



COVID-19 Detection using Deep Learning

Vandit Gupta

B.Tech Student, Department of IT, Maharaja Agrasen Institute of Technology, Delhi, India

To Cite this Article

Vandit Gupta, "COVID-19 Detection using Deep Learning", *International Journal for Modern Trends in Science and Technology*, 6(12): 421-425, 2020.

Article Info

Received on 16-November-2020, Revised on 09-December-2020, Accepted on 12-December-2020, Published on 18-December-2020.

ABSTRACT

Deep learning is an artificial intelligence function that imitates the workings of the human brain in processing data and creating patterns for use in decision making. Deep learning is a subset of machine learning in artificial intelligence (AI) that has networks capable of learning and recognizing patterns from data that is unstructured or unlabelled. It is also known as deep neural learning or deep neural network. Convolutional Neural Networks (ConvNets or CNNs) are a category of Neural Networks that have proven very effective in areas such as image recognition and classification. ConvNets have been successful in identifying faces, objects and traffic signs apart from powering vision in robots and self-driving cars. ConvNets can also be used for Radio Imaging which helps in disease detection.

This paper helps in detecting COVID-19 from the X-ray images provided to the model using Convolutional Neural Networks (CNN) and image augmentation techniques.

KEYWORDS: Deep Learning, Convolutional Neural Networks, COVID-19, Image Augmentation

I. INTRODUCTION

Covid illness 2019 (COVID-19) is an irresistible infection brought about by serious intense respiratory disorder Covid 2 (SARS-CoV-2). It was first distinguished in December 2019 in Wuhan, Hubei, China, and has brought about a progressing pandemic. Basic side effects incorporate fever, hack, weakness, windedness, and loss of smell and taste. While most of cases bring about gentle manifestations, some advancement to intense respiratory misery disorder (ARDS) perhaps accelerated by cytokine storm, multi-organ disappointment, septic stun, and blood clusters. The time from introduction to beginning of side effects is ordinarily around five days, however may go from two to fourteen days. The loss of life from the disease is heightening at a stressful rate and numerous well being frameworks

around the globe are battling to adapt. Social Distancing is among certain methodologies proposed by the World Health Organization (WHO) to control the spread of this viral contamination. Precise screening needs to be done so that Covid affected patients can be identified soon enough and can be isolated from the general public. This measure is necessary for controlling the spread of the virus.

Switch record polymerase chain response (RT-PCR) testing[2], which can identify SARS-CoV-2 RNA from respiratory examples, (for example nasopharyngeal or oropharyngeal swabs), is the brilliant screening technique for recognizing COVID-19 cases. The high affectability of RT-PCR testing is restricted by the accessibility of test packs. What's more, this method takes a day or two for the results. In this time period, many people

may get affected. Thus, there is a developing need to utilize quicker and dependable screening strategies that could

be additionally affirmed by the RT-PCR testing. A few examinations have proposed the utilization of imaging procedures, for example, X-Rays or Computed Tomography (CT) scans of the chest to search for visual clues related to SARS-CoV-2 viral disease.

It was found in early examinations that patients show anomalies in chest radiographs that are normal for COVID-19 contamination, with some proposing that radiography assessment could be utilized as an essential instrument for COVID-19 screening in affected regions.

Thus Machine Learning and Deep Learning techniques could be instrumental in detecting COVID-19 by using chest X-Rays.

METHODOLOGY

The fundamental point of this paper is to detect if a patient is suffering from covid-19 or not in real time.

The techniques used in this research paper are:

- Convolutional Neural Networks
- Image Augmentation

CONVOLUTIONAL NEURAL NETWORKS (CNN)

Convolutional Neural Networks (ConvNets or CNNs) are a category of Neural Networks that have proven very effective in areas such as image recognition and classification. [3] ConvNets have been successful in identifying faces, objects and traffic signs apart from powering vision in robots and self-driving cars. ConvNets can also be used for Radio Imaging which helps in disease detection.



Figure 1: Object Detection using Deep Learning

Thus, CNN's are widely used for image classification which has enormous number of applications.

