FIBROSIS PREDICTION SYSTEM

TEAM MEMBER

A0215513R ZHANG JIUYUN

A0186896B MA ZEYU

A0215300A JIANG HAO

MASTER OF TECHNOLOGY PROJECT REPORT

Hao Jiang[1,*], Zhang JiuYun[1,*], Ma ZeYu[1,*]

National University of Singapore

Contents

[*]Corresponding author.
Email addresses: `haojiang@u.nus.edu` (Hao Jiang), `e0535603@u.nus.edu` (Zhang JiuYun), `haojiang@u.nus.edu` (Ma ZeYu)

1. INTRODUCTION

Pulmonary fibrosis, a disorder with no known cause and no known cure, is caused by scarring of the lungs. The most tricky part of this disease lies in the fact that there is no effective treatment right now. However, considering the wide coverage of the pandemic-Corona Virus Disease 2019 (COVID-19), pulmonary fibrosis could bring more harm than we expect.

1.1. Project Background

There have been research showing that nearly 40%[1] of patients who are diagnosed with COVID-19 may suffer from pulmonary fibrosis. This could be a large number considering the wide spread of COVID-19 (more than 4 million patients according to WHO). Moreover, other noticeable diseases including SARS (SARS-CoV), MERS (Middle East Respiratory Syndrome) can also cause a varying degrees of Pulmonary fibrosis symptom. According to statistics (reference), there are around 5 million people suffering from pulmonary fibrosis while their symptom can vary dramatically from weeks sudden decline to years stability, which will cause an extreme anxiety to the patients. However, although pulmonary fibrosis is one of the most common complications of COVID-19, patients may be difficult to afford the expensive medical fee and the doctors cannot guarantee the cure of patients, neither. In recent years, more and more people have suffered from pulmonary fibrosis due to the change of environment. The Figure.1 shows the increasing number of patients in Sweden.

In order to provide a solid reference for treatment trail design and help patients understand their severity of lung decline better, we build a free web application to predict a patient's severity of decline in lung function based on a CT scan of their lungs. Specifically, our system uses only the patient's CT scan and basic information as input and predict the trend of patients' forced vital capacity (FVC) (i.e. the volume of air exhaled in one breath) within 140 weeks. Besides, our project will offer patients proper guidance based on the degree of their disease. If successful, patients and their families would better understand their prognosis when they are first diagnosed with this incurable lung disease. Improved detection would also positively impact treatment trial design and accelerate the clinical development of novel treatments.

1.2. Data Resource

Our data is mainly comes from the Open Source Imaging Consortium (OSIC) which is designed to enable rapid advances in the fight against Idiopathic Pul-

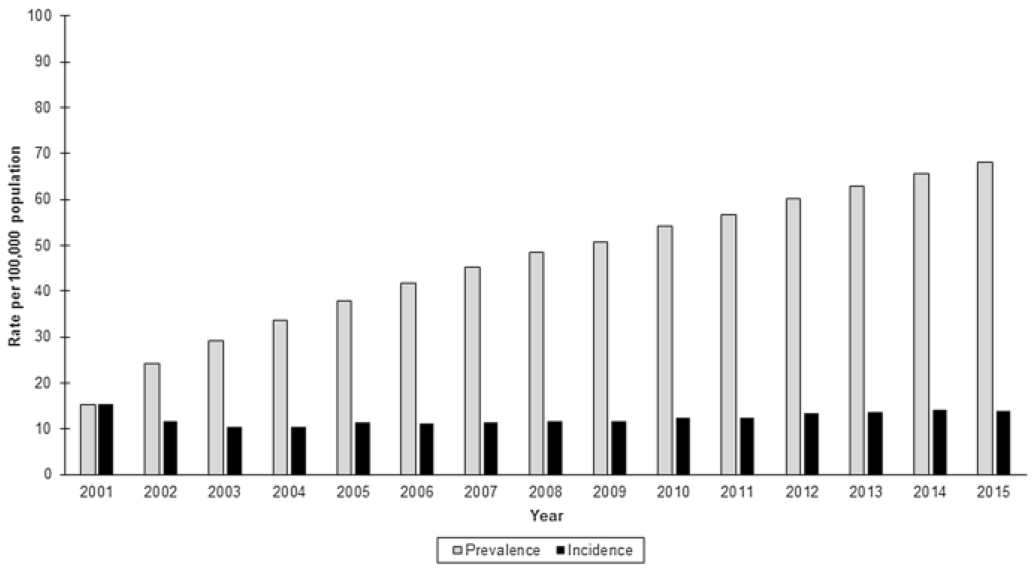


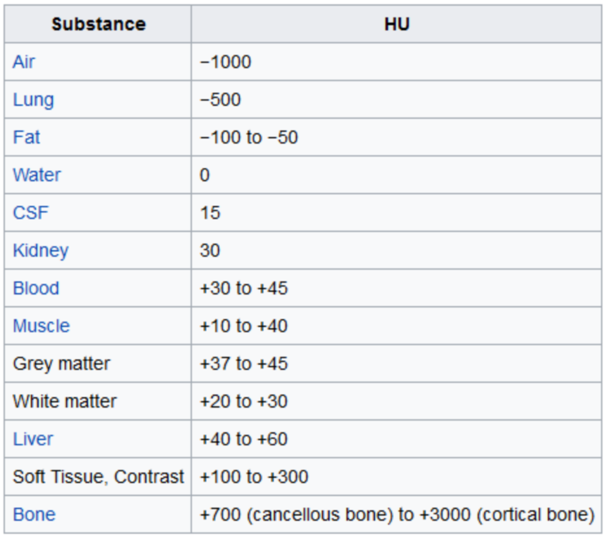
Figure 1: The trend of pulmonary fibrosis in Sweden

monary Fibrosis. Our training data contains 176 real-world patients' basic information, CT scans and corresponding examination data.

