Pneumonia Detection from Chest X-ray Images Using Deep Learning

Final Report

Halis Yigin

Table of Contents

1. Introduction
    1.1 Who is your client, and why do they care about this problem?
2. Dataset
3. Data preparation
    Label Images
    Resize Images
    Data Augmentation
4. Data Story
5.Modelling
6. Transfer Learning
7. Limitations
8. Conclusion and Next Steps

1. Introduction

Pneumonia is a very common type of infection in the world. More than 1 million adults are hospitalized with pneumonia and around 50,000 die from the disease every year in the US alone (CDC, 2017). The infection spreads in the lungs area of a human body. Pneumonia can be monitored via the chest X-ray(CXR). Physicians use this X-ray image to diagnose or monitor treatment for conditions of pneumonia. Also the chest X-rays is used in the diagnosis of different diseases like emphysema, lung cancer, line and tuberculosis.

The diagnosis of pneumonia on CXR is complicated because of a number of other conditions in the lungs such as fluid overload, bleeding, volume loss, lung cancer. In addition, clinicians are reading high volumes of images every shift. Sometimes being tired or distracted clinicians can miss some details in image. Pneumonia detection using deep learning can help the clinicians to diagnose the disease.

1.1 Who is your client, and why do they care about this problem?

Physicians can get help to diagnose the pneumonia from the algorithm. The algorithm helps them to separate pneumonia from other diseases. Physicians may see the details if they miss any detail in CXR. Hospitals can use the algorithm to improve the accuracy of the diagnosis.

2. Dataset

I will use the dataset of Chest X-Ray Images (Pneumonia) from kaggle. It is a dataset of chest X-Rays.

Dataset includes three folders which are test, train, val. In addition, each folder includes two sub folders which are ‘normal’ and ‘pneumonia’.

- Test/Normal has 234 images.
- Test/Pneumonia has 390 images.
- Train/Normal has 1341 images.
- Train/Pneumonia has 3875 images.
- Val/Normal has 8 images.
- Val/Pneumonia has 8 images.

3. Data preparation

Label Images

The dataset is divided into three sets: 1) train set 2) validation set and 3) test set. Each of the above directory contains two sub-directories:

NORMAL: These are the samples that describe the normal (no pneumonia) case.

PNEUMONIA: This directory contains those samples that are the pneumonia cases.

I inserted the train data into this list in (image, label) format and I created a data frame of train data. I also make a similar list for test data to make predictions.

Resize Images

Images in the dataset are different sizes. In classifying an image using a convolutional neural network (CNN), the input must be the same size for all images. Therefore, I resize the training images to dimensions(150,150) to train the model properly. Also, I resize the test images with the same dimensions to make a more accurate prediction.

