iJRASET
International Journal For Research in Applied Science and Engineering Technology



INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 10 **Issue:** IV **Month of publication:** April 2022

DOI: https://doi.org/10.22214/ijraset.2022.41674

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 10 Issue IV Apr 2022- Available at www.ijraset.com

Pneumonia and COVID-19 Detection using Convolutional Neural Network

Prof. M. R. Gorbal[1], Soham Vilas Sawant[2], Grishma Jagadish Tarmale[3], Sanket Mukund Nayakwadi[4]
[1]*Teacher,* [2, 3, 4]*B.E. Students, Department of Information Technology, Shivajirao S. Jondhale College of Engineering, University of Mumbai*

Abstract: ***The exponential increase in COVID-19 patients is overwhelming healthcare systems across the world. With limited testing kits, it is impossible for every patient with respiratory illness to be tested using conventional techniques (RT-PCR). The tests also have long turn-around time, and limited sensitivity. Detecting possible COVID-19 infections on Chest X-Ray may help quarantine high risk patients while test results are awaited. X-Ray machines are already available in most healthcare systems, and with most modern X-Ray systems already digitized, there is no transportation time involved for the samples either. In this work we propose the use of chest X-Ray to prioritize the selection of patients for further RT-PCR testing. This may be useful in an inpatient setting where the present systems are struggling to decide whether to keep the patient in the ward along with other patients or isolate them in COVID-19 areas. It would also help in identifying patients with high likelihood of COVID with a false negative RT-PCR who would need repeat testing. Further, we propose the use of modern AI techniques to detect the COVID-19 patients using X-Ray images in an automated manner, particularly in settings where radiologists are not available, and help make the proposed testing technology scalable. We present CovidAID: COVID-19 AI Detector, a novel deep neural network based model to triage patients for appropriate testing. On the publicly available covid-chestxray-dataset, our model gives 90% accuracy for the COVID-19 infection.***

Keywords: Pneumonia, COVID-19, Chest X-Ray, Convolutional Neural Network, VGG-16

I. INTRODUCTION

The sudden spike in the number of patients with COVID-19, a new respiratory virus, has put unprecedented load over healthcare systems across the world. In many countries, the healthcare systems have already been overwhelmed. There are limited kits for diagnosis, limited hospital beds for admission of such patients, limited personal protective equipment (PPE) for healthcare personnel and limited ventilators. It is thus important to differentiate which patients with severe acute respiratory illness (SARI) could have COVID-19 infection in order to efficiently utilize the limited resources. In this work we propose the use of chest X-Ray to detect COVID-19 infection in the patients exhibiting symptoms of SARI. Using our tool one can classify a given X-Ray in one of the four classes: normal, bacterial pneumonia, viral pneumonia, and covid pneumonia. The use of X-Ray has several advantages over conventional diagnostic tests:

1) X-Ray imaging is much more widespread and cost effective than the conventional diagnostic tests.
2) Transfer of digital X-Ray images does not require any transportation from point of acquisition to the point of analysis, thus making the diagnostic process extremely quick.
3) Unlike CT Scans, portable X-Ray machines also enable testing within an isolation ward itself, hence reducing the requirement of additional Personal Protective Equipment (PPE), an extremely scarce and valuable resource in this scenario. It also reduces the risk of hospital acquired infection for the patients.

The main contribution of this work is in proposing a novel deep neural network based model for highly accurate detection of COVID-19 infection from the chest X-Ray images of the patients. Radiographs in the current setting are in most cases interpreted by non-radiologists. Further, given the novelty of the virus, many of the radiologists themselves may not be familiar with all the nuances of the infection, and may be lacking in the adequate expertise to make highly accurate diagnosis. Therefore this automated tool can serve as a guide for those in the forefront of this analysis.

II. LITERATURE SURVEY

In the context of the COVID-19 pandemic, extensive research has been conducted to develop automated image-based COVID-19 detection and diagnostic systems. We hereafter review the proposed approaches for reliable detection systems based on chest X-ray and CT-scan imaging modalities. These techniques follow either one of the two main paradigms. On one hand, new deep network architectures have been developed and tailored specifically for detecting and recognizing COVID-19.

©IJRASET: All Rights are Reserved

COVID-Net represents one of the earliest convolutional networks designed for detecting COVID-19 cases automatically from X-ray images. The performance of the network showed an acceptable accuracy of 83.5% and a high sensitivity of 100% for COVID-19 cases. Hasan et al proposed a CNN-based network named the Coronavirus Recognition Network (CVR-Net) to automatically detect COVID-19 cases from radiography images. The network was trained and evaluated on datasets with X-ray and CT images. The results showed varying accuracy scores according to the number of classes in the underlying X-ray image dataset, and an average accuracy of 78% for the CT image dataset.

