

A Novel Architecture of Convolutional Neural Network to Diagnose COVID-19 Disease

Wasfi T. Kahwachi

Research Center Director- Tishk International University – Erbil, Iraq
wasfi.kahwachi@tiu.edu.iq

Khalida Ali Saed

Department of Project Management , University of Sulaimani, Sulaimaniya, Iraq
khalida.saed@univsul.edu.iq

Article Info

Page Number: 4831 - 4856

Publication Issue:

Vol 71 No. 4 (2022)

Abstract

COVID-19 is a rapidly spreading viral infection with its rapid spread and increasing numbers of patients that lasted until the time of writing this research caused which has led to a great loss all over the world, economically and socially, and it has become a necessity to rapidly diagnose the condition and contain it from spreading. We are aware that at the beginning of the pandemic, researchers in the field of artificial intelligence proposed a large number of automatic diagnosing models in an effort to aid radiologists and improve diagnosing accuracy based on X-ray images and Computed Tomography (CT) images, which have since been widely adopted to confirm positive COVID-19 RT-PCR tests, this was done due to the time-consuming nature of the Reverse Transcription Polymerase Chain Reaction (RT-PCR) tests and the false-negative rate. In this work, the chest CT scan images are classified and detected to COVID-19 and non-COVID-19 classes by using six automated DCNN architectures (VGG19, Inception-V3, Resnet-50, Inception-ResNet-V2, DenseNet121) and comparison between them by using some different activation functions. Our results demonstrate that Inception-V3 and DenseNet-121 produce superior outcomes with other activation functions instead of ReLU. For this, we suggest using these methods to aid the physician in making resolution in clinical practice, especially in impoverished areas with limited availability of radiologists with sufficient training in COVID-19 imaging. Finally, we conclude that the model and the dataset's behavior relative to the model determine which activation function is optimal. Additionally, we intend to compare simple activation functions with complex, adaptive activation functions using the newest application-specific architectures in our next work.

Index Terms: Deep Convolutional Neural Network (DCNN), Coronavirus disease (COVID-19), Chest Computed Tomography Scans (CT), Activation Functions (AFs).

Article History

Article Received: 25 March 2022

Revised: 30 April 2022

Accepted: 15 June 2022

Publication: 19 August 2022

1. Introduction:

Since the coronavirus disease caused by SARS-CoV-2 began in late 2019, it has rapidly spread over the entire world. The World Health Organization (WHO) declared an emergency case and an epidemic on January 30, 2020. [1] The disease has not yet been under control since then. [4] COVID-19 is a respiratory disease, which is caused by a severe acute respiratory syndrome that has common symptoms of sore throat, headache, fever, cough, short breathing, tiredness, aches, the vanishing of taste, loss of smell, and nasal blockage with diarrhea can also be observed in patients, [3] and the correct action to stop the spread of illness in healthy people is to separate an infected individual, but the problem is making an immediate diagnosis to distinguish positive patients from negative. The real-time polymerase chain reaction (RT-PCR) is the standard for detecting Covid-19, but its problem is that it is expensive and takes time to confirm the disease. [4] [5]. Global research efforts have concentrated on leveraging Artificial Intelligence (AI) technology on different medical data of COVID-19. In order to study the etiology of COVID-19 and quickly create ways to stop the development of this new disease, health officials collaborate with researchers from many fields. In fact, chest X-rays and Computed Tomography (CT) scan images are the most used image in medical image processing. [4] Computing tomography is a quick highly accurate and non-invasive imaging modality. [1] [10] Deep learning has recently produced remarkable results in the interpretation of medical images, which aids doctors in making wise conclusions regarding patient diagnosis. [1] Rapid diagnosis of COVID-19 with high reliability is essential in the early stages. [8] Whereas deep learning experts and medical specialists have attempted to quickly and cheaply differentiate COVID-19 cases from non-COVID-19 cases [9]. In order to address this issue more effectively, Artificial Intelligence (AI) researchers have built a variety of deep learning models and presented a number of studies that enable the diagnosis of anomalies in chest CT-scan images [4] [8] by using the Deep Convolutional Neural Networks (DCNN). [5]

On the other hand, the activation functions, are critical elements of any deep learning architecture. Although the oldest deep learning algorithms used a small number of layers throughout their architecture, however, the activation functions were used to perform diverse computations between the hidden layers and the output layer. [13] [14] LeNet5 from 1989, one of the earliest deep learning algorithms, has just five layers; since then, network layers have seen improvements in depth. [13] Additionally, as networks get more complex, it becomes imperative to comprehend the consecutive events occurring within the layers. [9] [7] The difficulty of training the neural network's algorithms has been a major challenge for researchers working on deep neural network architectures up until now, however, there are a number of interesting strategies for enhancing deep learning algorithm training that has been reported in the literature. [9]

In this work, we used automated methods to classify the chest CT-scan images for coronavirus patients, due to that, the experiments were conducted by using the 6 DCNN models (VGG-19, Inception-V3, Resnet-50, Exception, Inception ResNet-V2, DenseNet-121) for comparison between the different activation functions, these six techniques were chosen because of their great accuracy in published studies. [5] [27] [36] Moreover, we evaluated the performances of these

experimentation models by using various performance evaluation metrics to answer the following research questions:

- Q1.** Are there differences between the activation functions practice in DCNN models?
- Q2.** Is there any DCNN approach that distinctly outperforms other DCNN approaches?
- Q3.** Can DCNN be used to screen for coronavirus disease and make an early diagnosis using CT-scan images?
- Q4.** How accurate are the deep learning models in making diagnoses based on CT scan images?

The structure of this study is as follows: The study starts with the literature review in section 2, and section 3 is the methodology study contains the neural networks definition and the types of neural networks, activation function definition, and the types of activation functions, and finally, the evaluation criteria and performance are all covered in Section 4. We described the CNN model application in section 5, and the results are discussed in section 6 of the study. And the conclusions and future directions in section 7.

2. Related works

Several studies have been published recently to highlight the different activation functions that are used in deep learning and the need for activation functions in deep learning models. A review of some published studies on the use of activation functions is presented in this section.

(Naveen, P, 2022)[41] It has been proposed Phish, a novel activation function that is a composite function. Using images from the MNIST and CIFAR-10 datasets, the network was trained to reduce sparse categorical cross-entropy, where the Phish outperforms other activation functions for classification.

(Wang, X., et.al., 2022)[40] Proposed a new deep learning activation function (Smish) to improve the classification capacity of deep learning models. Whereas the characteristics of this function allow deep learning networks to perform better by making use of negative representation but compared to other activation functions, it is more complex. When the learning rates were increased, the effectiveness of Smish, outperformed ReLU, Swish, Mish, and Logh in Efficient Netflix.

(Dubey, S. R., et.al, 2021)[18] They presented an overview and the survey for activation functions in neural networks for deep learning. and covered the different classes of activation functions with several characteristics. Additionally, evaluations of the effectiveness of 18 cutting-edge activation functions were made using various networks and various types of data. Finally, they provided information on activation functions to aid researchers in conducting additional research and making different study options.

(Zhang, M., 2021)[15] Examine the relationship between the choice of the activation function and the accuracy of the image categorization in datasets of balanced or imbalanced COVID-19 chest X-rays. And give a thorough examination of ten activation functions. The experimental results showed

that the swish and soft plus functions improve the classification capability of cutting-edge networks and advanced specific recommendations for selecting suitable activation functions going forward.

(Ratnawati, D. E., et al., 2020) [2] Eleven activation functions were used, and the experiment's accuracy demonstrates that ELU and TanHRe, which can be employed in place of ReLU, where perform best in terms of average and maximum accuracy.

(Zhu, J., & Chen, Z., 2020) [23] List the benefits and drawbacks of four conventional activation functions before introducing Swish, FTS, and ReLU-Softplus, three innovative activation functions. They used a hybrid novel activation function on the CNN convolution layers that were built using VGGNet19, and they compared image recognition using the CIFAR-10 data set. They also discovered that the average time is shorter and the convergence speed is faster when using mixed activation functions rather than using a new activation function alone.

(Qiumei, Z., et al. 2019) [17] To speed exponential linear calculations and shorten the duration of network operation, they suggest a new activation function Fast Exponentially Linear Unit (FELU), also they test five conventional activation functions. The results of the trials show that the proposed activation function FELU not only accelerates exponential calculation, reducing the convolutional neural network's running time, and also significantly increases the noise robustness of the network to improve the accuracy of classification.

(Misra D., 2019) [22] Under most experimental circumstances, Mish outperforms Swish, ReLU, and Leaky ReLU in terms of empirical data. Future work in this topic includes significantly lowering the computational overhead and assessing how well the Mish activation function performs in other state-of-the-art models on various computer vision-related tasks.

