SegRELU: A Parameter-Free Activation Function for Deep Neural Network used in Medical Image Segmentation

Hong-Seng Gan[1] *, Viet-Vu Vu[2], Doan-Vinh Tran[3], Viet-Thang Vu[4], Akinobu Shimizu[5], Muhammad Hanif Ramlee[6], and Riries Rulaningtyas[7]

[1] Department of Data Science, Universiti Malaysia Kelantan, 16100 UMK City Campus, Pengkalan Chepa, Kelantan, Malaysia; ORCID: 0000-0003-3777-3640

[2] VNU Information Technology Institute, Vietnam National University, Hanoi, Vietnam; ORCID: 0000-0001-5579-8783

[3] University of Education, Vietnam National University, Hanoi, Vietnam; ORCID: 0000-0002-2357-0108

[4] Hanoi University of Industry, Hanoi, Vietnam; ORCID: 0000-0002-8830-9023

[5] Institute of Engineering, Tokyo University of Agriculture and Technology, 2-24-16, Naka-cho, Koganei, Tokyo 184-0012, Japan; ORCID: 0000-0002-2719-5923

[6] Bone Biomechanics Laboratory (BBL), School of Biomedical Engineering and Health Sciences, Faculty of Engineering, Universiti Teknologi Malaysia, 81310 Skudai, Johor, Malaysia; ORCID: 0000-0003-2705-8379

[7] Biomedical Engineering Study Program, Department of Physics, Faculty of Science and Technology, Universitas Airlangga, 60115 Surabaya, Indonesia; ORCID: 0000-0001-7058-1566

*****Corresponding author:

Email	:	hongseng1008@gmail.com
Corresponding Address	:	Department of Data Science, Universiti Malaysia Kelantan, 16100 UMK City Campus, Pengkalan Chepa, Kelantan, Malaysia
Tel	:	+6017-655-1561

Abstract

Medical images are characterized by varying texture, geometry, and pathological features. As a result, medical image segmentation remains a daunting task until today. Deep learning models depend on layer-by-layer convolutional feature representation learning coordinated by forward and backward propagation. An activation function guarantees the robustness of model learning. However, existing activation functions are either ineffective in tackling vanishing gradient problem or burdened with multiple parameters that require manual tuning. The limitations lead to uncertainty when visualizing the image data. Furthermore, existing research on activation function mainly concentrates on classification task by using natural images from MNIST, CIFAR-10 and CIFAR-100 datasets. Therefore, we propose a novel activation function, SegRELU for medical image segmentation. For positive inputs, SegRELU utilizes the nearly linear property to produce outputs without vanishing gradient. The negative inputs are converged polynomially towards its asymptotes. Hence, the function assures stable training performance. We tested the function on cell and knee datasets. SegRELU has reported the best segmentation accuracies (cell: 0.9604±0.0197, knee: 0.9399±0.0187). Its' easy-to-implement advantage is a coveted feature in hyperparameters-laden deep learning models. Future works should affirm the effectiveness of SegRELU on other tasks and other types of dataset with abnormality grades.

Keywords: Activation Functions; Medical Image Segmentation; Deep Learning; U-Net

Declaration of interest: None

Acknowledgement: This research has done under the research project QG.21.58 "(Researching and Developing Clustering Integrating Constraints and Deep Learning Algorithms)" of Vietnam National University, Hanoi and UMK Matching Grant (project title: Semi Supervised Spatial Graph Deep Learning Model for Medical Image Segmentation, grant no: UMK.B01.08.01.800-2/14(39) provided by Universiti Malaysia Kelantan.

1. Introduction

Modern medical image dataset is rich of texture, anatomical and pathological information. Therefore, medical image segmentation plays an increasingly crucial role in producing accurate segmented objects for high-quality medical image analysis. While segmentation is applied on various tasks such as brain [1], lung [2], knee [3] and cell [4], medical image segmentation remains a daunting task until today. For instance, the characteristics of medical image often distinctly vary according to type of imaging modality and different anatomical topology. Besides, most medical images suffer from inadequate contrast and brightness of certain extend, which contribute to low contrast and weak boundary problems.

Classical machine learning in medical image segmentation involves feature extraction and label classification via supervised or unsupervised learning approach. While most have reported good performance, those models are mainly restricted to small size dataset [5]. In recent years, deep learning [6] has demonstrated good accuracies in large scale image datasets. Among these models, U-net [7] is the state-of-art for medical image segmentation. Its' architecture consists of a stack of convolutions, activations, and pooling layers, which translates the input vector by applying linear affine transformation followed by a nonlinear mapping. Lastly, high level abstractions are extracted by fully connected layers.

As the model goes deeper, effective feature representation learning is governed by forward and backward propagations. An activation function is a key component to introduce nonlinearity within deep neural network to improve generalizability of training data. Multiple weights are adjusted during the training process. Nonetheless, modern medical image is associated with uncertainty when visualizing the data. The outcome has huge impact on the decision-making process in clinical daily routine [8]. As a result, an activation function that can express the uncertainty to improve the segmentation result is highly coveted. Majority of existing activation function studies, however, have concentrated on classification tasks by using MNIST, CIFAR-10 and CIFAR-100 datasets.

Herewith, we proposed a hybrid activation function named SegRELU suitable for medical image segmentation and examined on 2 different medical datasets of different anatomical complexities. The contributions of this paper are elaborated below:

1. SegRELU is parameter-free and monotonic. It is easy to implement within the segmentation framework, and the performance is not influenced by handcrafted parameter tuning.
2. A comprehensive performance analysis of SegRELU against existing activation functions in medical image segmentation.

The rest of this paper is organized as follows. We explain relevant works in Section 2, the materials and methods in Section 3 and present the results and discussions in Section 4. Lastly, we conclude the paper in Section 5.

2. Related Works

Careful design of activation function is an active area in deep learning research. Numerous descriptive [9, 10] and quantitative [11-14] performance analysis on activation function have been conducted. Sigmoid function [15] and Hyperbolic Tangent (Tanh) function [16] are two of the earliest continuous nonlinear activation functions. Sigmoid function has steep gradient between the range of $2 \leq x \leq -2$ to produce clear outputs. Nevertheless, it is more suitable for shallow neural networks. Sigmoid function also suffers from vanishing gradient problem because the change of output values at both ends is very minimal. Hence, the model fails to learn effectively and has reported non-zero centrality issue.

Tanh function is a smoother zero-centered function ranged between -1 and 1. The function maps negative inputs to negative outputs and vice versa to give better training performance. It is symmetric about the origin, of which the outputs are more likely on average closer to zero to aid the backpropagation process. Unfortunately, Tanh function also struggles to overcome the vanishing gradient issue faced by sigmoid function. New functions such as Softplus [17]; Rectified Linear Unit (RELU) [18], Leaky RELU (LRELU) [19], Parametric RELU (PRELU) [20]; Exponential Linear Unit (ELU) [21], Scaled ELU (SELU) [22]; Gaussian Error Linear Unit (GELU) [23]; SWISH [24] and MISH [25] have been proposed to replace these traditional functions.

Currently, RELU is the most widely applied activation function in deep learning applications. The function is simple to implement with gradient descent methods. Derivatives of RELU is a constant, which can guarantee fast

computation. Nonetheless, RELU is infamous for dying neuron problem when its input is negative and subsequent output is always zero [26]. By modifying the negative part of RELU, the resultant function becomes robust to weight update during entire backpropagation process. One major challenge, however, is the tuning of parameter in the modified part.

Previously, Manessi & Rozza have examined the combination of different basic activation functions i.e. linear function, RELU and Tanh on classification task. Fashion-MNIST, CIFAR-10 and ILSVRG-2012 datasets were used. The study found out combined activation functions usually outperform their corresponding basic activation function [27]. Some researchers have proposed hybrid activation function architectures. Li et al. have combined RELU with Tanh activation function. The proposed structure [28] leverages on Tanh function to introduce the information discarded by RELU units. Meanwhile, Yu et al. have modified the formulation of RELU by incorporating memristive window function. The resultant activation function preserves the characteristics of RELU and SWISH [29].

3. Materials and Methods

3.1 Image Datasets

Two large scale medical image datasets were used. The first dataset comprised of 2D microscopy images of cell obtained from the Kaggle Data Science Bowl Challenge. The dataset contains of a large number of segmented nuclei images acquired under a variety of conditions and diverse in the cell type, magnification, and imaging modality (brightfield vs. fluorescence). Second dataset consisted of normal 3D MR image of knee obtained from the Osteoarthritis Initiative (OAI). The images were acquired by using 3.0 Tesla (T) MRI Scanner (Siemens Magnetom Trio, Erlangen, Germany) with quadrature transmit receive knee coil (USA Instruments, Aurora, OH). We employed the dual echo steady state (DESS) sequence with water excitation (WE). All knee image datasets used in this experiment were chosen randomly. The DESS knee images have section thickness of 0.7mm and an in-plane resolution of 0.365×0.365 mm^2 (field of view = 140×140 mm, flip angle = 25^o, TR/TE = 16.3/4.7 msec, matrix size = 384×384, bandwidth = 185 Hz/pixel).

