Journal of Engineering Sciences
Vol 13, Issue 06, June/2022
ISSN NO:0377-9254

DETECTION OF COVID 19 FROM CHEST X RAY SCANS USING CONVOLUTIONAL NEURAL NETWORKS ARCHITECTURE

[1]Pasupuleti Hemantth Pankajj Ram, [2]Shaik Abdul Khalid
[3]Mohammed Arshad Saeed, [4]Shaik Imran
[5]B. Aamir Khan
[6]Dr. M. Rudra Kumar
[1,2,3,4,5] STUDENT [6] PROFESSOR
G. PULLAIAH COLLEGE OF ENGINEERING AND TECHNOLOGY -KURNOOL

ABSTRACT

The detection of severe acute respiratory syndrome coronavirus 2 (SARS CoV-2), which is responsible for coronavirus disease 2019 (COVID-19), using chest X-ray images has life-saving importance for both patients and doctors. In addition, in countries that are unable to purchase laboratory kits for testing, this becomes even more vital. In this study, we aimed to present the use of deep learning for the high-accuracy detection of COVID-19 using chest X-ray images. Publicly available X-ray images (1583 healthy, 4292 pneumonia, and 225 confirmed COVID-19) were used in the experiments, which involved the training of deep learning and machine learning classifiers. Thirty-eight experiments were performed using convolutional neural networks, 10 experiments were performed using five machine learning models, and 14 experiments were performed using the state-of-the-art pre-trained networks for transfer learning. Images and statistical data were considered separately in the experiments to evaluate the performances of models, and eightfold cross-validation was used. A mean sensitivity of 93.84%, mean specificity of 99.18%, mean accuracy of 98.50%, and mean receiver operating characteristics–area under the curve scores of 96.51% are achieved. A convolutional neural network without pre-processing and with minimized layers is capable of detecting COVID-19 in a limited number of, and in imbalanced, chest X-ray images.

1.INTRODUCTION

At the end of 2019, humankind was faced with an epidemic—severe acute respiratory syndrome coronavirus 2 (SARS CoV-2)–related pneumonia, referred to as coronavirus disease 2019 (COVID-19)—that people did not expect to encounter in the current era of technology. While the COVID-19 outbreak started in Wuhan, China, the significant spread of the epidemic around the world has meant that the amount of equipment available to doctors fighting the disease is insufficient. At the time of writing (September 8, 2020), there have been more than 27,000,000 confirmed cases and more than 875,000 confirmed deaths worldwide.1 Considering the time required for diagnosis and the financial costs of the laboratory kits used for diagnosis, artificial intelligence (AI) and deep learning research and applications have been initiated to support doctors who aim to treat patients and fight the illness.2 Although rapid point-of-care COVID-19 tests are expected to be used in clinical settings at some point, for now, turnaround times for COVID-19 test results range from 3 to more than 48 hours, and probably not all countries will have access to those test kits that give results rapidly. According to a recently published multinational consensus

Vol 13, Issue 06, June/2022
ISSN NO:0377-9254

statement by the Fleischner Society, one of the main recommendations is to use chest radiography for patients with COVID-19 in a resource-constrained environment.

PROBLEM STATEMENT:

Doctors and patients can both benefit from the use of chest X-ray imaging to detect the SARS coronavirus 2 (SARS CoV-2) that causes coronavirus illness 2019 (COVID-19). This is especially important in nations where it is difficult or impossible to get laboratory equipment for testing. The goal of this study was to demonstrate the high-accuracy detection of COVID-19 in chest X-ray pictures using deep learning. The tests employed publicly available X-ray pictures to train deep learning and machine learning classifiers on (1583 healthy, 4292 pneumonia, and 225 verified COVID-19).

2. LITERATURE SURVEY

HybridSN: Exploring 3-D–2-D CNN Feature Hierarchy for Hyperspectral Image Classification:

Authors: Swalpa Kumar Roy, Student Member, IEEE, Gopal Krishna, Shiv Ram Dubey, Member, IEEE, and Bidyut B. Chaudhuri, Fellow, IEEE