(Nwankpa, C., et al., 2018) [37] They give an overview of the existing activation functions (AFs) used in deep learning applications and the most recent trends in their use to see if there would be improved performance results in outperforming architectures with cutting-edge functions.

3. Methodology

The proposed methodology for COVID-19 image screening based on CT scans is presented in this section, along with testing and comparison of the various activation functions using the most well-known deep convolutional neural network algorithms in the published works in the deep learning field.

3.1 Neural networks and the types of Neural networks

3.1.1 Neural networks

Artificial neural networks are made up of networks of synthetic neurons that resemble various components of human brains. Practically, an ANN is simply a parallel computational system made up of numerous simple processing components connected in a particular way to carry out a specific task, such as recognizing relationships in a set of data by adapting to or learning from a set of training patterns through a process that simulates how the human brain works. [39][44]

3.1.2 Type of Neural networks

Based on how information spreads throughout the network, neural networks can be divided into two categories: **Networks using feed-forward**; There are no loops or cycles in the graph, and the information flow in a feed-forward network only occurs in one direction, which includes (CNN) and MLP. In this study, image processing was carried out using CNN's multilayer neural networks (computer vision). **Feed-back networks**; These networks are connected in a way that creates directed loops. This architecture can retain data and sequence relationships in their internal memory, enabling them to work on and construct sequences of any size. Recurrent neural networks (RNN) and Long-Short term memory (LSTM) are two examples of such designs. [11][43][44][48]

4. Convolutional Neural Network Architectures

4.1 Convolutional Neural Network (CNN) definition:

Convolutional neural networks (CNNs) are a subclass of Artificial Neural Networks that have become the standard approach for computer vision tasks. [15] In the last few years, a large number of CNN architectures have been built and developed that have made tremendous progress such that (AlexNet, VGG, ResNet, Inception, and Xception, DenseNet, MobileNet, NASNet, MSDNet, etc) with different versions for various images classification challenges including medical imaging. [5][29] In this subsection, we introduce successful CNN designs which are constructed using the basic building blocks that are relatively complex and built on top of conventional designs in computer vision.

4.2 Activation Function:

The main reason activation functions are used in neural networks is to introduce non-linearity to the network. Without this non-linearity, the network cannot perform tasks well because the output of a neural network is rarely linear. [13] Due to the fact that activation functions can recognize patterns in a dataset, it serves as the primary building block for the training and optimization of a neural network. [9] The two most popular activation functions (sigmoid and hyperbolic tangent) were initially employed. [14][22]

4.2.1 Definition of Activation Function

An activation function is a function $h: \mathbb{R} \rightarrow \mathbb{R}$, [5] which we use to determine the node's output when the neuron becomes active above a specific threshold and tries to place the output within a specific range. [12] The goal of using this function, which maps the results of the input signals and determines whether the neuron should be activated or not, determines the output of neural networks, where:

$$output = f\left(\sum x_i w_i\right) \quad \dots (1)$$

The weighted values are added, put through a limiting function, and then output to the following layer. [14] Let the output function reflect the total amount of activity sent to the following neuron.

$$y_i = f\left(\sum x_i w_i\right) \quad \dots (2)$$

Where $f(w_i x_i)$ is the activation function, which connects a neuron's weights w_i to its input x_i and establishes the neuron's activation state. Where the weights stand in for the actual amount of information that is transmitted between neurons.[9][18] We can basically divide the activation functions into two groups based on that:

4.2.2 First group: The hidden layer functions:

These activation functions are utilized with the hidden layers. The most popular hidden layer functions are listed below.[18][40]

4.2.2.1 The Rectified Linear Unit –ReLU- 2010 [21]

ReLU are a sort of hidden layer function, that is linear in the positive dimension but zero in the negative dimension. They are used most frequently in hidden layers, especially in convolutional neural networks.[41][47] It is a straightforward and computationally efficient activation function.[42] It is defined in eq. (3) and it is shown in figure (1)

$$f_{ReLU}(x) = \text{Max}(0, x) \quad \dots (3)$$

The ReLU function's popularity and effectiveness led to the development of variants versions, including the Leaky ReLU, parametric ReLU, ELU, and SELU.[40][48]

4.2.2.2 Leaky Rectified Linear Unit –L-ReLU-2011[18]

This activation function known as a (Leaky- ReLU) is based on a ReLU and has a small slope for negative values rather than a flat slope. This small slope is defined as an incredibly small linear component for negative values of x rather than setting the value of the ReLU function to zero. It is defined in eq. (4) and it is shown in figure (1)

$$f_{L-ReLU}(x) = \begin{cases} x, & \text{if } x \geq 0 \\ 0.01 * x, & \text{if } x < 0 \end{cases} \quad \dots (4)$$

The leak factor, which is a constant and usually set to a low value, is represented by the slope coefficient ($\alpha=0.01$).[17]

4.2.2.3 Parametric Rectified Linear Unit –P-ReLU-2011[24]

Rectified Linear Unit is an additional option, the P- ReLU function works better and is slightly different.[17] The generalized conventional rectified unit has a slope for negative values.[41][42] It is defined in eq. (5)

$$f_{P-ReLU}(x) = \begin{cases} x, & \text{if } x > 0 \\ \alpha x, & \text{if } x \leq 0 \end{cases} \quad \dots (5)$$

where the network training is used to learn the tunable leak parameter. [24] And this is the ReLU function when $\lambda = 0$, and this is equivalent to the Leaky ReLU when λ is fixed. [41]

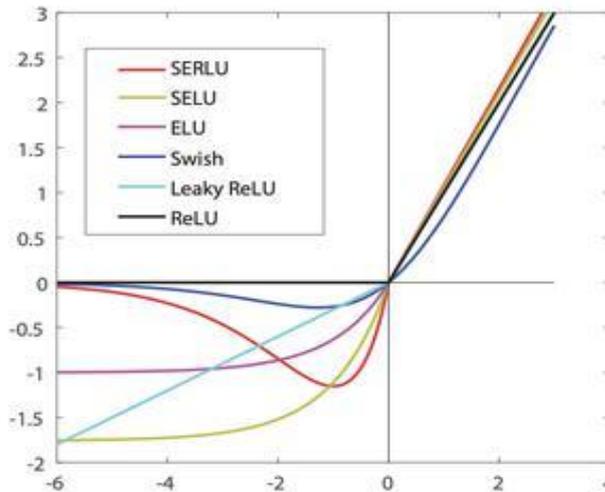


Figure (1) Curves of the ReLU, L- ReLU, ELU, SELU, Swish activation functions. [47]

4.2.2.4 Exponential Linear Unit –ELU-2015[26]

ELU is a neural network activation function. It is a modified form of ReLU, and in contrast to L-ReLU and P-ReLU, it provides some robustness to noise while addressing the gradient diminishing problem of ReLU. [18][41] It is defined in eq. (6) and it is shown in figure (1).

$$f_{ELU}(x) = \begin{cases} x & \text{if } x > 0 \\ \alpha (e^x - 1) & \text{if } x \leq 0 \end{cases} \quad \dots (6)$$

ELU defines the negative values using a log curve, which is differentiable, saturates for large negative inputs, and lessens bias shift. [17][18] Deep neural network learning is accelerated by ELU, which also improves classification accuracy. [26]

4.2.2.5 Scaled Exponential Linear Unit –SELU-2017[47]

The ELU is extended to Scaled ELU (SELU), when positive inputs are used, a scaling hyperparameter is used to increase the slope of the ELU. To automatically converge towards zero mean and unit variance, the SELU induces self-normalization. [18] It is defined in eq. (7) and it is shown in figure (1).

$$f_{SELU}(x) = \lambda * \begin{cases} x & \text{if } x \geq 0 \\ \alpha (e^x - 1) & \text{if } x < 0 \end{cases} \quad \dots (7)$$

The output range is $[-\lambda, \infty)$ where α is a hyperparameter. And α and λ are two fixed parameters, the values of $\alpha \approx 1.6732$, and $\lambda \approx 1.0507$. [16][47] See that when scale = 1, SELU is simply ELU.