1.2.1. CT Scan

As a modern method of detecting human's inner organs, CT scan will not just take one picture of the organ but take multiple pictures at different cross-sections at the same time, and then corresponding software program will combine these pictures into a complete three-dimensional image and restore the internal structure of the scanned object. Therefore, for each patient there will be one folder containing their 3D scan which is divided into different number of 2D images (refers to a slice at different section).

According to the rules in medicine, every 2D slice of the CT scan should be in the format of Digital Imaging and Communications in Medicine (DICOM). Besides, crucial information should be stored in the DICOM files including the patients' information, scan information and the pixel information. Among these information, the most important parts are the pixel arrays which provide the gray scale value at every pixel. Besides the pixel arrays, there are two more crucial attributes called RescaleIntercept and RescaleSlope which can used to calculate the Hounsfield Unit (HU) of the 2D image. The Hounsfield unit is a quantitative scale for describing different tissues' radio density in the CT scan, which will be useful in extracting useful features from the image. For example, the lung tissue has a



Substance	HU
Air	−1000
Lung	−500
Fat	−100 to −50
Water	0
CSF	15
Kidney	30
Blood	+30 to +45
Muscle	+10 to +40
Grey matter	+37 to +45
White matter	+20 to +30
Liver	+40 to +60
Soft Tissue, Contrast	+100 to +300
Bone	+700 (cancellous bone) to +3000 (cortical bone)

Figure 2: Different range of Hounsfield Unit for various materials

range of Hounsfield unit around -300 while other materials like air are around -1000. Above Figure.2 shows different ranges of various materials. We can obtain the HU value from the original pixel array by doing a linear transformation using RescaleIntercept and RescaleSlope attributes.

1.2.2. Tabular Data

According the description of the tabular data, after the CT scan at the base week, every patient will receive a FVC test every week within a period of 140 weeks. Other information provided is showed as follows

* patients' sex
* age
* smoking status (currently smoke, never smoke and ex-smoker)
* the week that received test
* percent (a computed field which approximates the patient's FVC as a percent of the typical FVC for a person of similar characteristics)
* FVC (the recorded lung capacity in ml)

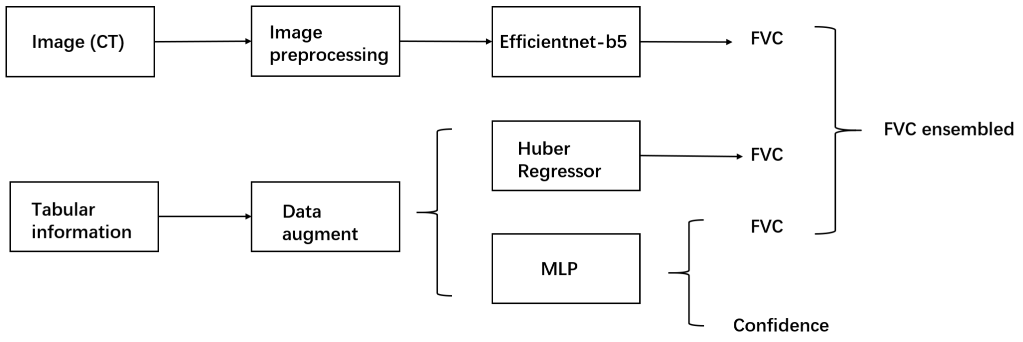


Figure 3: flow graph of prediction ensemble System

Within these feature, FVC is the target value that we needs predict and others are input features. As for the number of input rows, as mentioned above, every patients has several records at different weeks. However, different patient has different number of rows in the table ranging from 9 to 20, which is a challenge part for training machine models.

2. System design

Our prediction system can be decomposed into four main stages: (1) efficient net (2) MLP (3) HuberRegressor (4) UI

In these sub-models, we use (1) and (2) Shared by others, but we also Add some parts of our own improvement, like data augment and feature discretion. The flow graph of our system pipeline is shown in Figure.3.

2.1. EfficientNet

Our data is composed of patients' CT scans and the tabular data. Before training our model, we firstly need to do some pre-processing work on training data then we build deep neural networks for extracting features from CT scan. The brief flow chart in shown in Figure.4. The CT scans are treated as the input images of EfficientNet to extract useful information. And, after concatenate of the output of EfficientNet and some tabular information of patients, We also add another two fully-connected layers and dropout layer to fuse more knowledge and avoid overfitting.