Hyperspectral image (HSI) classification is widely used for the analysis of remotely sensed images. Hyperspectral imagery includes varying bands of images. Convolutional neural network (CNN) is one of the most frequently used deep learning-based methods for visual data processing. The use of CNN for HSI classification is also visible in recent works. These approaches are mostly based on 2-D CNN. On the other hand, the HSI classification performance is highly dependent on both spatial and spectral information. Very few methods have used the 3-D-CNN because of increased computational complexity. This letter proposes a hybrid spectral CNN (HybridSN) for HSI classification. In general, the HybridSN is a spectral–spatial 3-DCNN followed by spatial 2-D-CNN. The 3-D-CNN facilitates the joint spatial–spectral feature representation from a stack of spectral bands. The 2-D-CNN on top of the 3-D-CNN further learns more abstract-level spatial representation. Moreover, the use of hybrid CNNs reduces the complexity of the model compared to the use of 3-D-CNN alone. To test the performance of this hybrid approach, very rigorous HSI classification experiments are performed over Indian Pines, University of Pavia, and Salinas Scene remote sensing data sets. The results are compared with the stateof-the-art hand-crafted as well as end-to-end deep learning-based methods. A very satisfactory performance is obtained using the proposed HybridSN for HSI classification.

3. Existing system:

we study the end-to-end long-term trajectory prediction in dense traffic. The ‘‘long-term’’ here means that the model is capable to predict the trajectories of the entire process of nontrivial movements (movements except going straight) while keeping a low prediction error. The ‘‘dense traffic’’ here azmeans that every vehicle can influence trajectories of its surrounding vehicles, but the road is not fully blocked. In such scenarios, the motion of the vehicle is complicated due to the influence of surrounding vehicles.

Disadvantages:

- **Accuracy is less**

4. Proposed system:

we aimed to present the use of deep learning for the high-accuracy detection of COVID-19 using chest X-ray images. Publicly available X-ray images (1583 healthy, 4292 pneumonia, and 225 confirmed COVID-19) were used in the experiments, which involved the training of deep learning and machine learning classifiers. Thirty-eight experiments were performed using convolutional neural networks, 10 experiments were performed using five machine learning models, and 14 experiments were performed using the state-of-the-art pre-trained networks for transfer learning. Images and statistical data were considered separately in the experiments to evaluate the performances of models, and eightfold cross-validation was used. A mean sensitivity of 93.84%, mean specificity of 99.18%, mean accuracy of 98.50%, and mean receiver operating characteristics–area under the curve scores of 96.51% are achieved. A convolutional neural network without pre-processing and with minimized layers is capable of detecting COVID-19 in a limited number of, and in imbalanced, chest X-ray images.

Advantages:

- **Accuracy is high**

Methodology:

A total of 225 COVID-19 chest X-ray images were obtained from Cohen;18 they can be accessed from github.19 The average age for the COVID-19 group was 58.8±14.9 years, and it comprised 131 male patients and 64 female patients. Note that some patients' information is missing; this is because the dataset used in this study does not have accompanying complete metadata, because this is the very first publicly available COVID-19 X-ray image collection, and it was created in a limited time. In addition, 1583 normal and 4292 pneumonia chest X-ray images were obtained from Kermany et al.20 All images were in different dimensions, so they were resized to 640 × 480.

Design of Experiments:

Several categorized experiments were performed to evaluate the efficiency of the ConvNet on the considered image database and to compare ConvNet with other models using the basic statistical characteristics of the images, which can provide effective information for classification. Experiments were divided into three categories: ConvNetexperiments, statistical measurement experiments, and transfer learning experiments.

4.4IMPLEMENTATION:

MODULES:

- Upload Covid-19 Chest X-ray Dataset
- Preprocess Dataset
- Model Generation
- Build CNN Covid-19 Model
- Upload Test Data & Predict Disease
- Accuracy Comparison Graph

4.4 ALGORITHMS:

RESNET

A residual neural network (ResNet) is an artificial neural network (ANN) of a kind that builds on constructs known from pyramidal cells in the cerebral cortex. Residual neural networks do this by utilizing skip connections, or shortcuts to jump over some layers. Typical ResNet models are implemented with double- or triple- layer skips that contain nonlinearities (ReLU) and batch normalization in between.[1] An additional weight matrix may be used to learn the skip weights; these models are known as HighwayNets.[2] Models with several parallel skips are referred to as DenseNets.[3] In the context of residual neural networks, a non-residual network may be described as a plain network. A reconstruction of a pyramidal cell. Soma and dendrites are labeled in red, axon arbor in blue. (1) Soma, (2) Basal

Journal of Engineering Sciences
Vol 13, Issue 06, June/2022
ISSN NO:0377-9254

dendrite, (3) Apical dendrite, (4) Axon, (5) Collateral axon. There are two main reasons to add skip connections: to avoid the problem of vanishing gradients, or to mitigate the Degradation (accuracy saturation) problem; where adding more layers to a suitably deep model leads to higher training error.[1] During training, the weights adapt to mute the upstream layer [clarification needed], and amplify the previously-skipped layer. In the simplest case, only the weights for the adjacent layer's connection are adapted, with no explicit weights for the upstream layer. This works best when a single nonlinear layer is stepped over, or when the intermediate layers are all linear. If not, then an explicit weight matrix should be learned for the skipped connection (a HighwayNet should be used). Skipping effectively simplifies the network, using fewer layers in the initial training stages [clarification needed]. This speeds learning by reducing the impact of vanishing gradients, as there are fewer layers to propagate through. The network then gradually restores the skipped layers as it learns the feature space.