4.2.2.6 Gaussian Error Linear Unit –GELU-2016[47]

The activation function of the ReLU is approximated by GELU. The GELU nonlinearity weights inputs by their percentile. As a result, the GELU can be considered a smoother ReLU.[34] This non-convex, non-monotonic function is non-linear and curved everywhere in the positive domain. [18] The GELU activation function is represented by:

$$f_{GELU}(x) = x * P(x \leq x) = x * \frac{1}{2} [1 + \operatorname{erf}(\frac{x}{\sqrt{2}})]$$

$$\approx 0.5 * x \left(1 + \tanh \left[\sqrt{\frac{2}{\pi}} (x + 0.044715x^3) \right] \right) \quad \dots (8)$$

In the range $(-0.17, \infty)$. The major goal of such a function is to avoid the significant jump discontinuity that ReLU exhibits at $(0, 0)$ in the origin of the Cartesian coordinate system.[18]

4.2.2.7 Swish Activation function -2017[19]

The swish function is non-linear, and its derivatives never experience discontinuities, increasing continuously and passing through the origin $(0, 0)$. The output range of it is $(-\infty, \infty)$. [18][41] The Swish activation function is given by eq. (9), and it is shown in figure (1).

$$f_{Swish}(x) = x * \operatorname{Sigmoid}(\beta * x) \quad \dots (9)$$

Where β is a learnable parameter, here $\beta = 1$. Additionally, the value of the function may reduce even as the input values increase. When employed in deep learning training, the smoothness property makes it provide better optimization and generalization results.[17][18]

4.2.2.8 Hyperbolic Tangent function -Tanh-2000[19][41]

The analog hyperbolic tangent function is symmetrical in relation to the origin. That limits its output to a value between -1 and 1.[17] It is continuous and differentiable, according to [42]. Additionally, it is monotonic while its derivative is not, allowing for the separation of two classes, which are utilized in feed-forward nets along with the logistic sigmoid function. Multi-layer neural networks performed better as a result.[17] The tanh activation function is given by eq. (10), and it is shown in figure (4).

$$f_{tanh}(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \quad \dots (10)$$

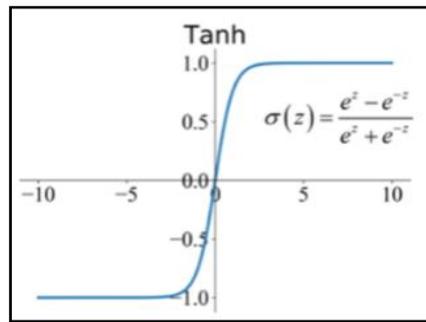


Figure (2) The Tanh activation function.[47]

4.2.2.9 Softplus activation function-2000[17][47]

ReLU and softplus are very similar, where softplus is seductively smooth and differentiable. Additionally, it is occasionally substituted for ReLU in neural networks.[18][19] The softplus activation function is given by eq. (11), and it is shown in figure (5).

$$f_{softplus}(x) = \ln(1 + \exp(x)) \quad \dots (11)$$

It is within the range $[0, \infty)$, to maintain enough gradients the softplus activation function has been proposed [17]. The model using the softplus function performs better in image classification.[15]

4.2.2.10 Softsgin activation function-2000[47]

Softsign is an alternative activation function to the hyperbolic tangent function, albeit it is not as widely used as tanh in practice applications. Additionally, it is zero-centered, which aids the propagation of the subsequent neuron.[30][31] Softsgin activation function is given by eq. (12), and it is shown in figure (6).

$$f_{softsgin}(x) = \frac{x}{(1 + |x|)} \quad \dots (12)$$

4.2.2.11 Rectified linear unit- 6 (RELU-6)-2017[47]

Rectified linear unit- 6, is a different modification activation function of the rectified linear unit that is frequently used in deep convolutional neural networks. It has a higher level of robustness when applied to low-precision computation. This helps the model to acquire sparse features.[47] [38]RELU-6 activation function is given by eq. (10), and it is shown in figure (7).

$$f_{ReLU-6}(x) = \min(\max(0, x), 6)$$

$$= \begin{cases} 0, & \text{if } x = 0 \text{ or } x = 6 \\ 1, & \text{if } 0 < x < 6 \end{cases} \dots (13)$$

4.2.2.12 Sigmoid Linear Unit (SiLU)-2017[47]

The Sigmoid Linear Unit activation function -SiLU, is calculated by multiplying the input by the sigmoid function. Elfving, S., et.al. (2017) introduced SiLU for neural network function approximation in reinforcement learning, and they showed that SiLUs in the convolutional layers beat ReLUs significantly. [6][47] SiLU activation function is given by eq. (14), and it is shown in figure (8).

$$f_{SiLU}(x) = \frac{x}{1 + e^{-x}} \dots (14)$$

4.2.2.13 Mish Activation Function-2019[47]

Mish is a neural network activation function defined by eq (15). Is similar to the swish activation function in that it has a range $[\approx -0.31, \infty)$, is bounded below, and outperforms ReLU in ResNet-50 in accuracy, while maintaining all other network parameters and hyperparameters constant. [40] [47] Mish activation function is given by eq. (15).

$$f_{mish}(x) = x \cdot \tanh(\text{softplus}(x)) \\ = x \tanh(\ln(1 + e^x)) \dots (15)$$

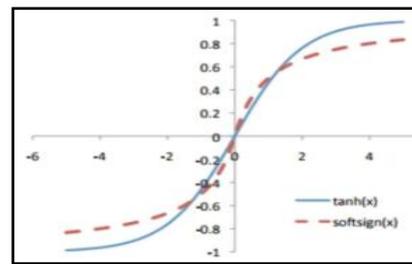
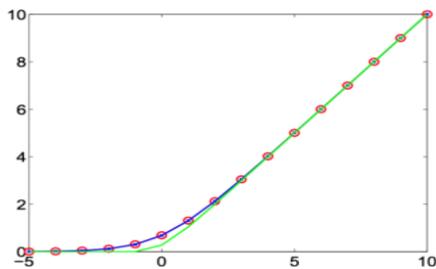


Figure (3) The Softplus activation function.[47] **Figure (4)** The Softsign activation function.[47]

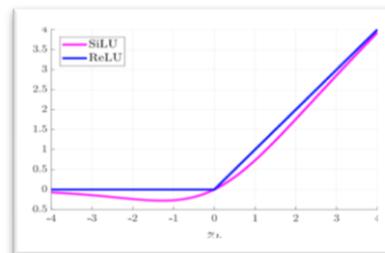
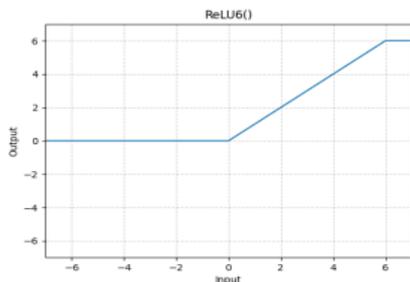


Figure (5) The ReLU 6 activation function.[47] **Figure (6)** The SiLU activation function.[47]

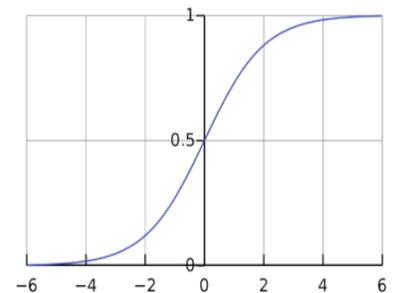
4.2.3 Second group: The output function:

The linear output activation function or (identity) function is one of those that is employed in the output layer for the categorization of the output because it does not alter the weighted sum of the input in any manner and merely returns the result. If the input exceeds the threshold, the neuron is activated in the first class; if the input falls below the threshold, the neuron is activated in a different class.[43]

4.2.3.1 Sigmoid Activation Function -2000

It is a specific kind of neural network activation function, that have slow convergence, gradient saturation, and strong damp gradients during the backpropagation from deeper hidden layers to inputs.[17] Additionally, it can function successfully when all of the training data's values are positive.[42] The sigmoid gradients at the extremities of the curve are almost completely saturated at 0.[17][32] The sigmoid activation function is given by eq. (16), and it is shown in figure (9).

$$f_{sigmoid}(x) = \frac{1}{1 + e^{-x}} \quad \dots (16)$$



Figure(7) The Sigmoid activation function.[47]

4.3. Dataset:

In the application, we used the publicly available SARS-CoV-2 CT-scan dataset, which has more images (2482 images) and was collected from hospitals in Sao Paulo, Brazil. containing 1252 CT-scan images from 60 Patients from males (32) and females (28) that are positive for infection (COVID-19) and 1230 CT-scan images from 60 Patients (30) males and (30), females normal (non-COVID-19).[1] the dataset is available at: <https://www.kaggle.com/plameneduardo/sarscov2-ctscan-dataset>.

4.4 Performance metrics and evaluation criteria:

This subsection presents performance measures, including: sensitivity, specificity, recall, accuracy, and F_1 - score values.[27][33][35] Here are the definitions for some well-known measures. True Positive (T_P), False Positive (F_P), True Negative (T_N), and False Negative (F_N) values are needed for these assessments. A confusion matrix is used to measure T_P , T_N , F_P , and F_N as indicated in table (1).

4.4.1 Confusion matrix:

A method for summarizing a classification algorithm's performance is the confusion matrix. Classifying the obtained data and assigning it to a particular class is the final step after extracting the suitable feature.[5]

Table (1) Is shown confusion matrix for the two classes COVID-19 and non-COVID-19.

