DAT 565E Final Project: X-Ray Image Diagnosis of COVID-19

Group 22   Section 21

Jimmy Qiu, Xinrui Jiao, Yi Wang, Annabelle Zhu

Abstract

Coronavirus disease 2019 (COVID-19) is becoming one of the most severe problems all over the world. According to data from Corona Resource Center, Johns Hopkins University, COVID-19 has already infected over 85 million people all over the world. Also, The World Health Organization (WHO) declared that the outbreak of COVID-19 constitutes a Public Health Emergency of International Concern. As an effective examination method to reflect the lung condition, X-ray images can quickly reflect the patient's lung condition, thus helping doctors to make diagnosis. As the medical equipment and experts in the hospitals are so limited, it is strongly required that an automatic identifying screening system could be created to identify infected patients. To fight against the disease, we can build deep learning models to identify the X-Ray images of infected COVID-19 patients, thus we can save the diagnosis time, improve the efficiency and save the cost

We applied the model to each of the 3 levels of diagnosis. First, we implemented binary classification models to detect the X-Ray image to see whether it is from Pneumonia infected patients. Then, we used multi-classification models to identify whether the patient is infected with Virus. Finally, we still used multi-classification models to identify whether the patient is infected with COVID-19. We used supervised models (Dense-Layer and Convolutional Neural Network) and self-supervised models (Auto-Encoder) for the classification.

For the level-1 diagnosis, CNN achieved the best recognition effect, with an average accuracy of over 0.95. For the level-2 diagnosis, AE achieved the best recognition effect, with a test accuracy of over 0.75. For the level-3 diagnosis, CNN achieved the best recognition effect, with a test accuracy of over 0.98. For the level-1and level-3 diagnosis, the model has achieved very good results. Considering that the diagnosis of pneumonia and COVID-19 infection are the two most important problems now, our model has very good problem-solving ability and also has good commercial application value.

Key word: Pneumonia, COVID-19, Supervised Learning, Self-Supervised Learning

Introduction

Coronavirus disease 2019 (COVID-19) is a contagious disease caused by severe acute respiratory syndrome coronavirus 2 (SARS-CoV-2). COVID-19 has become a major risk disease affecting the health of people around the world. Different countries have suffered different degrees of loss. Among them, the diagnosis of COVID-19 has become a top priority in the fight against the epidemic. As an effective examination method to reflect the lung condition, X-ray images can quickly reflect the patient's lung condition, thus helping doctors make diagnosis.

However, the shortage of medical resources and medical staff makes the process very slow, which is likely to aggravate the infection of patients with mild symptoms. In order to solve this problem, we can try to use the deep learning model specially dealing with image classification to speed up the diagnosis of whether a patient is infected with pneumonia and play an auxiliary role for doctors, thus shortening the treatment time, improving the survival rate and alleviating the suffering of patients. With the Chest X-Ray Dataset, we need to use X-Ray images and given labels to train a deep learning model with good accuracy.

The model we used in this final project not only includes DNN and CNN used in the previous research, but also attempts to use the automatic encoder in self-supervised learning to deconstruct and learn images, and then classify the original images. Not only that, but in a breakthrough, we divided the model into three levels, from determining whether the patient is infected with pneumonia, to determining the cause of pneumonia, and to determining the virus type of viral pneumonia. This is our breakthrough and attempt on the basis of previous research.

Review of the literature

COVID-19 is a challenging disease that has infected millions of people all around the world while the medical equipment and experts in the hospitals are so limited. It is strongly required that an automatic identifying screening system could be created to identify COVID-19 infected patients. Therefore, in order to solve the problem of COVID-19, several machine learning techniques have been applied into the computer-aided diagnosis area **Chen (et al. 2020)**. Take **Chandra (el al. 2020)** as an instance, **Chandra (el al. 2020)** employed hierarchical classification using conventional ML algorithms and radiomic texture descriptors to segregate normal, pneumonia, and nCOVID-19 infected patients, which achieved above 90% accurate rate. Moreover, many deep-learning models have been applied to automatically identify COVID-19. **Pesce (et al. 2019)** applied two neural network models to detect pulmonary lesions in chest X-ray images. The first model used the back-propagation mechanism, featuring an improved localization capability, which, in turn, boosts the classification performance while the second model implemented an extension of the Recurrent Attention Model, rewarding a higher classification score when the glimpses attended by the algorithms during training overlap with the correct lesion locations.

Gao (2020) developed a deep convolutional neural network (CNN) that is able to assist radiologists with diagnosis by distinguishing COVID-19 pneumonia from non-COVID-19 pneumonia in patients based on chest X-ray classification and analysis, reaching the average accuracy to above 95%. As **Kiran Purohit (et al., 2020)** mentioned, though RT-PCR methodology has developed into a mature methodology to detect coronavirus infection, it did not reach the desired accurate level. Thus, **Kiran Purohit (et al., 2020)** used a convolutional neural network (CNN) based multi-image augmentation technique for detecting COVID-19 in chest X-

Ray and chest CT scan images of coronavirus individuals. Using this approach, the accurate rate of the proposed models significantly enhanced, which turned out to be around 95.38% and 98.97% for CT scan and X-Ray images, respectively.

