

Fine-tuning U-net for medical image segmentation based on activation function, optimizer and pooling layer

Remah Younis¹, Rawan Ghnemat², Jaafer Al Saraireh³

Computer Science Department, King Hussein Faculty for Computing Sciences, Princess Sumaya University for Technology, Amman, Jordan

Article Info

Article history:

Received Oct 10, 2022

Revised Jan 2, 2023

Accepted Feb 4, 2023

Keywords:

Activation function

Fine-tune

Image segmentation

U-net

X-ray images

ABSTRACT

U-net convolutional neural network (CNN) is a famous architecture developed to deal with medical images. Fine-tuning CNNs is a common technique used to enhance their performance by selecting the building blocks which can provide the ultimate results. This paper introduces a method for tuning U-net architecture to improve its performance in medical image segmentation. The experiment is conducted using an x-ray image segmentation approach. The performance of U-net CNN in lung x-ray image segmentation is studied with different activation functions, optimizers, and pooling-bottleneck-layers. The analysis focuses on creating a method that can be applied for tuning U-net, like CNNs. It also provides the best activation function, optimizer, and pooling layer to enhance U-net CNN's performance on x-ray image segmentation. The findings of this research showed that a U-net architecture worked supremely when we used the LeakyReLU activation function and average pooling layer as well as RMSProb optimizer. The U-net model accuracy is raised from 89.59 to 93.81% when trained and tested with lung x-ray images and uses the LeakyReLU activation function, average pooling layer, and RMSProb optimizer. The fine-tuned model also enhanced accuracy results with three other datasets.

This is an open access article under the [CC BY-SA](https://creativecommons.org/licenses/by-sa/4.0/) license.



Corresponding Author:

Jaafer Al Saraireh

Computer Science Department, King Hussein Faculty for Computing Sciences, Princess Sumaya

University for Technology

Amman, Jordan, 11941

Email: j.saraireh@psut.edu.jo

1. INTRODUCTION

Convolutional neural network (CNN) is a known class of artificial neural networks (ANN) usually used in image analysis. CNN is the multi-layer architecture built by stacking alternative layers of filters to extract features and details from images, so conclusions such as classification, recognition, or segmentation are conducted. U-net-shaped architectures have been used in medical image segmentation as a reliable architecture. The essential advantage of U-net CNNs is their ability to increase the resolution of extracted features by acquiring the up-sampling process [1].

Medical image analysis with artificial intelligence (AI) tools has emerged as a helping tool in biomedical research and healthcare. Data collected from different sources are deployed to train AI models, which are later used to aid in diagnostic analysis. The data, usually images collected from different sources with variant specifications such as size, type, and resolution, should be processed and prepared for training and testing. Collecting a sufficient amount of data is also difficult due to privacy issues associated with patients' medical information. One such common utilization of AI in studying medical images is image

segmentation. Image segmentation aims to extract specific information from medical images, such as tumors or locating bleeding vessels. Since such information is not likely to be easily identified with the abstract eye, image segmentation applications are used to find corresponding information [2].

AI models used in medical data analysis can tolerate not much shortage in accuracy; due to the nature of actions built over them. For example, diagnostic decisions might lead to a therapy plan or surgical actions. Hence AI models used in medical applications should be reliable and accurate. At the time, researchers put much effort into analyzing the problem from a medical point of view. AI models used in studying the medical problem are not given the same degree of attention, leading to contradictory decisions between different models or even incomplete decisions [3].

This paper covers the need for adopting a CNN architecture that creates an optimal model for medical image segmentation using U-net CNNs. The performance of architecture varies with different activation functions and optimizers. Additionally, the performance of U-net CNNs is highly dependent on the type of pooling layer located between the feature selection side of the architecture and the up-sampling side.

This work is motivated by many facts, such as that a lousy selection of an activation function in CNNs can lead to vanishing or exploding gradient problems. At the same time, a good choice of optimizer can lead to faster and more accurate learning models. The pooling layer in U-net architectures lies between the architecture's feature selection and up-sampling sides. Hence it has a crucial role in the performance quality of the U-net. U-net architectures should be studied and analyzed to be driven into a comfortable area of trust so that they can be helpful in medical institutions with more confidence. CNN architectures like U-net consist of multiple components that need to be adequately assembled; thus, a perfect use can be achieved.

This work analyzes U-net CNN segmentation results in the context of accuracy and loss value with different shapes of the U-net CNN's varying activation function, the optimizer function, and the pooling layer. The data set used in the analysis consists of 5,500 x-ray images of lungs, collected to train models of value for coronavirus disease (COVID-19) diagnostics. Then three other datasets were used to investigate and confirm the study. What distinguishes this work is the deep analysis and research on the effect of parameters such as activation function and optimization method used in consecutive layers of the CNN architecture. This work presents guidelines for researchers to design an efficient U-net architecture for medical image segmentation which was not frankly covered in other related works. The proposed methodology was detailed to be generalized and used with different CNN architectures.

This paper is organized as: in the next section, the related works are discussed, are clarified. Then, section 3 includes the presentation of the methodology followed throughout this work. After that, in section 4, an analysis of the results is presented. Finally, the conclusion comes in section 5.

2. RELATED WORK

Medical image analysis using AI models has achieved many achievements and successes in the past few years. U-net-based AI systems have proved their ability to lift the performance of AI in medical research and diagnostic platforms. After the U-net model has shown proof of success in medical image analysis, especially when it comes to segmentation applications, many research works were built on it.

As cleared in Table 1, U-net CNNs were used with many applications in medical image segmentation. The work presented in [4] provided a U-net CNN model, which focuses on bladder segmentation from computed tomography (CT) images. A new model was suggested by [5]; they improved the model by enhancing the original U-net architecture, so the efficiency of the segmentation method outperformed U-net and ResNet101 architectures. U-net CNN segments nuclear magnetic resonance (NMR) images to create a model of left ventricular segmentation [6]. Moreover, the list continues to grow with exceptional results accomplished with the help of U-net CNNs, such as [2], [7]–[9], and many others, which presented models that aim to provide an efficient model for medical image segmentation in a specific field. Every model presented was prepared and designed to help provide the best results.

