Communications in Mathematics and Applications
Vol. 14, No. 2, pp. 949–968, 2023
ISSN 0975-8607 (online); 0976-5905 (print)
Published by RGN Publications
DOI: 10.26713/cma.v14i2.2152

http://www.rgnpublications.com

Research Article

Image Segmentation on Convolutional Neural Network (CNN) using Some New Activation Functions

Arvind Kumar* and Sartaj Singh Sodhi

Department of Computer Science and Engineering, University School of Information and Communication Technology, Guru Gobind Singh Indrapratha University Delhi, Sector 16C, Dwarka, Delhi 110078, India
***Corresponding author:** arvind.usict.134164@ipu.ac.in

Received: December 28, 2022 **Accepted:** March 3, 2023

Abstract. Image segmentation means subdividing the image into different objects. We use different methods for the segmentation of images. For getting different objects from a single image, we generally apply old methods, such as Fuzzy C-means, and K-means clustering techniques. In the case of getting different objects from huge amounts of image datasets, we generally apply the CNN technique. The Activation Function (AF) plays a major role during the segmentation of images through CNN. We can increase the power of CNN using activation functions. Generally, the exponential activation function is used for very complex and non-separable types of problems. So, in this paper, we segment the triangle and CamVid datasets on CNN using NewSigmoid AF and root2Sigmoid AF. Both AFs are exponential activation functions. We made a 12-layer CNN for the triangle segment and a 13-layer CNN for the CamVid segment. After that, we compare four activation functions on both CNNs. These activation functions are NewSigmoid, Relu, root2sigmoid, and Tanh. On the triangle image dataset, we have achieved global accuracy of 96.29% through NewSigmoid AF, 95.77% through Relu AF, 97.02% through root2sigmoid AF, and 96.47% through Tanh AF. On the CamVid dataset, we have achieved global accuracy of 93.97% through NewSigmoid AF, 94.01% through Relu AF, 94.19% through root2sigmoid AF, and 94.13% through Tanh AF. We also check the validation accuracy of the CamVid dataset. We take 10% of images for validation purpose. For the same dataset and same CNN network, we found validation accuracy of relu is 96.04%, root2simgoid is 96.26%, tanh is 96.20% and NewSimgois is 96.26%. Therefore, we can say that the root2sigmoid AF gives a better result as compared with NewSigmoid, Relu, and Tanh AFs.

Keywords. Convolutional Neural Network (CNN), Activation function, NewSigmoid, Tansigmoid, Training algorithm, Relu, Adam optimization

Mathematics Subject Classification (2020). 68T01, 68T07, 68T10, 68T45

1. Introduction

Image segmentation means subdividing the image into different objects. Thresholding-based methods, Edge Detection methods, Region-based methods, and Cluster-based methods are some famous image segmentation methods (Samaddar and Reddy [36]). Out of them, thresholding is the simplest method. It is also known as the pixel-based method. In this method we find the entire pixel having intensity values below or above some taken pixel value (Gonzalez *et al*. [13]). The initial segmentation of images is done by the pixel-based method (Zhang *et al*. [43]). We cannot use the thresholding method for the segmentation of texture images (Despotović *et al*. [11]). The gray level, colors, texture, shape, and model are some criteria of homogeneity. In the Region-growing based segmentation, we separate the Homogeneity of the regions. In the region growing methods, we find out the pixel values with compare to its neighbor's pixel values. We find out boundary values in the edge-based segmentation process. Some famous edge methods are Robert edge detection, Sobel edge detection (Gonzalez and Woods [12], and Gonzalez *et al*. [13]), Prewitt edge detection (Gonzalez and Woods [12]), Krish edge detection, Robinson edge detection, Marr-Hildreth edge detection, LoG edge detection, Canny edge detection (Canny [7], Davies [9], and Gonzalez and Woods [12]). Some of the famous corner detection methods are: Harris operator, BRISK, FAST (Features from accelerated Segment test, and SUSAN. Determinant of Hessian (DoH), Difference of Gaussian (DoG), and Laplacian of Gaussian (LoG) are some different types of the blod detection methods (Adhikary *et al*. [1]). We detect lines and curves in the hough transformation method. We calculate all of the pixels of the image in the Histogram Based Methods. We make clusters in the clustering process. Belonging to the same group is known as a cluster. This comes under the unsupervised method (Wang *et al*. [41]). Generally, the Euclidean distance formula is used for clustering purposes. Connectivity modes, Centroid clustering, Density models, Subspace models, Graph-based models, Distribution models, and Neural Network models are some famous method that comes under this process. K-means clustering and Fuzzy C-means clustering are the famous methods of centroid clustering. Equation (1) is a K-means clustering method (Kanungo *et al*. [20], MacQueen [28], Mao *et al*. [30], Rafique *et al*. [34]). Equation (2) is a Fuzzy C-Means Clustering method (Hoseini and Shahbahrami [18], and Zheng *et al*. [46], [1]).

If, $C1$ is a number of clusters, then

$$C1=\sum_{J=1}^{K}\sum_{i=1}^{n}\|v_j^i-\mu_j\|^2 \tag{1}$$

where, v_j^i is the ith sample of the jth class K_j and μ_j the center of the jth cluster defined as the mean of v_j.

The FCM clustering minimizes

$$C2=\sum_{J=1}^{K}\sum_{i=1}^{n}\mu_{ij}^m\|v_j^i-\mu_j\|^2, \quad 1\le m\le\infty, \tag{2}$$

where $C2$ is the number of samples n is the number of clusters. μ_j is the d-dimensional center of the cluster, v_j is the d-dimensional data and m is a weighting exponent.

[1] M. A. B. Siddique, R. B. Arif, M. M. R. Khan and Z. Ashrafi, Implementation of fuzzy c-means and possibilistic c-means clustering algorithms, cluster tendency analysis and cluster validation, *Preprint* **2018** (2018), 2018110581, DOI: 10.20944/preprints201811.0581.v1.

Following conditions must be satisfied for equation (2):

$$\sum_{i=1}^{C} \mu_{ij} = 1, \tag{3}$$

$$\mu_{ij}^{m} = \frac{1}{\sum_{i=1}^{n} \left(\frac{\|v_j - \mu_j\|}{\|v_j - \mu_j\|} \right)^{\frac{2}{m-1}}} \quad \text{and} \quad \mu_j = \frac{\sum_{i=1}^{n} \mu_{ij}.v_i}{\sum_{i=1}^{n} \mu_{ij}}. \tag{4}$$

If the image had fewer complexes, then the fuzzy C-means clustering gave better results; but if the image had more complex ones, then the K-means clustering gave better results (Kumar and Sodhi [23]). Bal *et al.* [6] segment brain tumors using *Rough-Fuzzy C-Means* (RFCM). They also implemented the K-means method in preprocessing steps. The authors find higher accuracy than HCM (*Hard C-Means*) and FCM. Sumathi *et al.* [37] detect brain tumors from MRI images with the help of kernel Fuzzy C-Means with PSO. They found tumor exaction accuracy of the model is 97.6%. Hemamalini *et al.* [16] segment the fruit images using K-means clustering. After that, they classified fruit images into damaged or not using machine learning approaches such as KNN, SVM, and C4.5. Gu [15] proposed K-means segmentation based on Improved Grey Wolf Optimizer for image segmentation. Saifullah *et al.* [35] proposed K-means segmentation based on Lab Color space for Embryo Detection in Incubated Eggs. Zhu [48] segment SAR images by Efficient Fuzzy C-Means framework with Adaptive Generalized Likelihood Ratio Nonlocal spatial information Embedded.

