

Transfer learning approach to Classify the X-ray image that corresponds to corona disease Using ResNet50 pretrained by ChexNet

Mahyar Bolhassani

Istanbul Technical University- bolhassani19@itu.edu.tr

Abstract

Corona virus adversely has affected people worldwide. There are common symptoms between Covid19 virus disease and other respiratory diseases like pneumonia or Influenza. Therefore, diagnosing it fast is crucial not only to save patients but also to prevent it from spreading. One of the most reliable method of diagnosis is through X-ray image of a lung. By the help of deep learning approaches, we can teach the deep model to learn the condition of an affected lung so that it can classify the new sample as if it is a Covid19 infected patient or not. In this project, we trained a deep model based on ResNet50 pretrained by ImageNet dataset and CheXNet dataset. Based on imbalanced CoronaHack Chest X-Ray dataset introduced by Kaggle we applied both binary and multi-class classification. Also, we compare the results when using Focal loss and Cross entropy loss.

Keywords: X-ray classification, Convolutional Neural Network, ResNet, Supervised learning, Transfer learning

1. Introduction

According to World Health Organization (WHO), Covid19 virus disease declared as a pandemic on January 2020. This problem was catastrophic for all nations which its pernicious effects on various domains like economy, politics, education, etc. Therefore, the priority action was to restrict the speed of spreading the disease since the virus is highly contagious. To fulfill this goal, medical systems need a reliable, fast, and accessible tool. The application of deep learning on medical image processing is one of the answers. So, how medical image processing can be helpful. The answer is that one of the most significant symptoms of Covid19 virus on human body is infected lungs which we can trace by taking X-ray scans. Although patients infected by Covid19 have common symptoms with pneumonia, Influenza, and some other respiratory diseases, the X-ray scans are a safe tool to distinguish between them. As a result, during the pandemic, several institutes collected a pack of X-ray scan images from normal to infected patients.

In this project, we starts with a binary classification between bacteria-infected case and virus-infected case. The reason behind it, is that the available dataset is highly imbalance using transfer learning method. In the next step, we apply the same methodology on a 4-label classification. Then, to improve the adverse effect of imbalance dataset, we chose to use Focal loss which its goal is to compensate the effects of imbalance data classes. Based on the results, we tried to improve the performance of the model even better by generating artificial input data using conditional GANs.

2. Method

Automatic classification of X-ray scan images is a challenging task when we have a highly imbalance dataset. Therefore, in this article, we are trying to find the best approach to increase the classification accuracy.

2.1. Benchmark datasets

In this study, we are using CoronaHack Chest X-Ray Dataset [Praveen](#) that has given by the organizer of the challenge to develop a deep model to classify corona infected x-ray scan cases. This dataset consists of four main categories, Normal, Bacteria, Virus, and Covid19 cases. The dataset is highly unbalance, 1575 normal cases, 2778 bacteria infected cases, 1494 virus infected cases, and 82 Covid19 infected cases. We plotted a class distribution diagram which perfectly shows the imbalanced dataset. Also, we showed in another histogram diagram the distribution of training and test samples that has been provided by the competition organizers. The diagram is illustrated in Figure 1.

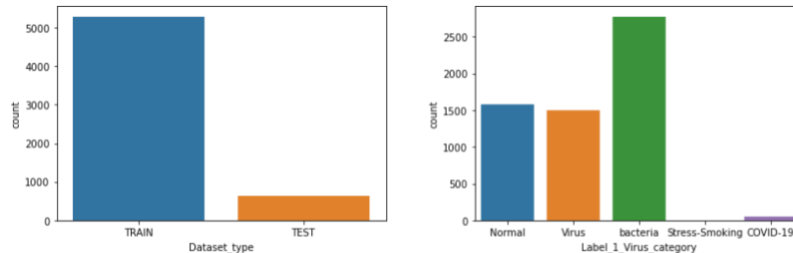


Figure 1: On the left: train-test distribution, on the right: Class distribution of dataset

In order to have better understanding about the input dataset, it is a good attempt to plot them to see how they are different to each other. The Figure 2, shows image two samples from each of the four class types and the intensity histogram distribution of each.

