JOURNAL LA MULTIAPP

VOL. 03, ISSUE 04 (174-197), 2022
DOI: 10.37899/journallamultiapp.v3i4.691

# Intelligent Diagnosis of Covid-19 Based on CNN-PNN

**Abbas Akram Khorsheed**[1]

[1]*Computer science department, College of science, Mustansiriyah University, Baghdad, Iraq*

**Corresponding Author: Abbas Akram Khorsheed*
*Email:* abbasarab2000@uomustansiriyah.edu.iq

Article Info

*Article history:*
Received 14 July 2022
Received in revised form 12 August 2022
Accepted 15 August 2022

*Keywords:*
COVID-19
PNN
CNN
CT scan
AI
D

**Abstract**

*Today the whole world suffers and fears the epidemic of the Coronavirus and the developed waves in it, as we have now reached the fourth wave, and this is a serious matter. Where the statistics of the Coronavirus in the current data showed that 213 countries are affected by this epidemic, and about 6 millions of deaths are recorded. This virus spreads rapidly, and patients mainly suffer from breathing. The patient who suffers from pre-existing health problems will be more likely to contract this disease, so there was an urgent need for artificial intelligence to enter to quickly detect this virus, so the world turned to deep learning, which is one of the most powerful methods and techniques for classification because of its use of Bayas Rule, where there is no possibility of error. This paper proposes CNN (Convolutional Neural Networks) and PNN (Proprestitc Neural Networks) mixed tomography scanning model to classify Covid-19 images, the proposed network called the CNN-PNN model. The CNN-PNN model can use CNN to compute the dependency and continuity features of the output of the middle layer of the PNN model, and correlate the properties of these middle levels with the final full network to predict the classification.*

## Introduction

The emerging coronavirus SARS-CoV-2 (Coronavirus causing severe acute respiratory syndrome 2), a new coronavirus, appeared in December 2019 to begin the Covid-19 pandemic, a respiratory disease that has proven difficult to treat, with symptoms ranging from mild to severe. Death as a result of organ dysfunction from a simple and self-limited respiratory infection to mortality, acute progressive pneumonia, multiple organ failure, and death (Dai et al., 2020). There are compelling reasons to worry about the repercussions of this viral infection as the epidemic grows, the number of confirmed cases increases, patients with severe respiratory failure and patients with cardiovascular problems (Song et al., 2021). Identifying relevant techniques to find answers to the challenges of Covid-19 has received a great deal of attention. However, another key issue that researchers and policymakers must address is the ever-increasing volume of data, known as big data, that poses challenges in the fight against HIV. This demonstrates how and to what extent artificial intelligence (ARTIFICIAL INTELLIGENCE) can play an important role in the global development and modernization of healthcare systems. According to Karasmanaki & Tsantopoulos (2021) Artificial intelligence has recently received increasing research efforts to deal with complex problems in a variety of areas, including engineering, health, economics, and psychology. As a result, such a critical situation requires the mobilization and maintenance of medical, logistical, and human resources, and artificial intelligence can facilitate this, but also save time in a period when one hour can save lives in all sites where CORONAvirus is spreading. You're killing lives. With the recent popularity of artificial intelligence, you are taking people's lives. With the current popularity of AI applications in healthcare settings, this may help reduce the number of unwanted deletions, increase productivity and efficiency in large sample research (Elsalem et al., 2021; Pokhrel & Chhetri, 2021) and improve the forecast and accuracy of the intended diagnosis (Moser et al., 2021). In addition, research on viral activity modeling in any country can benefit from the use of big data. The data is analyzed so that healthcare policymakers can better prepare their countries for epidemics and make better judgments. Bashitialshaaer et al. (2021) While ARTIFICIAL INTELLIGENCE can help with treatment and crisis management approaches, and diagnostic and improvement methods, such as medical imaging and image processing techniques, it has not yet been used in a desirable and well-customized way for healthcare delivery systems. They are fighting Covid-19, and they are winning. For example, image-based medical diagnosis may benefit greatly from the useful contribution of ARTIFICIAL INTELLIGENCE, allowing for the rapid and accurate diagnosis of COVID-19 and thus saving lives. Customizing AI technology to deal with COVID-19 issues may help bridge the gap between AI-based approaches and medical procedures and treatments (Ahmed et al., 2021).

## Literature Review

This paragraph will discuss a range of studies that have utilized COVID-19 detection with deep learning:

Covid-19 was diagnosed using smart technologies and vital indicators, which were examined, categorized, and compared. The AI-PSR system, which combines artificial intelligence and physiological sensors, might aid clinicians in making choices and predicting the development of respiratory failure in Covid-19 patients. To minimize direct interaction between the patient and the doctor or nursing staff, the physiological characteristics of Covid-19 patients can be remotely relayed to the doctor using specific wireless technologies such as Wi-Fi and Bluetooth. The AI-PSR model's outcome allows data to be gathered and connected to what will happen next to prevent respiratory failure and support the patient with a mechanical ventilator have been presented (Bharath et al., 2018).

The model was trained using an open online data set comprising blood samples from 485 patients in Wuhan, China, and a temporary deep learning strategy based on a time-conscious T-LSTM neural network. Other scientists neglect dynamic correlations in uniformly uncollangled time series, which our technology can understand. Our method, in particular, used biomarkers and irregular periods to estimate the diagnosis of COVID-19 patients. Then, for COVID-19, we used T-LSTM units to build patient representations that were used to determine patient stages and describe the progression of the disease have been presented (Mohanapriya & Babu, 2017).

CNN's patch-based approach was used with a small number of training parameters to diagnose COVID-19. The proposed approach was inspired by our statistical analysis of possible CXR imaging markers. According to empirical data, our approach delivers advanced performance and provides clinically interpretable line maps, which are used to diagnose COVID-19 and screen patients that have been presented (Nguyen et al., 2017).

A basic CNN architecture with a small number of parameters distinguishes COVID-19 from traditional X-rays. Second, a qualitative study was conducted using a technique known as Class Activation Maps to analyze CNN's judgments (CAM). CAMs may be utilized to map the activations that most affected the CNNs' choice to return to the original image to display the more distinct regions on the input image. The chest x-ray images utilized in this investigation were created using three publically available sources. A big dataset covering viral illnesses other than COVID-19, bacterial infections, and plain x-rays was employed, as well as two COVID-19 x-ray image datasets., have been presented (Omara et al., 2016).

New approaches for developing a CNN deep learning model to detect and characterize COVID-19 patients have been created using chest X-ray images. The findings suggest that image preprocessing improves the quality of the input image data utilized to train deep learning models. To feed all three channels of CNN models, the transfer learning strategy involves removing superfluous areas, adjusting the image contrast-to-noise ratio, and synthesizing pseudo-color images. The good classification performance discovered is particularly noteworthy, giving a solid foundation for future research into fine-tuning deep learning models for discovering, verifying, and performing COVID-19 cases utilizing vast and heterogeneous image datasets have been presented (Gore et al. nd).

COVID-19 categorization from natural and viral pneumonia on CXRs is automated using deep learning-based algorithms, as well as the identification of COVID-19 biomarker-indicative regions. To pre-process and segment lung areas, we employed the DeepLabV3+ technique and then clipped off lung regions. The cut lung sections were utilized as inputs to several deep convolutional neural networks for COVID-19 prediction (CNNs). The data collection was skewed; the great majority of the images were normal, with only a few COVID-19 and pneumonia images thrown in for good measure. Utilized five different approaches to deal with the unbalanced distribution and avoid skewed classification results: (1) widening the image to incorporate more images of minority concerns; (3) lack of sampling for majority groups; (iv) over-sampling of minority groups; and (5) a mixed re-sampling method of over-sampling and under-sampling. The best-performing techniques from each approach were combined as the group classifier using two voting procedures. Finally, we utilized a CNN stellar map to identify COVID-19 biomarker suggestive areas for interpretation that have been presented (Kandgaonkar et al., 2015).

**Artificial Intelligence and COVID-193**

The Corona pandemic illustrated how artificial intelligence has been ingrained in our daily lives and how it is fast and entering numerous professions in a variety of ways. Artificial intelligence is no longer a science-fiction notion, but a reality that was recently shown during the Corona outbreak. Shortly after the pandemic began, artificial intelligence aided humans in identifying SARS-CoV-2 (Severe acute respiratory syndrome coronavirus 2), the pathogenic coronavirus. Scientists also helped researchers find the virus's characteristics by assisting with the quick study of the virus's genetic information (DNA). Scientists utilized artificial intelligence to figure out how rapidly the virus may change, as well as design and test vaccines against the Coronavirus. Artificial intelligence and machine learning algorithms saved some lives during the coronavirus outbreak. Artificial intelligence and machine learning techniques were employed for diagnosis. It can read a large number of chest X-rays in a fraction of the time that individuals can. Clinicians have been able to identify and monitor coronavirus patients more quickly as a result of this (Vijay & Indumathi, 2021). In Nigeria, for example, technology has been utilized on a small but practical basis to help individuals estimate the risk of infection. People may be asked questions online, and medical advice may be offered remotely based on their replies, or they may be persuaded to travel to the hospital in extreme instances.

Artificial intelligence has benefited doctors in South Korea in recognizing ill patients quickly, which is one of the most important challenges in the Corona crisis. They were able to pinpoint the hotspots of the Coronavirus epidemic in the country at the start of the crisis as a result of this. A company in Seoul (Seegene) utilized artificial intelligence to develop tests to detect the Coronavirus in weeks, rather than months, as it would have taken without it. This was at a time when many countries around the world were still considering the possibility of imposing a general closure due to the pandemic.