COVID-19	T_P	F_N
Non- COVID-19	F_P	T_N

4.4.2 Accuracy:

The proportion of rectified instances to the total number of cases is the accuracy. It's represented in eq. (17).

$$\text{Accuracy} = \frac{T_P + T_N}{T_P + T_N + F_P + F_N} \quad \dots (17)$$

4.4.3 False Negative Rate (FNR):

False Negative Rate (FNR) is the percentage of positive classes that the classifier misclassified. Since we wish to correctly classify the positive classes, a higher TPR and a lower FNR are preferable. It's represented in eq. (18).

$$\text{FNR} = \frac{F_N}{T_P + F_N} \quad \dots (18)$$

4.4.4 Sensitivity / True Positive Rate / Recall

The computed proportion of the positive class was accurately identified due to sensitivity. And it's represented in eq. (19).

$$\begin{aligned} \text{Sensitivity} &= \frac{T_P}{T_P + F_N} \\ &= 1 - \text{FNR} \end{aligned} \quad \dots (19)$$

4.4.5 Specificity / True Negative Rate:

The percentage of the negative class that was correctly classified is known as specificity. It's represented in eq. (20).

$$\text{Specificity} = \frac{T_N}{T_N + F_P} \quad \dots (20)$$

4.4.6 Precision:

It is referred to as a positive predictive value and is a measurement of the proportion of pertinent examples among the cases that were retrieved. It's represented in eq. (21).

$$\text{Precision} = \frac{T_p}{T_p + F_p} \quad \dots (21)$$

Where Sensitivity and Specificity are the most important of these criteria.

4.4.7. F_1 – score:

The accuracy of a test is gauged by the F-Score or -measure in statistical analysis of binary classification. The harmonic mean of recall and precision is the F- score. An F-score can have a maximum value of 1, which denotes flawless precision and recall, and a minimum value of 0, which denotes 0% recall. It's represented in eq. (22).

$$F_1 = 2 * \frac{\text{Precision} * \text{Recall}}{\text{Precision} + \text{Recal}} \dots (22)$$

5. Application

We used the 6 DCNN models, the VGG-19, Resnet-50, Inception-V3, Xception, Inception ResnetV-2, and DenseNet -121 approaches. These six techniques were chosen because of their great accuracy in studies published recently, to compare between some different activation functions. And by using the SARS-CoV-2 CT-scan dataset, which contains 1252 pictures of COVID-19 positive cases and 1230 images for non-COVID-19 cases, and we used this CT-scan images from COVID-19-infected patients and normal cases to train the models, where previously trained on a larger dataset which is frequently adequate to learn a certain hierarchy to extract characteristics from images.

In figure (1) an example architecture of the pre-trained model approach, demonstrates the architecture for all models in the DCNN application baseline for our experiment models, where the modified architecture follows the steps below:

1. Input layer: it is the first layer, the inputs are CT-images All of the images in the dataset were resized to 224x224 pixels, with the exception of the Xception, Inception V3 model's images, which were resized to 299x299 pixels.
2. In all models, we used the convolutional and pooling layers without any changeling.
3. The activation function: In all models, we first used the ReLU function with a convolution layers (between layers), and then we utilized the other various activation functions (ReLU, L-ReLU, P-ReLU, tanh, Softplus, ELU, SeLU, GELU Softsgin, Swish, ReLU6, SiLU, Mish) in another time.
4. The fully connected layer uses cross-entropy loss to produce an output of a single vector from data supplied in the form of a simple vector (log loss).
5. Finally, as for the output layer, we use the sigmoid function because we have two classes of output COVID-19 and normal (non-COVID-19) cases.

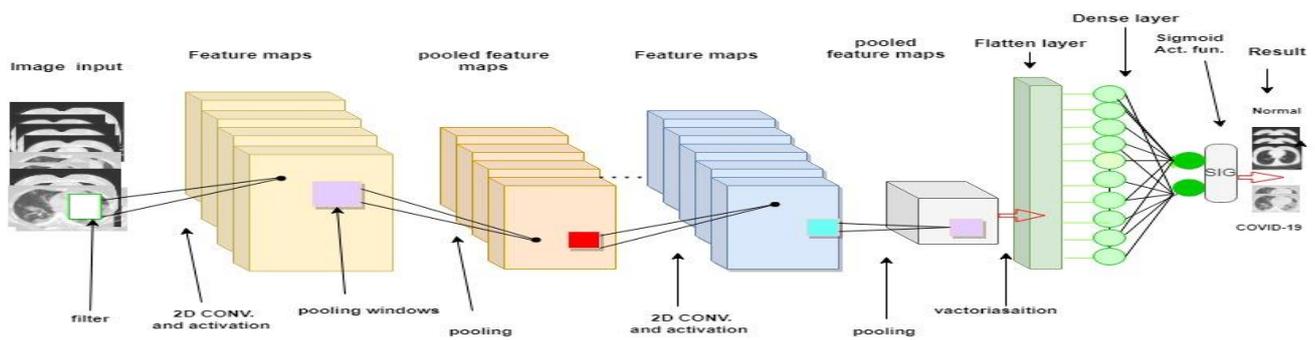


Figure (8) Principal design of our basic DCNN.[45]

5.1 Experimental results:

In this section, we present the result for a two-class classification obtained from the pre-trained models (VGG-19, Resnet-50, Inception - V3, Xception, Inception ResNet-V2, Densenet-121) by using CTscan images of COVID-19 and non-COVID-19 cases to the comparison between the different activation functions, by using the optimizer = Adam, learning rate (lr) = 0.001, with the 80% training and 20% testing of data. The results of the classification of the models are shown, along with the confusion matrices for the two classes. The methods have been validated using performance metrics for accuracy, sensitivity, and specificity. The primary goal of this classification is to identify confirmed COVID-19 cases, and it is done by contrasting the DCNN models using various activation functions. The loss and accuracy graphs, along with the confusion matrices results that were trained on a dataset, are the best tools for visualizing model training. The datasets that the model was testing are represented by rows, and the column represents the datasets on which the model was evaluated for two classes.

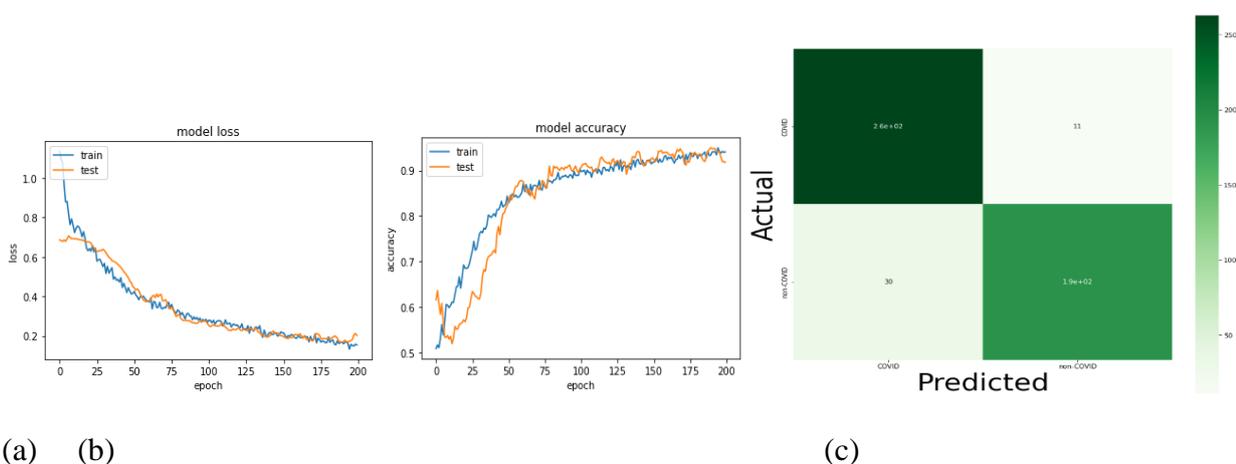


Figure (9) Explains the training and testing analysis for model InceptionV3 with ReLU, and sigmoid output function. (a), (b) The accuracy and loss graphs for testing and training. (c) Show the confusion matrix results of the model, from a total of 271 images were tested, out of which 260 at

the rate of 0.96 were correctly classed as COVID-19, and at the rate of 0.041 were found to be wrongly classed. And from 220 images, 190 were correctly categorized as non-COVID-19 or at the rate of 0.86, and at the rate of 0.14 were found to be wrongly classed. Table (2) presents the performance outcomes.

Table (2) Comparison between the different DCNN models by using the ReLU activation function.