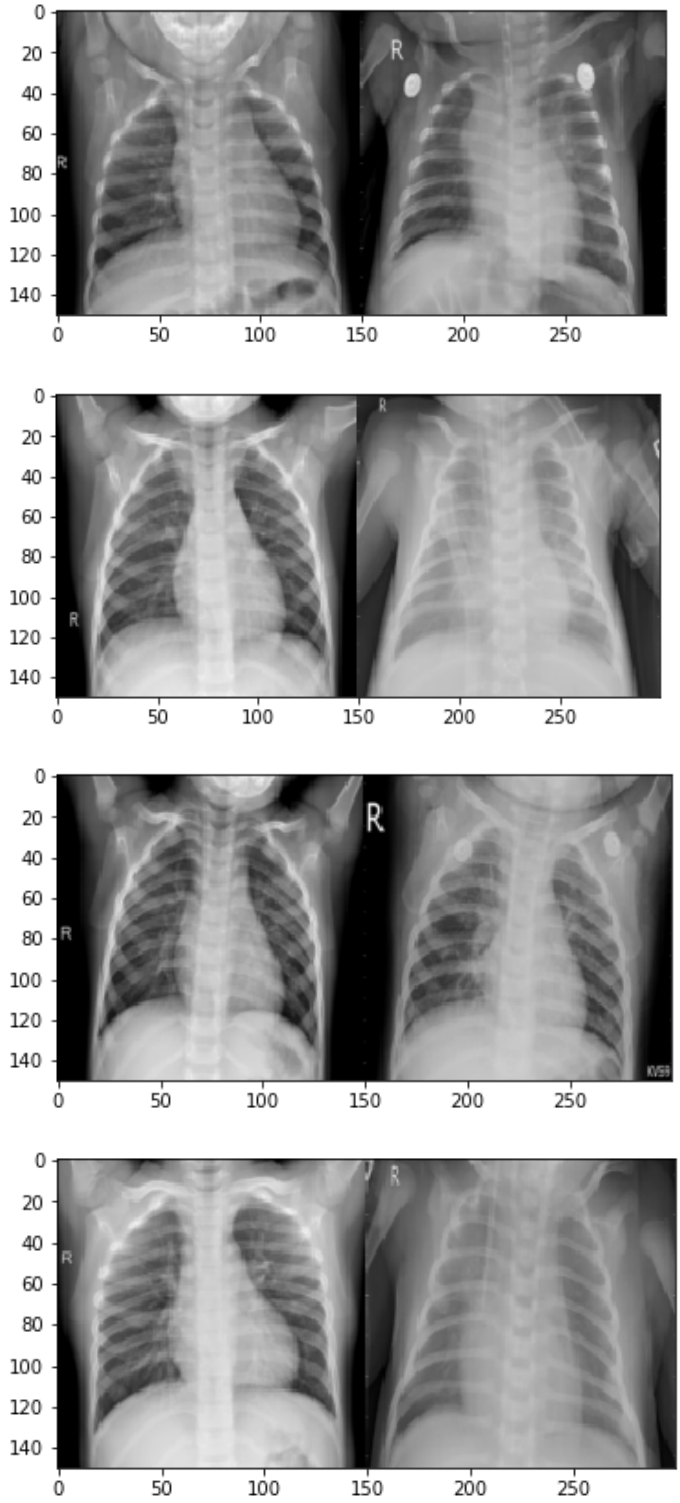
Data Augmentation

The practice of data augmentation is an effective way to increase the size of the training set. Augmenting the training examples allow the network to “see” more diversified, but still representative, data points during training. I applied data augmentation to the training and validation datasets. A data generator is capable of loading the required amount of data (a mini batch of images) directly from the source folder, converting them into *training data* (fed to the model). At testing time I do not apply data augmentation and simply evaluate my trained network on the unmodified testing data.

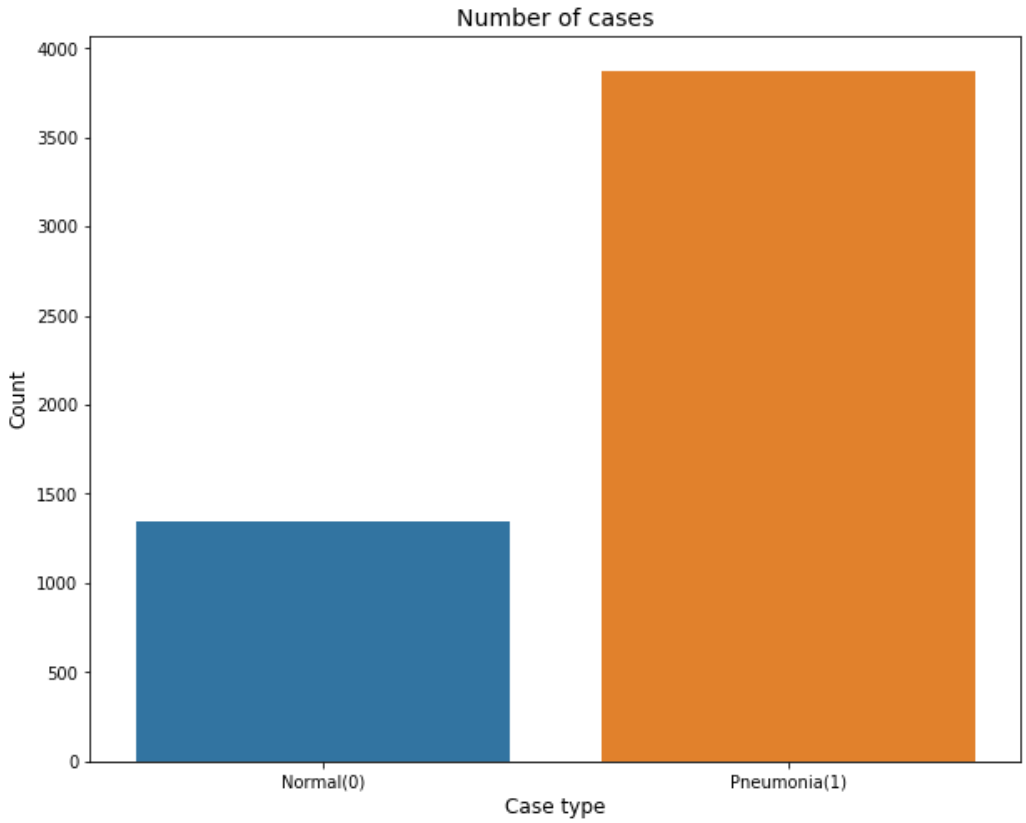
4. Data Story

I displayed some normal and pneumonia images to just have a look at how much different they look from the naked eye. It’s not easy to detect pneumonia if you are not a clinician.

