SIIM-FISABIO-RSNA COVID-19 Detection

Syed Hani Haider **Nikhil Jyoti**

Abstract

As COVID-19 has caused a lot of deaths in the past year, some of the symptoms of COVID-19 are the same as that of bacterial pneumonia: pulmonary infection that shows inflammation and fluid in lungs. Since it looks very similar to bacterial pneumonia, it is very difficult to differentiate between them. Our model will help doctors provide a quick and reliable diagnosis to tell if the person has the virus or not. This will help patients get the right treatment before the virus takes hold. For each test image, we are predicting if the image is: 'Negative', 'Typical', 'Indeterminate', 'Atypical'. This project seems interesting to us because COVID-19 has taken a lot of lives in the past couple of years, and it still is. [3]

Introduction

Due to the increasing threat of COVID-19 after a patient is positive for the virus, in the process of treating it time becomes an essential factor. Our goal was to identify and localize COVID-19 abnormalities on the chest radiographs. Doctors can detect and differentiate between pneumonia and COVID, but radiologists may sometimes not be able to do so. Our model will help to pre-classify radiology reports and hence assist such radiologists to decide faster.

We are going to use the BIMCV-COVID 19 [2] dataset. We obtained this existing dataset from kaggle, but the dataset was originally published by the Medical Imaging Databank of the Valencia Region (BIMCV) in cooperation with The Foundation for the Promotion of Health and Biomedical Research of Valencia Region (FISABIO), and the Regional Ministry of Innovation, Universities, Science and Digital Society (Generalitat Valenciana). The dataset consists of a collection of 6,334 chest scans (Figure 1.0) in DICOM format and 2 CSV files -study_level.csv - which has the study level metadata and image_level - which has the image level metadata. The study level metadata has the information of which class the image belongs to i.e whether it belongs to 'Negative', 'Typical', 'Indeterminate', or, 'Atypical'. The image level metadata gives us more information about the image by providing bounding boxes.

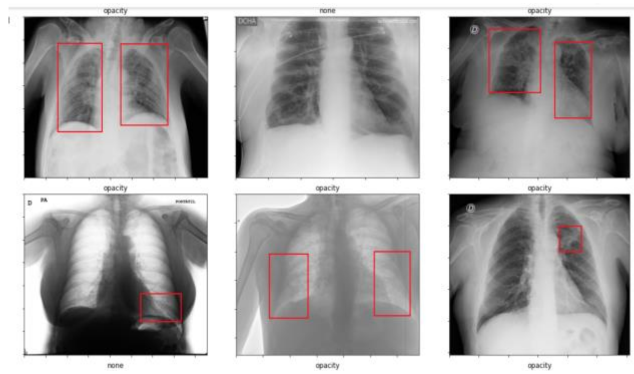


Figure 1.0

Background

Machine learning is a branch of artificial intelligence and computer science which focuses and different data and algorithm to find and predict the accuracy . In the field of data science machine learning is one of the important components. As big data is expanding and growing each day the demand for data scientist is increasing so that they can identify and predict the most relevant business question and answer them subsequently. Majorly the machine learning and algorithms are divided into three important parts which are: a decision process meaning making prediction or classification based on input, an error function: it is used to evaluate the prediction model this is used to do comparison so that we have better accuracy on our model and model optimization process: lastly we will optimize so that the data can be fit better and can also be train better meaning adjusting weight so that it reduces the discrepancy between the known example and model estimated [8]

Before we dive into how the algorithm works one needs to first have some background of machine learning. One should know the following algorithms on how they work to better understand how the image is being detected. The following are some background information one might need to know: 1) Image Classification, 2) Multiclass Classification, 3) Neural networks.

Image classification: It is a process of categorizing and labeling groups of pixels with help of certain specific rules. An image classifier is tasked to find various classes from the image, e.g., the training images given fig1.0 show what are we trying to predict, we are trying to predict the red box if any abnormality is existing. There are three main types of image classification technique: I) unsupervised image classification ii) supervised image classification iii) object-based image analysis. Supervised and unsupervised are the most used approaches.

Unsupervised Classification: If the pixels are grouped into clusters and then they base their properties, after that each cluster with a land covers a class.it is easy to understand the image as it does not need samples for unsupervised classification.

Supervised classification: it uses sample for each land of cover class and then uses theses training sites and applies them to the entire image to find the output.in supervised image classification one need to create training sample for example in our case we need to first create the bounding box for which the algorithm will learn form. One will continue creating the sample until you have created a training sample for each class. and then finally you will run the model to find the correct score which can be done from picking different kinds of classification algorithms.