Artificial intelligence has been widely utilized in the healthcare sector due to the virus's urgent challenges, particularly in the areas of diagnostics and drug and vaccine development, since the outbreak; in the public and private sectors around the world, artificial intelligence has been widely utilized to combat the "Covid-19" epidemic; in the public and private sectors around the world, artificial intelligence has been widely utilized to combat the "Covid-19" epidemic; in the healthcare sector, AI has been widely utilized due to the virus's urgent challenges, particularly in the areas of diagnostic.

Given the severity of the "Covid-19" epidemic, most nations have implemented physical barriers and encouraged individuals to keep as close to their houses as possible. Despite this, government organizations, businesses, and institutions were able to keep operating by deploying artificial intelligence capabilities to activate distant communication channels. The "Covid-19" outbreak revealed that robots can do the tasks required to respond to emergencies in environments that are deemed dangerous to humans, like the present pandemic. Robots have been built to roam the streets in numerous cities across the world, urging inhabitants to remain at a safe distance and follow safety steps to avoid human exposure to the virus. Artificial intelligence is being utilized effectively to combat the pandemic in several areas, the most noteworthy of which are the three mentioned below: In some cases, robots have been utilized to remotely or directly detect persons with disease symptoms. I health care resource management; (ii) Service and Research Department; and (3) drug and vaccine development, and in some cases, robots have been utilized to remotely or directly detect persons with disease symptoms. It also uses artificial intelligence to make treating patients safer for physicians and nurses. The "Covid-19" epidemic has demonstrated that public and private organizations and agencies can use machine learning methods – a subset of artificial intelligence utilized to improve systems and skills for self-learning without being programmed – to quickly achieve excellence in areas such as communications, provision of basic services, and customer care. Gaining a deeper knowledge of the core difficulties that help in the virus's battle (Resmi & Raju, 2019).

For example, BlueDot, an artificial intelligence company, utilized machine learning and other artificial intelligence applications to monitor infectious disease outbreaks by alerting other institutions to a sudden rise in pneumonia cases in the city of Wuhan, before the World Health Organization assessed the virus's outbreak and subsequently classified it as a pandemic9; this demonstrates how AI can help analyze data at different stages of an epidemic, thus demonstrating that Time is undoubtedly a critical factor in responding to a pandemic with rapid virus spread, as artificial intelligence applications can save thousands of lives by reaching the sequence of the "Covid-19" genome, which consists of 30,000 genetic bases, and thus performs an active evaluation and tracking function.

**AI utilizing hyper Deep Learning Classification**

Deep learning has emerged as a particularly useful technology over the last several decades. This may be attributed to its capacity to process massive quantities of data. Hidden layers have received interest in fields such as pattern recognition when standard methods are unsuccessful. Deep neural networks often take the form of convolutional neural networks (sometimes abbreviated as CNNs).

Since the 1950s, when artificial intelligence was still in its infancy, researchers have sought to construct a system that can grasp visual input. The years that followed ultimately led to the development of a discipline that is today known as computer vision. When compared to the most advanced image recognition algorithms, an artificial intelligence model that was developed by a team of researchers at the University of Toronto in 2012 represented a quantum leap in the field of computer vision. This model was able to recognize images in a significantly more accurate manner.

The artificial intelligence system known as AlexNet, which was named after its primary developer Alex Krizhevsky, triumphed in the ImageNet computer vision competition in 2012 with an astounding accuracy of 85 percent. The runner-up in the competition earned a decent score of 74% on the examination. AlexNet made use of convolutional neural networks, a kind of neural network that was designed to simulate human vision. CNN's have developed into an essential component of a wide variety of computer vision applications over time, and as a result, they are now included in every online computer vision training course. Now that we have that out of the way, let's have a look at how CNNs operate. Deep learning is a subfield of machine learning that employs the use of artificial neural networks (ANN), which are computer programs that mimic the structure of the human brain and can learn from very large volumes of data. Machine learning, for example, enables computers to learn from their own experiences and create capabilities without the need for human input at any stage of the process. An algorithm that uses deep learning will do a job several times, each time making little adjustments to the previous output to better it. This is analogous to the way that a person would gain knowledge via experience. Deep learning is a word that is used to describe neural networks that comprise several (deep) layers that make it possible for the network to learn. Deep learning may be used to solve any issue that involves "thinking," since it can be trained to solve such problems. Deep learning is a kind of machine learning that helps computers to solve difficult issues, even when they are given a massive, unstructured, and interconnected data set to work with. The more successfully the learning algorithms complete their tasks, the more advanced they are (Kabir et al., 2019). As shown in Fig.1

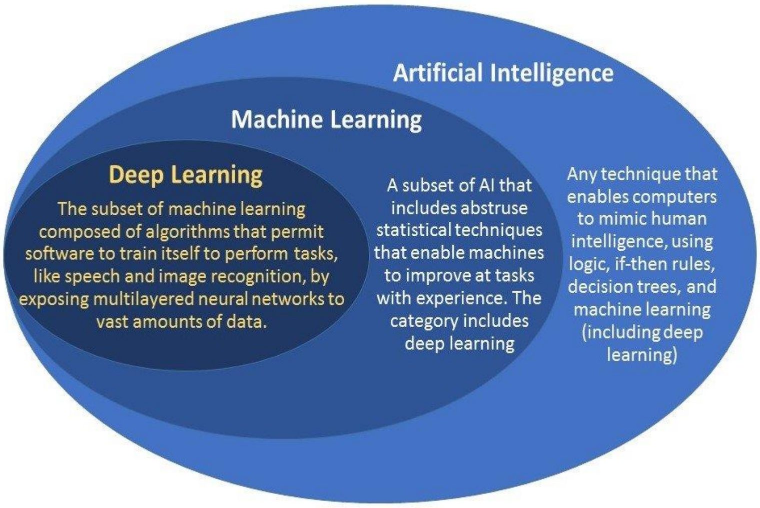



*Figure 1. AI self-learning [18].*

**Convolutional Neural Networks (CNN)**

CNN is a kind of deep learning that deals with collecting features from an image and video data, building a neural network by assigning its weights and converting them using a filter to categorize and choose an image. When it comes to handling any kind of image or video processing data, CNN is the first option for every data world. Using our layers also makes it simple to implement and customize the transportation learning model. The convolutional neural network (CNN) is a specialized kind of neural network that is used for redirection. This form of the neural network is employed in computer vision and artificial intelligence challenges since it is effective at solving these types of issues. The term "bypass neural networks" was coined because the circumvention relies on at least one of the levels of the network's layers as a fundamental step. CNN is made up of several layers, both visible and hidden, that lie in between its input and output layers. Layers may have many different forms, including bypass layers, assembly layers, and complete associated layers. The number of levels and kinds of layers that are employed in CNN designs is app-specific. Hamd & Mohammed (2019) CNN layers may include anything from a few to hundreds of filters that examine all channels and go through the input data.

The most abstract serial properties are extracted using CNN layers. At the initial levels, start with edges and corners and work your way to the entire faces and artifacts in the deeper layers. During training, the network learns the features that must be extracted to answer the problem. CNN has the advantage of not having to know well or humanly to develop features, making it a versatile solution to a wide range of computer vision challenges. Fig.2 shows the architecture of a CNN.

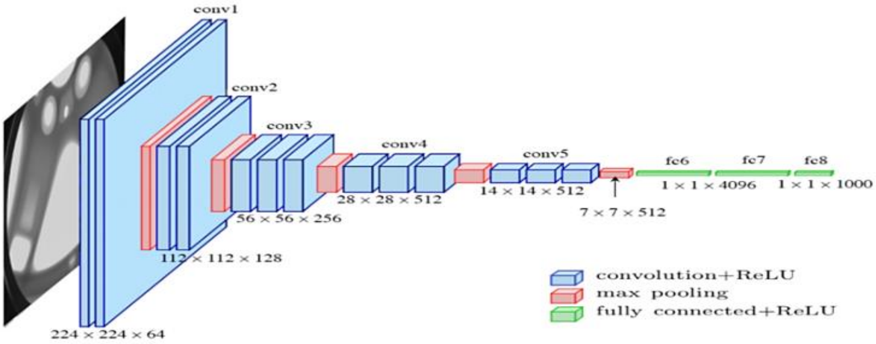



*Figure 2. Architecture of CNN*

In the subsections below, many CNN-related terms and terminology are explained:

**The input layer of CNN**

The input layer contains all of CNN's data. It is usually the image's pixel matrix in an image processing neural network. The grayscale image's width and height can be utilized to determine the size of the input image (Huo et al., 2015) as shown in Fig.3.