3.2 Overview of Model Implementation

Knee segmentation [30] plays an essential role in searching for pertinent imaging based-biomarkers. In clinical-related applications, segmented knee model is used to quantify the progression pattern of cartilage degradation and attain early detection of osteoarthritis (OA). Meanwhile, cell segmentation [31] helps in identifying indicative cellular phenotypic features that can reflect the cell's physiological state. The information provides insights into oncology research. We have computed the SegRELU along with 6 basic activation functions i.e. Sigmoid, RELU, LRELU, ELU, SWISH and MISH, and 2 multi activation functions i.e. RELU-tanh and RMAF and then trained on the state-of-art U-net model [7]. The knee segmentation was reconstructed into 3D result. Fig. 1 illustrates the overall flow of deep learning segmentation.

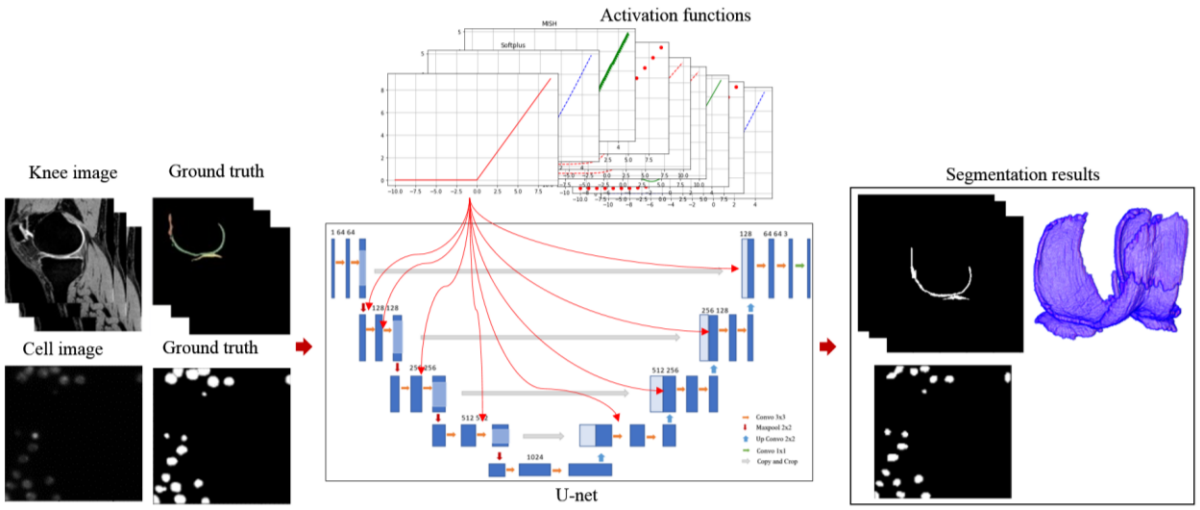


Fig. 1: Implementation of U-net model with different activation functions to perform medical image segmentation.

3.3 U-Net Model

We have implemented the U-net model with 23-layers for knee and cell segmentation. The U-net architecture consists of a contracting path and an expansive path. The contracting path comprises of numerous convolution layers, each followed by activation function and a max pooling layer with stride of two. Then, the expansive path would halve the feature channels via upsampling operators and propagate feature and spatial information to higher resolution layers. A total of 650 cell images and 5,120 knee images were used to train the U-net model independently. During the training of knee segmentation model, we have applied traditional augmentation via flipping and rotation to produce 5,120 knee images. The augmentation aims to improve the robustness of U-net model.

The U-net architecture encompasses of 10 convolutional layers where a 3×3 kernel was used at each of the downsampling. The kernel would double the feature channels and reduce the spatial information. Bottleneck layer connects in between the contracting path and expansive path and helps in transferring feature maps from contracting path to expansive path to allow precise location of objects and improve the flow of gradient during model training. In the upsampling, each input coming from bottleneck layer is up sampled by 2×2 by stride of two, followed by two 3×3 convolution layers. The last convolution layer of size 1×1 would map the 64-component feature vector to the desired number of classes. The U-net architecture is illustrated in Fig. 2.

Down-Convolution Block		Up-Convolution Block
Conv 3x3, Activation function Conv 3x3, Activation function		Up-Sample layer 2x2
Maxpool layer 2x2	Skip layer (connects every two layers of Down-Convolution block to Up-convolution block)	Conv 3x3, Activation function Conv 3x3, Activation function
Conv 3x3, Activation function Conv 3x3, Activation function		Up-Sample layer 2x2
Maxpool layer 2x2		Conv 3x3, Activation function Conv 3x3, Activation function
Conv 3x3, Activation function Conv 3x3, Activation function		Up-Sample layer 2x2
Maxpool layer 2x2		Conv 3x3, Activation function Conv 3x3, Activation function
Conv 3x3, Activation function Conv 3x3, Activation function		Up-Sample layer 2x2
Maxpool layer 2x2		Conv 3x3, Activation function Conv 3x3, Activation function
Conv 3x3, Activation function Conv 3x3, Activation function		Up-Sample layer 2x2
		Conv 3x3, Activation function Conv 3x3, Activation function
		Conv 1x1, Softmax

Fig 2: Architecture table of U-net model.

3.4 Activation Function

An artificial neural network is propelled by the forward and backward propagation. Given that $x = \{x_1, x_2, \dots, x_n\}$ are inputs to a neuron unit with weights $w = \{w_1, w_2, \dots, w_n\}$ and bias, b, the output is denoted as y. A neuron unit within the greater artificial neural networks is illustrated in Fig. 3.

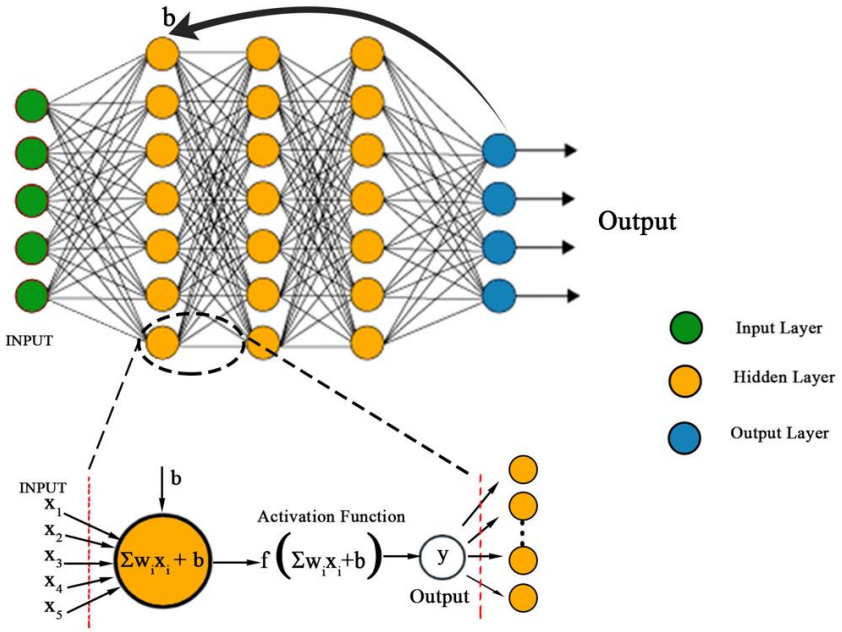


Fig. 3: A forward propagation occurs within a neuron unit. At input layer, all data is fed into the neuron unit. At hidden layer, the total input to each neuron is computed as a sum of weights and bias, $\sum w_i x_i + b$. Then, an activation function $f(\cdot)$ is applied to get the output y.

From a number of N predictions, mean squared error E is computed from the value difference between the predicted network output y_p and their corresponding targeted values t_p as shown in

$$E = \frac{1}{2}\sum_{N}(t_p - y_p)^2 \tag{1}$$

Learning is attained via adjusting the weights so y_p is as close as possible to the corresponding t_p. Thus, backpropagation refers to gradient-based optimization methods to calculate the mean squared error and adjust the weights $w^l, l \in \{1, \dots, L\}$; subsequently minimizes the loss function. A deep network is harder to train due to gradient instability issue.

