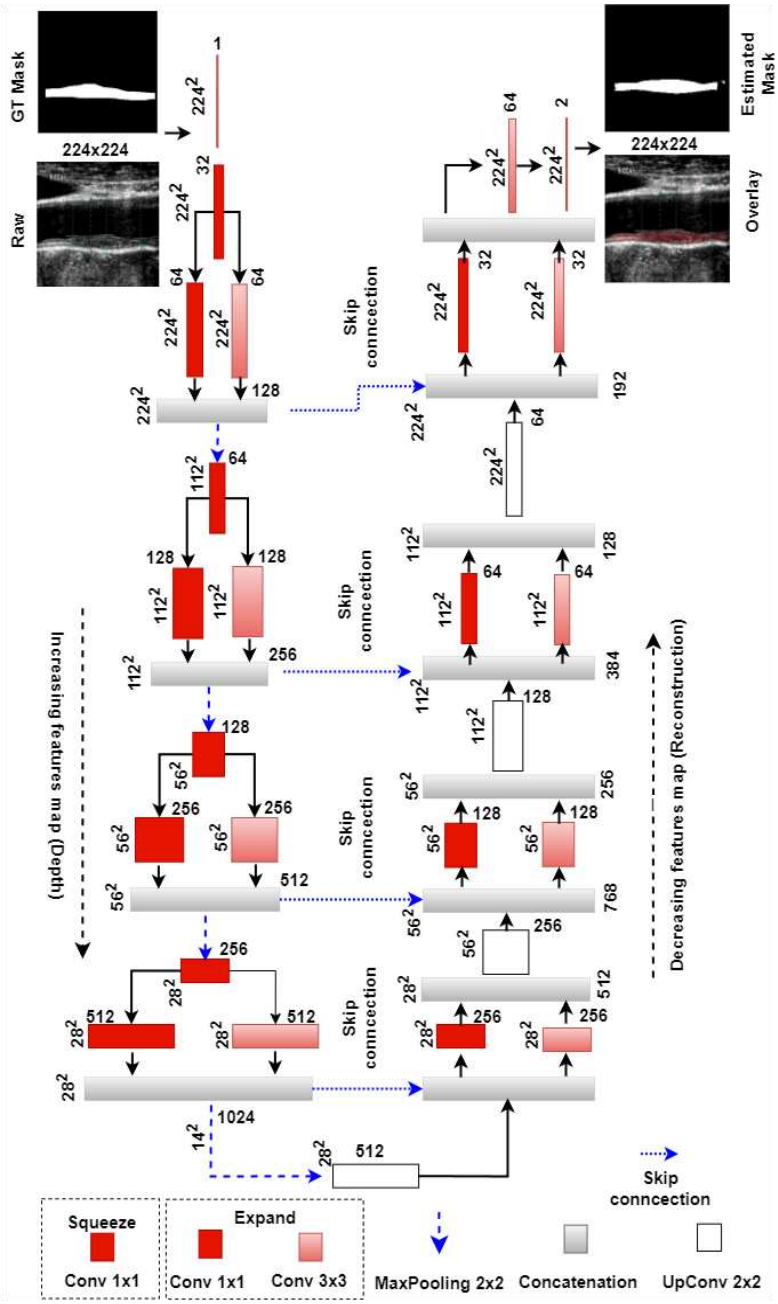

Figure 6.9 Squeeze-UNet HDL architecture with four fire models on the left (encoder) and four fire models on the right (decoder).

The *squeeze* layer is a 1x1 convolution layer, whereas the *expand* layer combines a 1x1 and 3x3 convolution layers in parallel. The *squeeze* layer feeds both *expand* layers. The combination of 'S' and 'E' is called a fire-module. Thus, our Squeeze-UNet hybrid model consists of four fire-modules with varying filter sizes on the encoder and decoder sides. The purpose of the fire-module is to reduce the number of parameters in the activation map, while preserving the accuracy. Output parameters of a convolution layer are calculated by the product of (i) the number of inputs from previous layer, (ii) filters depth (number of filters), and (iii) filter size. We follow two strategies to reduce the number of parameters. First, we choose a 1x1 (filter size) convolution layer for the *squeeze* layer, as the 1x1 convolution filter generates nine times less parameters compared to 3x3 filters. The second strategy is to reduce the number of feature map from previous layer to a 3x3 filter. Filters in the *squeeze* layer are chosen as half in number to that of *expand* layer, to achieve the second strategy. The cumulative effect of both strategies results in a reduced number of output parameters, which is discussed in the results and discussion sections.

6.3.5.3 Hybrid Type 3: Fractal-UNet Architecture

We introduce third hybrid architecture as Fractal-UNet. This architecture can be drawn as a "fractal block" [257] of the convolution defined by Eq. 6.2. It has two constituent parts A_1 and A_2, shown in Figure 6.10. Part A_1 is a cascade of two successive fractal components $C_0(S_{FM})$ and part A_2 is a convolutional layer $conv(S_{FM})$. The cascade operation of two elements in Eq. 6.2 is represented by "$\circ$" (empty circle), and the parallel connection is represented by "$\oplus$" (circle with plus). Part A_1 and A_2 are supplied with the same features map of size S_{FM} from the previous layer followed by maxpooling MP. Finally, both parts are concatenated.

$$C_1(S_{FM}) = \left[[C_0(S_{FM}) \circ C_0(S_{FM})] \circ MP\right] \oplus [conv(S_{FM})] \circ MP \qquad (6.2)$$

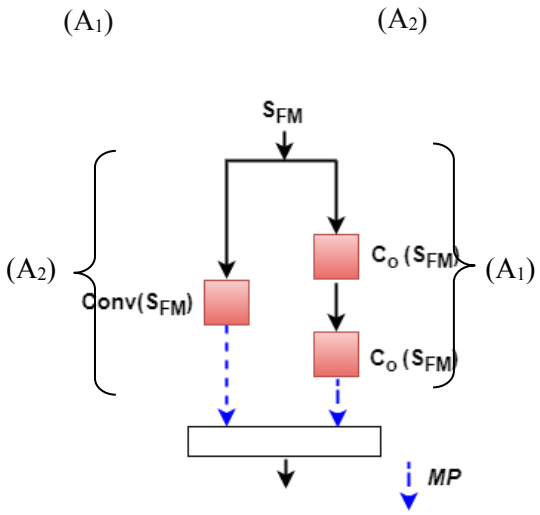

Figure 6.10 Fractal fundamental block used for designing Fractal-Unit architecture.

The complete architecture of the encoder-decoder side of Fractal-UNet is shown in Figure 6.11. The complete encoder architecture $C'_x(S_{FM})$ is defined by Eq. (6.3). In this equation, $C_x(S_{FM})$ represents successive intertwined layers of depth x defined by Eq. (6.4). In consecutive layers of the encoder side, the size of the features map is reduced successively after each "fractal fundamental block" which is controlled by the denominator term 2^{x-1}.