Zhou et al highlighted the importance of deep learning techniques and chest CT images for differentiating between COVID-19 and influenza pneumonias. The study was conducted on CT images for confirmed COVID-19 patients from different hospitals in China. Their study proved the potential of accurate COVID-19 diagnosis from CT images and the effectiveness of their proposed classification scheme to differentiate between the two pneumonia types. Deep Pneumonia was developed to identify COVID-19 cases (88 patients), bacterial pneumonia (100 patients), and healthy cases (86 subjects) based on CT images. The model achieved an accuracy of 86.5% for differentiating between bacterial and viral (COVID-19) pneumonia.

Very few studies employed handcrafted feature extraction methods and conventional classifiers. In texture, features were extracted from X-ray images using popular texture descriptors. The features were combined with those extracted from a pretrained InceptionV3 using different fusion strategies.

Then, various classifiers were used to differentiate between normal X-rays and different types of pneumonia. The best classification scheme achieved an F1-score of 83%.

Pneumonia detection in Chest X-Rays: Various deep learning based approaches have been developed to identify different thoracic diseases, including pneumonia. We choose CheXNetto build upon, which could detect pneumonia from chest X-Rays at a level exceeding practicing radiologists. CheXNet is trained on ChestX-ray14 (the largest publicly available chest X-ray dataset), gives better performance than previous approaches, and has a simpler architecture than later approaches. CheXNet is a 121-layer DenseNet based model trained on the ChestXray14 dataset comprising of 112,120 frontal-view chest X-Ray images. The model is trained to classify X-Ray images into 14 different thoracic disease classes, including pneumonia. Given the visual similarity of the input samples, we found this to be the closest pre-trained backbone to develop a model for identifying COVID-19 pneumonia.

COVID-19 detection in Chest X-Rays: Since the recent sudden surge of COVID-19 infections across the world, many alternative screening approaches have been developed to identify suspected cases of COVID-19. However there are only limited such open-source applications available for use which use chest X-Ray images. Publicly available data on chest X-Rays for COVID-19 are also limited. COVID-Net [10] is the only approach having an open source and actively maintained tool which has ability to identify COVID-19 as well as other pneumonia while showing considerable sensitivity for COVID-19 detection. COVID-Net uses a machine-driven design exploration to learn the architecture design starting from initial design prototype and requirements. It takes as input a chest X-Ray image and outputs a prediction among three classes: Normal, Pneumonia and COVID-19. We treat this model as our baseline, comparing our results with it.

Azemin et al used a deep learning method based on the ResNet-101 CNN model. In their proposed method, thousands of images were used in the pre-trained phase to recognize meaningful objects and retrained to detect abnormality in the chest X-ray images. The accuracy of this method was only 71.9%.

Yoo et al applied chest X-ray radiography (CXR) images to classify using a deep learning-based decision-tree classifier for detecting COVID-19. This classifier compared three binary decision trees based on the PyTorch frame. The decision tree classified CXR images as normal or abnormal, where the third decision tree achieved an average accuracy of 95%.

III. PROBLEM STATEMENT

This thesis objective is to examine if transfer learning improves a model's performance when using a small dataset consisting of radiographs. In this case, the model's goal is to classify healthy, pneumonia and COVID- 19. Transfer learning from different source domains improve performance for classifying COVID-19.

IV. PROPOSED SYSTEM

A. Data Generation

Radiology images of COVID-19 infected patients are rare. We used COVIDx dataset assembled by. They combined open source databases with chest radiology or CT images from. We only used X-ray images to train our model and no CT scan images were used. The total number of COVID-19 infected Chest images are 3616. Normal chest images are 2130 and Pneumonia images are 1345.

B. Preprocessing

We used minimal preprocessing of the dataset before it is fed to our model. The only preprocessing was resizing every image to a similar dimension. We used images of height 224 pixels, width 224 pixels, and the number of channel 3 (224*224*3). Minimal preprocessing makes our inference process faster, so when testing, we can generate the model's output (prediction and heatmap) in real-time.

C. Model Architecture

Our model is comprised of two parts, feature extractor, and classifier. For the feature extractor, we used Densenet-121, and for the classifier, we used a fully connected layer with softmax activation function. The main building block of DenseNet-121 is DenseBlock. These DenseBlocks consist of Convolution Layers. In general, CNN architectures are hierarchical, so feature maps of (l – 1)th layer are input to the lst layer. But in DenseNet, feature-maps off all preceding layers are concatenated and used as input for any particular layer. Also, it's own featuremaps are used as inputs for all subsequent layers. So, for lst layer, features maps of all preceding layers X0, X1, ..., Xl −1 are concatenated and used as it's input.