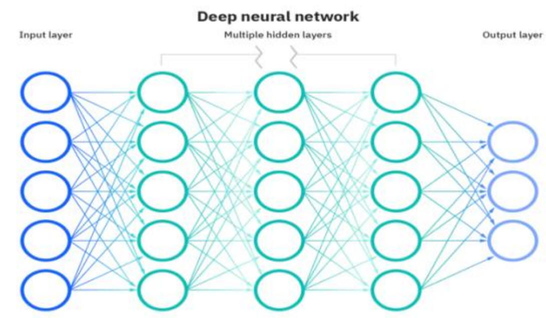
Multiclass classification: Given any kind of data be it image or textual there are problems where we need to classify the image into classes. Our assignments being classified as group of grades such as A, A^{-} or B is one such example of classification. When we are required to classify an image into a class among multiple classes it is called Multi class classification.[2]

Neural Network: It is the heart of machine learning algorithms. Neural networks majorly rely on the training data so that as it learns more the accuracy would increase. This means that as it is learning the algorithms fine tune itself and the accuracy increases. Neural network is a power tool in the field of computer science and artificial intelligence as it allows the cluster data with a higher pace. The model works in a way that it makes each node as each linear regression model which is composed input data, weights, a bias(threshold) and an output layer [5]

Related Work

Image classification has been widely used in a plethora of problems. They help and improve human life by completing tasks quickly and more efficiently which are difficult for us to solve directly. Due to the urgency to tackle the virus many attempts have been made to diagnose/detect/classify covid reports. The current methods however use Neural networks to classify these images. Many methods like Support Vector Machines and alterations of Convolutional Neural Networks have been used for this task. We are using a better model called Efficient Net to solve the problem.

There are a long list of new models/ algorithms being developed and in use for such tasks. One such Yolo stands for you only look once is object detection where it



uses a grid system. Each object is responsible for itself. In today time it is one of the best models for image detection. It uses a deep neural network which is very easy to retain the network on a custom data set. It is one of the best one there to find out if an image is covid or not.

Project Description

The aim of the project was to develop a model with high accuracy to detect the classes from patient chest scans. For this we were given this study level csv consisting of 5 columns: id - the study id for this specific row, and binary indicators: Negative for Pneumonia, Typical Appearance, Indeterminate Appearance, and Atypical Appearance. The image level csv consists of id - unique image id, boxes - dimensions of the bounding boxes if any, label: the correct prediction label, and studyInstanceUID - unique identifier for the specific study which gave us the bounding boxes and the labels.

For the scope of our project, we decided to predict the classes for the images. Initially, we developed an algorithm to compute the opacity count for the given images. The opacity count lets us understand the data better and it gives us hints of classification way before we start building our model. Opacity count tells us how many bounding boxes the image has and hence also tells indirectly which class it may belong to. It is not entirely sufficient to say that an opacity count of 0 means negative for pneumonia, but it does give us a high probability of this being true. Eg shown in figure 2.0.

```
Opacity Count = 0
----------------------------
Negative for Pneumonia        1736
Typical Appearance             153
Indeterminate Appearance        59
Atypical Appearance             92
```

Figure 2.0 - Image statistics for opacity count 0

We can clearly see that even if the opacity count is 0 there are 153 images which are typical appearance. Since we have these many outliers (Figure 3.0) we need to understand our data better.

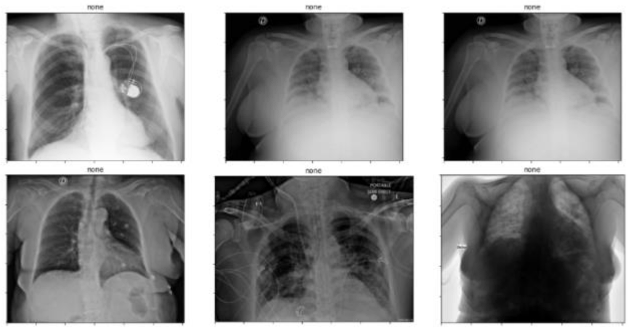


Figure 3.0 - Outliers for typical appearance but opacity count 0

To make the learning process of the model faster and more efficient all the images were resized to 150x150 and the pixel values for each image was scaled to an uniform range of 1~255. This resulted in more uniform images which are in the required format for modelling.

Initially, we built a convolutional neural network model to classify our images. The convolutional neural network is a type of neural network model which allows us to extract higher representations for the image content. Unlike the classical image recognition where you define the image features yourself, CNN takes the image's raw pixel data, trains the model, then extracts the features automatically for better classification.

