Multimedia Tools and Applications
https://doi.org/10.1007/s11042-022-14061-x

Segmentation of shoulder muscle MRI using a new Region and Edge based Deep Auto-Encoder

Saddam Hussain Khan[1,2,3] · Asifullah Khan[1,3,4] · Yeon Soo Lee[5] · Mehdi Hassan[6] · Woong Kyo Jeong[7]

Received: 31 August 2021 / Revised: 9 March 2022 / Accepted: 9 October 2022
© The Author(s), under exclusive licence to Springer Science+Business Media, LLC, part of Springer Nature 2022

Abstract
Automatic segmentation of shoulder muscle MRI is challenging due to the high variation in muscle size, shape, texture, and spatial position of tears. Manual segmentation of tear and muscle portion is hard, time-consuming, and subjective to pathological expertise. This work proposes a new Region and Edge-based Deep Auto-Encoder (RE-DAE) for shoulder muscle MRI segmentation. The proposed RE-DAE harmoniously employs average and max-pooling operations in the Convolutional encoder and decoder blocks. The DAE's region and edge-based segmentation encourage the network to extract homogenous and anatomical information, respectively. These two concepts, systematically combined in a DAE, generate a discriminative and sparse hybrid feature space (exploiting both region homogeneity and boundaries). Moreover, the concept of static attention is exploited in the proposed RE-DAE that helps effectively learn the tear region. The performances of the proposed RE-DAE architectures have been tested using a 3D MRI shoulder muscle dataset using the hold-out cross-validation technique. The MRI data has been collected from the Korea University Anam Hospital, Seoul, South Korea. Experimental comparisons have been conducted by employing innovative custom-made and existing pre-trained CNN architectures using transfer learning and fine-tuning. Objective evaluation on the muscle datasets using the proposed SA-RE-DAE showed a dice similarity of 85.58% and 87.07%, an accuracy of 81.57% and 95.58% for tear and muscle regions, respectively. The high visual quality and the objective result suggest that the proposed SA-RE-DAE can correctly segment tear and muscle regions in shoulder muscle MRI for better clinical decisions.

Keywords MRI · Shoulder muscle · Deep Auto-Encoder · Segmentation · Transfer learning · CNN

✉ Asifullah Khan
asif@pieas.edu.pk

Extended author information available on the last page of the article

Published online: 18 October 2022

Content courtesy of Springer Nature, terms of use apply. Rights reserved.

1 Introduction

Over the last two decades, there has been a significant advancement in medical imaging technology and pathological analyses [31]. Medical image processing (MIP) plays a vital role in the diagnosis and screening of diseases [46]. In the anatomical imaging community, pixel-based segmentation of medical images generally classifies pixels into different anatomic regions [3]. These anatomic regions are bones, muscles, and blood vessels, etc. Applications of segmentation include effective diagnosis, surgery, vision boosting, and the extraction of qualitative and quantitative measures from medical datasets [15, 25]. However, the manual study of magnetic resonance imaging (MRI) based images created under different contrast or intensity can be quite tricky and time-consuming for radiologists and researchers. Consequently, computer-aided diagnostics exploit automatic image segmentation techniques, providing a guideline for image interpretation and making a quick and accurate decision [28, 8].

Nowadays, many people suffer from rotator cuff tear problems, which cause severe pain and limit normal activities like motion and strength of muscles. There are four muscles named infraspinatus, subscapularis, supraspinatus, and teres minor, which comprise the rotator cuff and cover the shoulder's inner space [11]. Injuries of muscles occur gradually due to continued computer usage, sports like chin up, cliff climbing, swimming, and some degenerative shoulder diseases [44]. In this context, consequently, the numbers of rotator cuff tear patients are increasing day by day. Therefore, efficient diagnosis and surgery of rotator cuff have become essential. However, the lack of muscle recovery after the operation brings about severe and complicated problems [45]. Simultaneously, accurate quantitative measurement methods of an anatomical structure are crucial to assess the pre-and post-operation phases of the issues.

The correct segmentation of MR images is always challenging due to a sudden change in intensity, contrast level, size, shape, and global MR appearance of the physical structure of a muscle [1]. The main difficulty is in segmenting regions with missing edges and absence of texture contrast as well in separating the region of interest (ROI) from muscles and background. To address such segmentation complexities, most of the earlier systems were built on traditional segmentation methods [18]. Moreover, the features-based machine learning (ML) approaches have been introduced to address these challenges since showed improved segmentation. The complexities of these approaches have been considered as a significant limitation for them to be deployed on large medical image data. Furthermore, automated segmentation of shoulder MR images using deep learning (DL) techniques has been scarcely applied to supraspinatus muscle [24].

The current study mainly focuses on the exploitation of the learning capabilities of DL approaches for the shoulder muscle segmentation of MR images [16, 22]. In this research work, we have developed an automated muscle segmentation system, named "Region and Edge-based Deep Auto-Encoder (RE-DAE)", based on Convolutional Neural Network (CNN). The proposed RE-DAE is developed and evaluated on the tear-related shoulder muscle dataset. The significant contributions of this research study are as follow.

1 To the best of our knowledge, this is the first study that incorporates region and edge-based pixel-wise segmentation in CNN architectures. Thus, it proposes a fully automatic (RE-DAE) tear and supraspinatus muscle segmentation architecture. The proposed RE-DAE systematically employs region and edge-based segmentation in each encoder and decoder block.
2 The static attention concept is introduced in the proposed RE-DAE to effectively learn the sparse representation of the tear region for better segmentation.

Content courtesy of Springer Nature, terms of use apply. Rights reserved.

3 Custom-made CNN models based on the concept of semantic segmentation are developed and are trained from scratch.
4 To improve the convergence of the custom-made CNN models, the idea of transfer learning (TL) is employed during the segmentation. Fine-tuning of the custom CNN architectures is performed by borrowing weights from the corresponding layers of the existing pre-trained models.

The paper is planned as follows. Section 2 highlights the background studies. Sections 3 and 4 present the proposed framework and customized semantic segmentation CNNs. Section 5 offers an experimental setup. Sections 6 and 7 describe the results and discussions, respectively. Finally, Section 8 concludes the paper.

2 Background work

Previous studies reported several classical and automatic methods [23, 10]. Generally, these traditional segmentation approaches were based on edge detection or mathematical modeling and were quite effective under normal circumstances [7, 41, 47]. However, these methods have some limitations in generating correct segmentation due to intensity inhomogeneity, higher noise levels, and uneven illumination. A classical active contour segmentation approach was employed to extract the supraspinatus muscle configuration [23]. The reported work achieved a dice similarity (DS Score) of (95%) for supraspinatus muscle using the 3D magnetic resonance (MR) muscle dataset [38]. However, the active contour method required user input by initializing region (supraspinatus muscle) and parametric values. Therefore, an automatic supervised and unsupervised segmentation based on ML approaches were reported to extract hand-crafted features [26, 30, 39]. Designing and extracting these hand-crafted features have always been the primary concern for developing a correct segmentation system. However, hand-crafted features were tested mainly on a small dataset which is considered as a significant limitation to be deployed. On the other hand, DL achieved high visual quality results compared to traditional ML techniques in the field of MIP, especially in medical MRI segmentation [42]. Recently, DL has been employed for rotator cuff shoulder muscles (deltoid, infraspinatus, supraspinatus, subscapularis) segmentation [10]. The U-Net model has been employed for shoulder muscle segmentation using the pathological and healthy OBPP muscle dataset, containing 24 examinations [32]. The U-Net achieved a DS score of 78.3% for deltoid, 82.6% for infraspinatus, 65.7% for supraspinatus, and 81.4% for subscapularis muscles. Similarly, the V16U-Net combines both VGG-16's encoder and U-Net's decoder, has been employed and enhanced DS scores for deltoid (80%), supraspinatus (67.3%), and subscapularis (81.6%). Furthermore, TL has been utilized in V16U-Net and fine-tuned on healthy and pathological shoulder muscle images. The V16pU-Net improved DS scores of 82.4%, 82.0%, 71.0% and 82.8% for deltoid, infraspinatus, supraspinatus and subscapularis muscles, respectively.

Importantly, tears of supraspinatus tendons may also occur, and degeneration of muscles limits normal activities. However, these methods have some limitations in providing accurate tear-related measurements associated with rotator cuff volume, thus achieving a better tear problem. Moreover, most of the reported work presents semi-automatic techniques and hence required user input by initializing region or prior knowledge, which is computationally

Content courtesy of Springer Nature, terms of use apply. Rights reserved.

expensive on a large set of data. Furthermore, the aforementioned methods addressed rotator cuff and supraspinatus muscle analysis. To our information, no research has been reported related to tear infected muscle segmentation using DL.

3 The proposed muscle segmentation framework

The proposed muscle segmentation framework aims to accurately and precisely segment the tear portion from the entire region of the muscle and background region. The workflow of the segmentation framework is divided into four main phases: (1) the pre-processing prepare the data necessary for training the proposed segmentation network, (2) the proposed semantic segmentation models training phase, (3) the model deployment phase is evaluated by pixel-label-based semantic segmentation, and (4) the pixel attention to the proposed segmentation model. The aforementioned phases of the proposed segmentation framework are shown in Fig. 1. The details of each phase are provided in the forthcoming sections.