X1 = H1([X0, X1, …., Xl - 1]) (1)

Here Hl represents the lth layer, Xl is the output of the lth layer, and [X0, X1, ..., Xl − 1] represents the concatenation operation. This special design improves information flow through the network and alleviates vanishing gradient problem. Moreover, DenseNet enhances feature reuse and parameter efficiency and provides each layer the collective knowledge of the network. Another important reason for choosing DenseNet as our architecture is that dense connection has a regularization effect, and it reduces over-fitting on training with smaller data sets, which is our case. DenseNet-121 has four dense blocks and a transition layer between every two dense blocks (Figure 4). Each dense block consists of several convolution layers, and each transition layer consists of a batch normalization, a convolution, and an average pooling layer. To increase nonlinearity ReLU activation function is used in DenseNet, which can be described as:

$$ReLu(x) = \begin{cases} x & x > 0 \\ 0 & x \leq 0 \end{cases} \qquad (2)$$

In our model, the final layer of the Dense-121 is a global average pooling layer that generates the features from the input image. These features are used by the classifier to make the final prediction. For the classifier, we used a fully connected layer, followed by a softmax activation function. For 3-class classification, we used a fully connected layer of three units, and for 2-class classification, we used a fully connected layer of two units. The softmax activation normalizes the output of the fully connected layer and generates a probability distribution over the predicted output classes. The equation of the softmax function can be written as follows:

$$\sigma(\vec{z}_i) = \frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_i}} \qquad (3)$$

Here, ~z is the input vector of the softmax function, zi values are the components of the input vector, and K is the number of classes.

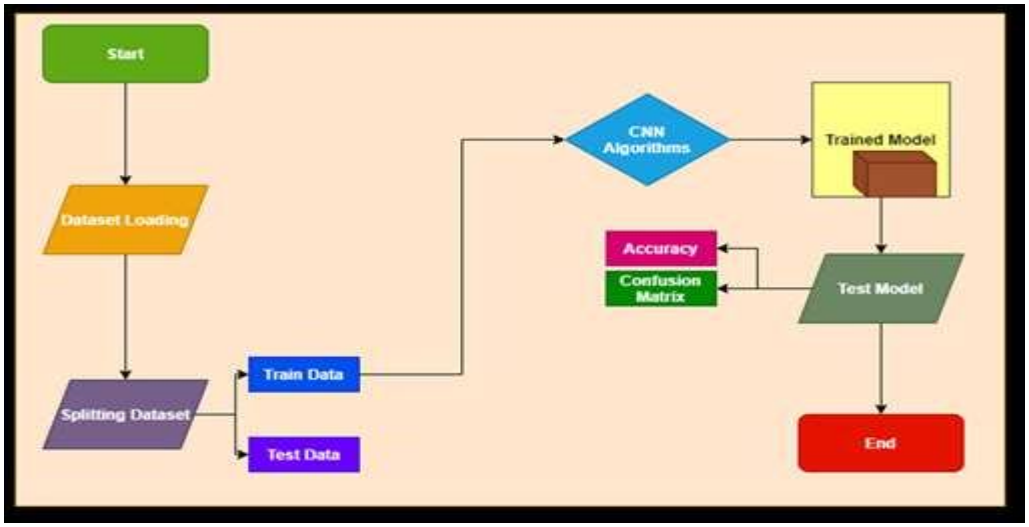


Fig. 1 Proposed System

V. SOFTWARE REQUIREMENT

A. Python

Python is a simple, general purpose, high level, and object-oriented programming language. Python is an interpreted scripting language also. Guido Van Rossum is known as the founder of Python programming. Python is a general purpose, dynamic, high level, and interpreted programming language. It supports Object Oriented programming approach to develop applications. It is simple and easy to learn and provides lots of high-level data structures. Python is easy to learn yet powerful and versatile scripting language, which makes it attractive for Application Development. Python's syntax and dynamic typing with its interpreted nature make it an ideal language for scripting and rapid application development. Python supports multiple programming pattern, including object-oriented, imperative, and functional or procedural programming styles. Python is not intended to work in a particular area, such as web programming. That is why it is known as multipurpose programming language because it can be used with web, enterprise, 3D CAD, etc. We don't need to use data types to declare variable because it is dynamically typed so we can write a=10 to assign an integer value in an integer variable. Python makes the development and debugging fast because there is no compilation step included in Python development, and edit-test-debug cycle is very fast. In most of the programming languages, whenever a new version releases, it supports the features and syntax of the existing version of the language, therefore, it is easier for the projects to switch in the newer version. However, in the case of Python, the two versions Python 2 and Python 3 are very much different from each other.

B. Flask

Flask is an API of Python that allows us to build up web-applications. It was developed by Armin Ronacher. Flask's framework is more explicit than Django's framework and is also easier to learn because it has less base code to implement a simple web-Application. A Web-Application Framework or Web Framework is the collection of modules and libraries that helps the developer to write applications without writing the low-level codes such as protocols, thread management, etc. Flask is based on WSGI (Web Server Gateway Interface) toolkit and Jinja2 template engine.