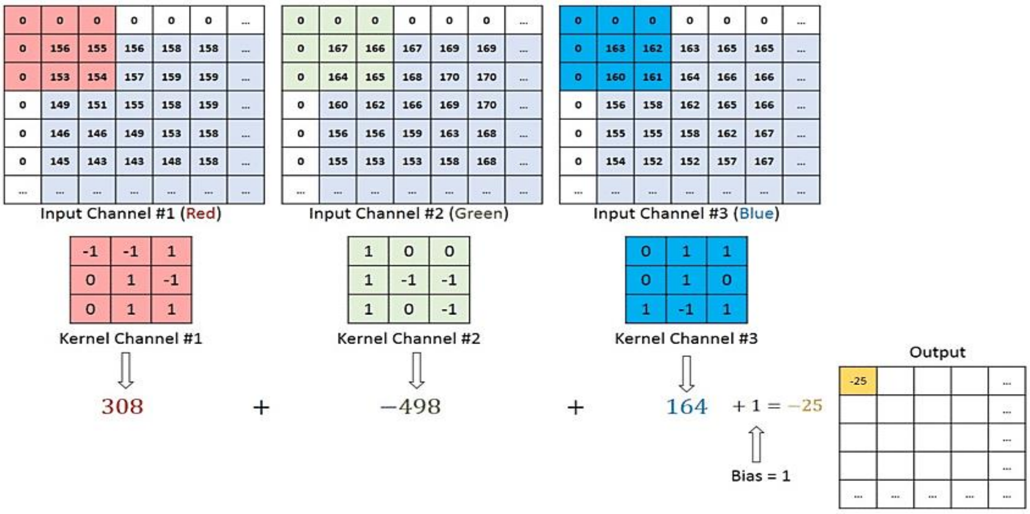



*Figure 3. input layer RGB Image of CNN (Huo et al., 2015).*

**Convolution Operation and Filter**

Image characteristics may be extracted with the help of the Convolution layer. The low-level bypass layer is responsible for the collection of shallow features (e.g. edges, lines, and angles). Through the combination of low-level characteristics, the high-level bypass layer can learn abstract qualities. By linking the wrap core of the preceding layer to the bypass layer at a specified size, as illustrated in equation 1, the bypass layer is given access to a large number of feature activation maps.

To extract image properties, the bypass layer is used. The low-level bypass layer is responsible for the collection of shallow features (e.g. edges, lines, and angles). B characteristics from lower levels, the high-level bypass can learn more abstract qualities. By linking the wrap core of the preceding layer to the bypass layer at a specified size, as illustrated in equation 1, the bypass layer is given access to a large number of feature activation maps.

$$X_j^l = f\left[\sum_{i \in M_j} (X_i^{l-1} * K_{ij}^l + b_j^l)\right], \quad (1)$$

The nucleus is made up of just a select few entries taken from the actual world. It's a scaled-down replica of the one we had before. The candidate is often arranged in the sequence 1 1, 3 3, 5 5, and 7 7, and it is modified throughout the training phase so that it matches the weights of the neural network. In mathematics, circumvention refers to the process of combining two functions whose values are determined to demonstrate how the behavior of one form is influenced by the behavior of the other. The method of avoiding detection. is seen in Fig.4.

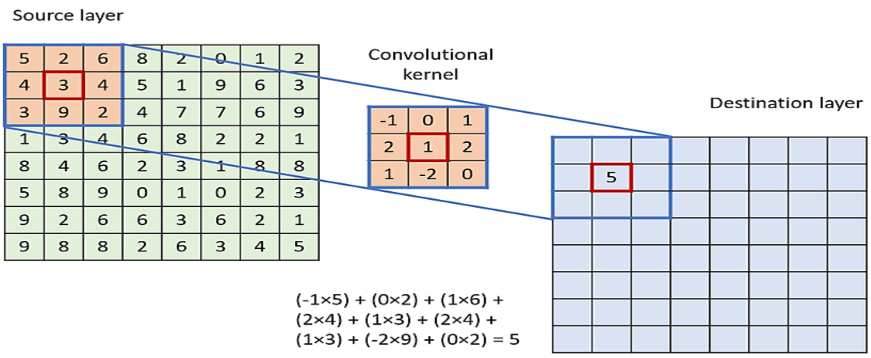



*Figure 4. Performing a convolutional operation*

The nucleus is a left-to-right movement in the image, and the bypass begins in the upper left corner. The nucleus moves one object down the hill before returning to the left when it reaches the top right corner of the image. This is repeated until the nucleus reaches the bottom right corner of the input image. Map features are the product of this process. The speed at which the filter passes through the image is controlled by a variable called stride. The filter will move only one pixel across the image if the step is set to 1.

### Image Features

Points of interest, edges, lines, and other characteristics in image graphs provide a plethora of information about the content of the image. They're utilized to characterize places in an image in a variety of image analysis applications such as reconstruction, identification, and matching (Sameh et al., 2021).

### Zero Padding

The process of adding zeros to the matrix of the input image to allow the maintenance of the original size of the input image throughout the whole of any convolution phase is referred to as zero padding. There are two different kinds of padding. The first one is referred to as legitimate, and it implies that there is no padding and that the convolutional layer is never padded at any point, which leads to a reduction in the amount of input data. The second kind is similar to the first in that the initial input image is padded before being convolved. As a direct consequence of this, the dimensions of both the output and the input are the same. Figure 5 does not have any padding. In a nutshell, the input volume that we provide serves as an indication of padding, which is required for our kernels to be able to accommodate the input matrices. Each zero padding, which consists of adding one row or column on either side of zero matrices, or valid padding may be used here (cutting away the section of the image that does not fit) (Sharifi & Eskandari, 2016)

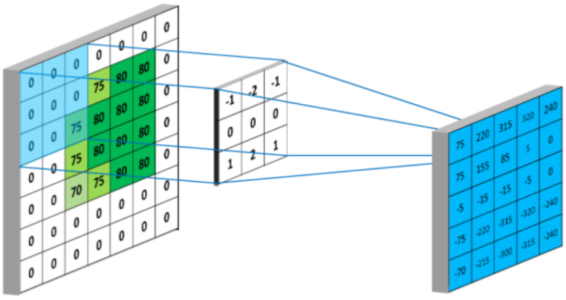


*Figure 5. Zero Pegging [22].*

**Batch Size, Iterations, and Epoch**

Batch sizes, iterations, and epochs become significant factors in machine learning because of the very large amount of data that must be processed. To overcome this challenge, we will need to break the data down into smaller pieces and then incorporate those pieces into our model in real time. Additionally, we will need to adjust the weights of the neural networks after each stage so that they are consistent with the data. The length of time that the network will continue to study data is represented by an epoch, which is a discrete period. To train the network for a single pass through the network during a single epoch, the whole data set is used. The batch size determines the total number of I/O pairs that are made available on the network at any one moment. On the other hand, iterations refer to the number of batches that must be completed to complete one period (Eskandari & Toygar, 2015).

**Cost Function**

The feedback on the performance of the network may be obtained via the use of the cost function. The elimination of this functionality is one of the outcomes of the process of deep learning, and it is the functionality that the network is working to eradicate. An optimizer is necessary to reduce the amount of functionality that a network costs. Adaptive Moment Estimation Optimizer, more often abbreviated by its acronym ADAM Optimizer, is the tool that should ideally be used in a scenario like this one. In place of the traditional random gradient descent approach, the ADAM optimization methodology may be used to recursively update network weights depending on training data. This is accomplished in place of the normal way of using random gradient descent (Alhamrouni, 2017).

**Layers of CNN**

CNN's layers A convolution neural network has a lot of layers. The following subsections outline the most significant layers:

***Layer of Convolutional Data***

The primary objective of the convolution process is to learn the characteristics of the image by applying a filter, also known as a kernel, to the input data. This allows the process to extract features from the input image while still preserving the spatial connection between the pixels in the image. The convolution layer serves as the basis for a convolutional neural network (CNN). Performs a wrapping operation on the image that has been given. It is used to extract the features of the image that is being entered. While the first convolution layer is responsible for collecting low-level properties such as lines, corners, and edges, subsequent levels of convolution are responsible for extracting higher-level features from the input image. The equation of layers is shown here by Equation (2) (Emeršič et al., 2017):

Where a1 refers to the set activations that were generated from the feature map, a0 stands for the activations that were input, w stands for the weight, b stands for the bias, * stands for the convolution operation, and a0 stands for the activations that were input. The height and width of the kernel are both less than the corresponding dimensions of the input image. The feature map is generated as a result of the kernel moving over the image (convolve with). The term "convolution" refers to the process of adding the product of the kernel element to the initial image.

***Pooling Layers***

To minimize volatility and computational complexity, neural networks (NN) use pooling. Beginners frequently use the assembly approach without understanding why it is utilized. The following is a comparison of three common assembly methods.

The following are the three types of aggregating operations:

Max pooling: A batch's maximum pixel value is provided.

Min pooling: Defines the batch's minimum pixel value.

Average pooling is the process of averaging the values of all pixels in a batch.

The term "batch" refers to a group of pixels with a size equal to the filter's size, which is defined by the image size. A 9x9 filter was chosen in the following example. The aggregation method's output differs depending on the filter size's variable value. As a result, the convolutional layers that form the feature maps are spatially large. The pooling layer reduces the accuracy of the features. The maximum and average pooling are shown in Fig.6. As a result, when we aggregate an image, we do not remove all of the data; instead, we create a summary of all of the values discovered.