Problem description

In our dataset, we have X-Ray images of COVID-19 infected and uninfected patients. We built deep learning models with high accuracy to identify X-Ray images of COVID-19 patients, which included supervised models (Dense Layers and Convolutional Neural Network) and self-supervised models (Autoencoder) for the classification. We tried different optimizers for each model to find the one with highest accuracy to detect COVID-19 X-Ray images. Compared to methods in previous articles, we improved the validation accuracy from 95% above to 96% above **Gao (2020)**. Especially, we tried a self-supervised model but not weakly labelled or annotated images **Pesce (et al. 2019)**.

In detail, first, we implemented binary classification models to detect the X-Ray image to see whether it is from Pneumonia infected patients; Then, we used multi-classification models to identify whether the patient is infected with Virus; Finally, we still used multi-classification models to identify whether the patient is infected with COVID-19. For each level of recognition, we want the accuracy of the verification set and test set to reach an ideal level. At the same time, we will choose the model which is most suitable for each level of recognition among the three kinds of neural network models.

Model description

1. Level-1 Diagnosis: Normal / Pneumonia - Binary Classification

In this part, We are trying to use a binary classification model to determine whether the patient is infected with pneumonia. Based on this preliminary prediction, we will further confirm the type of pneumonia and determine whether it is COVID-19. To ensure the comprehensiveness and accuracy of our analysis, we considered both supervised learning and unsupervised learning.

(1) Supervised Model

For supervised models, we use a simple neural network and convolutional neural network to see which model is more appropriate for our dataset. As you know, our research data is based on the X-ray images of patients' lungs. Therefore, we use a simple neural network to explore the feasibility of research and analysis, and then try CNN to see whether it can improve the accuracy.

- **Dense-Layer Neural Network (DNN)**

The dense layer is a neural network layer that is connected deeply, which means each neuron in the dense layer receives input from all neurons of its previous layer. The dense layer is found to be the most commonly used layer in the models. To test the data preliminarily, we use a three-layer model with 2 hidden layers. In the first hidden layer, we use 256 neurons and 64 neurons in the second hidden layer, with both activation functions of ReLU. In the output layer, we use Sigmoid. In this model, we choose binary-crossentropy as loss function and Adam as optimizer, and we track the accuracy of each epoch.

```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #   
=================================================================
flatten (Flatten)            (None, None)              0         
_________________________________________________________________
dense (Dense)                (None, 256)               3145984   
_________________________________________________________________
dense_1 (Dense)              (None, 64)                16448     
_________________________________________________________________
dense_2 (Dense)              (None, 1)                 65        
=================================================================
Total params: 3,162,497
Trainable params: 3,162,497
Non-trainable params: 0
_________________________________________________________________
```

Figure 1 Structure of DNN of Level-1 Diagnosis

- **Convolution Neural Network (CNN)**

Convolutional neural network (CNN) is a popular algorithm which can take in an input image, assign importance to various objects in the image and be able to differentiate one from the other. So, we choose this model in order to further improve the accuracy. The general idea is importing the image data and transforming into tensor data, then using convolutional layer and pooling layer to reduce the dimensions of the data, finally utilizing the dense layer to generate the output which is a classification output.

In our final convolutional model, we use 2 convolutional layers and 2 dense layers. The first convolution layer has 64 filters with size 3*3 and second convolution layer has 15 filters with size 3*3, both with max pooling. Then we flat the data and pass to a full-connected layer with 16 neurons then output the results using sigmoid. All activation functions in hidden layers are relu. We use Adam as the optimizer and binary-crossentropy as the loss function to train the data.

```
Model: "sequential_2"
_________________________________________________________________
Layer (type)                 Output Shape              Param #   
=================================================================
conv2d_4 (Conv2D)            (None, 62, 62, 64)        1792      
_________________________________________________________________
max_pooling2d_4 (MaxPooling2 (None, 31, 31, 64)        0         
_________________________________________________________________
conv2d_5 (Conv2D)            (None, 29, 29, 16)        9232      
_________________________________________________________________
max_pooling2d_5 (MaxPooling2 (None, 14, 14, 16)        0         
_________________________________________________________________
flatten_2 (Flatten)          (None, 3136)              0         
_________________________________________________________________
dense_4 (Dense)              (None, 16)                50192     
_________________________________________________________________
dense_5 (Dense)              (None, 1)                 17        
=================================================================
Total params: 61,233
Trainable params: 61,233
Non-trainable params: 0
_________________________________________________________________
```

Figure 2 Structure of CNN of Level-1 Diagnosis

(2) Self-Supervised Model - AutoEncoder (AE)

An autoencoder is a type of artificial neural network used to learn efficient data coding in an unsupervised manner. The aim of an autoencoder is to learn a representation (encoding) for a set of data, typically for dimensionality reduction, by training the network to ignore signal “noise”. It is an unsupervised artificial neural network that learns how to efficiently compress and encode data then learns how to reconstruct the data back from the reduced encoded representation to a representation that is as close to the original input as possible.

