Scalable Regression Model for Pulmonary Fibrosis†

Anuraj Singh
ABV-Indian Institute of Information Technology and Management
Gwalior, INDIA

Vaibhav Garg
ABV-Indian Institute of Information Technology and Management
Gwalior, INDIA

Praveen Kumar Sahu
ABV-Indian Institute of Information Technology and Management
Gwalior, INDIA

***Abstract*—By this work we try to build model which helps us to detect pulmonary fibrosis at an early stage. This model should be able to tell severity of the case using the Computed Tomography scan of the patient. The aim of this work is to also develop a self scaling model advantage of that would be that it can easily generalise for the other similar diseases also which require CT scan of lungs to prediction. By integrating this model with CT-scan machine itself it should be able to quickly tell the disease which is most probable if any and it's severity if possible. To make such auto-scaling model we will be using efficient-nets as the core of the model.**

***Index Terms*—Pulmonary Fibrosis, Computed Tomography scan, EfficientNets, Dropout Layers, Pooling Layer, Convolutional Neural Network**

I. Introduction

Diagnosis of lung pulmonary fibrosis is a complicated process. It begins with the clinical inspections followed by series of tests and finally a deep analysis is done. These series of test may contain different Imaging Tests like CT scan and Chest X-Rays, Lung Function tests and different types of Biopsy. Only way to treat pulmonary fibrosis at present is lung transplantation which is itself is risky as well as costly. Through non-surgical approaches we can only either keep the case stable as it is now or elongate the time till it get out of control but can not make the patient recover from the disease. But if we are able to identify it at an early stage biologist believe that it may be possible to treat Pulmonary Fibrosis using medicines or at worse could help in keeping situation stable helping in avoiding the surgery. After the widespread of Novel Corona-Virus the chances of pulmonary fibrosis has increased as it is predicted that Pulmonary Fibrosis can be sequel of COVID-19. Due to which detecting Pulmonary Fibrosis is important and we need to prepare for it.
This project aims to develop a model which can help in predicting the Pulmonary Fibrosis and its severity. We use EfficientNets as the core for this model. Efficientnet have a advantage over traditional neural-networks that they can auto scale and adjust their depth, width and resolution itself. Efficientnets is considered to be a step towards the Auto-ML.

II. Background

Pulmonary fibrosis was considered to be a rare disease. But now after the spread of the COVID-19 it is predicted that as an after effect of COVID-19 people may suffer from pulmonary lung fibrosis. Since there is no treatment for pulmonary fibrosis it is important to identify it at the early stage so that required precautions, medications and procedures can be done to keep that stable. Since the case study takes time to detect pulmonary fibrosis as well as is expensive which makes it important to find other way to identify it which are less time-taking, cheaper and reliable. To solve this problem, we propose a model which could take an input from users in form of a CT-Scan and the model should be able to tell the severity at an early stage so that the required steps can be taken. We aim to provide a cheaper and faster solution.

In 2019, EfficientNets was first proposed by Mingxing Tan, Quoc Le. They also compared the performance of the EfficientNet with the conventional CNNs. Rather than comparing by accuracy on a single example, they compared it in multiple cases. Mingxing and Quoc were working with Google's research team [1]. After the spread of the novel Corona Virus, there have been many attempts to stop developing ways to stop its spread and recognize it more accurately. Such attempts were based on applying the conventional CNN on the X-Ray images but which have the disadvantage of being rigid in their architecture [2][3]. CNN has been used in the medical field vary widely for detecting different types of diseases such as oral diseases[4][5][6]. In the future, we can also use EfficientNets on this disease to get even better results. Linh T. Duong et. Al tried used Efficient for fruit recognition it was one of the first use of EfficientNets after its discovery [7]. EfficientNets have also been used to apply on an image of the lesion to detect skin cancer and on X-Ray images to detect breast cancer. Pan Zhang, Ling Yang, and Daoliang Li attempted to detect greenhouse cucumber disease [8][9][10]. Pulmonary fibrosis is a rare disease which does not have any clinical treatment, so it is necessary to identify it at an early stage so that it could be controlled from getting critical [11][12][13].

Previous works focused on Pneumonia and other common diseases but not on pulmonary fibrosis due to it being a rare disease. Also, previously proposed models are more rigid and robust require manual effort to scale.
There are few algorithms applied on CT-Scan of Lungs to detect the Novel Corona-Virus and few algorithms which uses EfficientNets to predict other diseases. But there is no

† This work is supported by Mathematical Research Impact Centric Support scheme (MTR/2020/000477), Science and Engineering Research Board (SERB), India.