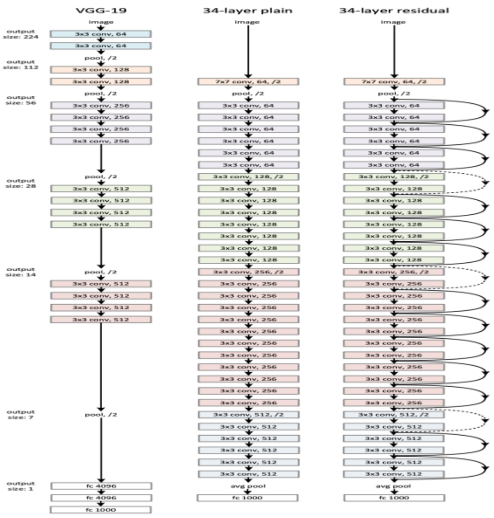
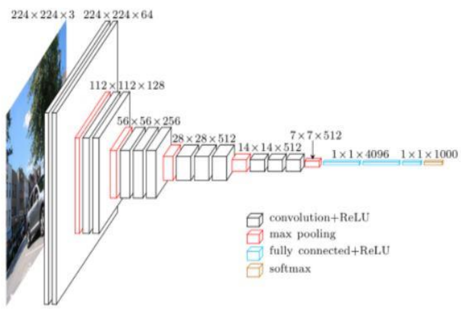


Figure 3. Example network architectures for ImageNet. **Left**: the VGG-19 model [41] (19.6 billion FLOPs) as a reference. **Middle**: a plain network with 34 parameter layers (3.6 billion FLOPs). **Right**: a residual network with 34 parameter layers (3.6 billion FLOPs). The dotted shortcuts increase dimensions. **Table 1** shows more details and other variants.

VGG16

VGG16 is a convolution neural net (CNN) architecture which was used to win ILSVR(Imagenet) competition in 2014. It is considered to be one of the excellent vision model architectures till date. Most unique thing about VGG16 is that instead of having a large number of hyper-parameters they focused on having convolution layers of 3x3 filter with a stride 1 and always used same padding and maxpool layer of 2x2 filter of stride 2. It follows this arrangement of convolution and max pool layers consistently throughout the whole architecture. In the end it has 2 FC (fully connected layers) followed by a softmax for output. The 16 in VGG16 refers to it has 16 layers that have weights. This network is a pretty large network and it has about 138 million (approx) parameters.I am going to implement full VGG16 from scratch in Keras.



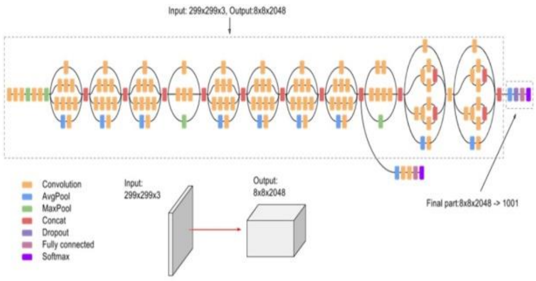
Architecture of VGG16

INCEPTION

Inception v3 is a widely-used image recognition model that has been shown to attain greater than 78.1% accuracy on the ImageNet dataset. The model is the culmination of many ideas developed by multiple researchers over the years. It is based on the original paper: "Rethinking the Inception Architecture for Computer Vision" by Szegedy, et. al. The model itself is made up of symmetric and asymmetric building blocks, including convolutions, average pooling, max pooling,

Vol 13, Issue 06, June/2022
ISSN NO:0377-9254

concatinations, dropouts, and fully connected layers. Batch normalization is used extensively throughout the model and applied to activation inputs. Loss is computed using Softmax.



