Applied Intelligence
https://doi.org/10.1007/s10489-022-03642-w

Semantic segmentation in medical images through transfused convolution and transformer networks

Tashvik Dhamija[1] · Anunay Gupta[2] · Shreyansh Gupta[3] · Anjum[4] · Rahul Katarya[4] · Ghanshyam Singh[5]

Accepted: 15 April 2022

Abstract
Recent decades have witnessed rapid development in the field of medical image segmentation. Deep learning-based fully convolution neural networks have played a significant role in the development of automated medical image segmentation models. Though immensely effective, such networks only take into account localized features and are unable to capitalize on the global context of medical image. In this paper, two deep learning based models have been proposed namely USegTransformer-P and USegTransformer-S. The proposed models capitalize upon local features and global features by amalgamating the transformer-based encoders and convolution-based encoders to segment medical images with high precision. Both the proposed models deliver promising results, performing better than the previous state of the art models in various segmentation tasks such as Brain tumor, Lung nodules, Skin lesion and Nuclei segmentation. The authors believe that the ability of USegTransformer-P and USegTransformer-S to perform segmentation with high precision could remarkably benefit medical practitioners and radiologists around the world.

Keywords COVID-19 · Deep learning · Image-segmentation · Transformer · U-net

1 Introduction

SINCE the dawn of the 21st Healthcare industry has started adopting technology rapidly. One area in which this adoption of technology has given remarkable results is the area of "Medical Imaging". Medical imaging refers to techniques and processes used to mimic the state of organs via images CT-Scan, X-rays and MRI [1]. Until the past decade, the advancements in medical imaging were more focused on optimizing and improving the process of creating organ images and enhancing the quality of medical images. While the process of inferring information was left untouched and immensely dependent on the availability of experts and trained professionals. The lack of innovation in information inferring processes has led to burdened healthcare infrastructures in many countries. The only way to reduce this burden is to automate certain aspects of inferring information from medical images. One domain that has a lot of scope automation is segmentation.

✉ Rahul Katarya
rahuldtu@gmail.com

Tashvik Dhamija
tashvik369@gmail.com

Anunay Gupta
aganunay@outlook.com

Shreyansh Gupta
shreyansh.gupta1299@gmail.com

Anjum
anjum_2792@yahoo.com

Ghanshyam Singh
ghanshyams@uj.ac.za

[1] Department of Electronics and Communication Engineering, Delhi Technological University, New Delhi, India

[2] Department of Electrical Engineering, Delhi Technological University, New Delhi, India

[3] Department of Civil Engineering, Delhi Technological University, New Delhi, India

[4] Department of Computer Science Engineering, Delhi Technological University, New Delhi, India

[5] Department of Electrical and Electronic Engineering Science, University of Johannesburg, Johannesburg, South Africa

Published online: 25 April 2022

Segmentation in medical imaging deals with labelling each pixel on the image with a class which is known as Semantic segmentation. Semantic Segmentation has a plethora of applications in the healthcare industry. Some of the areas where segmentation has been used are - detection of lung nodules from CT-Scans of lungs [2], Detection of Brain tumor [3], segmentation of skin lesion [4], Polyp detection [5] (Polyps are irregular tissue growth on body), Liver segmentation [6], Nuclei segmentation [7], etc. A lot of research has been put into developing segmentation models and algorithms using multiple toolboxes. A lot of research has been put into developing segmentation models and algorithms using multiple toolboxes. After going through literature, we found that, until now the problem of medical image segmentation has been tackled with algorithms that can be grouped into 4 broad categories namely, 1) machine learning-based algorithms, 2) early convolution neural networks (CNN), 3) optimized CNN, and 4) feature extracting convolution networks. Each category of algorithms is presented as follows [8-10].

1.1 Machine learning based algorithms

Machine learning methods in the medical image segmentation are broadly classified into two categories such as 1) supervised learning and 2) unsupervised learning. The supervised methods employed artificial neural networks [11]. However, the unsupervised methods consisted of extensive use of K-mean algorithm [12], Hard C-mean algorithms [13] and Fuzzy C-means algorithms [14]. The common problem among these techniques is that such techniques majorly rely on predefined features or some structure. The reliance on predefined features reduces the generalizing ability and robustness of the techniques.

1.2 Early convolutional networks

With the significant development in the deep learning, the task of medical image segmentation has tried to solved using Convolutional Neural Networks (CNN) [15] models. Early on, the patching-based CNN models are used over primitive algorithms as reported in [16]. Then, a ground-breaking improvement has been observed with the invent of the UNet [17]. It builds upon the concept of using a fully convolutional network (FCN) and it consists contracting segment that encodes the features and an expansive segment that generates the mask from the encoded features. This architecture is capable of learning high-resolution features to provide more precise outputs. The authors also introduced the use of data augmentation during training to aid the training of models on smaller datasets and to make them robust. The limitation of the UNet architecture is that they use simple skip connections which lack the ability to transfer spatial features across the encoder and decoder.

1.3 Optimized convolutional networks

In order to overcome the limitation of simple skip connections in the Unet, multiple architectures with complex skip connections are introduced. in [18], the authors have proposed an architecture named, MultiResUNet. This model replaces pairs of standard convolution layers present in the vanilla UNet with inception like MultiRes block to restore features learned at various scales while maintaining memory efficiency. The MultiRes block also consists of 1×1 convolutions to learn spatial information, owing to such additions MultiResUnet passes baselines set by Unet on multiple datasets. Further, the authors in [19] introduced DC-UNet that builds upon the MultiResUNet and aimed to improve the performance of image segmentation algorithms on tricky dataset. Further, the authors replace the simple residual connection in the MultiRes block with a sequence of 3×3 convolutions to increase the ability of the model to learn better spatial features. This architecture can learn features of different scales, reduce the semantic gap and learn spatial features. The DC-UNet is able to perform considerable well on infrared (IR) breast dataset, IEEE-ISBI dataset, and CVC_Clinic DB. Another approach as reported in [20] is taken to introduce efficient skip connections in the original UNet. The authors proposed a novel architecture, the UNET++ for semantic and instance image segmentation. The architecture differs from the original UNet architecture proposed in [17] in terms of optimized skip connections and connections between adjacent nodes. Thus, such modifications equip UNET++ to have considerable advantages over the traditional UNet architecture like always having the optimal depth and efficient fusion of features in the decoder. Further UNET++ achieves decent results on six problems in the domain of medical image segmentation, namely, Electron Microscopy (EM), Cell segmentation, Nuclei segmentation, Brain tumor segmentation, Liver Segmentation, and Lung Nodule Segmentation. The third way to improve the UNet by making the skip connections more effective has explored in [21] where the authors of BCDU-NET make use of Bi-ConvLSTM in the skip connections, which assist in relaying semantic information between the corresponding layers. The effectiveness of this result is proved by achieving state-of-the-art results on three benchmark datasets, namely, Drive Dataset, ISIC 2018 Dataset, and Lung Nodule Analysis (LUNA) dataset.

1.4 Feature extraction convolutional networks

In order to further enhance the performance of image segmentation algorithms, multiple newer architectures which focus on improving the feature extraction power of the UNet encoder have been introduced. In [22], the authors have proposed DoubleU-Net model for medical image segmentation which is based on the idea of using two UNets wherein the first UNet

uses a VGG-19 encoder pre-trained on ImageNet and the second UNet takes in the multiplication of the original image and the output of the first UNet as input. The DoubleU-Net performs well on datasets such as CVC-Clinic DB, ISIC 2018, and 2018 Data science Bowl challenge datasets. Further, a different approach has been taken by the authors of a novel segmentation architecture DRINet [23], which is based on DenseNet and Inception-ResNet. The dense blocks in the architecture of DRINet are used in the encoder part and residual inception blocks, along with unpooling from the decoder of the DRInet. Thus, such blocks are used to replace standard convolution blocks to learn distinctive features in cases of similar shape, intensity, location, and size. The DRINet performs well on CSF, CT, and multi-organ datasets. In [24], the authors have proposed another Novel architecture for 2D medical image segmentation, namely, CE-NET which has 3 major components such as Feature Encoder, Content extractor, and decoder in contrast to encoders, and decoders seen in typical UNets. The use of a context extractor that performs Atrios convolutions help to make the filters act on global features and thus provides the model more context. The CE-NET achieves excellent results in Optic disc image segmentation, Retinal Vessel Detection, Lung segmentation, and Cell contour segmentation.

A potential research challenge in the aforementioned architectures is that these models are only capable of extracting local context and unable to take into account global features, and long-range dependencies in the image data. Despite the heavy optimization in the extraction and usage of local features, the improvements in models' performance are significantly low. We hypothesize that this is due to the lack of consideration of global features and intending to overcome this research gap and fully capitalize on global features in segmenting medical images. We propose the USegTransformer-P and USegTransformer-S approach which leverage both local features from the full convolution networks and long-term dependencies obtained by transformers. We believe that such a combination will yield state-of-the-art results. The novelty and salient features of this paper is summarized as follows:

- We have developed a novel medical image segmentation models, USegTransformer-P and USegTransformer-S, which utilizes both the local and global features of the image using an amalgamation of Transformers based encoders and UNet based encoders.
- We have developed a fully convolutional ensemble decoder for transfusing both the local and global features obtained from a fully convolutional network and transformer network respectively, from an image in a learnable manner.
- We have analyzed the performance of sequential and parallel stacking of transformer-based encoder-decoder and convolution-based encoder-decoder in order to achieve a better configuration of the proposed models.
- We have proved the efficacy of the proposed models by comparing them with reported medical image segmentation models on benchmark datasets used in renowned competitions such as LUNA, ISIC-2018, and Kaggle Data science bowl.

The paper is further divided into 4 sections, namely, the methodology, results and experimentation, discussion, and conclusion and future scope. In the methodology section, we explain every intricate detail about our proposed models, USegTransformer-P and USegTransformer-S. In the result section, we explain and analyze the performance of our proposed model with the help of metrics and visualizations. In the discussion section we probe into what are qualitative and quotative effects of the use of global features in segmentation algorithms. In the final section that is conclusion and future work; we present our findings as well as discuss the direction future works that can be done in the field in the automated medical segmentation.