Table 1. Studies conducted on medical images using U-net CNNs

Reference	Application
[4]	Bladder segmentation
[5]	Tumour segmentation
[6]	Left ventricular segmentation
[7], [10]	Retinal thickness segmentation
[10]	Knee segmentation for age assessment
[8]	Cardiac images segmentation
[11]	Nerve Segmentation
[2]	Lung segmentation
[9]	Brain tumor segmentation
[12]	Chest x-ray image segmentation

A directed towards tuning U-net architecture applied on ultrasound images was presented by [1]. Their work focused on the impact of the layers used in a U-net CNN on its performance. Activation functions are the fundamental source of non-linearity in AI models, as shown in Table 2. Famous activation functions used in CNNs and introduced each were [13] presented. The performance of a CNN can vary depending on the activation function used [13]. The impact of activation function in AI models was investigated in [14], where they proved that choosing the proper activation function in a neural network implementation leads to accelerated training and improved performance. The performance of deep networks with different trending activation functions was presented by [15].

Table 2. Studies which analyzed activation function, optimizes pooling layers, and impact on a CNN

Reference	Contribution	Limitations
[13]	Presented and compared famous activation functions	The discussion took activation functions famously into consideration only. The performance of the activation function was not studied with standard datasets and architectures.
[14]	Proved that choosing the right activation function leads to improved performance. They also showed that tuning the initialization parameters and the activation function can accelerate the training and improve performance.	The results showed implications for Bayesian neural networks.
[15]	Studied the impact of activation function in a CNN	The performance of activation. The function was not studied with standard datasets and architectures.
[16]	Studied the impact of activation function face recognition applications and presented a new activation function that worked perfectly with the facial expression dataset.	None
[17]	Presented an Activation function for image classification	None
[18]	Optimizers' effect on a CNN was analyzed	The performance of only four optimizers was studied with a simple CNN only.
[19]	Analyzed the effect of seven optimizers on CNN models	None
[20]	Investigated the effect of optimizers on CNN models	The CNN they used in the study is not a famous or state-of-the-art CNN.
[21]	Analyzed the effect of optimizers in a plant disease classification model	The best nominated CNN takes a considerable amount of training time.
[22]	Analyzed the effect of optimizers in the image recognition model	Only three optimizers were studied.
[23]	Using dropout on the input to max-pooling layers of CNNs was studied	Neither CNN nor the dataset was famous in literature.
[24]	Suggested that pooling layers can act as a feature extraction layer in CNNs	None
[25]	Used average and max pooling layers of brain tumor segmentation	A complicated model requires high computations.
[26]	Global average and max pooling were analyzed, and a new pooling strategy was introduced.	Datasets with full, colored images did not achieve satisfying training results.
[27]	Analyzed the effect of pooling layers on image recognition models	None

Activation functions used with face recognition models were investigated in [16]. They also designed their special version of the activation function for image segmentation applications. This means that a specific category of images requires special requirements for medical images. An activation function that can be used with image classification was presented in [17]. Studies such as [28], [29] gave a deep analysis; of the cons and pros of widely used activation functions.

On the other hand, optimizers' effect on a CNN was studied and analyzed in the literature. Adagrad, Proximal Adagrad, Adam, and RMSProp optimizers were studied in [18], where adaptive moment estimation (ADAM) and RMSProp are believed to enhance the results collected by training an AI model with 1,200 medical images. While [19] analyzed the effect of seven optimizers on CNN models of hyperspectral remote sensing image (HSI) classification, they ended up nominating AdaMax over Adam. The study by [20] also agreed on the best performance being to AdaMax, as the survey was conducted on (HSI) images. For example, the work of [21] studied optimizers' impact in a plant disease classification model. At the same time, a new approach was conducted to study the effect of the optimizer chosen in image recognition models [22].

Pooling layers used in CNNs were investigated in [23]. They also aimed to understand the dropout effect in the input propagated into pooling layers. In the work of Bailer *et al.* [24], suggested that pooling layers can act as feature extraction layers in CNNs, and how this use of pooling layers can fasten the feature extraction process. In medical image applications, [25] located average and max pooling layers inside their

architecture to enhance the process of brain tumor segmentation. Global average and max pooling were analyzed and explored in [26], [27]; they also proposed their version of a deep generalized max pooling layer. Amiri *et al.* [1] focused on fine-tuning U-net CNN for ultrasound image segmentation and put effort into modifying the deep layers of the CNN to get satisfying results. A summary of the mentioned works is provided in Table 2.

In 2021 the research presented in [30] has proposed an automatic method through which the U-net CNN can fine-tune itself to adapt to the used dataset. The nnU-net in [30] has shown satisfying results regarding the accuracy of segmentation with many datasets. Yet, the nnU-net pretraining steps, as well as the training steps, are expensive regarding time and hardware usage. The nn-UNet is a general advanced approach that can be used with many medical images. This study proposes a manual fine-tuning process for the U-net architecture, which we believe can be used when limited time and hardware resources are available. Also, when the U-net is tuned to be used with a limited type of medical images.

On the other hand, at the time, practical and functional studies were conducted to fine-tune CNN architectures by focusing on a specific CNN component, such as activation functions. We, in this work, present a methodology to fine-tune a CNN architecture considering many components, such as activation function optimizers and pooling layers. The result is directed toward the state-of-the-art U-net CNNs, widely used in medical applications, and tailors an optimized architecture. Different datasets were used with various contents and sizes. Most of the studies mentioned in Table 2 focus on only one component of CNN. Many of them also were conducted on non-famous CNNs. In comparison, none of the studies conducted on famous CNN focused on U-net.

The current study presents an interest in the activation function influence on the performance of CNNs, and it is steered toward finding the best activation function for U-net architectures used in medical image segmentation. Unlike other images, medical images tend to be unclear, with lousy resolution and contrast features [31]. Hence medical images need further effort and study to be helpful in AI analysis. This paper presents a general methodology that can be used to design and implement the U-net CNN for optimized implementation. Activation functions, optimizers, and some essential role layers are investigated thoroughly in a specific order.