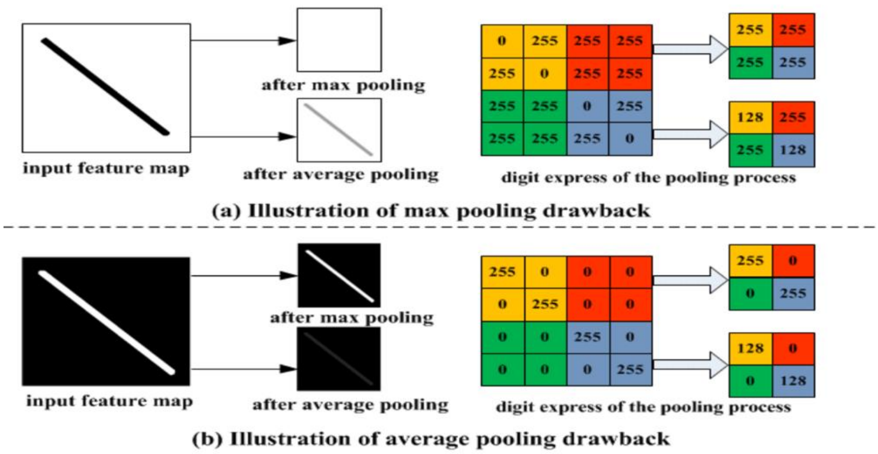



*Figure 6. max or average pooling.*

***Fully Connected Layer (FC)***

Weights and biases, in addition to neurons, are included in the FC layer, which links neurons between layers. These layers make up the last few levels of the CNN architecture before the output layer. This flattens the FC layer and feeds it the preceding layers' input image. To conduct arithmetic function operations, more FC layers are added on top of the flat vector. This is where the classifying process begins.

***Dropout***

As demonstrated in Figure, the essential premise of Dropout is to lose a percentage of the hidden layer neurons each time, which is similar to training on new networks each time, thus diminishing the connections between neurons (7). The Calculation technique for redirecting diffusion changes from equation (1) to equation (2) when the Dropout strategy is added (4).

$$r^{(-1)} \sim Bernolli\,(p) \tag{3}$$

$$x_j^l = f[\sum_{i \in M_j} (r^{(l-1)} * x_i^{l-1}) * W_{ij}^l + b_j^l] \tag{4}$$

When all features are connected to the FC layer in the training dataset, overfitting is common. Using a model that performed well on training data but now performs poorly on new data, where equation (3) states that each value in the vector (1) or is a Bernoulli distribution with probability p to generate values of 0 and 1, i.e., each layer of the model blocks is part of the 1lix input vector from the first layer l through the vector (1) LR, making the model approximate to the subnetwork model taken from the overall network model and the output o as shown in Fig.7.

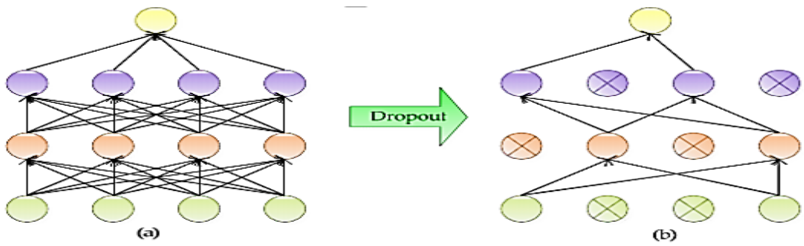



*Figure 7. Neural Network utilizing Dropout*

**Activation Function Layer**

Although not all neurons exhibit activating activity, as seen in Fig.8, The hidden neurons in the output layer have activation functions, but the neurons in the input layer do not.

Activation functions change the received input to keep the values within a safe range. Because the data in the input layer are generally zero-centric and have already been scaled appropriately, no transformation is necessary. When these values are multiplied by weights and put together, they quickly transcend the range of their initial scale, activating the activation functions, which bring the values back into this acceptable range and make them useable. Activation functions must be non-linear and continually differentiable to be useful. The nonlinearity of the neural network allows it to be utilized as a global approximation; the gradient-based optimization techniques need a continuously differentiable function, which allows for efficient error back-propagation across the network.

- The inside of a neuron:
- The activation function is received by a neuron or the entire neural layer.
- The weighted sum of the input values is added.
- The activation function is applied to the weighted sum of the input values, and the transformation occurs.
- This changed value is sent to the next rung [36]:

$$\text{sigmoid:} \quad f(x) = \frac{1}{1+e^{-x}}, \tag{5}$$

$$\text{tanh:} \quad f(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}, \tag{6}$$

$$\text{ReLU:} \quad f(x) = \max(0, x), \tag{7}$$

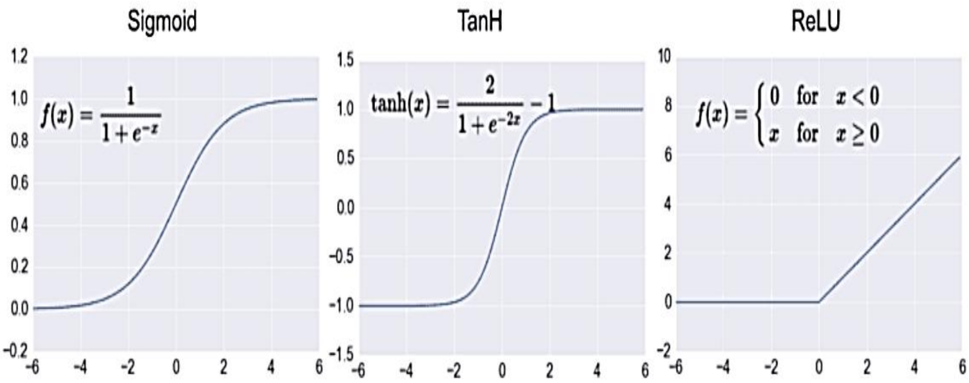



*Figure 8: activation function [36].*

**Probabilistic Neural Network (PNN)**

PNN is a radial basis network that is well-suited to classification problems. It uses the Bayes classification technique for decision-making, as well as the Pazen window, which uses training data to compute the probability density function (PDF) for each category. In 1999, Donald Specht introduced this neural model, demonstrating that the Bayes Parzen classifier may be broken down into a large number of simple sub-classifiers and run in parallel. This neural model has a fast training speed and can easily add more pattern neurons to the pattern layer, and it performs well in object identification with lower complexity (Jan, 2017).

**Architecture of Probabilistic Neural Network**

Figure 9 shows the structure of this model, a multi-layered front feeding network. The input layer, the padding layer, the assembly layer, and the output layer form the four layers. Uses a method of supervised learning training (Jan, 2017).

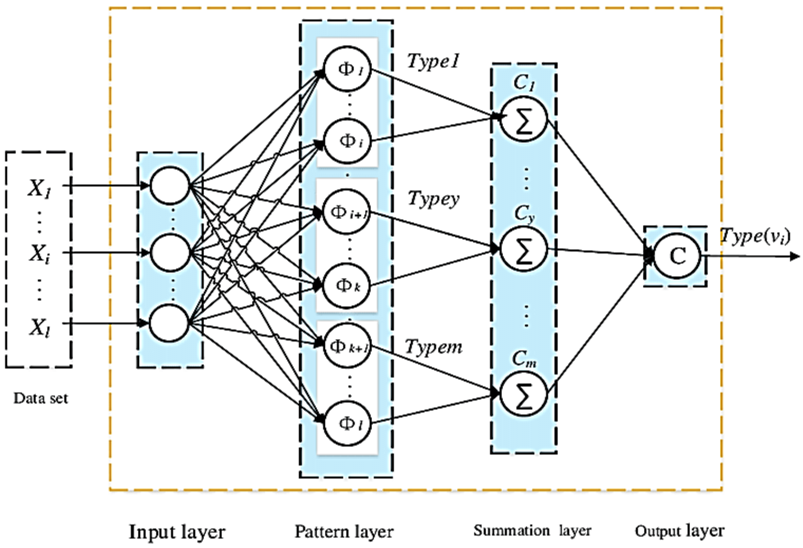



*Figure 9. Probabilistic Neural Network Architecture (Jan, 2017).*

PDF files for feature vectors obtained from training data for each category are estimated using the Gaussian function. Using these assessed densities, Bayes's decision-making is used to create the final classification option. You receive all the neural nodes in the pattern layer of the input layer, fully connected to them, vector input features (x). In the pattern layer, each training sample contains one neuron. The distances between the input feature vector and the training samples are calculated by each neural node in the pattern layer, and the result is a vector whose members reflect how close the input is to a training sample. : Any measure of distance, such as Euclid distance, square Euclid distance, and Manhattan distance, can be used to calculate distances.

$$G(x) = \exp(-D/(2\sigma^{\wedge}2)) \tag{8}$$

Where

- G (x): Output of the neuron pattern node.
- X: Input features vector to be assigned in Ci class
- D: The distance between the input features vector and the type vector that belongs to a particular category,(the Euclid distance as shown in equation (9) was used in this work).

$$D(x,y) = \sqrt{\sum_{j=1}^{nj}(x_j - y_j)^2} \tag{9}$$

$\sigma$: smoothing factor

The same combined neural node binds neurons in the pattern layer that belong to the same class. There is one neural node for each class in the plural layer, which combines the outputs of the pattern layer for each category and creates outputs that indicate probability for categories (obtaining and estimating the probability density function for each category), as mentioned earlier in equation (10).

$$O(x) = \frac{1}{nj}\sum_{j=1}^{nj} G(x) \tag{10}$$

Where

O (x): Output of the I plural node for the Ci class

nj: Number of samples in ci type layer

G (x): The product of the j-pattern node.

