



IJRASET

International Journal For Research in
Applied Science and Engineering Technology



INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 9 Issue: VI Month of publication: June 2021

DOI: <https://doi.org/10.22214/ijraset.2021.35595>

www.ijraset.com

Call:  08813907089

E-mail ID: ijraset@gmail.com

Lung Disease Detection Using CNN

Veeramalla Sowmya¹, Edla Saivarma², Bompally Bhanu Teja³, Kondam Deekshith Reddy⁴

^{1, 2, 3, 4}Electronics and Computer Engineering, Sreenidhi Institute of Science and Technology, India

Abstract: Covid Pneumonia is a life-threatening bacterial disease in humans that affects one or both lungs and is caused by the bacteria *Streptococcus pneumonia*. Also known as Covid-19, this is a respiratory illness that was first discovered in Wuhan, China. Expert radiotherapists must evaluate chest X-rays used to diagnose pneumonia. As a result, establishing an autonomous system for detecting pneumonia would be advantageous for treating the condition quickly, especially in distant places. The statistical results show that using pre trained CNN models and supervised classifier algorithms to analyse chest X-ray pictures, specifically to diagnose Pneumonia, can be highly advantageous. By constructing certain convolution neural network designs, we are developing a classifier model that accurately predicts if a person has covid or pneumonia.

Keywords: Support Vector Machine, Naïve Bayes, cyberbullying, social media, train data, test data, twitter, detection.

I. INTRODUCTION

Corona virus 2019 was a novel disease that appeared in 2019 that was previously unknown to humans (COVID-19). Corona viruses (CoVs) are a group of viruses that cause respiratory diseases such as SARS-CoV and Middle East respiratory sickness (MERS-CoV) (MERS-CoV). In December 2019, the novel corona virus first appeared in China, then spread to a number of other nations [1-3]. Automatic diagnostic tests can be very useful in decreasing the spread of this disease [4]. One of the most successful techniques of using artificial intelligence to diagnose COVID-19 infection from a medical imaging, such as an X-ray, is to conduct an in-depth study [2, 4]. COVID-19 was previously detected using detailed X-ray measurements from the chest. COVID-19 can work for up to 3 hours in air, 4 hours in copper, and 72 hours in plastic and stainless steel, according to new research. However, the public health study team has not yet agreed on the solutions to these topics, and they are still being researched. Infected people's lungs and tissues are attacked by COVID-19. Some people may not experience symptoms in the beginning of the period, when the flu and cough are the most common complaints. Body aches, sore throats, and potential headaches are some of the other symptoms. One of the primary suggestions in a recent statement of international consensus published by the Fleischner Society is to employ chest radiography of patients with COVID-19 in resource areas where computed tomography (CT) is accessible. Because their illnesses deteriorated throughout the photographing, all of the patients were already in solitary confinement and hospitalised. The sensitivity of chest X-rays was 68 percent (no information about the details provided). While X-rays are used in a small number of cases in select COVID-19 patients, the typical findings suggest that the most prevalent lung abnormalities are fluid and/or tissue in the lungs that prevent gas exchange in the lung alveoli. The lungs and lower limbs are also affected by the lack of ground glass and shadowing nodules. Medical and professional groups aim to provide recommendations due to the unclear nature of the data. The focus is on CT because of the data problem. Over the last decade, in-depth learning, which gets its name from specific concealed fields, has earned a unique place in the field of AI by producing useful results for both picture editing applications and retrospective challenges. Images or data must be pre-processed before being sent into a network. ConvNet, on the other hand, was originally defined as a neural network that requires very minor image processing before dedicating itself to the network, as well as a system that can extract objects from images to increase the neural network's learning function. There are three layers in Traditional ConvNet: convolution, integration, and fully integrated layers. The method of dividing images at a preset level of segments and employing filters to eliminate features from the image is done on a convolutional layer using a mask. The feature map is then built using the function to activate the values discovered in the mask, which estimates the feature on the 2D map. Activation activity makes neurons more sensitive to non-linear lines and lowers the neural network's calculation costs. The converted line unit (ReLU) is the most generally utilised activation function on CNN; because not all neurons work at the same time, it enables a speedy convergence when metals get the proper values to create a learned response during training. On a feature map meant to reduce image sizes, the blending work is done. A feature map is finally integrated into a vector and supplied to a fully connected layer. In a fully connected layer, the neural network is integrated with the separation of the input patterns, and the layer's objectives are based on the repetition of the error regeneration of the metals in this layer.