In business situations, most of the data we encounter are not labeled, or a lot of medical data may only retain the X-ray film left at the time of diagnosis, but there is no clear record of the patient's diagnosis result. Therefore, we consider using a self-monitoring model, ignoring the labels in the data. A successful self-monitoring model is used to predict the diagnosis results.

Autoencoders consist of 4 main parts. The first part is encoder, in which the model learns how to reduce the input dimensions and compress the input data into an encoded representation. We use 2 convolution layers and 2 full-connected layers here then output the results with 2 classes. The second part is the bottleneck, which is the layer that contains the compressed representation of the input data. This is the lowest possible dimension of the input data. Here we have 2 dimensions of data to represent each image. The third part is decoder, in which the model learns how to reconstruct the data from the encoded representation to be as close to the original input as possible. Finally, we need to define the reconstruction loss. This is the method that measures how well the decoder is performing and how close the output is to the original input. We use MSE as our loss function and Adam as optimizer.

Both encoder and decoder are transformed into multi-level CNN network, so as to ensure the learning of image mode has a small loss. In terms of classification, encoder is separated from AE and its parameter is set as untrainable. Meanwhile, a simple dense layer is connected for training, and then the images are classified and diagnosed.

```
input_img = Input(shape=(128,128,3))
def encoder(input_img):
    #encoder
    conv1 = Conv2D(32, (3, 3), activation='relu', padding='same')(input_img)
    conv1 = BatchNormalization()(conv1)
    conv1 = Conv2D(32, (3, 3), activation='relu', padding='same')(conv1)
    conv1 = BatchNormalization()(conv1)
    pool1 = MaxPooling2D(pool_size=(2, 2))(conv1)
    conv2 = Conv2D(64, (3, 3), activation='relu', padding='same')(pool1)
    conv2 = BatchNormalization()(conv2)
    conv2 = Conv2D(64, (3, 3), activation='relu', padding='same')(conv2)
    conv2 = BatchNormalization()(conv2)
    pool2 = MaxPooling2D(pool_size=(2, 2))(conv2)
    conv3 = Conv2D(128, (3, 3), activation='relu', padding='same')(pool2)
    conv3 = BatchNormalization()(conv3)
    conv3 = Conv2D(128, (3, 3), activation='relu', padding='same')(conv3)
    conv3 = BatchNormalization()(conv3)
    conv4 = Conv2D(256, (3, 3), activation='relu', padding='same')(conv3)
    conv4 = BatchNormalization()(conv4)
    conv4 = Conv2D(256, (3, 3), activation='relu', padding='same')(conv4)
    conv4 = BatchNormalization()(conv4)
    return conv4
```

```
def decoder(conv4):
        #decoder
        conv5 = Conv2D(128, (3, 3), activation='relu', padding='same')(conv4)
        conv5 = BatchNormalization()(conv5)
        conv5 = Conv2D(128, (3, 3), activation='relu', padding='same')(conv5)
        conv5 = BatchNormalization()(conv5)
        conv6 = Conv2D(64, (3, 3), activation='relu', padding='same')(conv5)
        conv6 = BatchNormalization()(conv6)
        conv6 = Conv2D(64, (3, 3), activation='relu', padding='same')(conv6)
        conv6 = BatchNormalization()(conv6)
        up1 = UpSampling2D((2,2))(conv6)
        conv7 = Conv2D(32, (3, 3), activation='relu', padding='same')(up1)
        conv7 = BatchNormalization()(conv7)
        conv7 = Conv2D(32, (3, 3), activation='relu', padding='same')(conv7)
        conv7 = BatchNormalization()(conv7)
        up2 = UpSampling2D((2,2))(conv7)
        decoded = Conv2D(3, (3, 3), activation='sigmoid', padding='same')(up2)
        return decoded
```

Figure 3 Structure of AE of Level-1 Diagnosis

2. Level-2 Diagnosis: Virus / Bacteria / Stress-Smoking - Multiple Classification

Level-3 Diagnosis: COVID-19 / SARS / Other - Multiple Classification

In the latter two levels of diagnosis, we will focus on the comparison between models rather than the optimization of the model's internal parameters. This is because the multi-classification problem is more complex than the dichotomy problem, and the performance difference between different models will be greater. DNN, CNN and AE were still used as models to solve these two multi-classification problems.