The output layer (resolution layer) chooses the maximum of one of these possibilities, as shown in equation (11)

$$\text{Target class}(x) = \text{maximum}(O(X)) \tag{11}$$

The de facto decision to appoint the signatories remains Gaussiansussians. The value of σ affects the results of identifying PNN works and should be carefully selected. More than one algorithm can be used to determine the homogeneity element of the range. In this work, minimum and maximum standard deviation values are used after calculating the average vector for each category to determine the σ range.

**The Proposed Method**

The proposed system includes the discovery and classification of infected persons using two deep learning methods called name probabilistic and conventional Neural networks. The proposed system consists of a data set that was collected and located in the Kaggle, and this data set needs pre-processing and then entering (CNN) once to extract the acquired characteristics and using (PNN) for the described classification process in Fig.10.

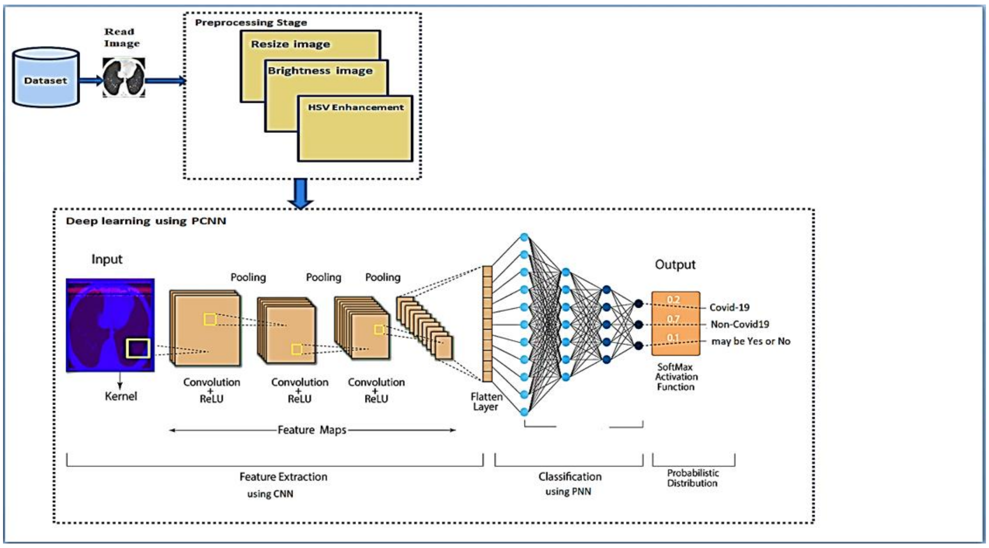



*Figure 10. The Proposed Method of COVID-19 detection using PCNN*

## Image Acquisition

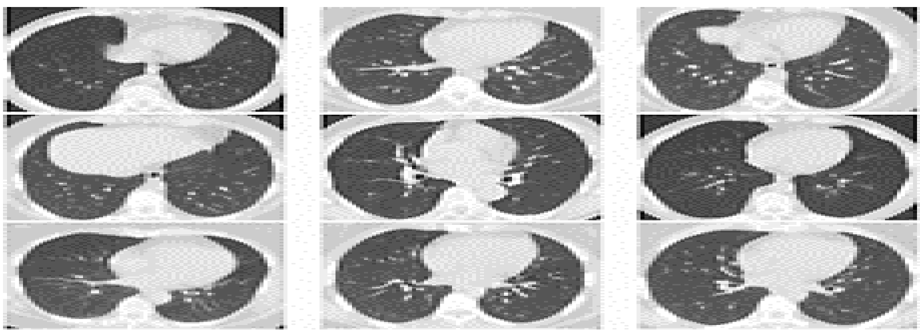
Image acquisition is always the first step in every vision system. After the image has been captured, a range of processing techniques may be utilized to execute a variety of image-related tasks. Processing is impossible without an image, which is why obtaining an image is always the initial step in a workflow sequence. Images may be captured using a variety of methods, including cameras and scanners. All of the attributes of the acquired image must be preserved. In this article, datasets were obtained for patients with and without Covid-19 illness, one of which was downloaded from the Kaggle Standard website.

## Standard Dataset

Images are collected from papers related to COVID19 from the bioRxiv, medRxiv, JAMA, and NEJM dubbed Lancet, (COVID-CT-Dataset: A CT Scan) Dataset on COVID-19.

Where the split-train-test is chosen randomly. There are 2500 training images, and 1800 test images, with nearly 50% of cases splitting non-Covid and Covid. The images are of different sizes, but all have been converted to .png for easy access. A good algorithm for processing these images needs to consider changing sizes and resembling the size you choose. The data set is left at its original size (without scaling uniformly) so that innovative precision-operated algorithms can operate without being constrained. This dataset is accessible and A sample of lung CT scan images from the dataset is shown in Fig.11.

**Non - COVID-19**



**COVID-19**

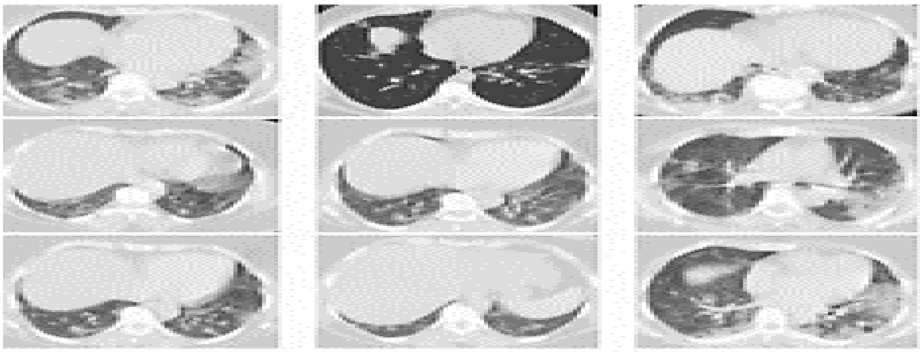


*Figure 11. sample of CT scan datasets*

***Preprocessing Stage***

Even though the CT images seldom include noise, the act of gathering the data set might obscure the image features, which is why the images need to be processed. Because the lighting had a role in how it turned out, we need to make some adjustments to it. In addition to that, the image has to be highlighted in the covid 19, which can be accomplished by converting the image and making use of HSV.

**Resize Image**

Due to the different sizes of the images for the data set utilized, we need to resize, the image is processed by resizing the image for covid-19 to 256 x 256.

**Brightness Image**

The proposed technique is utilized for preprocessing the method of enhancing the image brightness to show the details of the image well because sometimes the images may be taken in conditions that are not suitable in terms of light and noise, which can be taken into account. After all, any lack of lighting or confusion is noticeable when the images are taken and seen Although it was taken in a device and it is possible to display a simple model of those images, the data set that was trained from the global images, there was a problem in the process of capturing images because they did not take the appropriate lighting, so utilize the appropriate brightness transfer to improve the image as the brightness increases between 0 and 255.

**Hue saturation value (HSV) Enhancement**

Color and saturation are two factors that influence how the human eye perceives things. Both the original color image and the grayscale image to be colored affect the quality of the grayscale image colorization procedure. There are no objective criteria for determining the efficiency of a staining procedure. At most, a grayscale color image can be utilized as a source and recolored using colors from the original color image. The mean square error (MSE) and signal-to-noise ratio (PSNR) between the resulting image and the original image can then be calculated as a performance measure for the coloring method's quality, but the transfer of color attributes from the color source image to the target grayscale image using two different color spaces cannot be calculated HSV. We conclude that selecting a source image with a similar feature will target the image to acquire better staining results, and if done incorrectly, will result in poor staining. and Y is the luma component and CB and CR are the blue-difference and red-difference chroma components. (YCbCr) with various pixel window widths (2 x 2), (3 x 3) up to (10 x 10), we conclude that selecting a source image with a similar feature will target the image to acquire better staining results, and if done incorrectly, will result in poor staining. Depending on the characteristic of both the source color image and the target color image, as well as the target grayscale image, the pixel window size provides the best colorization result when utilizing an inappropriate source color image.

### Feature Extraction

Extracting properties from samples such as COVID19 CT graphs is the most important stage in diagnosing whether a person has Covid 19. In its basic version, the system uses CNN, where the bypass layer acts as the seventh feature extraction step.

### Input Layer

The width and height dimensions, in addition to the channel, each measure 28 pixels. The image's measurements are 28 by 28 by 1. As a result of digital data being composed of Covid-19 images, there are three different color channels. It is not essential to mix the data for this class since the train network does it on your behalf at the beginning of the training.

### Convolutional Layer

The landmark map is made by applying a weight slide mask to the source image and then striking the dot result. This is done on the feature extraction layer. The weight is generated at random, and then it is altered via the BN, Pooling, and RleU layers throughout numerous rounds until the weight that corresponds to this image's advantage is found; this weight is the optimum weight.

### Batch Normalization Layer (BN)

The training process is sped up by this layer, and the sensitivity of the network setup is eliminated as a result of the reduction in the number of channels. After the deviation of the mini-medium batch has been subtracted from and divided into the deviation of the standard mini-batch, each channel is activated. Next, the layer input is ultimately displaced and measured by the factor. The RleU layer and the bypass layer are both integrated with this layer.

### RleU Layer

The maximum assembly layer is used to remove features that are unnecessary or excessive, and it also returns an important milestone. This is accomplished by moving a mask with a known dimension on top of the map of features that were produced by the bypass layer that came before it. Since the maximum is empty, the result is the highest value that lies under this mask at every step. The maximum assembly layer is used to remove features that are unnecessary or excessive, and it also returns an important milestone. This is accomplished by moving a mask with a known dimension on top of the map of features that were produced by the bypass layer that came before it. Since the maximum is empty, the result is the highest value that lies under this mask at every step.