Now-a-day, neural network is used in many areas. Image classification, image recognition, and image segmentation are some areas where we used neural networks. There are many neural networks available for different works. Our research is based on segmentation on CNN. So, here we present some latest papers based on segmentation through CNN. Avenash and Vishwanath [5] made U-HardNet using CNN and Hard-Swish for semantic segmentation of Satellite Images. Li *et al.* [26] proposed MPCNN for nut image segmentation. Wang *et al.* [39] segment remote sensing images on *Fully Convolutional Network* (FCN) by using encoder-decoder architecture. Amer [3] segment medical images using *Multi-Scale Dilated Attention* (MDA-Unet). Li *et al.* [25] proposed optimized *Deep CNN* (DCNN) for the segmentation of urban scenes for Vehicle Visual Sensors. Huang *et al.* [19] segment ultrasound images by combined use of U-Net and VGG16. Anjum *et al.* [4] made a 13-layer 3-D semantic segmentation model for the segmentation of skin images. Chen *et al.* [8] propose *Error Feature Back-projection based Local-Global* (EB-LG) for the classification and segmentation of 3-D point cloud images. Kim and Heo [21] segment the cityscapes dataset using Spatio-Channel Dilated Convolution. Ho and Lin [17] segment person from some image datasets using CNN with Dilated Convolutions. Liu *et al.* [27] segment images based on Dilated Convolution and Multi-layer Feature Fusion. Tian *et al.* [38] proposed conditional convolutional for Instance and panoptic segmentation. Deo *et al.* [10] proposed an ensemble CNN-based algorithm for iron ore pellet size analysis. Kumawat and Raman [24] proposed a Local phase U-Net for fundus image segmentation. Zhou *et al.* [47] proposed LTS-Net for lung tissue segmentation from CT images. Man *et al.* [29]

segment Liver and Tumor from CT and MRI images using five types of CNN. Kim *et al*. [22] segment *Cambridge Driving labeled Video Dataset* (CamVid) and *Stanford Background Dataset* (SBD) using Grouped Dilated Convolutional Module. Mehta *et al*. [31] made a network based on CNN for semantic segmentation. Zhang *et al*. [45] proposed MFENet: Multi-Feature Extraction Net for remote sensing semantic segmentation. Piao and Lio [32] made an improved UNet network based on Dilated Convolution. Alghodhaifi *et al*. [2] made two CNN for predicting invasive ductal carcinoma in breast histology images. Zhang *et al*. [44] segment CT images of COVID-19 patients using a *multiscale dilated convolutional network* (MSDC-Net). Wei *et al*. [42] proposed revisiting Dilated Convolution for image segmentation.

2. Activation Functions

We increase the power of the Artificial Neural Networks with the help of Activation Functions (AFs). Performances of the networks are also increased by using AFs. Linear activation function and Non-linear activation function are two types of AFs. The linear activation function is useful for linear separability types of problems. Saturating Linear, Linear, Symmetric Saturating Linear, Positive linear, hard limit, and Symmetrical hard limit are some examples of the linear activation function. The linear AF is an unbounded function and output depends upon input value. In Table 1, we have presented some activation functions, their mathematical equations, derivatives and range.

Table 1. Different types of linear AFs, their mathematical equations, derivatives, and range

Sr.	Activation function	Equation	Derivative	Range
1	Linear or purelin	$f(n) = n$	$f'(n) = 1$	unbounded
2	Saturating Linear or satlin	$f(n) = \begin{cases} 0, & n = 0; \\ n, & 0 = n = 1; \\ 1, & n = 1 \end{cases}$	$f(n) = \begin{cases} 0, & n = 0; \\ 1, & 0 = n = 1; \\ 0, & n = 1 \end{cases}$	(depends on conditions)
3	Symmetric Saturating Linear or satlins	$f(n) = \begin{cases} -1, & n = -1; \\ n, & 0 = n = 1; \\ 1, & n = 1 \end{cases}$	$f(n) = \begin{cases} 0, & n = -1; \\ 1, & -1 = n = 1; \\ 0, & n = 1 \end{cases}$	(depends on conditions)
4	Positive linear	$f(n) = \begin{cases} n, & 0 = n; \\ 0, & n = 0 \end{cases}$	$f'(n) = \begin{cases} 1, & 0 = n; \\ 0, & n = 1 \end{cases}$	bounded
5	Hard limit	$f(n) = \begin{cases} 1, & 0 = n; \\ -1, & \text{otherwise} \end{cases}$	$f'(n) = \begin{cases} 1, & 0 = n; \\ -1, & \text{otherwise} \end{cases}$	bounded
6	Symmetric Hard-limit or hardlims	$f(n) = \begin{cases} 1, & 0 = n; \\ -1, & \text{otherwise} \end{cases}$	$f'(n) = \begin{cases} 1, & 0 = n; \\ -1, & \text{otherwise} \end{cases}$	bounded

Table 2. Different types of non-linear AFs, their mathematical equations, derivatives and range

Sr. No.	Activation function	Equation	Derivative	Range
1	Logsigmoid or Sigmoid	$f(n)=\frac{1}{1+e^{-n}}$	$f'(n)=\frac{e^{-n}}{(1+e^{-n})^2}$	$(0,1)$
2	Tansigmoid or Tanh	$f(n)=\frac{e^n-e^{-n}}{e^n+e^{-n}}$	$f'(n)=\frac{4e^{2n}}{(1+e^{2n})^2}$	$(-1,1)$
3	NewSigmoid	$f(n)=\frac{e^n-e^{-n}}{\sqrt{2(e^{2n}+e^{-2n})}}$	$f'(n)=\frac{e^n+e^{-n}}{\sqrt{2(e^{2n}+e^{-2n})}}-\frac{4(e^{-2n}-e^{2n})(e^{-n}-e^n)}{2(2e^{2n}+2e^{-2n})^{\frac{3}{2}}}$	$(-0.7071,0.7071)$
4	ISigmoid	$\alpha(x-a)+\text{Sigmoid}(a);\ x\geq a$ $\text{Sigmoid}(x);\ -a<x<a$ $\alpha(x+a)+\text{Sigmoid}(a);\ x<=-a$	$\alpha;\ \lvert x\rvert >= a$ $\text{Sigmoid}'(x);\ \lvert x\rvert<a$ where 'a' is the threshold α is the slop, and both of them are preset	(depends on conditions)
5	Bi-sig1	$f(n)=\frac{1}{2}\left(\frac{1}{1+e^{-n+1}}+\frac{1}{1+e^{-n-1}}\right)$	$f'(n)=\frac{1}{2}\left(\frac{e^{1-n}}{(1+e^{1-n})^2}+\frac{e^{-1-n}}{(1+e^{-n-1})^2}\right)$	$(0,1)$
6	Bi-sig2	$f(n)=\frac{1}{2}\left(\frac{1}{1+e^{-n}}+\frac{1}{1+e^{-n-1}}\right)$	$f'(n)=\frac{1}{2}\left(\frac{e^{-n}}{(1+e^{-n})^2}+\frac{e^{-1-n}}{(1+e^{-n-1})^2}\right)$	$(0,1)$
7	Bi-tanh1	$f(n)=\frac{1}{2}\left(\tanh\left(\frac{n}{2}\right)+\tanh\left(\frac{n+1}{2}\right)\right)+0.5$	$f'(n)=\frac{1}{4}\left(\text{sech}^2\left(\frac{n+1}{2}\right)+\text{sech}^2\left(\frac{n}{2}\right)\right)$	$(-0.5,1.5)$
8	Bi-tanh2	$f(n)=\frac{1}{2}\left(\tanh\left(\frac{n-1}{2}\right)+\tanh\left(\frac{n+1}{2}\right)\right)+0.5$	$f'(n)=\frac{1}{4}\left(\text{sech}^2\left(\frac{n+1}{2}\right)+\text{sech}^2\left(\frac{n-1}{2}\right)\right)$	$(-0.5,1.5)$
9	Elliott	$f(n)=\frac{0.5n}{1+\lvert n\rvert}+0.5$	$f'(n)=\frac{0.5}{(1+\lvert n\rvert)^2}$	$(0,1)$