Based on the Figure 2, we can conclude that the intensity distribution of samples from on class differ from other classes, which is a good parameter that shows we can successfully train a deep model to classify them.

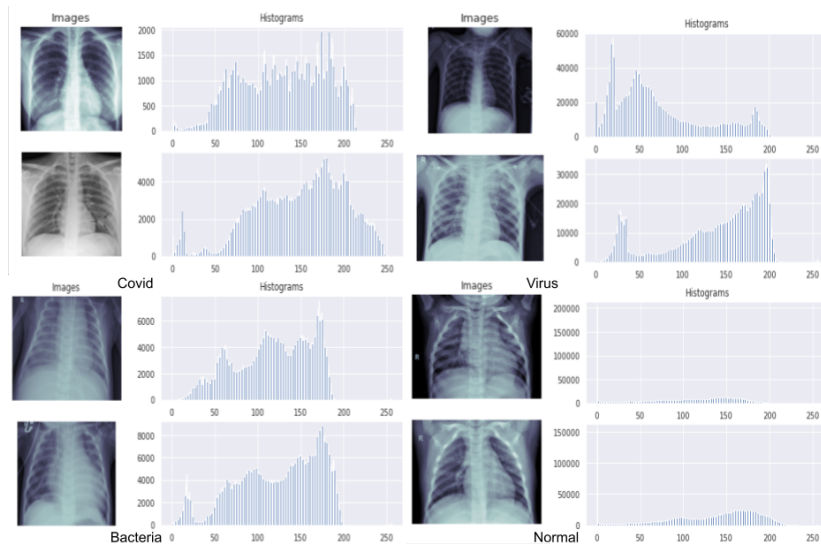


Figure 2: Histogram distribution of data samples

2.2. Data augmentation

Deep learning approaches demand a large number of annotated sample data in order to have an accurate prediction on test samples. Labeled medical images are hard to achieve, expensive, and time consuming, since we need the help of clinicians for annotating or labeling medical images. In our case, we need medical experts to classify lung X-ray scans varies from normal to Covid19 infected x-ray scans. Therefore, we have to come up with a solution to tackle the scarcity of the training samples.

One way to overcome this hurdle is data augmentation technique. By applying this method we will be able to improve the performance of the model by having variations on the training sample so that our model have the ability to see sample with different variations. In this project, we applied several augmentation techniques including random horizontal and vertical flip each with probability of 50%, and random rotation with probability of 30%.

2.3. Architecture

So as to train a model to learn classifying X-ray scan images of lungs, we decided to choose two different architectures based on the medical image processing literature, DenseNet, and ResNet. Both of them are shown that perform well on medical data.

2.3.1. ResNet

ResNet is developed to solve the vanishing gradient as a result of a deep convolutional network. This architecture provides a path for gradient to follow through the skip connections between blocks. A diagram which shows the detail implementation of ResNet [He et al. \(2015\)](#), is illustrated in figure 3.

2.3.2. DenseNet

The traditional convolutional neural network has a problem in a case when we use a much deeper network. In this case, since the path of information from input to the output become very long (as a result of a deep network) for both forward and backward path, it is likely to face vanishing gradient. DenseNet Huang et al. (2018) was introduced to solve the issue mentioned above. In this architecture, the output of each layer is passed to the input of all other layer. This way, not only does the gradient vanishing solve, but also we need to manage fewer parameter in compare to the vanilla CNN. The more details of DenseNet architecture is shown in the Figure 3 on left image.