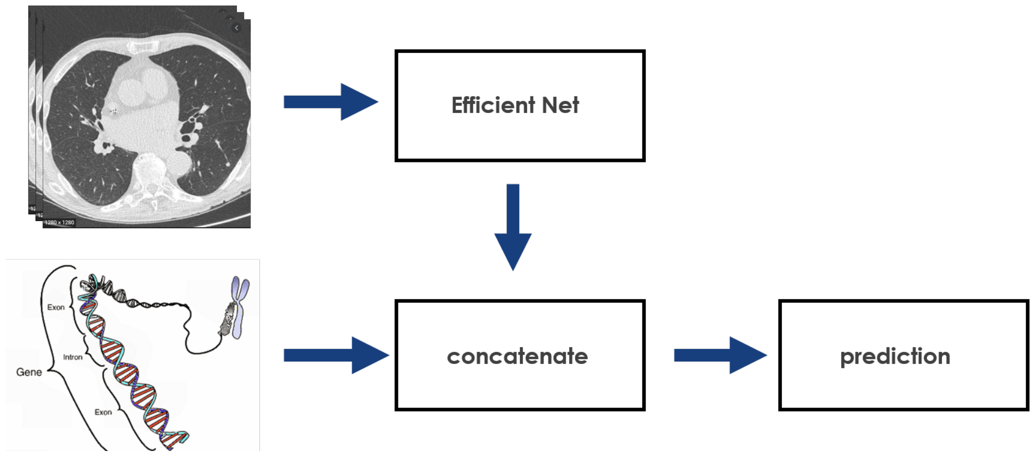


Figure 4: brief flow chart of image processing stage

2.1.1. Processing Data

For the CT scans images, the Hounsfield Unit(HU) transformation (explained in last section) should be conducted before training the model. Besides, the smoking status and sex features in the tabular data are both discrete data which needs one-hot encoding. Additionally, we also stack the patient's CT scans pixel arrays with his tabular information (sex, smoking status etc.). Therefore, the input data of our model is several patients' scan images together with his or her tabular information. As for the output, our model will predict the speed (the slope) of the patients' decline in lung function.

2.1.2. Building Model

1) EfficientNet Introduction

The backbone model that we used is the Efficient Net[2] proposed by Google in 2019. The model is obtained by Neural Architecture Search (NAS) in a search space that is similar to Mobile Net. The main contribution of Efficient Net is to find a compound scaling relationship between the depth, width and resolution of a deep learning model. The research firstly built a baseline model (Efficient B0) through NAS and introduced a method to compute the relationship between depth, width and resolution. Then the baseline model was gradually scaled up according to relationship and obtained several different models (Efficient B1-B7). Compared to previous deep learning models, the main advantage of Efficient models is efficiency

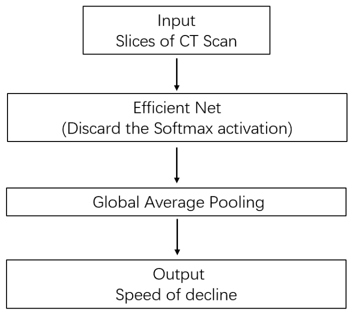


Figure 5: Backbone Model

and accuracy.

For example, GPipe model can get a 84.3 percent top-1 accuracy on ImageNet with 557 million parameters while EfficientNet B7 got a top-1 accuracy of 85 percent with 66 million parameters! Therefore, considering the hardware that we utilize (GPU, CPU, the size of memory), we decide to apply EfficientNet as our backbone model.

2) Model Training

For the training part, we firstly refer to the result of a Kaggle model which also used Efficient Net. The structure of the model shows as following Figure.5 As mentioned above, the Efficient Net has seven models (B0-B7) with different parameter sizes so we need to figure out the most suitable one for our CT scan. Additionally, there are several hyper-parameters needs to be tested including batch size, epoch and learning rate. The way we split training data is k-fold cross validation so it is also necessary to test different split. To train our model, we choose MAE as our model loss function and apply AdamW as our optimizer. Besides, we utilize a dacay learning rate to avoid local minimum. In the initial run, we set batch size to 8 with 10 epochs and test EfficientNet model from B0 to B7. Then we compare different combinations of batch size and epochs. Finally we test different split of training data using k-fold cross validation. The criteria we use for comparing the trained models is the score formulation defined in section 3.1. In the process of real training, we found that it is unrealistic for us to train EfficientNet B7 due to the limitation of GPU memory. Among the rest of the models, Efficient B6 performs best with a score of -6.914 so we