Table 2 *contd.*

Sr. No.	Activation function	Equation	Derivative	Range
10	Modified Elliott	$f(n)=\frac{n}{(1+n^2)^{\frac{1}{2}}}+0.5$	$f'(n)=\frac{0.5}{(1+n^2)^{\frac{3}{2}}}$	$(-0.5,1.5)$
11	Gaussian	$f(n)=e^{-n^2}$	$f'(n)=-2ne^{-n^2}$	$(0,1)$
12	Rootsig	$f(n)=\frac{n}{1+\sqrt{1+n^2}}+0.5$	$f'(n)=\frac{1}{1+n^2+\sqrt{1+n^2}}$	$(-0.5,1.5)$
13	Sech	$f(n)=\frac{2}{e^n+e^{-n}}$	$f'(n)=\frac{2(e^n-e^{-n})}{(e^n+e^{-n})^2}$	$(0,1)$
14	ReLU	x; $x>=0$; 0; $x<0$	1; $x>=0$ 0; $x<0$	$(0,\infty)$
15	RelTanh	$\tanh'(\lambda^+)(x-\lambda^+)+\tanh(\lambda^+)$; $x>=\lambda^+$ $\tanh(x)$; $\lambda^-\langle x\rangle=\lambda^+$ $\tanh'(\lambda^-)(x-\lambda^-)+\tanh(\lambda^-)$; $x<=\lambda^-$ $\lambda^+_{\text{lower}}<=\lambda^+<=\lambda^+_{\text{upper}}$ and $\lambda^-_{\text{lower}}<=\lambda^-<=\lambda^-_{\text{upper}}$	$\tanh''(\lambda^+)$; $x>=\lambda^+$ $\tanh'(x)$; $\lambda^-\langle x\rangle=\lambda^+$ $\tanh''(\lambda^-)$; $x<=\lambda^-$	(depends on conditions)
16	Modified activation function	$f(n)=\frac{e^{\frac{2n}{3}}-1}{(1+e^{2n})^{\frac{1}{3}}}$	$f'(n)=\frac{2e^{\frac{2n}{3}}}{3(1+e^{2n})^{\frac{1}{3}}}-\frac{2e^{2n}(e^{\frac{2n}{3}}-1)}{3(1+e^{2n})^{\frac{4}{3}}}$	$(-1,1)$
17	Swish	$f(n)=n.\alpha(n)$ where $\alpha(n)=\frac{1}{1+e^{-n}}$	$f'(n)=f(n)+\alpha(n).(1-f(n))$	(unbound above and bounded below)
18	Hexpo	$-a(e^{-\frac{x}{b}}-1)$, if $x=0$ $f(n)=c(e^{\frac{x}{b}}-1)$, if otherwise	$\frac{a}{b}e^{-\frac{x}{b}}$, if $x=0$ $f'(n)=\frac{c}{d}e^{\frac{x}{b}}$, if otherwise	$(-c,a)$

We generally use the non-linear AF in non-separability problems for getting better performance. The Elliot symmetric sigmoid (elliotsig), logsig, tanh, and softmax are some examples of non-linear AF. There is a lot of AFs present in these times. One AF is not suitable for every problem. The tansigmoid and logsigmoid do not always achieve better performance. If tansigmoid and logsigmoid do not give satisfactory results then we use some other activation function for getting better results. We have to use different AF for numerous problems to achieve better performance. For example, Qin *et al.* obtained a classification accuracy of 93.73% with the help of sigmoid, and 95.56% with the help of Improved logistic sigmoid (Isigmoid) on the MNIST dataset (Qin *et al.* [33]). The equation of an improved logistic sigmoid (Isigmoid) AF is shown in Table 2 (Qin *et al.* [33]). The equation of a ReLU AF is shown in Table 2. ReLU does not use exponential or trigonometric function so does not give a local effect. In the case of ReLU AF, the neurons will be deactivated if the output of the linear transformation is less than 0 (i.e., for the negative value). The equation of a RelTanh AF is shown in Table 2. This RelTanh AF is an improved AF of tanh. Wang *et al.* obtained 96.15% testing accuracy using RelTanh on the faulty dataset of CWRU (Wang *et al.* [40]). In Table 2, we have presented some activation functions, their mathematical equations, derivatives, and range.

Equation (5) is our NewSigmoid AF and equation (6) is our root2sigmoid AF.

$$f1(n) = \frac{e^{n} - e^{-n}}{\sqrt{2(e^{2n} + e^{-2n})}}, \tag{5}$$

$$f2(n) = \frac{(\sqrt{2})^{n} - (\sqrt{2})^{-n}}{2\sqrt{2}(2(2)^{2n} + 2(\sqrt{2})^{-2n})^{\frac{1}{2}}}, \tag{6}$$

where n is the input value and $f(n)$ is the output value. We put $\sqrt{2} = 1.414$ for this segmentation purpose.

NewSigmoid and root2sigmoid AFs have the following four properties:

Nonlinear. NewSigmoid and root2sigmoid AF use exponential function (equations (5) and (6)) like tanh and logsigmoid AFs. So, we can use these AFs for solving nonlinear problems.

Range and Shape. The range of the NewSigmoid AF is $[-0.7071, 0.7071]$. This range is finite. As per Table 3, if we fix the output value to the four decimal places, then root2sigmoid AF becomes a bounded range and its range becomes $[-0.25, 0.25]$. If we don not fix its four decimal places, then root2sigmoid AF becomes an unbounded range. Vanishing gradient and slow convergence happen when we take bounded values. The shape of both new AFs is 'S'.

Zero Centered. NewSigmoid and Root2sigmoid are zero-centered activation function because it shows symmetrical on both sides of zero. So like tanh AF, this function works better than logsigmoid. The logsigmoid is not symmetric around zero, but root2sigmoid and NewSigmoid AFs are symmetric around zero (as like tansigmoid). So, due to this reason, we may use these AFs for solving very complex non-linear problems; such as text, images, videos, audios or any other high dimensionality problems.

Continuously Differentiable. NewSigmoid and root2sigmoid AFs are continuously differentiable functions. Equation (7) is a derivative of NewSigmoid AF and equation (8) is a derivative of root2sigmoid AF, we can use both AFs in gradient-based Optimization methods.

Table 3. Obtained output values using root2sigmoid and tanh AFs

Sr. No.	root2sigmoid	tanh
1	0.111779751	0.7615942
2	0.18188608	0.9640276
3	0.217062425	0.9950548
4	0.233933946	0.9993293
5	0.242093731	0.9999092
6	0.246093684	0.9999877
7	0.248072602	0.9999983
8	0.249056776	0.9999998
9	0.249547596	1
10	0.249792723	1
11	0.249915233	1
12	0.249976485	1
13	0.250007114	1
14	0.250022432	1
15	0.250030093	1
16	0.250033925	1
17	0.250035841	1
18	0.2500368	1
19	0.250037279	1
20	0.250037519	1

$$f1'(n) = \frac{e^n + e^{-n}}{\sqrt{2(e^{2n} + e^{-2n})}} - \frac{4(e^{-2n} - e^{2n})(e^{-n} - e^n)}{2(2e^{2n} + 2e^{-2n})^{\frac{3}{2}}}, \tag{7}$$

$$\begin{aligned} f2'(n) = {} & \frac{(\sqrt{2})^n \cdot \log(\sqrt{2}) + (\sqrt{2})^{-n} \cdot \log(\sqrt{2})}{2\sqrt{2}(2(\sqrt{2})^{2n} + 2(\sqrt{2})^{-2n})^{\frac{1}{2}}} \\ & - \frac{[4(\sqrt{2})^{-2n} \cdot \log(\sqrt{2}) - 4(\sqrt{2})^{2n} \cdot \log(\sqrt{2})] * [(\sqrt{2})^{-n} - (\sqrt{2})^n]}{4\sqrt{2}(2(\sqrt{2})^{-2n} + 2(\sqrt{2})^{2n})^{\frac{3}{2}}}. \end{aligned} \tag{8}$$

Suppose $n = 0$, then the value of equation (5) is as under:

$$f1(0) = \frac{e^0 - e^{-0}}{\sqrt{2(e^{2*0} + e^{-2*0})}} = 0 \tag{9}$$

and the value of equation (7) is as under:

$$f1'(n) = \frac{e^0 + e^{-0}}{\sqrt{2(e^{2*0} + e^{-2*0})}} - \frac{4(e^{-2*0} - e^{2*0})(e^{-0} - e^0)}{2(2e^{2*0} + 2e^{-2*0})^{\frac{3}{2}}} = \frac{2}{2} - 0 = 1. \tag{10}$$

On the basis of equations (9) and (10), we can say $f1'(n)$ is continuous at 0. Therefore, NewSigmoid AF has an approximate identity at the origin point. During the optimization process, this AF becomes easier as compared to the logistic function.

During the optimization process, the root2sigmoid AF may give high accuracy with respect to tansigmoid (tanh), due to giving more variations as compared to tansigmoid (see Table 3). So this AF is more useful for other networks also such that RNN or LSTM.