In CNN model, an input $x_{i,j}$ is characterized by dimension $H \times W$ and a kernel $k_{m,n}$ is characterized by dimension $k_1 \times k_2$. $w^l_{m,n}$ is the weight matrix connecting neurons of layer l with neurons of layer $l-1$. Activation function exists in between the convolution and pooling layer where we define the output vector as $o^l_{i,j} = f(x^l_{i,j})$. Computation of convolved input vector at layer l is shown in

$$x^l_{i,j} = \sum_m \sum_n w^l_{m,n} o^{l-1}_{i+m,j+n} + b^l \tag{2}$$

Gradients $\delta_{i,j}$ were generated during backpropagation. The gradients at layer l is expressed in

$$\delta^l_{i,j} = \frac{\partial E}{\partial x^l_{i,j}} \tag{3}$$

Chain rule was utilized to calculate the gradients layer-by-layer. Ultimately, the gradient error propagated from L to $L-q$ can be presented as a recursive nested expression given in

$$\frac{\delta^{L-q}_{Q\in i,j}}{\delta^{L}_{Q\in i,j}} = \sum_{Q_{L-q+1}} \sum_{Q_{L-q+2}} \cdots \sum_{Q_L} \prod_{p=1}^{q} w_{Q_{L-p}Q_{L-p+1}} o_{L-p}\left(x^{L-p}_{Q_{L-p}}\right) \tag{4}$$

According to previous study, the initial weights and activation function are significant factors that lead to either vanishing gradient or explosive gradient [32]. Consequently, unsuitable activation function can aggravate the gradient instability.

3.4.1 Sigmoid

The Sigmoid is S-shaped bounded, monotonic function. It is also known as logistic function. Regardless of any type of input given to this function, the output will always be either zero or one. The Sigmoid is expressed in

$$f(x) = \frac{1}{1 + e^{-x}} \tag{5}$$

3.4.2 Rectified Linear Unit (RELU)

RELU's structure resembles a nearly linear function in the range from zero to infinity [18], which helps to preserve the properties of linear models; making them easy to optimize with gradient-descent methods. RELU is a threshold-based activation function and does not involve any exponential or divisions. Its derivative is always one for positive value inputs thus, contributing to faster computation compared to other activation functions. The RELU is expressed in

$$f(x) = max(0, x) = \begin{cases} x_i, & if\ x_i \geq 0 \\ 0, & if\ x_i < 0 \end{cases} \tag{6}$$

RELU output is the same as the input in positive value range. It will rectify the negative value inputs by forcing them to zero. Because the function does not have an asymptotic upper and lower bound, it is able to receive errors from the last layers to update all weights between layers. Hence, RELU does not report vanishing gradient problem. A major limitation of RELU is dying RELU problem i.e. an inactive unit always produces zero gradient and causes the weights not updated during gradient-based optimization [26]. The neurons will stop responding to variations of inputs and stuck in a perpetually inactive state.

3.4.3 Leaky RELU (LRELU)

LRELU [19] was introduced as a generalization of RELU to mitigate potential problems caused by hard zero activations reported in dying RELU problem. Hence, LRELU inherits an identical structure as of RELU except a "leaky" rectifier at the range of negative value inputs, where a constant α is added to the negative value inputs, to sustain weight update during entire propagation process. The LRELU is expressed in

$$f(x) = max(0, x) = \begin{cases} x, & if\ x \geq 0 \\ \alpha x, & if\ x < 0 \end{cases} \tag{7}$$

In Maas et al. work, the value of constant α was set at 0.01. Based on this modification, LRELU permits a small, non-zero gradient when the unit is saturated and inactive. Therefore, it contributes to a more robust gradient optimization. The operation implies a compromise to zero sparsity in order to compute the non-zero gradient. LRELU does not fix the problem of exploding gradient and the value of α still needs to be determined manually via carefully yet tedious training.

3.4.4 Exponential Linear Unit (ELU)

ELU [21] was proposed to accelerate the learning speed in deep neural network. The α parameter is maintained to help managing the value to which ELU saturates for negative value inputs. Hence, the vanishing gradient problem remained subdued given that its derivatives is one and not contractive. Then, an exponential operation is added into the negative value inputs. The ELU is expressed in

$$f(x) = \begin{cases} x, & if\ x \geq 0 \\ \alpha(e^x - 1), & if\ x < 0 \end{cases} \tag{8}$$

The output average of ELU is pushed closer to zero which contributes to quicker convergence. Moreover, units with non-zero mean activation will act as bias for the next layer. If such units do not cancel out each other, learning causes a bias shift for units in the next layer. Since ELU has negative values, it can decrease bias shift by pushing mean activation towards zero during training. Mean shifts toward zero speed up learning by bringing the normal gradient closet to the unit natural gradient. However, exploding gradient problem still persists in ELU.

3.4.5 SWISH

SWISH [24] is a smooth activation function that nonlinearly interpolates between the linear function and the RELU function. The degree of interpolation is governed by adding trainable parameter β into the function, which leads to the generalization of SWISH known as ESWISH [33]. The function is unbounded above and bounded below. SWISH are smooth and non-monotonic. The smooth property optimizes the output results while the non-monotonic property helps to preserve small negative inputs and thus, improves expressivity and gradient flow. The SWISH and ESWISH are expressed in Equation 9 and 10, respectively.

$$f(x) = x \cdot \sigma(x) \; where \; \sigma(x) = sigmoid(x) = \frac{x}{1 + e^{-x}} \tag{9}$$

$$f(x) = \beta x \cdot \sigma(\beta x) \; where \; \sigma(z) = \frac{\beta x}{1 + e^{-\beta * x}} \tag{10}$$

Experiment showed that SWISH has consistently outperformed RELU on deeper neural networks. The function is able to overcome the vanishing gradient problem reported by the sigmoid function. However, the better performance comes at the expense of greater computational cost when compared to RELU.

3.4.6 MISH

MISH [25] is a smooth and non-monotonic activation function inspired by SWISH. It is bounded below to encourage stronger regularization and unbounded above within the range $[\approx -0.31, \infty)$ to avert saturation that has dragged down the training speed due to near zero gradient. Based on the outcome of first derivative, the function of MISH is closely related to SWISH. The MISH is expressed in

$$f(x) = x \cdot tanh\big(softplus(x)\big) = x \cdot tanh\big(In(1 + e^{-x})\big) \tag{11}$$

Similar to SWISH, MISH inherits the self-gating property where the non-modulated input is multiplied with the output of a non-linear function of the input. MISH preserves a small amount of negative value inputs in order to ensure greater information flow and expressivity. The latter has demonstrated better robustness in improving the results of deep neural network compared to SWISH and RELU. Yet, the improved accuracy is a trade-off with higher computational cost and MISH was validated on classification task by using CIFAR datasets only.

3.4.7 RELU-Tanh

Li et al. combined both RELU and Tanh in their works. Tanh function complements the drawbacks of RELU, especially at the negative section [28]. Both RELU and Tanh are governed by parameter α and β, respectively. The RELU-Tanh is expressed in

$$f(x) = \alpha * Relu(Y) + \beta * Tanh(Y) \tag{12}$$

Both constants can be fixed or trainable. As a result, performance of RELU-Tanh highly depends on the right selection of constant value. The output of Tanh function is negative when RELU function outputs zero. The output of Tanh function could introduce the information neglected by RELU and avoids vanishing gradient.

3.4.8 RELU-Memristor-like-Activation Function (RMAF)

RMAF [29] uses a memristive window function technique to find a scalar activation function. A pair of threshold hyperparameters, p and j were introduced to scale the flatness of region, which aims to improve the accuracy of deep neural network. At the same time, constant α was also included as a trainable parameter. The RMAF is expressed in

$$f(x) = \alpha x \cdot \left[j \left(\frac{1}{(0.25 \cdot (1 + e^{-x}) + 0.75)^{p}} \right) \right] \tag{13}$$

For instance, RMAF is unbounded above and bounded below. The unboundedness is critical to avert saturation when the training is slow due to near-zero gradient. Meanwhile, the bounded below property contributes to stronger regularization effects. Despite its desirable properties, RMAF is limited by the selection of learnable parameter as well as the threshold of hyperparameters.

3.4.9 The Proposed Function: SegRELU

RELU has been serving as the base for most of the newer activation functions because of its intuitiveness at $x \geq 0$. However, it is hard saturated at $x < 0$; making its' activation neuron fragile and vulnerable to irreversible neuronal death. By summarizing from existing literature, we have decided the proposed activation function that is suitable for medical image segmentation should possess: (1) parameter-free setting to ease users' implementation, (2) light architecture to encourage reproducibility in future deep learning models, and (3) robust weight update at both the positive and negative part of the activation functions to preserve effective feature learning during training process. A new hybrid activation function, SegRELU, is introduced. The plots of commonly used activation functions and SegRELU are shown Fig. 4.