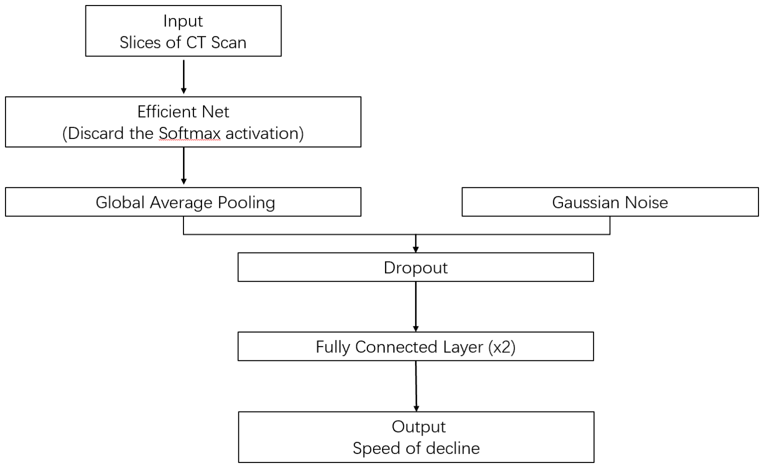


Figure 6: Improved Model

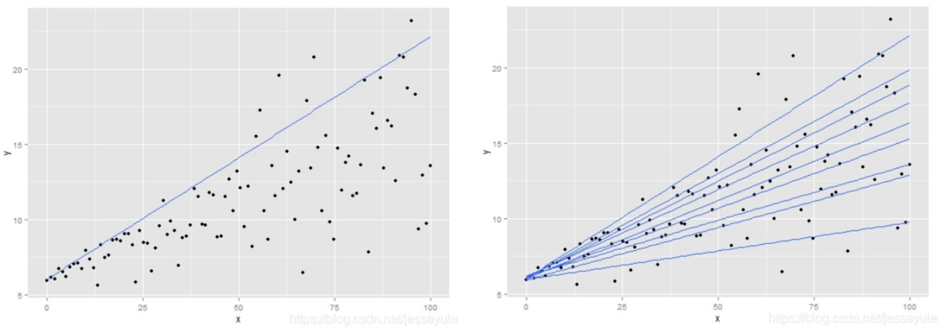
choose Efficient B6 as our backbone model. For the epoch and batch size, after try and examination we choose a batch size of 4 and epoch of 50.

3) Model Changes

   Although we improve the performance of our model through parameter tuning, it get stuck in the score around -6.86 (the calculation of score is described in Experiment section). Meanwhile, considering the size of the training set and the number of parameters of EfficientNet model. it is likely that our model get overfit and also extract features that are too abstract. Thus we add another two fully-connected layers after the output of EfficientNet and introduce another Gaussian Noise input and dropout layer to avoid overfit. Our improved model is shown in Figure.6. We also change our dropout rate to 0.5 and the nodes of the fully connected layer to 100. After parameter tuning, our model achieve a score of -6.843, which is better than our previous one.

2.2. MLP

It is hard to predict FVC among the whole 136 weeks perfectly because of thousands of external factors. So, we introduce Confidence to assess accuracy of forecast. Fortunately, quantile regression can achieve this. We can see in picture.7 (a), normal regression can only predict the expectation of prediction. According



(a) Normal Regression (b) Quantile Regression

Figure 7: The different between normal regression and quantile regression.

to picture.7 (b), quantile regression can calculate the distribution of prediction, which can reflect the stability directly.

So, we chose MLP to predict the Confidence value by regression, besides, to avoid overfitting and have robust performance, we only use simple fully connected layers to build the model.

2.2.1. fully connected layer

In the Multilayer perceptron, there may be have more than one linear layer (combinations of neurons). If we take the simple example the three-layer network, first layer will be the input layer and last will be output layer and middle layer will be called hidden layer. We feed our input data into the input layer and take the output from the output layer. We can increase the number of the hidden layer as much as we want, to make the model more complex according to our task. The diagram of MLP is shown in Figure.8. In our system, we use 4 fully connected layers for prediction ,and the final output has three neurons, which are for top situation (for possibility), normal situation and middle situation respectively (The middle one is predicted FVC, and Confidence can be calculated by the third output minus the first one).

2.2.2. Loss define

In a supervised classification system, each input vector is associated with a label, or ground truth, defining its class or class label is given with the data. The output of the network gives a class score, or prediction, for each input. To measure the performance of the classifier, the loss function is defined. The loss will be high if the predicted class does not correspond to the true class, it will be low otherwise.

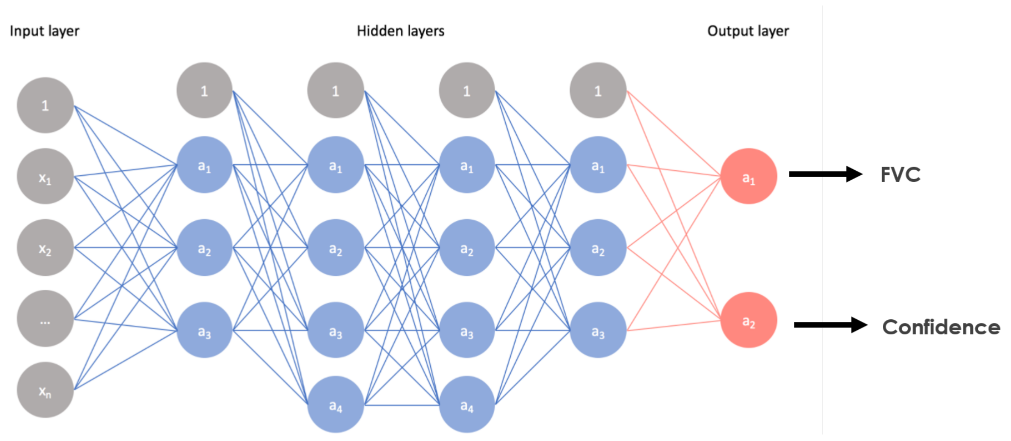


Figure 8: diagram of MLP

In our prediction task, MLP model is used to predict both FVC and confidence by regression, so we use quantile loss for back propagation. We define a list with three numbers to calculate the variability and instability of predict results , in our system, we use $[0.2, 0.5, 0.8]$ for prediction.

2.2.3. model use

The input of MLP is the tensor extract from tabular information of patients, in total is $N * M$, where N is the traning sample of dataset and M is the dimension of each sample such as age and sex. For FVC prediction part, we can assume that the middle output is our predict FVC because 0.5 the median possible output, for confidence prediction part, we can define the subtract of first and third output as our predction, because $[0.8, 0.2]$ is the instability of our possible output, which can represent instability.