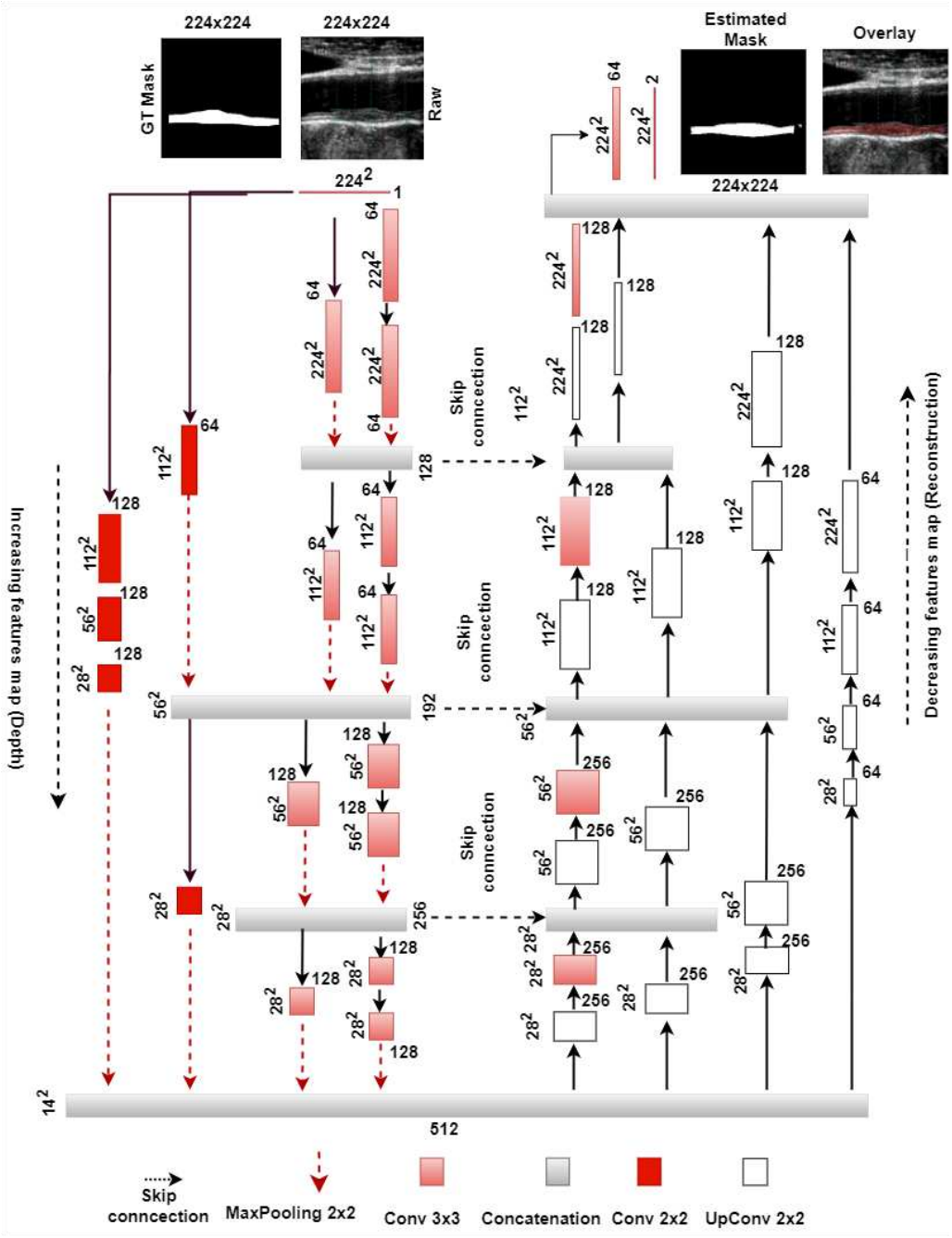

Fig 6.11 Fractal-UNet architecture.

$$C'_x(S_{FM}) = C_x(S_{FM}) \circ \left[conv\left(\frac{S_{FM}}{2^x}\right) \circ MP\right] \tag{6.3}$$

Where *MP* is maxpooling, $conv(S_{FM})$ is convolution operation with S_{FM} as the feature map input.

$$C_x(S_{FM}) = \left[\left[\left[C_{x-1}(S_{FM}) \circ C_{x-1}\left(\frac{S_{FM}}{2^{x-1}}\right)\right] \circ MP\right] \oplus \ldots \left[conv\left(\frac{S_{FM}}{2^0}\right) \circ MP\right] \circ \left[conv\left(\frac{S_{FM}}{2^1}\right) \circ MP\right] \circ \ldots\ldots \left[conv\left(\frac{S_{FM}}{2^{x-1}}\right) \circ MP\right]\right] \quad (6.4)$$

Note, "∘" (empty circle) represents cascade and "⊕" (circle with plus) represents parallel connection.

Equations (6.2), (6.5), (6.6), and (6.7) define four parallel layered encoder architecture of the Fractal-UNet.

$$C'_3(S_{FM}) = C_3(S_{FM}) \oplus [C_2(S_{FM}) \oplus \{C_1(S_{FM}) \oplus (C_0(S_{FM}))\}] \quad (6.5)$$

$$C_3(S_{FM}) = \left[\left[C_2(S_{FM}) \circ C_2\left(\frac{S_{FM}}{2^2}\right)\right] \circ MP\right] \ldots \Big\} (C_0)$$
$$\ldots \oplus \left[conv\left(\frac{S_{FM}}{2^0}\right) \circ MP\right] \ldots \Big\} (C_1)$$
$$\ldots \circ \left[conv\left(\frac{S_{FM}}{2^1}\right) \circ MP\right] \Big\} (C_2) \quad (6.6)$$

Part C_1 and C_2 are cascaded blocks of convolution layers and parallel with Part C_0. Part C_0 i.e. $C_2(S_{FM})$ is an intertwined layer defined by Eq. (6.7) and is given as:

$$C_2(S_{FM}) = \left[\left[C_1(S_{FM}) \circ C_1\left(\frac{S_{FM}}{2^1}\right)\right] \circ MP\right] \oplus \Big\} (B_1)$$
$$\ldots \left[conv\left(\frac{S_{FM}}{2^0}\right) \circ MP\right] \Big\} (B_2) \quad (6.7)$$

Part B_1 is a convolutional block parallel to part B. Part B is an intertwined layer defined by Eq. (6.2). The same equations can also define the decoder architecture of the Fractal-UNet. In each Fractal unit, the UpConv layer and Conv layer represent cascaded blocks parallel with the UpConv layer. Both UpConv layers in the fractal unit have the same filter size and strides (S=2) value which doubles the feature map of the previous layer using equation (6.1). Parallel organization of the Fractal-Units reduces the training time and required offline memory size. A detailed discussion about training time and memory sizes will be discussed in later sections.

6.3.6 Cross-Entropy Loss function

Finally features map are classified using a softmax classifier and loss function defined below is used to minimize the segmentation loss. The CE-loss function is labelled as L_{CE} for an image with N pixels can be mathematically given as eq. (6.8):

$$L_{CE} = -\frac{1}{N}\sum_{i=1}^{N} \underbrace{[(P_i \times \log a_i)}_{A} + \underbrace{(1-P_i)\times\log(1-a_i)]}_{B} \quad (6.8)$$

Above equation consists of two parts A and B. Part A consists of 'P_i' which stands for input GT (Ground truth) label 1 (class=plaque region), and 'ai' represents the model output i.e. SoftMax classifier probability for class 1. Part B in involves (1-P_i) which is for GT (Ground truth) label 0 (class=background or no plaque region), and (1-a_i) term represents model output for class 0. The above equation counts average of all pixels (i=1 to i=N).

6.4 Results

6.4.1 Hyperparameter optimization and performance evaluation

Deep learning model hyper-parameters setup plays a vital role in training loss and weight optimization. We have provided a comprehensive list of hyper-parameters in Table 6.1. All experiments were performed on Lenovo Workstation with Core i7 10th Gen CPU, NVIDIA Quadro P4000 GPU, and 16 GB RAM. However, a low learning rate slows down the training operations, whereas batch size requires a high GPU memory. We applied a learning rate of 10xe-3 and epochs of 100 for each DL model to provide an identical training length. Based on the DL model architectures in Figure 6.3 to Figure 6.11, models' weights are saved offline. These weights have a drastic change in memory requirement for hybrid models compared to solo models. The trainable parameters in each model are 31.03M, 31.61M, 32.71M, 10.72M, 10.36M, 5.73M, 21.49M, and 3.91M for UNet, UNet+, UNet++, Inception-UNet, Fractal-UNet, Squeeze-UNet, UNet3P, and Autoencoder respectively. We divided the complete dataset into 80% training and 20% testing by a well-known K5 partition method. Also, 20% of test data is replaced by a fresh pool of data from the remaining 80% in a cyclic fashion. Thus, the complete training and testing process is repeated five times, and five model weights are saved offline. Each image gets a chance to be part of training and testing at least once in this process. All experiments were repeated for CCA and ICA images with the same setup.

Table 6.1 Hyperparameters And Model Sizes.