3.1 Pre-processing

Pre-processing is essential in subsequent segmentation tasks, especially for medical image segmentation. Muscle MRI images predominantly suffer due to contrast or intensity variation, low illumination, clutter, and noise. Some pre-processing methods were employed to enhance the subjective quality of images and make the model training smooth and quantifiable before providing the segmentation network [14]. During the pre-processing, three main preparatory steps were employed: (1) 3D to 2D slice conversion and dataset splitting, (2) contrast enhancement of the images by using Histogram equalization to improve the visual quality, and (3) and data augmentation of the original dataset.

3.1.1 Conversion and splitting

The MRI images are in 3D format; therefore, the MRIcon software was used to convert 3D to 2D slices. Each slice represents a single MRI RGB image of size 304 × 304 × 3 pixels. The

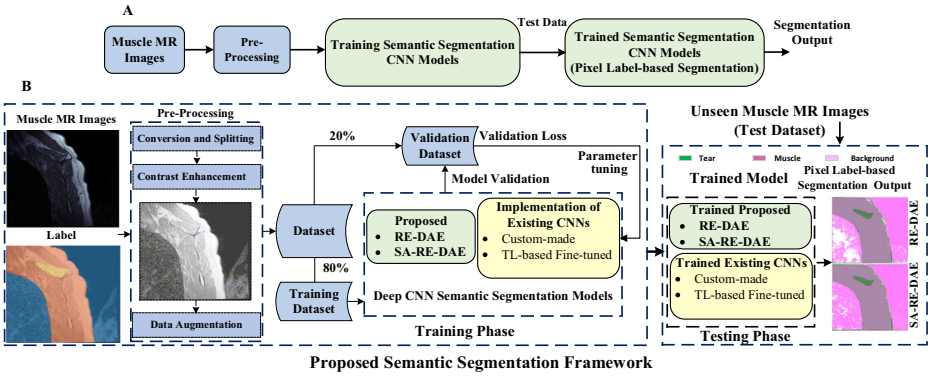


Fig. 1 **A** & **B** shows the brief and detailed **B** flow diagram of the proposed training and testing segmentation framework. The training images are preprocessed and then feed into the CNN network and their respective ground label

Content courtesy of Springer Nature, terms of use apply. Rights reserved.

dataset was partitioned into training and testing at the ratio of 8:2. The training images and their corresponding label were used for training end to end CNN networks.

3.1.2 Contrast enhancement

Histogram equalization of digital images stretches out the intensity range of an image into a broad range of intensity. Therefore, Histogram equalization has been applied in the pre-processing phase for enhancing the contrast and visual quality of an image, as shown in Fig. 1.

3.1.3 Data augmentation

There is a considerable chance of over-fitting when training a DL network on a small amount of dataset during the training phase. Therefore, data augmentation strategies were employed to increase the number of data samples. During training, the original and augmented images were combined and fed into CNN. While in on-the-fly strategy, different transformations like rotation, translation, scaling, and flipping was employed at run-time during training the CNNs. This type of data augmentation also helps in making the learner a robust one. The standard data augmentation, like a random reflection on both the X- and Y-axis, scaled factor at the range of [0.5 1], flipped along both the dimensions and rotated by ± 10 degrees.

3.2 The proposed Region and Edge-based Deep Auto-Encoder

In semantic segmentation, every image is partitioned using pixel-wise labeling, where each pixel belongs to a specific class. Generally, the semantic segmentation approaches are used to understand indoor/outdoor scenes with many classes. However, these semantic segmentation architectures also play a key role in medical MR images with a few classes. In this regard, region and edge-based deep auto-encoder (RE-DAE) semantic segmentation CNN is proposed for shoulder muscle MRI images.

The proposed RE-DAE introduces the idea of how to exploit the region and edge-based pixel-wise semantic segmentation in CNN architecture, as shown in Fig. 2. In CNN networks, Convolutional layers employ trainable filters on the input images and create feature maps that help in learning the input contents as shown in Eq. (1). The output feature maps are very sensitive to the position of the input contents. The small change in the position of the features in the input image will produce a different output. This problem might happen with shifting, scaling, flipping, and rotation of the original image. Therefore, the sub-sampling of feature maps overcomes the aforementioned problems. Sub-sampling is performed to achieve translation invariance for robust classification and the data having a small shifted variation. It is crucial to store spatial indices information while performing sub-sampling on the encoder side to achieve robustness. Based on stored indices, sub-sampling is performed in two different ways: max-pooling and average-pooling. When there is sufficient memory resource, then all the encoded features can be stored after sub-sampling.

3.2.1 Edge and region-based segmentation

The edge-based operation allows a network to learn the image structure and boundaries information like edges, and textures, making discriminative sparse features set [21]. Max-pooling is employed as an edge operation that extracts boundaries of spatial information for

Content courtesy of Springer Nature, terms of use apply. Rights reserved.

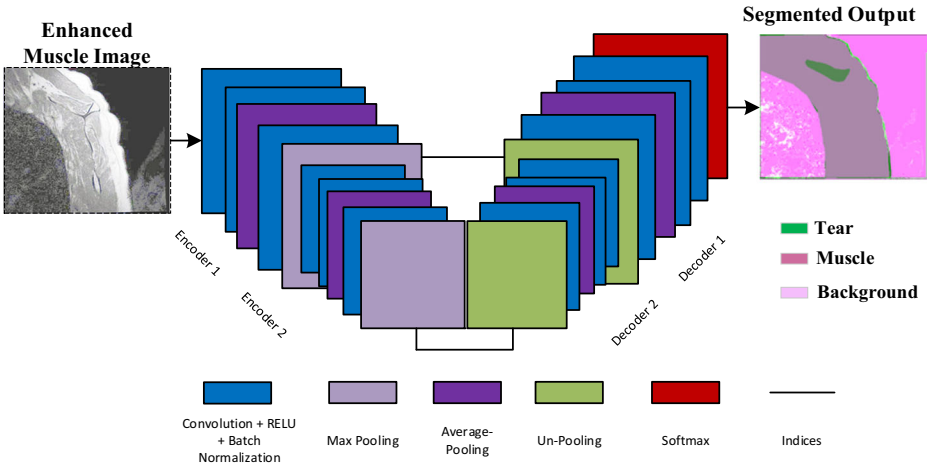


Fig. 2 The proposed (RE-DAE) CNN architecture uses two encoders and their corresponding matching decoders, with the final layer of pixel-based classification. The first encoder referred to as 'encoder1' is mapped with the first decoder named 'decoder1', and similarly, the second encoder mapped with the second decoder. Convolution layers, batch normalization, constitute the encoder network and activation function followed by $\mathbf{F}_{RE-e}$ operations as shown in Eq. 5, and as a result, robust low-resolution hybrid (smooth and boundaries) feature sets are achieved. Similarly, in the decoder convolution network, first, we up-sample (Un-pool) the low-resolution feature sets and then, convolution layer followed by batch normalization is employed. Finally, the smoothing operation is applied to produce dense hybrid features ($\mathbf{F}_{RE-d}$) as illustrates in Eq. 6, the same in spatial resolution to their corresponding input resolution

practical application as well as also stores the location indices of high discriminative features as shown in Eq. (2). The location indices are memorized in each pooling window at every encoder feature map. The pooling operation works independently on every input feature map and reduces the spatial dimension. When the stride size 'n' (non-overlapping window) is executed during the pooling operation, then the resultant feature map is down-sampled by a factor of 'n' along both width and height, thus reducing $\frac{3}{4}$n% of the activations. Conversely, up-sampling is performed by using the un-pooling layer in the decoder network. Specifically, up-sampling is performed using the memorized location indices calculated during the max-pooling operation of the corresponding encoder as illustrated in Eq. (3). This process removes the need for learning during up-sampling. The up-sampled feature maps are sparse, and when convolved with trainable filters, they will generate dense channels similar in size to the corresponding input resolution. Max-pooling is more efficient to store location indices as compared with memorizing the feature maps. Max-pooling uses less memory, which makes it suitable for practical applications. Region-based operation encourages the network to identify an object's smooth and regional homogeneity information by employing average-pooling [19, 37], as shown in Eq. (4). Smooth operator smoothens the region variations by computing the average values of each local region and suppresses the noise distortions acquired during MR image acquisition.

$$\mathbf{O}_{a,b}^{max-Down} = \sum_{i,j}^{p,q} \mathrm{f}_{p,q} \mathrm{C}_{a+p,b+q} \tag{1}$$

$$\mathbf{O}_{a,b}^{max-Down} = f_{max-Down}\left(\mathbf{O}_{a,b}\right) \tag{2}$$

Content courtesy of Springer Nature, terms of use apply. Rights reserved.

$$\mathbf{O}_{a,b}^{max-Up} = f_{max-Up}\left(\mathbf{O}_{a,b}\right) \tag{3}$$

$$\mathbf{O}_{a,b}^{avg} = f_{avg}\left(\mathbf{O}_{a,b}\right) \tag{4}$$

In Eq. (1), $\mathbf{C}$ is an input channel of size $(A \times B)$ and filter is represented by $\mathbf{f}$. The receptive field of the input channel is represented by (a, b) with respect to $\mathbf{f}$, and (p, q) shows the spatial dimension of the filter. Similarly $f_{max-Down}(.)$, $f_{max-Up}(.)$, and $f_{avg}(.)$ are max-pooling, un-sampling, and average operations, respectively. $\mathrm{O}_{a,b}^{max-Down}$, $\mathrm{O}_{a,b}^{max-Up}$, and $\mathrm{O}_{a,b}^{avg}$ show their respective outputs.