2.2.4. model changes

In the original method, we use age, sex, percent, smoking status, and base FVC for tabular information of per patients, specially, base FVC and age is the continuous features, and we do normalization before input, and smoking status, sex is the discrete features. However, because of the limited training dataset (we only have 176 patients with around 1500 training samples), the model trained is very unstable, so, to improve the accuracy of prediction, we treat every weeks in the training set of each patients to be the base week, and add 'week pass' feature. In this way, the input training data can be increased to 10 times. The diagram

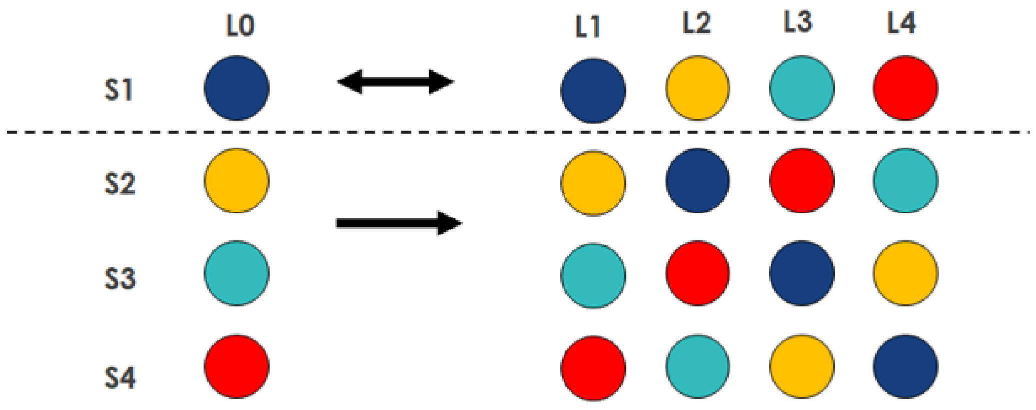


Figure 9: diagram of data augment

of this process is shown in Figure.9. According to Figure.9, $s1, s2, s3, s4$ is four week point of one sample, through our data augment method, we can have total 16 data input vector with 4 different base weeks instead of 1.

second, because some continuous features has less continuous information. For example, age feature does not perform well even after normalization process. Because of this, we use feature binning for our feature discretion. Specifically, we discrete age into 5 dimensional one hot vector, and the discreted points are $[64, 70, 75, 79, 88]$.

Third, because we still have very few patients information, so if we can use prior knowledge to calculate more features related to patients, it will be a huge improvement in data lavel. So, we use

$$fvc_{female} = (21.78 - 0.101a) * h \tag{1}$$

$$fvc_{male} = (27.63 - 0.112a) * h \tag{2}$$

where a is age in years, and h is height in cm. From these two equations, we can get patients' height information using the features we have.

2.3. *Huber Regressor*

2.3.1. *motivation*

Linear regression model that is robust to outliers. According to Figure.10, although there are some zigzag and some increase decrease repeatedly, the overall

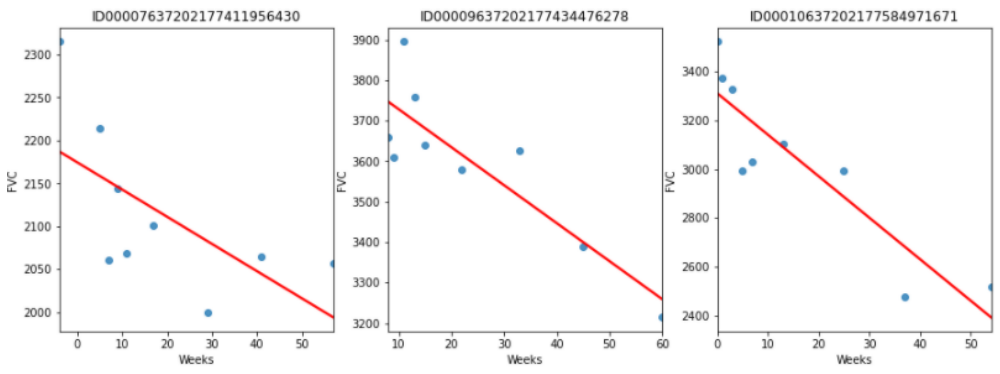


Figure 10: Visualization of three patients according to their FVC of each week point in training dataset

trend of patients FVC is decrease with a certain slope. So, if we can use a linear model to predict the slope, we can get a higher robustness model ensembled with the current model. Besides, the linear model is very easy to train, so the submodel has a high cost-effective and don't pay a large time resources.

2.3.2. model define

The Huber Regressor optimizes the squared loss for the samples where $|(y - X'w)/sigma| < epsilon$ and the absolute loss for the samples where $|(y - X'w)/sigma| > epsilon$, where w and sigma are parameters to be optimized. The parameter sigma makes sure that if y is scaled up or down by a certain factor, one does not need to rescale epsilon to achieve the same robustness. Note that this does not take into account the fact that the different features of X may be of different scales. This makes sure that the loss function is not heavily influenced by the outliers while not completely ignoring their effect. In our system, we import the Huber library in python library and the default hyperparameters.

In input data level, we also use the input preprocessed data described above.

2.4. FE-BE Design

We use Flask framework in our project. There are a number of frameworks for Python, including Flask, Tornado, Pyramid, and Django. Compare with other frameworks, Flask is more flexible, simple, low development cost, and it still have a basic MTV (Model-View-Template) structure.