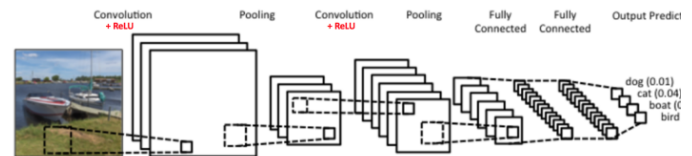


Figure 2: CNN Architecture

The Convolutional Neural Network [4] in Figure is similar in architecture to the original LeNet and classifies an input image into four categories: dog, cat, boat or bird (the original LeNet was used mainly for character recognition tasks). As evident from the figure above, on receiving a boat image as input, the network correctly assigns the highest probability for boat (0.94) among all four categories. The sum of all probabilities in the output layer should be one.

There are four main operations in the ConvNet as shown in Figure above:

- Convolution
- Non Linearity (ReLU) (Activation Function)
- Pooling or Sub Sampling
- Classification (Fully Connected Layer)

Convolution: The objective of the Convolution Operation is to extract the high-level features such as edges, from the input image. ConvNets need not be limited to only one Convolutional Layer. Conventionally, the first ConvLayer is responsible for capturing the Low-Level features such as edges, color, gradient orientation, etc. With added layers, the architecture adapts to the High-Level features as well, giving us a network which has the wholesome understanding of images in the dataset.

ReLU: The Rectified Linear Unit is the most commonly used activation function in deep learning models. The function returns 0 if it receives any negative input, but for any positive value x it returns that value back. So it can be written as $f(x) = \max(0, x)$.

Its main advantage over other activation functions is that in this function overcomes the problem of vanishing gradients.

Pooling: Pooling layer is responsible for reducing the spatial size of the Convolved Feature. This is to decrease the computational power required to process the data through dimensionality reduction. Furthermore, it is useful for extracting dominant features which are rotational and positional invariant, thus maintaining the process of effectively training of the model.

There are two types of Pooling: Max Pooling and Average Pooling. Max Pooling returns the maximum value from the portion of the image covered by the Kernel. On the other hand, Average Pooling returns the average of all the values from the portion of the image covered by the Kernel.

Flattening: Flattening transforms a two-dimensional matrix of features into a vector that can be fed into a fully connected neural network classifier.

Image Augmentation

At the time of writing this paper there were very less images available of Covid-19 affected Chest X-rays, so in order to increase data image augmentation techniques were used.

CNN, due to its in-variance property, can classify objects even when visible in different sizes, orientations or different illumination[1]. CNN need a huge number of images for the model to be trained effectively. This helps to increase the performance of the model by generalizing better and thereby reducing overfitting. Most popular data sets for classification and object detection data sets have a few thousand to millions of images. Hence, we can take the small dataset of images and transform the objects to different sizes by zooming in or zooming out, flipping them vertically or horizontally or changing the brightness whatever makes sense for the image/X-ray. This way a rich, diverse data set with variations is generated for training.

For image augmentation Keras **Image DataGenerator** class was used.

Some of the operations done by Image DataGenerator class are[5]:

- Rescaling
- Flipping
- Rotation
- Cropping

Rescaling: Rescaling multiplies the data by the value provided in the parameter (after applying all other transformations).

Flipping [8]: Flips the X-ray images laterally or vertically for increased data and better results

Rotation[7]: Rotates the image by a specified degree.

Cropping: Cropping the X-ray to remove unnecessary parts of the X-ray images.

DATASETS USED

Two different datasets were used for Covid-19 Detection namely:

1. **For Normal Images** – For covid negative patients (Normal Patients) the dataset used was Chest X-Ray Images (Pneumonia) provided by Paul Mooney on Kaggle.

Link:

<https://www.kaggle.com/paultimothymooney/chest-xray-pneumonia>

2. **For Covid Positive Images** – For covid positive patients the dataset used was covid-chestxray-dataset provided by ieee8023 on Github.

Link:

<https://github.com/ieee8023/covid-chestxray-dataset>

Chest X-Ray Images (Pneumonia) Dataset:

This dataset contains 5863 images of 2 categories:

1. Normal Patients
2. Pneumonia Patients

This dataset contains 3 types of data: Training data, Validation data and Testing data. Each of these 3 data contain 2 Folders containing images of Normal patients (Disease Free) and second of those who are suffering from Pneumonia. Healthy patients Chest X-Rays are extracted from this dataset.