3. PROPOSED METHOD FOR FINE-TUNING U-NET

A U-net network is composed of down-sampling and up-sampling processes. Down-sampling is where the feature extraction comes from; a thumbnail for the image is generated with the deep image features included. Deep features extracted in the down-sampling process are enlarged during the up-sampling process. The bottleneck layer comes between the down and up-sampling processes, usually a pooling layer containing the most semantic features.

3.1. Activation functions

Activation functions are used in a CNN to add non-linearity to the network; without an activation function, the network will become a linear representation of all the data included in the training. It also might result in vanishing or exploding gradient problems. The vanishing gradient problem means that some weights in the network are receiving a negligible number of updates on their values, which means that it would take a long time before it reaches a sufficient value. The exploding gradient, on the other hand, means that the values of the weights are receiving a significant update, sometimes causing weights to become larger than 1. Since the relation between the input and the output in a CNN is not linear, not using a proper activation function will result in an erroneous representation of the network. Tensorflow provides many activation functions that can be utilized for building an AI model. In this paper, we have trained the same U-net architecture with these functions to find the activation function which results in the best metrics.

Some activation functions such as rectified linear unit (ReLU), sigmoid, tanh, scaled exponential linear unit (SELU), LeakyReLU, and ReLU6 are used in the internal layers, while Softmax, SoftPlus, and Swish are used in the output layers. Here we provide a brief discussion for each of them. In later sections, we discuss how we used them throughout the work and our results.

- ReLU function: ReLU or rectified linear activation function is described in (1). The output is the same as the input if the input is a positive value, while the output is zero otherwise. In other words, the ReLU activation function activates neurons with positive input and deactivates others. These functions can bring the problem of dead neurons into the CNN since some neurons will never be activated. On the other hand, it beats the vanishing gradient problem. ReLU is a preferable activation function to models, which comes with many convolutional layers since it has proved its reliability [15], [16].

$$Relu(x) = \max(0, x) \tag{1}$$

- Sigmoid function: Activating or deactivating a neuron reflects whether a neuron map to a feature we are interested in or not. The sigmoid functions value lies between 0 and 1, so it maps to the existence of a feature or absence. It also comes with high computational requirements, unlike ReLU, so it is not likely used. We examined the performance of our U-net CNN since we are highly interested in performance. The mathematical representation of the sigmoid is included in (2) [16].

$$h(x) = \frac{1}{1+e^{-Tx}} \quad (2)$$

- Tanh function is interpreted in (3) and can be considered similar to the sigmoid activation function [16], except that it maps the output to values between -1 and 1. This means a stronger mapping for irrelevant values since they take negative values instead of 0. Tanh is usually used for classification between two classes.

$$h(x) = \frac{1}{1+e^x} \quad (3)$$

- SELU function: the scaled exponential linear unit, or SELU activation function described in (4) [15], aims to normalize all the weights. Hence, they have a mean value of zero and a standard deviation of one.

$$SELU(x) = \max(\lambda * \alpha * e^x - \alpha, \lambda * x) \quad (4)$$

- Leaky ReLU function: it is presented in (5). It is an updated version of ReLU; instead of canceling the effect of negative input, a portion of it is considered. Usually, the output is 1% of the input. When? is any other value than 0.01, it is considered randomized Leaky ReLU [16].

$$leaky_relu(x) = \max(\beta * x, x) \quad (5)$$

- ReLU6: it is the same as ReLU but with a restriction on the max allowed value for the output as presented in (6).

$$Relu6(x) = \min(\max(0, x), 6) \quad (6)$$

- SoftMax: it is a generalized activation function used in the output layers. It is used for multiclass classification. The equation for the SoftMax activation function is shown in (7) [15].

$$Softmax(x_i) = \frac{\exp(x_i)}{\sum \exp(x_j)} \quad (7)$$

3.2. Optimizers

This work investigates the effect of optimizers used in U-net CNN on its performance in terms of accuracy and error. Optimizers update the value of the weights according to the error value calculated at the output layer. This paper interests the optimizers stochastic gradient descent (SGD) RMSprop, Adam, Adadelta, Adagrad, Adamax, Nadam, and Ftrl. Gradient descent (GD) optimization depends on using the first derivative function through the backpropagation process, aiming to drive the error function into its minima. Error function reaches its minima with the correct weights of weights used, so gradient descent updates the weights in a CNN according to the calculated error value.

The error is calculated based on all training examples using the GD optimizer. SGD acts the same as SD but estimates the error depending on randomly selected training examples. SDG accelerates learning time and reacquires less memory space. Gradient descent with momentum algorithms was presented to control the convergence speed into local minima. The required convergence should not be slow so that it would extend learning time. It also should not be fast, so it overshoots local minima. The RMSprop optimizer limits the oscillations in the vertical direction. Hence it is possible to increase the learning rate, and the algorithm can take significant steps in the horizontal direction converging faster. RMSprop and gradient descent differ in how the gradients are calculated.

Adagrad modifies the learning rate for each parameter by working on the derivative of the error. Adagard tunes the learning rate of a CNN well but is also considered computationally expensive. Adagard optimizer suffered from a decaying learning rate problem AdaDelta optimizer was found to deal with the problem by limiting the accumulated past gradients into a predefined value. AdaDelta requires high computational power.

Adam optimizer was built to use the best properties of RmsProp and Adagrad. Like RmsProp, it uses squared gradients to scale the learning rate, and like Adagrad, it depends on the derivative of the gradient to calculate the momentum value. Nadam and AdaMax are improved versions of Adam, which are supposed to give better results. A deeper analysis of optimizers in CNNs is included in [19].

3.3. Pooling layers

Pooling layers are a basic building block in CNNs that aim to summarize the massive amount of data produced in preceding convolutional layers. The rules in selecting data in pooling layers to be passed to the following layers result in different types of pooling. The pooling layers types we are interested in this work are max pooling, average pooling, global max pooling, and global average pooling.