In our root directory, we have a templates folder that stores a series of HTML template. The FE control cooperates with JS to store the target CT scan in source folder, BE analyzes patient's CT scan and meta data, such as age, gender , calculates the most possible FVC value in more than 100 weeks in the feature- . At last, the FE js controls and reads the result , it will display the prediction result in a line chart. We hope both doctors and patients can get what they want from our line chart.

3. Experiment and Conclusion

We run our code and algorithms in colab notebook, in the final system design, we save as py file and test in local environment.

For the Efficient net, MLP and Huber training, we use cross validation to have a better see the performance of model and have more training data. In our experiment, we use 5 fold for training. It will be around 40 minutes to run the total system.

3.1. score of measure our system performance

Because we have two output results, the one is FVC, and the other is Confidence to reflect the accuracy of FVC. We define a metric to measure the performance of both the two things. The equations are shown below,

$$\sigma_{clipped} = max(\sigma, 70) \tag{3}$$

$$\Delta = min(|FVC_{true} - FVC_{predicted}|, 1000) \tag{4}$$

$$metric = -\frac{\sqrt{2}\Delta}{\sigma_{clipped}} - \ln(\sqrt{2}\sigma_{clipped}) \tag{5}$$

where $\sigma_{clipped}$ is Confidence we predict .

3.2. Experiment in efficient net

For the input image of efficient net is $2D$ image, in every training epoches, we select the $2D$ slice in a pile of patients' slices randomly, so, with more epoch in a certain extent, the more accuracy (higher score) we can get. Here are the model performance of different $BatchSize$ and $Epoch$.

According to table.1, we try different $BatchSize$ and $Epoch$ to fine tune our model, we can see that efficient net performs best when $batchsize = 4$ and $epoch = 50$ ($scrore = -6.862$). This is because with more epoches, we can input more $2D$ CT scans to model to train, which will capture more useful information.

Table 1: The varying performance with respect to $Batch(B)Size$ and $Epoch(E)$ upon EfficientNet training process. The best results are bold.

E / B	B=4	B=8	B=16	B=32
E=10	-6.912	-6.914	-6.918	-6.926
E=20	-6.901	-6.905	-6.913	-6.92
E=30	-6.887	-6.892	-6.902	-6.908
E=40	-6.866	-6.879	-6.892	-6.904
E=50	**-6.862**	-6.875	-6.887	-6.896

Table 2: Different performance with different submodels (branches) ensembled. The best results are bold.

Method	$Score$
Efficient net	-6.862
MLP	-6.857
Huber Regressor	-6.868
Efficient net + MLP	-6.849
Efficient net + MLP + Huber Regressor	**-6.841**

3.3. Experiment in while ensembled model

According to Table.2, we can see the final score of our model can get -6.841.

The final visualization of output results and truth label in training set are shown in Figure.11. According to Figure.11, we can see our model has a perfect fitness for ground truth globally. Specifically, $q50$ is the predict FVC, and $q75 - q25$ is the confidence we predict.

3.4. Findings and discussion

In our project, we did a hard effort to find and construct models. What's more, we spend a lot of time to fine tune the model to improve the model performance on FVC and Confidence prediction. In this system design project, we knew data is the fundamental role in the model performance, data preprocessing is a tough but advanced method to learn. Secondly, we use a lot of knowledge we learnt in class to our system, including ensmeble learning, data augment and construction of fully convolutional layer. In the ensemble level, we just fuse the output of these models and add with different weights. In our system, we distribute 0.16 for efficient net, 0.64 for MLP and 0.2 for Huber Regressor.

We also done some other changes and novel changes, like $3D{-}reconstruction$ subscribed in next section, but we don't add them in our current system (you can find in our github address with same link) because of the time limit.

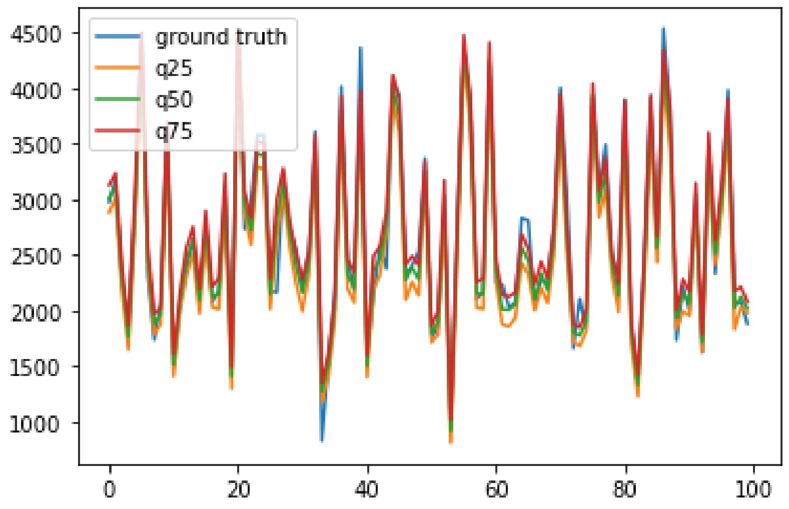


Figure 11: Visualization of three patients according to their FVC of each week point in training dataset

3.5. 3D reconstrction

In our project, we also reconstruct the 3D CT scan, The method we used is in [3] Here are the steps

1) Loading every slice of a patient together to obtain a 3D-array using pydicom package.
2) Obtain the slice thickness of a patient by slice location in the metadata to help calculate the original size of the lung in the following steps.
3) Hounsfield Unit (HU) transformation.
4) Re-sampling all the scan to a certain isotropic resolution (1mmx1mmx1mm) to avoid distortion of the final 3D image.
5) Zoom the 2d image to the same size, remain the number of slices position. This step is added by ourselves. Every size of the output array size after re-sampling is different from each other, which is common because in real life everyone has a different size of lungs. However, this could be difficult for applying deep learning models. So we resize the 2d image to the same and keep their channels different.