Covid-chestxray-dataset Dataset:

This dataset contains images of 11 categories:

1. COVID-19 Patients
2. ARDS Patients
3. SARS Patients
4. Pneumocystis Patients
5. Streptococcus Patients
6. Chlamydomphila Patients
7. E.Coli Patients
8. COVID-19 and ARDS Patients (Both)
9. Klebsiella Patients
10. Legionella Patients
11. No Finding (Normal) Patients

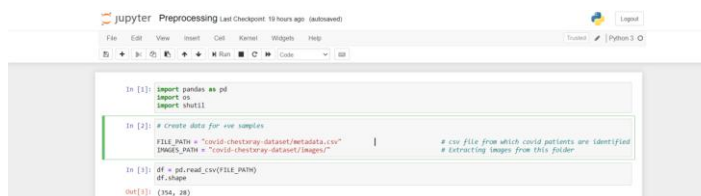
The images are extracted from covid-chestxray-dataset/images/ "metadata.csv" contains the data about the patients and the disease they are suffering from. This file helps in filtering out which patients are affected by Covid-19.

IMPLEMENTATION

The following steps are followed:

- Data Preparation
- Image Preprocessing & Model Creation

1. Data Preparation: In this section the images are sorted into Training and Validation Datasets which would be later used in the next section for training the model.



```
In [1]: import pandas as pd
import os
import shutil

In [2]: # create data for use samples
FILE_PATH = "covid-chestxray-dataset/metadata.csv"
IMAGES_PATH = "covid-chestxray-dataset/images/"

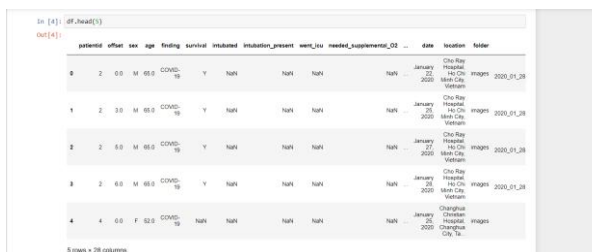
In [3]: df = pd.read_csv(FILE_PATH)
df.shape
Out[3]: (154, 28)
```

Figure 3: Paths of Files used

FILE_PATH points to the “metadata.csv”.
IMAGES_PATH points to the x-ray images.

Now we take an overview of the metadata.csv
Important Columns:

- finding – This column gives the information about the patient’s disease.
- view – Only X-rays of PosteroAnterior (PA) views are selected.
- filename – This column contains the filename



patient_id	offset	sex	age	finding	survival	isolated	isolation_present	ventricle	needed_supplemental_o2	date	location	folder
0	2	0.0	M	60.0	COVID-19	Y	N/A	N/A	N/A	January 22, 2020	Cho Ray Hospital, Hq Cho Ray, Vietnam	images 2020_01_28
1	2	0.0	M	60.0	COVID-19	Y	N/A	N/A	N/A	January 22, 2020	Cho Ray Hospital, Hq Cho Ray, Vietnam	images 2020_01_28
2	2	0.0	M	60.0	COVID-19	Y	N/A	N/A	N/A	January 27, 2020	Cho Ray Hospital, Hq Cho Ray, Vietnam	images 2020_01_28
3	2	0.0	M	60.0	COVID-19	Y	N/A	N/A	N/A	January 22, 2020	Cho Ray Hospital, Hq Cho Ray, Vietnam	images 2020_01_28
4	4	0.0	F	52.0	COVID-19	N/A	N/A	N/A	N/A	January 22, 2020	Changchun Chuanhe, Changchun, China	images 2020_01_28

Figure 4: Overview of metadata.csv

The relevant images are copied to TARGET_DIR for next step.



```
In [3]: cnt = 0
for (i, row) in df.iterrows():
    if row["finding"] == "COVID-19" and row["view"] == "pa":
        filename = row["filename"]
        image_path = os.path.join(SOURCE_PATH, filename)
        target_path = os.path.join(TARGET_DIR, filename)
        shutil.copy2(image_path, target_path)
        print("Copying Covid Images")
        cnt += 1
```

5: Filtering Images

There are only 141 images of patients suffering from only COVID-19 and the X-ray view is PA. Now the same process is repeated for Processing the X-rays of patients not suffering from any disease using the Pneumonia Chest X-Ray Dataset but as there are only 141 images of Covid Patients, less x-ray images of Normal images were chosen so that the distribution remains even in both the classes and model accuracy is not biased. After Both the Processed datasets are created, divide the data into Training and Validation datasets.