DenseNet 121

DenseNet (Dense Convolutional Network) is an architecture that focuses on making the deep learning networks go even deeper, but at the same time making them more efficient to train, by using shorter connections between the layers. DenseNet is a convolutional neural network where each layer is connected to all other layers that are deeper in the network, that is, the first layer is connected to the 2nd, 3rd, 4th and so on, the second layer is connected to the 3rd, 4th, 5th and so on. This is done to enable maximum information flow between the layers of the network. To preserve the feed-forward nature, each layer obtains inputs from all the previous layers and passes on its own feature maps to all the layers which will come after it. Unlike Resnets it does not combine features through summation but combines the features by concatenating them. So the 'ith' layer has 'i' inputs and consists of feature maps of all its preceding convolutional blocks. Its own feature maps are passed on to all the next 'I-i' layers. This introduces '(I(I+1))/2' connections in the network, rather than just 'I' connections as in traditional deep learning architectures. It hence requires fewer parameters than traditional convolutional neural networks, as there is no need to learn unimportant feature maps. DenseNet consists of two important blocks other than the basic convolutional and pooling layers. they are the Dense Blocks and the Transition layers.

Layers	Output Size	DenseNet-121	DenseNet-169	DenseNet-201	DenseNet-264
Convolution	112×112	7×7 conv, stride 2			
Pooling	56×56	3×3 max pool, stride 2			
Dense Block (1)	56×56	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 6$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 6$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 6$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 6$
Transition Layer (1)	56×56	1×1 conv			
	28×28	2×2 average pool, stride 2			
Dense Block (2)	28×28	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 12$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 12$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 12$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 12$
Transition Layer (2)	28×28	1×1 conv			
	14×14	2×2 average pool, stride 2			
Dense Block (3)	14×14	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 24$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 32$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 48$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 64$
Transition Layer (3)	14×14	1×1 conv			
	7×7	2×2 average pool, stride 2			
Dense Block (4)	7×7	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 16$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 32$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 32$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 48$
Classification Layer	1×1	7×7 global average pool			
		1000D fully-connected, softmax			

DenseNet architectures for ImageNet. The growth rate for all the networks is $k = 32$. Note that each "conv" layer shown in the table corresponds the sequence BN-ReLU-Conv.

CNN:

To build such automated opinion detection author is suggesting to build CNN model which can work like human brains. This CNN model can be generated for any services and we can make it to work like automated decision making without any human interactions. To suggest this technique author already describing concept to implement multiple models in which one model can detect or recognize human hand written digits and second model can detect sentiment from text sentences which can be given by human about government schemes. In our extension model we added another model which can detect sentiment from person face image. Person face expressions can describe sentiments better than words or sentences. So, our extension work can predict sentiments from person face images.

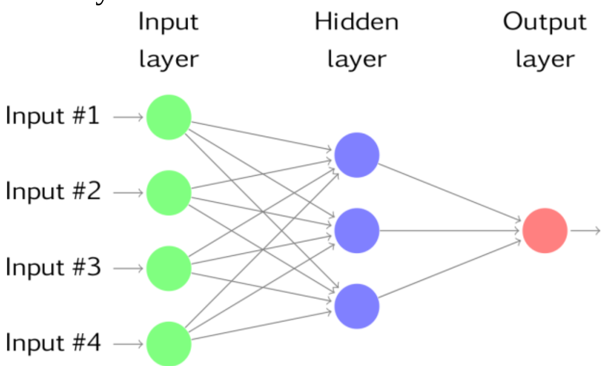
To demonstrate how to build a convolutional neural network-based image classifier, we shall build a 6-layer neural network that will identify and separate one image from other. This network that we shall build is a very small network that we can run on a CPU as well. Traditional neural networks that are very good at doing image classification have many more parameters and take a lot of time if trained on normal CPU. However, our objective is

Journal of Engineering Sciences
Vol 13, Issue 06, June/2022
ISSN NO:0377-9254

to show how to build a real-world convolutional neural network using TENSORFLOW.

Neural Networks are essentially mathematical models to solve an optimization problem. They are made of neurons, the basic computation unit of neural networks. A neuron takes an input (say x), do some computation on it (say: multiply it with a variable w and adds another variable b) to produce a value (say; z= wx+b). This value is passed to a non-linear function called activation function (f) to produce the final output(activation) of a neuron. There are many kinds of activation functions. One of the popular activation function is Sigmoid. The neuron which uses sigmoid function as an activation function will be called sigmoid neuron. Depending on the activation functions, neurons are named and there are many kinds of them like RELU, TanH.

If you stack neurons in a single line, it's called a layer; which is the next building block of neural networks. See below image with layers



To predict image class multiple layers operate on each other to get best match layer and this process continues till no more improvement left.