3.4. Methodology

This work focuses on finding the activation function, optimizer, and pooling layer, which enhances U-net CNN performance measured by accuracy and loss error. Figure 1 shows the steps followed through the work. A sequential process is applied to start from analysis to find the activation function that provides the best accuracy and loss error results. Then, the U-net implementation is updated to use the activation function selected from the previous step. The same steps were followed to find the best optimizer; the U-net implementation was modified again with the best optimizer realized. The final step is to find the best pooling layer. It is necessary to state here that the initial implementation of the U-net used the “ReLU” activation function, “Adam” optimizer, and “global max pooling”.

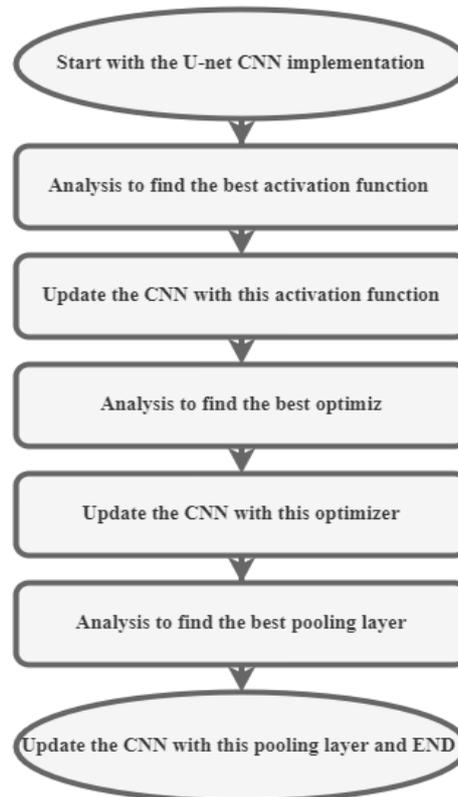


Figure 1. Steps followed through fine-tuning the U-net architecture

An experimental approach is used by varying the activation function in a U-net CNN implementation to find the activation function, which maximizes accuracy and minimizes loss error. Six activation functions are used in this analysis. The activation functions are ReLU, ReLU6, SELU, LeakyReLU, Sigmoid, and TanH. The same experimental approach used to select the activation function is applied for optimizer selection. After finding the activation function with the best performance, the CNN performance is analyzed with different optimizers. The tested optimizers are Adam, AdamMax, Adgrad, Adadelta, RMSProb, Nadam, and Ftrl. After updating the CNN to use, the best optimizer effect of the

pooling layer is investigated. We tested four famous pooling layers, average pooling, max pooling, global average pooling, and global max pooling.

The approach used through the analysis is believed to be an inside-out analysis. It starts from the depth of the CNN presented by the activation function. Activation functions directly impact the weights calculations during the forward propagation of data. Essential decisions are built over the output generated in activation functions. Optimizers' role comes later and is reflected in how errors are calculated and corrected through the backpropagation of data. The backpropagation aims to adjust the error in the calculations resulting from the forward propagation process. Hence optimizers are analyzed after analyzing the optimizer effect. The bottlenecks layer role in U-net CNNs is essential in the performance quality since it lies at the end of the feature selection process, which passes the values believed to be the most important ones into later up-sampling steps. Hence the impact of the pooling layer was also studied and investigated. After passing through the three fine-tuning steps mentioned earlier and listed in Figure 1, we used three other datasets than the one we used to fine-tune the CNN. We trained and tested the U-net initial architecture and the final, fine-tuned architecture with the datasets and compared the accuracy and loss results. As will be shown shortly, the fine-tuned architecture gave better accuracy results.

We have selected the most famous and reliable activation functions. Then we moved to the most famous and used optimizers. We fixed the CNN implementation, used the best activation function, which gave the most accurate results, and redone the test again but with the most famous optimizers. After that, we tested four pooling layers after fixing the U-net CNN with the best optimizer. We are suggesting here in this work to detail our process, which we followed to produce a U-net CNN architecture that gave the best results with medical images such as lung X-rays. We described the process, the tested components, and the results.

4. EXPERIMENTAL RESULTS

The data used in this work is an open public dataset of chest X-rays collected from patients suspected to be COVID-19 positive or infected with other viral and bacterial pneumonia [32]. This work focused only on the segmentation dataset regardless of the diagnostics. The accuracy and loss error in the segmented lung image is measured. Five thousand five hundred lung X-ray images are included in the work, each with the corresponding mask image. One thousand one hundred images are used for validation results, and the rest are used in training. In Figure 2 we present a screen shot for the u-net CNN architecture used in this experiment prior to tuning. We also present an example for the used images.