3.2.2 The systematic operation of both region and edge-based

Max-pooling extracts salient and discriminative sparse features for object segmentation and classification. In comparison, the combined operations of both the max and average-pooling extract hybrid feature space (both discriminative as well as smoothing features) to improve the segmentation performance. On the other hand, individual operations of both max and average-pooling have their drawbacks. Max-pooling extracts the maximum value within the pooling region, which leads to an unacceptable result because sometimes, most of the pooling region elements are of high intensity. Then, selecting a single high-intensity value within the pooling region may vanish the distinguishable features after max-pooling.

Conversely, average-pooling operates better on a cluster mode with a single centroid. But, its performance may be affected by data distribution that contains more than one centroid or outliers. Also, in a case where the features set have equivalent positive and negative values, the resultant activation could be small and may not provide a significant advantage. Sometimes, variance in features is not essential, and incorporating both types of pooling will produce approximately the same results. Typically, images in nature are ever-changing, and there is a high possibility that the drawback of both the pooling operations employed individually may have adverse effects on CNNs.

Therefore, in this work, the combined systematic operations of average local pooling (region) and max-pooling (edge) concepts are employed in the proposed RE-DAE as illustrated in Eq. (5). Normally, max-pooling skip the regional information, which is recovered by average-pooling. Similarly, the basic intuition behind max-pooling is that with the loss of edge features due to the average operation, the network recovers all the edge information of an object. The excellent addressing ability is that the proposed combination of the pooling layers controls the number of features to learn the CNN network and avoids overfitting [21]. Experimental results demonstrate that the proposed combination of region and edge-based operation yields information-rich feature space, superior to individually applying the region and edge-based operations. The proposed combination has two main steps: (1) preserves smooth or regional information, (2) extracts high-intensity edge and discriminative features as depicts in Eqs. (5 and 6). When choosing pooling strategies, there is always a trade-off between information and noise. Since noise can be a clue for featuring, the noise will not be suppressed.

$$\mathbf{F}_{RE-e} = f_c\left(\mathbf{O}_{x,y}^{avg} || \mathbf{O}_{x,y}^{max-Down}\right) \tag{5}$$

Content courtesy of Springer Nature, terms of use apply. Rights reserved.

$$\mathbf{F}_{RE-d} = f_c\left(\mathbf{O}_{x,y}^{max-Up} || \mathbf{O}_{x,y}^{avg}\right) \tag{6}$$

Finally, $\mathbf{F}_{RE-e}$ and $\mathbf{F}_{RE-d}$ are region and edge-based operation at encoder (e) and decoder (d) block of the proposed RE-DAE architecture as shown in Eqs. (5 and 6) and Fig. 2.

3.3 Pixel-label-based segmentation

The CNN layers are used as a training network in the proposed RE-DAE semantic segmentation architectures. CNN layers in these architectures perform dynamic feature extraction in a stepwise fashion. The trained network's final layer is based on pixel classification in the last decoder, which performs pixel label-based segmentation. The softmax as an activation function was used to identify the class probabilities for each pixel. The pixel classification layer classifies each pixel based on the maximum feature maps and the MRI pixel values into three different regions, tear, muscle, and background.

3.4 Static attention (SA)-based implementation of the proposed RE-DAE

In this study, the idea of assigning static attention (SA) to each pixel of the dominant tear region (class) is incorporated in the proposed RE-DAE architecture [17, 37]. The proposed SA-RE-DAE enhances the tear region by assigning high weightage. In comparison, the background region is suppressed by assigning a less weight to their pixels. Weighted class name pairs perform the pixel classification, and cross-entropy is used as a loss function to achieve the final segmentation output. As a result, the semantic segmentation metrics and high visual quality of correct segmented regions are computed.

4 Implementation of customized deep segmentation CNNs

The deep semantic segmentation has two different segmentation experimental setups. (1) Custom-made, and (2) TL-based fine-tuned deep CNN segmentation models. The existing CNN architectures are customized by modifying the initial and final layers to be compatible with the input feature map dimensions and output multi-class challenge (3-class). The following customized CNN models are employed for automatic muscle segmentation. Figure 3 shows the details of customized models.

4.1 Custom VGG (visual geometry Group)

The VGG-16/19 encoder have 13/16 Convolutional and 3 fully connected (FC) layers [13]. These architectures were initially designed for object classification and achieved the second position in classification at the 2014-ILSVRC competition [40]. VGG, for the first time, replaced large filter size 11 × 11, 7 × 7, and 5 × 5 with 3 × 3 and experimentally verified that a small size filter makes the receptive field effective and produces excellent results as compared with large size filters. Secondly, the use of a small size filter reduces the number of parameters and reduces low computation complexity. VGG was initially designed for object recognition and classification. However, the classification layers (FC layers) are replaced with

Content courtesy of Springer Nature, terms of use apply. Rights reserved.

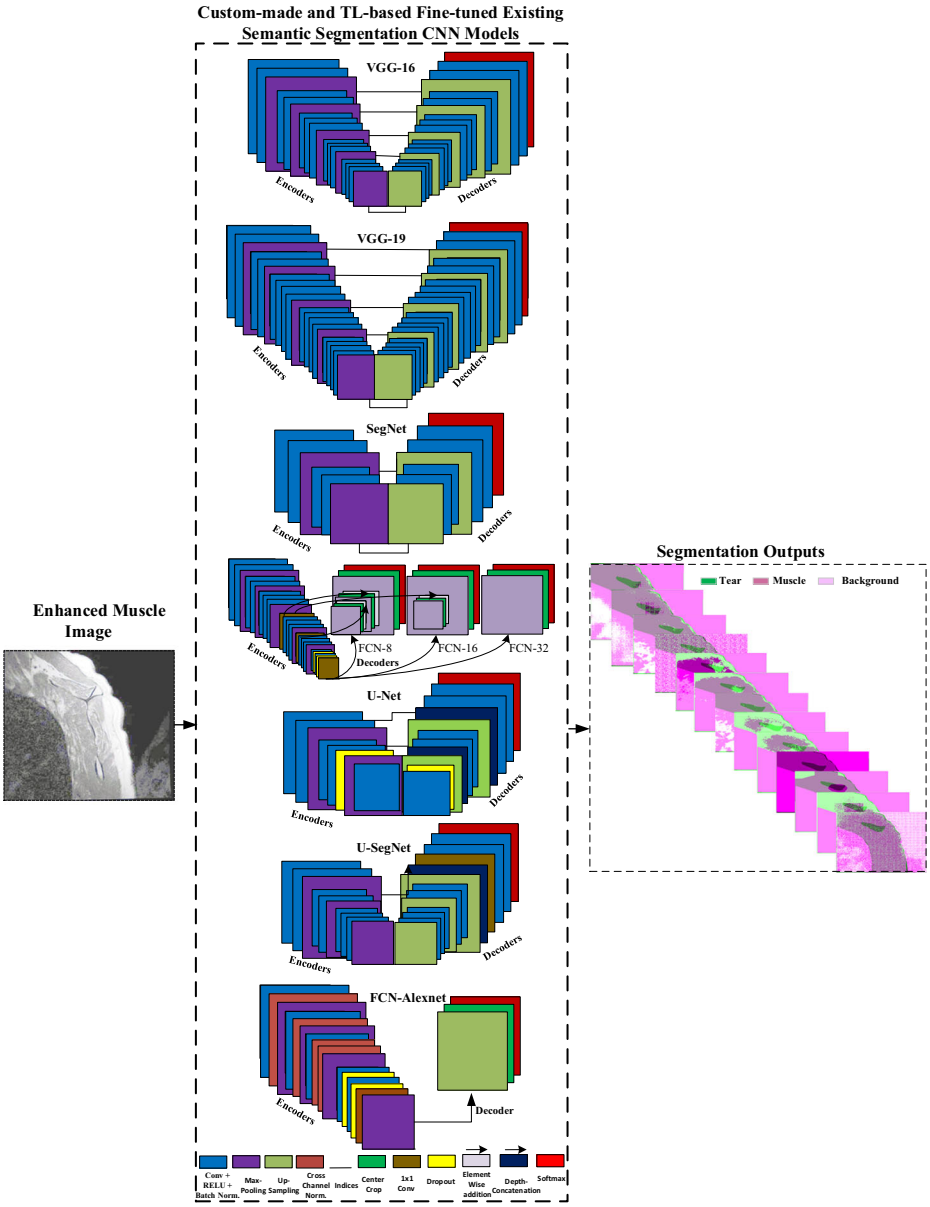


Fig. 3 Segmentation framework of the muscle MR images using customized CNN models

convolution layers at the decoder side to retain high-resolution and same dimension feature maps at the deepest corresponding input encoder for the segmentation task.

4.2 Custom SegNet

SegNet is a semantic pixel-wise segmentation CNN model [5]. SegNet has been initially proposed for outdoor scene understanding applications, specifically for an autonomous car, but

Content courtesy of Springer Nature, terms of use apply. Rights reserved.

recently has been effectively used for indoor scene understanding. SegNet has limited applications for indoor scenes, especially in the field of medical sciences. The segmentation architecture of SegNet, as shown in Fig. 3, is similar to VGG16 but has a different number of encoders and decoders. VGG has 5, while SegNet contains 2 encoders and their corresponding decoders [40]. Each encoder and decoder contains non-linear processing layers. Usually, at the end of each encoder network, non-linear down-sampling is performed using the max-pooling, and the high-frequency indices are stored. Each corresponding decoder network gets sparse low-resolution feature maps. Then, up-sampling is performed using the already stored indices calculated during the corresponding encoder's max-pooling operation. Also, the other advantages of pooling operations are: (1) retains the location information in terms of high-intensity of the segmented images, (2) Reduces the total number of trainable parameters during un-pooling at the decoder side.