5.SOFTWARE ENVIRONMENT

Deep Neural Networks

Deep learning (also known as deep structured learning) is part of a broader family of machine learning methods based on artificial neural networks with representation learning. Learning can be supervised, semi-supervised or unsupervised.

Deep-learning architectures such as deep neural networks, deep belief networks, deep reinforcement learning, recurrent neural networks and convolutional neural networks have been applied to fields including computer vision, speech recognition, natural language processing, machine translation, bioinformatics, drug design, medical image analysis, material inspection and board game programs, where they have produced results comparable to and in some cases surpassing human expert performance.

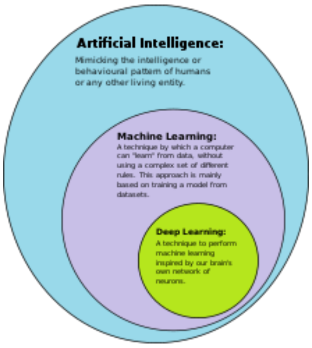
Artificial neural networks (ANNs) were inspired by information processing and distributed communication nodes in biological systems. ANNs have various differences from biological brains. Specifically, artificial neural networks tend to be static and symbolic, while the biological brain of most living organisms is dynamic (plastic) and analogue.

The adjective "deep" in deep learning refers to the use of multiple layers in the network. Early work showed that a linear perceptron cannot be a universal classifier, but that a network with a nonpolynomial activation function with one hidden layer of unbounded width can. Deep learning is a modern variation which is concerned with an unbounded number of layers of bounded size, which permits practical application and optimized implementation, while retaining theoretical universality under mild conditions. In deep learning the layers are also permitted to be heterogeneous and to deviate widely from biologically

informed connectionist models, for the sake of efficiency, trainability and

Journal of Engineering Sciences
Vol 13, Issue 06, June/2022
ISSN NO:0377-9254

understandability, whence the "structured" part.



Deep learning creates many layers of neurons, attempting to learn structured representation of big data, layer by layer.

Architecture of the network:

Network models

• Deep neural networks are mathematical models of intelligence designed to mimic human brains.

• Network models define a set of network layers and how they interact.

• Questions to answer while designing a network models include: –

Which layer type to use? – How many neurons to use in each layer? – How are layers arranged? – And more

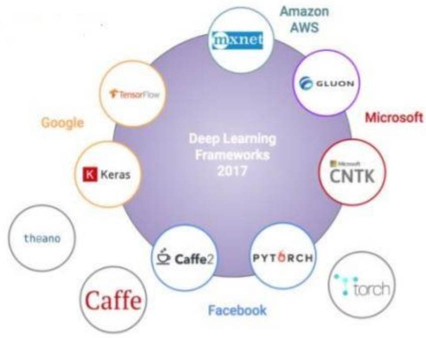
• There are many standard CNN models available today which work great for many standard problems. Examples being AlexNet, GoogleNet, Inception-ResNet, VGG, etc

Deep learning frameworks

• Building a deep learning solution is a big challenge because of its complexity.

• Frameworks are tools to ease the building of deep learning solutions.

• Frameworks offer a higher level of abstraction and simplify potentially difficult programming tasks.



Popular Frameworks:

• TensorFlow: – Developed by Google – The most used deep learning framework – Based on Github stars and forks and Stack Overflow activity

• Caffe: – Developed by Berkeley Vision and Learning Center (BVLC) – Popular for CNN modeling (imaging/computer vision applications) and its Model Zoo (a selection of pre-trained networks) Next to all for deep learning today is Keras. Keras is a high-level deep learning API, written in Python.

Deep learning development flow

1. Selection of a framework for development

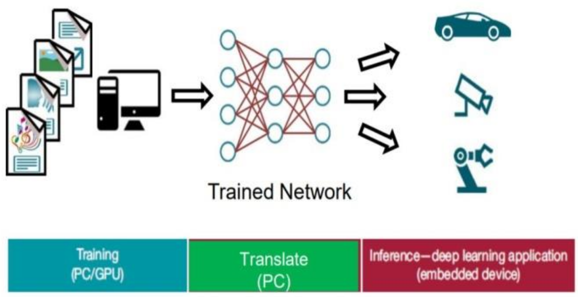
2. Selecting labeled data set of classes to train the network upon