Model	#E[1]	LR[2]	LF[3]	Optimizer	BS[4]	TP[5]	MS[6] (MB)	TT[7] (min)	It/E[8]
UNet	100	1e-3	CE	SGDM	8	31,032,840	355	116.22	97
UNet+	100	1e-3	CE	SGDM	8	31,606,402	362	154.81	97
UNet++	100	1e-3	CE	SGDM	6	32,712,322	375	190.63	129
Inception-UNet	100	1e-3	CE	SGDM	4	10,719,176	129	228.5	194
Fractal-UNet	100	1e-3	CE	SGDM	8	10,355,906	54.5	60.25	97
Squeeze-UNet	100	1e-3	CE	SGDM	8	5,730,754	53.1	85.69	97
UNet3P	100	1e-3	CE	SGDM	8	21,488,200	252	94.59	97
Autoencod	100	1e-3	CE	SGDM	8	3,910,784	67.5	26.95	97

[1]E = Epochs; [2]LR= Learning Rate; [3]LF = Loss Function; [4]BS = Batch Size; [5]TP = Total Parameters; MS[6] = Model Sizes; [7]TT=Training Time; [8]It/E=Iteration per epoch.

6.4.2 Visual results of Plaque Segmentation

Figure 6.12 below shows visual results of the estimated plaque area (EPA) in green and GTPA by red colour for CCA and ICA experiments. On top of Figure 6.12 are binary masks of moderate (R1C1)

and low (R1C2) plaque images of CCA and high (R1C3) and moderate (R1C4) plaque images of ICA. In Figure 6.12, the second row represents an overlay of binary masks over raw images. The rest of the images are the estimated plaque area of DL models (UNet, UNet+, UNet++, Inception-UNet, Fractal-UNet, and Squeeze-UNet) and GTPA. In DL output images, red represents the estimated plaque area, whereas green represents an absolute difference between GT and the estimated plaque area.

6.4.3 Segmentation performance

Tables 6.2 and 6.3 represent the segmentation performance of all DL model experiments performed on CCA and ICA images, respectively. We determined some standard parameters such as accuracy, sensitivity, specificity, precision, MCC (Mathew's correlation coefficient), dice-similarity coefficient (DSC), and Jaccard-index (JI). By looking into both tables, we can conclude that the solo and hybrid DL models have very close results in terms of segmentation parameters. All DL models have DSC of 87.29±5.86, 87.06±6.05, 88.37±5.07, 78.88±8.25, 86.00±6.82, and 86.81±6.20 for CCA experiments and 90.02±3.53, 89.72±3.57, 89.32±4.16, 87.86±4.03, 88.78±4.50, and 89.52±4.01 respectively for ICA experiments.

Table 6.2 CCA Plaque Segmentation Results.

Model	**[1]ACC (%)**	**[2]SENS (%)**	**[3]SPEC (%)**	**[4]PREC (%)**	**[5]MCC (%)**	**[6]DICE (%)**	**[7]JI (%)**
UNet	99.05±0.45	83.54±7.47	99.71±0.35	92.33±8.25	87.14±5.66	87.29±5.86	77.88±8.48
UNet+	99.04±0.45	83.20±7.21	99.71±0.38	92.27±8.61	86.93±5.80	87.06±6.05	77.54±8.49
UNet++	99.14±0.36	84.57±7.33	99.75±0.24	93.28±6.34	88.22±4.82	88.37±5.07	79.51±7.44
Inception-UNet	98.60±0.53	68.89±10.29	99.81±0.30	94.12±8.36	79.54±7.42	78.88±8.25	65.83±10.26
Fractal-UNet	98.97±0.51	81.46±8.71	99.72±0.30	92.19±8.46	85.91±6.51	86.00±6.82	76.00±9.36
Squeeze-UNet	99.02±0.48	82.41±8.50	99.74±0.28	92.74±7.51	86.71±5.84	86.81±6.20	77.17±8.69
UNet3P	98.98±0.52	81.71±9.84	99.73±0.29	92.49±7.84	86.13±6.49	86.16±6.99	76.27±9.69
Autoencoder	98.80±0.83	87.79±9.96	99.29±0.63	83.94±12.27	84.86±9.35	85.07±9.76	75.10±12.88

[1]ACC=Accuracy; [2]SENS=Sensitivity; [3]SPEC = Specificity; [4]PREC= Precision; [5]MCC= Mathew's Correlation Coefficient; [6]DICE= Dice Similarity Coefficient; [7]JI= Jaccard Index.

Table 6.3 ICA Plaque Segmentation Results.

Model	[1]ACC (%)	[2]SENS (%)	[3]SPEC (%)	[4]PREC (%)	[5]MCC (%)	[6]DICE (%)	[7]JI (%)
UNet	98.58±0.61	87.43±5.45	99.48±0.40	93.11±4.61	89.39±3.63	90.02±3.53	82.03±5.66
UNet+	98.56±0.57	86.65±5.59	99.50±0.39	93.39±4.55	89.11±3.62	89.72±3.57	81.55±5.71
UNet++	98.51±0.65	85.72±6.56	99.53±0.38	93.70±4.55	88.74±4.14	89.32±4.16	80.94±6.45
Inception-UNet	98.37±0.57	81.83±6.55	99.67±0.31	95.33±4.00	87.39±3.89	87.86±4.03	78.57±6.24
Fractal-UNet	98.41±0.74	85.68±6.71	99.45±0.43	92.61±5.35	88.13±4.58	88.78±4.50	80.10±6.92
Squeeze-UNet	98.53±0.63	86.29±6.01	99.51±0.39	93.40±4.95	88.91±4.09	89.52±4.01	81.25±6.30
UNet+++	98.51±0.66	86.73±6.35	99.47±0.41	92.90±4.84	88.87±3.96	89.49±3.90	81.19±6.16
Autoencoder	96.83±2.30	68.89±21.10	99.25±0.68	87.12±11.15	75.04±16.34	75.14±17.41	62.91±19.79

[1]ACC=Accuracy; [2]SENS=Sensitivity; [3]SPEC = Specificity; [4]PREC= Precision; [5]MCC= Mathew's Correlation Coefficient; [6]DICE= Dice Similarity Coefficient; [7]JI= Jaccard Index

Similarly, JI values for the same models are 77.88±8.48, 77.54±8.49, 79.51±7.44, 65.83±10.26, 76.00±9.36, and 77.17±8.69 for CCA experiments and 82.03±5.66, 81.55±5.71, 80.94±6.45, 78.57±6.24, 80.10±6.92, and 81.25±6.30 for ICA experiments. Other segmentation parameters for hybrid models have close values with solo models.