4.3 Custom FCN (Fully Convolution Network)

The VGG-16 network is customized into FCN 32s, FCN 16s, and FCN 8s networks, having different up-sampling factors. FCN FC layers are replaced with fully convolution layers for object segmentation, and up-sampling is performed through transposed convolution layers. In FCN 32s at the decoder side, stride size 32 (non-overlapping window) was employed [36]. As a result, the resulting feature map was up-sampled by a factor of 32 across both width and height. FCN 32s extracts coarser features that are helpful for object classification, detection, and localization. On the other hand, FCN16s and FCN 8s extract both coarse and fine features and improve segmentation performance [6]. Both FCN16s and FCN 8sup-sample the low-level resolution contents obtained from the encoder network to get a precise output using 16 and 8 up-sampling factors, respectively.

In FCN 16s, the outcome of the network is the product of the up-sampling of two layers. These layers are up-sampled from the 4th pooling layer and (7th Convolutional layer) x 2. But in FCN 8s, the outcome is a product of the up-sampling of three layers. These layers are the up-sampling of the 3rd pooling layer, (4th pooling layer) x 2 and (7th Convolutional layer) x 4.

4.4 Custom U-Net

U-Net architecture is based on FCN and is considered one of the most popular convolution encoder-decoder architecture for biomedical applications [35, 9]. Comparing U-Net with FCN-8, there are two main differences: First, U-net is symmetric, and second, the skip connections are introduced between encoder and decoder feature map. The skip connection in segmentation architecture combines the same resolution features maps of both encoder and decoder in the up-sampling path to use both the coarse and fine features. The architecture is computationally efficient and trained end to end on a small amount of data. The encoder decreases the spatial dimension by using the down-sampling layer to extract the finely detailed features. While on the decoder side, these fine feature maps are up-sampled through transposed layers to get coarse detail for the segmentation (of the same dimensions to the corresponding input resolution map). U-Net can be directly be employed on 2D volume as well as on 3D volume data. While employed on the 3D volume data, it is computationally expensive and consumes high memory [48].

Content courtesy of Springer Nature, terms of use apply. Rights reserved.

4.5 Custom U-SegNet

U-SegNet combines both SegNet and U-Net CNN architectures, largely used for semantic segmentation [27]. Most of the contemporary reported DL based semantic segmentation architectures are computationally expensive and require many learning parameters. These architectures also require a large amount of training dataset, but the availability of extensive medical data labeled by a pathologist is generally difficult. This motivates developing a U-SegNet architecture that gives a reasonable performance for medical image segmentation, provided that the amount of available data is limited and has a smaller number of parameters.

In U-SegNet, skip connection is incorporated in SegNet inspired from U-Net architecture to concatenate both coarser and more delicate details for final pixel-wise segmentation and reduce the number of parameters by using a 1×1 Convolutional layer. Because, at the decoder side, SegNet up-samples the coarser detail by using an un-pooling layer and utilizes a smaller number of parameters and trains quickly. Similarly, U-Net, by applying transposed Convolutional layer while up-sampling, starts from the extraction of coarser features and reaches to finer level information. As compared to SegNet, U-Net uses the concatenation of multi-scale information at the decoding side, which requires many learning parameters and trains slowly compared to SegNet. The U-SegNet architecture focuses on the local information as compared with global information for the segmentation. Because U-SegNet divides images into small patches, these patches are enough to achieve adequate information for segmentation. The depth of SegNet architecture decreases to the input patch size for compatibility. U-Net skips a connection at upper layers to incorporate feature maps with fine detail and reduces the number of parameters. U-SegNet at the decoding side uses a 1×1 Convolutional layer to concatenate both coarse and fine details for segmentation. It also reduces the number of parameters for the final convolution layer.

4.6 Custom FCN-AlexNet

FCN-AlexNet is a custom-made CNN architecture based on both the FCN and Alexnet network, as shown in Fig. 3. AlexNet has been employed on ImageNet for object classification and won the ImageNet competition in 2012. Generally, Convolutional layers are used for feature extraction and maintain spatial information, while FC layers are used for feature classification. The network is made competitive for the segmentation task by replacing the FC layer of Alexnet with a corresponding Convolutional layer and up-sampling layers.

4.6.1 TL-based fine-tuned segmentation CNNs

A CNN can effectively perform when the training data is abundant and has a 1D (vector) or 2D (image) grid-like topology. When enough amount of data is not available, the training performance of CNN mostly suffers. Especially in the medical field (muscle dataset), sometimes a small amount of labeled dataset is available. Therefore, TL has been utilized to produce a satisfactory performance on a limited amount of medical datasets [2, 33, 34]. TL is one of the most popular methods in ML that allows us to build an accurate network and save time and resources. Instead of training CNN from scratch, TL methods have been used to exploits the pre-trained network. The pre-trained networks were trained on different benchmark datasets, and suitable weights and biases have been obtained by solving a similar task to the problem-specific domain.

Content courtesy of Springer Nature, terms of use apply. Rights reserved.

VGG16 and VGG19 CNNs were considered to have been already trained on the ImageNet dataset [40]. The ImageNet dataset contains 15 million images of 1000 different classes, such as a mouse, keyboard, pencil, animals, etc. The network learned prominent features due to the considerable variation in the dataset. These pre-trained networks are imported and fine-tuned on problem-specific muscle datasets to exploit the concept of TL. The Fine-tuning of a pre-trained network with TL is much faster than training a network from scratch. The pre-trained network already learns the rich feature maps, and when fine-tuned, the network learns specific features according to the problem-specific domain. If the dataset is sufficient, then TL is not so fast as compared with learning from scratch. Similarly, SegNet [13], U-Net [40], Fully Convolutional Network (FCN) [4], named as FCN-8s, FCN-16s, and FCN-32s [36], were also initialized using layers and weights from the VGG network.

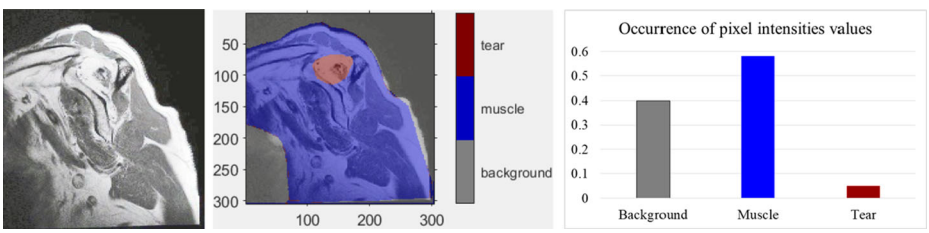
5 Experimental setup

5.1 Dataset

The dataset was obtained from the Korea University Anam Hospital, Seoul, South Korea. The dataset contains 15 patients' data, and an augmentation strategy was employed to create more samples. The dataset consists of mainly tear affected muscles. The ground truth was made through manual annotation as a pathological expert (the surgeon among the authors) provided the class knowledge. The tear affected images were partitioned into three classes named: tear, muscle, and background. Figure 4 illustrates the pixel distribution of each region of the input image. Labels assigned to these classes were 0, 1, and 2 for background, muscle, and tear, respectively. All the images are in RGB format, and the foreground region is mostly located in the center. Images were captured from different views; therefore, the tear's size and position vary from different angles. Hence, the differences in the tear's size and position made the diagnosis of the tear very challenging. In practice, the expert physician knows the direction of captured MR images.

5.2 Implementation details

Regarding the network training, the hyper-parameter selection is important to control the optimization and convergence of the network. The parameter optimization was carried out using the hold-out cross-validation strategy. The "Stochastic Gradient Descent with Momentum (SGDM)"



(a) (b) (c)

Fig. 4 **a** Shows pre-processed shoulder muscle image, **b** Color label, and **c** Pixel distribution of tear, supraspinatus muscle, and background regions, respectively

Content courtesy of Springer Nature, terms of use apply. Rights reserved.

algorithm was used for optimization. The learning rate was kept 10^{-4}, epochs = 30, the linear momentum = 0.95 and mini-batch = 2. After the selection of the hyper-parameters, CNNs were trained on the training set to get optimized weights (parameters) by reducing training error. Once the optimized parameters were selected, the validation dataset was also added to the training set.

All experiments were performed on MATLAB R2020b using Dell Core i7, 16 GB RAM. MATLAB DL toolbox was used for the implementation of segmentation. To train the CNN network is computationally expensive. Therefore, to reduce computational time, MatConvNet (a MATLAB based DL library) was used for experiments [43]. The total training time for each network took approximately 3–6 min on GPU NVIDIA GeForce GTX Titan X.

5.3 Performance evaluation

Generally, the results of segmentation networks are compared by using standard measures like accuracy. But in the case of medical segmentation, accuracy is not enough to achieve the true picture of a network's performance. Therefore, accuracy, dice similarity (DS Score), and Jaccard Coefficient (intersection over union (IOU)) were used as performance metrics to evaluate the segmentation network's true performance. The evaluation of these parameters on each tested image was calculated using the manually annotated labels.