VI. IMPLEMENTATION

A. Methodology

Convolutional Neural Networks (CNN) have completely dominated the machine vision space in recent years. A CNN consists of an input layer, output layer, as well as multiple hidden layers. The hidden layers of a CNN typically consist of convolutional layers, pooling layers, fully connected layers and normalization layers (ReLU). Additional layers can be used for more complex models.3 hidden layers. The hidden layers of a CNN typically consist of convolutional layers, pooling layers, fully connected layers and normalization layers (ReLU). Additional layers can be used for more complex models. Examples of a typical CNN can be seen in and it is depicted in Figure 1. The CNN architecture has shown excellent performance in many Computer Vision and Machine Learning problems. CNN trains and predicts in an abstract level, with the details left out for later sections. This CNN model is used extensively in modern Machine Learning applications due to its ongoing record breaking effectiveness. Linear algebra is the basis for how these CNNs work. Matrix vector multiplication is at the heart of how data and weights are represented. Each of the layers contains a different set of characteristics for an image set. For instance, if a face image is the input into a CNN, the network will learn some basic characteristics such as edges, bright spots, dark spots, shapes etc., in its initial layers. The next set of layers will consist of shapes and objects relating to the image which are recognizable such as eyes, nose and mouth. The subsequent layer consists of aspects that look like actual faces, in other words, shapes and objects which the network can use to define a human face. CNN matches parts rather than the whole image, therefore breaking the image classification process down into smaller parts (features). A 3x3 grid is defined to represent the features extraction by the CNN for evaluation. The following process, known as filtering, involves lining the feature with the image patch. One-by-one, each pixel is multiplied by the corresponding feature pixel, and once completed, all the values are summed and divided by the total number of pixels in the feature space. The final value for the feature is then placed into the feature patch. This process is repeated for the remaining feature patches followed by trying every possible match- repeated application of this filter, which is known as a convolution. The next layer of a CNN is referred to as "max pooling", which involves shrinking the image stack. In order to pool an image, the window size must be defined (e.g. usually 2x2/3x3 pixels), the stride must also be defined (e.g. usually 2 pixels). The window is then filtered across the image in strides, with the max value being recorded for each window. Max pooling reduces the dimensionality of each feature map whilst retaining the most important information. The normalization layer of a CNN, also referred to as the process of Rectified Linear Unit (ReLU), involves changing all negative values within the filtered image to 0.

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |

This step is then repeated on all the filtered images, the ReLU layer increases the non-linear properties of the model. The subsequent step by the CNN is to stack the layers (convolution, pooling, ReLU), so that the output of one layer becomes the input of the next. Layers can be repeated resulting in a “deep stacking”. The final layer within the CNN architecture is called the fully connected layer also known as the classifier. Within this layer every value gets a vote on determining the image classification. Fully connected layers are often stacked together, with each intermediate layer voting on phantom “hidden” categories. In effect, each additional layer allows the network to learn even more sophisticated combinations of features towards better decision making. The values used for the convolution layer as well as the weights for the fully connected layers are obtained through back propagation, which is done by the deep neural network. Back propagation is whereby the neural network uses the error in the final answer to determine how much the network adjusts and changes.

The Inception-v3 model is an architecture of convolutional networks. It is one of the most accurate models in its field for image classification, achieving 3.46% in terms of “top-5 error rate" having been trained on the ImageNet dataset. Originally created by the Google Brain team, this model has been used for different tasks such as object detection as well as other domains through Transfer Learning.

B. Transfer Learning

Transfer Learning is a Machine Learning technique whereby a model is trained and developed for one task and is then re-used on a second related task. It refers to the situation whereby what has been learnt in one setting is exploited to improve optimization in another setting. Transfer Learning is usually applied when there is a new dataset smaller than the original dataset used to train the pre-trained model. This paper proposes a system which uses a model (Inception-v3) in which was first trained on a base dataset (ImageNet), and is now being repurposed to learn features (or transfer them), to be trained on a new dataset (CIFAR-10 and Caltech Faces). With regards to the initial training, Transfer Learning allows us to start with the learned features on the ImageNet dataset and adjust these features and perhaps the structure of the model to suit the new dataset/task instead of starting the learning process on the data from scratch with random weight initialization. TensorFlow is used to facilitate Transfer Learning of the CNN pre-trained model. We study the topology of the CNN architecture to find a suitable model, permitting image classification through Transfer Learning. Whilst testing and changing the network topology (i.e. parameters) as well as dataset characteristic to help determine which variables affect classification accuracy, though with limited computational power and time.