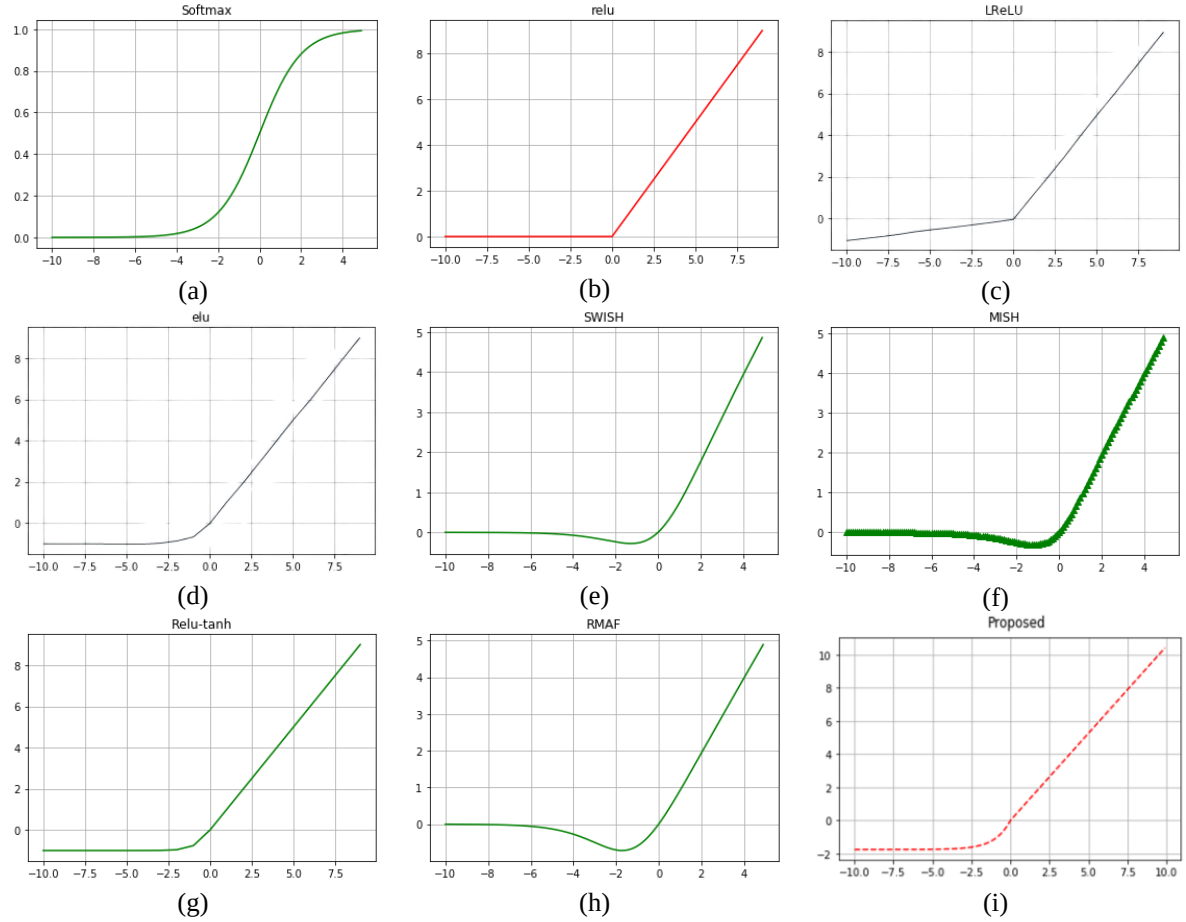


Fig 4. Piecewise activation functions. (a) Sigmoid (b) RELU (c) LRELU (d) ELU (e) SWISH (f) MISH (g) RELU-Tanh, (h) RMAF and (i) SegRELU.

Traditional activation functions such as Tanh and ELU grow exponentially, their derivatives or gradients are complex which leads to late convergence in training process of deep learning models. SegRELU is developed based on the complementary of RELU and Softsign function. We have retained the nearly linear property of RELU at $x > 0$ in order to maintain the light computation and architecture. To ensure effective learning and smooth convergence, we have identified Softsign to compensate the negative part. Softsign has solid theoretical foundation. The function belongs to the quadratic activation function family, which has proven to be invariant to the translation of a subject across a background of zeros [34]. The principle they draw on is that of spatial correlation and can be written as linear functions of the spatial auto-correlation of an image [35].

Softsign grows polynomially and has smoother asymptote line to demonstrate greater degree of non-linearization. The non-linearity nature presented by quadratic function is highly regarded in neural network research to preserve pertinent features. SegRELU inherits the non-linearity, hence it can easily delineate complicated object boundary in medical image. Moreover, SegRELU can generate an activation when calculating gradient in negative part,

dead neuron problem during the training can be averted. Formulations of SegRELU and its derivative are given in Equation 14 and 15, respectively.

$$f(x) = \begin{cases} \frac{x}{1+|x|} & x \leq 0 \\ x & x > 0 \end{cases} \tag{14}$$

$$f'(x) = \begin{cases} \frac{1}{1+|x|^2} & x \leq 0 \\ 1 & x > 0 \end{cases} \tag{15}$$

Lastly, we summarized the characteristics of different activation functions in terms of vanishing gradient, negative activation, bound, monotonic and order of continuity in Table 1.

Table 1. A summary of characteristics demonstrated by different activation functions

Activation function	Vanishing Gradient	Negative Activation possible	Bound Activation function	Monotonic Function	Order of Continuity
Sigmoid	Yes	Yes	Yes	No	C^{∞}
RELU	No	No	No	Yes	C^0
LRELU	No	Yes	No	Yes	C^0
ELU	Yes, for negative values	Yes	No	If alpha>0	C^1 if alpha = 1 C^0 otherwise
SWISH	No	Yes	No	No	C^{∞}
MISH	No	Yes	No	No	C^{∞}
RELU-Tanh	No	Yes	Yes, for negative inputs	Yes	C^{∞}
RMAF	No	Yes	Yes, for negative inputs	Yes	C^{∞}
SegRELU	No	Yes	Yes, for negative inputs	Yes	C^1

4. Results and Discussion

4.1 Experiment Setting

Each knee image dataset has size of 384 × 384 × 160. Whole cartilage was segmented from knee images. A total of 32 training datasets, 32 augmented datasets and 2 validation datasets were involved. The knee segmentation produces an array of binary segmentation output, which was reconstructed into 3D. In cell image dataset, each image has a size of 128 × 128. We used 650 images for training and 10 for validation. The output is a binary segmentation to extract whole cell from background image.

The model was trained from scratch on a NVIDIA GeForce RTX 3070 GPU (Santa Clara, CA, US) with Tensorflow and Pytorch library as backend. ADAM was used to optimize the training of model and batch normalization was used to stabilize the training. The model adopted learning rate of 0.01, batch size of 8 and 25 epochs when training with cell image. Then, we adopted the learning rate of 0.01, batch size of 2 and 50 epochs when training with knee image. An energy function was computed by a pixel-wise softmax over the final feature map combined with cross entropy loss function. For each activation function, the model training took approximately 4-5 hours for cell images and 7-8 hours for knee images.

Segmentation accuracies produced by U-net with different activation functions were assessed by using Dice Similarity Coefficient (DSC), a benchmark evaluation metric in medical image segmentation [36]. DSC is defined as the degree of similarity between the segmented object, X and groundtruth, Y, as shown in Equation 16. We also applied Intersection over Union (IoU) and Volumetric Overlap Error (VOE). IoU is alternatively known as the Jaccard index. In Equation 17, the metric measures the number of pixels shared by X and Y divided by the total number of pixels present across both masks.

$$DSC = \frac{2|X \cap Y|}{|X| + |Y|} \tag{16}$$

$$IoU = \frac{|X \cap Y|}{|X \cup Y|} \tag{17}$$

Meanwhile, VOE is the corresponding error measure expressed in

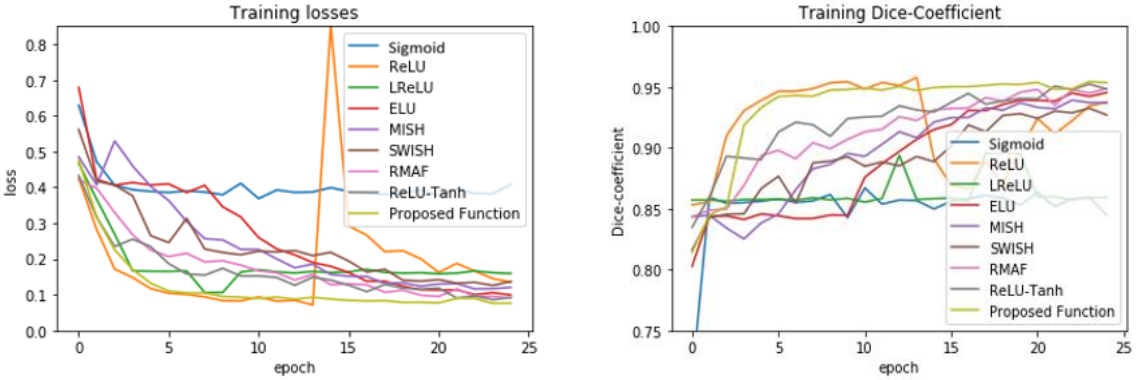
$$VOE = 1 - \frac{|X \cap Y|}{|X \cup Y|} \tag{18}$$