II. MATERIALS AND METHOD

A. Flow Chart

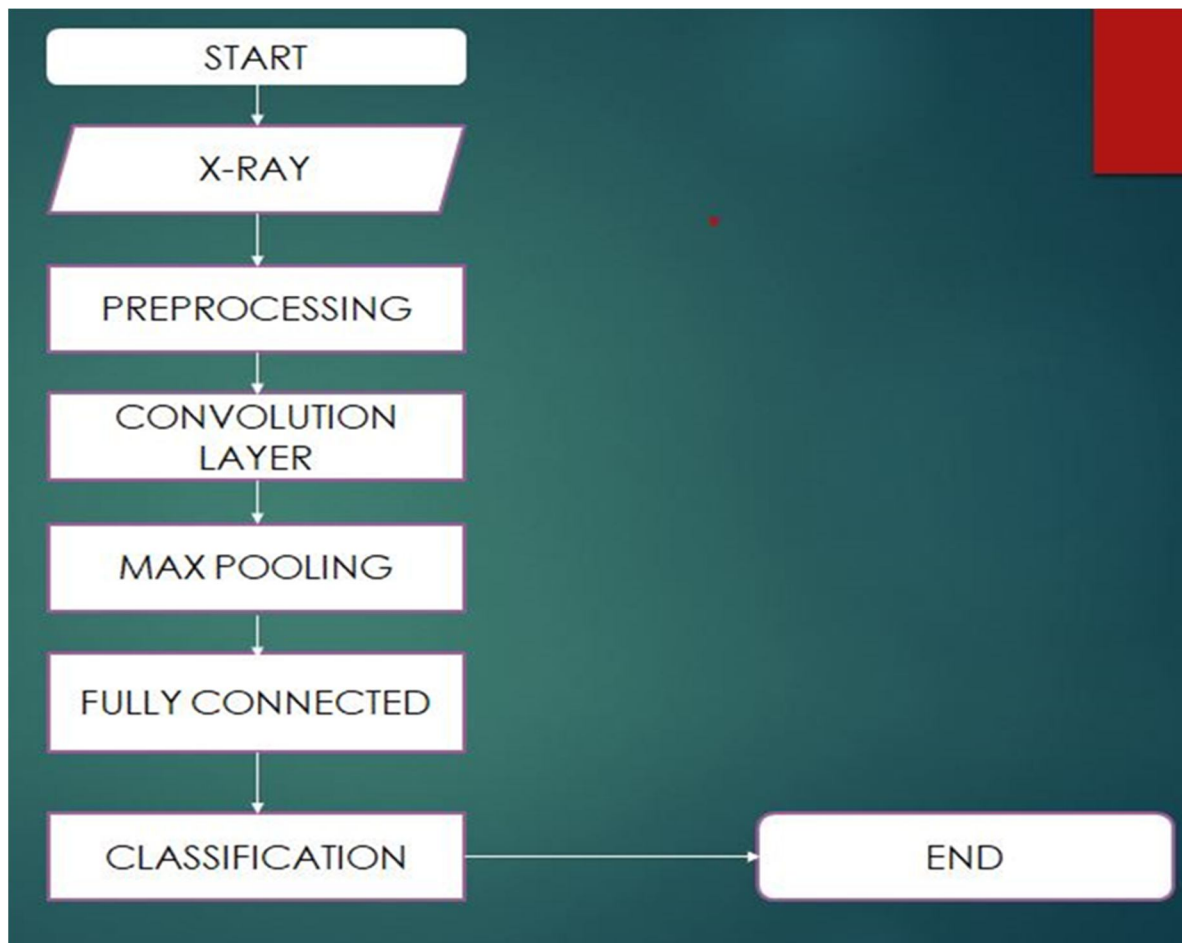


Fig 1. Architecture

B. Dataset Collection

Cohen provided a total of 3616 COVID-19 chest X-ray images. The COVID-19 group had 112 male patients and 45 female patients, with a mean age of 58.8 14.9 years. Because this is the first publicly available picture collection for the COVID-19 X-ray, the data utilised in this study does not include the whole associated metadata, and it was done in a restricted period, some patient details are not available. In addition, Kaggle revealed 501 regular and 3616 pneumonia chest X-ray photos. Because the photographs were all different sizes, they were all resized to 125 by 125 pixels.