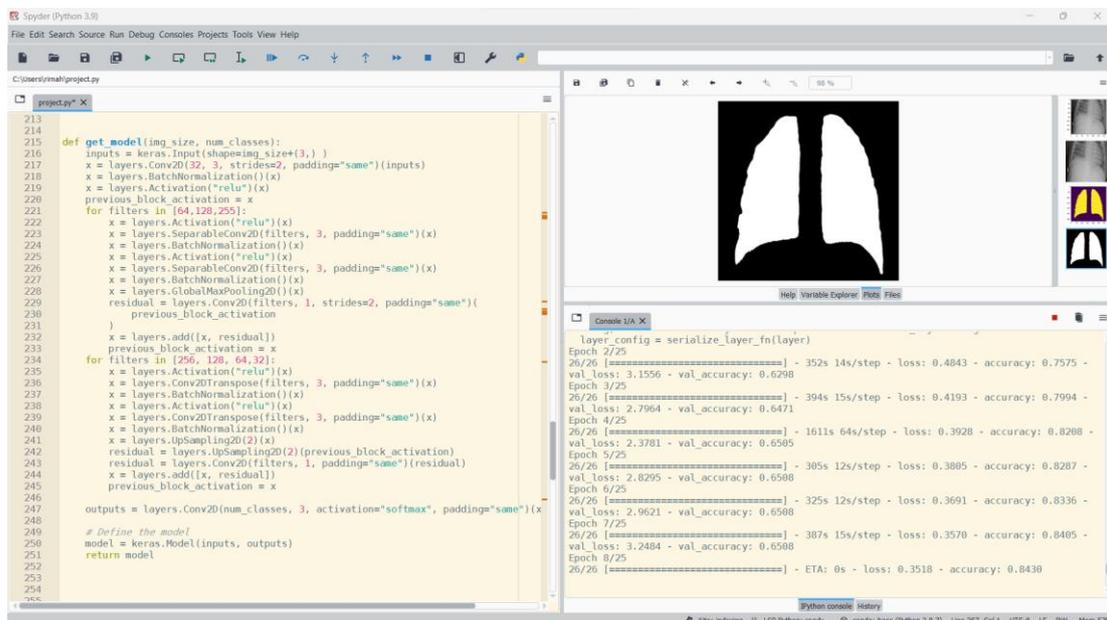


Figure 2. A screen shot for the work environment

The mask images are available in the “.png” file. While image instances are downloaded using the “URL” address included in “.json” files. The JSON file contained much information regarding the diagnosis

and the patient’s survival through the disease. In this study, we are only interested in segmentation results, so information other than X-ray images is not used.

All images were downloaded, resized to 160×160 pixels, and saved as “.jpeg” images before the training started so they would be read from the PC rather than downloaded during training every time the training began. The patch size used throughout the work is 50, and the number of epochs used is 25. The training is done using “Spyder” from the “Anaconda” platform on top of a 2.8 GHz 11th generation Intel core i7 processor and 16 GB RAM. The activation functions’ performance under the same circumstances gives almost similar results. Figure 3 shows that the best accuracy and loss results are generated when “LeakyReLU” is used, while the “Sigmoid” function is the least efficient among the six activation functions. LeakyReLU is created in the first place to overcome the dead neurons problem in CNN in the ReLU activation function SELU and ReLU6 are also designed to enhance the ReLU activation function, and they all give better results than ReLU.

Figure 4 shows the accuracy and loss values with the different optimizers; Adam, Nadam, And Adamax are close results. RmsProp exceeded their performance. The pooling layers effect is shown in Figure 5. The average pooling layer gave the best results and could provide the U-net architecture with the last leap toward better performance. The improvement in experimental results can be viewed in Figure 6. The accuracy is raised from 89.59 to 93.81%. The best results are gained when the U-net architecture is tuned with the LeakyReLU activation function, RmsProp optimizer, and the pooling layer is set to the average pooling layer.

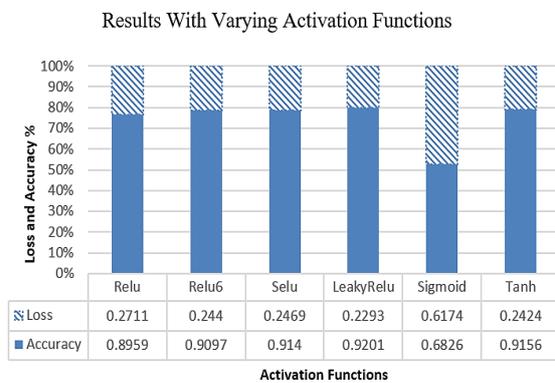


Figure 3. Activation functions performance

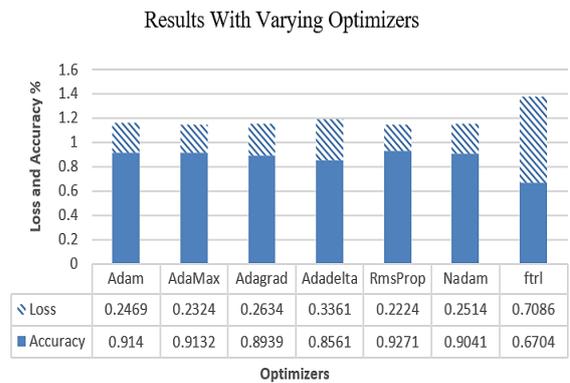


Figure 4. Optimizers performance

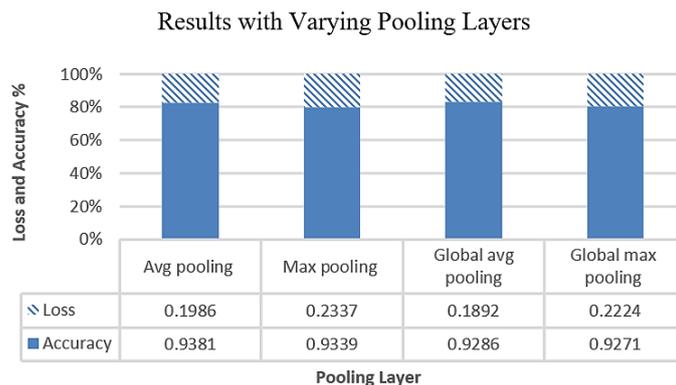


Figure 5. Pooling layers performance

A smaller dataset was used to support and confirm the results shown in Figure 6; the dataset randomly picked 1,000 images from the 6,500 images we used before. The testing experiment started by training the original U-net architecture with the new data set. Then “ReLU” activation function is replaced by “LeakyReLU”. As clarified in Figure 7, the accuracy and loss results are enhanced after applying this step. Then we used the “RmsProp” optimizer instead of “Adam” and recorded the improved results shown in Figure 7. Finally, the pooling layer is set as the “average pooling” layer.

The results shown in Figure 7 confirmed the efficiency of the model produced after tuning parameters and the methodology used. Figure 7 shows how the model’s accuracy is improved after using the LeakyReLU activation function. It is enhanced when the RmsProp optimizer was used and enhanced after using the average pooling layer with another dataset than the one that started our study.