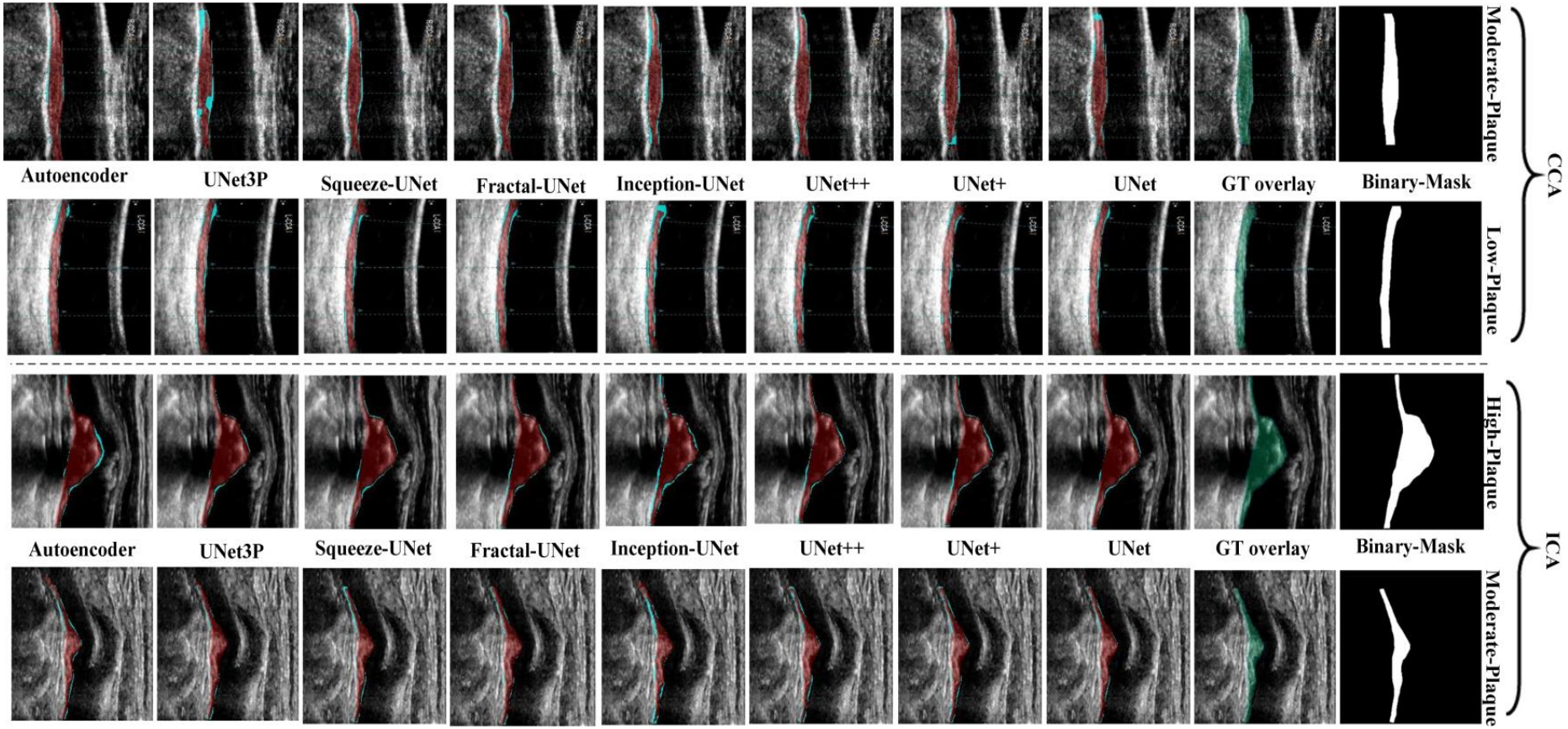

Figure 6.12 Estimated (red) and the absolute difference between estimated and gold standard (green) plaque area in CCA and ICA images. Column 1: Moderate-plaque CCA; Column 2: Low-plaque CCA; Column 3: High-plaque ICA; Column 4: Moderate-plaque ICA. Row1: Binary mask; Row2: Ground truth region; Row 3: UNet; Row 4: UNet+; Row 5: UNet++; Row 6: Inception-UNet; Row 7: Fractal-UNet; Row 8: Squeeze-UNet.

6.5 Performance Evaluations

We evaluate the performance of the eight models using three kinds of criteria. (i) Correlation between the output area with the ground truth area; (ii) evaluation using Bland-Altman plots; (iii) AUC evaluation from the ROC curves along with the p-values.

Table 6.4 Performance Parameters Of CCA And ICA Data Experiment.

Performance Parameters for CCA Data				
Model	**CC**	**AUC**	**FOM**	**Mean ΔPA**
UNet	0.96	0.970	87.11	5.52±5.5
UNet+	0.96	0.969	86.72	5.56±5.28
UNet++	0.98	0.974	88.43	4.89±4.52
Inception-UNet	0.96	0.969	72.62	11.37±7.31
Fractal-UNet	0.96	0.962	85.72	6.19±6.44
Squeeze-UNet	0.96	0.969	86.17	5.97±5.52
UNet3P	0.96	0.960	85.15	6.40±6.62
Autoencoder	0.92	0.952	85.18	5.48±7.37
Performance Parameters for ICA Data				
UNet	0.99	0.99	91.85	3.89±3.80
UNet+	0.99	0.989	91.20	4.06±3.74
UNet++	0.98	0.988	90.14	4.57±4.48
Inception-UNet	0.99	0.989	85.66	6.40±4.54
Fractal-UNet	0.98	0.986	90.53	4.52±4.74
Squeeze-UNet	0.98	0.989	90.83	4.33±4.15
UNet3P	0.98	0.988	90.92	4.28±4.47
Autoencoder	0.76	0.895	76.43	12.62±16.09

6.5.1 Correlation Analysis using Regression Curves

Regression curve analysis is a powerful and vital parameter to find a correlation between two measured quantities. The correlation coefficient takes values between '0' and '1', where values close to '1' represent a very high correlation, and those close to '0' represent a low correlation. We can see in Table 6.4 column 2, the correlation coefficient between the estimated area and GT area are 0.96, 0.96, 0.98, 0.96,0.96, 0.96,0.96, and 0.92 (all have $p<0.0001$) for CCA experiments and 0.99, 0.99, 0.98, 0.99, 0.98, 0.98, 0.98, and 0.76 (all have $p<0.0001$) for ICA experiments, respectively for UNet, UNet+, UNet++, Inception-UNet, Fractal-UNet, Squeeze-UNet, UNet3P, and Autoencoder respectively. All SDL and HDL models have encouraging CC values. Figures 6.13 and 6.14 show the regression curves between GT and estimated plaque area for all U-series models for CCA and ICA experiments, respectively.

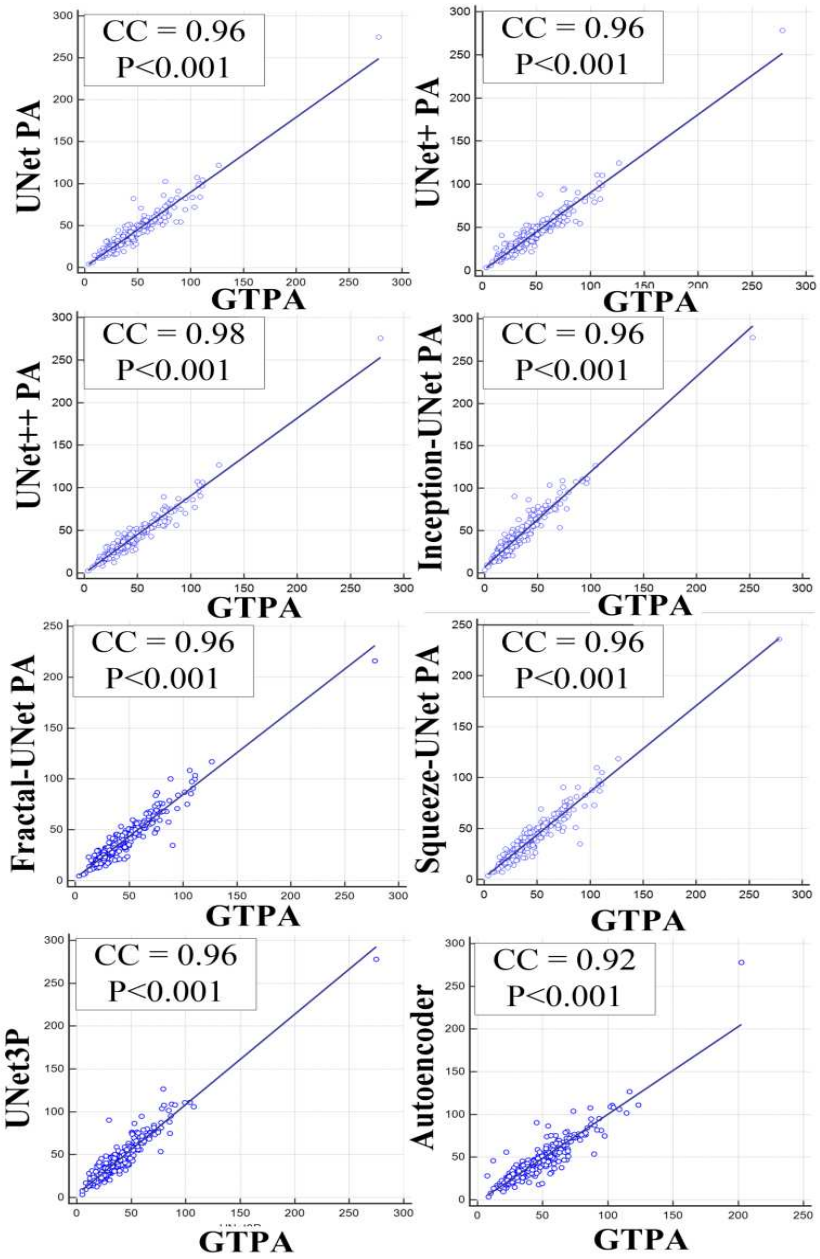

Figure 6.13 CCA Regression curves between estimated PA of models vs. ground truth plaque area.