For our first model we started to build a CNN and fine-tuning it to check with multiple permutations which model performs better and found out that the model given in Figure 4.0 performs the best among our set of CNN models.

Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 254, 254, 64)	640
max_pooling2d (MaxPooling2D)	(None, 127, 127, 64)	0
conv2d_1 (Conv2D)	(None, 125, 125, 64)	36928
max_pooling2d_1 (MaxPooling2	(None, 62, 62, 64)	0
conv2d_2 (Conv2D)	(None, 60, 60, 128)	73856
max_pooling2d_2 (MaxPooling2	(None, 30, 30, 128)	0
conv2d_3 (Conv2D)	(None, 28, 28, 128)	147584
max_pooling2d_3 (MaxPooling2	(None, 14, 14, 128)	0
flatten (Flatten)	(None, 25088)	0
dropout (Dropout)	(None, 25088)	0
dense (Dense)	(None, 128)	3211392
dense_1 (Dense)	(None, 4)	516

Figure 4.0 - CNN model summary

CNNs(Figure 5.0) are regularized versions of multilayer perceptrons. Multilayer perceptrons usually mean fully connected networks, that is, each neuron in one layer is connected to all neurons in the next layer. A CNN is basically a multilayer perceptron. A typical convolutional neural network consists of some combination of the following layers: convolutional layers, pooling layers and dense layers. It consists of activation functions such as ReLU, sigmoid, tanh, etc. We decided to use ReLU in our model for the first Dense Layer and softmax for the last dense layer. ReLU stands for Rectified Linear Unit and the major benefits of ReLUs are sparsity and a reduced likelihood of vanishing gradient. Generally, we use softmax activation instead of sigmoid with the cross-entropy loss because softmax activation distributes the probability throughout each output node.

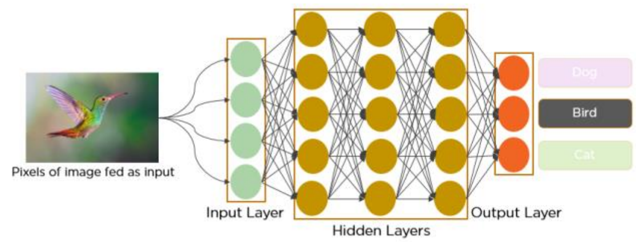


Figure 5.0 - an example of CNN model[6]

We used max pooling layers to downsample the input along its height and width. It does this by taking the maximum value over a window which is in turn defined by the pool size for every channel of the input. We added a flatten layer since it reshapes the tensor to make the shape same as the number of elements in the tensor without the batch dimension. A dropout layer was added which helps in preventing overfitting by randomly setting input units to 0. The most important layer, the dense layer, is nothing but a fully connected neural network layer. It performs the below operation

$$output = activation(dot(input, kernel) + bias)$$

and the output is then fed forward to the next layer which in turn classifies the image into one of the four classes.

On observing the basic CNN model we thought of doing better and after researching we decided to use the EfficientNet Model. In any CNN model there are three scaling dimensions namely depth, width and resolution. Depth corresponds to how deep the network is i.e. the number of layers in the model. Width on the other hand measures how wide the network is i.e. the number of channels in a layer. Resolution is basically the resolution of the image that is being passed to the input layer of CNN.

Increasing the depth does allow the network to learn more features but this tends to push the network towards vanishing gradients and hence become difficult to train. When we scale the width such as it is done in models like ResNet, increasing width prevents from learning complex features and if we increase/scale the resolution of the image it does help in learning more fine grained data but it alone cannot increase the accuracy gains in a substantial way.

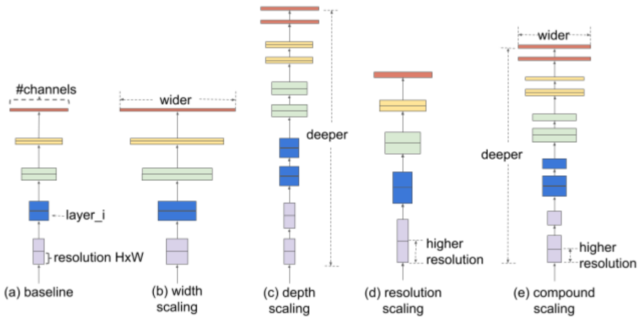


Figure 6.0 - Compound scaling[7]

To enable scaling of our model and improve its accuracy at the same time we need to find a way to scale all the 3 features i.e., the width, height, and resolution in a balanced and harmonic way. This is where Efficient Net comes in. The efficient net model uses compound scaling to uniformly scale the network. The technique used to do this is as follows:

$$\begin{aligned} depth\ d\ &= \alpha^{\phi} \\ width\ w\ &= \beta^{\phi} \\ resolution\ r\ &= \gamma^{\phi} \\ s.t.\ \alpha . \beta^{2} . \gamma^{2}\ &\approx\ 2 \\ where\ \alpha \geq 1, \beta &\geq 1\ , \gamma\ \geq 1 \end{aligned}$$

This scaling method helps the model learn more information from the input and also enables the computational floating point operations (FLOPS) to be less than 2^ϕ.