2 Proposed architecture

The UNet-based model extracts localized features in the high-level representations such as the CNN models. On the other hand, transformer-based models like ViT extract global context and long-range dependencies. In a task like an image segmentation, we would require both kinds of feature representations. Therefore, in this proposed system model, we come up with two different methods to use the best of both worlds. These methods take advantage of the different feature extraction abilities of a transformer model and CNN model. By doing this, the proposed system model is capable of understanding the local features as well as the global context. The first model, known as the USegTransformer-P shown in Fig. 2a, is a parallel model and the second model that is the USegTransformer-S as shown in Fig. 2b, is a sequential model inspired from the work reported in [22].

The proposed system models derive their name from the use of a transformer-based encoder-decoder architecture stacked with U-Net inspired encoder-decoder convolutional architecture to segment the bestowed image. The suffix of USegTransformer (-P and -S) signifies the type of stacking. For the given image $img \in R^{H \times W \times C}$ with a spatial resolution of $H \times W$, where H and W denote height and width of the conferred image and C channels. The main goal is to yield a segmentation mask $msk \in R^{H \times W}$. The architecture of USegTransformer-P and USegTransformer-S consists of two common parts, namely, the transformer-based encoder-decoder and an UNet inspired encoder-decoder convolutional block.

These components and the stacking details of the proposed models are discussed in the following subsections.

2.1 Transformer based image encoder-decoder

The vision Transformer models have been quite successfully used in both computer vision [25] and natural language processing [26] and have provided promising results in both fields. The primary edge that the transformers have over other techniques is efficiency in terms of computational resource usage and its efficacy in performing various tasks. This success of the vision transformers model can be attributed to parallelization, where, unlike in Recurrent Neural Networks (RNNs) and Long Short-Term Memory (LSTM), all time-steps can be passed in a single step making the model computationally efficient. Secondly, the transformers use a self-attention mechanism to gain global attention features from the data which attributes to the results that the transformers achieve.

2.1.1 Transformer encoder

This research study used a transformer-based self-attention architecture to encode the images into high-level features with a global context. We have first divided the image $\{img^i \in R^{H \times W \times C}\}$ into two-dimensional (2D) flattened patches $\left\{img^i_{pat} \in R^{P^2.C}\right\}$, where $i \in [1 \ldots N]$ and each patch is of size $P \times P$ with $N = \frac{W.H}{P^2}$ to make the image data sequential and time distributed as reported in [25]. Further, we have also added a linear layer called the projection layer to project these flattened patches into a L dimension vector where L is the length of image sequence and these patches are known as Projected Patches. To include the spatial information of the input image in the proposed model, we added these projected patches with a positional encoding matrix $\left\{E_{Pos} \in R^{N \times Embed_{Size}}\right\}$ where EmbedSize is given as $Embed_{Size} = L$. The positional encodings matrix can be developed in different ways however, we have kept the positional encoding matrix as a learnable parameter in this proposed research, i.e., the model will also update the elements of this positional embedding matrix during back-propagation. The positional encodings and the flattened patches are combined and mathematically presented as:

$$P_{embed} = \left[x^1_t E, x^2_t E, x^3_t E, \ldots, x^N_t E\right] + E_{Pos} \tag{1}$$

where E is the linear projection matrix such that $E \in R^{\left(P^2 C \times L\right)}$ and E_{Pos} represent the learnable position encoding. After the image's encoding is performed, we move on to apply the Multi-Headed-Self-Attentions (MHSA) and Position-wise Feed-Forward Networks (FFNs) layers. The transformer consists of H Multi-Head-Self-Attention (MHSA) and H Position-wise Feed-Forward Networks (FFNs) blocks. The Eqs. (2) and (3) explain the operations performed during the self-attention phase.

$$\ddot{z}_j = MHSA\left(LinNorm\left(z_j\right)\right) + z_{j-1} \tag{2}$$

$$z_j = FFN\left(LinNorm\left(\ddot{z}_j\right)\right) + \ddot{z}_{j-1} \tag{3}$$

where $LinNorm(\,)$ denote the Linear Layer Normalization and z_j is a feature tensor. The FFN block includes linear layers with the rectified linear unit (ReLU) activation function, and the MHSA contains L self-attention heads (SAs) connected in parallel. In each SA head, we cmputed the attention by mapping queries and key-value pairs to an output. The key (K), value (V), and query (Q) matrixes are formed by the encoded input matrices. Now, we compute the dot of the query matrix with all the key matrices and then scale each of them by $\sqrt{d_k}$ where d_k is the dimensions of the key matrix and query matrix and apply the SoftMax function to obtain the self-attention weight matrix, which then multiplied with the value matrix. The Eq. (4) below explains this procedure mathematically.

$$Attention(K, Q, V) = Softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{4}$$

The outputs from all the attention heads are concatenated into a single matrix, which is represented as:

$$MHSA(z) = [h_1, h_2, h_3 \ldots, h_H] W_{MHSA} \tag{5}$$

where W_{MHSA} represents the weights of the learnable weight matrix of different SAs. The flow of input through the transformer encoder layer can also been understood by analysing Fig. 1.

2.1.2 Convolution based decoder

After obtaining the encoded images through the transformer-based encoder, we pass the encoded vector, which is first reshaped and rearranged to a 2D matrix $Img_{encode} \in R^{\ddot{w} \times \ddot{h} \times c}$ and then passed through a convolutional auto-encoder inspired decoder as reported in [27]. Here, we have used decoder convolutional up-sampling blocks. Each convolutional up-sampling block includes a Conv2D layer with a kernel size of 3 × 3 with a Batch-Normalization layer and ReLu activation with a bilinear up-sampling layer that has a scale factor of 2. The image from the bottleneck encoder is passed through each decoding step where at every step, we up-sample the image by a factor of 2. This process is continued until we achieve the

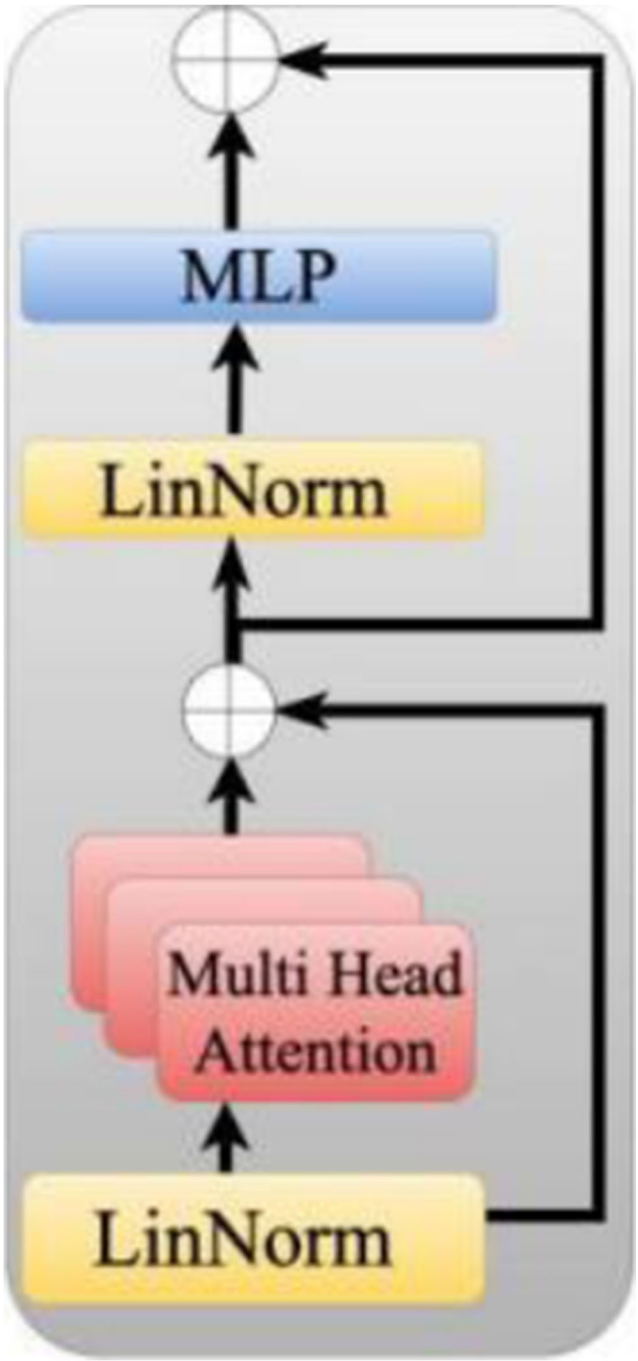

Fig. 1 Schematic of the transformer based image encoder

original image resolution. At the final stage, we use a convolution layer with 1 × 1 kernel size and with the sigmoid activation in the end.

2.2 U-net inspired encoder-decoder block

As pellucid from the above discussion, the transformer-based encoder provides a global high-level representation of features. Therefore, we need a fully convolutional network to yield localized high-level feature representation to identify subtly local regions in the masked image. Hence, we use a UNet-inspired fully convolutional encoder-decoder architecture for this purpose. The encoder block consists of a sequence of 2D convolution, batch normalization, and activation, repeated twice, followed by a pooling function to downscale. The decoder block consists of a sequence of linear or bilinear up-sampling followed by 2D convolution, batch normalization, and activation function, repeated twice. Additionally, there exists a stacking of encoder outputs to decoder inputs at the same dimension across the encoder and the decoder through skip connections. The flow of input image in a UNet can be mathematically expressed as:

$$R^{\epsilon H\times W\times 3}\Rightarrow R^{\epsilon h\times w\times c}\Rightarrow R^{\epsilon H\times W\times C} \tag{6}$$

In the experiments, all convolutions have a 3 × 3 kernel with padding = 1. The activation function uses a standard ReLU function. The pooling is applied using max-pooling. The input image passes through a series of encoder blocks wherein, after each block, the number of channels scales up by a factor of 2. The final encoded activation map has 1024 channels. It passes through a series of decoder blocks wherein, after each block, the number of channels scales down by a factor of 2 shown in Fig. 2.