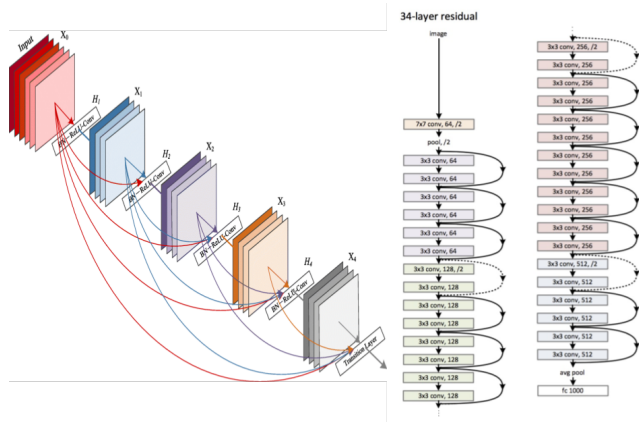


Figure 3: On the left:DenseNet, on the right: ResNet

2.4. Loss function

Next step is defining our loss function for the training part. We know that choosing the right loss function which is a hyperparameter depends on the problem we are facing. Therefore, to tackle a multi classification problem multi class cross entropy loss (Categorical Cross Entropy Loss Function) seems a wise choice. Equation (1) is the formula for this loss function.

$$CE = - \sum_{i=1}^{C'=2} t_i \log(s_i) = -t_1 \log(s_1) - (1 - t_1) \log(1 - s_1) \quad (1)$$

Though looking deeper to the problem we have, something forces us to be more cautious and it is the imbalanced data samples that we have as the input of our model. Focal loss Lin et al. (2018) is another choice that we can leverage its properties to enhance the performance of our model. this loss function tries to generate class weighting system in order to balance the samples in each batch size of data. Equation (2) shows the details regarding this function.

$$FL(pt) = -\alpha t(1 - pt)^\gamma \log(pt) \quad (2)$$

2.5. Metric

One of the most important steps in the deep learning process is to define metrics which means how we evaluate the performance of our model. We chose to use accuracy which is the simplest metric. Accuracy is defined by finding the the numbers of predictions per class divided by all numbers of predictions on each epoch. The average accuracy that we report in this project is obtained by averaging accuracy of each epoch.

2.6. weighted class

Our dataset is not balance so that we are not sure that in each mini batch of data all class samples exist or not. Therefore, one of our solution for mentioned problem is to use `weightedRandomSampler` function in Pytorch. We first calculated the numbers of samples in each class. Then, we got the reciprocal value of them and considered them as weights of each class. In the next step, we passed the class weights to each class labels.

3. Detail implementation

Transfer learning is our dominant approach to classify covid-19 infected cases. According to the fact that we have a small and imbalance dataset in which covid-19 class is considerably smaller than other classes. This issue can deteriorate our prediction, as a result we need to seek for a method to alleviate the unbalanced class distribution of the classes.

We started with ResNet50 architecture pretrained by ImageNet dataset. Then, we applied weight sampling approach to have equal number of samples from each class. We, also, examine using both categorical cross entropy and focal loss to see the effect of focal loss on our imbalanced dataset. Although our attempts paid off and the accuracy on both training and validation increased, still we tried to improve the accuracy of the model. We know that medical images are different from natural images, for instance, medical images are mostly in gray scale. Hence, a question rises here whether using a model pretrained by ImageNet which is a collection of natural images, is a wise choice or not. This question motivated us to research on this issue so that we ended up to [Mangal et al. \(2020\)](#) in which they pretrained their model using CheXNet. CheXNet [Rajpurkar et al. \(2017\)](#) is a model based on DenseNet architecture that trained on a large brain MRI scan images. We decided to use the same approach to check if it improve the performance of our model or not. Finally, we add some covid-19 images to the dataset (our dataset has a very limited number of covid-19 samples).

For training the model, we chose epoch number to 20 for time and GPU restrictions. Also, we used Adam optimizer with initial learning rate with the value of 0.001. Learning rate is updating every 10 epochs and multiplying with a factor of 0.5 in order to avoid vanishing or exploding gradient.

4. Experimental results

In order to show our results, we divided each selection of hyperparameters into a section. This will allow us to explain the output and drawbacks of each in detail. We considered Adam optimiser with the learning rate scheduler explain in section 3. In all figures in this section, images on the left represent Training and validation accuracy vs number of epochs, while the images on the right illustrates Training and validation loss vs number of epochs.

The experimental results of each subsection are gathered in table 1.

Table 1: Accuracy results for each proposed model

Models	PRCE	PRCW	PRFL	PDCXCE	PDCXFL	RCE
Tr Acc.	81.06	86.06	85.84	85.28	87.95	81.608
Val Acc.	65.39	73.43	75.0	76.062	73.87	69.93
Tst Acc.	62.1	80.2	75.5	80.0	75.56	71.11

According to the results provided in Table 1, pretraining DensNet on CheXNet dataset outperforms other models.