3. Methodology

Deep learning is a modern branch of machine learning. We train the network of deep learning from its experience. *Convolutional Neural Network* (CNN) (Goodfellow *et al*. [14]) is a part of an *Artificial Neural Network* (ANN). CNN is made with the help of multiple convolutional layers and one or multiple fully connected layers. For example: VGG-16, Alexnet, VGG-19, etc. Therefore, CNN has two parts. The first part consists of many multiple layers of convolutional layers, pooling layers, activation layers, and normalization layers, and the second part consists fully connected layer, softmax layer, and pixelClassification layer or classification layer. In this research, we have also made two CNNs. Because CNN takes more time for running a network (sometimes more days), so we have made a small network (maximum of 13-layer CNN). A CNN has mainly three layers. These layers are the Convolutional layer, the Pooling layer, and the Fully connected layer.

3.1 Convolutional Layer

We do convolutional operations on the convolutional layer. In the convolutional operation, we do dot products between input (learnable parameters) and kernel. Mainly we used three types of convolutional operation in convNets. They are 1-D convolutional operation, 2-D convolutional operation, and 3-D convolutional operation.

1-D convolutional operation. Equation (11) is a 1-D convolutional operation in the case of continuous data,

$$s(t) = (x * w)(t) = \int x(a)w(t-a)da\,, \tag{11}$$

where w is a kernel, x is the input, t is the time index, and a is the age of measurement.

Equation (12) is a 1-D convolutional operation in the case of discrete data is

$$s(t) = (x * w)(t) = \sum_{x=-\infty}^{\infty} x(a)\cdot w(t-a)\,. \tag{12}$$

2-D convolutional operation. Equation (13) is a 2-D convolutional operation in the case of discrete data is

$$s(i,j) = (I * K)(i,j) = \sum_m \sum_n I(m,n)\cdot k(i-m,\ j-n)\,. \tag{13}$$

3-D convolutional operation is used for a colour images.

3.2 Pooling Layer

The output values are changed with the help of this layer at a certain location. This location is a summary statistic of the nearby output. Generally, we used two types of pooling layers in the CNN. The first is average (avg) pooling and the second is maximum (max) pooling. For a 2-D image, we used 2-D avg pooling or max pooling. For a 3-D image, we used 3-D avg pooling or max pooling.

Suppose a matrix is given as under

2	3
3	4

Then, the avg pooling $= avg(2,3,3,4) = 3$, and the max pooling value $= \max(2,3,3,4) = 4$.

3.3 Fully Connected Layer

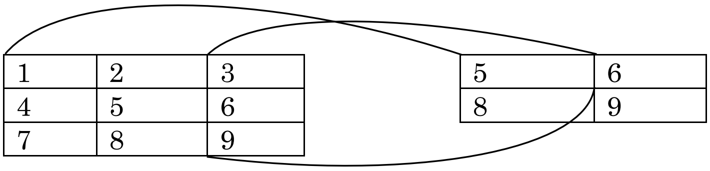
This is the last layer of a CNN. The neurons of this layer are fully connected with all other neurons. Function of this layer is similar to the MLFFNN.

Structure of a CNN (convNet). Figure 1 is a typical structure of a layer of a convNet.

```
layer=[
    imageInputlayer([64 64 3])
    convolution2dlayer(5,32, ‘padding’,1)
    batchnormalizationlayer
    maxPooling2dlayer(2, ‘stride’, 2)
    fullyconnectedlayer(10)
    softmaxlayer
    classificationlayer or pixelclassificationLayer
    ]
```

Figure 1. A typical structure of convNet or CNN

The first part of this layer is an image input layer. In this figure, the size of the image is given. The height is 64, the width is 64 and the number of colors (or channels) is 3. If we have taken a gray image then we have to take 1 instead of 3. The second part of this layer is the convolution2dlayer. A convolutional layer consists of filters and stride, dilated convolution, feature maps, zero padding, output size, number of neurons, learning parameters, and number of layers. A feature region is learned by a filter size. A filter size consists of a set of weights that is applied to that region. In Figure 1 the size of a filter is $5*5$ and the number of filters is 32. A stride is a step size. Filter moves with the help of a stride value.



1	2	3
4	5	6
7	8	9

5	6
8	9

Figure 2. Function of max pooling

Figure 2 is an example of max pooling with stride 1.

The number of weights of a filter $= h*w*c$, where h is the height, w is the weight and c is the number of channels.

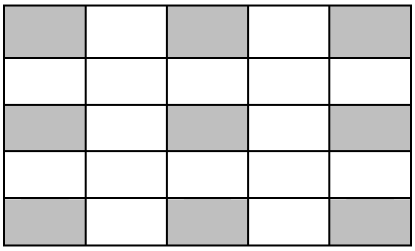


Figure 3. Dilated convolution with factor size [2 2]

The expansion of the spaces between the elements of the filter is decided by a dilated convolutional value. Figure 3 is an example of dilated convolutional with factor size [2 2]. In this figure, the size of the figure is $3*3$. So, the effective filter size of this image is [5 5] with zeros between the elements. The equation of the effective filter size is calculated with the help of equation (14).

$$\text{effective\_filter\_size} = ((\text{filter}_{\text{size}} - 1)\cdot \text{dilation\_factor} + 1 = (3-1)*2+1 = 5. \tag{14}$$

All the weights and biases are consists of feature maps.

If the bias value is 1, then the total number of inputs or parameters (n) is

$$n = (h*w*c+1)*\text{number\_of\_filters}. \tag{15}$$

For controlling the output size we apply the padding value. The output size of a convolutional layer is calculated with the help of equation (16).

$$\text{output\_size} = (\text{input\_size} - (\text{filter}_{\text{size}} - 1)*\text{dilation\_factor} + 1) + 2*\text{padding})/\text{stride} + 1. \tag{16}$$

The total number of neurons in a convolutional layer is equation (17)

$$\text{total} = \text{map\_size} * \text{number\_of\_filters}. \tag{17}$$

The third part of Figure 1 is batch normalization. Batch normalization is used for the speeding up a convolutional neural networks and reducing the sensitivity. The fourth part of this figure is ReLU layer. This is a non-linear AF. There are many types of non-linear AF. ReLU, tanh, and swish are some famous non-linear AF used in CNN. We have presented an explanation of some non-linear AF in Section 2. The fifth part of Figure 1 is maxpooling 2-D layer. The size of this maxpooling is [2 2] and the stride value is 2. The sixth part of this figure is a fully connected layer. The size of the fully connected layer is 10. The seventh part of Figure 1 is the softmax layer. The last part of Figure 1 is the classification layer. A classification layer computes the cross entropy loss value. For a continuous value, we use a regression layer in place of a classification layer. Half mean square error loss values are calculated within this layer. Half-mean-square error is calculated with the help of equation (18).

$$\text{loss} = \frac{\sum_{i=1}^{R}(t_i - y_i)^2}{2}. \tag{18}$$

After designing the CNN, we train the network using a solver and the number of epochs. There are many types of solvers. Out of them, 'adam' (adaptive moment estimation), 'rmsprop' (root mean square), and 'sgdm' (stochastic gradient descent with momentum) are famous solvers. The classification layer is used for the classification of the datasets. A pixel classification layer provides a categorical label for each image pixel or voxel. This layer is used for segmentation purposes.

4. Experiments

We have done this experiment on Intel Core i7, Window 10, and MATLAB 2022b [2].

Algorithm: We did the following four steps for this research:

Step 1 (*Datasets*): We had taken 2 types of datasets. These datasets were triangle Figure 4.a2, and Camvid Figure 4.b[3]. As per Table 4, we have taken a number of samples for training and testing purposes. Output values for both datasets are segmented images. The CamVid dataset has 32 classes. Some of these classes are traffic, light, sky, road, pole, luggage, bicyclist, pedestrian, and car. Out of 32 classes, we have taken four classes for this research. These classes are sky, pole, pedestrian, and bicyclist.

Step 2 (*Network Selection*): In all the above two datasets input images and corresponding segment images are given. For segmenting our datasets on CNN, we had taken two new activation functions. They are NewSigmoid and root2sigmoid AF. We made different layers of CNN for the different datasets. As per Table 5, we made 12-layer CNN for the triangle segment and 13-layer CNN for the CamVid segment.

Step 3 (*Selecting activation function*): As per Table 5, we compare our NewSigmoid and root2sigmoid AF at the sr. no. 7 of both CNN. One-by-one we have taken NewSigmoid, relu, root2Sigmoid, and tanh activation function at sr. no. 7. At sr. no. 14, we have taken the Softmax activation function, which is used for the multi-classification model.