2.3 USegTransformer-P: Fully convolutional ensemble decoder

USegTransformer-P is a parallel stacked model of the transformer-based self-attention model and the UNet based model using a fully convolution-based decoder. The image is input into both the networks individually to output segmentation masks. These masks are stacked together and are convoluted using a 1 × 1 kernel. By using a 1 × 1 convolutional kernel, the model projects the best features by transfusing local and global features to produce a more accurate segmentation map. The flow of images in the network is mathematically represented as:

$$mg\in R^{H\times W\times 3}\Rightarrow feature\ matrix\in R^{H\times W\times 1} \tag{7}$$

$$img\in R^{H\times W\times 3}\Rightarrow feature\ matrix\in R^{H\times W\times 1} \tag{8}$$

$$R^{H\times W\times 1}+R^{H\times W\times 1}\Rightarrow concatenated\ matrix\in R^{H\times W\times 2} \tag{9}$$

$$R^{H\times W\times 2}\Rightarrow mask\in R^{H\times W\times 1} \tag{10}$$

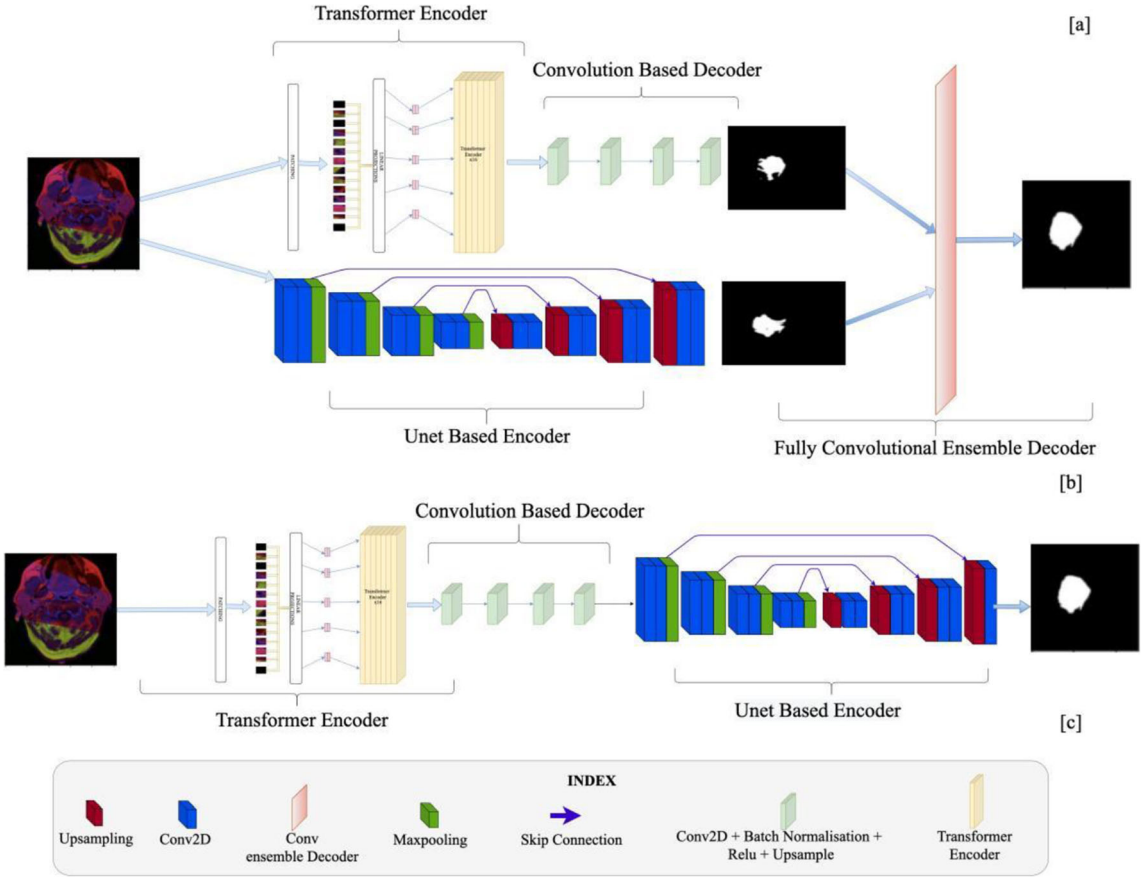

Fig. 2 Schematic of the proposed architecture (**a**) USegTransformer-P, (**b**) USegTransformer-S, and (**c**) index for the figure

where Eq. (7) represents the output of transformer-based encoder-decoder, (8) represents the outputs UNet inspired encoder decoder. The Eqs. (9) and (10) explain the image transformations in the fully convolution ensemble decoder. This model is trained using Algorithm-1.

Algorithm-1: Training of a USegTransformer-P Model

Input: A batch of image tensors
Output: Trained USegTransformer-P model

```
for step t = 1,2…T do
    for batch b = 1,2…N do
        x ← image tensor [b]
        x1 ← x
        for layer l = 1,2...L do
            x1 ← transformer_encoder(x1)
        End
        x1 ← convolution_based_decoder(x1)
        x2 ← unet_based_encoder_decoder_block(x)
        y_predicted[b] ← fully_conv_ensemble_decoder(x1,x2)
    End
    loss ← loss_function (y_predicted, y_true)
    Backpropagating loss error and weight update
End
```

The above training Algorithm-1 runs for T epochs, each step training on N batches making up the whole dataset. Each image in a batch runs through L layers of transformer encoder. The encoder applies attention to its input at each layer giving out a feature space that has considered and attended to the correlation between the image tokens. The final feature map is decoded using a convolutional decoder. In parallel, the image is also passed through a UNet based encoder-decoder to capture the spatial features and correlations. The two feature maps from both branches are combined and decoded to produce the output. The ensemble decoder combines the two feature spaces from the two branches and considers long range dependencies as well as spatially close dependencies to decide what features are more important where and then produces the segmented mask, which is an amalgamation of both branches. The output is compared to the ground truth and the error is backpropagated to update weights. The error is calculated using a loss function.

2.4 USegTransformer-S: Sequential stacking

The USegTransformer-S is a sequential stacked model of the UNet based model and inspired by the transformer-based self-attention model. The activation map produced by the transformer-based encoder-decoder is multichannel (in the proposed experiments, it has two channels). This multichannel

activation map is then input to the UNet based encoder-decoder. We intend to progressively down sample the number of channels in the output of each decoder. This helps in the transfusion of global features obtained by Transformer encoder and local feature obtained by Unet based encoder-decoder. The flow of the image in the network is mathematically represented as:

$$R^{\epsilon H\times W\times 3}\Rightarrow R^{\epsilon h\times w\times c}\Rightarrow R^{\epsilon H\times W\times 2}-1 \tag{11}$$

$$R^{\epsilon H\times W\times 2}\Rightarrow R^{\epsilon h'\times w'\times c'}\Rightarrow R^{\epsilon H\times W\times 1}-1 \tag{12}$$

The USegTransformer-S is trained using Algorithm-2.

Algorithm-2: Training of a USegTransformer-S Model

```
Input: A batch of image tensors
Output: Trained USegTransformer-S model
    for step t = 1,2…T do
        for batch b = 1,2…N do
            x ← image tensor [b]
            for layer l = 1,2...L do
                x ← transformer_encoder(x)
            end
            x ← convolution_based_decoder(x)
            x ← unet_based_encoder_decoder_block(x)
            y_predicted[b] ← x
        end
        loss ← loss_finction (y predicted, y_true)
        Backpropagating loss error and weight update
    end
```

The above training Algorithm-2 runs for T epochs, each step training on N batches making up the whole dataset. Each image in a batch runs through L layers of transformer encoder. The encoder applies attention to its input at each layer giving out a feature space that has considered and attended to the correlation between the image tokens. The final feature map is decoded using a convolutional decoder to give a multi-channel feature space. This feature space is further passed through a UNet based encoder decoder that capture the spatial features and relations in the high-level long-range feature space. The feature space also goes through a change in number of channels incorporating features at multiple levels. The output is, hence, a convoluted output of independent long-range features. This output is then compared to the ground truth and the error is backpropagated to update weights. The error is calculated using a loss function. Experimentally, as discussed in the next section, we have found that both ensembling methods perform better or at par with the previous state-of-the-art models wherein, the parallel model achieves superior results. Therefore, the fully convolutional ensemble decoder exploits the feature extraction abilities of both networks in a better way.

3 Experimental results

In order to investigate the capabilities of the proposed models, we test them on benchmark datasets. Each dataset chosen have a different set of challenges with a unique application in the field of medical imaging. The following section discusses in detail the various intricacies of each dataset as well as analyzes the results obtained on each dataset.

3.1 Dataset description

3.1.1 MRI dataset

The dataset [28] consists of MRI scans and corresponding segmentation masks of 110 patients obtained from the cancer imaging archive (TCIA). The patients are from the cancer Genome Atlas (TCGA) with low-grade glioma with fluid-attenuated inversion recovery (FLAIR) sequence. We have divided the chosen data into 70% training set(2838 images), 15% validation set(501 Images) and 15%(510 Images) testing set.

3.1.2 Lungs segmentation dataset

The dataset [29] is from the Lung Nodule Analysis (LUNA) competition from 2016. The competition aims to obtain automatic module detection algorithms. It contains 267 2D CT scans of lungs and their corresponding segmentation masks. We train the proposed models on 216 images (80%), validate the proposed models on 24 images (~10%), and test on 27 images (~10%).

3.1.3 Nuclei segmentation dataset

This dataset was part of the prestigious annual Kaggle data science bowl 2018 competition [30]. The dataset provides us with 670 images and along with them the segmented mask of each nucleus. The dataset contains images captured under varied conditions such as magnification, brightfield and fluorescence. The difference in quality images makes this dataset a true challenge for any deep learning model. In this paper, we use 603 (90%) images for training and 67 (10%) images for testing.