VII. RESULTS

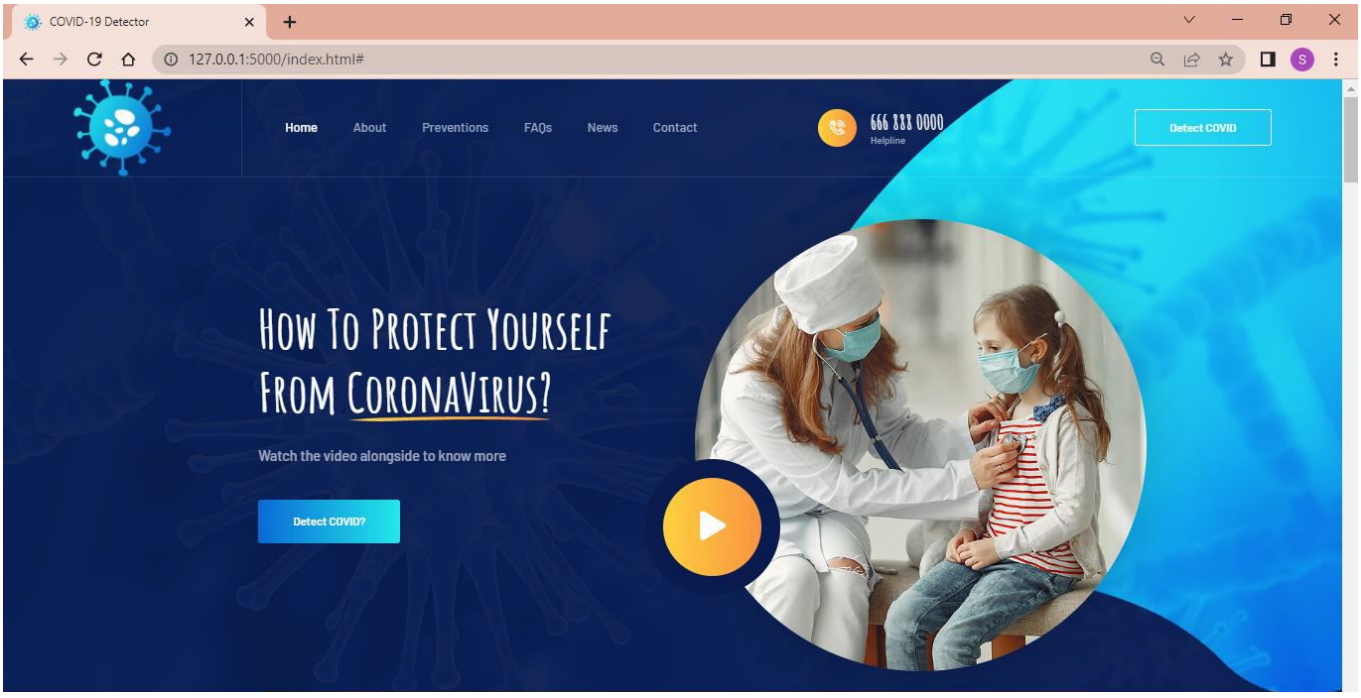


Fig. 2 Home Page for COVID-19 Detector

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 10 Issue IV Apr 2022- Available at www.ijraset.com