(1)

2. Image Preprocessing & Model creation:

Model Creation:



```
# CNN based model in keras
model = Sequential()
model.add(Conv2D(32, kernel_size=(3, 3), activation='relu', input_shape = (224, 224, 3)))
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Dropout(0.25))

model.add(Conv2D(64, (3, 3), activation='relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Dropout(0.25))

model.add(Conv2D(128, (3, 3), activation='relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Dropout(0.25))

model.add(Conv2D(256, (3, 3), activation='relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Dropout(0.25))

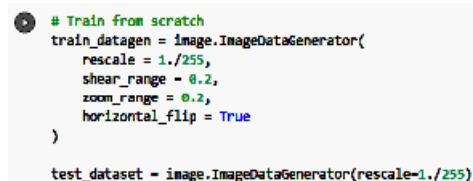
model.add(Flatten())
model.add(Dense(512, activation='relu'))
model.add(Dropout(0.5))
model.add(Dense(1, activation='sigmoid'))

model.compile(loss=keras.losses.binary_crossentropy, optimizer='adam', metrics=['accuracy'])
```

Figure 6: Model Creation

Convolution Neural Networks model was created for training the images and then using the test images for final results.

Image Pre-Processing / Augmentation:



```
# Train from scratch
train_datagen = image.ImageDataGenerator(
    rescale = 1./255,
    shear_range = 0.2,
    zoom_range = 0.2,
    horizontal_flip = True
)

test_datagen = image.ImageDataGenerator(rescale=1./255)
```

Figure 7: Image Pre-Processing

In this step the images were rescaled, zoomed, flipped for more images and better accuracy.

After Pre-Processing 224 images of both classes (Covid, Normal) are chosen for training and 60 images for validation datasets.

Figure

```
[8] train_generator = train_data_gen.flow_from_directory(
    'CovidDataset/train/',
    target_size = (224,224),
    batch_size = 32,
    class_mode = 'binary'
)

test_dataset = image.ImageDataGenerator(rescale=1./255)

Found 224 images belonging to 2 classes.

[9] train_generator.class_indices
{'covid': 0, 'normal': 1}
```

Figure 8: Class Creation

Model Training:After Data filtering, model creation and image augmentation, the model is trained.

```
101 model = model_fit_generator(
102     train_generator,
103     steps_per_epoch = 8,
104     epochs = 10,
105     validation_data = validation_generator,
106     validation_steps = 2
107 )

Epoch 1/10: 100% [-----] 17s 24/step - loss: 0.8206 - accuracy: 0.5218 - val_loss: 0.8011 - val_accuracy: 0.5000
Epoch 2/10: 100% [-----] 9s 16/step - loss: 0.4878 - accuracy: 0.7088 - val_loss: 0.6880 - val_accuracy: 0.5111
Epoch 3/10: 100% [-----] 10s 16/step - loss: 0.4325 - accuracy: 0.6797 - val_loss: 0.5827 - val_accuracy: 0.7000
Epoch 4/10: 100% [-----] 9s 16/step - loss: 0.4369 - accuracy: 0.6867 - val_loss: 0.3410 - val_accuracy: 0.6667
Epoch 5/10: 100% [-----] 9s 16/step - loss: 0.3443 - accuracy: 0.8711 - val_loss: 0.1799 - val_accuracy: 0.6900
Epoch 6/10: 100% [-----] 10s 16/step - loss: 0.3074 - accuracy: 0.8962 - val_loss: 0.1882 - val_accuracy: 0.6833
Epoch 7/10: 100% [-----] 9s 16/step - loss: 0.1791 - accuracy: 0.9054 - val_loss: 0.1874 - val_accuracy: 0.6833
Epoch 8/10: 100% [-----] 9s 16/step - loss: 0.1668 - accuracy: 0.9088 - val_loss: 0.1185 - val_accuracy: 0.7000
Epoch 9/10: 100% [-----] 10s 16/step - loss: 0.1287 - accuracy: 0.9375 - val_loss: 0.0912 - val_accuracy: 0.6833
Epoch 10/10: 100% [-----] 9s 16/step - loss: 0.1486 - accuracy: 0.9613 - val_loss: 0.0956 - val_accuracy: 0.6667
```