3.1.4 Skin lesion segmentation dataset

We use the dataset provided by the annual ISIC competition in 2018 [31, 32]. This dataset is meant to accomplish three tasks namely Lesion segmentation, lesion attribute detection and Disease classification. But since we intend to show our proposed models' proficiency in segmenting medical images, we concentrate on the first task. The dataset contains 2694 images and their corresponding masks. The masks are annotated by a

committee of experts. In this paper, we train our model on 75% of the data which is 2075 images and test the model's efficiency on 20% which is 518 images of the data and validated our model on 5% ~ 100 images.

3.1.5 COVID-19 CT scan segmentation dataset

This is a part of COVID-19 CT Image Segmentation competition hosted on Kaggle [33]. The dataset consists of 9 segmented axial volumetric CTs from Radiopedia. This dataset contains entire quantities and hence both positive and negative slices (373 out of the total of 829 slices have been evaluated by a radiologist as positive and segmented). These slices are then converted and normalized. Here the masks in the dataset consists of 4 classes or channels (ground glass, consolidations, lungs other, and background) out which 2 of those (ground glass and consolidations) have been used for evaluation of our proposed model. We trained the proposed model of 85% of these slices i.e., 704 slices and tested on 15% i.e., 125 slices.

The data sample splitting is conducted in the experiments for the different medical segmentation datasets as shown in Fig. 3. The split is made such that they are similar to splits in previous state-of-the-art and baseline results for the most appropriate comparison.

3.2 Pre-processing and training

All the experiments have been conducted in the PyTorch [34] framework on Google Collaboratory Pro. The GPU used was Tesla P-100. All the models were trained using the Ams grad [35] variant of Adam optimizer with the most appropriate learning rate and model hyperparameters according to the available resources. The data was pre-processed with a combination of some primary image augmentations like Random rotate, Random flip, Random Affine, and more. These augmentations were applied using Torchvision and Albumentations libraries in python. All the experiments were conducted under major resource constraints. Improvement in the following results might be seen by changing and tuning various variables given the required resources are available such as better GPUs, more disk space, and RAM.

3.2.1 Metrics

Three metrics are used for quantifying the performance of the proposed model which are Dice similarity coefficient, intersection over union and pixel accuracy [36]. The pixel accuracy is the percentage of pixels in the predicted mask that match the expected pixel class in the ground truth mask. It is the most simple and primitive metric which is prone to class imbalance. The mathematical formula to determine pixel accuracy is presented as:

$$Pixel\ Accuracy = \frac{N_{TP} + N_{TN}}{N_{TP} + N_{TN} + N_{FP} + N_{FN}} \tag{13}$$

where, N_{TP}, N_{TN}, N_{FP}, and N_{FN} are the correctly classified pixels as Class A, correctly classified pixels as not Class A, incorrectly classified pixels as class A, and incorrectly classified pixels as not Class A, respectively. The intersection over union (IoU) is one of the widely accepted metrics of measuring the efficacy of the segmentation algorithm. It measures the overlap between the ground truth and the predicted masks by dividing the area of overlap by the area of union. The mathematical expressions to determining IoU is given as:

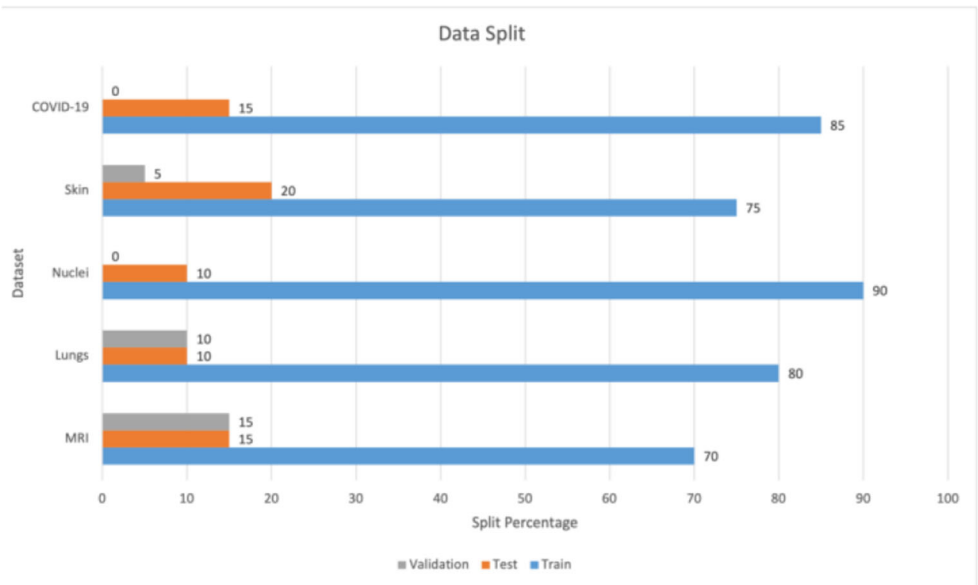

Fig. 3 Split in datasets

$$IoU = \frac{|N_A \cap N_B|}{|N_A \cup N_B|} \tag{14}$$

where, N_A, N_B, $N_A \cap N_B$, and $N_A \cup N_B$ are the pixels of Image A, pixels of Image B, area of overlap and area of union, respectively. The Dice coefficient is one of the best metrics to measure the efficacy of segmentation algorithm. It measures the similarity between the ground truth and predicted masks by dividing the number of overlapping pixels by total number of pixels in both images and multiplying the results by two. Equation (15) shows the mathematical expressions for determining Dice.

$$Dice = \frac{2*|N_A \cap N_B|}{|N_A| + |N_B|} \tag{15}$$

where, N_A, N_B, and $N_A \cap N_B$ are the pixels of image A, pixels of image B, and the area of overlap, respectively.

3.2.2 Loss functions

Binary cross entropy loss function is a standard loss function used in classification and segmentation task. The primary advantage of binary cross entropy loss function is that it provides smooth loss curves which contributes towards faster training of models.

$$BCE = -\frac{1}{N}\sum_{i=0}^{N} y_i \cdot \log\left(\widehat{y}_i\right) + (1-y_i) \cdot \log\left(1-\widehat{y}_i\right) \tag{16}$$

The Dice Loss is a loss function that makes the model strive towards producing images which are similar to the ground truth. It is slowly becoming a popular choice owing to its inherent maximization of dice coefficient and its salubrious effect on class imbalance. Mathematically dice loss is calculated by simply subtracting dice coefficient from 1.

$$Dice\ Loss = 1 - \sum_{i=0}^{i=N} \frac{2*\left|y_i \cap \widehat{y}_i\right|}{|y_i| + \left|\widehat{y}_i\right|} \tag{17}$$

All the models are trained using the standard binary cross-entropy (BCE) Dice loss, since both BCE loss and Dice loss when combined are observed to have a symbiotic relationship. The combined loss has relatively smooth loss curves owing to the component contributed by BCE loss as well Dice Loss's inherent tendency to maximize Dice coefficient and handle class imbalance which is prevalent in medical imaging datasets.

$$Combined\ Loss = -\frac{1}{N}\sum_{i=0}^{N} y_i \cdot log\left(\widehat{y}_i\right) + (1-y_i) \cdot log\left(1-\widehat{y}_i\right) + 1 - \sum_{i=0}^{i=N} \frac{2*\left|y_i \cap \widehat{y}_i\right|}{|y_i| + \left|\widehat{y}_i\right|} \tag{18}$$

3.3 Results

In this subsection, we have discussed in detail as well as analyze the prediction made by the models on various datasets. The quantitative analysis is done on the aforementioned metrics however, only one metric could be used for comparative analysis with prior models since most datasets were a part of a competition which required reporting that specific metric. Further visual predictions made by proposed model are also presented in order to analyze the performance of the proposed model qualitatively.

3.3.1 Brain tumor segmentation

One of the most crucial application of medical image segmentation is Brain tumor segmentation from Brain MRI. We choose Brain MRI LGG dataset from Kaggle to evaluate our proposed model. From Table 1, it is depicted that the USegTransformer-P performs the best across all segmentation metrics, that is, Dice score, Intersection over union and accuracy. The USegTransformer-P achieves an accuracy of 99.71 while achieve an IOU score of 0.9734 and a dice of 0.8934 while converging approximately at the same time with transformer backbone as seen in Fig. 4b. Figure 4c shows the validation curve (BCE Dice Loss Vs Epoch) for all three models, here even though the transformer backbone outperforms USegTransformer-S and USegTransformer-P on BCE Dice loss, yet USegTransformer-S and USegTransformer-P achieve a better dice coefficient and a better IoU score on validation split as well. Furthermore, these results can be visually analyzed from Fig. 4a. The transformer Backbone also performs efficiently achieving a dice score 8770, an IOU of 0.9381 and an accuracy of 0.9961. It is interesting to note that

Table 1 Key performance indicator of MRI segmentation

Model	Dice	IoU	Accuracy
U-Net [17]	0.8200	–	–
SynthSeg [37]	0.8610	–	–
Transformer Backbone	0.8770	0.9381	0.9961
USegTransformer-P	**0.8934**	**0.9746**	**0.9971**
USegTransformer-S	**0.8504**	**0.9634**	**0.9954**