Now, we have all the components to make our classification prepared. The steps we followed to classify the images: After loading our dataset of images and the 2 csv files. We first merged the two files together, created a new feature Opacity Count, resized the images using openCV, fed these resized images along with their labels to our model and then compared the results.

Empirical Results

The task of image classification requires knowledge of vast algorithms if we need to do it perfectly. The task of object detection upon this is even more difficult. Initially we planned to classify the images and then further detect the anomalies using object detection. We learned about YOLO for this task but were not able to implement it in time. After researching and learning about new and breakthrough models for the classification task we decided to compare results of different models and have listed a few of them below.

Convolutional Neural Network model

This model uses the conventional neural network learning process with no compound scaling and hence took a hit in predicting the right classes with a low accuracy score as we can see from the table (1.0)

EfficientNet B0

The efficient net model (Figure 7.0) was able to perform much better than the given CNN model since this efficient model performs compound scaling to learn the data.

```
Model: "sequential_2"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
efficientnetb0 (Functional)  (None, 7, 7, 1280)        4049571
_________________________________________________________________
flatten_2 (Flatten)          (None, 62720)             0
_________________________________________________________________
dropout_2 (Dropout)          (None, 62720)             0
_________________________________________________________________
dense_4 (Dense)              (None, 256)               16056576
_________________________________________________________________
dense_5 (Dense)              (None, 4)                 1028
=================================================================
Total params: 20,107,175
Trainable params: 16,057,604
```

Figure 7.0 - Efficient Net B0 summary

EfficientNet B7

The efficient net model (Figure 8.0) performed the best as we can see from table 1.0. This is because the compound scaling gradually increases from B0 to B7 however given our dataset the improvement was not as significant as that of CNN to B7.

Model	Accuracy%
1. CNN	47.5
2. EfficientNetB0	60.0%
3. EfficientNetB3	60.5%
4. EfficientNetB5	61.0%
5. EfficientNetB7	62.0%

Table 1.0 - Accuracy of all models

Based on our experiment and analysis it is clear that the efficient net outperforms the CNN model. We also compared these results with other models such as SVM but acquired even low scores in comparison.

Conclusion / Future directions

Undertaking this project in such unprecedented times was a challenging task but the realistic nature of the project kept us going. We learned that theoretical concepts if understood well have enormous power to complete complex tasks. Reading and learning about the new technologies such as EfficientNet, ResNet, MobileNet, openCV, YOLO not only open up a world of new information for us but compel us to understand our basics too. We feel that we have a far greeted understanding of the concepts we learned in class due to the project.

Given more time we would extend our project to the object detection task and would have loved to actually participate in the kaggle competition in a more competitive way. Our model does perform greatly as one of the top rankers but only in the classification phase.

Our advice to the future students of DS 5220 would be to make sure you try and understand the basics of everything you learn in class since all the models or state-of-the-art models are based on these concepts. Another advice would be not shy away from a competition and try to extend yourselves into learning.

References

[1]SIIM-FISABIO-RSNA COVID-19 Detection | Kaggle." SIIM Machine Learning
Committee, 2021, www.kaggle.com/c/siim-covid19-detection.

[2]ME, P. G. (n.d.). Image classification. Image Classification - an overview | ScienceDirect Topics. https://www.sciencedirect.com/topics/engineering/image-classification.

[3]Korstanje, J. (2020, September 28). Yolo v5 object detection tutorial. Medium. https://towardsdatascience.com/yolo-v5-object-detection-tutorial-2e607b9013ef.

[4]Clancey, W. J. 1984. Classification Problem Solving. In *Proceedings of the Fourth National Conference on Artificial Intelligence*, 49-54. Menlo Park, Calif.: AAAI Press.

[5]By: IBM Cloud Education. (n.d.). What are neural networks? IBM. https://www.ibm.com/cloud/learn/neural-networks.

[6]https://www.analyticsvidhya.com/blog/2021/05/convolutional-neural-networks-cnn/

[7]https://ai.googleblog.com/2019/05/efficientnet-improving-accuracy-and.html

[8] IBM Cloud Education. (n.d.). Machine Learning

IBM.https://www.ibm.com/cloud/learn/machine-learning