Step 4 (*Run the network*): Here we train/run the network with sgdm algorithm. We had taken a maximum number of epochs is 30 for the CamVid dataset and 300 for the triangle dataset, so that our network stops training/running; when the network reaches these epochs. We kept the default value of other parameters, because, in this paper, we want to show the performance of our activation function with compare to the other activation function on CNN in the image segmentation cases.

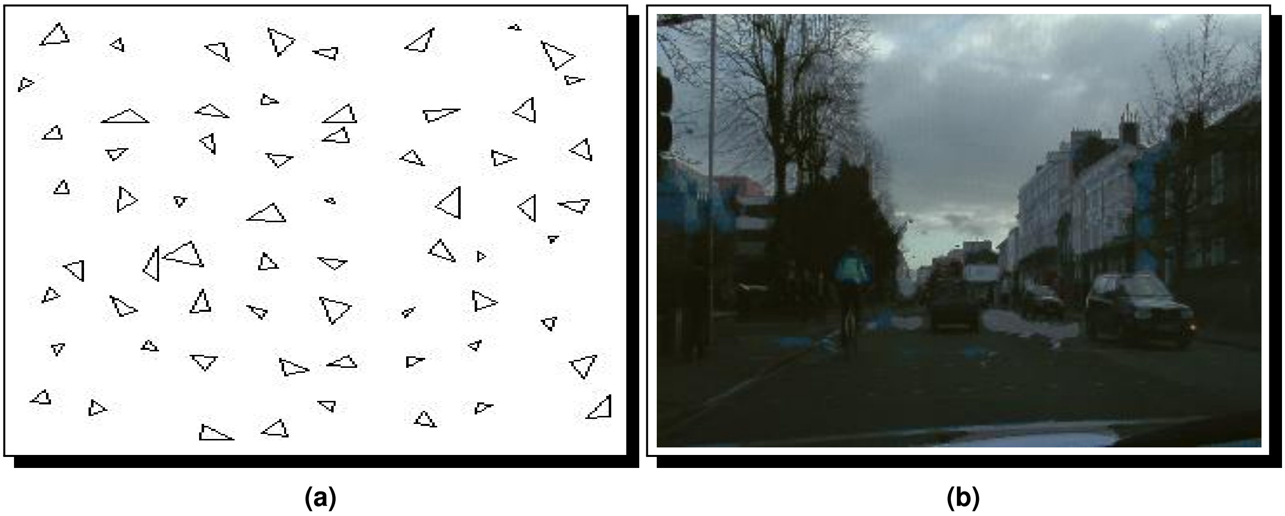


Figure 4. (a) Triangle database, (b) CamVid dataset

[2] MATLAB program, URL: https://www.mathworks.com/help/deeplearning.

[3] CamVid dataset, URL: https://www.kaggle.com/datasets/carlolepelaars/camvid.

Table 4. Training and testing samples of different datasets

Datasets	Number of samples for training purpose	Number of samples for testing purpose
Triangle	200	100
CamVid	33	11

Table 5. Different CNN layer for different datasets

Sr. No.	Layer of CNN	Size (S) and number (n) of CNN	
		triangle dataset (12-layer CNN)	CamVid dataset (13-layer CNN)
1	ImageInput layer	$S = [32\ \ 32\ \ 1]$	$S = [720\ \ 960\ \ 3]$
2	Convultion2d layer	$S = [3\ \ 3]$ and $n = 32$	$S = [5\ \ 5]$ and $n = 32$
3	Relu	(not taken)	Relu
4	batchnormalization	batchnormalization	batchnormalization
5	Convultion2d layer	$S = [3\ \ 3]$ and $n = 32$	$S = [5\ \ 5]$ and $n = 32$
6	batchnormalization	batchnormalization	-(not taken)
7	**NewSigmoid/relu/root2sigmoid/tanh**	**NewSigmoid/relu/root2sigmoid/tanh**	**NewSigmoid/relu/root2sigmoid/tanh**
8	batchnormalization	-(not taken)	batchnormalization
9	Convultion2d layer	$S = [3\ \ 3]$ and $n = 32$	$S = [5\ \ 5]$ and $n = 32$
10	Relu	-(not taken)	Relu
11	batchnormalization	batchnormalization	batchnormalization
12	Relu	Relu	-(not taken)
13	Convultion2d layer	Convultion2d layer	Convultion2d layer
14	Softmax	Softmax	Softmax
15	pixelClassification	pixelClassification	pixelClassification

5. Results and Discussion

Segmentation is wholly dependent upon user requirements. Application to application we can change the ground truth values. With the help of two (triangle and CamVid) image datasets, for getting better segmentation, we used *Convolutional Neural Network* (CNN) using different types of activation functions. After completing the above training process we have taken accuracy and some other metrics. We measured the Global Accuracy, Mean Accuracy, Mean IoU, Weight IoU, and MeanBF Score after completing the training on the above four activation functions.

Global accuracy: Suppose, correctly classified pixels is P and the total number of pixels is Q, then

$$\text{Global accuracy} = \frac{P}{Q}. \tag{19}$$

Mean Accuracy: Suppose, TP is the value of the True Positive and FN is the value of the False Negative, then

$$\text{accuracy} = \frac{TP}{TP+FN}. \tag{20}$$

The average accuracy of all four taken classes is known as mean accuracy. IoU (*Intersection over Union*) formula is

$$\text{IoU} = \frac{TP}{TP+FP+FN}, \tag{21}$$

where FP means false positive.

The average IoU of all these four classes is known as MeanIoU. The average weighted IoU of all four classes which are assigned by a number of pixels is known as WeightIoU. The contour matching score of all true values with predicted values is known as the boundary F1 (BF) value. The average BF score value is known as the MeanBF Score value. In Table 6, we showed these results.

Table 6. Obtained metrics with the help of the four activation functions on CNN

Activation functions	Global accuracy	Mean accuracy	Mean IoU	Weight IoU	MeanBF Score
For Triangle Image					
NewSigmoid AF	0.9629	0.9734	0.7560	0.9422	0.5792
Relu AF	0.9577	0.9706	0.7368	0.9354	0.5536
root2Simgoid AF	**0.9702**	0.9773	0.7864	0.9519	0.6175
Tanh AF	0.9647	0.9745	0.7630	0.9445	0.5987
For CamVid Image					
NewSigmoid AF	0.9397	0.6845	0.4711	0.9174	0.6452
Relu AF	0.9401	0.6533	0.4591	0.9172	0.6423
root2Simgoid AF	**0.9419**	0.6802	0.4741	0.9192	0.6415
Tanh AF	0.9413	0.6712	0.4688	0.9189	0.6524