The difference is that we set the number of iterations of the model to 20 and the batch size to 64. Also, we build DNN and CNN with more neurons, and use Relu and SoftMax as activation functions, hoping to get better training effect. Not only that, we also increased the input size of the image to 128*128 pixels, which will have a positive effect on the training of the model.

```
Model: "sequential_2"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
flatten_3 (Flatten)          (None, None)              0
_________________________________________________________________
dense_7 (Dense)              (None, 512)               25166336
_________________________________________________________________
dense_8 (Dense)              (None, 128)               65664
_________________________________________________________________
dense_9 (Dense)              (None, 32)                4128
_________________________________________________________________
dense_10 (Dense)             (None, 3)                 99
=================================================================
Total params: 25,236,227
Trainable params: 25,236,227
Non-trainable params: 0
```

```
Model: "sequential_1"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d (Conv2D)              (None, 126, 126, 16)      448
_________________________________________________________________
max_pooling2d (MaxPooling2D) (None, 63, 63, 16)        0
_________________________________________________________________
flatten_1 (Flatten)          (None, 63504)             0
_________________________________________________________________
dense_3 (Dense)              (None, 16)                1016080
_________________________________________________________________
dense_4 (Dense)              (None, 2)                 34
=================================================================
Total params: 1,016,562
Trainable params: 1,016,562
Non-trainable params: 0
```

Figure 4 Structure of DNN and CNN of Level-2 & Level-3 Diagnosis

```python
input_img = Input(shape=(128,128,3))
def encoder(input_img):
    #encoder
    conv1 = Conv2D(32, (3, 3), activation='relu', padding='same')(input_img)
    conv1 = BatchNormalization()(conv1)
    conv1 = Conv2D(32, (3, 3), activation='relu', padding='same')(conv1)
    conv1 = BatchNormalization()(conv1)
    pool1 = MaxPooling2D(pool_size=(2, 2))(conv1)
    conv2 = Conv2D(64, (3, 3), activation='relu', padding='same')(pool1)
    conv2 = BatchNormalization()(conv2)
    conv2 = Conv2D(64, (3, 3), activation='relu', padding='same')(conv2)
    conv2 = BatchNormalization()(conv2)
    pool2 = MaxPooling2D(pool_size=(2, 2))(conv2)
    conv3 = Conv2D(128, (3, 3), activation='relu', padding='same')(pool2)
    conv3 = BatchNormalization()(conv3)
    conv3 = Conv2D(128, (3, 3), activation='relu', padding='same')(conv3)
    conv3 = BatchNormalization()(conv3)
    conv4 = Conv2D(256, (3, 3), activation='relu', padding='same')(conv3)
    conv4 = BatchNormalization()(conv4)
    conv4 = Conv2D(256, (3, 3), activation='relu', padding='same')(conv4)
    conv4 = BatchNormalization()(conv4)
    return conv4
```

```python
def decoder(conv4):
        #decoder
        conv5 = Conv2D(128, (3, 3), activation='relu', padding='same')(conv4)
        conv5 = BatchNormalization()(conv5)
        conv5 = Conv2D(128, (3, 3), activation='relu', padding='same')(conv5)
        conv5 = BatchNormalization()(conv5)
        conv6 = Conv2D(64, (3, 3), activation='relu', padding='same')(conv5)
        conv6 = BatchNormalization()(conv6)
        conv6 = Conv2D(64, (3, 3), activation='relu', padding='same')(conv6)
        conv6 = BatchNormalization()(conv6)
        up1 = UpSampling2D((2,2))(conv6)
        conv7 = Conv2D(32, (3, 3), activation='relu', padding='same')(up1)
        conv7 = BatchNormalization()(conv7)
        conv7 = Conv2D(32, (3, 3), activation='relu', padding='same')(conv7)
        conv7 = BatchNormalization()(conv7)
        up2 = UpSampling2D((2,2))(conv7)
        decoded = Conv2D(3, (3, 3), activation='sigmoid', padding='same')(up2)
        return decoded
```

Figure 5 Structure of AE of Level-2 & Level-3 Diagnosis

Data and Experimental Result

- **Data**

We collect the public data from a Kaggle dataset called CoronaHack-Chest X-Ray-Dataset, which is mainly from a project approved by the University of Montreal's Ethics Committee. It is a 1.2 GB collection directly from hospital for the project, including the chest X-ray images of 1576 healthy people, 1493 virus infected patients, 58 covid-19 infected patients, 4 SARS infected patients and 2 stress-smoking people. The data provides image names and labels for these images. We chose this data because image data linked with clinically relevant attributes is quite ideal for machine learning and also meaningful for business.

By using machine learning diagnostic models, doctors can confirm their assessment of a patient's condition in one second. The data also provides images names and labels, which give us a good chance to train our supervised model and an ideal tool to validate our diagnostic performance of both the supervised and self-supervised version model in the end.

- **Experimental Result**

1. Level-1 Diagnosis: Normal / Pneumonia - Binary Classification

(1) Dense-Layer Neural Network (DNN)

In our best dense layer model, we got accuracy around 0.96.