C. Preprocessing

All raw data conversions must be completed prior to the provision of learning equipment or an in-depth learning algorithm. Training a convolutional neural network on immature photos, for example, could result in seizures. Images are available in a range of sizes and forms. They also have a variety of sources. Some photographs, for example, are what we term "natural photos," meaning they were captured in colour in the real world. For instance, a fruit image is a natural image, whereas an X-ray image is not. To think about all of this diversity, we need to pre-process any visual data.

RGB is the most common encoding type, and RGB is used in the majority of the "nature images" we see. Also, make photos of the same size during the first step of pre-processing data. The network design and input data format must be carefully considered when creating an effective neural network model. The latter is the subject of this text. Image number, image height, picture width, number of channels, and number of levels per pixel are the most frequent image input parameters. The colours Red, Green, and Blue (RGB) are frequently represented by three data channels with a pixel level of [0,255].

D. Image Scaling

It's time to measure each image appropriately once we've determined that they're all square (or have certain pre-defined dimensions). Images having a width and height of 100 pixels were chosen. Each image's width and height must be multiplied by a factor of 0.4 (100/250). Lifting techniques come in a variety of forms, and we frequently employ the library function to do so

E. Normalizing Image Inputs

The data setup phase assures that each input parameter (in this case, the pixel) has the same data distribution. This speeds up the integration process while the network is being trained. By subtracting the mean per pixel and then separating the result by standard deviation, data can be standardised. The Gulf of Zero-centered curve will be used to describe the distribution of such data. We require pixel numbers for image input to work properly, thus we can measure standard data in the range [0, 1] or [0, 255].

F. Dimensionality Reduction

On a grey scale, we can opt to blend RGB channels into one channel. It's commonly thought of as a magnitude reduction when the neural network's function is permitted to be unstable at that level, or to make the training challenge more manageable.

G. Data Augmentation

Enlarging existing data sets with disruptive versions of old photos is another typical processing method. Affinity adjustments such as measurement, rotation, and other affine alterations are prevalent. This is done in order to expose the neural network in a number of different ways. As a result, the neural network is less likely to discover undesirable elements in the data set.

H. Numpy

NumPy is a Python library that provides functions to operate on a straight algebra, fourier transform, and matrices and is used to work with the same members. Python of Numbers is referred to as NumPy. We have a list in Python that serves the same purpose as the members but isn't as slow to process. NumPy aspires to provide members with a service that is 50 times faster than regular Python lists. In NumPy, the object with the same members is named ndarray, and it has a lot of help functions that make dealing with ndarray a lot easier. In data science, when speed and resources are critical, scheduling is often used.

I. Pandas

Pandas is a Python library that creates fast, flexible, and descriptive data structures for performing structured operations (structured, diverse, and possibly sophisticated) on simple and accurate chronological series data. It aspires to be a cutting-edge framework for performing real-world data analysis in Python. Furthermore, it aspires to be the most complete and versatile data analysis and decryption tool available in any language. The two basic categories of pandas data, Series (1-dimensional) and Data Frame (2-dimensional), can handle the majority of financial, mathematics, social scientific, and engineering problems. Data Frame provides all of R's data frame features, plus a lot more. Pandas is a Python library built on top of NumPy that is designed to work well in the field of computer science and with a variety of other third-party libraries.

J. Matplotlib

Matplotlib is a graph editing package that also serves as a visualisation tool. John D. Hunter was the one who came up with the idea. It is open source, and we are allowed to use it.

K. Keras

Keras is a Python-based in-depth learning API that extends the TensorFlow machine learning framework. Built with the goal of facilitating speedy testing. The ability to shift from one point of view to another is crucial to doing good. Keras is a cutting-edge Tensor Flow 2 API that provides an accessible and highly productive interface for tackling machine learning issues, with a focus on today's in-depth learning. Provides important releases and components for building high-iteration-rate learning and shipping solutions. Engineers and researchers may use Keras to fully exploit TensorFlow 2's flexibility to scale up and down platforms: you can run Cameras on TPU or big GPU clusters, and transmit your Keras models to work on a browser or mobile device.

L. Tensor Flow

Tensor Flow is an open source machine learning platform that runs from start to finish. It features a comprehensive ecosystem of tools, libraries, and community resources that enable academics to condense machine learning technology and developers to create and deploy strong machine learning applications quickly and easily.