3. Designing initial network model

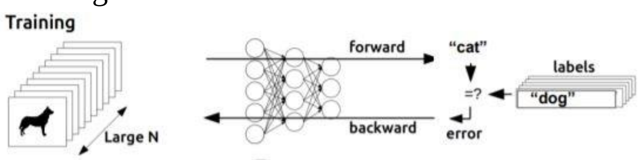
4. Training the network

5. Saving the parameters and architecture in a binary file

6. Inference



Training data:



Journal of Engineering Sciences
Vol 13, Issue 06, June/2022
ISSN NO:0377-9254

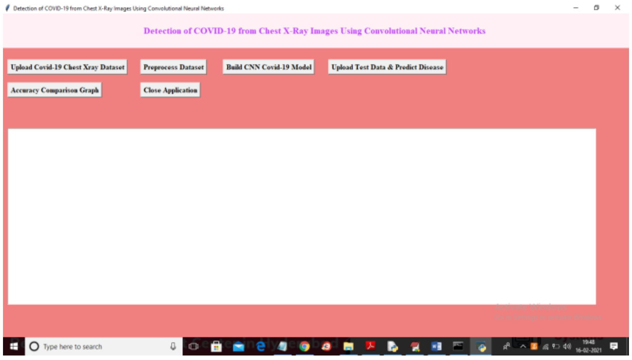
CNN:

Convolution leverages three important ideas that can help improve a machine learning system: sparse interactions, parameter sharing and equivalent representations. Moreover, convolution provides a means for working with inputs of variable size. Traditional neural network layers use matrix multiplication by a matrix of parameters with a separate parameter describing the interaction between each input unit and each output unit. This means every output unit interacts with every input unit. Convolutional networks, however, typically have sparse interactions This is accomplished by making the kernel smaller than the input. For example, when processing an image, the input image might have thousands or millions of pixels, but we can detect small, meaningful features such as edges with kernels that occupy only tens or hundreds of pixels.

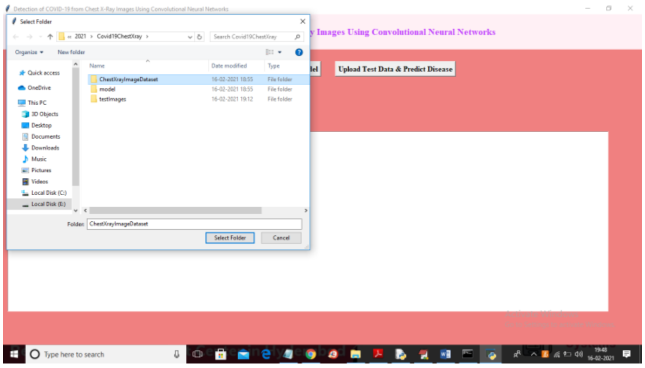
Store fewer parameters, which both reduces the memory requirements of the model and improves its statistical efficiency. It also means that computing the output requires fewer operations. These improvements in efficiency are usually quite large For many practical applications, it is possible to obtain good performance on the machine learning task while keeping several orders of magnitude less data. This allows the network to efficiently describe complicated interactions between many variables by constructing such interactions from simple building blocks that each describe only sparse interactions.

SCREENSHOTS

To run project double, click on 'run.bat' file to get below screen



In above screen click on 'Upload Covid-19 Chest X-ray Dataset' button and upload dataset

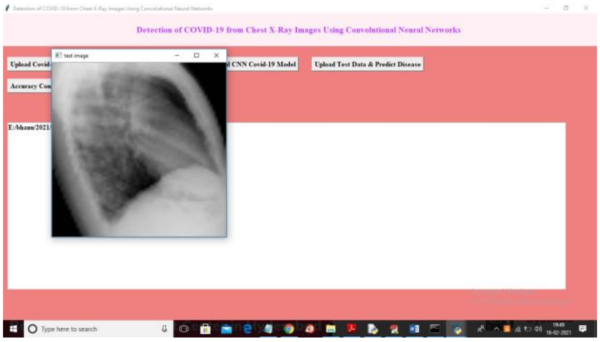


In above screen selecting and uploading 'ChestXrayImageDataset' folder which contains dataset images and then click on 'Select Folder' button to get below screen

In above screen dataset loaded and now click on 'Preprocess Dataset' button to read all images and then convert all images into equal size and then normalize all pixels of images to have better prediction result