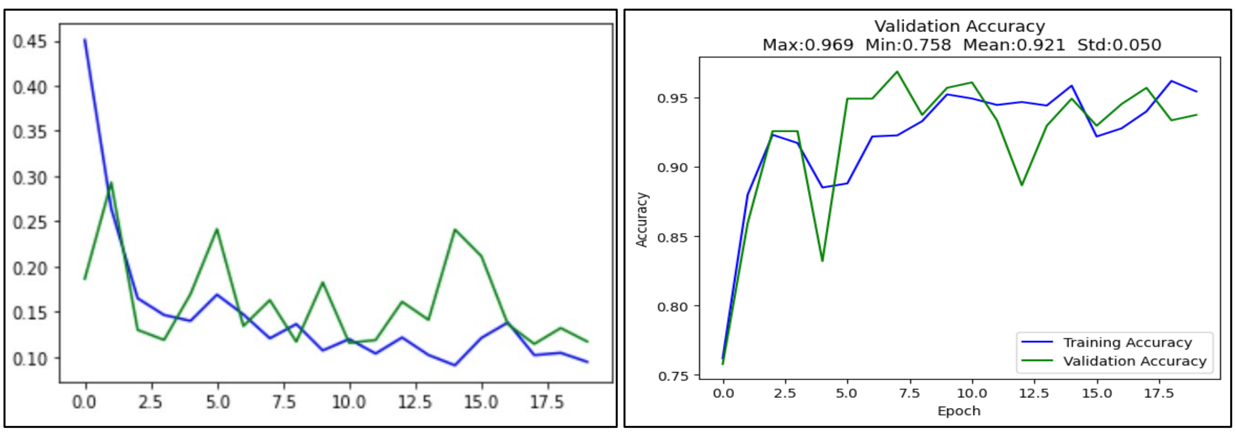

Figure 6 Best Loss and Accuracy of DNN of Level-1 Diagnosis

Model complexity: We tried to build a more complex model since we believe such kind of problem need more parameters to generalize the features which means more layers , but the improvement of accuracy is not obvious, or even not improved, so we choose a three-layer neural network to avoid the over fitting problem caused by the complexity of the model. In the more complex model, the accuracy is still around 0.96, thus we decide to use the 3-layer model by removing some convolution layer and decreasing the nodes.

We have changed the activation function as ReLU or tanh, optimizer as Adam or RMSprop and output layer activation function as softmax or sigmoid. Then we found that only the combination of ReLU and Adam can get high accuracy.

Activation function: We tried a variety of activation functions, and we finally found that using ReLU in hidden layers and sigmoid in the output layer can generate best accuracy. Using sigmoid in hidden layers and SoftMax in the output layer generates underfitting problems and only achieving an accuracy about 0.7. After a long time of adjustment, we determined the final model with ReLU in hidden layers and sigmoid in the output layer.

Optimizer: We tried several kinds of optimizers, Adam optimizer can produce higher accuracy, but Rmsprop optimizer makes the result more stable. Other optimizers got low accuracy in this model. Thus, we choose Adam as our final optimizer.

Loss function: MSE only results in an accuracy around 0.75, less than using binary-crossentropy.

(2) Convolution Neural Network (CNN)

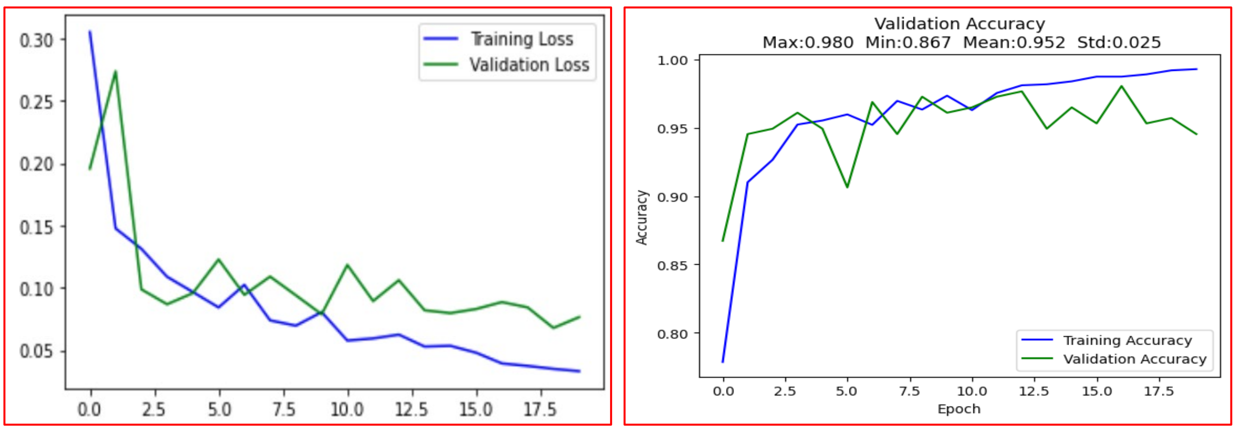

Figure 7 Best Loss and Accuracy of CNN of Level-1 Diagnosis

The situation in CNNs is similar to dense layer models. We keep using Adam optimizer and ReLU activation function. **The average validation accuracy of CNN model is better than DNN.**

(3) Self-Supervised Model - AutoEncoder (AE)

In our best auto-encoder model, we got loss nearly to 0. Then we use the encode to do the classification. However, the accuracy is not higher than CNN model.