Tensor Flow has a variety of output levels from which you can choose the best one for your purposes. Create and train models with the powerful Keras API, which makes Tensor Flow and machine learning more accessible. If you require greater flexibility, enthusiasm permits you to execute faster and fix errors more accurately. Use the Distribution Strategy API for distributed training on diverse hardware configurations without modifying the model description for large ML training workloads.

M. Convolution Layer

A CNN is made up of numerous layers of neurons, each of which is a non-linear function that transforms the effects of the preceding layer in a direct manner. Convolutional layers and integration layers are the most common types of layers. Convolutional layers have a weight that must be learned, whereas composite layers use a constant function to change performance. Convolutional neural networks' main building pieces are flexibility layers. Convolution is a straightforward application of the input filter that yields outcomes. The feature map, which depicts the positions and strengths of the feature identified in the input, such as an image, is created when the same filter is used repeatedly in the installation. The capacity to automatically read a large number of filters directly to the training database under the conditions of a specific predictive modelling issue, such as image separation, is a new architecture of convolutional neural networks. As a result, several unique elements can be detected across the input photographs.

The convolutional layer has a number of filters with settings that must be investigated. The filters' height and weight are less than the input volume. Each filter has an input volume that is used to calculate the activation map created by neurons. In other words, the filter moves the width and height of the input, and all sub-areas calculate the dot products between the input and the filter. Inserting maps to activate all filters by magnitude depth yields the volume of the convolutional layer extraction. Because each filter's width and height are smaller than the input, each neuron in the activation map is only connected to a small portion of the input volume area. In other words, each neuron's allowable field size is modest and equal to the filter size. The creation of the visual cortex in animals [22], where the acceptable levels of cells are minimal, triggers spatial connection. The convolutional layer's local connectivity allows the network to read the most responsive filters in the input region, making input location adjustment easier (input image, pixel is more compatible with nearby pixels than remote pixels). Furthermore, because the activation map is created by convolutioning the filter and the input, filter parameters are dispersed throughout all local positions. The number of factors for speech efficiency, learning efficiency, and general performance are reduced when weights are shared. The first layer to extract features from an input image is convolution. By reading image features with small squares of input data, conversion retains the link between pixels. It's a mathematical function with two inputs: an image matrix and a filter or kernel.

N. Maxpooling Layer

The feature map for convolutional layers has the drawback of recording the exact location of the input items. This means that a little change in the input image's feature region will result in a new feature map. This can be accomplished by scaling, rotating, and making other minor adjustments to the supplied image. Down sampling is the most popular method for dealing with this issue caused by the use of a signal. This is when a low-resolution version of the input signal is made with major or key structural features but without the tiny details that aren't needed for this activity. To make the feature maps smaller, pool insertion layers are employed. Reduce the amount of learning parameters and computers established in the network as a result. The features contained in the feature map region created by the convolution layer are summarised by the integration layer. As a result, some operations are carried out on shortened elements rather than the precisely set elements generated by the convolution layer. This increases the model's ability to change the shape of items in the input image. Kernel releases a considerable amount of convincing space during the debugging phase known as Max Pooling. In the Convolutional Neural Network, Max Pooling essentially states that we will only develop information if it is the most intelligent intelligence available.

Step 2: Max-pooling with a $2 * 2$ kernel on a $4 * 4$ station. $2 * 2$ Kernel is known to exist. The first $2 * 2$ set has four values of 8,3,4,7, with the kernel centred on the channel. The value of Max-Pooling is greater than "8."

Shift Invariance, Rotational Invariance, and Scale Invariance are all added via Max Pooling. As we can see in image $2 * 2$, even a small adjustment or change does not make a difference. Shift invariance is the term for this. Max Pooling, on the other hand, circulates slowly and steadily.

A popular strategy for ordering layers within a convolutional neural network is to add a consolidation layer after the convolutional layer. This pattern can be repeated once or more in a given model. The Merge layer operates on each feature map separately to build a new set of integrated feature maps with the same number of features. Merging contains a merge function that works similarly to a filter for inserting maps. The size of the merge or filter is less than the size of the feature map; in particular, $2 * 2$ pixels per line of 2 pixels is nearly often used.