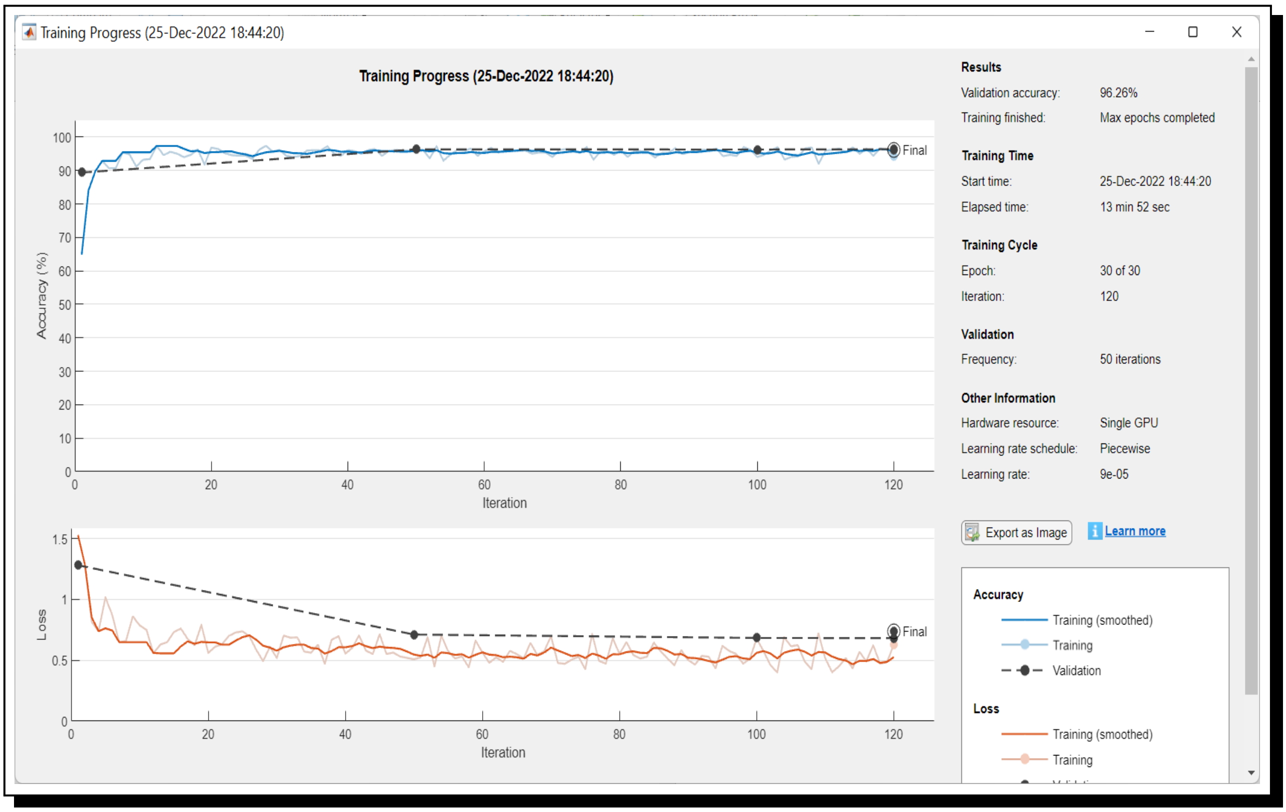
In the case of the triangle image dataset: Through NewSigmoid AF we found Global Accuracy is 96.29%, Mean Accuracy is 97.34%, Mean IoU is 75.60%, Weight IoU is 94.22%, and MeanBF Score is 57.92%. Through Relu AF we found Global Accuracy is 95.77%, Mean Accuracy is 97.06%, Mean IoU is 73.68%, Weight IoU is 93.54%, and MeanBF Score is 55.36%. Through `root2sigmoid` AF we found Global Accuracy is 97.02%, Mean Accuracy is 97.73%, Mean IoU is 78.64%, Weight IoU is 95.19%, and MeanBF Score is 61.75%. Through tanh AF we found Global Accuracy is 96.47%, Mean Accuracy is 97.45%, Mean IoU is 76.30%, Weight IoU is 94.45%, and MeanBF Score is 59.87%.

In the case of the CamVid image dataset: Through NewSigmoid AF we found Global Accuracy is 93.97%, Mean Accuracy is 68.45%, Mean IoU is 47.11%, Weight IoU is 91.74, and MeanBF Score is 64.52%. Through Relu AF we found Global Accuracy is 94.01%, Mean Accuracy is 65.33%, Mean IoU is 45.91%, Weight IoU is 91.72%, and MeanBF Score is 64.23%. Through `root2sigmoid` AF we found Global Accuracy is 94.19%, Mean Accuracy is 68.02%, Mean IoU is 47.41%, Weight IoU is 91.92%, and MeanBF Score is 64.15%. Through tanh AF we found Global Accuracy is 94.13%, Mean Accuracy is 67.12%, Mean IoU is 46.88%, Weight IoU is 91.89%, and MeanBF Score is 65.24%.

For more satisfaction, we also check the validation accuracy of the CamVid dataset. We take 10% of images (11 images) for validation purposes. For the same dataset and same CNN network validation accuracy of relu is 96.04%, root2simgoid is 96.26%, tanh is 96.20 and NewSimgois is 96.26%. In Figure 5, we show the training progress of CNN on the CamVid dataset obtained by the root2sigmoid activation function.

From all the above results, we can say that root2sigmoid performs better results with compare to NewSigmoid, Relu, and tanh AF in the case of image segmentation on CNN.

In Figure 6, we show an original image and its segmented image of the triangle dataset. In Figure 7, we show an original image and its segmented image of the CamVid dataset.



Results
Validation accuracy: 96.26%
Training finished: Max epochs completed

Training Time
Start time: 25-Dec-2022 18:44:20
Elapsed time: 13 min 52 sec

Training Cycle
Epoch: 30 of 30
Iteration: 120

Validation
Frequency: 50 iterations

Other Information
Hardware resource: Single GPU
Learning rate schedule: Piecewise
Learning rate: 9e-05

Figure 5. Training progress of CNN on CamVid dataset obtained by root2sigmoid activation function. The validation accuracy, in this case, is 96.26%