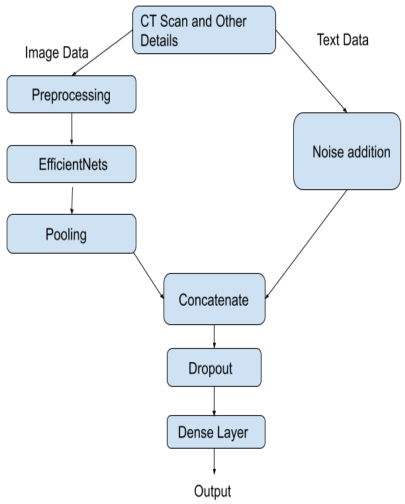


Fig. 1. Flowchart for the proposed model

algorithm which uses EfficientNets to predict the pulmonary fibrosis. Since Pulmonary fibrosis was considered as rare disease not much emphasis was given to it. Also our model can be easily modified to be used to predict Pneumonia and COVID-19 from CT scan. But since there are other tests which are more successful to predict COVID-19 and Pneumonia so it is not required currently.

III. Methodology

Fig 1 depicts the flow of the model. The system begins by taking input an image of CT-scan and details like smoking status, age and gender of the patient as input. Then image data and other details are process separately till they are concatenated into single tensor. Image is first passed to pre-processing step after which it it goes to EfficientNet and then it goes to pooling layer. And in the text details noise is added. Output after pooling and noise addition in image data and Text details is concatenated. Concatenated output is passed to dropout layer after which it is passed to dense layer. Then the dense layer gives the final output.

A. Pre-Processing

This is the first step of the process this step may vary according to the dataset being used. First step in pre-processing is to remove the salt and pepper noise for which we used median filter. In application if we are directly integrating it with the test machines performing scans we will not need this step. As in that case the salt and pepper noise will not be much significant. In this step, next size of the images must be reduced so that the neural network becomes more effective. This step seems to be simple step but the challenge in this step is to reduce the pixels of the image without losing the relevant data. We propose to keep the pixels between 400 to 800 in both height and width. So one of the best way for the same is to take the central pixels. We have taken 512 by 512 pixels to train our model.

B. Noise Addition To text data

During the time of training it is a good practice to add noise to data it avoids over-fitting and it makes sure that neural networks do not memorise the examples. We have decided to use Gaussian Noise to the data set. We have used standard deviation as 0.1 for adding Gaussian noise after converting the combination of 3 fields Age, Gender and Smoking status after converting them to input tensor.

C. Pooling Layer

Convolutional Neural Networks are effective when they are applied to images. CNN tend to extract the low-level features of the images due to deep layering. Convolutional neural networks give a very high importance to the location of these features so it is important to add a pooling layer to them to avoid this. Adding the pooling layer hides the low level features giving more importance to high level features and the process is termed as down sampling.

D. Dropout layer

Dropout layer is important as it helps in solving the problem that every part of the layer learns a similar feature or gets trained to the same feature. This problem leads to over-fitting. To solve this problem there are two options one is to train to different models. Then take the weighted mean of output from all models to get the final output. This method comes under the category of Ensemble Methods. The major drawback of this method is very high computational requirement. In neural networks using dropout layer is a better option. By dropout layer we randomly drop the nodes of the previous layer. This dropout layer helps in regularization. Adding dropout layer becomes much more important when it is expected that the data available while building the model and the data while real-time application of model is expected to highly vary.

E. Dense-Layer Neural Network

Dense layer in a neural network are simple to understand these are the layers which are completely connected to each other. This means each neuron in previous layer are connected to each neuron in this layer. We are using three layers dense layers after the concatenation of the outputs from efficientNet model and the patient text data which we feel might be important for the prediction. The number of neuron in layers in dense layer neural network are 100, 100 and 3. And the input size of the tensor to dense neural network would be 9 and the output would be 1. Number of trainable parameter in each layer can be calculated by (IS * OS) + OS. Where the first term corresponds to the weight of each input for the layer and the second term corresponds to the bias.