Mathematically, accuracy is expressed in Eq. (7), which calculates the correct estimation against the total number of cases like global, local, and means accuracy. The accuracy has been considered for muscle segmentation since the number of accurate distributions of pixels to the specific class would affect the overall pixel classification result.

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{FP} + \text{FN} + \text{TN}} \tag{7}$$

$$\text{Jaccard Coefficient (IOU)} = \frac{\text{TP}}{\text{TP} + \text{FP} + \text{FN}} \tag{8}$$

$$\text{Dice Similarity (DS Score)} = \frac{2*\text{TP}}{(2*\text{TP} + \text{FP} + \text{FN})} \tag{9}$$

$$\text{Confidence Interval} = \text{z}\sqrt{\frac{\text{E}(1-\text{E})}{\text{Total Images}}} \tag{10}$$

TP, FP is the amount of true and false-positive pixels. Whereas TN, FN is the amount of true and false-negative pixels of each pixel, respectively. Equations (8) and (9) mathematically express IOU and DS Score. For comparative analysis, the images are converted into label images as ground truth. During testing, the pixel of each class was decided based on the majority voting of class labels obtained from overlapping patches. DS Score is an essential parameter for determining how the ground truth and simulated images are closely related. IOU is used to determine its close spatially matching with accurate mapping. Moreover, Confidence Interval (CI) is employed as a statistical test to evaluate the uncertainty of the segmentation models [2029]. The error (E) for DS is computed at a 95% CI for z = 1.96, in Eq. (10).

Content courtesy of Springer Nature, terms of use apply. Rights reserved.

6 Results

Images of twelve patients from muscle datasets were considered for training (including augmented images), and the rest of them were used for testing. The highest performance was achieved when pathological data were augmented to train the proposed architectures SA-RE-DAE and RE-DAE. The result achieved with a DS score: 84.67%, 85.58% for the tear and 86.30%, 87.07% for the supraspinatus muscle region by employing SA-RE-DAE and RE-DAE on test 24 images, as shown in Table 1, respectively. The accuracy for the proposed segmented architectures RE-DAE and SA-RE-DAE was 81.57%, 95.98% and 80.52%, 95.26% for the tear and supraspinatus muscle region, respectively. The overall global accuracy and Weighted IoU were 85.73% and 74.27% for the proposed SA-RE-DAE. The cluster-wise accuracy and IoU are also mentioned in Table 1. The obtained results appear satisfactory and visually distinguishable. Figure 5a contains the original muscle MR image. Figure 5b shows a pre-processed image, while Fig. 5c shows ground truth in RGB format. The segmented images using the proposed SA-RE-DAE and RE-DAE are also illustrated in Fig. 5d and e. In

Table 1 Performance analysis of the proposed and custom-made semantic segmentation models on test images

Model	Region	Objective Evaluation Metrics				
		DS±E %	Accuracy %	IOU %	Global Acc%	Weighed IOU%
SA-RE-DAE	Muscle	**87.07±13.06**	**95.98**	**77.43**	**85.73**	**74.27**
	Tear	**85.58±14.56**	**81.57**	**75.77**		
RE-DAE	Muscle	**86.30±13.84**	**95.26**	**76.79**	**84.86**	**73.33**
	Tear	**84.67±15.48**	**80.52**	**74.81**		
SegNet	Muscle	84.30±15.86	93.13	75.22	83.21	71.29
	Tear	82.96±17.21	77.71	71.98		
FCN-8	Muscle	84.21±15.95	91.92	73.29	82.11	71.19
	Tear	82.77±17.40	77.22	72.12		
U-SegNet	Muscle	84.05±16.11	94.19	73.48	81.82	68.92
	Tear	80.82±19.37	72.45	68.39		
UNet	Muscle	83.55±16.61	93.12	75.69	81.97	70.82
	Tear	81.06±19.13	74.37	70.90		
VGG-19	Muscle	82.60±17.57	92.52	70.61	80.72	67.22
	Tear	78.43±21.79	75.18	68.47		
FCN-16	Muscle	81.45±18.74	94.18	71.13	79.51	66.16
	Tear	78.92±21.29	67.90	64.69		
VGG-16	Muscle	78.21±22.01	82.37	64.90	76.01	61.89
	Tear	76.53±23.70	74.26	62.88		
FCN-Alexnet	Muscle	45.46±55.09	45.14	36.60	44.34	32.70
	Tear	40.17±60.43	43.73	34.24		
FCN-32	Muscle	41.46±59.13	35.94	26.60	34.76	24.70
	Tear	38.17±62.45	33.73	24.14		
U-Net [10]*	Muscle	65.68±34.66	---	55.70	---	---
	---	---	---	---		
V16U-Net [10]*	Muscle	67.30±33.03	---	56.98	---	---
	---	---	---	---		
V16pU-Net [10]*	Muscle	70.98±29.31	---	72.71	---	---
	---	---	---	---		

Modified U−Net [10]* results have been reported on OBPP muscle dataset [32]

Boldface texts are the results of the proposed SA-RE-DAE and RE-DAE segmentations CNNs to highlight its outperformanceM

Content courtesy of Springer Nature, terms of use apply. Rights reserved.

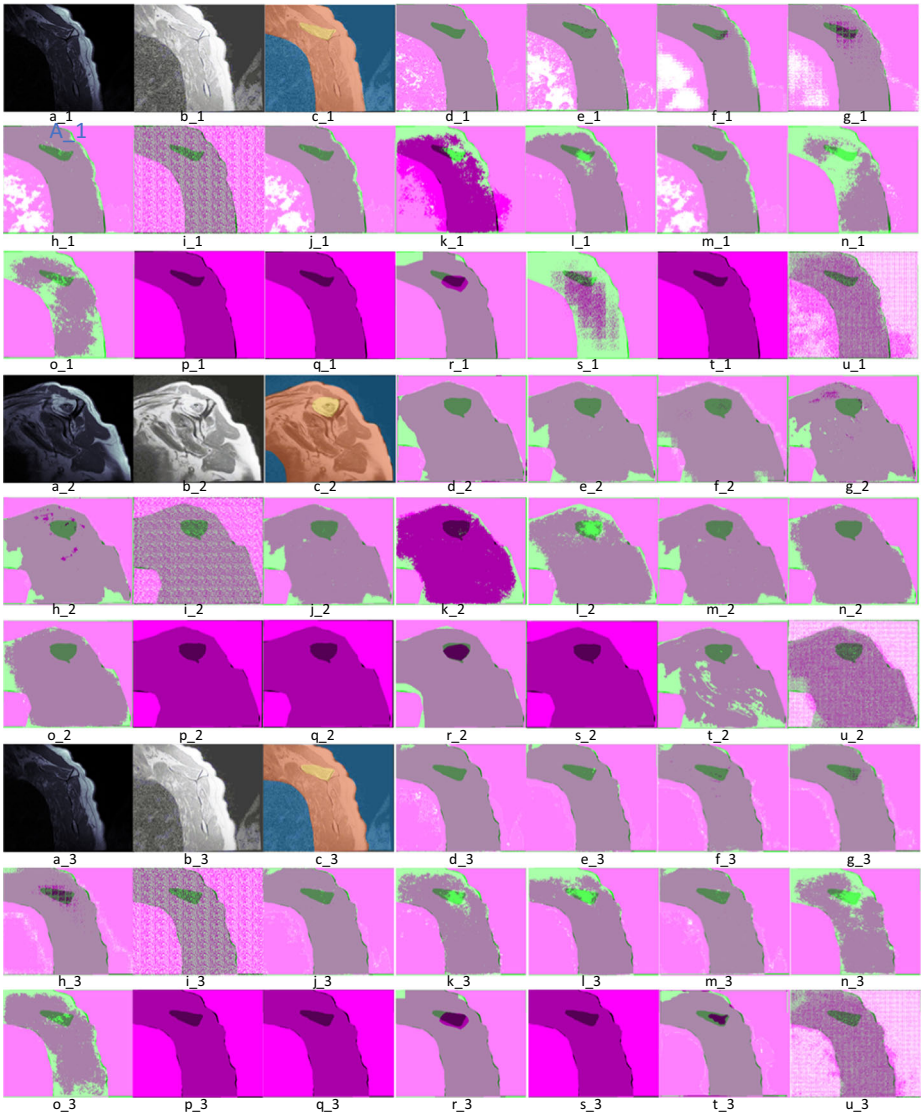


Fig. 5 **a** Shoulder muscle image, **b** Pre-processed image, **c** Color label, **d** the Proposed SA-RE-DAE, **e** the Proposed RE-DAE, **f** Pre-trained SegNet, **g** SegNet, **h** FCN-8, **i** FCN-16, **j** U-SegNet, **k** Pre-trained FCN-32, **l** FCN-Alexnet, **m** VGG19, **n** FCN-32, **o** Pre-trained FCN-Alexnet, **p** Pre-trained VGG16, **q** Pre-trained VGG19, **r** Pre-trained U-Net, **s** Pre-trained FCN-8, **t** VGG16, and **u** pre-trained FCN-16, shows segmented images. Whereas greenish, orchid, and magenta color represents tear, supraspinatus muscle, and background regions, respectively

comparison, Fig. 5 from (f) to (u) shows the segmented image of the existing CNN models. The segmented image is represented by the color map and further compared with the labeled image to achieve the accuracy, DS, and IOU score of each class, namely; tear, and muscle. Furthermore, the segmented images using the proposed architectures are also illustrated in Fig. 6.