This indicates that the aggregate layer will always cut each element's map size by two elements, i.e., each element's map size will be halved, decreasing the number of pixels or values on the map to one-quarter of its original size. A composite layer applied on a 6 6 (36 pixel) feature map, for example, will result in a combined 3 3 (9 pixel) integrated feature map. The aggregate layer is used to compile and construct feature-mapped feature maps, and the result is a condensed version of the features discovered in the input. Small changes in the feature area in the input recognised by the convolutional layer result in an integrated feature map with the feature in the same area, which is advantageous. Model variability in local translation refers to this enhanced ability to integrate.

Maxpool2D:- Data performance in a 2D area with several locations. Takes a greater number than the input window (size provided by pool size) for each input channel to reduce the input by area (height and width). The roads in each grade move the window.

O. Activation

Working with him is like taking a break from work. You can feed data into one of the activation tasks and get the same output, which we refer to as the function. Yes, just as we have a distinct function in arithmetic that accepts a specific matrix of numbers as input and returns a result. For example, $y = mx$ is a function that returns a different y depending on the x value. The similar thing happens in the convolution's first layer. Identity, tanh, Arc Tan, and Relu are some of the action functions utilised on CNN (modified line unit). For these functions, input is provided, as well as compliance. The objective of showing these layers is to create discrepancies, as if you'll notice that these functions aren't available online, resulting in product incompatibility. When linearity is present in your neural network, training becomes faster and more precise. When the Relu function is invoked, all negative values are converted to 0 or positive.

P. ReLU

The correct line rectification function, or ReLU for short, is a thin line function that, if accurate, returns a direct input; otherwise, it returns zero. Because the model it utilises is easy to train and typically delivers higher performance, it has become a function of automation in many types of neural networks. The goal of employing corrective action is to amplify the disparity in our images. We'd like to do this because images aren't always compatible. The modifier works to further divide the linearity in order to create a unity that can establish the tone when used for convolution.

To see how this works in practise, consider the following image and the changes that occur when we continue our job of convolution followed by rectification.

Q. Fully Connected Layer

Layout that is completely connected Simply put, it improves neural networks. The network's final tiers are Fully Connected Lists. The outcome of the final Pooling or Convolution Layer, which is softened and placed in a fully connected layer, is input to a fully connected layer. This vector is flattened and joined to a few fully connected layers that perform the same mathematical operations as Artificial Neural Networks. Following the fully linked layers, the last layer employs the softmax activation function (rather than ReLU) to locate input parameters in a certain category (fragmentation). In neural networks, the layers connected to are those in which all of the inputs from one layer are connected to the complete launch unit of the following layer. The last few layers of the most popular machine learning models are full layers that incorporate data from previous layers to produce the final result. It is Convolution's second most expensive layer. The convolution layers' output represents the data's high-level characteristics. While the output layer can be increased and linked to the output layer, adding a completely integrated layer is a cheap (usually) approach to understand the inaccuracy of these feature combinations. In essence, convolution layers offer sound, low, and anomalous features, while a fully connected layer trains a function (potentially non-linear) in that space.

The vector of the input element is represented by CNN's fully connected layer (FC). Important installation details are contained in this vector / tensor / layer feature. This feature vector is used to proceed with modification, decryption, or installation in another network such as an RNN to translate to another output after a network has been trained. It's a type of embedded vector. This feature vector is utilised during training to determine losses and to assist the network in locating the train. The convolution layers that come before the FC layers provide information on the insert image's local features, such as edges, blobs, and forms. You can have numerous filters for each conv layer, each representing one of the local features. All of the most significant convolution layers' data is compiled and merged in the FC layer.

The final Pooling Layer release serves as an input into the so-called Fully Connected Layer at the end of CNN. One or more of these layers could be involved ("fully connected" means that every node in the first layer is connected to all nodes in the second layer). Fully linked layers distinguish between each other using features extracted from preceding layers.

This layer is often a standard ANN with the softmax activation function, which eliminates probability (number ranging from 0-1) for each model label you're trying to forecast. FC layers simply learn the elements' indirect functions and categorise photos based on high-quality attributes.

Every neuron in one layer is coupled to every neuron in another layer in fully connected layers. To separate the images, the flat matrix travels through a completely connected layer. The convolution layers' output represents the data's high-level characteristics. While the output layer can be increased and linked to the output layer, adding a completely integrated layer is a cheap (usually) approach to understand the inaccuracy of these feature combinations.