Location	Input Size	Output Size	Number of Parameter
Between Input and 1st layer	9	100	1000
Between 1st and 2nd layer	100	100	10100
Between 2nd and 3rd layer	100	3	303
Between 3rd and Output Layer	3	1	4
Total trainable Parameters			11407

TABLE I
DESCRIPTION OF TRAINABLE PARAMETERS IN DENSE LAYER NEURAL NETWORK

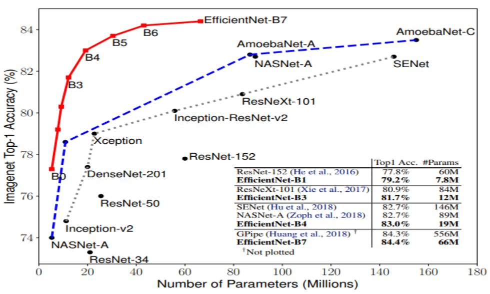


Fig. 2. Graph taken from the work by Mingxing Tan and Quoc Le[1]

F. EfficientNets

EfficientNets comes under the category of Convolutional Neural Networks. An EfficientNet consists of different CNN layers and the speciality of efficient nets is that it scales itself to give better outputs by changing the number of these layers used, complexity of the layers used and by changing order in which these layers used dynamically. This property of EfficientNets makes them more effective and practical when compared to traditional Neural Networks. EfficientNets are also considered as automatic scaling mechanism rather than a new type of model. Generally, EfficientNets are expected to give better result than traditional CNN Models in most cases. Two important factors affecting the performance of Efficient-Nets is the base model or architecture which is being used and the scaling up mechanism being used. This is the reason why same EfficientNet Model or with little modifications can be trained for similar problems. Example is like a similar code be reused with little modification to predict pulmonary fibrosis and pneumonia. In our model EfficientNet is being used to extract the features from images.

Fig 2 illustrates the comparison between the common convolutional neural networks with different types of EfficientNets. The 8 types of EfficientNet B0 to B7 all differ in terms of how they scale up. From the figure we can clearly see that not only EfficientNets are providing better accuracy but also with the less number of parameters. Fig 3 shows the comparison between how the EfficientNet scales as compared to ordinary CNNs. So this is due to compound scaling used by EfficientNet they are able to give alot better results than normal CNNs.An EfficientNet B5 can give accuracy as good as the other complex neural network with limited size. That is why we have decided to use B5 model for our problem. But while implementation if model doesn't achieve the required

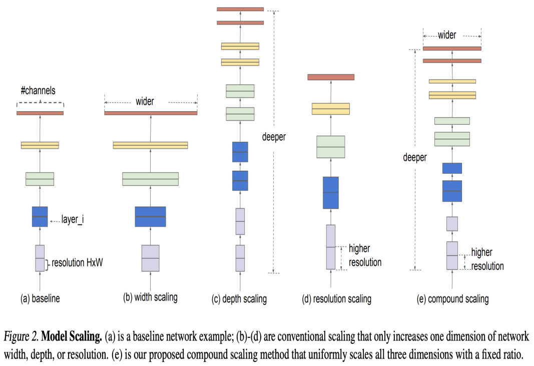


Fig. 3. Difference in scaling method of normal CNN and EfficientNet[1]

accuracy it can be desirable to switch to B6 or B7. Both Fig 1 and Fig 2 were taken from the first paper on Efficient Nets by Mingxing Tan, Quoc Le..

G. Model Parameters

- EfficientNet B5 is chosen and used to make this model. Reason behind preferring B5 over other models was that it is much more effective than the B0 model. Whereas even though choosing B7 would have provided even better results but would have been much more complex. The amount of parameters in B7 are huge in comparison to the increase in accuracy provide. Considering these reasons we have choosen B5 as the model. But when used in application we can switch to B7 for better results.
- The overall available data was divided into training and testing data in the ratio of 4:1.
- Number of Epochs for which model ran was 800. This number was decided keeping the size of the data set and model in mind. We also observed that the results started to saturate after it.
- Global Average Pooling layer is used as a pooling layer.
- For training the model K-Fold technique is used with the value of K as 5.

IV. RESULT AND DISCUSSION

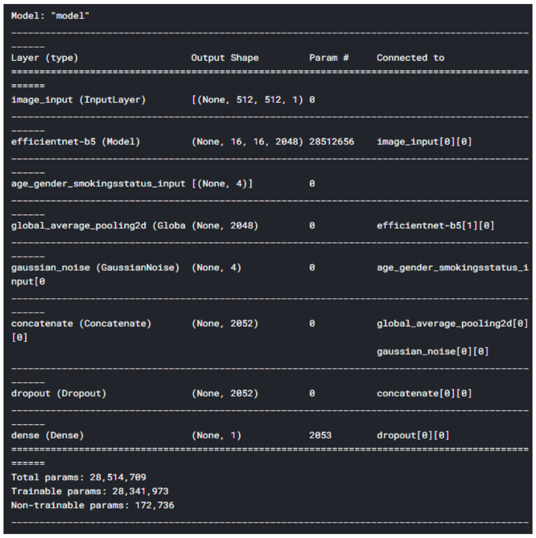
In this Section we will discuss the results obtained by applying discussed model. We will also the discuss the architecture of the model. For measuring the performance of the model various metrics are calculated.