Journal of Engineering Sciences

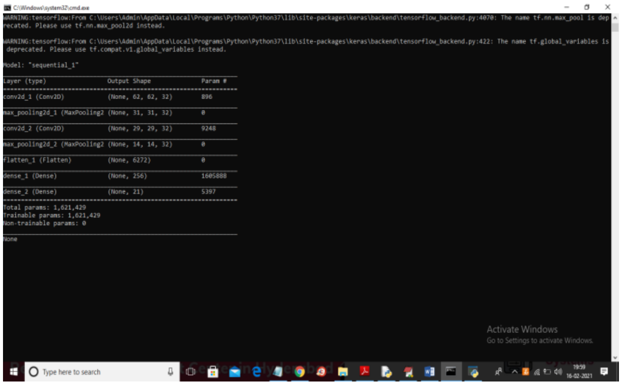
Vol 13, Issue 06, June/2022
ISSN NO:0377-9254



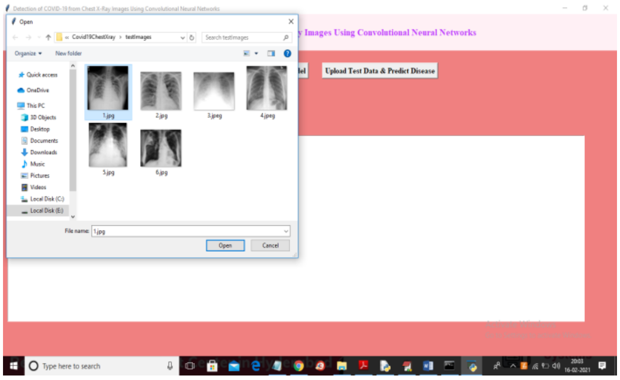
In above screen dataset processed and to test whether application reading all images properly so I am displaying one loaded sample image and now close above image to get below screen

In above screen application found total 820 images and now images are ready and now click on 'Build CNN Covid-19 Model' button to generate CNN model on loaded dataset and to get below screen

In above screen CNN model generated and its prediction accuracy is 89% and we can see below black console to see CNN layer details or its summary



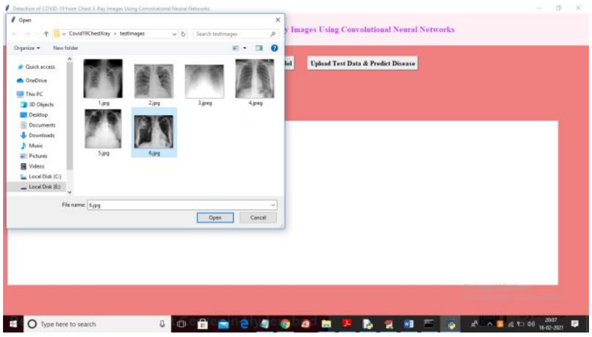
In above console we can see images are filtered at different layer with different image sizes where at first layer 62 X 62 image size was used and in second layer 31 X 31 and goes on. Now CNN model is ready and now click on 'Upload Test Data & Predict Disease' button to upload new test image and then application will predict disease from that image



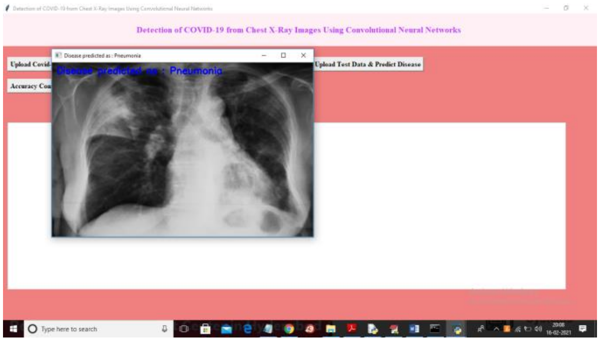
In above screen selecting and uploading '1.jpg' and then click on 'Open' button to load image and to get below prediction result

In above screen in blue colour text printing detected disease in uploaded image and now upload another image and test

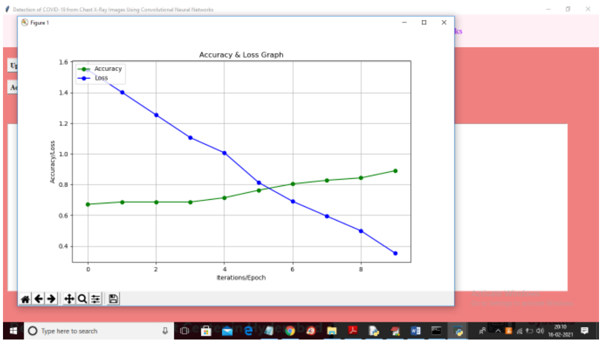
Journal of Engineering Sciences
Vol 13, Issue 06, June/2022
ISSN NO:0377-9254



In above screen selecting and uploading '6.jpg' and then click on 'Open' button to get below prediction result



In above screen disease predicted as 'Pneumonia' and similarly you can upload other images and get prediction result. Now click on 'Accuracy Comparison Graph' button to get below graph



In above graph green line represents accuracy and blue line represents LOSS. In above graph x-axis represents epoch/iteration and y-axis represents accuracy and loss values and to build CNN i took 10 iterations and we can see at each increasing iteration Accuracy get increase and LOSS get decrease.