Metric	Accuracy (%)	Sensitivity (Recall) (%)	Specificity (%)	Precision (%)	F1- Score
Model by ReLU					
VGG-19	0.6802	0.30942	0.98175	0.920	0.8123
Resnet-50	0.8753	0.7763	0.95238	0.92896	0.8458
Inception -V3	0.9175	0.8655	0.95985	0.9453	0.9026
Xception	0.6539	0.96413	0.40152	0.57895	0.7237
Inception ResNet -V2	0.8773	0.8072	0.93431	0.9091	0.8 551
Densenet-121	0.9115	0.8850	0.93525	0.9174	0.9009

From table (2), it can be shown that the Inception - V3 classifier with the ReLU activation function outperformed other methods in terms of all measures, which may be useful for experts. In this instance, the obtained accuracy is 0.9175, sensitivity 0.8655, specificity 0.95985, precision 0.9453, and F1- score 0.9026.

Table (3) Comparison between the different DCNN models by using the Leaky ReLU activation function.

Metric	Accuracy (%)	Sensitivity (%)	Specificity (%)	Precision (%)	F1- Score
Method by leaky ReLU					
VGG-19	0.55131	0.0	1.0	0	0
Resnet-50	0.8773	0.8036	0.9386	0.91371	0.8551
Inception - V3	0.89135	0.8386	0.9343	0.91346	0.8756
Xception	0.76861	0.9283	0.6387	0.6796	0.7851
Inception ResNet-V2	0.8813	0.7758	0.96715	0.9497	0.85213
Densenet-121	0.94165	0.9460	0.9386	0.9251	0.9354

From table (3) above, it can be seen the Densenet-121 classifier with Leaky ReLU activation function outperformed other methods in terms of all measures. In this instance, the obtained accuracy is 0.94165, sensitivity 0.9460, specificity 0.9386, precision 0.9251, and F1 -score 0.9354.

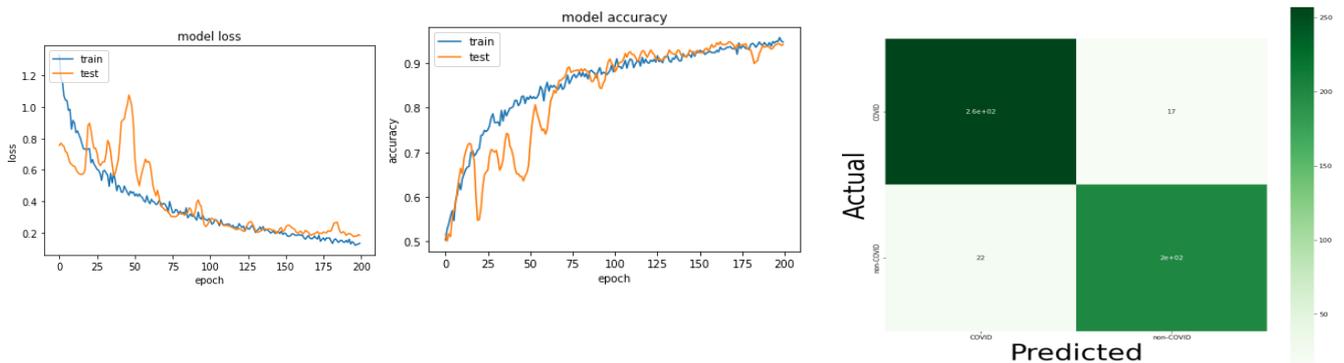


Figure (10) Explain the Inception - V3 classifier with the SeLU activation function with the confusion matrix analyses of the model, from a total of 277 images were tested, out of which 260 were correctly classed as COVID-19at the rate of 0.94. And from 222 images, 200 were correctly categorized as non-COVID-19at the rate of 0.90.Table (4) presents the performance outcomes.

Table (4) Comparison between the different DCNN models by using the SELU activation function.

Metric	Accuracy %	Sensitivity %	Specificity %	Precision%	F1- Score
Model by SELU					
VGG-19	0.82495	0.9910	0.6898	0.7213	0.8349
Resnet-50	0.88934	0.8444	0.926	0.9048	0.8736
Inception - V3	0.9215	0.9014	0.938	0.9217	0.9112
Xception	0.77264	0.9552	0.6241	0.67742	0.7925
Inception ResNet-V2	0.91549	0.9507	0.8867	0.8714	0.9091
Densenet-121	0.9034	0.8889	0.91575	0.8969	0.89285

From table (4) The Inception - V3 classifier with the SELU activation function provided better results in terms of all metrics than the other approaches. Where the achieved accuracies are 0.9215, sensitivity 0.9014, specificity 0.938, precision 0.9217, and F1 score 0.9112.

Table (5)Comparison between the different DCNN models by using the Swish activation function.

Metric	Accuracy %	Sensitivity	Specificity %	Precision %	F1- Score
Model by Swish					
VGG-19	0.7244	1.0	0.5	0.6111	0.7586
Resnet-50	0.9235	0.9333	0.9158	0.9013	0.9170
Inception - V3	0.8068	0.6682	0.9197	0.8721	0.7576
Xception	0.6740	0.9193	0.4745	0.5882	0.7169
Inception ResNet-V2	0.8853	0.9058	0.8686	0.8475	0.8753
Densenet-121	0.93306	0.94595	0.9225	0.9091	0.9272

function.

From table (5) it can be shown that the DenseNet-121 classifier with the swish activation function outperformed other methods in terms of all measures. In this instance, the obtained accuracy is 0.93306, sensitivity 0.94595, specificity 0.9225, precision 0.9091, and F1 score 0.9272.

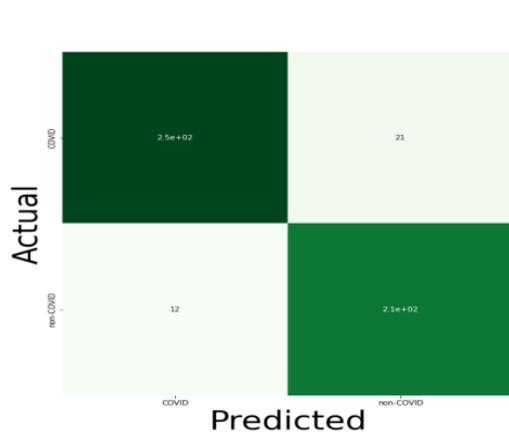


Figure (11)

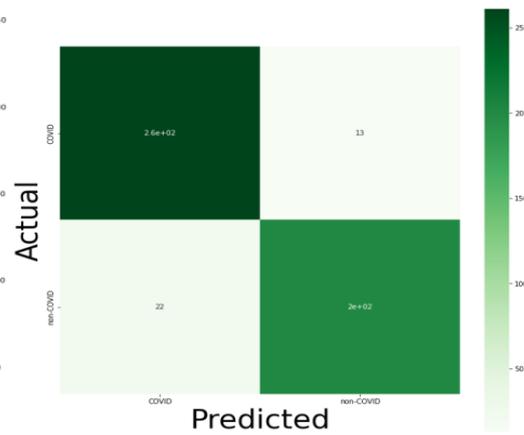


Figure (12)

Figure (11) Explains the DenseNet-121 with a swish AF with the confusion matrix results, from a total of 271 images were tested, out of which 250 at the rate of 0.92 were correctly classed as COVID-19, and at the rate of 0.077 were found to be wrongly classed. And from 222 images, 210 images were correctly categorized as non-COVID-19 or at the rate of 0.95, and at the rate of 0.054 were found to be wrongly classed. Table (5) presents the performance outcome.

Figure (12) Explains the InceptionV3 classifier with the ELU AF with the confusion matrix results, from a total of 273 images were tested, out of which 260 at the rate of 0.95 were correctly classed as COVID-19, and 13 images at the rate of 0.048 were found to be wrongly classed. And from 222 images, 200 images at the rate of 0.90 were correctly categorized as non-COVID-19, and 22 images at the rate of 0.099 were found to be wrongly classed. See table (6).

Table (6) Comparison between the different DCNN models by using the ELU activation function.

Metric	Accuracy%	Sensitivity %	Specificity %	Precision%	F1- Score
Method by ELU					
VGG-19	0.7767	0.54261	0.9672	0.93023	0.6877
Resnet-50	0.9175	0.91324	0.91912	0.9009	0.9070
Inception - V3	0.92958	0.9014	0.95238	0.939	0.9333
Xception	0.74648	0.94619	0.58394	0.6563	0.7749
Inception ResNet-V2	0.91348	0.8879	0.93431	0.91743	0.9029
Densenet-121	0.85714	0.7273	0.95941	0.9357	0.81841

From table (6) above, it can be shown that the Inception - V3 classifier with the ELU activation function outperformed other methods in terms of all measures. In this instance, the obtained accuracy is 0.92958, sensitivity 0.9014, specificity 0.95238, precision 0.939, and F1 -score 0.9333.