### Fully Connected Layer

It is referred to as a feature vector and it is comprised of the most vital information about inputs. During the training process, collects characteristics from all of the preceding bypass layers so that they may be utilized for classification later. To put it another way, a hidden layer has the capability of being taught to predict the likelihood of each class.

### Softmax Layer

Give the candidate class a high probability while decreasing the chance of other categories, which is generated by the Softmax layer, which generates a probability value between 0 and 1.

### Loss Function Laye

In each trading period, the loss function is invoked to compute the amount of loss, also known as error. In addition, the loss function is an essential part of the rear propagation weight update process since it displays the gap between the predicted output and the actual label.

### COVID -19 Classification utilizing PNN

PNN is a multi-layer supervised learning approach that employs four layers to accomplish its goals. This section will outline the structure of the four levels of feature sets utilized in this project.

**Input layer (distribution layer):** The initial feature sets in the input layer are made up of seven input neurons. The second feature set of the input layer consists of eight input neurons. As a consequence of the CNN method utilized to extract the features, these traits are considered highly strong.

**Pattern layer:** The first two sets of attributes make up the pattern layer, which has 260 pattern nodes. One modular ganglion per training sample per class; eight training patterns; each neuron node in the pattern layer computes the distance between unknown input vectors and training vectors represented by neurons and executes a Gaussian function;

**Abstraction layer:** The first and second attributes indicate that the layer has thirty nodes (the number of classes). Each collecting node receives the output from the pattern nodes associated with each class; which. Each class has its neuron, which combines the values of neurons belonging to that class in the pattern layer to generate an estimated probability density function (PDF) for that class.

**Output layer:** The first and second feature sets are made up of a single node.

The main purpose of the training phase is to determine the optimum smoothing coefficient value. In this study, the maximum and lowest values of the categories' standard deviations are utilized to establish the range.

### Experimental Results

Covid-19 will be identified at this stage by passing through various phases of the suggested approach, which will be explained below.

### Preprocessing Stage

At this stage, will be preprocessing the image as shown below :

### Brightness1. Image

Optimization processing was applied to a CT scan of a Covid-19 image with its graph after and before optimization pretreatment. In the process, select a range of parameters to control the desired brightness. The best parameter is 0.1 as shown in Fig.12.

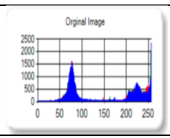
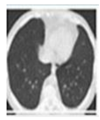
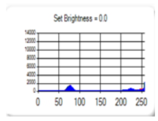
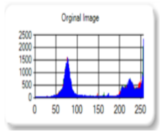
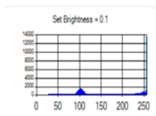
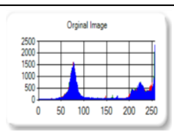
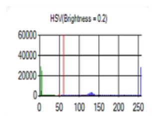
| Original Image | Histogram | parameter | Brightness Image | Histogram |
|---|---|---|---|---|
|  |  | **0.0** |  |  |
|  |  | **0.1** |  |  |
|  |  | **0.2** |  |  |

*Figure 12: Brightness CT Scan Covid-19 Image*

### HSV Enhancement

Results of CT conversion of a Covid-19 image model to an HSV model The H, S, and V parameters were not fixed but flexible in the order utilized to obtain an acceptable HSV image. The best parameter for HSV is H[0-60], S[61-180], V[181-255] as shown in Fig.13.

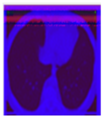
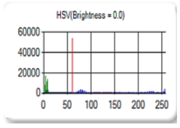
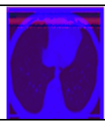
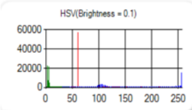
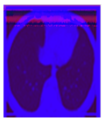
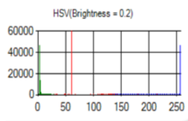
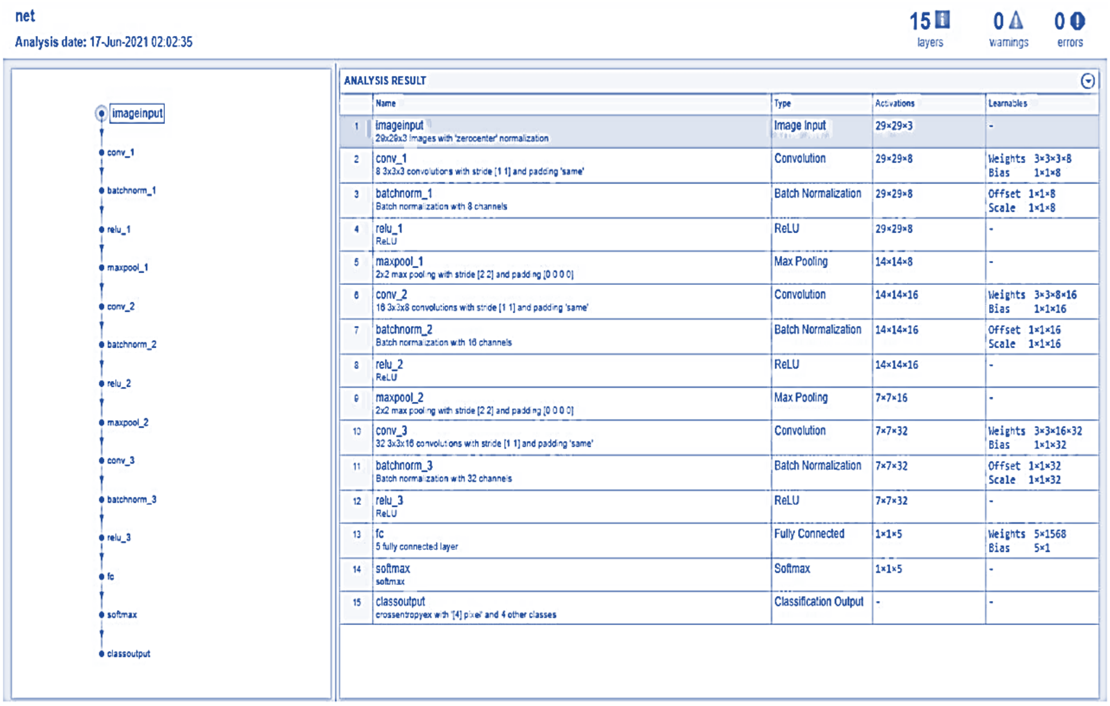
| Original image | Parameter | | | HSV | Histogram |
|---|---|---|---|---|---|
| | H [From, To] | S [From, To] | V [From, To] | | |
|  | [0 – 60] | [61-180] | [181-255] |  |   |
|  | [0-30] | [30-130] | [131-255] |  |   |
|  | [0-80] | [81-150] | [151-255] |  |   |

*Figure 13. HSV CT Scan Covid-19 Image*

### Feature Extraction using CNN

At this stage, the features will be extracted through CNN which includes eight important layers as explained below, where each layer will be utilized once as shown in the Tables below. and Fig.14.




| | Name | Type | Activations | Learnables |
|---|---|---|---|---|
| 1 | imageinput<br>29x29x3 images with 'zerocenter' normalization | Image Input | 29×29×3 | - |
| 2 | conv_1<br>8 3x3x3 convolutions with stride [1 1] and padding 'same' | Convolution | 29×29×8 | Weights 3×3×3×8<br>Bias 1×1×8 |
| 3 | batchnorm_1<br>Batch normalization with 8 channels | Batch Normalization | 29×29×8 | Offset 1×1×8<br>Scale 1×1×8 |
| 4 | relu_1<br>ReLU | ReLU | 29×29×8 | - |
| 5 | maxpool_1<br>2x2 max pooling with stride [2 2] and padding [0 0 0 0] | Max Pooling | 14×14×8 | - |
| 6 | conv_2<br>16 3x3x8 convolutions with stride [1 1] and padding 'same' | Convolution | 14×14×16 | Weights 3×3×8×16<br>Bias 1×1×16 |
| 7 | batchnorm_2<br>Batch normalization with 16 channels | Batch Normalization | 14×14×16 | Offset 1×1×16<br>Scale 1×1×16 |
| 8 | relu_2<br>ReLU | ReLU | 14×14×16 | - |
| 9 | maxpool_2<br>2x2 max pooling with stride [2 2] and padding [0 0 0 0] | Max Pooling | 7×7×16 | - |
| 10 | conv_3<br>32 3x3x16 convolutions with stride [1 1] and padding 'same' | Convolution | 7×7×32 | Weights 3×3×16×32<br>Bias 1×1×32 |
| 11 | batchnorm_3<br>Batch normalization with 32 channels | Batch Normalization | 7×7×32 | Offset 1×1×32<br>Scale 1×1×32 |
| 12 | relu_3<br>ReLU | ReLU | 7×7×32 | - |
| 13 | fc<br>5 fully connected layer | Fully Connected | 1×1×5 | Weights 5×1568<br>Bias 5×1 |
| 14 | softmax<br>softmax | Softmax | 1×1×5 | - |
| 15 | classoutput<br>crossentropyex with '[4] pixel' and 4 other classes | Classification Output | - | - |

*Figure 14. Feature Extraction for CNN*

### Convolution Layer

The image size (padding) is the same in this layer of eight filters with a filter size of 3 x 3. Because the filter action is one pixel, the default step value of one is suitable. This layer kernel, also known as the filter, makes use of it. The purpose of using numerous filters to extract distinct features is to find out if the original image contains particular qualities or patterns. The filter is tiny enough to scan the whole image and do the necessary computations between the filter and pixel values to extract features. The initially hidden layers are utilized to extract simple and virtual features such as edges in opposing directions. As we go further into the network's secret tiers, the intricacy of the attributes that must be identified and retrieved grows. As shown in Tab.1.