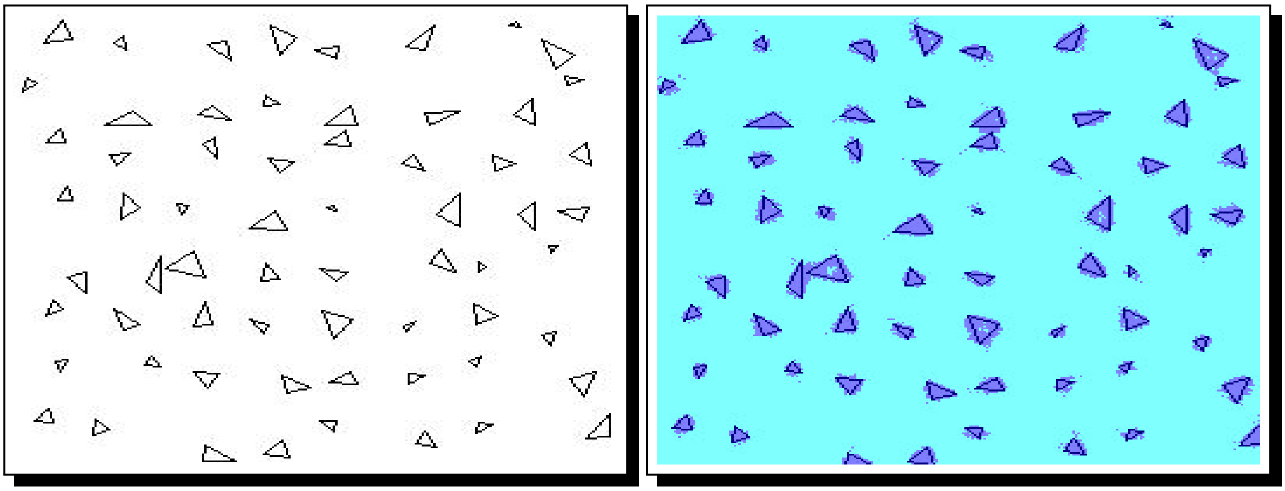


Figure 6. (a) original image, (b) segmented image of the triangle dataset

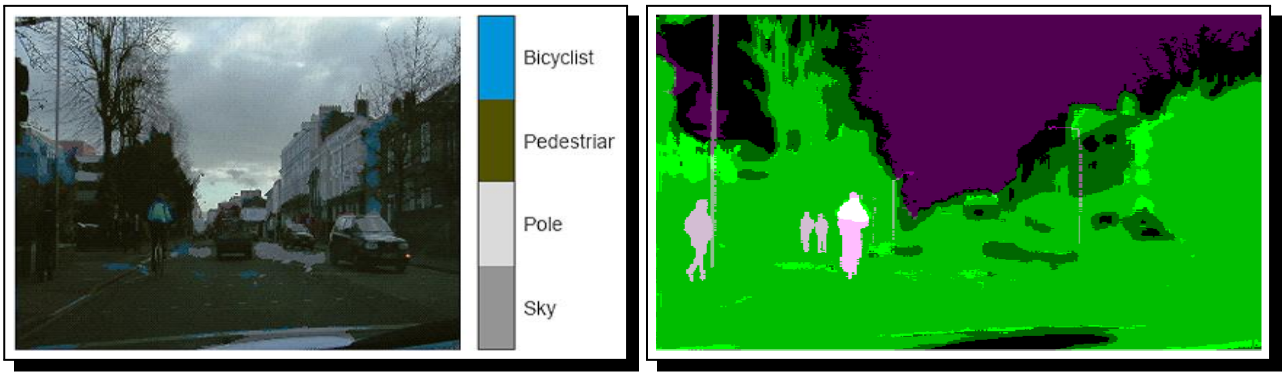


Figure 7. (a) original image, (b) segmented image of the CamVid dataset

6. Conclusion and Future Work

In this research paper, we have segmented the image on CNN using four types of activation functions. We used NewSigmoid, Relu, root2sigmoid, and Tanh on CNN. We have compared these four AFs on the same dataset and same networks for finding the better AF on CNN for image segmentation purposes. NewSigmoid, root2sigmoid, and Tanh are exponential activation functions. Our other main aim of this research is to implement some new activation functions in CNN for image segmentation purposes. We succeeded to find this result. We successfully implement our Newsigmoid and root2sigmoid AF in CNN for segments of the images. We made 12-layer CNN for the triangle segment and 13-layer CNN for the CamVid segment. After that, we compare all these four activation functions on both CNNs. On the triangle image dataset, we have achieved global accuracy of 96.29% through NewSigmoid AF, 95.77% through Relu AF, 97.02% through root2sigmoid AF, and 96.47% through Tanh AF. On the CamVid dataset, we have achieved global accuracy of 93.97% through NewSigmoid AF, 94.01% through Relu AF, 94.19% through root2sigmoid AF, and 94.13% through Tanh AF. We also check the validation accuracy of the CamVid dataset. We take 10% of images for validation purposes. For the same dataset and same CNN network, we found validation accuracy of relu is 96.04%, root2simgoid is 96.26%, tanh is 96.20% and NewSimgois is 96.26%. Therefore, we may say that for the same dataset and the same network the root2sigmoid AF gives better results as compared with NewSigmoid, Relu, and Tanh AFs.

In the future, we may use both AFs for image segmentation in CNN or any other neural network.

Competing Interests

The authors declare that they have no competing interests.

Authors' Contributions

All the authors contributed significantly in writing this article. The authors read and approved the final manuscript.

References

[1] S. Adhikary, S. P. Tiwari and S. Banerjee, Realtime oil spill detection by image processing of synthetic aperture radar data, *2022 IEEE International Conference on Electronics, Computing and Communication Technologies* (CONECCT), Bangalore, India (2022), pp. 1 – 5, DOI: 10.1109/CONECCT55679.2022.9865768.

[2] H. Alghodhaifi, A. Alghodhaifi and M. Alghodhaifi, Predicting invasive ductal carcinoma in breast histology images using convolutional neural network, in: *IEEE National Aerospace and Electronics Conference* (NAECON 2019), Dayton, OH, USA (2019), pp. 374 – 378, DOI: 10.1109/NAECON46414.2019.9057822.

[3] A. Amer, T. Lambrou and X. Ye, MDA-Unet: A multi-scale dilated attention U-net for medical image segmentation, *Applied Sciences* **12**(7) (2022), 3676, DOI: 10.3390/app12073676.

[4] M. A. Anjum, J. Amin, M. Sharif, H. U. Khan, M. S. A. Malik and S. Kadry, Deep semantic segmentation and multi-class skin lesion classification based on convolutional neural network, *IEEE Access* **8** (2020), 129668 – 129678, DOI: 10.1109/ACCESS.2020.3009276.

[5] R. Avenash and P. Viswanath, Semantic segmentation of satellite images using a modified CNN with hard-Swish activation function, in: *Proceedings of the 14th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications*, Vol. **4** (2019), 413 – 420, Prague, Czech Republic, DOI: 10.5220/0007469604130420.

[6] A. Bal, M. Banerjee, A. Chakrabarti and P. Sharma, MRI brain tumor segmentation and analysis using rough-fuzzy c-means and shape based properties, *Journal of King Saud University-Computer and Information Sciences* **34**(2) (2022), 115 – 133, DOI: 10.1016/j.jksuci.2018.11.001.

[7] J. Canny, A computational approach to edge detection, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **PAMI-8**(6) (1986), 679 – 698, DOI: 10.1109/TPAMI.1986.4767851.

[8] J. Chen, Y. Zhang, F. Ma and Z. Tan, EB-LG module for 3D point cloud classification and segmentation, *IEEE Robotics and Automation Letters* **8**(1) (2023), 160 – 167, DOI: 10.1109/LRA.2022.3223558.

[9] E. R. Davies, *Computer Vision: Principles, Algorithms, Applications, Learning*, 5th edition, Academic Press, 858 pages (2018), DOI: 10.1016/C2015-0-05563-0.

[10] A. J. Deo, A. Sahoo, S. K. Behera and D. P. Das, Iron ore pellet size analysis: A machine learning-based image processing approach, *IEEE Industry Applications Magazine* **29**(1) (2023), 67 – 79, DOI: 10.1109/MIAS.2022.3214020.

[11] I. Despotović, B. Goossens and W. Philips, MRI segmentation of the human brain: Challenges, methods, and applications, *Computational and Mathematical Methods in Medicine* **2015** (2015), Article ID 450341, 23 pages, DOI: 10.1155/2015/450341.

[12] R. C. Gonzalez and R. E. Woods, *Digital Image Processing*, 3rd edition, Pearson Education, Inc., New Jersey, xxii + 954 pages (2008), URL: https://dl.ebooksworld.ir/motoman/Digital.Image.Processing.3rd.Edition.www.EBooksWorld.ir.pdf.

[13] R. C. Gonzalez, R. E. Woods and S. L. Eddins, *Digital Image Processing using MATLAB*, 2nd edition, Tata McGraw-Hill (2010), URL: https://in.mathworks.com/academia/books/digital-image-processing-using-matlab-gonzalez-in.html.

[14] I. Goodfellow, Y. Bengio and A. Courville, *Deep Learning*, MIT Press, 800 pages (2016), URL: https://mitpress.mit.edu/9780262035613/deep-learning.

[15] S. Gu, K-means image segmentation based on improved grey wolf optimizater, in: *3rd International Conference on Electronic Communication and Artificial Intelligence* (IWECAI 2022), Zhuhai, China (2022), pp. 467 – 471, DOI: 10.1109/IWECAI55315.2022.00096.

[16] V. Hemamalini, S. Rajarajeswari, S. Nachiyappan, M. Sambath, T. Devi, B. K. Singh and A. Raghuvanshi, Food quality inspection and grading using efficient image segmentation and machine learning-based system, *Journal of Food Quality* **2022** (2022), Article ID 5262294, 6 pages, DOI: 10.1155/2022/5262294.

[17] D. J. Ho and Q. Lin, Person segmentation using convolutional neural networks with dilated convolutions, in: *Proceedings of the IS&T International Symposium on Electronic Imaging: Imaging and Multimedia Analytics in a Web and Mobile World*, (2018), 455-1 – 455-7, DOI: 10.2352/ISSN.2470-1173.2018.10.IMAWM-455.

[18] F. Hoseini and A. Shahbahrami, An efficient implementation of fuzzy C-means and watershed algorithms for MRI segmentation, in: *8th International Symposium on Telecommunications* (IST2016), Tehran, Iran (2016), pp. 178 – 184, DOI: 10.1109/ISTEL.2016.7881806.

[19] A. Huang, Q. Wang, L. Jiang and J. Zhang, Automatic segmentation of median nerve in ultrasound image by a combined use of u-net and VGG16, in: *IEEE International Ultrasonics Symposium* (IUS 2021), Xi'an, China (2021), pp. 1 – 4, DOI: 10.1109/IUS52206.2021.9593861.

[20] T. Kanungo, D. M. Mount, N. S. Netanyahu, C. D. Piatko, R. Silverman and A. Y. Wu, An efficient k-means clustering algorithm: analysis and implementation, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **24**(7) (2002), 881 – 892, DOI: 10.1109/TPAMI.2002.1017616.

[21] J. Kim and Y. S. Heo, Efficient semantic segmentation using spatio-channel dilated convolutions, *IEEE Access* **7** (2019), 154239 – 154252, DOI: 10.1109/ACCESS.2019.2949076.

[22] D. S. Kim, Y. H. Kim and K. R. Park, Semantic segmentation by multi-scale feature extraction based on grouped dilated convolution module, *Mathematics* **9**(9) (2021), 947, DOI: 10.3390/math9090947.

[23] A. Kumar and S. S. Sodhi, Comparative analysis of fuzzy c-means and k-means clustering in the case of image segmentation, in: *8th International Conference on Computing for Sustainable Global Development* (INDIACom 2021), New Delhi, India (2021), pp. 194 – 200, URL: https://ieeexplore.ieee.org/document/9441121.