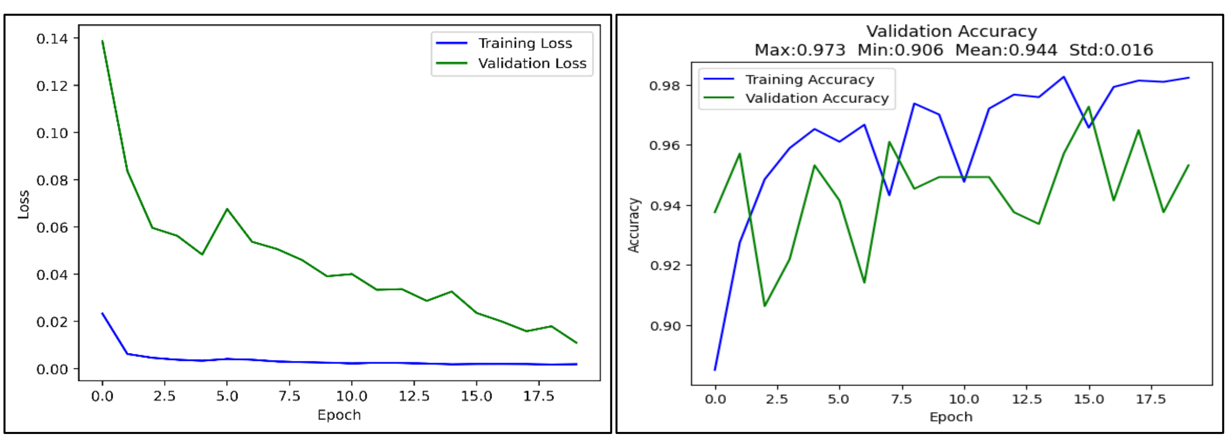

Figure 8 Loss of AutoEncoder and Accuracy of Encoder Classifier of Level-1 Diagnosis

Optimizer: We have tried other optimizers like RMSprop and Adam, other activation functions like tanh and sigmoid and other loss functions like binary-crossentropy, we found that in most models loss was decreasing continuously, but the accuracy remained at 0.3 and did not rise. The reason may be that the selection of the optimizer leads to the final result falling into the local optimal solution, so we change the optimizer to Adam, loss function to mse and activation function to ReLU, the accuracy is improved to close to 1.

Learning rate: By changing the learning rate from 0.0002 to 0.02, we found that the effect of the model for different learning rates is very significantly different. Finally, we found that the model with learning rate of 0.018 has the highest accuracy.

2. Level-2 Diagnosis: Normal / Pneumonia - Binary Classification

As can be seen from the following results, **AE** has the best recognition effect and the highest diagnostic accuracy. **The test accuracy is higher than 0.75.**