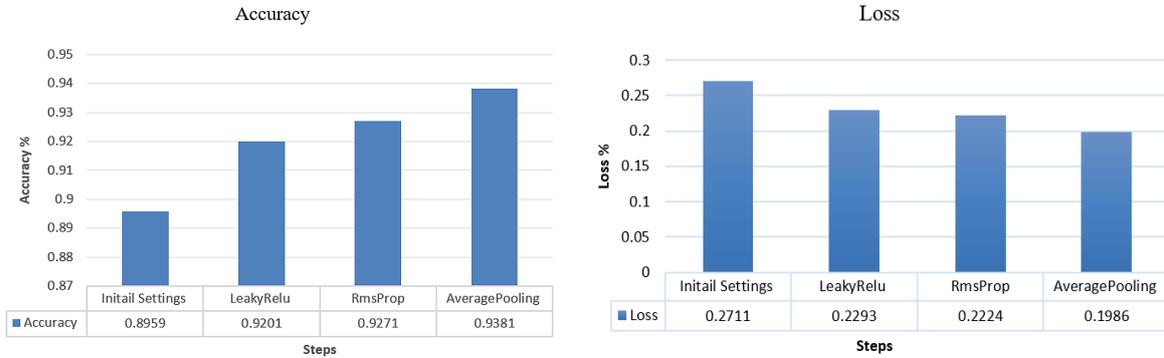


Figure 6. Experimental accuracy and error enhancement

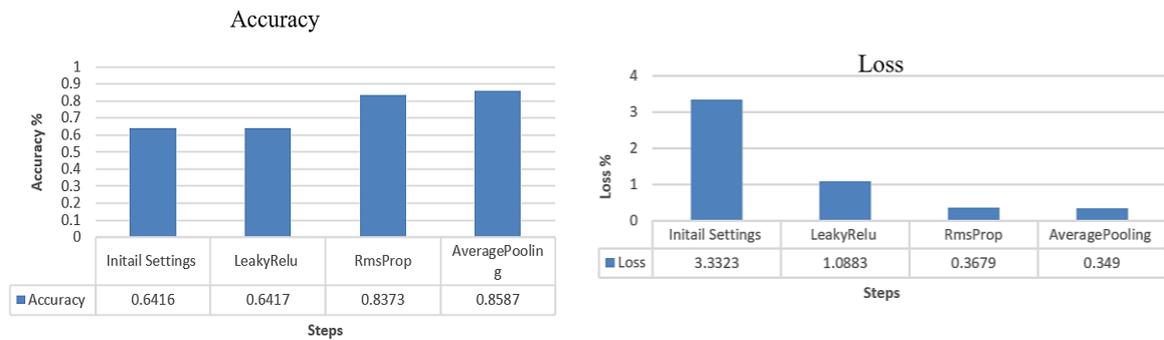


Figure 7. Enhancement results on a different dataset

The final step was to prove the methodology’s efficiency and the enhanced U-net efficiency. CNN’s original and enhanced versions used and trained two other datasets. The data set specifications are mentioned in Table 3. The lung CT-scan images, which are 267 images, were resized from 512×512 to 256×256 pixels; 50 images were used for validation. While for the Breast cancer x-ray images, 58 images only are used, 15 of them are used for validation and their sizes are 896×768 pixels. In Table 3, we also present the results of training the datasets with the non-tuned CNN and the tuned CNN. The table shows that tuning the CNN has improved the model’s accuracy; even when the initial results were not satisfying, these results are improved.

Table 3. Results summary

DataSet	Num of images	Accuracy with the original CNN	Accuracy With the Enhanced CNN
Lung x-ray [33]	5,500	0.8959	0.9381
Lung x-ray [33]	1,000	0.6416	0.8587
Lung CT scan images	267	0.1067	0.3965
Breast Cancer x-ray images [32]	58	0.9822	0.9836

Fine-tuning a CNN is proposed in literature mainly to be used in transfer learning models, where AI models are trained with large datasets and then fine-tuned to be used with smaller datasets and give excellent results. Works such as [34]–[36] proposed fine-tuned models to be used with medical images, but the details of the process of how these models were tuned were not detailed. The works started with AI models whose performance was close to perfect, while the tuning aimed to give similar results to smaller or similar datasets. The work in [37] extended the research so the fine-tuned model would extract more features from the dataset it was trained with before the tuning process took place and work sufficiently with new images.

Finally, we consider the amount of enhancement gained in the accuracy results rather than the result itself. For example, with the 5,500 lung x-ray images, the accuracy started with a value of 89.59% and raised to 93.81%. When a subset of 1,000 images out of the 5,500 images was used, the accuracy increased from 64.16 to 85.87%. This proves that comparing initial with final results, the improvement was of satisfying degree of enhancement.

Finally, a multifold cross-validation process was performed to confirm the efficiency of the fine-tuned architecture. For testing, a 5-fold stratified cross-validation technique is used [38]. The 5-fold validation was applied with two U-net architectures; the u-net architecture which we started with and the architecture which we ended with, so a comparison can be made between the initial and the fine-tuned architectures. The 1,000-picture dataset mentioned in Table 3 was used through this process.

Table 4 shows that the fine-tuned architecture has performed better with all 5-folds. The results with the 5-folds are with low variance as well. It is worth mentioning here that this study aims to fine-tune the U-net architecture to produce a more accurate model. It also shows how the model's accuracy changes with each parameter and component used.

Table 4. Multi-fold validation results

	Fold1	Fold2	Fold3	Fold4	Fold5
Initial architecture	0.8081	0.7078	0.783	0.7682	0.806
Fine-tuned architecture	0.8506	0.8401	0.8444	0.8433	0.8575

5. CONCLUSION

This work presented a methodology for designing and tuning U-net CNN parameters. The methodology gives bold lines that can be followed to enhance U-net CNN performance. The experiment is conducted and validated using x-ray and CT images. Hence the considerable activation function, optimizer, and pooling layer that can be used for medical image segmentation are investigated and presented. The work proves that the LeakyReLU activation function, RmsProp optimizer, and average pooling can be deployed in U-net CNN to enhance its performance in x-ray image segmentation systems.