R. Flatten

Flattening data into a one-dimensional array allows it to be inserted into the next layer. We investigate removing convolutional layers in order to create a single long feature vector. It's also linked to a completely connected layer, which is the final separation model. To put it another way, we combine all of the pixel data into a single line and connect it to the last layer. The next step is to soften the integrated map once it has been detected. The method of house softening entails turning a complete map of a single column feature into a single column and then feeding it to a neural network for processing.

S. Dense

Dense uses the following functionality: $\text{output} = \text{output} (\text{dot} (\text{input}, \text{kernel}) + \text{selection})$, where the activation argument is the most intelligent activation function, the kernel is a weight matrix created by the layer, and the bias vector veas is generated by default (only works if the use of application is True). All of these characteristics are Dense. A dense layer is a network layer that is always linked. The most popular and often utilised layer. The dense layer returns the outcome of the function performed below the input.

$\text{Output} = (\text{dot} (\text{input}, \text{kernel}) + \text{bias}) \text{ activation}$.

T. Dropout

During training, the random dump layer sets the input units to 0 with the average frequency every step, preventing overpopulation. The total of all inputs is not changed since inputs that are not set to 0 are increased by $1 / (1 - \text{average})$.

Because there are no prices lowered at the time of consideration, the disposal layer only applies when the training is set to Truth. If you use `mod.fit`, the training will be set to True by default, and you can also explicitly set the `kwarg` to True when calling a layer in some circumstances.

U. Strides

Stride is a component of convolutional neural networks, which are neural networks that compress image and video data. The stride parameter of a neural network filter transforms the amount of movement in an image or video. When the neural network step is set to 1, for example, the filter moves one pixel (or unit) at a time. The step is normally set to a whole integer rather than a fraction or a decimal because the filter size impacts the output volume included.

III. OUTPUTS AND SCREENSHOTS

```
from keras.preprocessing import image

test_image = image.load_img(imagePath, target_size = (150,150,3))
test_image = image.img_to_array(test_image)
test_image = np.expand_dims(test_image, axis = 0)

result = model.predict(test_image)
if(result[0][0] <= 1.0):
    print("Normal, No covid Detected")
else:
    print("Covid 19 Detected")
```

Normal, No covid Detected

Fig 2. Output of covid negative

Here we can see that how the data is being detected for offensive language used in the dataset.

```
from keras.preprocessing import image

test_image = image.load_img(imagePath, target_size = (150,150,3))
test_image = image.img_to_array(test_image)
test_image = np.expand_dims(test_image, axis = 0)

result = model.predict(test_image)
if(result[0][0] <= 1.0):
    print("Normal, No pneumonia Detected")
else:
    print("Pneumonia Detected")
```

Normal, No pneumonia Detected

Fig 3. Output of no pneumonia detected

This is a sample from the dataset, which is used in our model the above dataset is used to create training and testing datasets.

```
Epoch 46/50
11/11 [=====] - 20s 2s/step - loss: 0.3047 - accuracy: 0.8833 - val_loss: 0.2732 - val_acc
uracy: 0.9000
Epoch 47/50
11/11 [=====] - 20s 2s/step - loss: 0.2498 - accuracy: 0.9106 - val_loss: 0.2332 - val_acc
uracy: 0.9167
Epoch 48/50
11/11 [=====] - 21s 2s/step - loss: 0.3204 - accuracy: 0.8864 - val_loss: 0.2444 - val_acc
uracy: 0.9056
Epoch 49/50
11/11 [=====] - 20s 2s/step - loss: 0.2791 - accuracy: 0.8822 - val_loss: 0.2624 - val_acc
uracy: 0.9111
Epoch 50/50
11/11 [=====] - 21s 2s/step - loss: 0.2183 - accuracy: 0.9318 - val_loss: 0.2657 - val_acc
uracy: 0.9000
```

Fig 10. Accuracy of the model

IV. FUTURE WORK

Other lung diseases, such as tuberculosis, may develop in the future. Positive or negative class predictions by CNNs should be viewed with caution unless they are qualitatively evaluated or segmented regions of interest are fed into CNNs in both the training and testing phases. Figures 4 and 5 show various examples of CNNs using texts and medical device traces on X-rays to prevent them from learning the disease's true features.

Segmenting the lung region from chest X-rays and eliminating additional artefacts such as text and medical device traces from chest X-rays are among the future research goals and ongoing study. To construct CNN models that can be generalised and not biased towards a certain country or a targeted demographic, data from various sources must be used.