Content courtesy of Springer Nature, terms of use apply. Rights reserved.

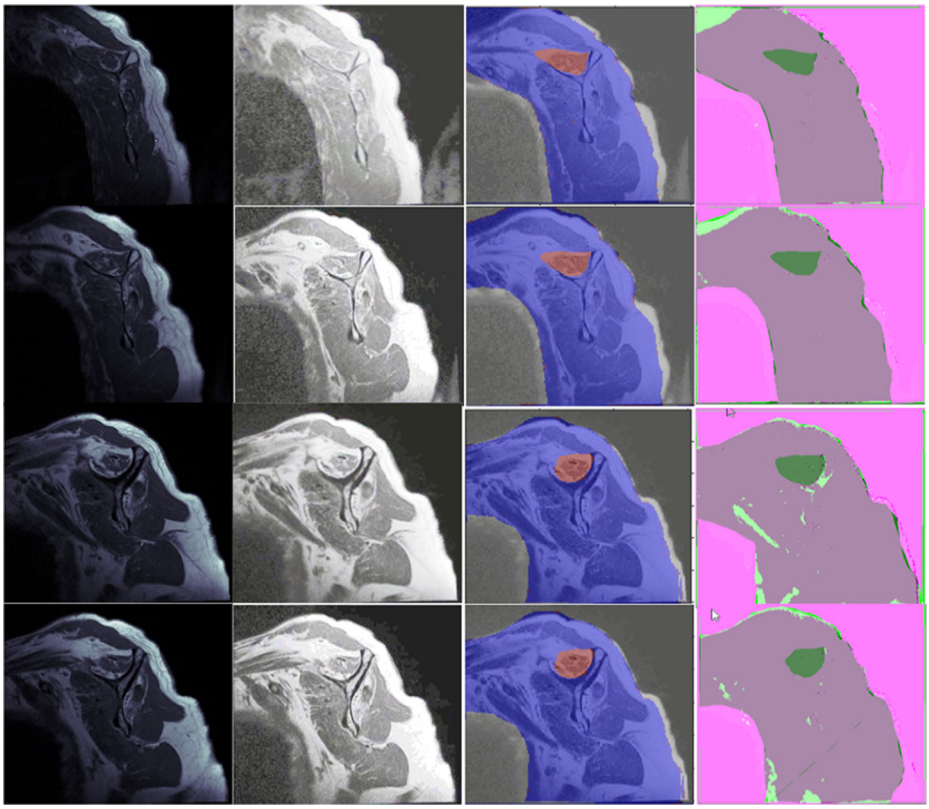


Fig. 6 **a** Shoulder Muscle Image, **b** Pre-Processed Image, **c** Color label, and **d** the Proposed SA-RE-DAE Segmented image. Whereas greenish, Orchid, and magenta color represents tear, supraspinatus muscle, and background regions, respectively

7 Discussion

Initially, training the network has a high error rate; therefore, SGD fluctuates heavily to minimize the cross-entropy loss. With the decrease of loss, SGD movement becomes stable and converges to a better solution. After reaching convergence, the training was stopped. The obtained result shows that out of the entire validation set, the network could correctly classify pixels around 96% and 85% of muscle and tear, respectively. The validation result shows that the segmentation network was fully trained on the problem-specific dataset. Additionally, the optimized parameters are obtained and now ready to perform segmentation on test images. Furthermore, the computed results of both custom and TL-based fine-tuned pre-trained semantic segmentation architectures have also been analyzed.

The maximum cluster accuracy within all the custom architectures against the maximum within all the TL-based fine-tuned pre-trained semantic segmentation architectures is (77.71% against 80.99%) for the tear region. Consequently, DS score and IOU are (82.96% against 83.52%) and (71.98% against 73.88%), respectively. Accuracy scores for a segmented supraspinatus muscle and tear region are better. Supraspinatus muscle is thin, elongated in shape and variable contrast across the patients. Therefore, it recommends that muscles with

Content courtesy of Springer Nature, terms of use apply. Rights reserved.

very strong injury (tear) may be treated separately with weighted attention, only using pathological labeled augmented data [12].

Moreover, the proposed SA-RE-DAE and RE-DAE appears to be globally suited from minor to major muscle impairments. Also, to improve the performance of the proposed architectures, the pathological and augmented data are combined and are considered to be a better strategy than TL-based fine-tuned pre-trained networks. Furthermore, exploiting annotations for the diseased shoulder muscles, the same training strategy is also employed for custom, and TL-based fine-tuned semantic architectures.

7.1 Performance comparison of the proposed architectures

We have validated the learning capacity of our proposed architectures RE-DAE and SA-RE-DAE by comparing their performance with different existing segmentation models. The proposed architecture globally outperforms existing custom, and TL-based fine-tuned the pre-trained semantic segmentation CNNs model. The detailed comparisons of all the models using the test data are illustrated in Tables 1 and 2. The outperformance of the proposed architectures is due to incorporating a hybrid feature set, both smooth and boundary information of tear and muscle region for correct segmentation. Training accuracy and loss values for the proposed RE-DAE and the second-best model (Custom SegNet) are illustrated in Fig. 7. The plot clearly shows that RE-DAE has initially good convergence as compared to other architectures. Moreover, the spatial and temporal complexity of the proposed RE-DAE is compared with existing CNN, shown in Table 3.

7.1.1 Segmentation using Static Attention (SA)-based RE-DAE

In the muscle data, the background and muscle regions dominate the tear region. This dominancy normally affects the segregation of tear region and as result reduces the performance of segmentation models. To address this issue, we incorporate a pixel attention strategy in the proposed RE-DAE. The pixel attention strategy improves the segmentation

Table 2 Performance analysis of TL-based fine-tuned pre-trained semantic segmentation models

Model	Region	Objective Evaluation Metrics				
		DS±E %	Accuracy %	IOU %	Global Acc%	Weighed IOU%
SegNet	Muscle	84.63±15.52	88.93	74.91	82.84	71.32
	Tear	83.52±16.64	80.99	73.88		
FCN-8	Muscle	84.61±15.54	92.17	74.09	82.78	71.29
	Tear	83.77±16.39	77.90	72.40		
VGG16	Muscle	83.96±16.20	94.30	74.71	82.67	71.13
	Tear	81.19±19.00	75.18	71.82		
U-Net	Muscle	83.95±16.21	92.12	74.39	82.52	71.07
	Tear	82.16±18.02	76.37	71.30		
VGG19	Muscle	83.21±16.96	92.82	70.92	81.12	68.21
	Tear	78.73±21.48	75.98	68.89		
FCN-16	Muscle	82.65±17.52	94.30	72.13	80.71	67.26
	Tear	79.29±20.92	67.90	65.49		
FCN-AlexNet	Muscle	73.57±26.69	84.01	62.88	71.71	58.79
	Tear	70.43±29.87	62.21	57.66		
FCN-32	Muscle	61.56±38.82	61.14	46.60	64.76	45.70
	Tear	58.27±42.15	53.63	44.14		

Content courtesy of Springer Nature, terms of use apply. Rights reserved.

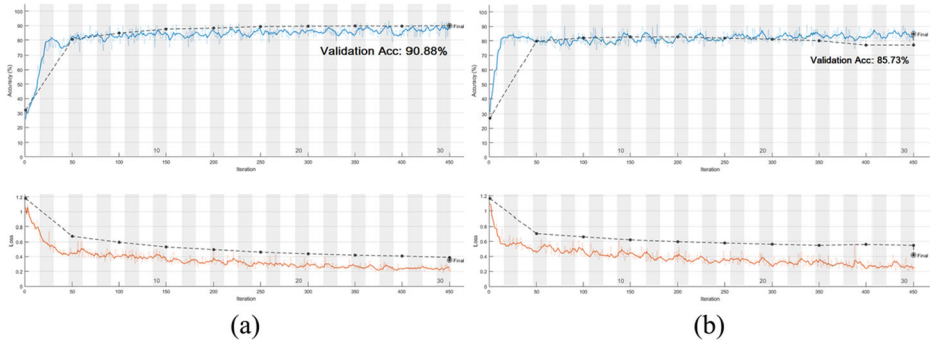


Fig. 7 Training and Validation plot of **a** RE-DAE, and **b** SegNet

performance, evident from the subjective quality (Figs. 4 and 5) and objective performance measure (Fig. 8; Table 1).

7.1.2 Performance comparison of TL-based fine-tuned and custom-made models

The performances of TL-based CNN architectures are compared with custom-made. Each pre-trained CNN model has been trained on the different problem domains like ImageNet and then fine-tuned on the problem-specific muscle dataset using TL. The TL-based fine-tuned pre-trained architectures improved the performance from 5 to 7% in terms of accuracy compared to the corresponding custom CNN models. Whereas analyzing the DS Score, the DS performance is also increased to 5–6%, as shown in Fig. 8. Both the Custom and TL-based fine-tuned pre-trained segmentation CNN has the approximately same model. But the difference is: in custom-made, the initial and final layer is customized, and no learned patterns are transferred from a pre-trained architecture. The custom models collectively performed poorly, as shown in Fig. 8, as compared to the fine-tuned pre-trained CNN models. The poor performance of custom segmentation models shows that these models have difficulty in convergence due to the depth of CNN and the small size of the training set. Custom semantic models are trained from scratch on the muscle dataset. However, it is observed that these models like VGG, SegNet, and U-Net, etc., have previously shown good performance on the availability of

Table 3 Spatial and temporal complexity of CNN models

Networks	Conv. Layers	Training (~Min.)
Proposed SA-RE-DAE	**12**	4
Proposed RE-DAE	**12**	4
FCN-Alexnet	06	3
SegNet	08	3
UNet	08	3
FCN-16	13	5
FCN-32	13	5
FCN-8	13	5
U-SegNet	15	6
VGG-16	26	8
VGG-19	32	10

Boldface texts are the results of the proposed SA-RE-DAE and RE-DAE segmentations CNNs to highlight its outperformance

Content courtesy of Springer Nature, terms of use apply. Rights reserved.