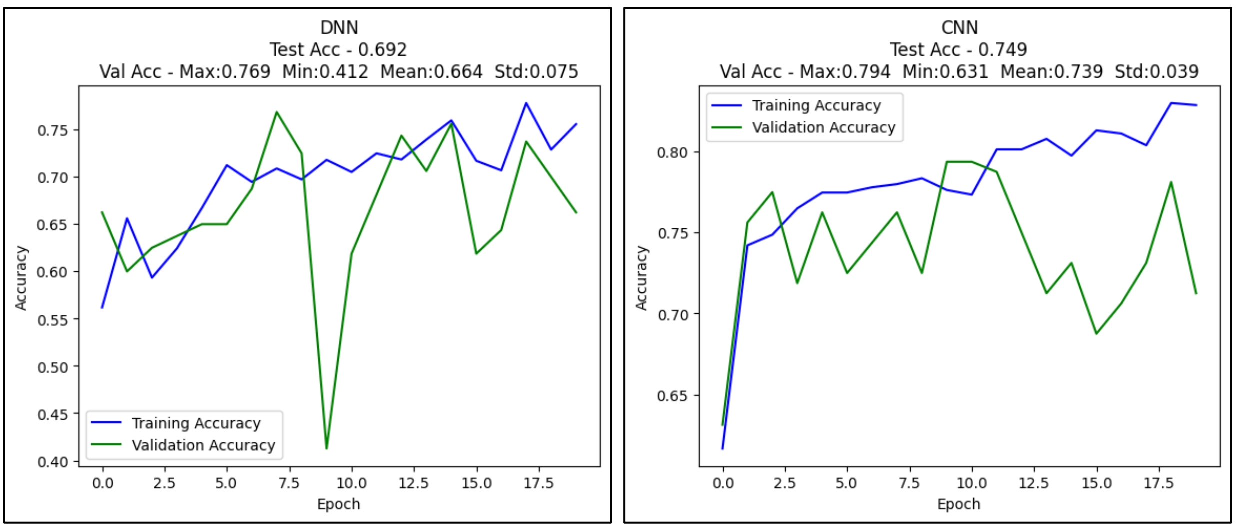

Figure 9 Best Loss and Accuracy of DNN and CNN of Level-2 Diagnosis

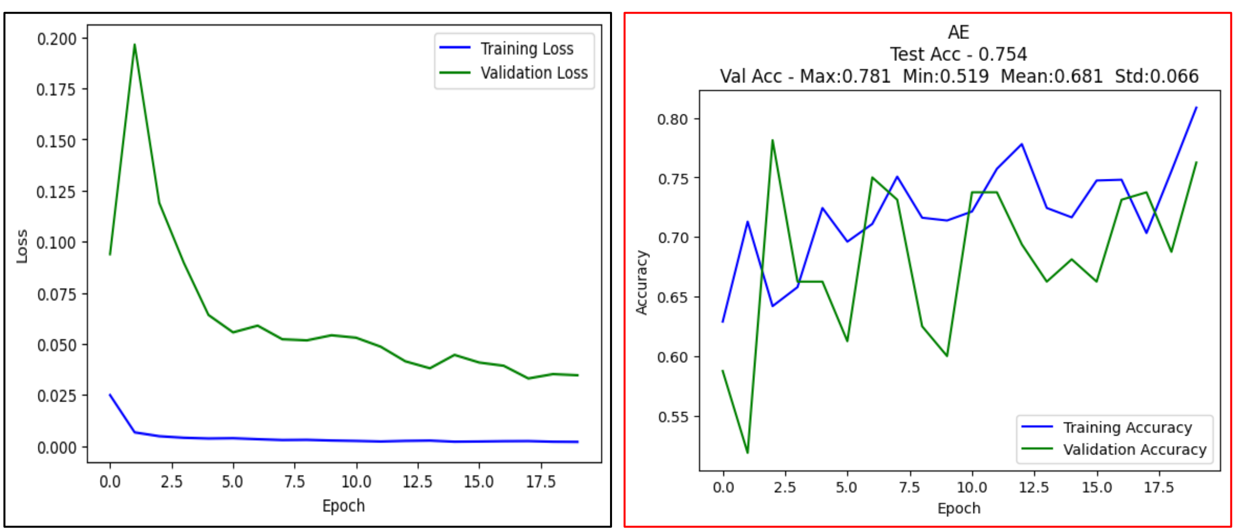

Figure 10 Loss of AutoEncoder and Accuracy of Encoder Classifier of Level-2 Diagnosis

3. Level-3 Diagnosis: COVID-19 / SARS / Other - Multiple Classification

As can be seen from the following results, **CNN** has the best recognition effect and the highest diagnostic accuracy. **The test accuracy of validation is higher than 0.98.**