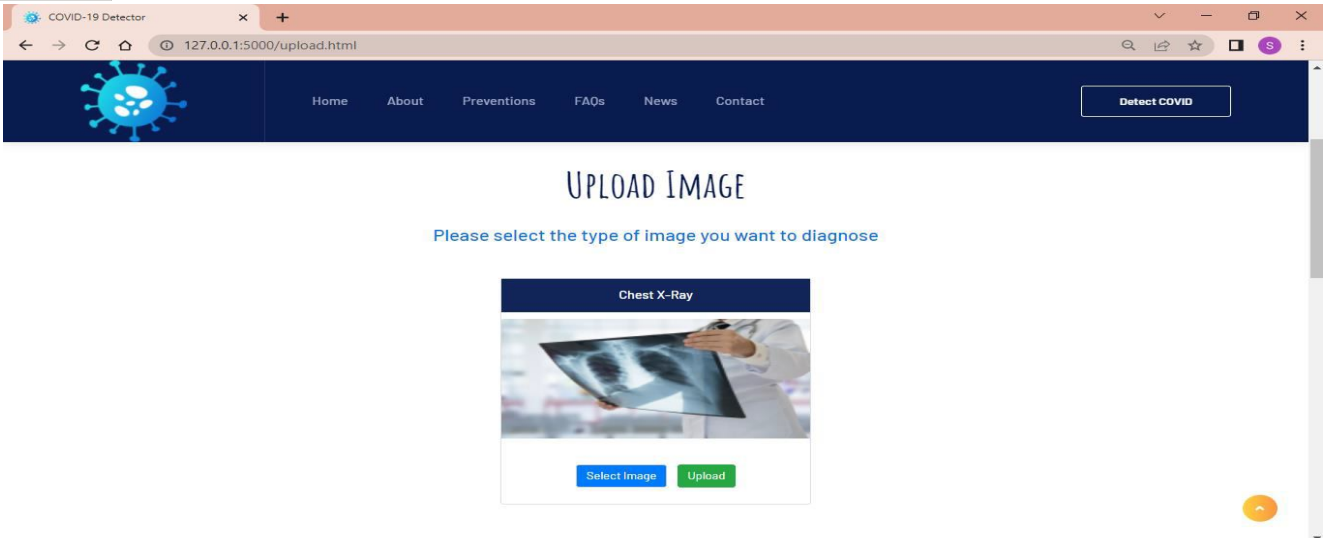


Fig. 3 Image Uploading Window

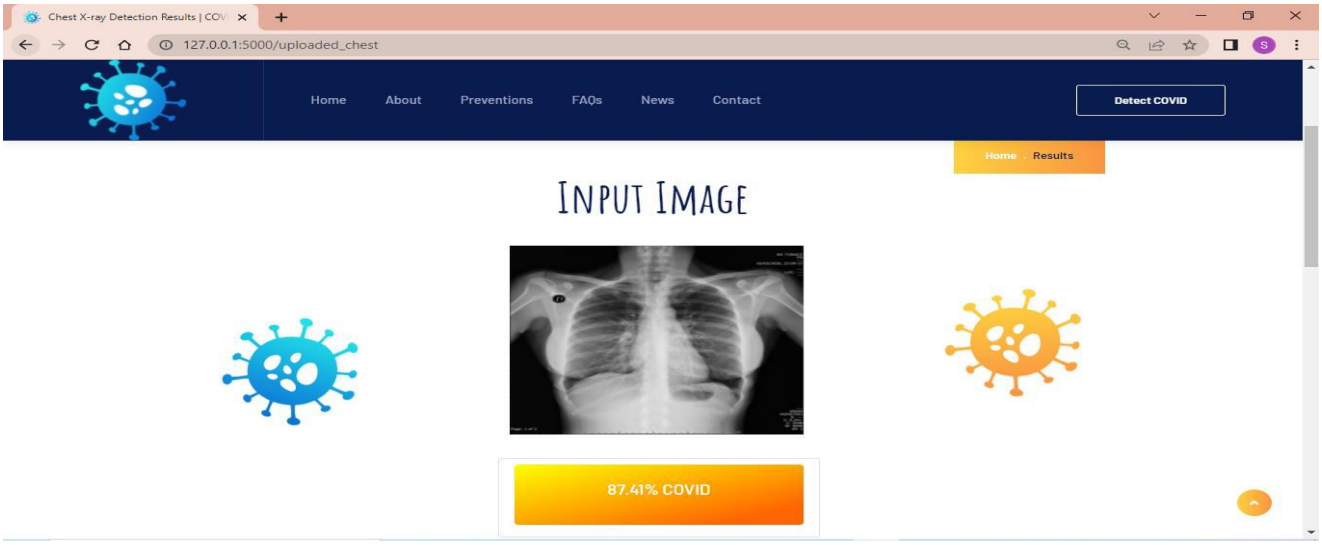


Fig. 4 COVID Detected

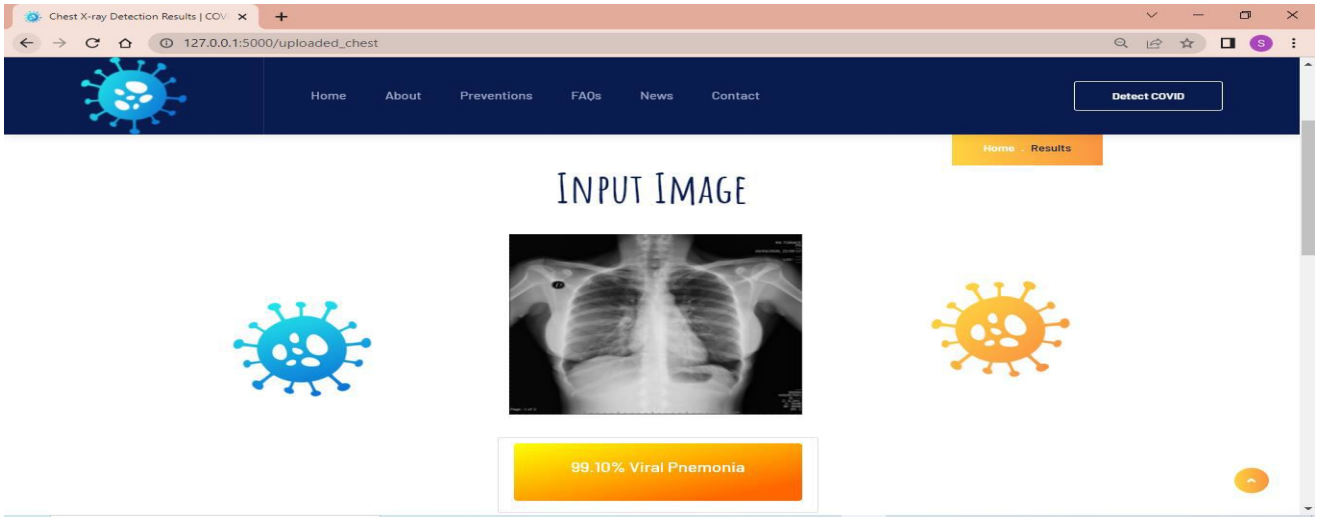


Fig. 5 Pneumonia Detected

```
Epoch 1/5
75/75 [==============================] - 30s 396ms/step - loss: 0.4704 - acc: 0.8438 - val_loss: 0.3692 - val_acc: 0.8750
Epoch 2/5
75/75 [==============================] - 28s 372ms/step - loss: 0.4328 - acc: 0.8475 - val_loss: 0.3489 - val_acc: 0.8800
Epoch 3/5
75/75 [==============================] - 28s 378ms/step - loss: 0.3936 - acc: 0.8542 - val_loss: 0.2921 - val_acc: 0.8967
Epoch 4/5
75/75 [==============================] - 28s 378ms/step - loss: 0.3847 - acc: 0.8604 - val_loss: 0.2626 - val_acc: 0.9083
Epoch 5/5
75/75 [==============================] - 28s 377ms/step - loss: 0.3684 - acc: 0.8625 - val_loss: 0.2672 - val_acc: 0.9033
```

Fig. 6 EPOCH

As we have used VGG16 model for our web application it gives more accuracy compare to other model like resnet, densenet etc. As we have kept EPOCH = 5, to get the best accuracy. EPOCH is total number of iteration for training the machine learning model with all the training data in one cycle for getting the best accuracy.