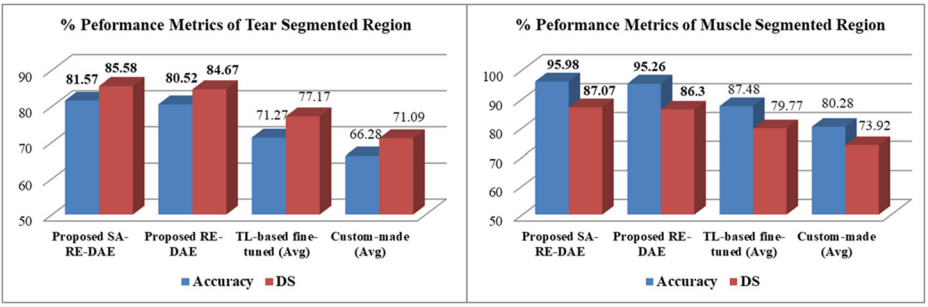


Fig. 8 Performance comparison of the proposed architectures (SA-RE-DAE, RE-DAE), TL-based fine-tuned (average) and custom-made (average) semantic segmentation models

a large amount of dataset for different domains like outdoor as well as indoor. This motivates that incorporating TL into these segmentation models might improve the segmentation performance. Figure 8; Tables 1 and 2 clearly show that TL-based fine-tuning of the pre-trained architectures achieves a little improvement in segmentation results compared to just using custom semantic architectures. One reason for this slight improvement is the different source domains (natural outdoor or indoor dataset like ImageNet) and the target domains (medical dataset of the muscle). Individually, the worst-segmented model (FCN-32) DS result is 38.17% and 41.46% for the tear and muscle region, respectively. The poor performance of FCN-32 is due to up-sampling with a high rate (up = 32) which may lose important information. Moreover, pre-processing methods also improved the accuracy of the proposed architecture and refined by an efficient data augmentation strategy.

7.2 Clinical application

The current study mainly focuses on the segmentation of tear infected shoulder muscle regions using MR images. Correct segmentation of the tear and muscle region is always challenging due to a sudden change in intensity, contrast level, size, shape, and structural appearance. These changes cause problems of missing edges and texture regions and rectifying the ROI from the background. Therefore, in this work, the proposed SA-RE-DAE extract hybrid feature space (discriminative and smoothing features) and gives substantial attention to ROI for improving the segmentation performance. The promising performance suggests that the proposed SA-RE-DAE can effectively segment and analyze MR images' tear-infected and muscle regions for better clinical decisions. The deployment of the proposed system in clinical surgery for muscle repair may be beneficial as it gives accurate objective and subjective measurements of tear and muscle regions.

8 Conclusion and future work

This research study presents an automatic shoulder muscle segmentation using a new semantic segmentation RE-DAE architecture. The regional homogeneity and boundary related important details are exploited by incorporating the concept of region and edge method, whereby it is systematically employed in the encoder and decoder block of the proposed RE-DAE. This in turn enables the proposed technique to automatically segment closely related tear regions of shoulder muscle in MR images using pixel-label-based segmentation. Additionally, the pixel

Content courtesy of Springer Nature, terms of use apply. Rights reserved.

weightage (static attention idea) has been incorporated in the tear region. We employed different custom-made and TL-based fine-tuned semantic segmentation CNN architectures for comparative analysis. The proposed segmentation SA-RE-DAE model outperforms the existing segmentation models(DS: 85.58% and 87.07%, accuracy: 81.57% and 95.58%, for tear and muscle region). In the future, the proposed region and edge concept may be employed in the existing semantic segmentation models to improve the segmentation performances. The proposed architectures have the potential to be applied to other muscle types in applications such as neuromuscular diseases and other muscle diseases. Moreover, the proposed framework will be employed on a large dataset to improve efficacy and reliability for real-time segmentation. In this regard, the training sets will be augmented by the generative adversarial network.

Acknowledgements This work was conducted with the support of the PIEAS IT endowment fund under the Pakistan Higher Education Commission (HEC). This study was also supported by the research grant of National Research Foundation of Korea (2017R1A2B2005065).As well as, we thank Pattern Recognition Lab (PR-Lab) and Pakistan Institute of Engineering, and Applied Sciences (PIEAS), for providing necessary computational resources and a healthy research environment.

Declarations

Conflict of interest Authors declared no conflict of interest.

References

1. Ahmad P, Jin H, Qamar S et al (2021) RD2A: densely connected residual networks using ASPP for brain tumor segmentation. Multimed Tools Appl 80:27069–27094
2. Ahmed U, Khan A, Khan SH et al (2019) Transfer learning and meta classification based deep churn prediction system for telecom industry. arXiv
3. Alipour N, Hasanzadeh RPR (2021) Superpixel-based brain tumor segmentation in MR images using an extended local fuzzy active contour model. Multimed Tools Appl 80:8835–8859
4. Badrinarayanan V, Mishra B, Cipolla R (2015) Understanding symmetries in deep networks
5. Badrinarayanan V, Kendall A, Cipolla R (2017) SegNet: a deep convolutional encoder-decoder architecture for image segmentation. IEEE Trans Pattern Anal Mach Intell 39:2481–2495
6. Biltz NK, Meyer GA (2017) A novel method for the quantification of fatty infiltration in skeletal muscle. Skelet Muscle 7:1
7. Bresson X, Esedoglu S, Vandergheynst P et al (2007) Fast global minimization of the active contour/snake model. J Math Imaging Vis 28:151–167
8. Chahal ES, Patel A, Gupta A et al (2021) Unet based Xception model for prostate cancer segmentation from MRI images. Multimed Tools Appl. https://doi.org/10.1007/s11042-021-11334-9
9. Çiçek Ö, Abdulkadir A, Lienkamp SS et al (2016) 3D U-net: Learning dense volumetric segmentation from sparse annotation. Lect Notes Comput Sci (including Subser Lect Notes Artif Intell Lect Notes Bioinformatics) 9901 LNCS:424–432
10. Conze P-H, Brochard S, Burdin V et al (2020) Healthy versus pathological learning transferability in shoulder muscle MRI segmentation using deep convolutional encoder-decoders. Comput Med Imaging Graph 83:101733. https://doi.org/10.1016/j.compmedimag.2020.101733
11. DEPALMA AF (1963) Surgical anatomy of the rotator cuff and the natural history of degenerative periarthritis. Surg Clin North Am 43:1507–1520
12. Devi D, Namasudra S, Kadry S (2020) A boosting-aided adaptive cluster-based undersampling approach for treatment of class imbalance problem. Int J Data Warehous Min 16:60–86. https://doi.org/10.4018/IJDWM.2020070104
13. Ioffe S, Szegedy C (2015) Batch normalization: accelerating deep network training by reducing internal covariate shift. 32nd Int. Conf. Mach. Learn. ICML 2015 1, pp 448–456
14. Javed SG, Majid A, Mirza AM, Khan A (2016) Multi-denoising based impulse noise removal from images using robust statistical features and genetic programming. Multimed Tools Appl 75:5887–5916

Content courtesy of Springer Nature, terms of use apply. Rights reserved.