Metric	Accuracy %	Sensitivity %	Specificity %	Precision%	F1- Score
Model by PRelu					
VGG-19	0.7827	0.5471	0.9745	0.9445	0.6916
Resnet-50	0.9195	0.9546	0.8889	0.875	0.9130
Inception - V3	0.9396	0.9148	0.9343	0.913	0.9333
Xception	0.6781	0.95516	0.4523	0.5833	0.72414
Inception ResNet-V2	0.9135	0.8744	0.9453	0.9302	0.9029
Densenet-121	0.93159	0.89286	0.96296	0.9524	0.9217

Table (7) Comparison between the different DCNN models by using the P-ReLU activation function.

From table (7) above, we see the Inception - V3 and Densenet-121 approaches with the PReLU activation function provided approximate results in terms of all metrics than the other approaches. Where the achieved accuracies are 0.9396, 0.93159, sensitivity 0.9148, 0.89286, specificity 0.9343, 0.96296, precision 0.913, 0.9524 and F1 score are 0.9333, 0.9217 respectively.

Table (8) Comparison between the different DCNN models by using the Softplus activation function.

Metric	Accuracy %	Sensitivity %	Specificity %	Precision%	F1- Score
Model by Softplus					
VGG-19	0.8531	0.9866	0.7445	0.758621	0.8577
Resnet-50	0.92153	0.9649	0.8856	0.8765	0.9186
Inception - V3	0.9396	0.9552	0.9270	0.9130	0.9333
Xception	0.7103	0.96861	0.500	0.6111	0.7496
Inception ResNet-V2	0.91952	0.9238	0.9161	0.9013	0.9130
Densenet-121	0.8753	0.9910	0.7810	0.7857	0.8765

From table (8) above, it can be shown that the Inception - V3 classifier with the softplus activation function outperformed other methods in terms of all measures, in this instance, the obtained accuracy is 0.9396, sensitivity 0.9552, specificity 0.9270, precision 0.9130, and F1 score 0.9333.

Table (9) Comparison between the different DCNN models by using the GELU activation function.

Metric	Accuracy %	Sensitivity %	Specificity %	Precision %	F1- Score
Method by GELU					
VGG-19	0.8511	0.9821	0.7445	0.5641	0.856
Resnet-50	0.9175	0.9333	0.9058	0.895	0.9111
Inception - V3	0.9235	0.87444	0.9635	0.9524	0.9132
Xception	0.6539	0.95964	0.405	0.5676	0.7131
Inception ResNet-V2	0.9155	0.8969	0.9307	0.91324	0.9050
Densenet-121	0.8773	0.75892	0.9747	0.9605	0.8479

function.

From table (9) above, it can be shown that the Inception - V3 classifier with GELU outperformed other methods. In this instance, the obtained accuracy is 0.9235, sensitivity 0.87444, specificity 0.9635, precision 0.9524, and F1 -score 0.9132, also for other models.

Table (10) Comparison between the different DCNN models by using the Softsign activation function.

Metric	Accuracy %	Sensitivity %	Specificity %	Precision%	F1- Score
Model by Softsign					
VGG-19	0.71227	1.0	0.478103	0.61111	0.7586
Resnet-50	0.8974	0.9333	0.8696	0.8537	0.8917
Inception - V3	0.9417	0.9327	0.9489	0.9375	0.9354
Xception	0.73642	0.969	0.54745	0.6471	0.7760
Inception ResNet-V2	0.9075	0.8655	0.9416	0.9223	0.8920
Densenet-121	0.90745	0.9866	0.8425	0.8365	0.9053

From table (10) above, it can be shown that the Inception - V3 classifier with the softsign activation function outperformed other methods in terms of all measures. In this instance, the obtained accuracy is accuracies are 0.9417, sensitivity 0.9327, specificity 0.9489, precision 0.9375, and F1 score 0.9354.

Table (11) Comparison between the different DCNN models by using the ReLU6 activation function.

Metric	Accuracy %	Sensitivity %	Specificity %	Precision%	F1- Score
Method by ReLU6					
VGG-19	0.5694	0.04036	1	1	0.07895
Resnet-50	0.9256	0.9597	0.8978	0.8824	0.919
Inception - V3	0.8934	0.8263	0.9526	0.9325	0.8717
Xception	0.5815	1	0.2412	0.5116	0.6770
Inception ResNet-V2	0.9296	0.9193	0.9382	0.9217	0.9195
Densenet-121	0.9095	0.9731	0.8577	0.8494	0.9072

From table (11) we can see the Inception ResNet-V2 classifier with the ReLU6 activation function outperformed other methods in terms of all measures. In this instance, the obtained accuracy is 0.9296, sensitivity 0.9193, specificity 0.9382, precision 0.9217, and F1 score 0.9195.

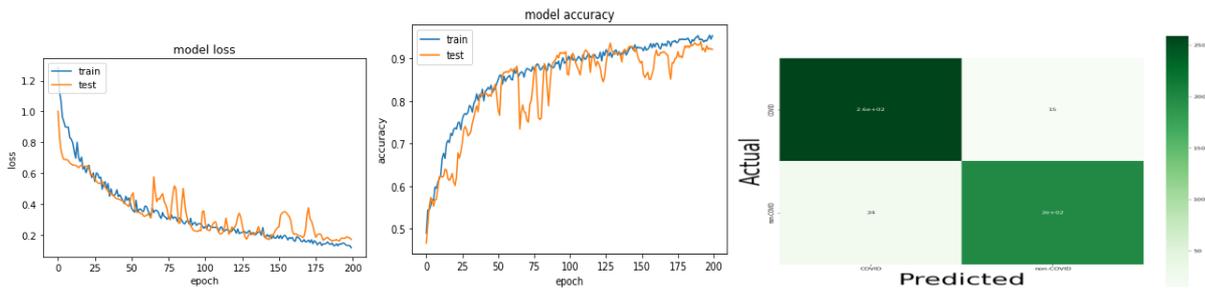


Figure (13) Explains the inceptionV3 classifier with the SiLU AF with the confusion matrix results, from a total of 275 images were tested, out of which 260 at the rate of 0.95 were correctly classed as COVID-19, and 15 images were found to be wrongly classed. And from 224 images, 200 images were correctly categorized as non-COVID-19, and 24 images were found to be wrongly classed, table (9) presents the performance outcome.

Table (12) Comparison between the different DCNN models by using the SiLU activation function.

Method by SiLU	Metric	Accuracy %	Sensitivity (%)	Specificity %	Precision%	F1- Score
VGG-19		0.8332	0.9821	0.71168	0.7358	0.8413
Resnet-50		0.91952	0.9238	0.9161	0.9013	0.9131
Inception - V3		0.92153	0.8924	0.9453	0.9302	0.9112
Xception		0.5513	0	1	0	0
Inception ResNet-V2		0.9074	0.8614	0.9453	0.927	0.892
Densenet-121		0.9115	0.8520	0.964	0.9453	0.8962

From table (12) above, it can be shown that the Inception - V3 classifier with the SiLU activation function outperformed other methods in terms of all measures. Where the achieved accuracies are 0.92153, sensitivity 0.8924, specificity 0.9453, precision 0.9302, and F1 score 0.9112. Also, the Resnet-50 and the Densenet-121 are that have good performance respectively.

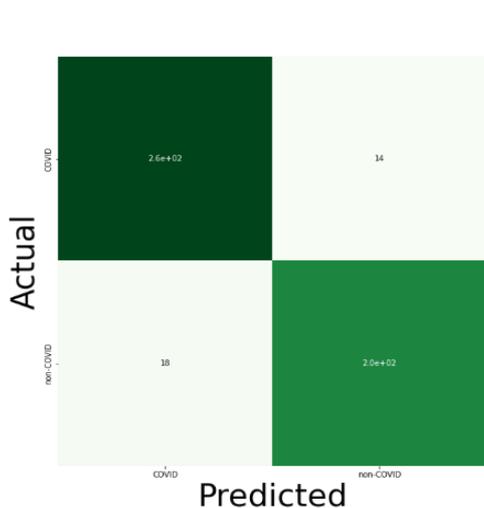


Figure (14)

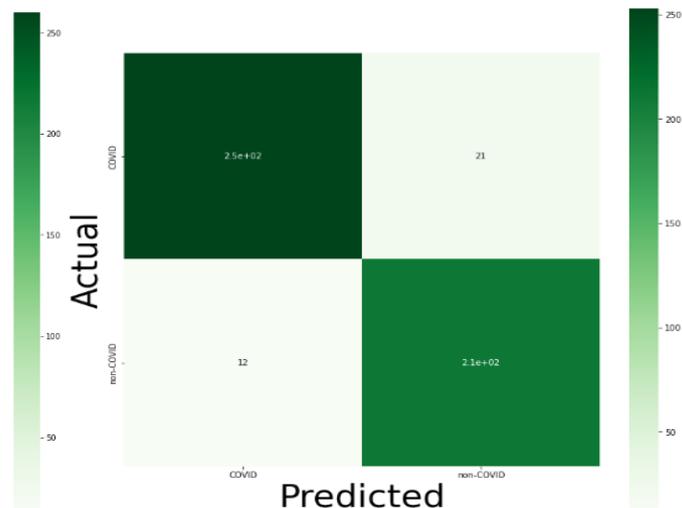


Figure (15)

Figure (14) Explains the Inception ResNetV2 classifier with the tanh AF with the confusion matrix analyses of the model, from a total of 274 images were tested, out of which 260 at the rate of 0.95 were correctly classed as COVID-19, and 14 images at the rate of 0.05 were found to be wrongly classed. And from 218 images, 200 images at the rate of 0.92 were correctly categorized as non - COVID-19, and 18 images at the rate of 0.08 were found to be wrongly classed.