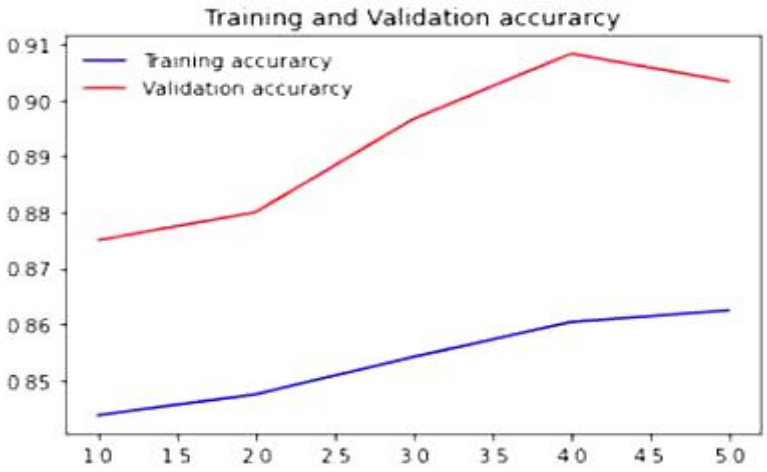


Fig. 7 Graph for Training and Validation Accuracy

In the above graph, red line represents validation accuracy (val_acc) from Fig. 6, blue line represents training accuracy (acc) from Fig. 6.



Fig. 8 Graph for Training and Validation Loss

In the above graph, red line represents validation loss (val_loss) from Fig. 6, blue line represents training loss (loss) from Fig. 6.

VIII. CONCLUSIONS

In this work, we showed a novel transfer learning-based approach to detect COVID-19. To assure that our model can differentiate COVID-19 radiology images from both healthy persons and pneumonia patients, we performed both 2-class and 3-class classifications. To guarantee the robustness and consistency of our model, we implemented patient-wise 10- fold cross-validation. Moreover, we performed an explain ability analysis to interpret and visualize how our model works. Our extensive experiments suggest that COVID-Dense Net can be used effectively for detecting COVID-19 from chest radiology images.