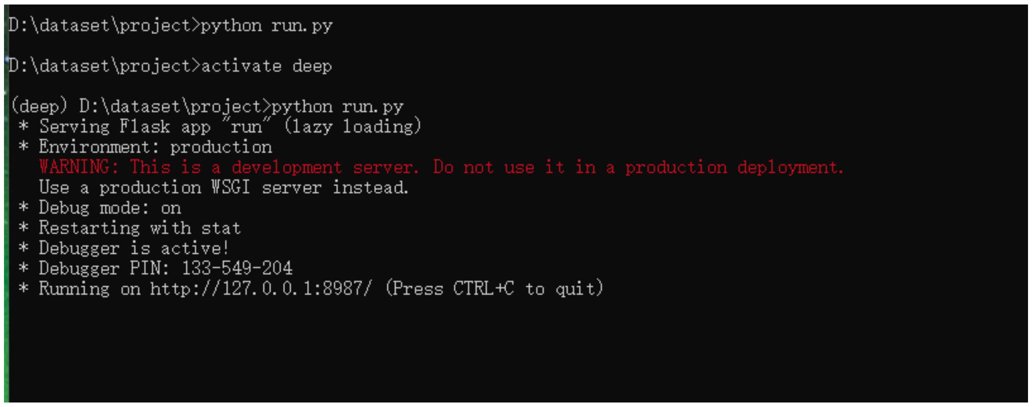
6) Segment the lung tissue from surrounding environment by labeling the largest connected components.

4. Appendices

4.1. Installation

1.Download extract zip from address
2.Install sklearn
3.Make sure tensorflow==2.2.0
4.Make sure flask >= 1.1.1
5.Make sure pydicom >= 2.0.0
6.Make sure seaborn >= 0.11.0
7.Make sure efficientnet == 1.1.1
8.Make sure tensorflow-addons >= 0.11.2
9.Make sure plotly == 4.11.0

4.2. User Guide

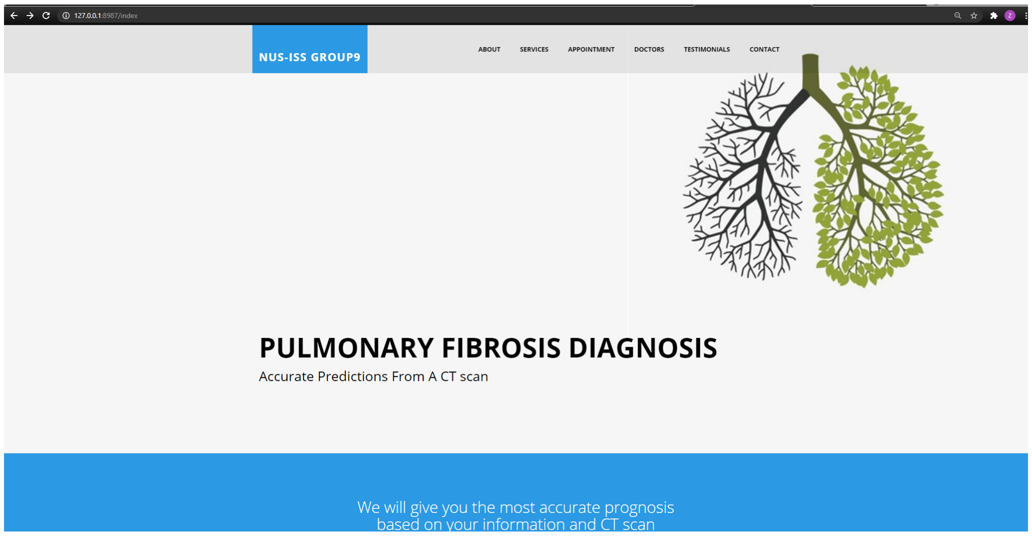


```
D:\dataset\project>python run.py

D:\dataset\project>activate deep

(deep) D:\dataset\project>python run.py
 * Serving Flask app "run" (lazy loading)
 * Environment: production
   WARNING: This is a development server. Do not use it in a production deployment.
   Use a production WSGI server instead.
 * Debug mode: on
 * Restarting with stat
 * Debugger is active!
 * Debugger PIN: 133-549-204
 * Running on http://127.0.0.1:8987/ (Press CTRL+C to quit)
```

1.After doing installation , run run.py in root directory.