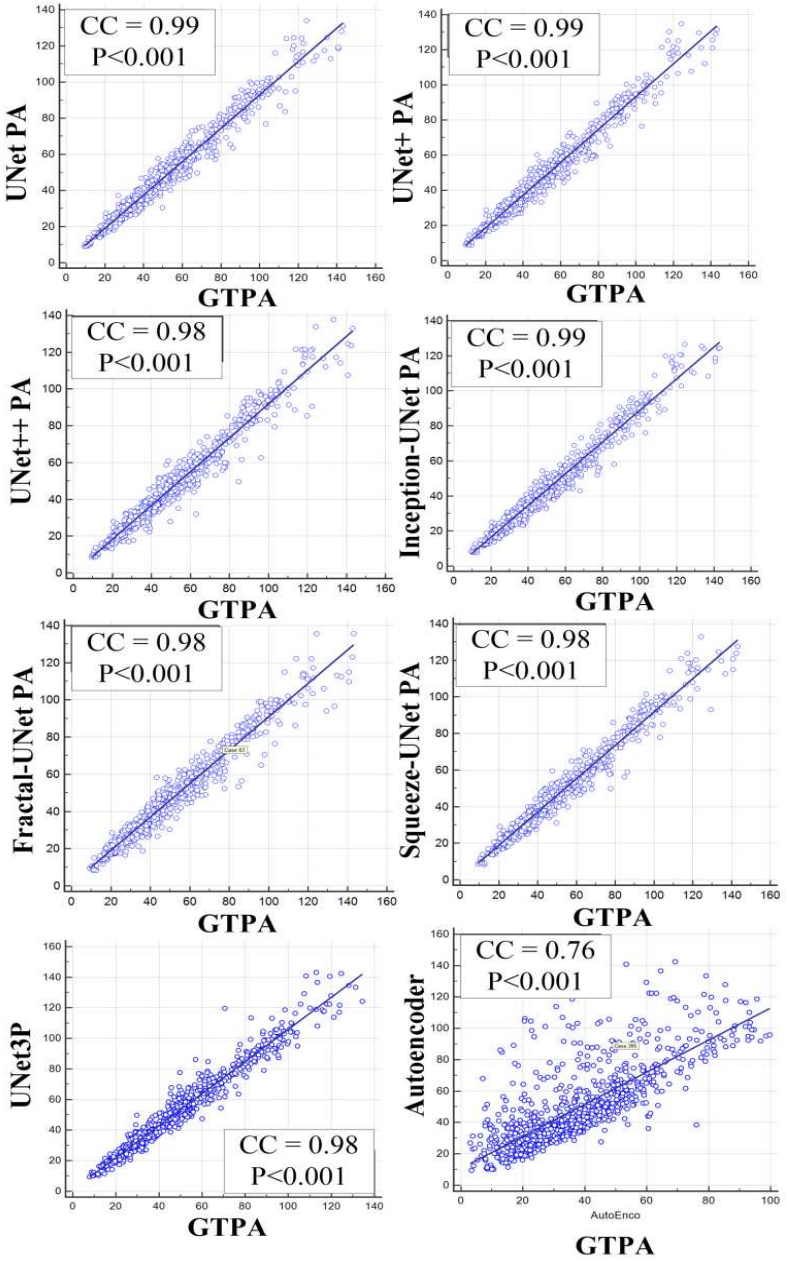

Figure 6.14 ICA Regression curves between estimated PA of models vs. ground truth plaque area.

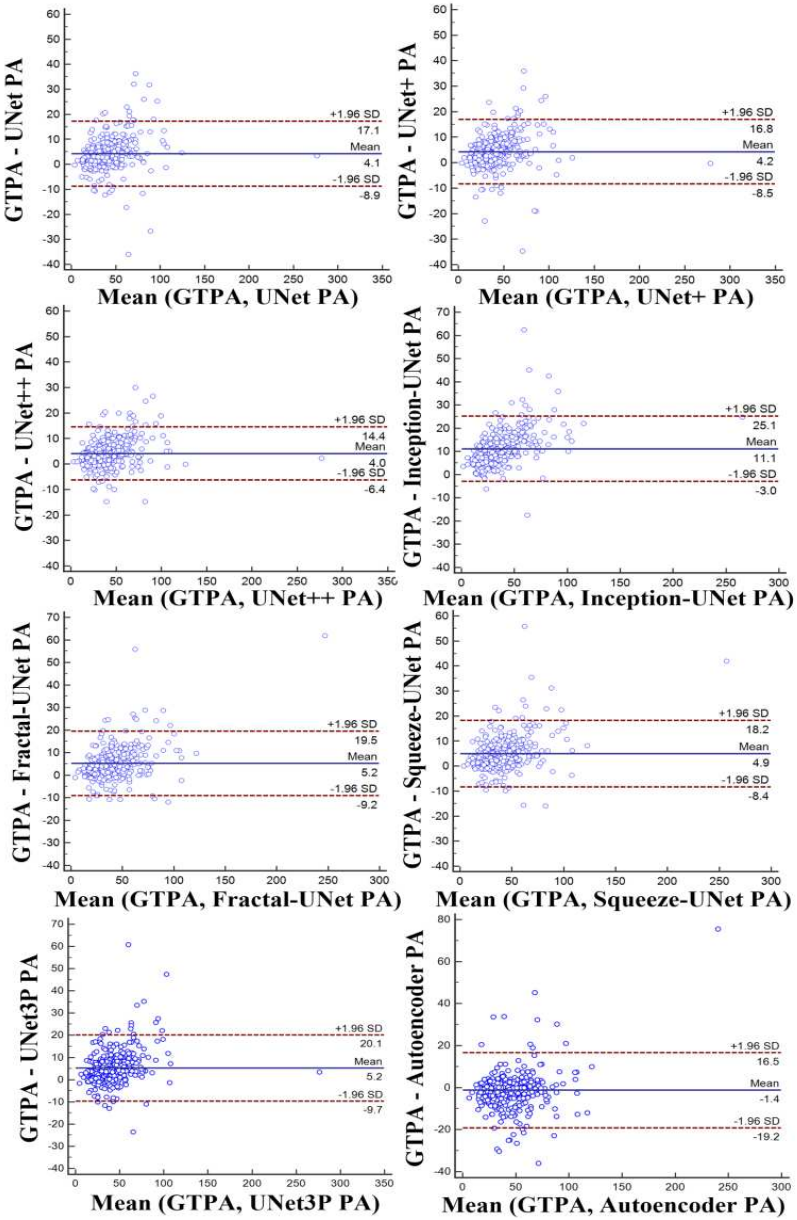

Figure 6.15 Bland Altman's plot between CCA estimated PA of models vs. ground truth plaque area.

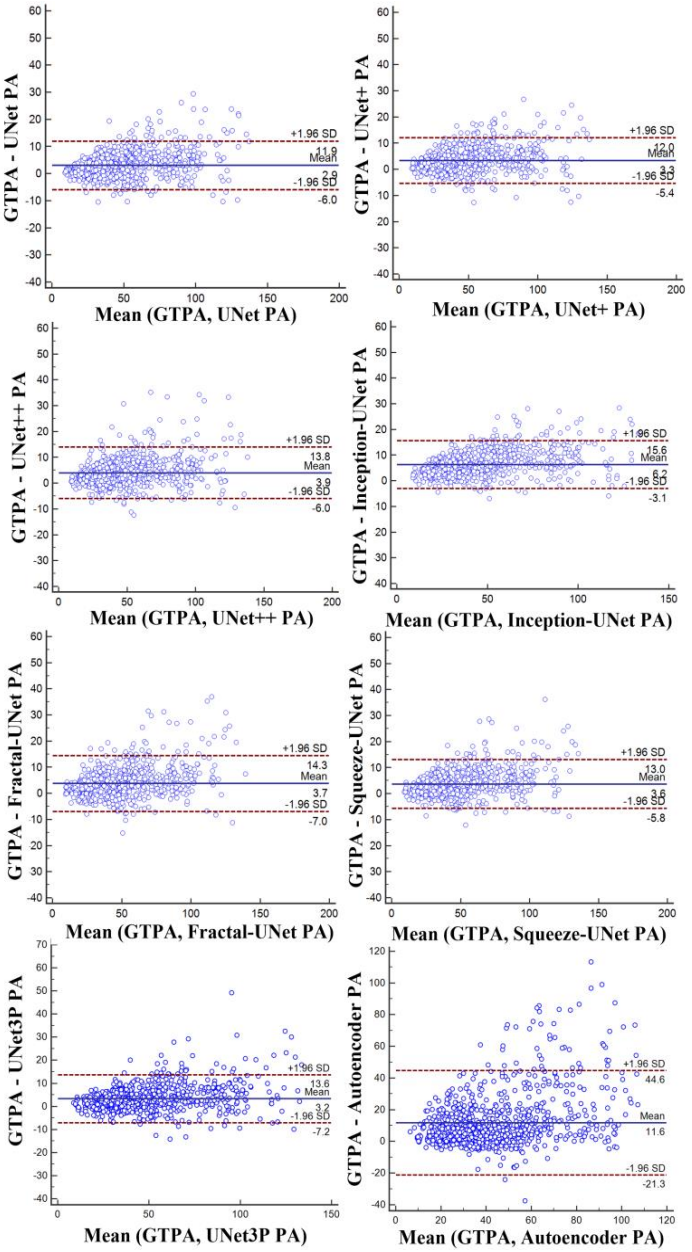

Figure 6.16 Bland Altman's plot between ICA estimated PA of models vs. ground truth plaque area.