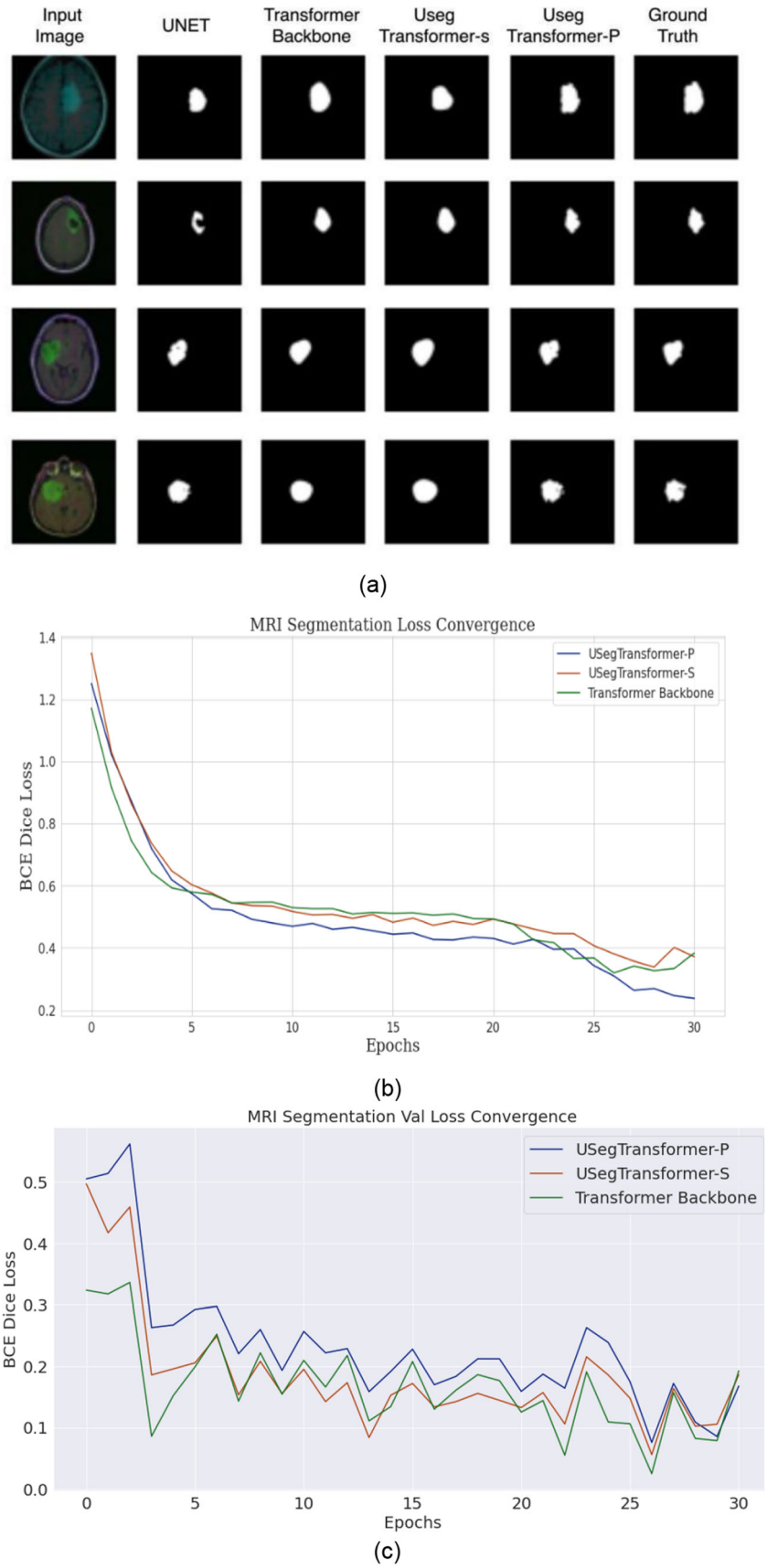

Fig. 4 The Brain MRI segmentation (**a**) visual depiction, (**b**) train loss convergence and (**c**) validation Loss

on this particular dataset transformer backbone performs better than the USegTransformer-S which is a deeper model. The USegTransformer-S achieves a dice score of 0.8504, an IOU of 0.9634 and an accuracy of 0.9954.The subpar performance of the USegTransformer-S on this dataset can be attributed to the lack of images in the dataset in comparison to the depth of the model. As shown by Table 1, the model outperforms state-of-the-art of models by a considerable overhead.

3.3.2 Lung nodule segmentation

Another crucial application where an artificial intelligence can revolutionize medical diagnosis is Lung Nodule analysis. We choose the dataset provided in the famous challenge LUNA to evaluate the proposed model's efficacy in segmentation of lung nodules. As presented in Table 2, the USegTransformer-P performs the best amongst Transformer Backbone, the USegTransformer-S and USegTransformer-P. The USegTransformer-P achieves a Dice score of 0.9807, an IOU of 0.9462 and an accuracy of 0.9913. While the USegTransformer-S achieves a Dice score 0.9777, an IOU of 0.9316 and an accuracy of 0.9913. Figure 4b and Fig. 4c shows the convergence of the models on BCE Dice Loss. The loss convergence and the results on the test split clearly show how USegTransformer-P generalizes on the dataset. The results are visualized in Fig. 5a. The Transformer Backbone, which is the least complex model achieves a Dice score of 0.9582, an IOU of 0.8946 and an accuracy of 0.9816. It can be observed from the Table 2 that the proposed model either outperforms most of the state-of-the-art models while performing at par with others.

Additionally, we performed 3-fold cross validation on the lungs segmentation dataset resulting in the model to be trained and tested on the data being divided into 2:1 ratio (66% training and 33% testing split). The results in Table 3 showcases that the model upholds its high dice, IoU and accuracy metrics with very low standard deviation across the 3 folds. This proves that the proposed architectures are robust to the data split.

Table 2 Key performance parameters of lungs segmentation

Model	Dice	IoU	Accuracy
U-Net [17]	–	–	0.9872
RU-Net [38]	–	–	0.9836
CE-Net [24]	–	–	0.9900
ET-Net [39]	–	–	0.9868
BCDU-Net [21]	–	–	0.9946
Transformer Backbone	0.9582	0.8946	0.9816
USegTransformer-P	**0.9807**	**0.9462**	**0.9913**
USegTransformer-S	**0.9777**	**0.9316**	**0.9894**

3.3.3 Skin lesion segmentation

The Skin Lesion Segmentation is a vital process in medical diagnosis since it forms the basis for more complex analysis. Table 4 presents the key performance parameters of proposed model on the ISIC 2018 competition dataset. The USegTransformer-P is the most effective in skin lesion segmentation achieving an accuracy of 0.9514, an IOU of 0.8672 and a Dice score of 0.8701. Figure 6a provides a visual depiction of these results. Further, the USegTransformer-P shows better and faster convergence and does not overfit on the data as evident Fig. 6b and Fig. 6c. The performance of Transformer Backbone and USegTransformer-S is similar. The USegTransformer-S achieves a Dice score of 0.8420, an IOU of 0.8374 and an accuracy of 0.9431. While the Transformer Backbone achieves a Dice score of 0.8503, an IOU of 0.8511 and an accuracy of 0.9447. As evident by Table 3, the best model proposed in this paper is able to significantly outperform various state of segmentation architectures.

3.3.4 Nuclei segmentation

An automation in the field of nuclei segmentation can prove to be a game changer in bio-medical research and thus is an important area of research. We evaluate the efficacy of proposed model in segmenting nuclei using the dataset provided in Kaggle 2018 data science bowl. From Table 5, it is evident that the proposed model that is USegTransformer-P has outperformed all the other models by achieving a Dice score of 0.9004, an accuracy of 0.9761 and IoU of 0.8470. These exemplary results can be seen in Fig. 7a. On the other hand, USegTransformer-S has achieved a dice score of 0.8517, accuracy of 0.9653 and an IoU of 0.7949. The Transformer Backbone has however outperformed the USegTransformer-S model, achieved a Dice score of 0.8780, accuracy of 0.9694 and IoU of 0.8287. The results explained above are reiterated by analyzing loss curves in Fig. 7b which shows that the USegTransformer-P is able to converge better.

Further, we performed 3-fold cross validation on the nuclei segmentation dataset resulting in the model to be trained and tested on the data being divided into 2:1 ratio (66% training and 33% testing split). The results in Table 6 showcase that the model upholds its high dice, IoU and accuracy metrics with very low standard deviation across the 3 folds. This proves that the proposed architectures showcase high degree of robustness.

3.3.5 Covid −19 CT-scan segmentation

An automatic segmentation also proves to be a boon to the bane of humanity that is Covid-19 pandemic. A major effect of Covid −19 is consolidation in chest CT. We believe that