Figure (15) Explains the analysis for modeling the Densenet-121 classifier with the Mish activation function with the confusion matrix analyses of the model, from a total of 271 images were tested, out of which 250 at the rate of 0.94 were correctly classed as COVID-19, and 21 images found to be wrongly classed. And from 33 images, 21 images were correctly categorized as non-COVID-19, and 12 images were found to be wrongly classed, table (14) presents the performance outcome.

Table (13) Comparison between the different DCNN models by using the Mish activation function.

Metric	Accuracy %	Sensitivity %	Specificity %	Precision%	F1- Score
Model by Mish					
VGG-19	0.6802	0.30942	0.98175	0.920	0.8123
Resnet-50	0.8753	0.7763	0.95238	0.92896	0.8458
Inception - V3	0.9175	0.8655	0.95985	0.9453	0.9026
Xception	0.6539	0.96413	0.40152	0.57895	0.7237
Inception ResNet-V2	0.8773	0.8072	0.93431	0.9091	0.8 551
Densenet-121	0.9115	0.8850	0.93525	0.9174	0.9009

From table (13) it can be shown that the Inception - V3 classifier with the Mish activation function outperformed other methods in terms of all measures. Where the achieved accuracies are 0.09175, sensitivity 0.8655, specificity 0.95985, precision 0.9453, and F1-score 0.9026.

6. Discussion:

By the (2-13) different tables above in the past section, we compared the various pre-trained models using thirteen commonly different activation functions (classic with modern), according to accuracy, specificity, sensitivity, precision, and F1-score. The results showed that Inception-V3 gave a good performance in classification with most of the activation functions, followed by Densnet-121, Inception ResNet V2, and ResNet-50. On the other hand, VGG19 and Xception perform the worst when compared to other DCNN architectures because they aid in achieving the lowest accuracy using the activation functions utilized in this study.

Next, we explain and discuss the training and testing analysis results for the six DCNN models with some activation functions and a comparison between them by final loss, and the final accuracy for models employed in this experiment. Table (14) presents the performance outcome.

In the current study, we started with a ReLU function, employed a few other functions in the development of the activation functions path, and compared the most popular deep learning architectures based on final accuracy and loss for these models. Additionally, we observe the following:

1. The Inception -V3 model gave good classification performance with most activation functions and approximated in other activation functions compared with other DCNN architectures, from

where minimum loss and maximum accuracy, where have (18.72% of loss and 94.17% of accuracy) with the softsign AF compared with ReLU AF, followed by Densnet-101 that have (16.81% of loss and 94.165% of accuracy) with the Leaky ReLU AF, followed by Incpetion_Resnet_V2 that have (23.25% of loss and 93.56% of accuracy) with the than AF, and ResNet-50 that have (20.61% of loss and 92.35% of accuracy) with the swish AF. In contrast, VGG19 and Xception perform poorly since, when compared to other DCNN designs, they help to achieve the highest loss and the lowest accuracy with all thirteen activation functions. Table (14) explains everything.

Table (14) A comparison of the thirteen activation functions used by the six different DCNN models, based on final accuracy and loss.

Name of Activation function	Inception ResNet-V2											
	Inception-V3		VGG-19		Xception		ResNet-50		DenseNet121		Inception ResNet-V2	
	Loss	Accuracy	Loss	Accuracy	Loss	Accuracy	Loss	Accuracy	Loss	Accuracy	Loss	Accuracy
ReLU-2010	0.2017	0.9175	0.618	0.6801	0.5813	0.6539	0.3001	0.8753	0.2137	0.9115	0.2464	0.8914
Leaky ReLU	0.2564	0.8914	0.41813	0.8572	0.4612	0.7686	0.2975	0.8823	0.1681	0.94165	0.2864	0.8813
Softplus	0.18759	0.93964	0.41688	0.8531	0.5348	0.7103	0.18254	0.921529	0.18759	0.93964	0.2272	0.9195
Tanh	0.28605	0.89537	0.2668	0.89336	0.38036	0.8209	0.25515	0.8954	0.26576	0.8873	0.2325	0.9356
Softsign	0.18272	0.94165	0.84153	0.71227	0.51811	0.7364	0.2393	0.89738	0.24004	0.9075	0.233	0.9075

			0.50	0.782	0.55	0.678			0.17	0.931		
PReLU	0.18 76	0.939 64	906	7	483	1	0.229	0.919	946	59	0.21 42	0.913 5
							0.746	0.243	0.909			0.913
Elu	0.20 364	0.929 58	0.49 883	0.776 66	0.47 754	5	6	5	0.30 418	0.857 2	0.22 1	5
				0.851		0.653	0.217	0.917	0.26	0.877		0.915
GELU	0.19 642	0.923 54	0.46 315	107	0.58 141	9	44	5	81	3	0.22 6	5
			0.53		0.45		0.217	0.917	0.21	0.903	0.24	0.915
SELU	0.20 62	0.921 5	01	0.825	33	0.772 6	44	5	437	4	46	5
ReLU 6	0.24 34	0.893 4	1.28 89	0.569 4	0.63 58	0.581 5	0.217 2	0.925 6	0.22 81	0.909 5	0.18 08	0.929 6
	0.32 844	0.863 18	0.92 955	0.688 1	0.58 131	0.674 1	0.206 1	0.923 54	0.18 284	0.933 6	0.27 384	0.885 3
Swish												
SILU	0.17 13	0.921 53	0.47 012	0.833 4	0.69 17	0.551 31	0.216 7	0.919 5	0.19 46	0.911 47	0.22 44	0.907 5
												0.863
Mish	0.48 01	0.770 6	0.78 31	0.750 5	0.51 81	0.714 3	0.246 9	0.895 4	0.18 284	0.933 6	0.37 013	2

- Also, we notice that all different activation functions have good performance compared with ReLU AF in all DCNN architectures used in this study.
- From all past result tables (12 tables), we compared the most popular deep learning architectures to identify COVID-19 and non-COVID-19 cases using CT-scan images, the Inception-V3, Densnet-121, ResNet-50, and partially Inception ResNet V2 classifiers with the common varied activation functions perform better in terms of all metrics for DCNN models evaluation, these results may be useful for specialists to identify and categorize coronavirus pneumonia. The identical proposal is made in [36] as well.
- There are differences between the activation functions practice in these DCNN models used in this study, but not found a better activation function for building a robust model with good performance to all CNN models.

7. Conclusion

The use of artificial intelligence (AI) technology on various medical data of COVID-19-positive patients has recently been the focus of research efforts across the globe. Health officials collaborated with researchers from several fields to study the COVID-19 disease and quickly came up with plans to stop its spread. In this study, to classify the chest CT-scan images for coronavirus patients, to answer the research questions in section(1), we used six automated DCNN architectures (VGG19, Inception-V3, Resnet-50, Inception-ResNet-V2, DenseNet121) and comparison between them by using some of the activation functions. And toward this end, were conducted experiments, and the performances of these experiments were evaluated using various performance metrics. From the obtained results. With different activation functions, we were able to acquire data demonstrating that (Inception-V3 and DenseNet-121) offer better outcomes when compared to other architectures listed in this study, where (accuracy is higher than 94%) with the alternative activation functions of ReLU. Due to the good performance achieved by these models, we suggest utilizing these methods to support medical professionals' decision-making in clinical images, especially in underdeveloped regions with limited access to well-trained radiologists with adequate COVID-19 imaging expertise.[25][36] And notice that one cannot find the perfect activation function which fits every CNN model and produces accurate results, and has a good performance, we conclude that the best activation function completely depends on the model and how the dataset is behaving with respect to the model. (Dubey, S. R., et, al, 2021) [18] (Nwankpa, C., et.al., 2018) [37] Therefore, research is still going on in this domain to find better activation functions for building robust models. Finally, we intend to compare basic activation functions with complicated and adaptive activation functions on most DCNN architectures in the next work.