Table 1. Iteration Of Convolution Layer

| Iteration | Activation | Pyramid Level |
|---|---|---|
| 1 | 12.183 | 1 |
| 2 | 18.384 | 1 |
| 3 | 28.258 | 1 |
| 4 | 68.894 | 1 |
| 5 | 128.593 | 1 |
| 6 | 134.125 | 1 |
| 7 | 166.603 | 1 |
| 8 | 186.746 | 1 |
| 9 | 243.594 | 1 |
| 10 | 255.901 | 1 |

**Batch Normalization**

Standardize inputs to the layer for each small batch. This helps stabilize the learning process and reduces the number of training courses needed to develop deep networks. As shown in Tab.2.

Table 2. Iteration Of Batch Normalization

| Iteration | Activation | Pyramid Level |
|---|---|---|
| 1 | 0.001 | 1 |
| 2 | 0.456 | 1 |
| 3 | 0.764 | 1 |
| 4 | 0.922 | 1 |
| 5 | 0.987 | 1 |
| 6 | 1. 445 | 1 |
| 7 | 1. 095 | 1 |
| 8 | 1.134 | 1 |
| 9 | 1.156 | 1 |
| 10 | 1.178 | 1 |

**ReLU layer**

The Corrected Linear Activation Function Exit (ReLU) directly equals the input value. Otherwise, it’s zero. ReLU is a linear function with several definitions. Because it is faster in training and produces generally superior results, it has become the default activation function of many types of neural networks. As shown in Tab.3.

Table 3. Iteration of ReLU

| Iteration | Activation | Pyramid Level |
|---|---|---|
| 1 | 0.661 | 1 |
| 2 | 0.647 | 1 |
| 3 | 0.926 | 1 |
| 4 | 1.046 | 1 |
| 5 | 1.056 | 1 |

*ISSN: 2716-3865 (Print), 2721-1290 (Online)*

| | | |
|---|---|---|
| 6 | 1.067 | 1 |
| 7 | 1.088 | 1 |
| 8 | 1.119 | 1 |
| 9 | 1.119 | 1 |
| 10 | 1.11 9 | 1 |

**Max Pooling**

The purpose of the assembly layer is to make activation maps as small as possible, so more than one filter can be used. This not only reduces computing time but also prevents models from over-installing. To reduce the size of the huge arrays, use the following two functions: using the Max formula, look for as much value as possible in each window. Calculate the computational average of data in each window. Average: However, the first option is the most popular: maximum assembly. The main purpose is to maintain higher values within each window while scanning the activation map (or feature matrix) to reduce the size of the map. As shown in Tab.4.

Table 4. Iteration of Max-pooling

| Iteration | Activation | Pyramid Level |
|---|---|---|
| 1 | 0.846 | 1 |
| 2 | 1.370 | 1 |
| 3 | 1.551 | 1 |
| 4 | 1.548 | 1 |
| 5 | 1.693 | 1 |
| 6 | 1.667 | 1 |
| 7 | 1.891 | 1 |
| 8 | 1.843 | 1 |
| 9 | 1.876 | 1 |
| 10 | 0.846 | 1 |

**5. Fully Connected**

All neurons in the network layer are connected to all neurons in the previous layer. As shown in Tab.5.

Table 5. Iteration of Fully Connected

| Iteration | Activation | Pyramid Level |
|---|---|---|
| 1 | 0.35 | 1 |
| 2 | 2.25 | 1 |
| 3 | 3.69 | 1 |
| 4 | 5.08 | 1 |
| 5 | 6.18 | 1 |
| 6 | 7.32 | 1 |
| 7 | 8.92 | 1 |
| 8 | 9.29 | 1 |
| 9 | 9.80 | 1 |
| 10 | 10.13 | 1 |

**Softmax layer**

Use the softmax function to summarize a set of real K numbers, which take a set of real K values and add 1. Softmax converts positive, negative, zero, or many input values into 0 to 1 values, allowing them to be understood as probabilities. Softmax may deal with little or no input in one of two ways: low probability or high probability. No matter how small or negative the input is, the probability will always be between zero and one. As shown in Tab.6.

Table 6. Iteration Of Softmax

| Iteration | Activation | Pyramid Level |
|---|---|---|
| 1 | 0.20 | 1 |
| 2 | 0.76 | 1 |
| 3 | 0.85 | 1 |
| 4 | 0.96 | 1 |
| 5 | 0.98 | 1 |
| 6 | 1 | 1 |
| 7 | 1 | 1 |
| 8 | 1 | 1 |
| 9 | 1 | 1 |
| 10 | 1 | 1 |

**Classification using PNN**

After extracting features from CNN through a series of layers, the number of features is reduced because they vary from image to image by calculating an average and standard deviation, and features are then entered into PNN layers and the calculation of the characteristic is collected to classify whether a person has covid 19. As shown in Tab.7

Table 7. Classification of PNN

| **As shown in Tab.7** | | **As shown in Tab.7** | | **As shown in Tab.7** |
|---|---|---|---|---|
| **Feature** | **value** | **Class** | **Sum** | **Class2 = Positive COVID-19 Sum=**0.2850 |
| F(0) | 130.0567 | 1 | 0.2356 | |
| F(1) | 150.2563 | 2 | 0.2850 | |
| F(2) | 0.078463 | 3 | 0.2619 | |
| F(3) | 16.01987 | 4 | 0.2619 | |
| F(4) | 7.51229 | 5 | 0.0622 | |
| F(5) | 64.0659 | 6 | 0.0784 | |
| F(6) | 34.0563 | 7 | 0.0321 | |
| F(7) | 0.16202 | 8 | 0.0962 | |
| F(8) | 5.146365 | 9 | 0.0345 | |
| F(9) | 1.51979 | 10 | 0.0915 | |
| F(10) | 3.62023 | 11 | 0.1200 | |
| F(11) | 12.6641 | 12 | 0.0721 | |
| F(12) | -0.0040 | 13 | 0.0643 | |
| F(13) | 1.18152 | 14 | 0.0954 | |
| F(14) | 0.524876 | 15 | 0.0341 | |
| F(15) | 55.91278 | 16 | 0.3451 | |
| F(16) | 164.5747 | 17 | 0.1102 | |
| F(17) | 0.13420 | 18 | 0.2019 | |
| F(18) | 18.3671 | 19 | 0.1567 | |
| F(19) | 8.78339 | | | |

**Evaluation of CNN-PNN**

To assess the proposed model, it was necessary to calculate the accuracy of the proposed system, particularly after its application and the discovery of those infected and infected, with 70 percent of the system trained and the system tested at 30 percent of the standing data set, as in Figure 15. These ratios show that the system can detect them when calculating the accuracy of standard measurements to determine the value of performance: 100%. In the table (8) which shows the proposed CNN-PNN method, covid 19's discovery is 100% accurate. Factors that define the structure of the network (such as the number of hidden units, for example), as well as variables that govern the manner in which the network is trained, are referred to as hyperbolic parameters (e.g., the learning rate). The network's ability to swiftly change its parameters is directly proportional to the learning rate. The learning process moves more slowly but steadily forward when it has a low learning rate. Learning is accelerated by a greater learning rate, however, it is possible that the two rates will not converge. In most cases, the pace of learning that declines with time are favored.

Several eras are the number of times entire training data is displayed online during training. The accuracy of the mini-payment reported during training corresponds to the accuracy of the mini-batch assigned in the specified iteration. It's not an average run on iterations. During random gradient training with momentum (SGDM), the algorithm collects the full data set in small, unrelated batches. The iteration corresponds to calculating network gradients for each small batch. The era corresponds to the transition through each small batch available. The error is calculated for each example in the training data set, but it updates the form only after evaluating all examples of training.

When we train the accuracy of the model and its loss in the form of validation data it can vary with different situations. Usually, with each era increasing, the loss must decrease and accuracy must rise.

Many cases can be possible such as below:

val_loss begins to increase, and val_acc begins to decrease. This means that the model is cornering values and not learning.

val_loss begins to increase, and val_acc also increases. This may be a case of excess or varied probability values in cases where softmax is used in the output layer

val_loss begins to decrease, and val_acc begins to increase. This is also good because this means that the model that has been built learns and works well.