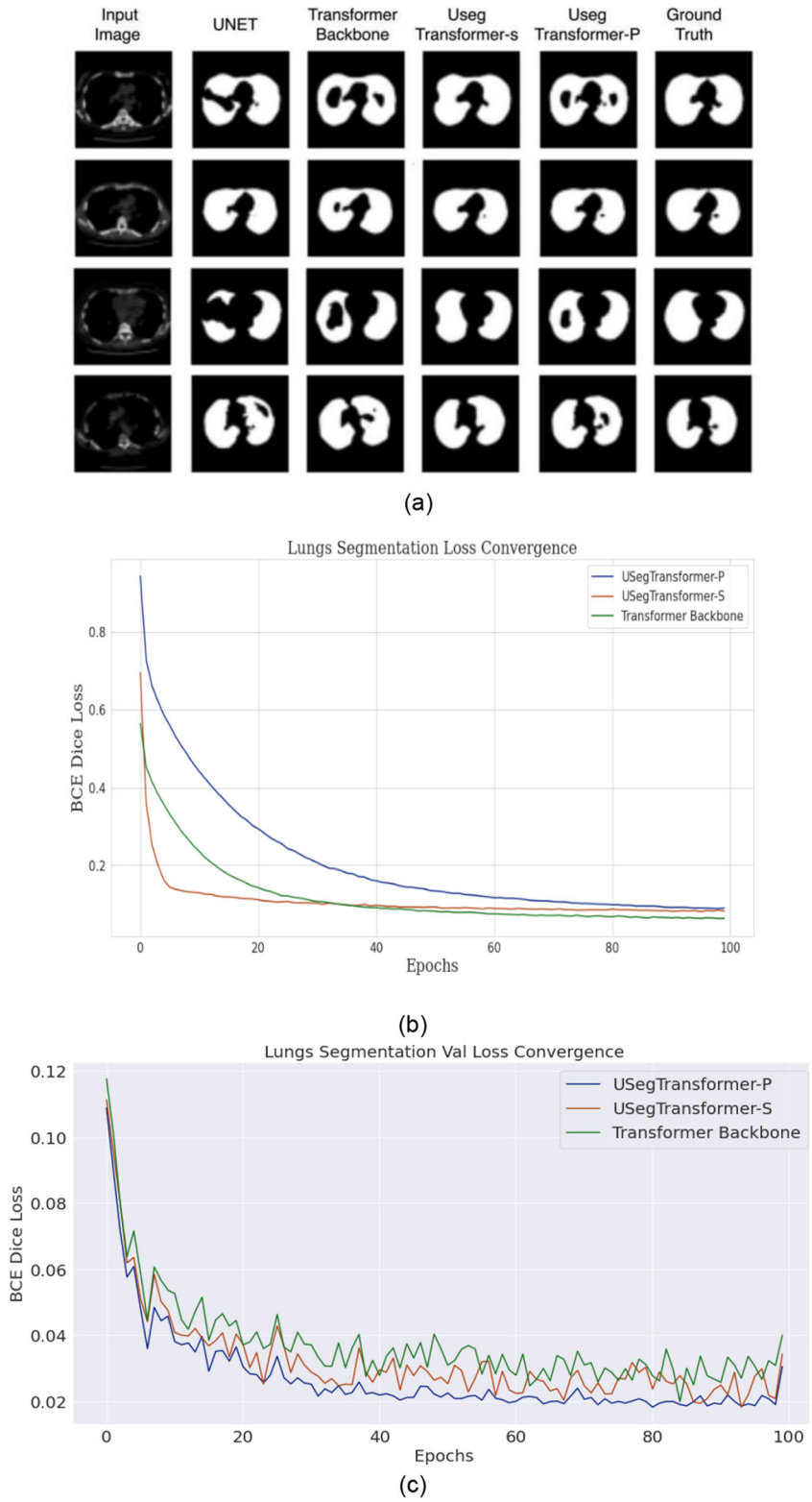

Fig. 5 The lungs segmentation (**a**) visual depiction (**b**) train loss convergence and (**c**) validation loss convergence

segmenting these consolidations will lead to better qualitative analysis of patients. Table 7 shows the quantitative analysis of the COVID-19 Consolidation mask dataset. The USegTransformer-P achieved a Dice score of 0.8295, an accuracy of 0.9981 and an IoU of 0.9685. The USegTransformer-P has achieved a Dice score of 0.6811, accuracy of 0.9919 and IoU of 0.9218 for ground class predictions. Figure 8a provides a qualitative analysis of these results.

Table 3 3-Fold Cross Validation Performance on lungs segmentation

Model	Mean Metric ± Std Deviation		
	Dice	IoU	Accuracy
USegTransformer-P	**0.9728±0.0052**	**0.9333±0.0046**	**0.9901±0.0013**
USegTransformer-S	**0.9601±0.0083**	**0.9256±0.0066**	**0.9811±0.0023**

The trend is also reiterated in the loss convergence graph in Fig. 8b.

Moreover, we performed 3-fold cross validation on the COVID-19 segmentation dataset resulting in the model to be trained and tested on the data being divided into 2:1 ratio (66% training and 33% testing split). The results in Table 8 showcase that the model upholds its high dice, IoU and accuracy metrics with very low standard deviation across the 3 folds. This proves that the proposed architecture is robust to the data split.

4 Discussions

The medical image segmentation is one of most crucial tasks in the diagnosis obtained from analyzing medical images. The current trend in automated image segmentation is to improve the performance of traditionally used fully convolution networks by optimizing skip connection, increasing the strength of convolution encoders. However, one area where very limited research has been done is making deep learning models account long range dependencies. We believe that such consideration of such long-range dependencies holds the key to a fully independent deep learning based medical segmentation system. In the proposed work, we try to account long range dependencies by virtue of transformer encoders. The two proposed models in this study utilize both, spatial and global features and transfuse them in two unique manners. The first model, the USegTransformer-P theoretically lets the image run through the transformer as well as the UNet encoder

Table 4 Performance on skin lesion segmentation

Model	Dice	IoU	Accuracy
BCDU-Net [21]	–	–	0.937
U-Net [17]	–	–	0.890
R2U-Net [38]	–	–	0.880
Attention R2U-Net [38]	–	–	0.904
Attention U-Net [40]	–	–	0.897
Transformer Backbone	0.8503	0.8511	0.9447
USegTransformer-P	**0.8701**	**0.8671**	**0.9514**
USegTransformer-S	**0.8420**	**0.8374**	**0.9431**

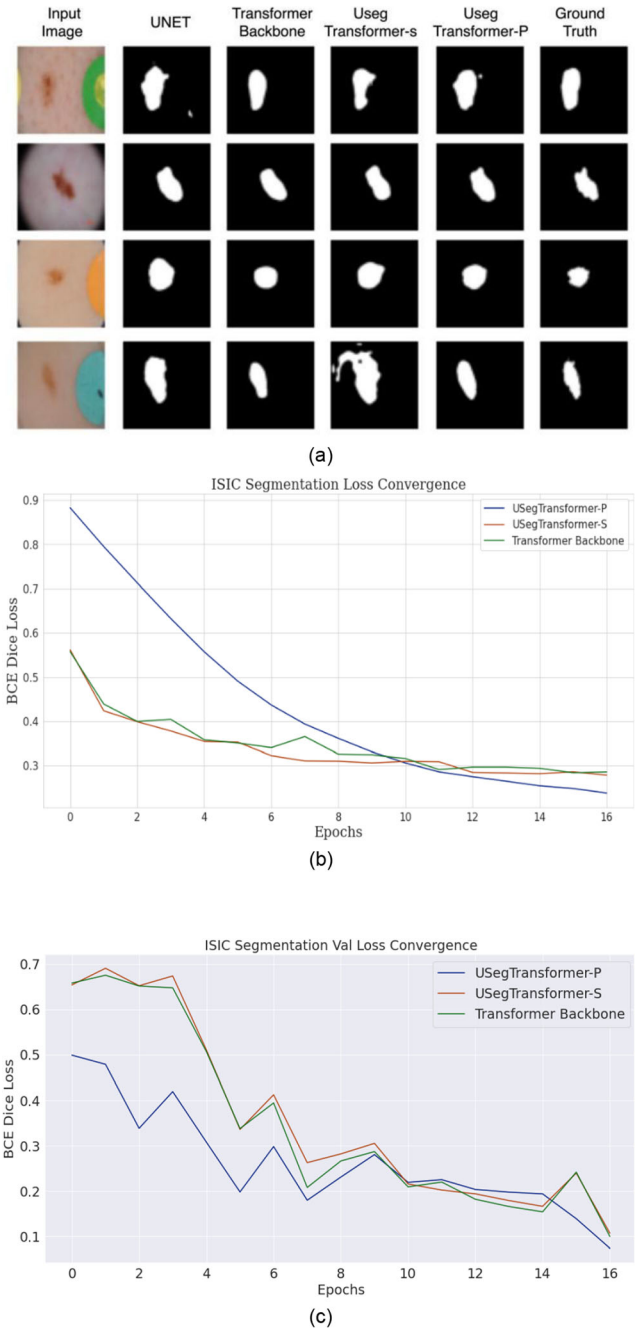

Fig. 6 The skin lesion segmentation (**a**) visual depiction, (**b**) train loss convergence and (**c**) validation loss convergence

decoders and finally combines and chooses between the local and global features through the novel ensemble decoder, from their respective feature maps. On the other hand, the second model, USegTransformer-S, first outputs a multichannel feature map considering long range features which is then used to find local features. This leads to an affective transfusion of both types of features. On analysing Fig. 9, specifically the third row, the effects that global features induce in the quality of masks becomes clearer. We can observe from Fig. 9, that the masks produced in UNET have very detailed boundaries

Table 5 Key performance indicators on nuclei segmentation

Model	Dice	IoU	Accuracy
U-Net [17]	0.7573	–	–
U-Net++ [20]	0.8974	–	–
32xU-Net/FPN (top coders) [30]*	–	–	0.6316
1xFC-FPN (Jacobkie) [30]*	–	–	0.6147
1xMask-RCNN (Deep Retina) [30]*	–	–	0.6140
Transformer Backbone	0.8780	0.8287	0.9694
USegTransformer-P	**0.9004**	**0.8470**	**0.9761**
USegTransformer-S	**0.8517**	**0.7949**	**0.9653**

*Winning teams from the original competition

while the masks produced by transformer backbone model have plain boundary. However, masks produced UNET are not able to emulate the broader shape of mask as effectively as the transformer. Such observation can be attributed to the type of features each model is using, UNET uses convolution which is able to exploit the local information and explicitly determine sharp edges present in boundaries while transformer which use attention mechanism to form correlation between

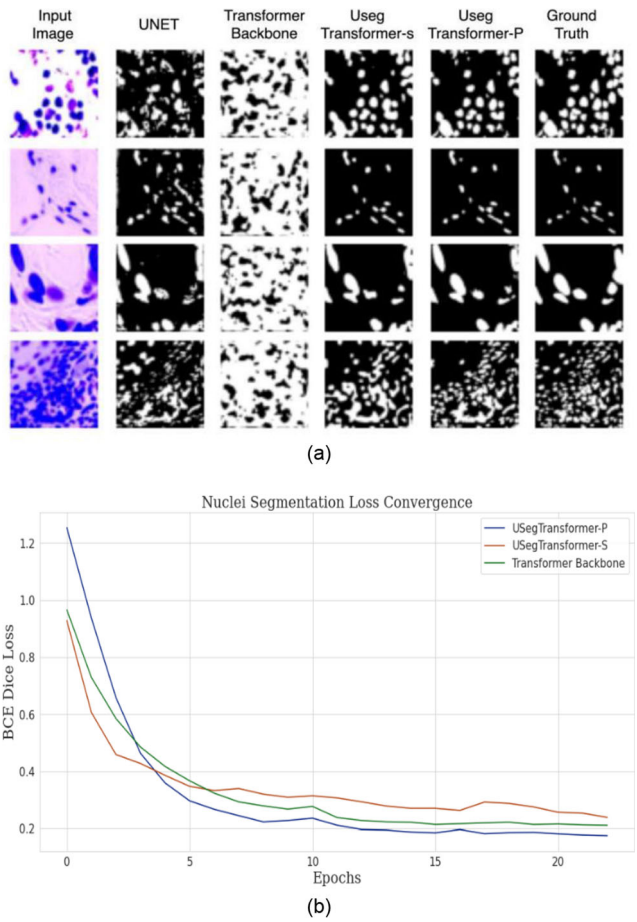

Fig. 7 The nuclei segmentation (**a**) visual depiction, and (**b**) loss convergence

Table 6 3-Fold Cross Validation Performance on nuclei segmentation

Model	Mean Metric ± Std Deviation		
	Dice	IoU	Accuracy
USegTransformer-P	**0.8887±0.0095**	**0.8351±0.0073**	**0.9703±0.0031**
USegTransformer-S	**0.8311±0.0114**	**0.7790±0.0098**	**0.9588±0.0054**

input has a large scope which leads to better understanding of general shape. The mask produced by USegTransformer-P.