4.2 Model Training and Validation Performance

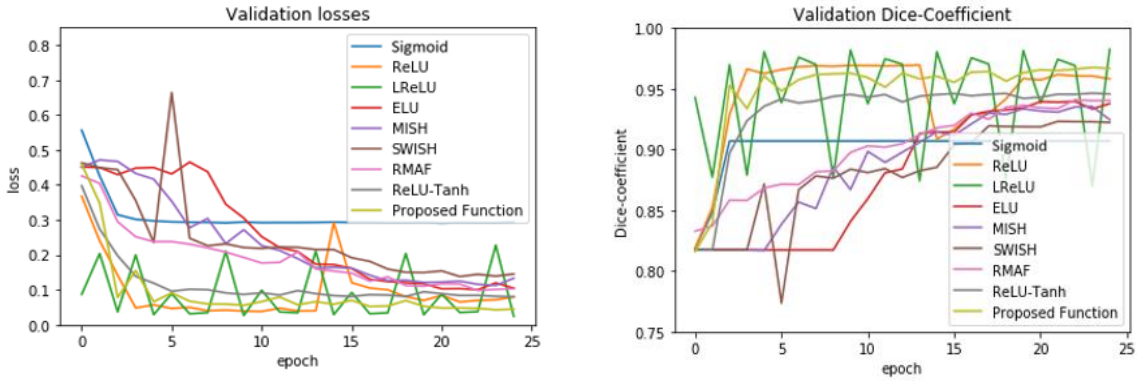
During model training, the accuracy was measured by using DSC. Training and validation performance of U-net by using cell images is depicted in Fig. 5. Parameters α and β of RELU-Tanh function were set at 1 and 0, respectively, while the parameter α as well as hyperparameters p and j of RMAF were set at 0.1, 1, and 1, respectively. Based on Fig. 5(a), Sigmoid has converged too swiftly to a suboptimal solution while ELU, MISH and SWISH are training slowly as the weights update progress is little. RELU has demonstrated sudden divergence in the middle of training.

Among the activation functions, only SegRELU has demonstrated good and stable learning performance throughout the model training. Then, we assessed the overfitting endured by U-net during training. By referring to Fig. 5(b) and (d), SegRELU has presented consistent training and validation curves wherein it converges around DSC of 0.9529±0.0188. The small gap between both curves suggests the proposed function does not suffer from overfitting. Comparatively, we can observe slight overfitting endured by Sigmoid and chaotic validation accuracy performance exhibited by LRELU.

In Fig. 6, the training and validation performance of U-net by using knee images are depicted. Parameters α and β of RELU-Tanh function were set at 1 and 0, respectively. The parameter α and hyperparameters p and j of RMAF were set at 0.1, 1.5, and 1, respectively. Overall, these plots are showing more wiggles than Fig. 5 due to the small batch size value. By referring to Fig. 6(a), SegRELU has exhibited good learning curve and converged smoothly. Sigmoid has demonstrated chaotic learning performance. Besides, its training does not converge. RELU demonstrates relatively low learning rate. Then, we compared the training and validation accuracy plots to check for model overfitting. By referring to Fig. 6(b) and (d), SegRELU does not report overfitting problem given that the gap between both curves is small. Sigmoid experiences volatile training accuracy performance.

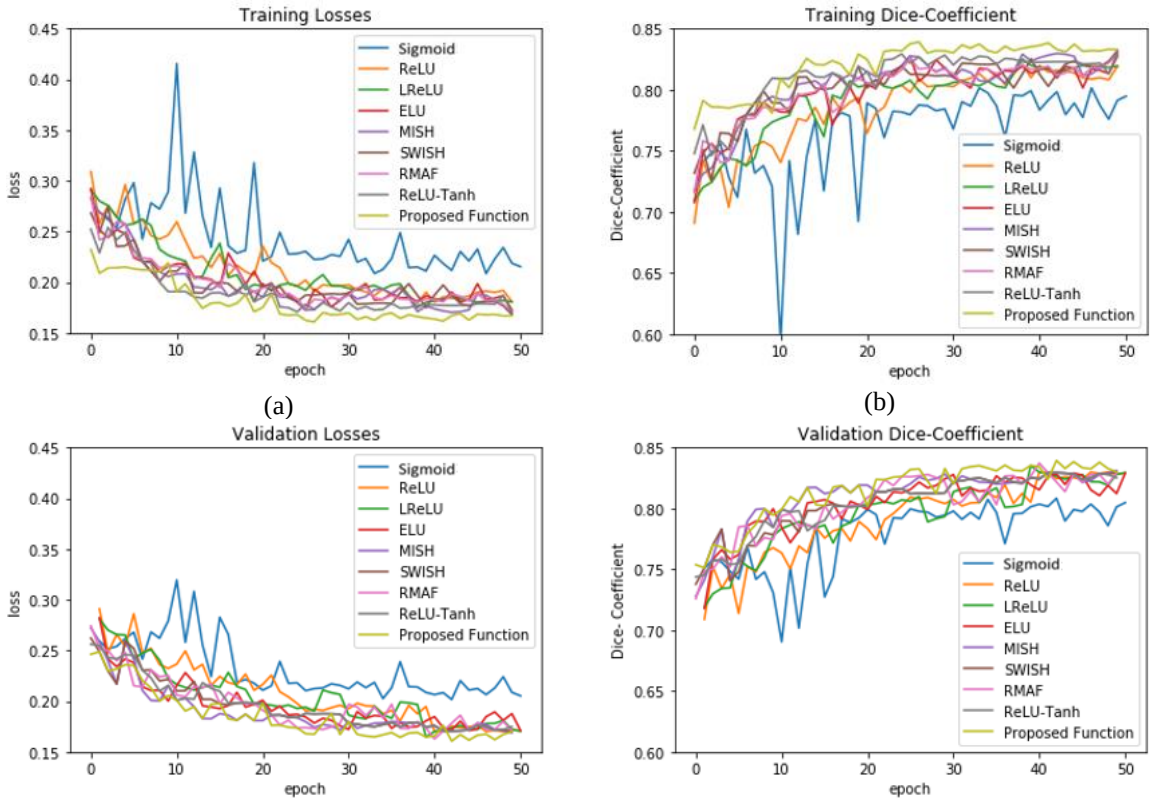


(a) (b)



(c) (d)

Fig. 5: U-net model performance by using cell dataset: (a) training losses, (b) training accuracy, (c) validation losses, and (d) validation accuracy.



(a) (b)

(c) (d)

Fig. 6: U-net model performance by using knee dataset: (a) training losses, (b) training accuracy, (c) validation losses, and (d) validation accuracy.

Performance of U-net model training and validation with different activation functions were summarized in Table 2. U-net with SegRELU outperforms other activation functions. It has demonstrated the best training accuracy and loss of 0.9287±0.0189 and 0.1443±0.0034 in cell dataset and 0.8449±0.0172 and 0.1543±0.0033 in knee dataset. Sigmoid has exhibited the worst training accuracy and loss of 0.8410±0.0262 and 0.5020±0.0158 in cell dataset and 0.7848±0.0269 and 0.2740±0.0096 in knee dataset. Combined with its training characteristics in Fig.5 and 6, we conclude that the Sigmoid is unsuitable to be deployed in deep learning segmentation model. Meanwhile, the training and validation results for other activation functions are mixed. For example, RMAF performs well in cell dataset (Training DSC: 0.9228±0.0273, Validation DSC: 0.9308±0.0224) but performs mediocrely in knee dataset (Training DSC: 0.8235±0.0196, Validation DSC: 0.8001±0.0188). Similar observation is made in RELU-Tanh. As such, the inconsistency of training performance is influenced by the selection of parameters required by both activation functions.