A. Result and Its Evaluation

The final architecture of the model is shown in the Figure 4. Total no of parameters in B5 model we used are around 28.5 million. Also from this figure we can also visualise the flowchart of the model.

Figure 5 shows the comparison of actual value and the predicted value for randomly chosen 100 input from the test data set. We can see the model is able to detect the trend to a large extend.

The distribution of the difference between the predicted and the actual value can be summarized as. Root Mean Square



```
Model: "model"
______________________________________________________________________________
Layer (type)                    Output Shape         Param #     Connected to
==============================================================================
image_input (InputLayer)        [(None, 512, 512, 1) 0
______________________________________________________________________________
efficientnet-b5 (Model)         (None, 16, 16, 2048) 28512656    image_input[0][0]
______________________________________________________________________________
age_gender_smokingsstatus_input [(None, 4)]          0
______________________________________________________________________________
global_average_pooling2d (Globa (None, 2048)         0           efficientnet-b5[1][0]
______________________________________________________________________________
gaussian_noise (GaussianNoise)  (None, 4)            0           age_gender_smokingsstatus_i
nput[0
______________________________________________________________________________
concatenate (Concatenate)       (None, 2052)         0           global_average_pooling2d[0]
[0]
                                                                 gaussian_noise[0][0]
______________________________________________________________________________
dropout (Dropout)               (None, 2052)         0           concatenate[0][0]
______________________________________________________________________________
dense (Dense)                   (None, 1)            2053        dropout[0][0]
==============================================================================
Total params: 28,514,709
Trainable params: 28,341,973
Non-trainable params: 172,736
______________________________________________________________________________
```

Fig. 4. Final Architecture of the model after integration

Error for the model was about 244 for the predicted values of FVC and actual values of FVC. Mean of the Absolute difference between the predicted values and the actual values of FVC was about 168. More than 70 percent of the predicted values have less than 200 absolute difference in the FVC's predicted and actual. Max absolute difference in FVC's predicted and actual is 612.42. Only 8 data-points were there which had difference in FVC of more than 500 out of 700 examples which were used for testing. More than 35 percent of the predicted values and the actual value were below 60 which can be considered as highly desirable or accurate.

In the figure 6 shows the distribution of absolute values of the difference between the predicted and the actual values. From the graph we can see that the chances of higher values are decreasing drastically.

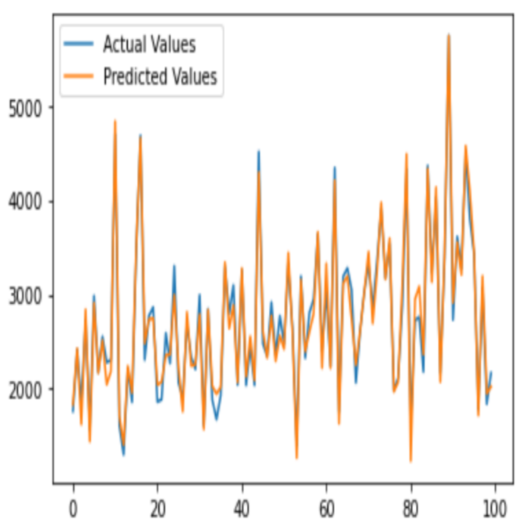


Fig. 5. Predicted and actual values for randomly chosen 100 data points

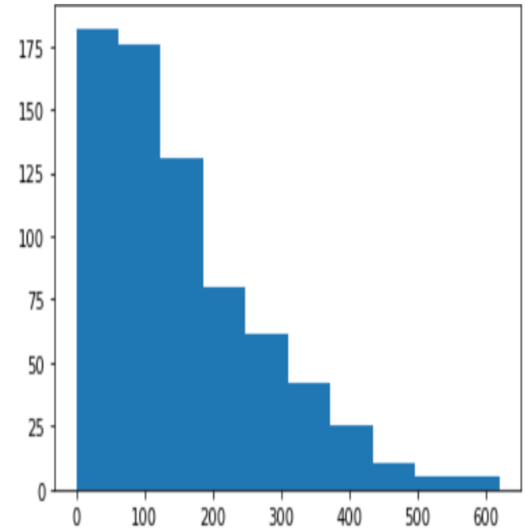


Fig. 6. Histogram on Absolute difference of predicted and actual values

B. *Conclusion*

The developed model was able to give satisfactory results. RMSE of 244 in where the range of highest value and lowest value is more than 3500 is highly desirable. We were able to make a model which is scalable and adaptable for future changes. With a very little modification the B5 model which is being used can be replaced by more complex efficientNet or simpler one as per requirements.