6.5.2 Bland-Altman's Plot analysis

In clinical studies, Bland-Altman's (BA) plot analysis is performed when two methods measure a parameter. In our work, estimated plaque areas by all DL models are compared against the GTPA using BA plots. It can be noticed from BA plots of all CCA and ICA data experiments that the significant populations are well within the 95% confidence interval lines (±1.96 SD lines). Figures 6.15 and 6.16 show BA plots of U-series models for CCA and ICA experiments, respectively.

6.5.3 Receiver operating characteristics

The ROC curve plays a vital role in assessing the performance of a segmentation model. In our case, we plotted the ROC curve using the binary values generated by putting a threshold of 40 mm^2 in the ground truth plaque area ('1' for GTPA>40 mm^2, and '0' for GTPA<40 mm^2) [97, 152, 192]. Further, these binary values are used as classification variables against the estimated plaque areas by all DL models. As seen in Table 6.4 column 3, area under the ROC curve (AUC) values for all CCA experiments are 0.97, 0.969, 0.974, 0.969, 0.962, 0.969, 0.960, and 0.952 same for ICA experiments

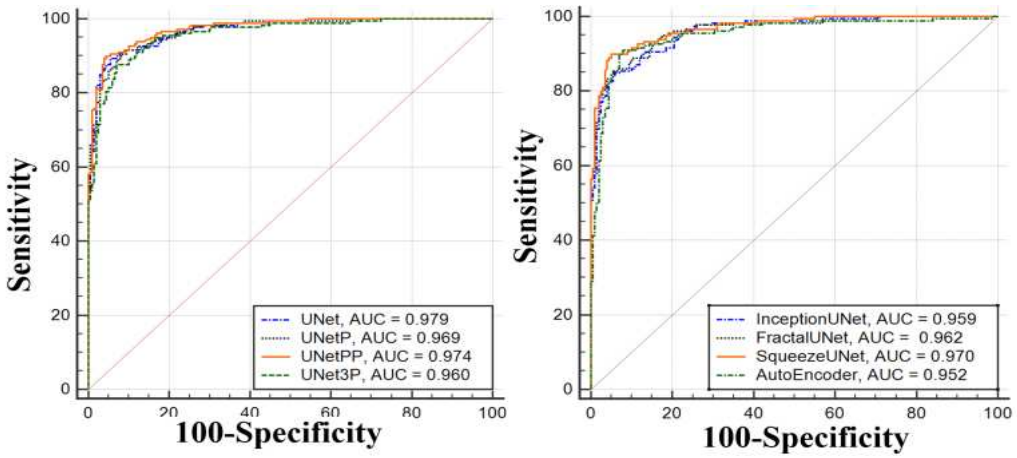

Figure 6.17 AUC for CCA analysis.

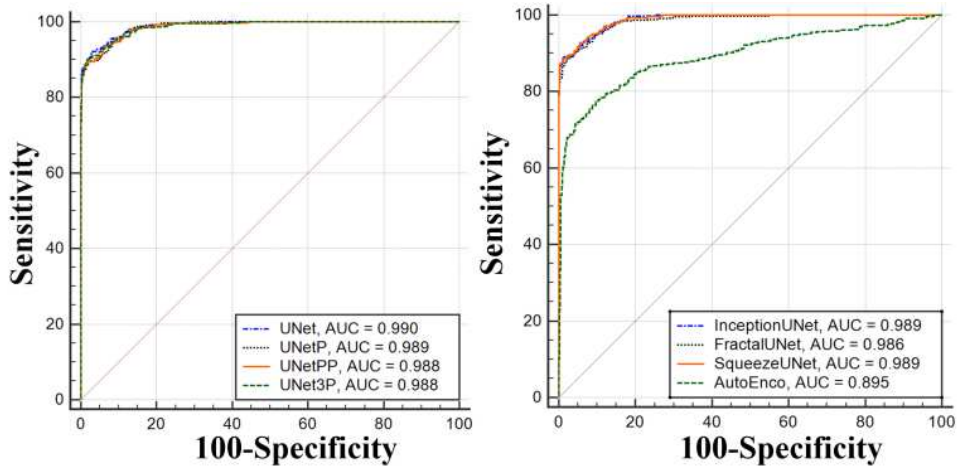

Figure 6.18 AUC for ICA analysis.

are 0.99, 0.989, 0.988, 0.989,0.986, 0.989, 0.988, and 0.895 respectively. Figure 6.17 (a) and (b) show ROC curves for three SDL and three HDL models for CCA experiments, whereas, Figure 6.18 (a) and (b) show the same for ICA experiments.

6.5.4 Figure of Merit

The central tendency of the error is defined as the Figure of Merit (FoM). This concept is also used to assess the performance of the system. Let, $A_{ai}(m,i)$ and $A_{gt}(i)$ represent the plaque area using AI model 'm' and GT, respectively, for the image 'i'. Thus for the total number of scans 'N', $\bar{A}_{ai}$(m) and $\bar{A}_{gt}$ represents corresponding mean area for model m and GT. Then, the FoM for DL model 'm' can be defined as Eq. (6.9):

$$\left| FoM(m) = 100 - \left[\frac{|\bar{A}_{ai}(m) - \bar{A}_{gt}|}{|\bar{A}_{gt}|} \right] \times 100 \right| \qquad (6.9)$$

Where, $\bar{A}_{ai}(m)$ and $\bar{A}_{gt}(m)$ are defined as equations (10) and (11):

$$\bar{A}_{ai}(m) = \frac{\sum_{i=1}^{N} A_{ai}(m,i)}{N} \qquad (6.10)$$

$$\bar{A}_{gt} = \frac{\sum_{i=1}^{N} A_{gt}(i)}{N} \qquad (6.11)$$

Table 6.4 column 4 shows the FoM values for CCA and ICA data experiments.

6.5.5 Model size, training time, and training parameters

We hypothesize earlier that the hybrid models have some inherent advantages by virtue of their parallel connections. Our research found that the parallel connections reduce the training time and the offline model weight size. Table 6.1 shows that training time changes drastically from SDL to HDL. For UNet, UNet+, UNet++, Inception-UNet, Fractal-UNet, Squeeze-UNet, UNet3P, and Autoencoder hold training time (in minutes) of 116.22, 154.81, 190.63, 228.5, 60.25, 85.69, 94.59, and 26.95 respectively. Corresponding training parameters (in Million) of training models are 31.03, 31.61, 32.71, 10.72, 10.36, 5.73, 252, and 67.5 respectively. Further, the memory requirements (in MB) for the same offline models are 355, 362, 375, 129, 54.5, 53.1, 214.89, and 3.91 respectively. A bar chart showing model size (blue), training time (red) and training parameters (green) is shown in Figure 6.19. The y-axis represents the model size (in MB), training time (in minutes) and training parameters (in Million) jointly. Thus a trade-off between memory size, training time and training parameters gives a better perspective for choosing between SDL and HDL.