Bibliography

1. Soares, E., Angelov, P., Biaso, S., Froes, M. H., & Abe, D. K. (2020). SARS-CoV-2 CT-scan dataset: A large dataset of real patients CT scans for SARS-CoV-2 identification. MedRxiv.
2. Ratnawati, D. E., Marjono, Widodo, & Anam, S. (2020, September). Comparison of activation function on extreme learning machine (ELM) performance for classifying the active compound. In AIP Conference Proceedings (Vol. 2264, No. 1, p. 140001). AIP Publishing LLC.
3. Wang, L., Lin, Z. Q., & Wong, A. (2020). Covid-net: A tailored deep convolutional neural network design for detection of covid-19 cases from chest x-ray images. *Scientific Reports*, 10(1), 1-12.
4. Bhandary, A., Prabhu, G. A., Rajinikanth, V., Thanaraj, K. P., Satapathy, S. C., Robbins, D. E., ... & Raja, N. S. M. (2020). Deep-learning framework to detect lung abnormality—A study with chest X-Ray and lung CT scan images. *Pattern Recognition Letters*, 129, 271-278.
5. Shazia, A., Xuan, T. Z., Chuah, J. H., Usman, J., Qian, P., & Lai, K. W. (2021). A comparative study of multiple neural networks for detection of COVID-19 on chest X-ray. *EURASIP journal on advances in signal processing*, 2021(1), 1-16.
6. Elfwing, S., Uchibe, E., & Doya, K. (2018). Sigmoid-weighted linear units for neural network function approximation in reinforcement learning. *Neural Networks*, 107, 3-11.
7. Aggarwal, C. C. (2018). *Neural networks and deep learning*. 10, 978-3.

8. Gaál, G., Maga, B., & Lukács, A. (2020). Attention u-net based adversarial architectures for chest x-ray lung segmentation.:2003.10304.
9. Ratnawati, D. E., Marjono, Widodo, & Anam, S. (2020, September). Comparison of activation function on extreme learning machine (ELM) performance for classifying the active compound. In AIP Conference Proceedings (Vol. 2264, No. 1, p. 140001). AIP Publishing LLC.
10. Jia, H., Zhao, J., & Arshaghi, A. (2021). COVID-19 Diagnosis from CT Images with Convolutional Neural Network Optimized by Marine Predator Optimization Algorithm. *BioMed Research International*, 2021.
11. Roberts, D. A., Yaida, S., & Hanin, B. (2021). The principles of deep learning theory. arxiv preprint arXiv:2106.10165.
12. Elfving, S., Uchibe, E., & Doya, K. (2018). Sigmoid-weighted linear units for neural network function approximation in reinforcement learning. *Neural Networks*, 107, 3-11.
13. Jeyanthi, S., & Subadra, M. (2014, May). Implementation of single neuron using various activation functions with FPGA. In 2014 IEEE International Conference on Advanced Communications, Control and Computing Technologies (pp. 1126-1131).
14. Guccione, S. A., & Gonzalez, M. J. (1994, July). A neural network implementation using reconfigurable architectures. In Selected papers from the Oxford 1993 international workshop on field programmable logic and applications on More FPGAs, Oxford: Abingdon EE&CS Books (pp. 443-451).
15. Zhang, M. (2021). A Comparison of Image Classification with Different Activation Functions in Balanced and Unbalanced Datasets (Doctoral dissertation, Virginia Tech).
16. Klambauer, G., Unterthiner, T., Mayr, A., & Hochreiter, S. (2017). Self-normalizing neural networks. *Advances in neural information processing systems*, 30.
17. Qiumei, Z., Dan, T., & Fenghua, W. (2019). Improved convolutional neural network based on fast exponentially linear unit activation function. *Ieee Access*, 7, 151359-151367.
18. Dubey, S. R., Singh, S. K., & Chaudhuri, B. B. (2021). A comprehensive survey and performance analysis of activation functions in deep Learning. arXiv preprint arXiv:2109-14545.
19. Ramachandran, P., Zoph, B., & Le, Q. V. (2017). Searching for activation functions. arXiv preprint arXiv:1710.05941.
20. Sharma, S., Sharma, S., & Athaiya, A. (2017). Activation functions in neural networks. *towards data science*, 6(12), 310-316.
21. Agarap, A. F. (2018). Deep learning using rectified linear units (relu). arxiv preprint arxiv:1803.08375.
22. Misra, D. (2019). Mish: A self-regularized non-monotonic neural activation function. arXiv preprint arXiv:1908.08681, 4(2), 10-48550.
23. Zhu, J., & Chen, Z. (2020, November). Comparative analysis of various new activation functions based on convolutional neural network. In *Journal of Physics: Conference Series* (Vol. 1676, No. 1, p. 012228). IOP Publishing.
24. He, K., Zhang, X., Ren, S., & Sun, J. (2015). Delving deep into rectifiers: Surpassing human-level performance on imageNet classification. In *Proceedings of the IEEE international conference on computer vision* (pp. 1026-1034).
25. Pooja, A, et.al. 2021 Deep learning system detection of COVID-19, NVEO Natural Volatiles & Essential Oils, *Nat. Volatiles & Essent. Oils*, 2021; 8(5): 3165-3173.
26. Clevert, D. A., Unterthiner, T., & Hochreiter, S. (2015). Fast and accurate deep network learning by exponential linear units (elus). arXiv preprint arXiv:1511.07289.

27. El Asnaoui, K., Chawki, Y., & Idri, A. (2021). Automated methods for detection and classification pneumonia based on x-ray images using deep learning. In Artificial intelligence and blockchain for future cybersecurity applications (pp. 257-284).
28. Zafar, I., Tzanidou, G., Burton, R., Patel, N., & Araujo, L. (2018). Hands-on convolutional neural networks with TensorFlow: Solve computer vision problems with modeling in TensorFlow and Python. Packt Publishing Ltd.
29. Xiao, J., Wang, J., Cao, S., & Li, B. (2020, April). Application of a novel and improved VGG-19 network in the detection of workers wearing masks. In Journal of Physics: Conference Series (Vol. 1518, No. 1, p. 012041).
30. <https://analyticsindiamag.com/the-number-game-behind-advanced-activation-functions-in-machine-learning>
31. <https://sefiks.com/2017/11/10/softsign-as-a-neural-networks-activation-function/>
32. Pedamonti, D. (2018). Comparison of non-linear activation functions for deep neural networks on MNIST classification task. arxiv preprint arXiv:1804.02763.
33. Murugan, R., & Goel, T. (2021). E-DiCoNet: Extreme learning machine-based classifier for diagnosis of COVID-19 using deep convolutional network. Journal of Ambient Intelligence and Humanized Computing, 1-12.
34. Hendrycks, D., & Gimpel, K. (2016). Gaussian error linear units (gelus). arxiv preprint arxiv:1606.08415.
35. Silva, P., Luz, E., Silva, G., Moreira, G., Silva, R., Lucio, D., & Menotti, D. (2020). COVID-19 detection in CT images with deep learning: A voting-based scheme and cross-datasets analysis. Informatics in medicine unlocked, 20, 100427.
36. El Asnaoui, K., & Chawki, Y. (2021). Using X-ray images and deep learning for automated detection of coronavirus disease. Journal of Biomolecular Structure and Dynamics, 39(10), 3615-3626.
37. Nwankpa, C., Ijomah, W., Gachagan, A., & Marshall, S. (2018). Activation functions: Comparison of trends in practice and research for deep learning. arXiv preprint arxiv:1811.03378.
38. Krizhevsky, A., & Hinton, G. (2010). Convolutional deep belief networks on cifar-10.40(7), 1-9.
39. Rosebrock, A. (2017). Deep learning for computer vision with python: starter bundle. PyImageSearch.
40. Wang, X., Ren, H., & Wang, A. (2022). Smish: A Novel Activation Function for Deep Learning Methods. Electronics, 11(4), 540.
41. Naveen, P. (2022). Phish: A Novel Hyper-Optimizable Activation Function.
42. Kamath, U., Liu, J., & Whitaker, J. (2019). Deep learning for NLP and speech recognition (Vol. 84).
43. Ketkar, N., & Santana, E. (2017). Deep learning with Python (Vol. 1). Berkeley: Apress.
44. Gad, A. F., Gad, A. F., & John, S. (2018). Practical computer vision applications using deep learning with CNNs. New York, NY, USA: Apress.
El Asnaoui, K., Chawki, Y., & Idri, A. (2021). Automated methods for detection and classification pneumonia based on x-ray images using deep learning. In Artificial intelligence and blockchain for future cybersecurity applications (pp. 257-284).
45. Haldar, M., Abdool, M., Ramanathan, P., Xu, T., Yang, S., Duan, H., ... & Legrand, T. (2019, July). Applying deep learning to airbnb search. In Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (pp. 1927-1935).
46. <https://paperswithcode.com/methods/category/activation-functions>
47. Khan, S., Rahmani, H., Shah, S. A. A., & Bennamoun, M. (2018). A guide to convolutional neural networks for computer vision. Synthesis Lectures on Computer Vision, 8(1), 1-207.