4.1. ResNet pretrained on ImageNet

4.1.1. Considering CE loss (PRCE)

In this section, we train our model based on ResNet50 pretrained by ImageNet. The loss function is categorical cross entropy.

4.1.2. Considering CE loss and weighted-class (PRCEW)

In this section, we train our model based on ResNet50 pretrained by ImageNet. The loss function is categorical cross entropy while including weighted classes to the model to decrease the adverse effects of imbalanced data on the results.

4.1.3. Considering FL loss (PRFL)

In this section, we train our model based on ResNet50 pretrained by ImageNet. The loss function is focal loss to reduce imbalanced distribution of sample data.

4.2. DenseNet pretrained on CheXNet

4.2.1. Considering CE loss (PDCXCE)

In this section, we train our model based on DenseNet121 pretrained on CheXNet considering CE loss.

4.2.2. Considering FL loss (PDCXFL)

In this section, we train our model based on DenseNet121 pretrained by CheXNet considering FL loss.

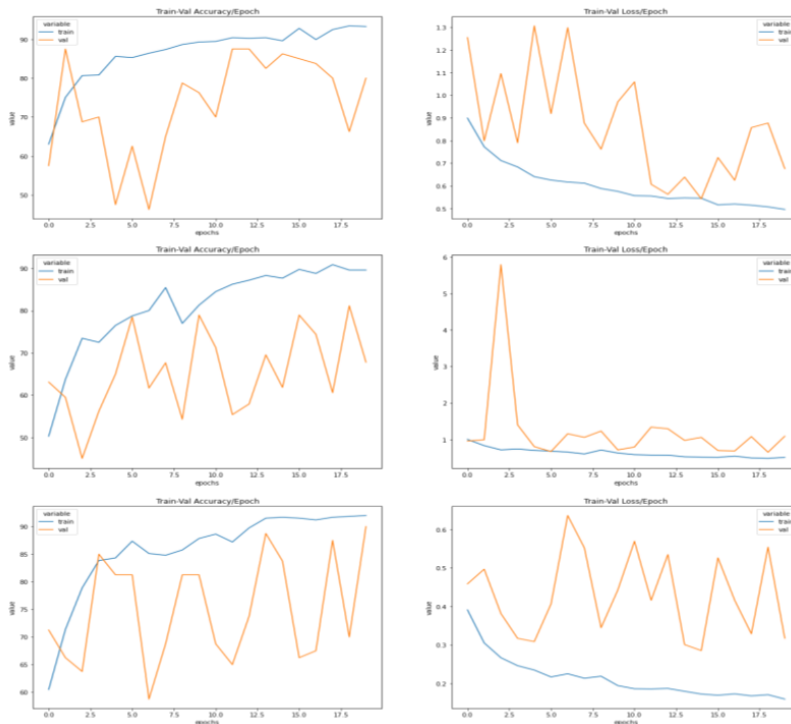


Figure 4: First row: ResNet pretrained model with CE loss, Second row: ResNet pretrained model with weighted class CE loss, Third row: ResNet pretrained model with FL loss

4.3. ResNet50 (RCE)

In this section, we train our model based on ResNet50 without considering transfer learning. The loss function is focal loss to reduce imbalanced distribution of sample data.

4.4. Extra covid-19 samples

Limited numbers of covid-19 samples is a problematic facts that prevent our model to perform at its best. Thus, at fist we intended to develop a GANs model to generate covid-19 images. Though, because of lack of enough time and GPU memory, and the fact that GANs models are very sensitive to the hyper-parameters our developed GANs couldn't generate accurate images. Therefore, we decided to add some existing covid-19 X-ray images on the Internet to our dataset. The results shown in Table 2.