IX. FUTURE SCOPE

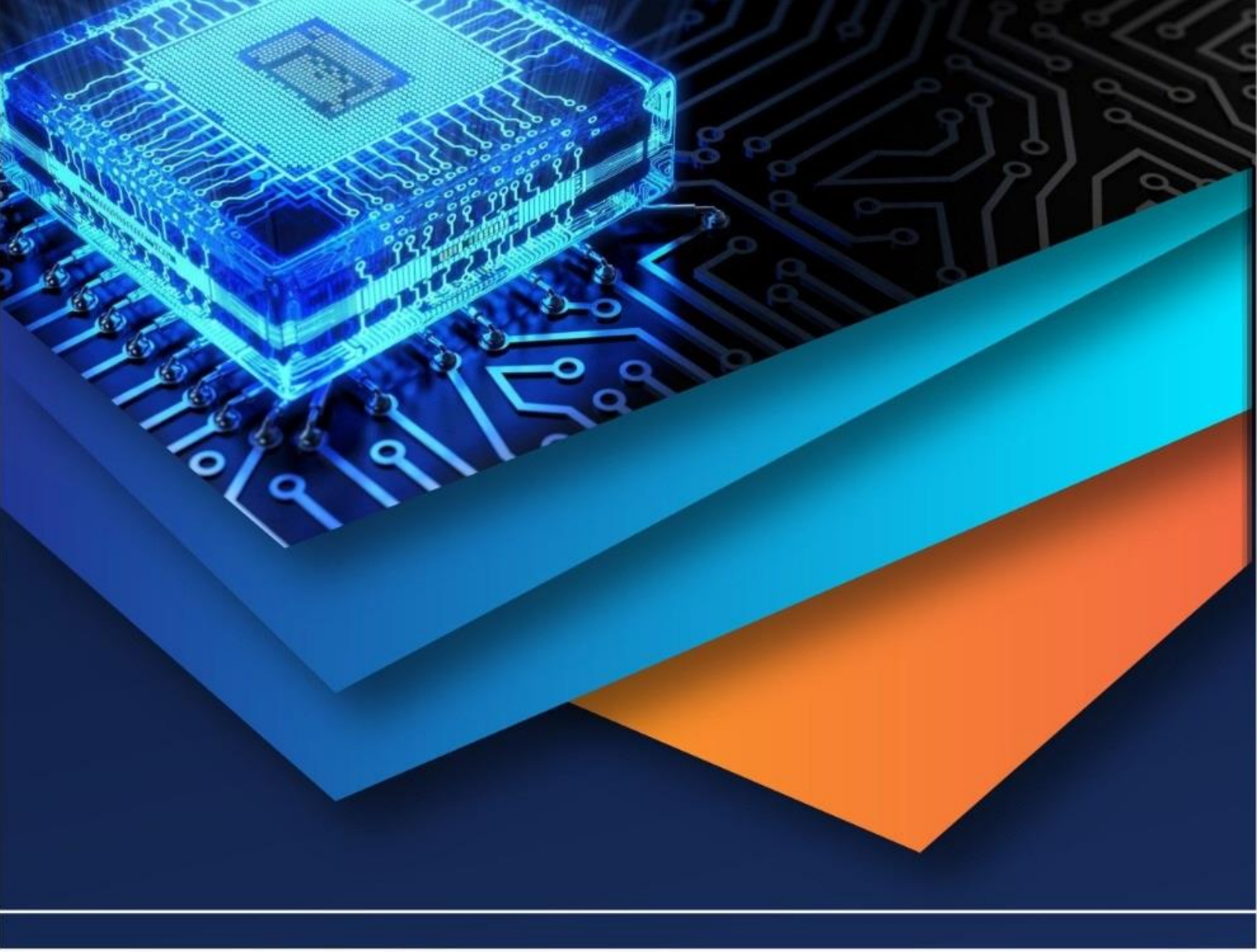
By improving the networks, we can achieve 100% accuracy. The future work for this model is to add Grad Cam techniques to visually see the COVID infected areas for better results. This Model proves that Convolutional Neural Networks can able to work superlatively in the medical field also.

X. ACKNOWLEDGMENT

After the completion of this work, words are not enough to express feelings about all those who helped us.
It is a great pleasure and moment of immense satisfaction for us to express our profound gratitude to our guide, whose constant encouragement enabled us to work enthusiastically. Her perpetual motivation, patience and excellent expertise in the discussion during progress of the project work have benefited us to an extent, which is beyond the expression.
We would also like to give our sincere thanks to Dr. Savita Sangam, Head of Department, Prof. M. R. Gorbal, Project Guide, for her guidance, encouragement and support during the project work. We are also thankful to Dr. Pramod R. Rodge, Principal, for providing an outstanding academic environment and also for providing the adequate facilities.
Last but not the least we would also like to thank all those who directly or indirectly helped us in completion of our work.

REFERENCES

[1] Alexander Wong and Linda Wang. Covid-net: A tailored deep convolutional neural network design for detection of covid-19 cases from chest radiography images. 03 2020.
[2] Gao Huang, Zhuang Liu, Kilian Weinberger.Densely, and Laurens van der Maaten connected convolutional networks. 07 2017.
[3] T Huang. Computer vision: Evolution and promise. 10 2020.
[4] Chi Chen. AN INTRODUCTION TO COMPUTER VISION IN MEDICAL IMAGING, pages 1–16. 01 2014.
[5] Andrew Zisserman and Karen Simonyan. Very deep convolutional networks for large-scale image recognition. arXiv 1409.1556, 09 2014.
[6] A Frangi, J Hornegger, N Navab, Olaf Ronneberger, Philipp Fischer, Thomas Brox, and W Wells. Medical image computing and computerassisted intervention–miccai 2015. Cham: Springer, pages 234–241, 2015.
[7] Ioannis D. Apostolopoulos and Tzani A. Mpesiana. Covid-19: automatic detection from xray images utilizing transfer learning with convolutional neural networks. Physical and Engineering Sciences in Medicine, 43(2):635–640, April 2020.
[8] Abdullah Faqih Al Mubarok, Abdullah Faqih Al Mubarok, Dominique Jeffrey A. M., Pneumonia Detection with Deep Convolutional Architecture, 2019.
[9] Brandon G. Sibbaluca, Nanette V. Dionisio, Sammy V. Militante, Pneumonia and COVID-19 Detection using Convolutional Neural Networks, 2020.
[10] www.kaggle.com
[11] www.keras.com



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Call : 08813907089 (24*7 Support on Whatsapp)