An important trend, we try to set through the proposed model is to expand the use of deep learning models to segmentation for detection of COVID-19 which was earlier on limited to classification problem statements. It has been illustrated that the USegTransformer-S model which is a product of sequential stacking performed at par with the existing models and didn't perform better than that of the USegTransformer-P model. Since, the sequential model is a relatively deeper model and for the benchmark datasets, we have smaller amounts of data. Hence, we hypothesize that this model holds the capacity to work much better if it is trained on huge industrial datasets with resources much greater than ours. In the proposed research work, we have been able to propose a novel model that holds various advantages over other networks that makes the proposed models much more suitable for the task of medical image segmentation. As explained earlier, the proposed models have the capacity to outperform other previous state-of-the-art networks qualitatively and quantitatively. It has the capacity to produce segmentation masks with accurate high-level form and precise boundaries and spacings which entitles these predicts to be more trustworthy and reliable for real world applications. On the flip side, it is important to note that the USegTransformer-P and USegTransformer-S are large computationally complex models that require huge training data to function at their best capacities.

5 Conclusion

In this paper, we have proposed an end-to-end deep learning frameworks for segmenting medical images named USegTransformer-P and USegTransformer-S which are capable of aiding medical professionals as well as accelerating the

Table 7 Key Performance indicators on COVID-19 segmentation

Model	Mask Feature	Dice	IoU	Accuracy
USegTransformer-P	Consolidations	0.8295	0.9685	0.9981
USegTransformer-P	Ground Glass	0.6811	0.9218	0.9919

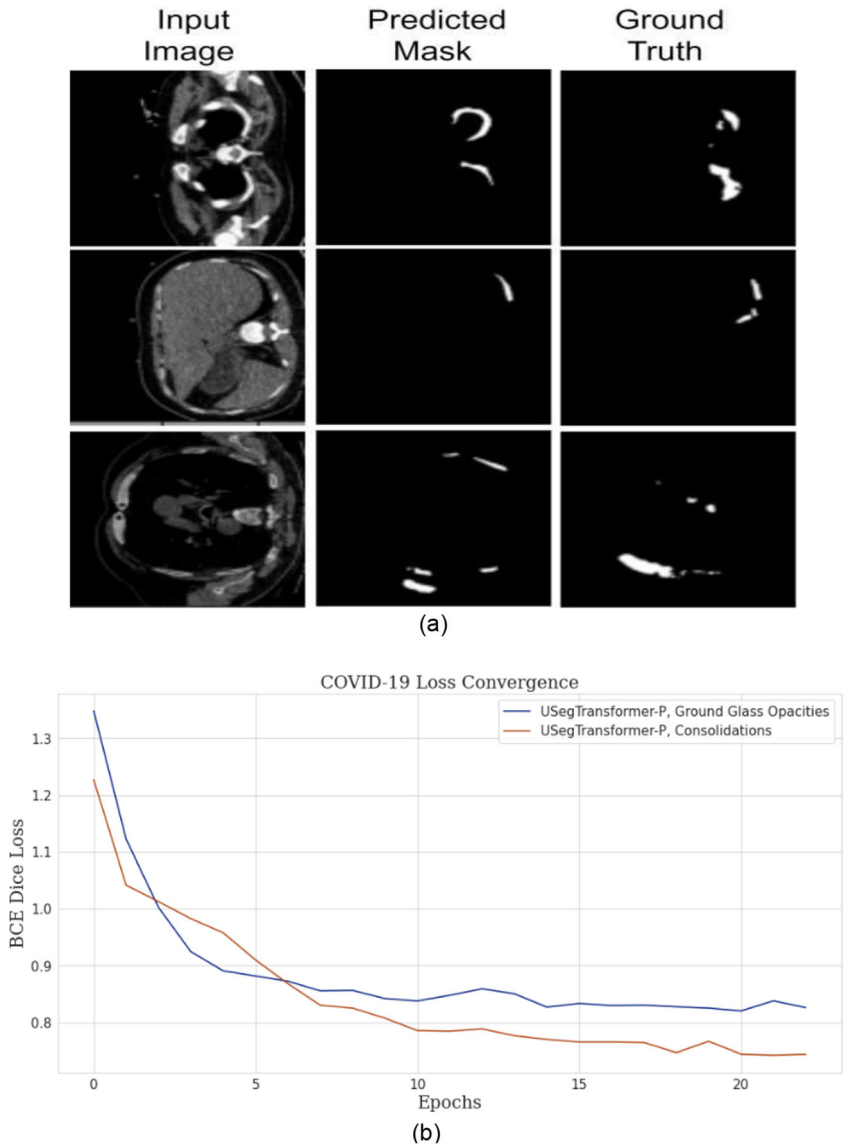

Fig. 8 The COVID-19 CT scan segmentation (**a**) visual depiction, and (**b**) loss convergence

process of medical diagnoses. Further, we illustrated the efficiency of utilizing transformer-based encoding and FCN based encoding together in the model. Also, we have presented two different techniques of stacking FCN-based segmentation models and transformer-based segmentation models. Furthermore, we have presented the efficiency of proposed model by evaluating it on varied benchmark datasets like LGG, LUNA, ISIC, and Data Science Bowl 2018 Dataset where the proposed model USegTransformer-P beat the current state of the models by achieving accuracies of 99.71%, 99.13%, 95.14% and 97.61% as well as USegTransformer-S achieved accuracies of 99.54%, 98.94%, 94.31% and 96.53%, respectively. Moreover, we have proved that the localized features obtained from FCN based networks and global context features obtained from transformer-based networks complement each other in improving a model's segmentation ability of medical datasets. By the virtue of these qualitative and quantitative

Table 8 3-Fold Cross Validation Performance on COVID-19 segmentation

Mask Feature	Mean Metric ± Std Deviation		
	Dice	IoU	Accuracy
USegTransformer-P (Consolidations)	**0.7854±0.0141**	**0.9746±0.0029**	**0.9989±0.0001**
USegTransformer-P (Ground Glass)	**0.6776±0.0159**	**0.9149±0.0131**	**0.9910±0.0014**

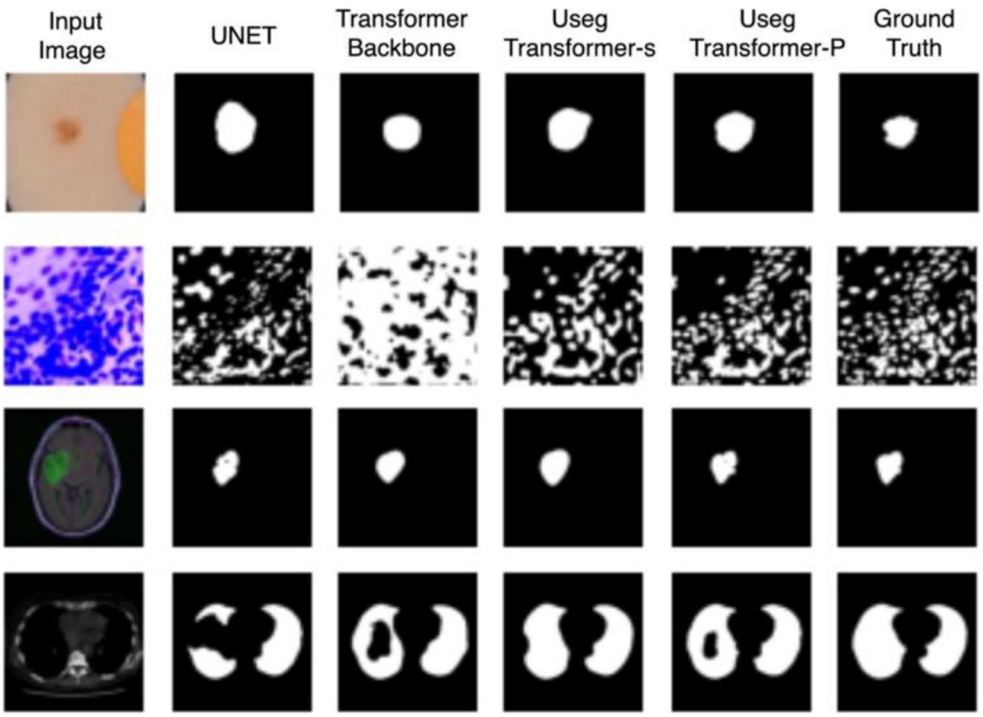

Fig. 9 Effects of global and local features

improvements, the proposed models are trustable and appropriate for real-world clinical applications. These predictions have the capacity to work in a diagnosis system to help analyze medical scans and reports to facilitate medical workers in the process of making health infrastructure accessible, available, and fruitful potential users.

In the future, experiments can be conducted on developing a more complex patching strategy in order to make the process of patching dynamic. Currently, the patching in a transformer branch is rigid and depends on manually parameterized sizes which can hold the transformer back in however more optimal patches can produce better results. In order to make the process of patching dynamic, we have to develop and conduct a more complex patching strategy. Further, an experiment could be conducted to develop an attention which is optimized for vision tasks instead of the self-attention mechanism which is a direct translation of the concept originally establish in the NLP domain that forces the transformer to look at a series of image tokens as a sentence with positional encoding. Moreover, different variants of UNet could be used to experiment with different types of feature extraction since the improvement of image segmentation is by respecting different types of features. Further, the full potential of USegTransformer-S can be analyzed by evaluating the model on more complex and large datasets.