V. CONCLUSION

In this paper we have detected different types of lung diseases using the convolution neural networks. We have developed an accurate model that would detect various types of lung diseases. We have used the types of convolution neural networks which include convolution layer, max pooling layer and the fully connected layer.

We have described how accurately our model will detect the different types lung diseases which include pneumonia, covid-19. We have collected the dataset from the kaggle and we have overcome the problems of image cropping through the preprocessing technique. We have used the convolution layer for features detection which included in three phases then we have used the highest value max pooling technique which reduces the time complexity and we have used fully connected layer for finding correct weights to the hidden layer and also used the fully connected layer to flatten and dropout such that there will be no time complexity issues. And finally we have created a model which will accurately detect the lung diseases.

We have fostered a methodology towards the discovery of cyberbullying conduct. In the event that we can effectively recognize such posts which are not appropriate for youths or teens, we can viably manage the wrongdoings that are perpetrated utilizing these stages. An algorithm which utilizes Supervised learning to find and restrict twitter cyberbullying. Our model has its roots on both Support Vector Machine and Naive Bayes, additionally for mining, we utilized the TFIDF vectorizer. As the outcomes show us that the precision for identifying Social media bullying content has likewise been extraordinary for Support Vector Machine which is superior to Naive Bayes. The model which is made by us will help individuals from the assaults of web-based media menaces.

VI. ACKNOWLEDGMENT

We owe a great many thanks to a great many people who have helped and supported us throughout this project, which would not have taken shape without their co-operation. Thanks to all.

We express our profound gratitude to Dr.T.Ch.Siva Reddy, Principal and indebtedness to our management Sreenidhi Institute of Science and Technology, Ghatkesar for their constructive criticism.

We would like to specially thank our beloved Head of Department, ECM, Prof. Dr. D.Mohan, for his guidance, inspiration and constant encouragement throughout this research work.

We would like to express our deep gratitude Mrs. N.Swapna, Associate professor of ECM and Mr. M. Nanda Kumar, Associate professor of ECM(Project Coordinators) and Mrs. V.Sowmya, Assistant professor of ECM(Internal Guide), For their timely guidance, moral support and personal supervision throughout the project.

These few words would never be complete if we were not to mention our thanks to our parents, Department laboratory, staff members and all friends without whose co-operation this project could not have become a reality.

REFERENCES

- [1] Liu, Kui, Guixia Kang, Ningbo Zhang, and Beibei Hou. "Breast cancer classification based on fully-connected layer first convolutional neural networks." *IEEE Access* 6 (2018): 23722-23732.
- [2] Kalchbrenner, N., Grefenstette, E. and Blunsom, P., 2014. A convolutional neural network for modelling sentences. arXiv preprint arXiv:1404.2188.
- [3] O'Shea, Keiron, and Ryan Nash. "An introduction to convolutional neural networks." arXiv preprint arXiv:1511.08458 (2015).
- [4] Simard, Patrice Y., David Steinkraus, and John C. Platt. "Best practices for convolutional neural networks applied to visual document analysis." In *Icdar*, vol. 3, no. 2003. 2003.
- [5] Lawrence, S., Giles, C.L., Tsoi, A.C. and Back, A.D., 1997. Face recognition: A convolutional neural-network approach. *IEEE transactions on neural networks*, 8(1), pp.98-113
- [6] Yamashita, Rikiya, Mizuho Nishio, Richard Kinh Gian Do, and Kaori Togashi. "Convolutional neural networks: an overview and application in radiology." *Insights into imaging* 9, no. 4 (2018): 611-629.
- [7] Lin, Xiaofan, Cong Zhao, and Wei Pan. "Towards accurate binary convolutional neural network." arXiv preprint arXiv:1711.11294 (2017).
- [8] Zhang, X., Zhou, X., Lin, M., & Sun, J. (2018). Shufflenet: An extremely efficient convolutional neural network for mobile devices. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 6848-6856).
- [9] Fauci, A. S., Lane, H. C., & Redfield, R. R. (2020). Covid-19—navigating the uncharted.
- [10] Yang, Li, Shasha Liu, Jinyan Liu, Zhixin Zhang, Xiaochun Wan, Bo Huang, Youhai Chen, and Yi Zhang. "COVID-19: immunopathogenesis and Immunotherapeutics." *Signal transduction and targeted therapy* 5, no. 1 (2020): 1-8.



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)