[24] S. Kumawat and S. Raman, Local phase U-net for fundus image segmentation, in: *IEEE International Conference on Acoustics, Speech and Signal Processing* (ICASSP 2019), Brighton, UK (2019), pp. 1209 – 1213, DOI: 10.1109/ICASSP.2019.8683390.

[25] Y. Li, J. Shi and Y. Li, Real-time semantic understanding and segmentation of urban scenes for vehicle visual sensors by optimized DCNN algorithm, *Applied Sciences* **12**(15) (2022), 7811, DOI: 10.3390/app12157811.

[26] J. Li, L. Su, J. Huang, Y. Ma, L. Liu and K. Luo, A modified pulse coupled neural network model for nut image segmentation, in: *Chinese Automation Congress* (CAC 2018), Xi'an, China (2018), pp. 2554 – 2558, DOI: 10.1109/CAC.2018.8623754.

[27] J. Liu, Z. Wu, Y. Hong, G. Zhong and M. Liu, Image semantic segmentation based on dilated convolution and multi-layer feature fusion, in: *IEEE International Conference on Artificial Intelligence and Industrial Design* (AIID 2021), Guangzhou, China (2021), pp. 339 – 342, DOI: 10.1109/AIID51893.2021.9456560.

[28] J. MacQueen, Some methods for classification and analysis of multivariate observations, *Berkeley Symposium on Mathematical Statistics and Probability* **1967** (1967), 281 – 297, URL: https://projecteuclid.org/ebooks/berkeley-symposium-on-mathematical-statistics-and-probability/Proceedings-of-the-Fifth-Berkeley-Symposium-on-Mathematical-Statistics-and/chapter/Some-methods-for-classification-and-analysis-of-multivariate-observations/bsmsp/1200512992?tab=ChapterArticleLink.

[29] L. Man, H. Wu, J. Man, X. Shi, H. Wang and Q. Liang, Machine learning for liver and tumor segmentation in ultrasound based on labeled CT and MRI images, in: *IEEE International Ultrasonics Symposium* (IUS 2022), Venice, Italy (2022), pp. 1 – 4, DOI: 10.1109/IUS54386.2022.9957634.

[30] J. Mao, Z. Cao, H. Wang, B. Zhang, Z. Guo and W. Niu, Agricultural robot navigation path recognition based on K-means algorithm for large-scale image segmentation, *2019 14th IEEE Conference on Industrial Electronics and Applications* (ICIEA), Xi'an, China (2019), pp. 1233 – 1237, DOI: 10.1109/ICIEA.2019.8834296.

[31] D. Mehta, A. Skliar, H. B. Yahia, S. Borse, F. Porikli, A. Habibian and T. Blankevoort, Simple and efficient architectures for semantic segmentation, in: *IEEE / CVF Conference on Computer Vision and Pattern Recognition Workshops* (CVPRW 2022), New Orleans, LA, USA, (2022), 2627 – 2635, DOI: 10.1109/CVPRW56347.2022.00296.

[32] S. Piao and J. Liu, Accuracy improvement of UNet based on dilated convolution, *Journal of Physics: Conference Series* **1345**(5) (2019), 052066, DOI: 10.1088/1742-6596/1345/5/052066.

[33] Y. Qin, X. Wang and J. Zou, The optimized deep belief networks with improved logistic sigmoid units and their application in fault diagnosis for planetary gearboxes of wind turbines, *IEEE Transactions on Industrial Electronics* **66**(5) (2019), 3814 – 3824, DOI: 10.1109/TIE.2018.2856205.

[34] A. A. Rafique, A. Jalal and A. Ahmed, Scene understanding and recognition: statistical segmented model using geometrical features and gaussian Naïve bayes, *2019 International Conference on Applied and Engineering Mathematics* (ICAEM), Taxila, Pakistan (2019), pp. 225 – 230, DOI: 10.1109/ICAEM.2019.8853721.

[35] S. Saifullah, R. Dreżewski, A. Khaliduzzaman, L. K. Tolentino and R. Ilyos, K-means segmentation based-on lab color space for embryo detection in incubated egg, *Jurnal Ilmiah Teknik Elektro Komputer dan Informatika* **8**(2) (2022), 175 – 185, DOI: 10.26555/jiteki.v8i2.23724.

[36] S. Samaddar and A. R. Reddy, Comparative study of image segmentation techniques on chronic kidney diseases, *International Journal of Pure and Applied Mathematics* **118**(14) (2018), 235 – 239, URL: https://acadpubl.eu/jsi/2018-118-14-15/articles/14/34.pdf.

[37] R. Sumathi, A. Saravanan, P. Nagaraj and R. Venkatesh, Detection of tumor region in MRI images using kernel fuzzy c means with PSO, in: *International Conference on Computer Communication and Informatics* (ICCCI 2022), Coimbatore, India (2022), pp. 1 – 6, DOI: 10.1109/ICCCI54379.2022.9741007.

[38] Z. Tian, B. Zhang, H. Chen and C. Shen, Instance and Panoptic segmentation using conditional convolutions, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **45**(1) (2023), 669 – 680, DOI: 10.1109/TPAMI.2022.3145407.

[39] L. Wang, R. Li, C. Duan, C. Zhang, X. Meng and S. Fang, A novel transformer based semantic segmentation scheme for fine-resolution remote sensing images, *IEEE Geoscience and Remote Sensing Letters* **19** (2022), Article number 6506105, DOI: 10.1109/LGRS.2022.3143368.

[40] X. Wang, Y. Qin, Y. Wang, S. Xiang and H. Chen, ReLTanh: An activation function with vanishing gradient resistance for SAE-based DNNs and its application to rotating machinery fault diagnosis, *Neurocomputing* **363** (2019), 88 – 98, DOI: 10.1016/j.neucom.2019.07.017.

[41] J. Wang, Z. Xu and Y. Liu, Texture-based segmentation for extracting image shape features, *2013 19th International Conference on Automation and Computing*, London, UK (2013), pp. 1 – 6, URL: https://ieeexplore.ieee.org/document/6662034.

[42] Y. Wei, H. Xiao, H. Shi, Z. Jie, J. Feng and T. S. Huang, Revisiting dilated convolution: a simple approach for weakly- and semi-supervised semantic segmentation, in: *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Salt Lake City, UT, USA (2018), pp. 7268 – 7277, DOI: 10.1109/CVPR.2018.00759.

[43] Y. Zhang, M. Brady and S. Smith, Segmentation of brain MR images through a hidden Markov random field model and the expectation-maximization algorithm, *IEEE Transactions on Medical Imaging* **20**(1) (2001), 45 – 57, DOI: 10.1109/42.906424.

[44] J. Zhang, X. Ding, D. Hu and Y. Jiang, Semantic segmentation of COVID-19 lesions with a multiscale dilated convolutional network, *Scientific Reports* **12** (2022), Article number: 1847, DOI: 10.1038/s41598-022-05527-x.

[45] C. Zhang, X. Lu, Q. Ye, C. Wang, C. Yang and Q. Wang, MFENet: Multi-feature extraction net for remote sensing semantic segmentation, in: *7th International Conference on Intelligent Computing and Signal Processing* (ICSP 2022), Xi'an, China (2022), pp. 1986 – 1990, DOI: 10.1109/ICSP54964.2022.9778622.

[46] Y. Zheng, B. Jeon, D. Xu, Q. M. Wu and H. Zhang, Image segmentation by generalized hierarchical fuzzy C-means algorithm, *Journal of Intelligent & Fuzzy Systems* **28**(2) (2015), 961 – 973, DOI: 10.3233/IFS-141378.

[47] L. Zhou, X. Xu, K. Zhou and J. Guo, LTS-NET: Lung tissue segmentation from CT images using fully convolutional neural tetwork, in: *11th International Conference on Information Science and Technology* (ICIST 2021), Chengdu, China (2021), pp. 125 – 131, DOI: 10.1109/ICIST52614.2021.9440594.

[48] J. Zhu, F. Wang and H. You, SAR image segmentation by efficient fuzzy c-means framework with adaptive generalized likelihood ratio nonlocal spatial information embedded, *Remote Sensing* **14**(7) (2022), 1621, DOI: 10.3390/rs14071621.