REFERENCES

- [1] M. Amiri, R. Brooks, and H. Rivaz, "Fine tuning U-Net for ultrasound image segmentation: which layers?," in *Domain Adaptation and Representation Transfer and Medical Image Learning with Less Labels and Imperfect Data: First MICCAI Workshop, DART 2019, and First International Workshop, MIL3D 2019, Shenzhen, Held in Conjunction with MICCAI 2019, Shenzhen, China*, Feb. 2020, pp. 235–242.
- [2] D. Müller, I. S. Rey, and F. Kramer, "Automated chest CT image segmentation of COVID-19 lung infection based on 3D U-Net," *Informatics in Medicine Unlocked*, vol. 25, Jun. 2020.
- [3] J. Wu and S. Shang, "Managing uncertainty in AI-enabled decision making and achieving sustainability," *Sustainability*, vol. 12, no. 21, Oct. 2020, doi: 10.3390/su12218758.
- [4] X. Xu, F. Zhou, and B. Liu, "Automatic bladder segmentation from CT images using deep CNN and 3D fully connected CRF-RNN," *International Journal of Computer Assisted Radiology and Surgery*, vol. 13, no. 7, pp. 967–975, Jul. 2018, doi: 10.1007/s11548-018-1733-7.
- [5] K. Men *et al.*, "Cascaded atrous convolution and spatial pyramid pooling for more accurate tumor target segmentation for rectal cancer radiotherapy," *Physics in Medicine and Biology*, vol. 63, no. 18, Sep. 2018, doi: 10.1088/1361-6560/aada6c.
- [6] Q. Tao *et al.*, "Deep learning-based method for fully automatic quantification of left ventricle function from cine MR images: a multivendor, multicenter study," *Radiology*, vol. 290, no. 1, pp. 81–88, Jan. 2019, doi: 10.1148/radiol.2018180513.
- [7] F. G. Venhuizen *et al.*, "Robust total retina thickness segmentation in optical coherence tomography images using convolutional neural networks," *Biomedical Optics Express*, vol. 8, no. 7, Jul. 2017, doi: 10.1364/BOE.8.003292.
- [8] Q. Zheng, H. Delingette, N. Duchateau, and N. Ayache, "3-D consistent and robust segmentation of cardiac images by deep learning with spatial propagation," *IEEE Transactions on Medical Imaging*, vol. 37, no. 9, pp. 2137–2148, Sep. 2018, doi: 10.1109/TMI.2018.2820742.
- [9] X. Feng and C. Meyer, "Patch-based 3d u-net for brain tumor segmentation," in *International MICCAI Brainlesion Workshop*, 2017.
- [10] P.-L. Pröve *et al.*, "Automated segmentation of the knee for age assessment in 3D MR images using convolutional neural networks," *International Journal of Legal Medicine*, vol. 133, no. 4, pp. 1191–1205, Jul. 2019, doi: 10.1007/s00414-018-1953-y.
- [11] H. Zhao and N. Sun, "Improved U-net model for nerve segmentation," in *Image and Graphics: 9th International Conference, ICIG 2017, Shanghai, China, September 13-15, 2017, Revised Selected Papers, Part II 9*, 2017, pp. 496–504.
- [12] A. S. Musallam, A. S. Sherif, and M. K. Hussein, "Efficient framework for detecting COVID-19 and pneumonia from chest X-ray using deep convolutional network," *Egyptian Informatics Journal*, vol. 23, no. 2, pp. 247–257, Jul. 2022, doi: 10.1016/j.eij.2022.01.002.
- [13] S. Sharma, "Activation functions in neural networks," *Towards Data Science*. 2017, Accessed: Oct. 10, 2022. [Online]. Available: <https://towardsdatascience.com/activation-functions-neural-networks-1cbd9f8d91d6>.
- [14] S. Hayou, A. Doucet, and J. Rousseau, "On the impact of the activation function on deep neural networks training," *arXiv preprint arXiv: 1902.06853*, Feb. 2019.