Table 2. Training and validation of U-net with different activation functions during model training

	Cell Image Dataset				Knee Image Dataset			
	Train		Validation		Train		Validation	
	DSC	Loss	DSC	Loss	DSC	Loss	DSC	Loss
Sigmoid	0.8410 ±0.0262	0.5020 ±0.0158	0.8720 ±0.0310	0.3085 ±0.0098	0.7848 ±0.0269	0.2740 ±0.0096	0.7959 ±0.0261	0.2402 ±0.0083
RELU	0.9073 ±0.0231	0.1910 ±0.0057	0.9350 ±0.0249	0.0929 ±0.0024	0.8210 ±0.0225	0.1907 ±0.0053	0.8018 ±0.0230	0.1903 ±0.0046
LRELU	0.9104 ±0.0219	0.1904 ±0.0049	0.9015 ±0.0231	0.1747 ±0.0158	0.8324 ±0.0206	0.1897 ±0.0047	0.8041 ±0.0193	0.1901 ±0.0053
ELU	0.8798 ±0.0253	0.2648 ±0.0077	0.8649 ±0.0223	0.2549 ±0.0068	0.8409 ±0.0215	0.1896 ±0.0051	0.8103 ±0.0192	0.1862 ±0.0043
SWISH	0.8787 ±0.0244	0.2589 ±0.0078	0.8652 ±0.0219	0.2644 ±0.0058	0.8348 ±0.0206	0.1707 ±0.0041	0.8091 ±0.0218	0.1890 ±0.0051
MISH	0.9090 ±0.0239	0.1936 ±0.0068	0.8320 ±0.0234	0.1970 ±0.0060	0.8340 ±0.0218	0.1890 ±0.0045	0.8009 ±0.0209	0.1864 ±0.0046
RELU-Tanh	0.9101 ±0.0238	0.1945 ±0.0059	0.9209 ±0.0259	0.1131 ±0.0029	0.8258 ±0.0262	0.1657 ±0.0041	0.8018 ±0.0211	0.1804 ±0.0048
RMAF	0.9228 ±0.0273	0.1540 ±0.0043	0.9308 ±0.0224	0.1149 ±0.0033	0.8235 ±0.0196	0.1787 ±0.0043	0.8001 ±0.0188	0.1890 ±0.0044
SegRELU	**0.9287 ±0.0189**	**0.1443 ±0.0034**	**0.9529 ±0.0188**	**0.0929 ±0.0020**	**0.8449 ±0.0172**	**0.1543 ±0.0033**	**0.8340 ±0.0167**	**0.1607 ±0.0032**

4.3 Segmentation Results

Accurate segmentation outcome is fundamental to medical image analysis [37]. In this experiment, we have used 320 knee images and 10 cell images to test the segmentation performance of U-net with different activation functions. Table 3 reports the segmentation results by using IoU, VOE and DSC. U-net with SegRELU outperformed other activation functions in cell dataset (IoU: 0.9149±0.0181, VOE: 0.06450±0.0014 and DSC: 0.9604±0.0197) and knee dataset (IoU: 0.8878±0.0191, VOE: 0.1443±0.0027 and DSC: 0.9399±0.0187). Sigmoid has the worst segmentation results in cell dataset (IoU: 0.5486±0.0178, VOE: 0.4513±0.0169 and DSC: 0.7085±0.0231) and knee dataset (IoU: 0.8309±0.0262, VOE: 0.1690±0.0051 and DSC: 0.9076±0.0287). The bad segmentation results are inherited by chaotic training performance experienced by the function. Meanwhile, other activation functions have demonstrated mixed accuracies ranking in different image datasets.

Table 3. Average accuracy scores of different activation functions by using cell and knee datasets during model testing

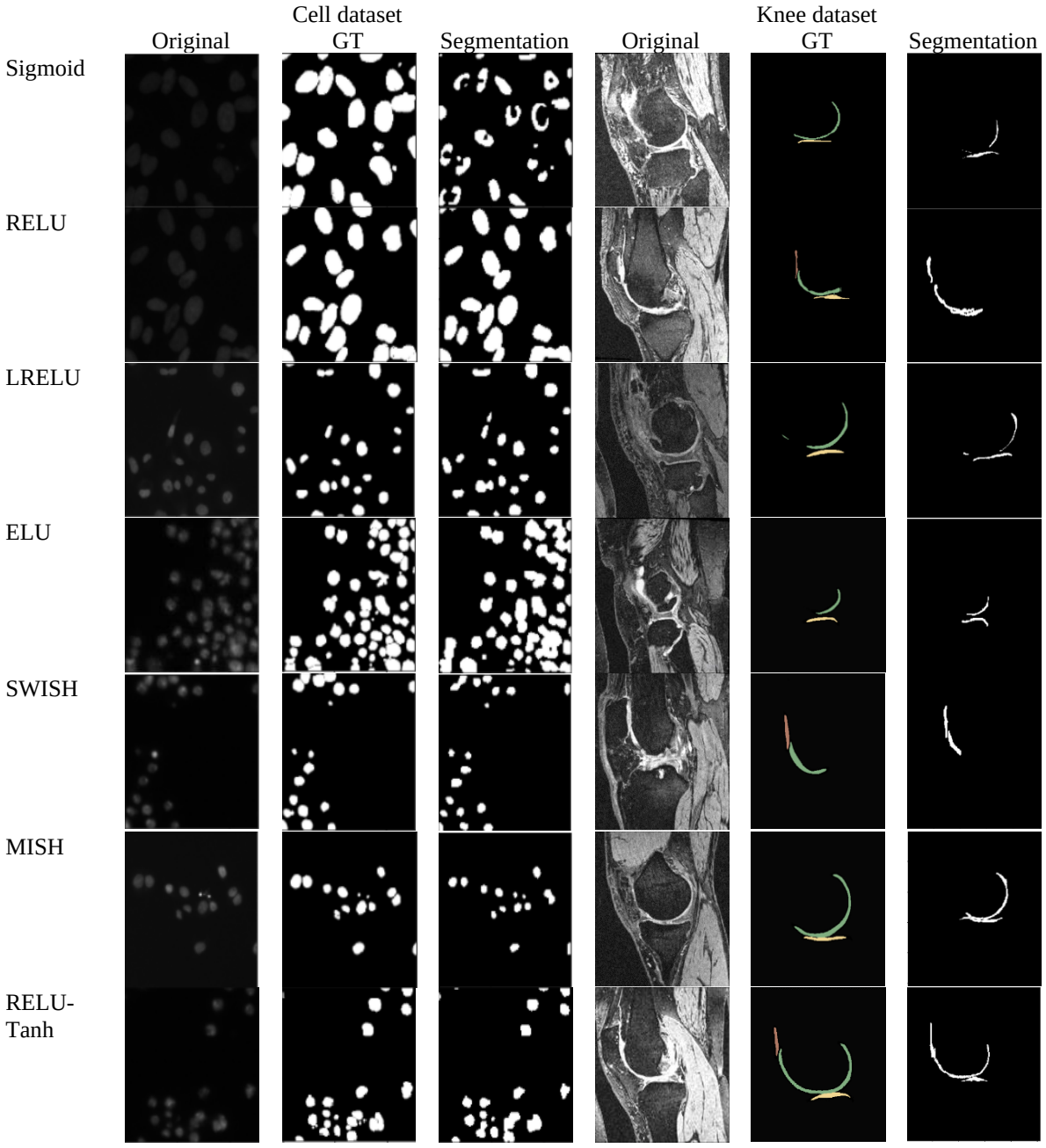
	2D Cell Dataset			3D Knee Dataset		
	IoU	VOE	DSC	IoU	VOE	DSC
Sigmoid	0.5486 ±0.0178	0.4513 ±0.0169	0.7085 ±0.0231	0.8309 ±0.0262	0.1690 ±0.0051	0.9076 ±0.0287
RELU	0.7849 ±0.0173	0.2150 ±0.0058	0.8795 ±0.0217	0.8449 ±0.0221	0.1640 ±0.0037	0.9098 ±0.0231
LRELU	0.7086 ±0.0175	0.2913 ±0.0076	0.8294 ±0.0215	0.8403 ±0.0193	0.1670 ±0.0047	0.9084 ±0.0208
ELU	0.8454 ±0.0221	0.1545 ±0.0042	0.9162 ±0.0219	0.8546 ±0.0213	0.1549 ±0.0041	0.9129 ±0.0214
SWISH	0.7756 ±0.0209	0.2243 ±0.0062	0.8736 ±0.0236	0.8560 ±0.0226	0.1507 ±0.0037	0.9160 ±0.0214
MISH	0.8885 ±0.0208	0.1110 ±0.0029	0.9409 ±0.0252	0.8606 ±0.0217	0.1506 ±0.0036	0.9167 ±0.0239
RELU-Tanh	0.9123 ±0.0190	0.08760 ±0.0020	0.9541 ±0.0253	0.8578 ±0.0216	0.1530 ±0.0037	0.9223 ±0.0215
RMAF	0.8126 ±0.0186	0.1873 ±0.0044	0.8966 ±0.0242	0.8494 ±0.0215	0.1601 ±0.0038	0.9109 ±0.0268
SegRELU	**0.9149 ±0.0181**	**0.06450 ±0.0014**	**0.9604 ±0.0197**	**0.8878 ±0.0191**	**0.1443 ±0.0027**	**0.9399 ±0.0187**

In Fig. 7, the segmentation results of cell and knee images are depicted and compared with respective ground truth. In cell segmentation, extraction of intact cell shape and correct number of cell are important. U-net with Sigmoid has produced seriously undersegmented cell shape, which cannot be utilized for clinical application.