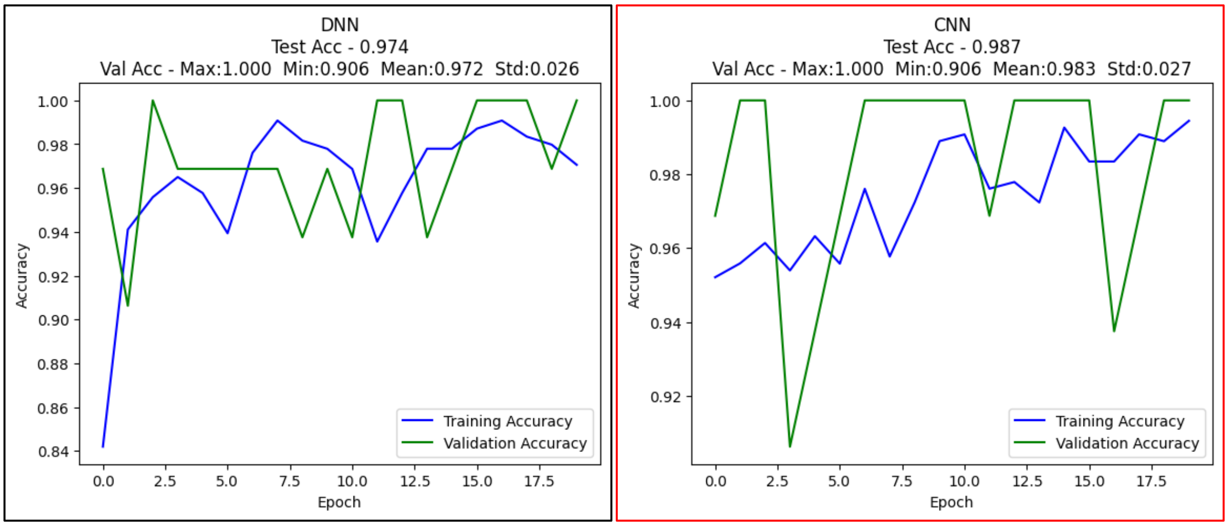

Figure 11 Best Loss and Accuracy of DNN and CNN of Level-3 Diagnosis

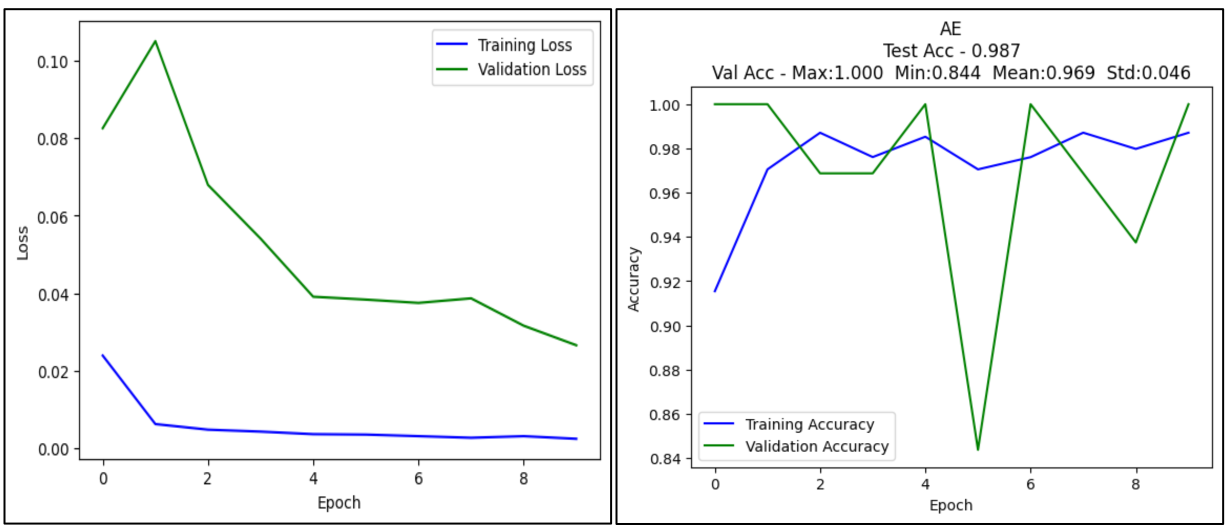

Figure 12 Loss of AutoEncoder and Accuracy of Encoder Classifier of Level-3 Diagnosis

- **Business Application for Model**

By applying the CNN model into business, doctors can confirm their assessment of a patient's condition in a few seconds by feeding the patient's chest X-ray image into the deep learning model. Patients don't need to wait for analysis of radiologist or nucleic acid test.

It can not only improve the speed of diagnosis, but also reduce the cost of pneumonia detection, so the model has a good commercial application value.

Conclusions, Discussions, and Recommendations

In this final project, We try supervised models including dense Layer Model and CNN Model, and self-supervised model using autoencoder. For the binary classification problem, after changing the complexity, activation function, optimizer, loss function, finally we found the average validation accuracy of CNN is around 0.96 and the average validation accuracy of the autoencoder is also higher than 0.98. CNN is the best model for level-3 diagnosis. We can almost perfectly classify the normal healthy people and infected people by using our deep learning models. For the multiple classification problem, After three different models were tried, AE and CNN are the best models for the level-2 and level-3 diagnosis, respectively. However, there is still much room for improvement in the accuracy of the level-2 diagnosis

In order to further improve the accuracy of diagnosis and the reliability of the model, we can further optimize the existing model, such as using more layers of CNN network, better activation function, optimizer and so on. We can also try to improve the resolution of the image and read pixel values. Finally, we can use more complex and sophisticated networks (such as U-NET, ResNet, and so on) to train images in the future.

Video Presentation

The Zoom link of our presentation video for the final project is:

https://wustl.zoom.us/rec/share/JPEh0282OQk_vhd1bA2Y20hsdMwfvPCx27QH94Xs2I24OIE6_ACDFCBFn5iE9qD7.7kf0xME8kA_gUNDS?startTime=1609772846000

The length of the video is about 18 minutes, You can have a quick look at our project. Also, the detailed models and results are shown in the Colab file we submitted on Canvas.

Reference

[1] Chandra, T. B., Verma, K., Singh, B. K., Jain, D., & Netam, S. S. (2020). Coronavirus disease (COVID-19) detection in Chest X-Ray images using majority voting-based classifier ensemble. *Expert systems with applications*, *165*, 113909.

[2] Chen N., Zhou M., Dong X., Qu J., Gong F., Han Y.…Zhang L. Epidemiological and clinical characteristics of 99 cases of 2019 novel coronavirus pneumonia in Wuhan, China: A descriptive study. The Lancet. 2020;395(10223):507–513. doi: 10.1016/S0140-6736(20)30211-7.

[3] Gao, T. (2020). Chest X-ray image analysis and classification for COVID-19 pneumonia detection using Deep CNN. *medRxiv*.

[4] Pesce, E., Withey, S. J., Ypsilantis, P. P., Bakewell, R., Goh, V., & Montana, G. (2019). Learning to detect chest radiographs containing pulmonary lesions using visual attention networks. *Medical image analysis*, *53*, 26-38.

[5] Purohit, K., Kesarwani, A., Kisku, D. R., & Dalui, M. (2020). COVID-19 detection on chest x-ray and ct scan images using multi-image augmented deep learning model. *BioRxiv*.