15. Jiang J, Liu X, Zhang K et al (2017) Automatic diagnosis of imbalanced ophthalmic images using a cost-sensitive deep convolutional neural network. Biomed Eng Online 16:132
16. Khagi B, Kwon GR (2018) Pixel-label-based segmentation of cross-sectional brain MRI using simplified segnet architecture-based CNN. J Healthc Eng 2018:2018
17. Khan SH, Sohail A, Khan A, Lee YS (2020) Classification and region analysis of COVID-19 infection using lung. CT images and deep convolutional neural networks
18. Khan SH, Yousaf MH, Murtaza F, Velastin S (2020) Passenger detection and counting for public transport system. NED Univ J Res XVII:35–46
19. Khan SH, Sohail A, Zafar MM, Khan A (2021) Coronavirus disease analysis using chest X-ray images and a novel deep convolutional neural network. Photodiagnosis Photodyn Ther 35:102473. https://doi.org/10.1016/j.pdpdt.2021.102473
20. Khan SH, Sohail A, Khan A et al (2021) COVID-19 detection in chest X-ray images using deep boosted hybrid learning. Comput Biol Med 137:104816
21. Khan SH, Sohail A, Khan A, Lee YS (2022) COVID-19 detection in chest X-ray images using a new channel boosted CNN. Diagnostics 12:267
22. Khan A, Hussain Khan S, Saif M et al A survey of deep learning techniques for the analysis of COVID-19 and their usability for detecting omicron
23. Kim S, Lee D, Park S et al (2017) Automatic segmentation of supraspinatus from MRI by internal shape fitting and autocorrection. Comput Methods Programs Biomed 140:165–174
24. Kim JY, Ro K, You S et al (2019) Development of an automatic muscle atrophy measuring algorithm to calculate the ratio of supraspinatus in supraspinous fossa using deep learning. Comput Methods Programs Biomed 182:105063
25. Kollias D, Tagaris A, Stafylopatis A et al (2018) Deep neural architectures for prediction in healthcare. Complex Intell Syst 4:119–131
26. Kumar GA, Sridevi PV (2021) E-fuzzy feature fusion and thresholding for morphology segmentation of brain MRI modalities. Multimed Tools Appl 80:19715–19735
27. Kumar P, Nagar P, Arora C, Gupta A (2018) U-segnet: fully convolutional neural network based automated brain tissue segmentation tool. arXiv
28. Lee H, Troschel FM, Tajmir S et al (2017) Pixel-level deep segmentation: artificial intelligence quantifies muscle on computed tomography for body morphometric analysis. J Digit Imaging 30:487–498
29. Li MW, Wang YT, Geng J, Hong WC (2021) Chaos cloud quantum bat hybrid optimization algorithm. Nonlinear Dyn 103:1167–1193
30. Mandić M, Rullman E, Widholm P et al (2020) Automated assessment of regional muscle volume and hypertrophy using MRI. Sci Rep 10:2239
31. Pavel M, Jimison HB, Wactlar HD et al (2013) The role of technology and engineering models in transforming healthcare. IEEE Rev Biomed Eng 6:156–177
32. Pons C, Sheehan FT, Im HS et al (2017) Shoulder muscle atrophy and its relation to strength loss in obstetrical brachial plexus palsy. Clin Biomech 48:80–87
33. Qureshi AS, Khan A (2018) Adaptive transfer learning in deep neural networks: wind power prediction using knowledge transfer from region to region and between different task domains. arXiv
34. Qureshi AS, Khan A, Zameer A, Usman A (2017) Wind power prediction using deep neural network based meta regression and transfer learning. Appl Soft Comput J 58:742–755
35. Ronneberger O, Fischer P, Brox T (2015) U-net: Convolutional networks for biomedical image segmentation. Lect Notes Comput Sci (including Subser Lect Notes Artif Intell Lect Notes Bioinformatics) 9351:234–241
36. Shelhamer E, Long J, Darrell T (2017) Fully convolutional networks for semantic segmentation. IEEE Trans Pattern Anal Mach Intell 39:640–651
37. Schlemper J, Oktay O, Schaap M et al (2019) Attention gated networks: learning to leverage salient regions in medical images. Med Image Anal 53:197–207
38. Slabaugh MA, Friel NA, Karas V et al (2012) Interobserver and intraobserver reliability of the goutallier classification using magnetic resonance imaging. Am J Sports Med 40:1728–1734
39. Singh LK, Pooja, Garg H et al (2021) An analytical study on machine learning techniques, pp 137–157
40. Simonyan K, Zisserman A (2015) Very deep convolutional networks for large-scale image recognition. 3rd Int. Conf. Learn. Represent. ICLR 2015 - Conf. Track Proc., pp 1–14
41. Tian Y, Duan F, Zhou M, Wu Z (2013) Active contour model combining region and edge information. Mach Vis Appl 24:47–61
42. van G STJJM, L DMJ et al (2017) Deep Learning for fully-automated localization and segmentation of rectal cancer on multiparametric MR. Sci Rep 7:5301
43. Vedaldi A, Lenc K (2015) MatConvNet: Convolutional neural networks for MATLAB. MM 2015 - Proc. 2015 ACM Multimed. Conf., pp 689–692

Content courtesy of Springer Nature, terms of use apply. Rights reserved.

44. Ward AD, Hamarneh G, Ashry R, Schweitzer ME (2007) 3D shape analysis of the supraspinatus muscle. Acad Radiol 14:1229–1241
45. Ward AD, Hamarneh G, Ashry R, Schweitzer ME (2007) 3D shape analysis of the supraspinatus muscle. A clinical study of the relationship between shape and pathology. Acad Radiol 14:1229–1241
46. Zafar MM, Rauf Z, Sohail A et al (2022) Detection of tumour infiltrating lymphocytes in CD3 and CD8 stained histopathological images using a two-phase deep CNN. Photodiagnosis Photodyn Ther 37:102676. https://doi.org/10.1016/j.pdpdt.2021.102676
47. Zhang Z, Hong WC (2021) Application of variational mode decomposition and chaotic grey wolf optimizer with support vector regression for forecasting electric loads. Knowl Based Syst 228:107297
48. Zhang C, Hua Q-Q, Chu Y-Y, Wang P-W (2021) Liver tumor segmentation using 2.5D UV-Net with multi-scale convolution. Comput Biol Med 133:104424. https://doi.org/10.1016/j.compbiomed.2021.104424

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Springer Nature or its licensor holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.

Affiliations

Saddam Hussain Khan [1,2,3] · Asifullah Khan [1,3,4] · Yeon Soo Lee [5] · Mehdi Hassan [6] · Woong Kyo Jeong [7]

Saddam Hussain Khan
hengrshkhan822@gmail.com

Yeon Soo Lee
yeonsoolee@cu.ac.kr

Mehdi Hassan
mehdi.hassan5@gmail.com

Woong Kyo Jeong
drshoulder@korea.ac.kr

1 Pattern Recognition Lab, Department of Computer & Information Sciences, Pakistan Institute of Engineering & Applied Sciences, Nilore, Islamabad 45650, Pakistan

2 Department of Computer Systems Engineering, University of Engineering and Applied Sciences (UEAS), Swat 19060, Pakistan

3 PIEAS Artificial Intelligence Center (PAIC), Pakistan Institute of Engineering & Applied Sciences, Nilore, Islamabad 45650, Pakistan

4 Center for Mathematical Sciences, Pakistan Institute of Engineering & Applied Sciences, Nilore, Islamabad 45650, Pakistan

5 Department of Biomedical Engineering, College of Medical Science, Catholic University of Daegu, Daegu, South Korea

6 Department of Computer Science, Air University, Islamabad, Pakistan

7 Department of Orthopaedic Surgery, College of Medicine, Korea University, Seoul, Korea

Content courtesy of Springer Nature, terms of use apply. Rights reserved.

Terms and Conditions

Springer Nature journal content, brought to you courtesy of Springer Nature Customer Service Center GmbH ("Springer Nature").

Springer Nature supports a reasonable amount of sharing of research papers by authors, subscribers and authorised users ("Users"), for small-scale personal, non-commercial use provided that all copyright, trade and service marks and other proprietary notices are maintained. By accessing, sharing, receiving or otherwise using the Springer Nature journal content you agree to these terms of use ("Terms"). For these purposes, Springer Nature considers academic use (by researchers and students) to be non-commercial.

These Terms are supplementary and will apply in addition to any applicable website terms and conditions, a relevant site licence or a personal subscription. These Terms will prevail over any conflict or ambiguity with regards to the relevant terms, a site licence or a personal subscription (to the extent of the conflict or ambiguity only). For Creative Commons-licensed articles, the terms of the Creative Commons license used will apply.

We collect and use personal data to provide access to the Springer Nature journal content. We may also use these personal data internally within ResearchGate and Springer Nature and as agreed share it, in an anonymised way, for purposes of tracking, analysis and reporting. We will not otherwise disclose your personal data outside the ResearchGate or the Springer Nature group of companies unless we have your permission as detailed in the Privacy Policy.

While Users may use the Springer Nature journal content for small scale, personal non-commercial use, it is important to note that Users may not:

1. use such content for the purpose of providing other users with access on a regular or large scale basis or as a means to circumvent access control;
2. use such content where to do so would be considered a criminal or statutory offence in any jurisdiction, or gives rise to civil liability, or is otherwise unlawful;
3. falsely or misleadingly imply or suggest endorsement, approval , sponsorship, or association unless explicitly agreed to by Springer Nature in writing;
4. use bots or other automated methods to access the content or redirect messages
5. override any security feature or exclusionary protocol; or
6. share the content in order to create substitute for Springer Nature products or services or a systematic database of Springer Nature journal content.

In line with the restriction against commercial use, Springer Nature does not permit the creation of a product or service that creates revenue, royalties, rent or income from our content or its inclusion as part of a paid for service or for other commercial gain. Springer Nature journal content cannot be used for inter-library loans and librarians may not upload Springer Nature journal content on a large scale into their, or any other, institutional repository.

These terms of use are reviewed regularly and may be amended at any time. Springer Nature is not obligated to publish any information or content on this website and may remove it or features or functionality at our sole discretion, at any time with or without notice. Springer Nature may revoke this licence to you at any time and remove access to any copies of the Springer Nature journal content which have been saved.

To the fullest extent permitted by law, Springer Nature makes no warranties, representations or guarantees to Users, either express or implied with respect to the Springer nature journal content and all parties disclaim and waive any implied warranties or warranties imposed by law, including merchantability or fitness for any particular purpose.

Please note that these rights do not automatically extend to content, data or other material published by Springer Nature that may be licensed from third parties.

If you would like to use or distribute our Springer Nature journal content to a wider audience or on a regular basis or in any other manner not expressly permitted by these Terms, please contact Springer Nature at

onlineservice@springernature.com