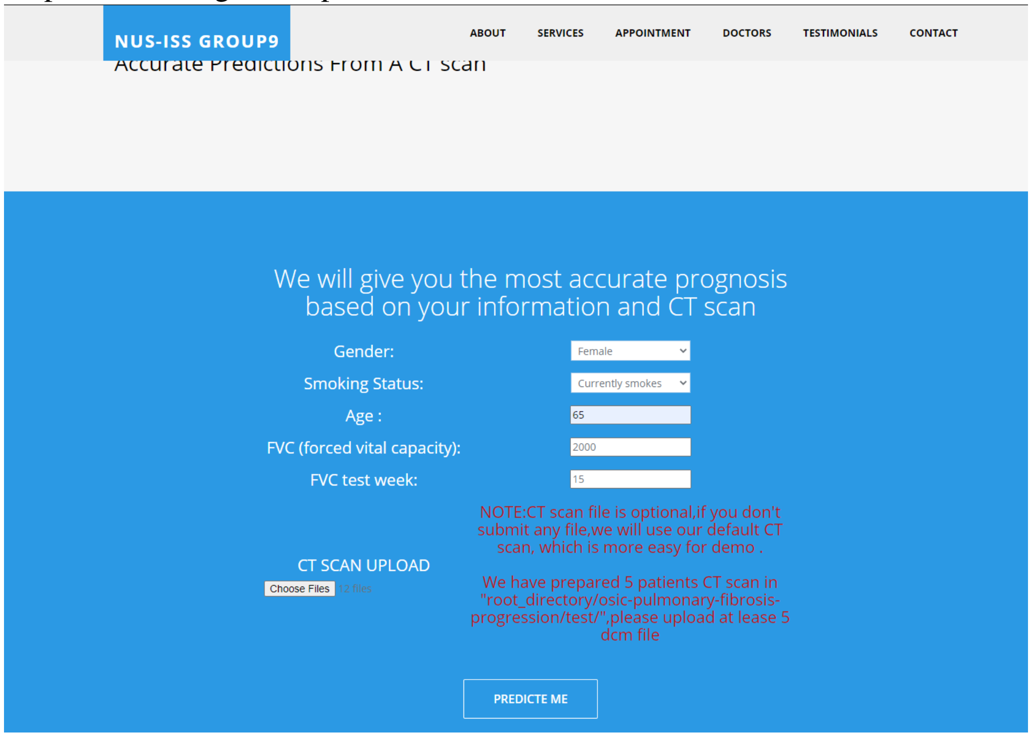


PULMONARY FIBROSIS DIAGNOSIS

Accurate Predictions From A CT scan

We will give you the most accurate prognosis based on your information and CT scan

2.open browser, go to http://127.0.0.1:8987/index.



Accurate Predictions From A CT scan

We will give you the most accurate prognosis based on your information and CT scan

Gender:	Female
Smoking Status:	Currently smokes
Age :	65
FVC (forced vital capacity):	2000
FVC test week:	15

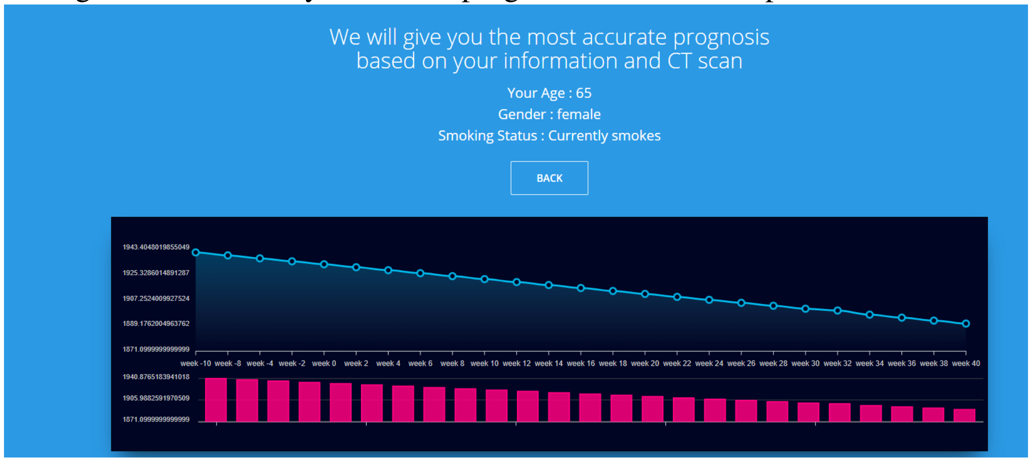
CT SCAN UPLOAD

NOTE:CT scan file is optional,if you don't submit any file,we will use our default CT scan, which is more easy for demo .

We have prepared 5 patients CT scan in "root_directory/osic-pulmonary-fibrosis-progression/test/",please upload at lease 5 dcm file

3.Enter patient's detail, upload CT scan (can skip CT scan upload, we will use default CT scan), and click predict me.

4. Waiting for 4 - 10 mins, you can see progress in terminal output



We will give you the most accurate prognosis
based on your information and CT scan

Your Age : 65
Gender : female
Smoking Status : Currently smokes

5. In FE , we can see output line chart,
6. Video introduction: address

References

[1] E. Vasarmidi, E. Tsitoura, D. A. Spandidos, N. Tzanakis, K. M. Antoniou, Pulmonary fibrosis in the aftermath of the covid-19 era, Experimental and therapeutic medicine 20 (3) (2020) 2557–2560.

[2] H. X. Bai, R. Wang, Z. Xiong, B. Hsieh, K. Chang, K. Halsey, T. M. L. Tran, J. W. Choi, D.-C. Wang, L.-B. Shi, et al., Ai augmentation of radiologist performance in distinguishing covid-19 from pneumonia of other etiology on chest ct, Radiology (2020) 201491.

[3] B. Schlueter, K. B. Kim, D. Oliver, G. Sortiropoulos, Cone beam computed tomography 3d reconstruction of the mandibular condyle, The Angle Orthodontist 78 (5) (2008) 880–888.