Figure 9: Model Training

After training the model shows 96% accuracy on training data and 96.6% accuracy on testing data.

```
[15] model.save_weights("Model.h5")

16 model.evaluate_generator(train_generator) # Loss , Accuracy
[0.25007036328315735, 0.9598214030265808]

17 model.evaluate_generator(validation_generator)
[0.09115566313266754, 0.9666666388511658]
```

Figure 10: Results

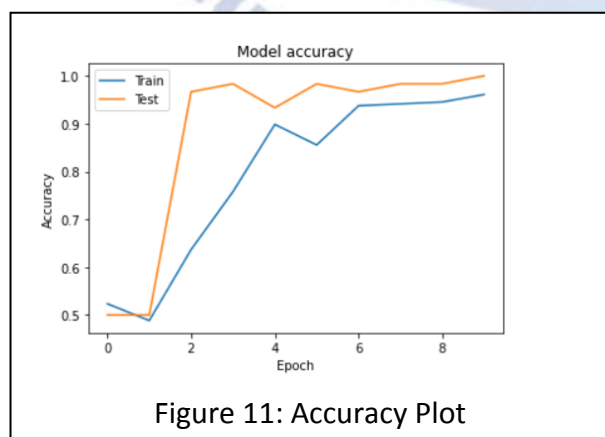


Figure 11: Accuracy Plot

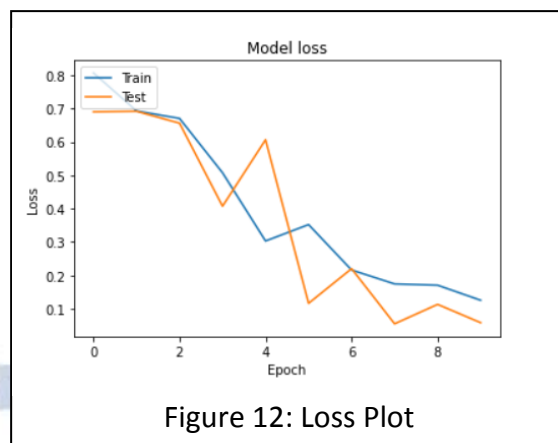


Figure 12: Loss Plot

CONCLUSION

Based on the review performed, it can be concluded that there is a huge scope for machine learning algorithms in disease prediction especially in the case of lung diseases due to the abundance of available data and extensive research done in this field and CNN's are quite powerful to detect COVID-19 patients from their X-Rays.

REFERENCES

- [1] Connor Shorten and Taghi M. Khoshgoftaar(2019), "A survey on Image Data Augmentation for Deep Learning",Springer Conference
- [2] Taban Majeed, Rasber Rashid, Dashti Ali, and Aras Asaad(2020),"Covid-19 detection using CNN transfer learning".
- [3] Keiron O'Shea and Ryan Nash (2015), "An Introduction to Convolutional Neural Networks"
- [4] Rachna Jain, Meenu Gupta, Soham Taneja & D. Jude Hemanth,"Deep learning based detection and analysis of COVID-19 on chest X-ray images",Springer Conference (2020)
- [5] M. Naveenkumar,A. Vadivel, "OpenCV for Computer Vision Applications",Proceedings of National Conference on Big Data and Cloud Computing (NCBDC'15), March 20, 2015
- [6] Agnieszka Mikołajczyk, Michał Grochowski,"Data augmentation for improving deep learning in image classification problem".
- [7] Valery Starovoitov,Dmitry Samal,(1999),"Comparison of rotation algorithms for digital images".
- [8] Alcala, Rony R. Arceo, Zuedmar G. Baterisna, Jonathan N. Morada, Jimmy O. Ramirez, Julius Oliver D,(2017),"Image Processing: Orthogonal Image Rotation and Flipping Using Matrix Operations".

[1]