Meanwhile, U-net with RELU and LRELU have generated mildly oversegmented cell, where we observed some cells are interconnected. Lastly, U-net with SegRELU has been able to produce desirable cell segmentation results.

Segmentation of knee, on the other hand, is more daunting because of the irregular and changing anatomical geometries across slices. Knee cartilage is divided into femoral, tibial and patellar cartilage. The knee image background is characterized by numerous musculoskeletal tissues such as bones, ligaments, fatty tissues and muscles. The complex landscape exhibited by knee image can easily mislead segmentation model of ineffective learning. Good knee segmentation should preserve accurate cartilage shape (without undersegmentation or oversegmentation) and clear boundary delineation between different types of cartilage.

Based on Fig. 7, U-net with Sigmoid has produced the worst segmentation result. The segmented femoral cartilage has been severely deformed where the anterior section has been missing when compared to the ground truth. RELU has encountered oversegmentation in tibial cartilage and LRELU has encountered undersegmentation in middle femoral cartilage. Meanwhile, RELU-Tanh has produced satisfactory result but has reported minor oversegmentation in patella cartilage. MISH has experienced slight undersegmentation in the anterior femoral cartilage. Only SegRELU is able to produce desirable knee segmentation result.



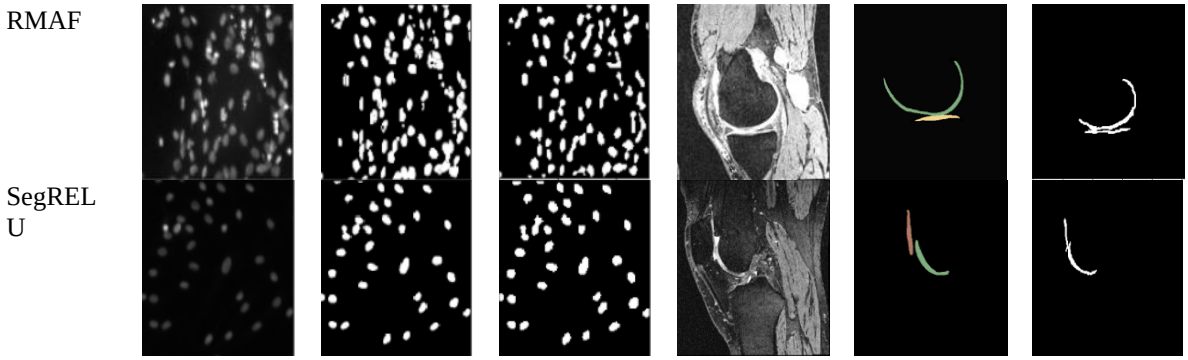


Fig. 7: Segmentation results generated by U-net using different activation functions.

Fig. 8 shows the 3D knee segmentation results from anterior view. Knee cartilage produced by Sigmoid has jaggy boundary and undersegmentation at the anterior and posterior lateral femoral cartilage. Similar undesirable results are observed by knee segmentation model with RMAF. The anterior lateral femoral cartilage has experienced serious undersegmentation while the posterior medial femoral cartilage has shown uneven surface delineation. Result produced by segmentation model with RELU has produced undersegmented anterior lateral cartilage and undefined anterior medial cartilage boundary with medial tibial cartilage. As a result, the whole cartilage model is deformed. Segmentation result produced by model with SegRELU has produced cartilage model with smooth boundary surface and correct anatomical shape. The boundary delineation between femoral and tibial is clear.

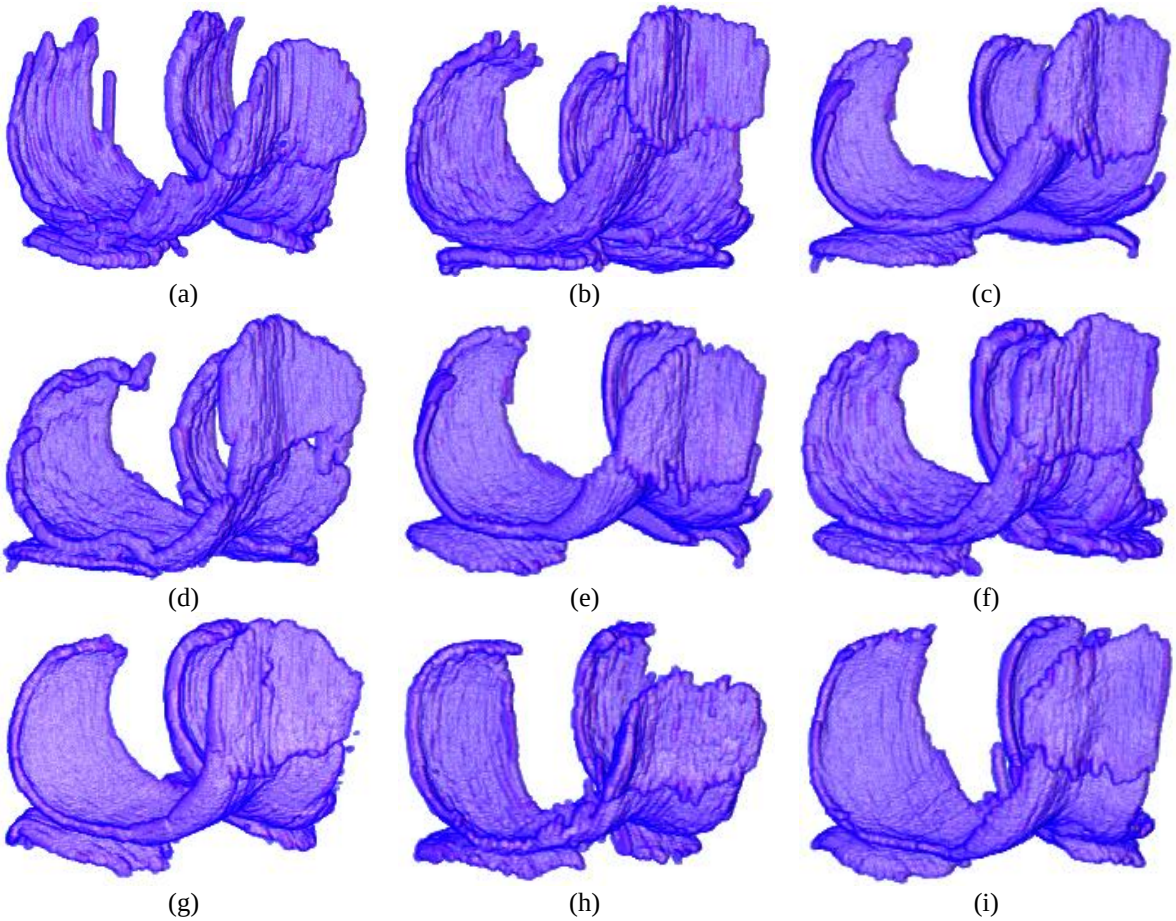


Fig. 8: 3D results of cartilage segmentation generated by U-net using (a) Sigmoid (b) RELU (c) LRELU (d) ELU (e) SWISH (f) MISH (g) RELU-Tanh, (h) RMAF and (i) SegRELU.

5. Conclusion

In this paper, we proposed a novel activation function known as SegRELU. The function is characterized by its parameter-free easy implementation for deep learning segmentation models, stable training performance and consistent good segmentation accuracies across different medical image datasets. Retrospectively, advanced activation functions such as RELU-Tanh and RMAF requires careful parameter tuning to obtain desirable results while basic activation functions such as Sigmoid is deemed unfit for segmentation task. Other activation functions such as RELU, ELU and LRELU have reported mixed results. For future work, we will evaluate the impact of SegRELU on other task such as classification and expand the experiment into more types of medical image dataset of different abnormality grades.

References

[1] V. Cherukuri, P. Ssenyonga, B. Warf, A. Kulkarni, V. Monga, and S. Schiff, "Learning Based Segmentation of CT Brain Images: Application to Postoperative Hydrocephalic Scans," *IEEE Transactions on Biomedical Engineering,* vol. 65, pp. 1871-1884, 2018.

[2] S. Wang *et al.*, "Central focused convolutional neural networks: Developing a data-driven model for lung nodule segmentation," *Medical Image Analysis,* vol. 40, pp. 172-183, 2017/08/01/ 2017, doi: https://doi.org/10.1016/j.media.2017.06.014.

[3] H.-S. Gan, K. A. Sayuti, M. H. Ramlee, Y.-S. Lee, W. M. H. Wan Mahmud, and A. H. Abdul Karim, "Unifying the seeds auto-generation (SAGE) with knee cartilage segmentation framework: data from the osteoarthritis initiative," *International Journal of Computer Assisted Radiology and Surgery,* vol. 14, no. 5, pp. 755-762, 2019/05/01 2019, doi: 10.1007/s11548-019-01936-y.