With few changes this or similar model can be extended for other similar diseases. Integrating this model with the CT-Scan machine itself can help us detect the pulmonary fibrosis at an early stage.

EfficientNets scaling mechanism is designed to scale up well when it is the only model in the system. But in our case since we are using the dense layer Neural-Network so this assumption is not completely true. In future development, the scaling mechanism can be improved to consider overall model architecture to scale up. Also as a future work a single model can be designed to work for multiple disease rather than having dedicated models for each diseases.

References

[1] Tan, M., & Le, Q. (2019, May). Efficientnet: Rethinking model scaling for convolutional neural networks. In International Conference on Machine Learning (pp. 6105-6114). PMLR.

[2] Marques, Gonçalo, Deevyankar Agarwal, and Isabel de la Torre Díez. "Automated medical diagnosis of COVID-19 through EfficientNet convolutional neural network." Applied Soft Computing 96 (2020): 106691.

[3] Chowdhury, Nihad Karim, et al. "ECOVNet: An Ensemble of Deep Convolutional Neural Networks Based on EfficientNet to Detect COVID-19 From Chest X-rays." arXiv preprint arXiv:2009.11850 (2020).

[4] Anantharaman, Rajaram, Matthew Velazquez, and Yugyung Lee. "Utilizing mask R-CNN for detection and segmentation of oral diseases." 2018 IEEE International Conference on Bioinformatics and Biomedicine (BIBM). IEEE, 2018.

[5] Hattikatti, Pratiksha. "Texture based interstitial lung disease detection using convolutional neural network." 2017 International Conference on Big Data, IoT and Data Science (BID). IEEE, 2017.

[6] Varshni, Dimpy, et al. ”Pneumonia detection using CNN based feature extraction.” 2019 IEEE International Conference on Electrical, Computer and Communication Technologies (ICECCT). IEEE, 2019.
[7] Duong, Linh T., et al. ”Automated fruit recognition using EfficientNet and MixNet.” Computers and Electronics in Agriculture 171 (2020): 105326.
[8] Miglani, Vandana, and M. P. S. Bhatia. ”Skin Lesion Classification: A Transfer Learning Approach Using EfficientNets.” International Conference on Advanced Machine Learning Technologies and Applications. Springer, Singapore, 2020.
[9] Wang, Jun, et al. ”Boosted EfficientNet: Detection of Lymph Node Metastases in Breast Cancer Using Convolutional Neural Networks.” Cancers 13.4 (2021): 661.
[10] Zhang, Pan, Ling Yang, and Daoliang Li. ”EfficientNet-B4-Ranger: A novel method for greenhouse cucumber disease recognition under natural complex environment.” Computers and Electronics in Agriculture 176 (2020): 105652.
[11] Lederer, David J., and Fernando J. Martinez. ”Idiopathic pulmonary fibrosis.” New England Journal of Medicine 378.19 (2018): 1811-1823.
[12] Gross, Thomas J., and Gary W. Hunninghake. ”Idiopathic pulmonary fibrosis.” New England Journal of Medicine 345.7 (2001): 517-525.
[13] Thannickal, V. J., Toews, G. B., White, E. S., Lynch Iii, J. P., & Martinez, F. J. (2004). Mechanisms of pulmonary fibrosis. Annu. Rev. Med., 55, 395-417.
[14] K. Pal and B. V. Patel, ”Data Classification with k-fold Cross Validation and Holdout Accuracy Estimation Methods with 5 Different Machine Learning Techniques,” 2020 Fourth International Conference on Computing Methodologies and Communication (ICCMC), 2020, pp. 83-87, doi: 10.1109/ICCMC48092.2020.ICCMC-00016.
[15] T. Wong and P. Yeh, ”Reliable Accuracy Estimates from k-Fold Cross Validation,” in IEEE Transactions on Knowledge and Data Engineering, vol. 32, no. 8, pp. 1586-1594, 1 Aug. 2020, doi: 10.1109/TKDE.2019.2912815.