8.CONCLUSION

Detection of COVID-19 from chest X-ray images is of vital importance for both doctors and patients to decrease the diagnostic time and reduce financial costs. Artificial intelligence and deep learning are capable of recognizing images for the tasks taught. In this study, several experiments were performed for the high-accuracy detection of COVID-19 in chest X-ray images using ConvNets. Various groups—COVID-19/Normal, COVID-19/Pneumonia, and COVID-19/Pneumonia/Normal— were considered for the classification. Different image dimensions, different network architectures, state-of-the-art pre-trained networks, and machine learning models were implemented and evaluated using images and statistical data. When the number of images in the database and the detection time of COVID-19 (average testing time = 0.03 s/image) are considered using ConvNets, it can be suggested that the considered architectures reduce the computational cost with high performance. The results showed that the convolutional neural network with minimized convolutional and fully connected layers is capable of detecting COVID-19 images within the two-class, COVID-19/Normal and COVID-19/Pneumonia classifications, with mean ROC AUC scores of 96.51 and 96.33%, respectively. In addition, the second proposed architecture, which had the second-lightest architecture, is capable of detecting COVID-19 in three-class, COVID-19/Pneumonia/Normal images, with a macro-averaged F1score of 94.10%. Therefore, the use of AI-based automated high-accuracy technologies may provide valuable assistance to doctors in diagnosing COVID-19. Further studies, based on the results obtained in this study, would provide more information about the use of CNN architectures with COVID-19 chest X-ray images and improve on the results of this study.

9. REFERENCES

1. World Health Organization. WHO Coronavirus Disease (COVID-19) Dashboard. https://covid19.who.int.

2. Apostolopoulos, I. D.; Mpesiana, T. Covid-19: Automatic Detection from X-Ray Images Utilizing Transfer Learning with Convolutional Neural Networks. Phys. Eng. Sci. Med. 2020, 43, 635–640.

3. Rubin, G. D.; Ryerson, C. J.; Haramati, L. B.; et al. The Role of Chest Imaging in Patient Management during the COVID19 Pandemic: A Multinational Consensus Statement from the Fleischner Society [published online ahead of print, 2020 Apr 7]. Chest. 2020, S0012-3692, 30673–30675. doi:10.1016/j. chest.2020.04.003.

4. Ozsahin, I.; Sekeroglu, B.; Mok, G. S. P. The Use of Back Propagation Neural Networks and 18F-Florbetapir PET for Early Detection of Alzheimer's Disease Using Alzheimer's Disease Neuroimaging Initiative Database. PLoS One. 2019, 14, e0226577.

5. Dai, S.; Li, L.; Li, Z. Modeling Vehicle Interactions via Modified LSTM Models for Trajectory Prediction. IEEE Access. 2019, 7, 38287–38296.

6. Yılmaz, N.; Sekeroglu, B. Student Performance Classification Using Artificial Intelligence Techniques. In: 10th International Conference on Theory and Application of Soft Computing, Computing with Words and Perceptions (ICSCCW) [2019]. Adv. Intel Sys. Comm. 2019, 1095, 596–603.

7. Meng, C.; Zhao, X. Webcam-Based Eye Movement Analysis Using CNN. IEEE Access. 2017, 5, 19581–19587. doi:10.1109/ACCESS.2017.2754299.

8. Deng, X.; Zhang, Y.; Yang, S.; et al. Joint Hand Detection and Rotation Estimation Using CNN. IEEE Trans. Image Proc. 2018, 27, 1888–1900, doi:10.1109/TIP.2017.2779600.

9. LeCun, Y.; Haffner, P.; Bottou, L.; et al. Object Recognition with Gradient-Based Learning: Shape, Contour and Grouping in Computer Vision. Lect. Notes Comput. Sci. 1999, 1681, 319–345.

10. Roy, S. K.; Krishna, G.; Dubey, S. R.; et al. HybridSN: Exploring 3-D–2-D CNN Feature Hierarchy for Hyperspectral Image Classification. IEEE Geosci. Remote. Sens. Lett. 2020, 17, 277–281.