[4] T. Song, V. Sanchez, H. EIDaly, and N. M. Rajpoot, "Dual-Channel Active Contour Model for Megakaryocytic Cell Segmentation in Bone Marrow Trephine Histology Images," *IEEE Transactions on Biomedical Engineering,* vol. 64, no. 12, pp. 2913-2923, 2017, doi: 10.1109/TBME.2017.2690863.

[5] H. S. Gan, T. Tan, A. H. A. Karim, K. A. Sayuti, and M. R. A. Kadir, "Multilabel graph based approach for knee cartilage segmentation: Data from the osteoarthritis initiative," in *IEEE Conference on Biomedical Engineering and Sciences (IECBES)*, 8-10 Dec. 2014 2014, pp. 210-213, doi: 10.1109/IECBES.2014.7047487.

[6] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature,* vol. 521, no. 7553, pp. 436-444, 2015/05/01 2015, doi: 10.1038/nature14539.

[7] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional Networks for Biomedical Image Segmentation," in *International Conference on Medical Image Computing and Computer-Assisted Intervention MICCAI 2015*, Cham, 2015, vol. 9351: Springer International Publishing, in Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015, pp. 234-241.

[8] C. Gillmann, D. Saur, T. Wischgoll, and G. Scheuermann, "Uncertainty-aware Visualization in Medical Imaging - A Survey," *Computer Graphics Forum,* vol. 40, no. 3, pp. 665-689, 2021, doi: https://doi.org/10.1111/cgf.14333.

[9] C. Nwankpa, W. Ijomah, A. Gachagan, and S. Marshall, *Activation Functions: Comparison of trends in Practice and Research for Deep Learning*. 2018.

[10] J. Feng and S. Lu, "Performance Analysis of Various Activation Functions in Artificial Neural Networks," *Journal of Physics: Conference Series,* vol. 1237, p. 022030, 2019/06 2019, doi: 10.1088/1742-6596/1237/2/022030.

[11] B. Ding, H. Qian, and J. Zhou, "Activation functions and their characteristics in deep neural networks," in *2018 Chinese Control And Decision Conference (CCDC)*, 9-11 June 2018 2018, pp. 1836-1841, doi: 10.1109/CCDC.2018.8407425.

[12] D. Pedamonti, "Comparison of non-linear activation functions for deep neural networks on MNIST classification task," *ArXiv,* vol. abs/1804.02763, 2018.

[13] X. Liu, J. Zhou, and H. Qian, "Comparison and Evaluation of Activation Functions in Term of Gradient Instability in Deep Neural Networks," in *2019 Chinese Control And Decision Conference (CCDC)*, 3-5 June 2019 2019, pp. 3966-3971, doi: 10.1109/CCDC.2019.8832578.

[14] T. Szandała, "Review and Comparison of Commonly Used Activation Functions for Deep Neural Networks," in *Bio-inspired Neurocomputing*, vol. 903, A. K. Bhoi, P. K. Mallick, C.-M. Liu, and V. E. Balas Eds. Singapore: Springer Singapore, 2021, pp. 203-224.

[15] B. L. Kalman and S. C. Kwasny, "Why tanh: choosing a sigmoidal function," in *[Proceedings 1992] IJCNN International Joint Conference on Neural Networks*, 7-11 June 1992 1992, vol. 4, pp. 578-581 vol.4, doi: 10.1109/IJCNN.1992.227257.

[16] K.-I. Funahashi, "On the approximate realization of continuous mappings by neural networks," *Neural Networks,* vol. 2, no. 3, pp. 183-192, 1989/01/01/ 1989, doi: https://doi.org/10.1016/0893-6080(89)90003-8.

[17] C. Dugas, Y. Bengio, F. Bélisle, C. Nadeau, and R. Garcia, "Incorporating Second-Order Functional Knowledge for Better Option Pricing," in *NIPS*, 2000.

[18] V. Nair and G. E. Hinton, "Rectified linear units improve restricted boltzmann machines," presented at the Proceedings of the 27th International Conference on International Conference on Machine Learning, Haifa, Israel, 2010.

[19] A. L. Maas, A. Y. Hannun, and A. Y. Ng, "Rectifier Nonlinearities Improve Neural Network Acoustic Models," in *International Conference on Machine Learning (ICML)* 2013.

[20] K. He, X. Zhang, S. Ren, and J. Sun, "Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification," in *2015 IEEE International Conference on Computer Vision (ICCV)*, 7-13 Dec. 2015 2015, pp. 1026-1034, doi: 10.1109/ICCV.2015.123.

[21] D.-A. Clevert, T. Unterthiner, and S. Hochreiter, "Fast and Accurate Deep Network Learning by Exponential Linear Units (ELUs)," *Under Review of ICLR2016 (1997),* 11/23 2015.

[22] G. Klambauer, T. Unterthiner, A. Mayr, and S. Hochreiter, "Self-normalizing neural networks," in *Advances in Neural Information Processing Systems (2017)*, 2017, pp. 971-980.

[23] D. Hendrycks and K. Gimpel, "Bridging Nonlinearities and Stochastic Regularizers with Gaussian Error Linear Units," *ArXiv,* vol. abs/1606.08415, 2016.

[24] P. Ramachandran, B. Zoph, and Q. V. Le, "Swish: a Self-Gated Activation Function," *arXiv: Neural and Evolutionary Computing,* 2017.

[25] D. Misra, *Mish: A Self Regularized Non-Monotonic Neural Activation Function*. 2019.

[26] L. Lu, Y. Shin, Y. Su, and G. Karniadakis, "Dying ReLU and Initialization: Theory and Numerical Examples," *ArXiv,* vol. abs/1903.06733, 2019.

[27] F. Manessi and A. Rozza, "Learning Combinations of Activation Functions," in *2018 24th International Conference on Pattern Recognition (ICPR)*, Beijing, China, 20-24 Aug. 2018 2018, pp. 61-66, doi: 10.1109/ICPR.2018.8545362.

[28] X. Li, Z. Hu, and X. Huang, "Combine Relu with Tanh," in *2020 IEEE 4th Information Technology, Networking, Electronic and Automation Control Conference (ITNEC)*, 12-14 June 2020 2020, vol. 1, pp. 51-55, doi: 10.1109/ITNEC48623.2020.9084659.

[29] Y. Yu, K. Adu, N. Tashi, P. Anokye, X. Wang, and M. A. Ayidzoe, "RMAF: Relu-Memristor-Like Activation Function for Deep Learning," *IEEE Access,* vol. 8, pp. 72727-72741, 2020.

[30] G. Hong-Seng, K. A. Sayuti, and A. H. A. Karim, "Investigation of random walks knee cartilage segmentation model using inter-observer reproducibility: Data from the osteoarthritis initiative," *Bio-Medical Materials and Engineering,* vol. 28, pp. 75-85, 2017, doi: 10.3233/BME-171658.

[31] E. Meijering, "Cell Segmentation: 50 Years Down the Road [Life Sciences]," *IEEE Signal Processing Magazine,* vol. 29, no. 5, pp. 140-145, 2012, doi: 10.1109/MSP.2012.2204190.

[32] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," *Neural Computation,* vol. 9, no. 8, pp. 1735-1780, 1997.

[33] E. Alcaide, "E-swish: Adjusting Activations to Different Network Depths," 01/22 2018.

[34] B. James, D. Guillaume, L. Pascal, and B. Yoshua, "Quadratic Polynomials Learn Better Image Features " *Technical Report 1337. Département d'Informatique et de Recherche Opérationnelle, Université de Montréal.,* vol. 2009, pp. 1-11, 2009.

[35] N. Rust, O. Schwartz, J. Movshon, and E. P. Simoncelli, "Spatiotemporal Elements of Macaque V1 Receptive Fields," *Neuron,* vol. 46, pp. 945-956, 2005.

[36] H. S. Gan and S. Khairil Amir, "Comparison of Improved Semi-Automated Segmentation Technique with Manual Segmentation: Data from the Osteoarthritis Initiative," *American Journal of Applied Sciences,* vol. 13, no. 11, pp. 1068-1075, 2016.

[37] H. S. Gan, K. A. Sayuti, A. H. A. Karim, R. A. M. Rosidi, and A. S. A. Khaizi, "Analysis on Semi-Automated Knee Cartilage Segmentation Model using Inter-Observer Reproducibility: Data

from the Osteoarthritis Initiative," presented at the Proceedings of the 7th International Conference on Bioscience, Biochemistry and Bioinformatics, Bangkok, Thailand, 2017. [Online]. Available: https://doi.org/10.1145/3051166.3051169.