Declarations The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

1. Ganguly D, Chakraborty S, Balitanas M, Kim TH (2010) Medical imaging: a review. Commun Comput Inf Sci 78(CCIS):504–516. https://doi.org/10.1007/978-3-642-16444-6_63
2. Cao H, Liu H, Song E, Hung CC, Ma G, Xu X, Jin R, Lu J (2020) Dual-branch residual network for lung nodule segmentation. Appl Soft Comput J 86:105934. https://doi.org/10.1016/j.asoc.2019.105934
3. Hashemzehi R, Mahdavi SJS, Kheirabadi M, Kamel SR (2020) Detection of brain tumors from MRI images base on deep learning using hybrid model CNN and NADE. Biocybern Biomed Eng 40:1225–1232
4. Garcia-Arroyo JL, Garcia-Zapirain B (2019) Segmentation of skin lesions in dermoscopy images using fuzzy classification of pixels and histogram thresholding. Comput Methods Prog Biomed 168:11–19. https://doi.org/10.1016/j.cmpb.2018.11.001
5. Nogueira-Rodríguez A, Domínguez-Carbajales R, López-Fernández H, Iglesias Á, Cubiella J, Fdez-Riverola F, Reboiro-Jato M, Glez-Peña D (2021) Deep neural networks approaches for detecting and classifying colorectal polyps. Neurocomputing 423:721–734. https://doi.org/10.1016/j.neucom.2020.02.123
6. Chlebus G, Schenk A, Moltz JH, van Ginneken B, Hahn HK, Meine H (2018) Automatic liver tumor segmentation in CT with fully convolutional neural networks and object-based postprocessing. Sci Rep 8:15497
7. Lal S, Das D, Alabhya K, Kanfade A, Kumar A, Kini J (2021) NucleiSegNet: robust deep learning architecture for the nuclei segmentation of liver cancer histopathology images. Comput Biol Med 128:104075. https://doi.org/10.1016/j.compbiomed.2020.104075
8. Sharma N, Aggarwal LM (2010) Automated medical image segmentation techniques. J Med Phys 35(1):3–14. https://doi.org/10.4103/0971-6203.58777
9. Ramesh N, Yoo JH, Sethi IK (1995) Thresholding based on histogram approximation. IEE Proc Vision, Image Signal Process 142:271. https://doi.org/10.1049/ip-vis:19952007

10. Sharma N, Ray A, Sharma S, Shukla KK, Pradhan S, Aggarwal LM (2008) Segmentation and classification of medical images using texture-primitive features: application of BAM-type artificial neural network. J Med Phys 33:119–126. https://doi.org/10.4103/0971-6203.42763
11. Gletsos M, Mougiakakou SG, Matsopoulos GK, et al (2003) A computer-aided diagnostic system to characterize CT focal liver Lesions: Design and Optimization of a Neural Network Classifier. IEEE Trans Inf Technol Biomed. https://doi.org/10.1109/TITB.2003.813793
12. Zheng X, Lei Q, Yao R, Gong Y, Yin Q (2018) Image segmentation based on adaptive K-means algorithm. Eurasip J Image Video Process 2018. https://doi.org/10.1186/s13640-018-0309-3
13. Ahmadi N (2020) A hybrid intelligent approach for image segmentation and feature extraction using fuzzy clustering, lattice boltzmann and GLDM techniques. J Soft Comput Decis Support Syst 7:1–5. https://doi.org/10.5815/ijigsp.2012.06.01
14. Bezdek JC, Ehrlich R, Full W (1984) FCM: the fuzzy c-means clustering algorithm. Comput Geosci 10:191–203. https://doi.org/10.1016/0098-3004(84)90020-7
15. Krizhevsky A, Sutskever I, Hinton GE (2017) ImageNet classification with deep convolutional neural networks. Commun ACM 60:84–90. https://doi.org/10.1145/3065386
16. Ciresan DC, Giusti A, Gambardella LM, Schmidhuber J (2012) Deep neural networks segment neuronal membranes in electron microscopy images. In: NIPS, pp 2852–2860
17. Ronneberger O, Fischer P, Brox T (2015) U-Net: convolutional networks for biomedical image segmentation. MICCAI. https://doi.org/10.48550/arXiv.1505.04597
18. Ibtehaz N, Rahman MS (2020) MultiResUNet: rethinking the U-net architecture for multimodal biomedical image segmentation. Neural Netw 121:74–87. https://doi.org/10.1016/j.neunet.2019.08.025
19. Lou A, Guan S, Loew M (2020) DC-UNet: rethinking the U-net architecture with dual channel efficient CNN for medical images segmentation. arXiv. https://doi.org/10.48550/arXiv.2006.00414
20. Zhou Z, Siddiquee MMR, Tajbakhsh N, Liang J (2020) UNet++: redesigning skip connections to exploit multiscale features in image segmentation. IEEE Trans Med Imaging 39:1856–1867. https://doi.org/10.1109/TMI.2019.2959609
21. Azad R, Asadi-Aghbolaghi M, Fathy M, Escalera S (2019) Bi-directional ConvLSTM U-net with densley connected convolutions. In: Proceedings - 2019 international conference on computer vision workshop, ICCVW 2019. https://doi.org/10.48550/arXiv.1909.00166
22. Jha D, Riegler MA, Johansen D et al (2020) DoubleU-net: a deep convolutional neural network for medical image segmentation. In: Proceedings - IEEE Symposium on Computer-Based Medical Systems. https://doi.org/10.48550/arXiv.2006.04868
23. Chen L, Bentley P, Mori K, Misawa K, Fujiwara M, Rueckert D (2018) DRINet for medical image segmentation. IEEE Trans Med Imaging 37:2453–2462. https://doi.org/10.1109/TMI.2018.2835303
24. Gu Z, Cheng J, Fu H, Zhou K, Hao H, Zhao Y, Zhang T, Gao S, Liu J (2019) CE-net: context encoder network for 2D medical image segmentation. IEEE Trans Med Imaging 38:2281–2292. https://doi.org/10.1109/TMI.2019.2903562
25. Dosovitskiy A, Beyer L, Kolesnikov A et al (2020) An image is worth 16x16 words: transformers for image recognition at scale. ICLR 2021. https://doi.org/10.48550/arXiv.2010.11929
26. Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, Kaiser Ł, Polosukhin I (2017). Attention is all you need. In: Proceedings of the 31st International Conference on Neural Information Processing Systems (NIPS'17). Curran Associates Inc., Red Hook, NY, USA, pp 6000–6010
27. Nishio M, Nagashima C, Hirabayashi S, Ohnishi A, Sasaki K, Sagawa T, Hamada M, Yamashita T (2017) Convolutional auto-encoders for image denoising of ultra-low-dose CT. Heliyon 3:e00393. https://doi.org/10.1016/j.heliyon.2017.e00393
28. Brain MRI segmentation | Kaggle (n.d.) https://www.kaggle.com/mateuszbuda/lgg-mri-segmentation
29. Finding and Measuring Lungs in CT Data | Kaggle (n.d.) https://www.kaggle.com/kmader/finding-lungs-in-ct-data/data/
30. Caicedo JC, Goodman A, Karhohs KW, Cimini BA, Ackerman J, Haghighi M, Heng CK, Becker T, Doan M, McQuin C, Rohban M, Singh S, Carpenter AE (2019) Nucleus segmentation across imaging experiments: the 2018 data science bowl. Nat Methods 16:1247–1253. https://doi.org/10.1038/s41592-019-0612-7
31. Codella NC, Rotemberg VM, Tschandl P, Celebi ME, Dusza SW, Gutman D, Helba B, Kalloo A, Liopyris K, Marchetti MA, Kittler H, Halpern AC (2019) Skin lesion analysis toward melanoma detection 2018: A challenge hosted by the international skin imaging collaboration (ISIC). ArXiv, abs/1902.03368
32. Tschandl P, Rosendahl C, Kittler H (2018) Data descriptor: the HAM10000 dataset, a large collection of multi-source dermatoscopic images of common pigmented skin lesions. Sci Data 5:180161. https://doi.org/10.1038/sdata.2018.161
33. COVID-19 - Medical segmentation (n.d.) http://medicalsegmentation.com/covid19/
34. Paszke A, Gross S, Massa F, Lerer A, Bradbury J, Chanan G, Killeen T, Lin Z, Gimelshein N, Antiga L, Desmaison S, Köpf A, Yang E, DeVito Z, Raison M, Tejani A, Chilamkurthy S, Steiner B, Fang L, Bai J, Chintala S (2019) PyTorch: An imperative style, high-performance deep learning library. NeurIPS. https://doi.org/10.48550/arXiv.1912.01703
35. Reddi SJ, Kale S, Kumar S (2018) On the convergence of Adam and beyond. In: 6th international conference on learning representations, ICLR 2018 - conference track proceedings. https://doi.org/10.48550/arXiv.1904.09237
36. Rizwan I, Haque I, Neubert J (2020) Deep learning approaches to biomedical image segmentation. Informatics Med Unlocked 18:100297/1–100297/10029711. https://doi.org/10.1016/j.imu.2020.100297
37. Billot B, Greve DN, van Leemput K et al (2020) A learning strategy for contrast-agnostic MRI segmentation.Proceedings of the Third Conference on Medical Imaging with Deep Learning 121:75-93. https://doi.org/10.48550/arXiv.2003.01995
38. Alom MZ, Yakopcic C, Hasan M, Taha TM, Asari VK (2019) Recurrent residual U-net for medical image segmentation. J Med Imaging 6:1. https://doi.org/10.1117/1.jmi.6.1.014006
39. Zhang Z, Fu H, Dai H, Shen J, Pang Y, Shao L (2019) ET-Net: A generic edge-aTtention guidance network for medical image segmentation. In: Medical Image Computing and Computer Assisted Intervention – MICCAI 2019. Springer-Verlag, Berlin, Heidelberg, pp 442–450. https://doi.org/10.1007/978-3-030-32239-7_49
40. Oktay O, Schlemper J, Folgoc LL, Lee MJ, Heinrich MP, Misawa K, Mori K, McDonagh SG, Hammerla NY, Kainz B, Glocker B, Rueckert D (2018) Attention U-Net: learning where to look for the pancreas. ArXiv, abs/1804.03999