5. Conclusion

We examined several approaches to classify highly imbalanced dataset provided by Kaggle. We divided our solution into three main categories: pretrained

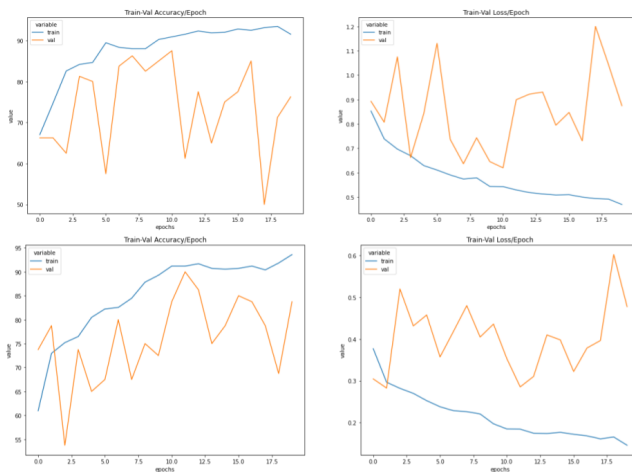


Figure 5: First row: DenseNet pretrained on CheXNet with CE, Second row: DenseNet pretrained on CheXNet with FL

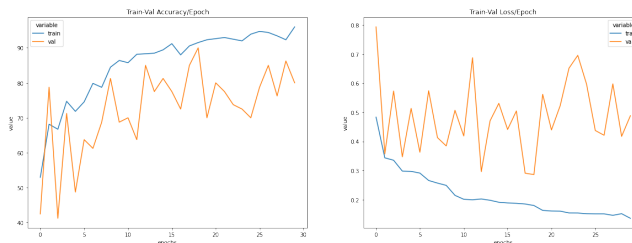


Figure 6: ResNet50 without pretraining considering FL

ResNet50 on ImageNet, pretrained DenseNet121 on CheXNet, and ResNet50 from scratch. We wanted to compare them all to get an idea about which approach is the best choice for lung X-ray scan images. based on results and our first impression, pretrained models on CheXNet have better performance on our dataset and outperformed other scenarios. Also, the results show that using focal loss improves the performance on test and validation set since the duty of this loss is to reduce the imbalanced data as input. Even though, categorical cross entropy achieves better training accuracy. Unfortunately the numbers of covid-19 samples are too limited that act as a hurdle in obtaining better results. Therefore, we add extra covid-19 samples from another challenge in Kaggle. Applying the models we developed in section 4 on the extended dataset shows a significant improvement.

Table 2: Accuracy results after adding extra covid-19 samples

Models	PRCE	PDCXCE	RCE
Tr Acc.	91.7	91.37	88.48
Val Acc.	78.75	81.83	80.26
Tst Acc.	72.69	84.6	71.11

References

- He, K., Zhang, X., Ren, S., Sun, J., 2015. Deep residual learning for image recognition. [arXiv:1512.03385](https://arxiv.org/abs/1512.03385).
- Huang, G., Liu, Z., van der Maaten, L., Weinberger, K.Q., 2018. Densely connected convolutional networks. [arXiv:1608.06993](https://arxiv.org/abs/1608.06993).
- Lin, T.Y., Goyal, P., Girshick, R., He, K., Dollár, P., 2018. Focal loss for dense object detection. [arXiv:1708.02002](https://arxiv.org/abs/1708.02002).
- Mangal, A., Kalia, S., Rajgopal, H., Rangarajan, K., Namboodiri, V., Banerjee, S., Arora, C., 2020. Covidaid: Covid-19 detection using chestx-ray. arXiv 2004.09803 URL: <https://github.com/arpnmangal/CovidAID>.
- Praveen, . Coronahack chest x-ray dataset classify the x ray image which is having corona. URL: <https://www.kaggle.com/praveengovi/coronahack-chest-xraydataset?select=Coronahack-Chest-XYray-Dataset>.
- Rajpurkar, P., Irvin, J., Zhu, K., Yang, B., Mehta, H., Duan, T., Ding, D., Bagul, A., Langlotz, C., Shpanskaya, K., Lungren, M.P., Ng, A.Y., 2017. Chexnet: Radiologist-level pneumonia detection on chest x-rays with deep learning. [arXiv:1711.05225](https://arxiv.org/abs/1711.05225).