6.6 Discussion

6.6.1 Principal findings

This is the first study of its kind that presented eight types of UNet architectures, three solo, namely, UNet, UNet+, and UNet++, and three hybrids, namely Inception-UNet, Fractal-UNet, and Squeeze-UNet showing the parallel concept of deep learning networks for carotid plaque segmentation. Two types of carotid scans (CCA and ICA) and three sources of data sets: UK (970 images), Japanese (379 images), and Hong Kong (300 images) were used in seen and unseen frameworks. Using the cross-validation framework of the K5 protocol, the UNet series showed encouraging results and proved to be very powerful for carotid plaque segmentation and area measurements. The mean CC and mean AUC over all eight models were 0.96 and 0.987 ($p<0.0001$), respectively, which are the best in the AI framework industry. Further, the system is applicable in clinical settings.

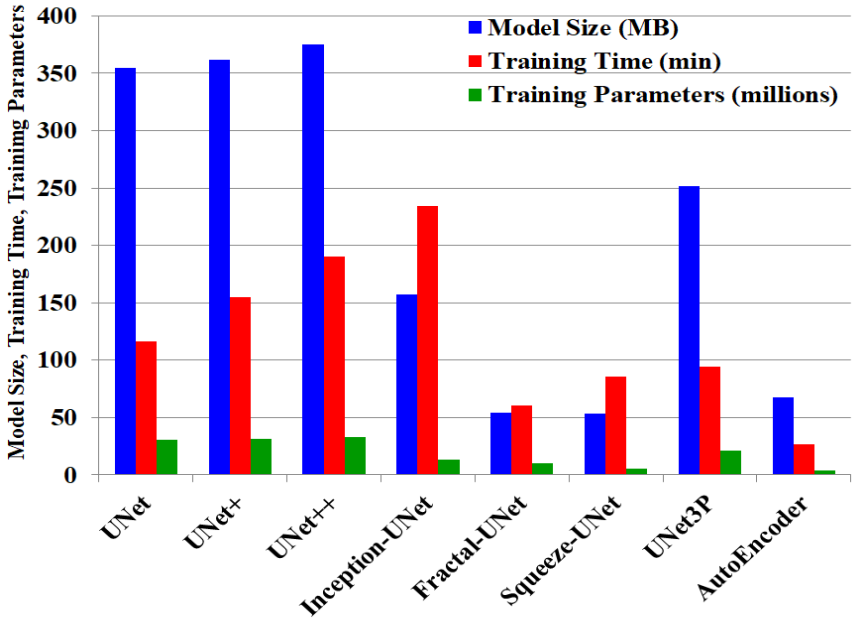

Figure 6.19 Bar chart showing model size, training time, and training parameters.

6.6.2 Short note on HDL and parallel connection

In the present work, we presented the new idea of HDL models and presented three novel architectures using parallel convolutions. The parallel architecture suits the modern graphics processor with a large number of cores [258]. The parallel jobs reduce the training time even after a large number of convolutional layers are deployed. Also, smaller filter sizes reduce the total trainable parameters, resulting in a small modal size. A seven-time reduction in memory requirement can be seen (see Table 6.1) in Fractal-UNet and Squeeze-UNet using the parallel architecture.

6.6.3 Stroke Risk Assessment - Relationship with Quantitative Measurements

The area of the wall plaque has a direct bearing with the stroke risk. Larger wall plaque area in the carotid artery signifies the chances of rupturing the wall due to stress in the walls of the carotid artery. Once the wall ruptures, the wall material such as fibrosis, fibrin, calcium, and macrophages ooze out of the wall region and flow in the direction of blood. These plaque components then reach the brain's small vessels and block the blood flow, causing the brain's cells to malfunction. Over time, these brain cells die, leading to a lack of brain functionality and stroke. Thus, larger the plaque region in the carotid wall, higher is the chances of stroke.

6.6.4 Benchmarking

Table 6.5 below is a benchmarking table that shows similar work related to plaque area segmentation using DL techniques. We have benchmarked our work with previous deep learning-based carotid plaque segmentation works. These works are listed here using first author's name.

Jain *et al.* proposed the concept of fusing two SDLs to form an HDL model [97]. Their work includes SegNet-UNet and SegNet-UNet+ HDL models, which outperformed UNet, UNet+ and SegNet SDL models in various performance parameters. They compared two loss functions, namely cross-entropy loss and dice-similarity coefficient loss functions, for each model to see the performance of the models with each loss function. In all experiments, SDL and HDL models yielded higher performance indices for CE-loss compared to the DSC-loss function-based model. Further, their SDL and HDL models achieved 3.49 mm^2 and 4.21 mm^2 PA errors. However, their models suffer from large memory sizes and sluggish operations. Also, they didn't validate their model with any established model.

Recently Zhou *et al.* presented a research paper using an SDL model UNet++ which includes dense skip connections between the encoder and decoder stage [98]. Their model achieved the best TPA error of 5.55±4.34 mm^2. However, they didn't discuss the memory size and training time issue. The major pros of UNet++ architecture are feature extraction capability in low to moderate plaque images. Also, they used a multi-institution database for their research. A major disadvantage of their system is the smaller test dataset of 33, 34, and 33. We implemented a similar UNet++ model and found that this model suffers from a large memory size and training time. Also, they combined CCA and ICA images to find the plaque deposition. It is to be noted that the CCA and ICA have different plaque severity.

Elisa *et al.* [43] also proposed a DL-based technique for plaque area calculation. Their technique includes an image pre-processing step, a two-stage DL model for cIMT detection and geometric total plaque area gTPA measurement. In the beginning pre-processing step is applied to

remove the information part from the ultrasound images. In the first stage (run) of DL model, lumen borders are detected. In the second stage, IAD borders are detected using the ground truth values of lumen and IAD boundaries. Once LD and IAD borders are detected, cIMT values are calculated using the polyline distance metric (PDM) algorithm. Also, in the second stage cylindrical-fitting approach is applied to calculate the geometric total plaque area (gTPA). They used VGG16 model in an encoder-decoder format. Initial 13 layers were used as an encoder to extract features and the remaining 3 layers were used as decoder. Finally, a post-processing system was used to smooth out the LI/MA borders. Their research suggested a PA threshold of 38 mm^2 for moderate to high-risk patients. Also measured mean PA for 396 patients were 20.52 mm^2 and 19.44 mm^2 using two models. Major advantage of their system is that it's a complete package, which means the same system can be used for cIMT and gTPA detection. However, the model is very bulky and runs two times for LD and IAD detection. Once the LI/MA borders are detected, then only gTPA is calculated. Thus, the complete package is a tedious and complex system.

Further, Jain *et al.* [96] tested two SDL and HDL models on 379 low- to moderate-risk CCA images. This work was an extension of the previous work mentioned above. The best performing models from the above section were used for low-to-moderate risk CCA plaque images. CCA plaque segmentation is a widespread area of research where many authors have presented their work earlier. Therefore, in this paper, the authors benchmarked their results against the state-of-the-art AtheroEdge2.0 package developed by AtheroPoint, LLP, USA. Thus, their measured PA error was 4.07±4.71 mm^2 for SDL and 3.12±3.93 mm^2 for HDL models.

The benchmark model AtheroEdge2.0 had PA error of 3.72±4.76 mm^2. Thus, their SegNet-UNet HDL model outperformed the CCA plaque area measurement benchmark model. Major advantage of their system is that it outperforms the previous benchmark model AtheroEdge2.0, and is fully automated. However, there is research focused only on SDL and HDL model feasibility on low plaque CCA images and didn't consider other aspects such as memory size and time complexity, which is a significant disadvantage of their work.

Table 6.5 Benchmarking Table Comprising Recent Studies for PA Measurement Using Dl Models.