- [15] C. E. Nwankpa, W. L. Ijomah, A. Gachagan, and S. Marshall, "Activation functions: comparison of trends in practice and research for deep learning," in *2nd International Conference on Computational Sciences and Technology, Jamshoro, Pakistan*, 2018, pp. 124–133.
- [16] Y. Wang, Y. Li, Y. Song, and X. Rong, "The influence of the activation function in a convolution neural network model of facial expression recognition," *Applied Sciences*, vol. 10, no. 5, pp. 1897–1913, Mar. 2020, doi: 10.3390/app10051897.
- [17] H.-X. Wang, J.-Q. Zhou, C.-H. Gu, and H. Lin, "Design of activation function in CNN for image classification," *Journal of Zhejiang University (Engineering Science)*, vol. 53, no. 7, pp. 1363–1373, 2019.
- [18] A. M. Taqi, A. Awad, F. Al-Azzo, and M. Milanova, "The impact of multi-optimizers and data augmentation on tensorflow convolutional neural network performance," in *2018 IEEE Conference on Multimedia Information Processing and Retrieval (MIPR)*, Apr. 2018, pp. 140–145, doi: 10.1109/MIPR.2018.00032.
- [19] S. Bera and V. K. Shrivastava, "Analysis of various optimizers on deep convolutional neural network model in the application of hyperspectral remote sensing image classification," *International Journal of Remote Sensing*, vol. 41, no. 7, pp. 2664–2683, Apr. 2020, doi: 10.1080/01431161.2019.1694725.
- [20] S. Vani and T. V. M. Rao, "An experimental approach towards the performance assessment of various optimizers on convolutional neural network," in *2019 3rd International Conference on Trends in Electronics and Informatics (ICOEI)*, Apr. 2019, pp. 331–336, doi: 10.1109/ICOEI.2019.8862686.
- [21] M. H. Saleem, J. Potgieter, and K. M. Arif, "Plant disease classification: a comparative evaluation of convolutional neural networks and deep learning optimizers," *Plants*, vol. 9, no. 10, Oct. 2020, doi: 10.3390/plants9101319.
- [22] S. Postalcioglu, "Performance analysis of different optimizers for deep learning-based image recognition," *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 34, no. 2, Feb. 2020, doi: 10.1142/S0218001420510039.
- [23] H. Wu and X. Gu, "Max-pooling dropout for regularization of convolutional neural networks," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, Dec. 2015, vol. 9489, pp. 46–54, doi: 10.48550/arxiv.1512.01400.
- [24] C. Bailer, T. Habtegebrail, K. Varanasi, and D. Stricker, "Fast feature extraction with CNNs with pooling layers," *arXiv preprint arXiv: 1805.03096*, May 2018.
- [25] J. Chang *et al.*, "A mix-pooling CNN architecture with FCRF for brain tumor segmentation," *Journal of Visual Communication and Image Representation*, vol. 58, pp. 316–322, Jan. 2019, doi: 10.1016/j.jvcir.2018.11.047.
- [26] V. Christlein, L. Spranger, M. Seuret, A. Nicolaou, P. Kral, and A. Maier, "Deep generalized max pooling," in *2019 International Conference on Document Analysis and Recognition (ICDAR)*, Sep. 2019, pp. 1090–1096, doi: 10.1109/ICDAR.2019.00177.
- [27] D. Scherer, A. Muller, and S. Behnke, "Evaluation of pooling operations in convolutional architectures for object recognition," in *Artificial Neural Networks-ICANN 2010: 20th International Conference, Thessaloniki, Greece, September 15-18, 2010, Proceedings, Part III 20*, 2010, pp. 92–101.
- [28] T. Szadala, "Review and comparison of commonly used activation functions for deep neural networks," *Bio-inspired neurocomputing*, pp. 203–224, 2021.
- [29] S. R. Dubey, S. K. Singh, and B. B. Chaudhuri, "Activation functions in deep learning: A comprehensive survey and benchmark," *Neurocomputing*, vol. 503, pp. 92–108, Sep. 2022, doi: 10.1016/j.neucom.2022.06.111.
- [30] F. Isensee, P. F. Jaeger, S. A. A. Kohl, J. Petersen, and K. H. Maier-Hein, "nnU-Net: a self-configuring method for deep learning-based biomedical image segmentation," *Nature Methods*, vol. 18, no. 2, pp. 203–211, Feb. 2021, doi: 10.1038/s41592-020-01008-z.
- [31] X. Li and J. Zhao, "A novel multi-modal medical image fusion algorithm," *Journal of Ambient Intelligence and Humanized Computing*, vol. 12, no. 2, pp. 1995–2002, Feb. 2021, doi: 10.1007/s12652-020-02293-4.
- [32] E. D. Gelasca, J. Byun, B. Obara, and B. S. Manjunath, "Evaluation and benchmark for biological image segmentation," in *2008 15th IEEE International Conference on Image Processing*, 2008, pp. 1816–1819, doi: 10.1109/ICIP.2008.4712130.
- [33] S. Edwardsson, "COVID-19 xray dataset," *GitHub*, 2020. <https://github.com/v7labs/covid-19-xray-dataset/> (accessed Feb. 16, 2022).
- [34] D. Yang, C. Martinez, L. Visuña, H. Khandhar, C. Bhatt, and J. Carretero, "Detection and analysis of COVID-19 in medical images using deep learning techniques," *Scientific Reports*, vol. 11, no. 1, Oct. 2021, doi: 10.1038/s41598-021-99015-3.
- [35] T. D. Pham, "Classification of COVID-19 chest X-rays with deep learning: new models or fine tuning?," *Health Information Science and Systems*, vol. 9, no. 1, Dec. 2021, doi: 10.1007/s13755-020-00135-3.
- [36] L. Alzubaidi *et al.*, "MedNet: pre-trained convolutional neural network model for the medical imaging tasks," *arXiv preprint arXiv: 2110.06512*, Oct. 2021.
- [37] T. Chauhan, H. Palivela, and S. Tiwari, "Optimization and fine-tuning of DenseNet model for classification of COVID-19 cases in medical imaging," *International Journal of Information Management Data Insights*, vol. 1, no. 2, Nov. 2021, doi: 10.1016/j.ijime.2021.100020.
- [38] J. Al-Saraireh and M. R. AlJa'afreh, "Keystroke and swipe biometrics fusion to enhance smartphones authentication," *Computers and Security*, vol. 125, Feb. 2023, doi: 10.1016/j.cose.2022.103022.

BIOGRAPHIES OF AUTHORS



Remah Younisse    a Ph.D. candidate at Princess Sumaya University for Technology (PSUT). Received the B.Sc. degree in Computer engineering from Jordan University for Technology in 2010 and M.S. in electrical engineering from PSUT in 2019. Currently, her research interests are medical image processing, AI models, and encryption. Machine learning embedded in network security. Also, interested in XAI models analysis. She can be contacted at email: r.baniyounisse@psut.edu.jo.



Rawan Ghnemat    received her Ph.D. in Computer Science, with distinction, from Le Havre University, France in 2009. Her study was funded by a French government scholarship. In 2009, she worked as assistant professor in the department of Computer Science at German-Jordanian University. She currently works as associate professor in the department of Computer Science at Princess Sumaya University for Technology. She has several scientific publications in artificial intelligence and data mining. She can be contacted at email: r.ghnemat@psut.edu.jo.



Jaafar Al Saraireh    received a B.Sc. degree in computer science from Mutah University, Karak, Jordan, in 1994, an M.Sc. degree in computer science from the University of Jordan, Amman, Jordan, in 2002, and a Ph.D. degree in computer science from Anglia Ruskin University, U.K., in 2007. He has been a professor of computer science/cyber security with Princess Sumaya University for Technology, since 2014. He is currently the Director of the consultancy and training center. His research interests include mobile, wireless network security, QoS, penetration test, hacking tacking and database. He is currently working as a full Professor at Princess Sumaya University for Technology, Jordan. He can be contacted at email: j.saraireh@psut.edu.jo.