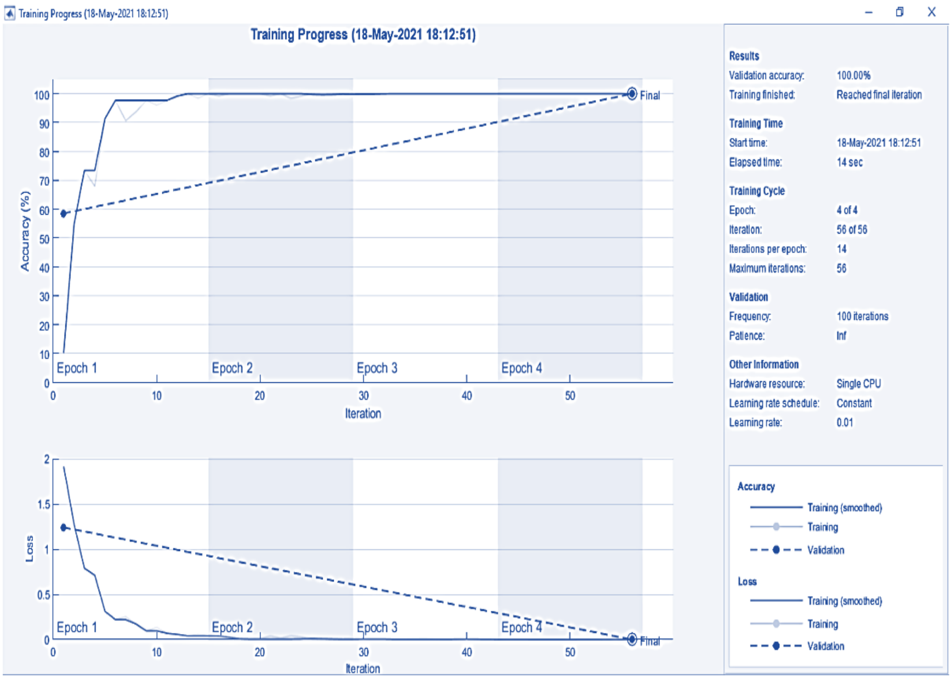



*Figure 15. Training and Testing Of Covid-19 For CNN-PNN*

Table 8. Accuracy of CNN-PNN

| # | Iteration | Time Elapsed (hh:mm:ss) | Mini-batch Accuracy | Validation Accuracy | Mini-batch Loss | Validation Loss | Base Learning Rate |
|---|---|---|---|---|---|---|---|
| 1 | 1 | 00:00:01 | 15 % | 60% | 4.0718 | 1.7120 | 100% |
| 2 | 10 | 00:00:03 | 98% | 98% | 0.0721 | 0.0609 | 100% |
| 3 | 20 | 00:00:07 | 100% | 99% | 0.0077 | 0.0183 | 100% |
| 4 | 30 | 00:00:09 | 100% | 100% | 0.0096 | 0.0084 | 100% |
| 5 | 40 | 00:00:13 | 100% | 100% | 0.0056 | 0.0065 | 100% |
| 7 | 50 | 00:00:14 | 100 % | 100 % | 0.0044 | 0.0049 | 100% |
| 8 | 60 | 00:00:15 | 100 % | 100% | 0.0082 | 0.0037 | 100% |
| 9 | 70 | 00:00:16 | 100% | 100.00% | 0.0018 | 0.0026 | 100% |

## Conclusion

We proposed an improved model using two deep learning methods of CNN-PNN, which has a certain improvement in the accuracy of image recognition issues, because the disease's spread is very dangerous, and because we need to know whether a person has Covid 19 or not, we proposed an improved model using two deep learning methods of CNN-PNN, which has a certain improvement in the accuracy of image recognition issues. However, it will add to the model's complexity. It will slow down training and raise the danger of overtraining. To lower the model's complexity, further organizing elements are required. We can easily extract a feature from the CNN layer in this work by categorizing the PNN as half of the individual is injured or not. There are a plethora of additional extraction techniques and formulations to investigate. However, because we overlook the fact that this data may be required to pre-process the image due to the possibility of noise during transmission, we can dismiss this concern. Before entering deep learning, several methods of image optimization were utilized, and the image was entered into the system once from the CNN layers to obtain the characteristics, and eight levels were utilized to give the best results for the characteristics, and the classification process was performed using PNN levels to give high accuracy by classifying. The system was 70% taught and 30% tested, with the findings indicating that the system's accuracy is 100% and that this proportion is not susceptible to mistakes.

## Acknowledgements

The Authors would like to thank Mustansiriyah University (https://uomustansiriyah.edu.iq) Baghdad -Iraq for its support in the present work.

## References

A. M. Alhamrouni. (2017). Iris recognition by using image preprocessing techniques," A MASTER'S THESIS In Computer Engineering Atilim University, 2017.

Ahmed, F. R. A., Ahmed, T. E., Saeed, R. A., Alhumyani, H., Abdel-Khalek, S., & Abu-Zinadah, H. (2021). Analysis and challenges of robust E-exams performance under COVID-19. *Results in Physics*, *23*, 103987.

Bashitialshaaer, R., Alhendawi, M., & Lassoued, Z. (2021). Obstacle comparisons to achieving distance learning and applying electronic exams during COVID-19 pandemic. *Symmetry*, *13*(1), 99.

Bharath, S., MuraliKrishnan, K., Padmanaban, M. A., & Jenefer, B. M. (2018). An Innovative Ear Based Recognition System for Automatic Attendance Monitoring. *International Journal of Latest Trends in Engineering and Technology e-ISSN*, 037-042.

Dai, W. C., Zhang, H. W., Yu, J., Xu, H. J., Chen, H., Luo, S. P., ... & Lin, F. (2020). CT imaging and differential diagnosis of COVID-19. *Canadian Association of Radiologists Journal*, *71*(2), 195-200

Elsalem, L., Al-Azzam, N., Jum'ah, A. A., & Obeidat, N. (2021). Remote E-exams during Covid-19 pandemic: A cross-sectional study of students' preferences and academic dishonesty in faculties of medical sciences. *Annals of Medicine and Surgery*, *62*, 326-333.

Emeršič, Ž., Štruc, V., & Peer, P. (2017). Ear recognition: More than a survey. *Neurocomputing*, *255*, 26-39.

Eskandari, M., & Toygar, Ö. (2015). Selection of optimized features and weights on face-iris fusion using distance images. *Computer Vision and Image Understanding*, *137*, 63-75.

Gore, D. B., Kawthekar, S. S., & Borole, V. Y. Ear Recognition Techniques for Biometrics Using Digital Image Processing: A Review.

Hamd, M. H., & Mohammed, M. Y. (2019). Multimodal biometric system based face-iris feature level fusion. *Int. J. Mod. Educ. Comput. Sci*, *11*(5), 1-9.

https://www.kaggle.com/c/covidct/data

Huo, G., Liu, Y., Zhu, X., Dong, H., & He, F. (2015). Face–iris multimodal biometric scheme based on feature level fusion. *Journal of Electronic Imaging*, *24*(6), 063020.

Jan, F. (2017). Non-circular iris contours localization in the visible wavelength eye images. *Computers & Electrical Engineering*, *62*, 166-177.

Kabir, W., Ahmad, M. O., & Swamy, M. N. S. (2019). A multi-biometric system based on feature and score level fusions. *IEEE Access*, *7*, 59437-59450.

Kandgaonkar, T. V., Mente, R. S., Shinde, A. R., & Raut, S. D. (2015). Ear biometrics: A survey on ear image databases and techniques for ear detection and recognition. *IBMRD's Journal of Management & Research*, *4*(1), 88-103.

Karasmanaki, E., & Tsantopoulos, G. (2021). Impacts of social distancing during COVID-19 pandemic on the daily life of forestry students. *Children and youth services review*, *120*, 105781.

Khlifi, Y., & El-Sabagh, H. A. (2017). A novel authentication scheme for E-assessments based on student behavior over e-learning platform. *International Journal of Emerging Technologies in Learning*, *12*(4).

Mohanapriya, K., & Babu, M. M. (2017). Ear recognition by feature extraction using force field transformation. *International Journal of Engineering and Computer Science*, *6*(3), 20742-20750.

Moser, K. M., Wei, T., & Brenner, D. (2021). Remote teaching during COVID-19: Implications from a national survey of language educators. *System*, *97*, 102431.

Nguyen, K., Fookes, C., Ross, A., & Sridharan, S. (2017). Iris recognition with off-the-shelf CNN features: A deep learning perspective. *IEEE Access*, *6*, 18848-18855.

Omara, I., Li, F., Zhang, H., & Zuo, W. (2016). A novel geometric feature extraction method for ear recognition. *Expert Systems with Applications*, *65*, 127-135.

Pokhrel, S., & Chhetri, R. (2021). A literature review on impact of COVID-19 pandemic on teaching and learning. *Higher Education for the Future*, *8*(1), 133-141.

Resmi, K. R., & Raju, G. (2019). Automatic 2D Ear Detection: A Survey. *International Journal of Scientific & Technology Research*, *8*(11), 3643-3647.

Sameh, S. M., Moustafa, H. E. D. S., & Zaki, F. W. (2021). Feature Level Fusion of Iris and Voice. *MEJ. Mansoura Engineering Journal*, *40*(5), 55-63.

Sharifi, O., & Eskandari, M. (2016). Optimal face-iris multimodal fusion scheme. *Symmetry*, *8*(6), 48.

Song, Y., Zheng, S., Li, L., Zhang, X., Zhang, X., Huang, Z., ... & Yang, Y. (2021). Deep learning enables accurate diagnosis of novel coronavirus (COVID-19) with CT images. *IEEE/ACM transactions on computational biology and bioinformatics*, *18*(6), 2775-2780.

Vijay, M., & Indumathi, G. (2021). Deep belief network-based hybrid model for multimodal biometric system for futuristic security applications. *Journal of Information Security and Applications*, *58*, 102707.