Authors	Artery Segment	AI method	Measure	#P/#Images	PA or TPA Performances	Pros	Cons
Jain *et al.* [97]	ICA	HDL, SDL	PA	970/97	PA error 3.49 mm^2 for SDL 4.21 mm^2 for HDL	Hybrid DL model Two loss functions compared, many statistical/performance parameters	Large memory size, Sluggish model, no multiethnic database study No benchmarking with established model
Zhou *et al.* [98]	CCA, ICA	UNet++, UNet	PA, TPA	144/510 497/638	Best TPA error: 5.55±4.34 mm^2	UNet model with dense skip connection, multiethnic database study	Small test dataset, Large memory size, Sluggish model No benchmarking with established model
Elisa *et al.* [43]	CCA	FCN	gTPA, cIMT	204/396	Mean PA 20.52 mm2 , 19.44 mm2	One system to find cIMT and gTPA, very reliable results	Very complex model, two stage to find area, no benchmark model, multiethnic dataset not used
Jain *et al.* [96]	CCA	SDL, HDL	PA	190/379	Absolute PA error 4.07±4.71 mm^2, and 3.12±3.93 mm^2	SDL and HDL model checked for low-to moderate CCA plaque feasibility, Outperforms benchmark model AtheroEdge2.0	Sluggish model, no multiethnic database study
Proposed method	CCA	SDL, HDL	PA	190/379	Absolute PA error SDL 4.89±4.52mm^2 HDL 5.97±5.52 mm^2	Faster training, small memory size Outperforms Autoencoder, based model, Multiethnic database used, both ICA and CCA sections are used for study	
	ICA	SDL, HDL	PA	97/970	Absolute PA error SDL 3.89±3.80 mm^2 HDL 4.33±4.15 mm^2		

#P: number of patients; #Images: number of images.

Table 6.6 Scientific Validation on Unknown Data.

	[1]ACC (%)	[2]SENS (%)	[3]SPEC (%)	[4]PREC (%)	[5]MCC (%)	[6]DICE (%)	[7]JI (%)	[8]CC	[9]AUC
UNet	99.01±0.33	82.36±7.67	99.72±0.22	91.80±8.55	86.32±6.69	86.54±6.71	76.74±6.71	0.93	0.967
UNet+	99.00±0.32	83.07±7.17	99.68±0.28	91.23±7.91	86.36±5.57	86.57±5.82	76.74±8.09	0.92	0.967
UNet++	98.96±0.36	80.65±7.72	99.73±0.23	92.33±6.87	85.61±5.58	85.75±5.80	75.47±8.08	0.92	0.954
Inception-Unet	99.00±0.32	81.60±7.45	99.73±0.21	92.41±6.48	86.15±5.39	86.31±5.84	76.31±7.72	0.92	0.962
Squeeze-UNet	98.70±0.48	76.06±10.40	99.67±0.30	90.52±8.00	82.02±7.18	82.02±7.74	70.16±9.77	0.83	0.913
Fractal-UNet	98.91±0.34	80.32±8.20	99.69±0.24	91.59±6.59	85.00±5.34	85.13±5.88	74.51±7.80	0.90	0.957
UNet3P	98.93±0.35	80.69±9.03	99.70±0.25	91.73±6.75	85.22±6.20	85.33±6.82	74.91±8.58	0.90	0.953
Autoencoder	98.70±0.55	77.82±12.83	99.61±0.26	88.85±7.80	82.27±8.02	82.09±9.86	70.58±11.70	0.81	0.891

[1]ACC=Accuracy; [2]SENS=Sensitivity; [3]SPEC = Specificity; [4]PREC= Precision; [5]MCC= Mathew's Correlation Coefficient; [6]DICE= Dice Similarity Coefficient; [7]JI= Jaccard Index; [9]CC= Correlation coefficient; [10]AUC=Area under the curve

6.6.5 Precision and Recall

The precision refers to the percentage of results which are positive. It is the ratio of TP (acceptable result) and total positive results (TP+FP) and is represented by equation (6.12).

$$\text{Precision} = \frac{TP}{TP+FP} \qquad (6.12)$$

The recall refers to the percentage of total acceptable positive results which are correctly classified by the system. It is the ratio of the TP (acceptable result) and total acceptable results produced by the system (TP+FN) and is represented by equation (6.13).

$$\text{Recall} = \frac{TP}{TP+FN} \qquad (6.13)$$

PP: Plaque Pixels; BP: Background Pixels

True Positive (PP:PP)	False Positive (BP:PP)
False Negative (PP:BP)	True Negative (BP:BP)

Total positive results (TP+FP) include prediction of plaque pixels (positive class) only. [True Positive (PP:PP), False Positive (BP:PP)]. Thus precision is ratio of prediction of correct plaque pixel to total positive results.

Whereas, total acceptable positive results (TP+FN) are correct prediction of plaque pixels (positive class) and background pixels (negative class). Thus recall is the ratio of prediction of correct plaque pixels to total acceptable positive results.

Our purpose is to not only correctly classify plaque and background pixels but to successfully reject the chances of background pixels to be classified as plaque pixels. Keeping the above concept in view, recall seems to play a vital role in system optimization. Based on recall values, we found UNet and Squeeze UNet models best for ICA database experiments. Similarly, we found UNet++ and Autoencoder models best for CCA database experiments. However, due to inconsistency performance for ICA and CCA database experiments and keeping other parameters such as CC, ROC, and AUC in view, Autoencoder is not a good choice for plaque segmentation.

6.6.6 Scientific Validation

We also performed scientific validation on *unseen* data. For this purpose, we selected a database of 300 images of CCA images from Hong Kong. A more detailed description of the database is given in reference [110]. The scientific validation aims to show the feasibility of the designed system with the *unseen* data. We used the same system parameters and didn't optimize the system for the *unseen* database and performed all experiments. Table 6.6 shows the all experimental and performance evaluation parameters of the U-Series eight architectures on unseen US scans. As seen from the table, all parameters align with the previous experiments.

6.6.7 Strength, Weakness, and Extensions

Our work demonstrated encouraging segmentation performance on ICA and CCA databases using SDL and HDL models. The parallel organization of convolutional layers enhances the training process and reduces the memory size while preserving the segmentation performance. We hypothesized that the HDL models are superior to the SDL models. The second major strength of our system is that we used two different CCA and ICA images databases for our experiments. ICA experiments analyse moderate to high plaque, whereas CCA experiments analyse low to moderate plaque images. Earlier studies concentrated only on one artery segment or combined both artery segments, which doesn't justify the plaque morphology.

Further, we performed each experiment five times using the K5 partition method; thus, each image gets a chance to be part of training and testing. A major weakness of our system is that the HDL and SDL models are equally performing the segmentation task. However, HDL models have other benefits, such as low memory requirement and low training time. Even though we did not perform variability analysis with any other tracer, our previous variability analysis with the same database shows the results within 1%, which follows the regulatory criteria of less than 5%. As part of an extension, advanced image processing tools can be applied to improve the reliability and design of the loss function in deep learning [148, 151]. The segmented wall can also be used to understand the plaque region's symptoms [139]. Application to video imaging is required where image registration will play an important role [151, 259, 260]. The method can also be extended to multi-label [119, 261] database, and scale-space [41, 247] applications. Also, this method is currently applied to Far wall plaque measurement only, which could be extended to near wall plaque measurement or both simultaneously. Finally, these evolutionary methods can be used for storage reduction and speed improvement [233].

Superior UNet models such as attention maps can be further incorporated into the skip connections for better feature extraction thereby improving the segmentation paradigm. UNets can be

enhanced by fusion of UNet with advanced conventional methods such as level sets . Risk stratification methods can be incorporated [262] for understanding the burden of carotid strokes [263]. There is also need to improve the storage capacity of UNet algorithms during paradigms [233, 264]. Advanced methods for AI-bias estimation can be incorporated for improving the AI performances [194, 265-268]. Lastly one can undergo cardiovascular risk stratification [36, 139, 175] in SDL/HDL frameworks.

6.7 Conclusion

We presented a novel work on U-series plaque segmentation models using three SDL and three HDL models. We validated our hypothesis that the HDL models have advantages over the SDL models. All models were analysed using various segmentation parameters and performance indices. UNet++ was the best performing model under the SDL category with CC, AUC, and FoM of 0.98, 0.974, and 88.43%, respectively, and plaque area (Mean±SD) of 4.89±4.52 mm2 for CCA experiments. Similarly, Squeeze-UNet was the best performing model under the HDL model category with CC, AUC, and FoM of 0.96, 0.969, and 86.17%, respectively, and plaque area (Mean±SD) of 5.97±5.52 mm2, along with seven times reduction in model size. The small memory size makes the HDL models the most suitable choice for web-based applications.

6.8 Proposed Extension for Next Chapter

The present chapter fills all the gaps from previous work, such as the artery subsection problem covered by considering ICA and CCA both, unseen model is covered by considering multi-ethnic, multicentre database. However, the method can still not properly segment a few critical images. Therefore, a robust way or change is required in the current models to segment all types of images in the database. We have incorporated a novel feature modification method which can perform better feature engineering in current models. The same method is described in the next chapter.