(Left) - No Pneumonia vs (Right) - Pneumonia



I checked the training data images. There are 3875 pneumonia images and 1341 normal images in the train dataset.



5.Modelling

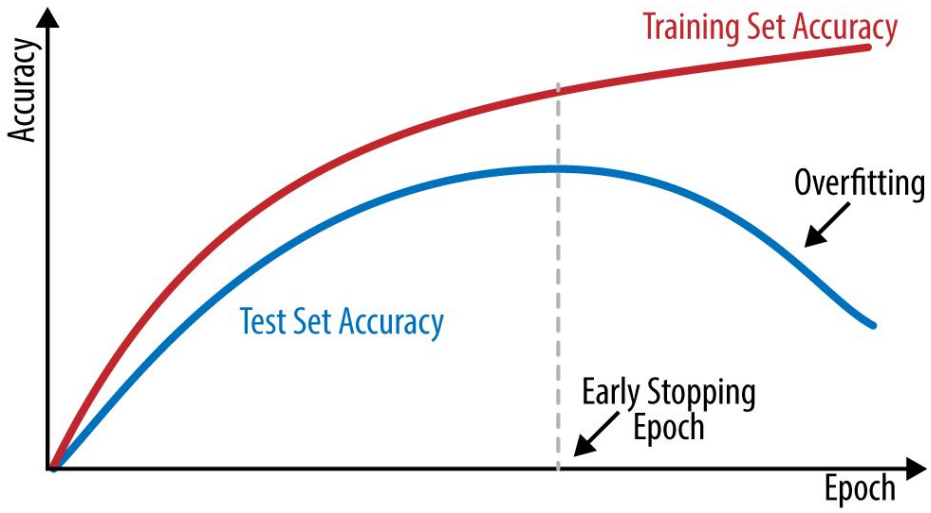
I built a CNN(convolutional neural network) model. This can be described in the following steps.

- I used five convolutional blocks with convolutional layers and max-pooling.
- I used a flatten layer and followed it by fully connected layers.
- Also I have used dropout to reduce overfitting.
- Activation function was Relu throughout except for the last layer where it was Sigmoid as this is a binary classification problem.
- I have used RMSprob as the optimizer and cross-entropy as the loss.

Before training the model I define some callbacks which are ModelCheckpoint and EarlyStopping.

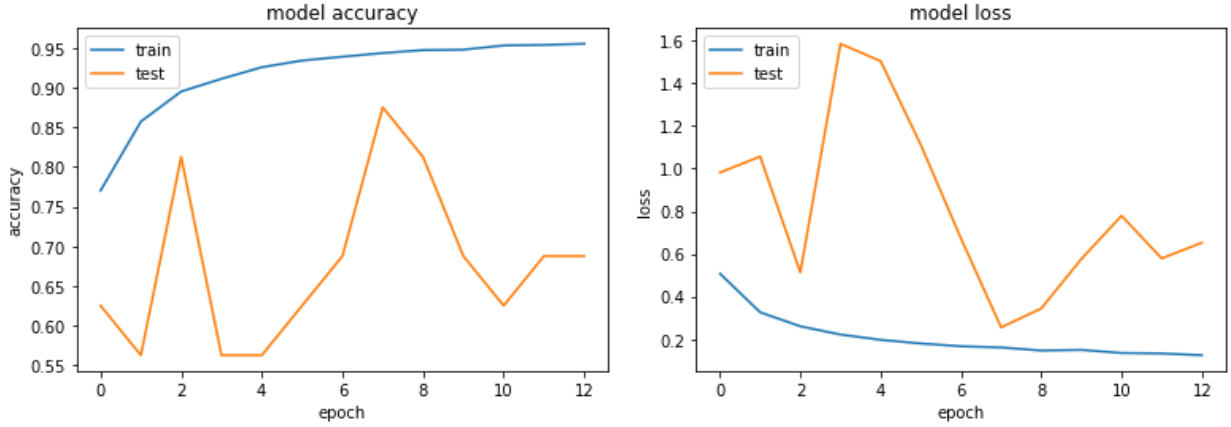
ModelCheckpoint helps us save our model for each epoch, so we can put our model to train and not worry about possible issues that might happen.

EarlyStopping is basically stopping the training once your loss starts to increase (or in other words validation accuracy starts to decrease)



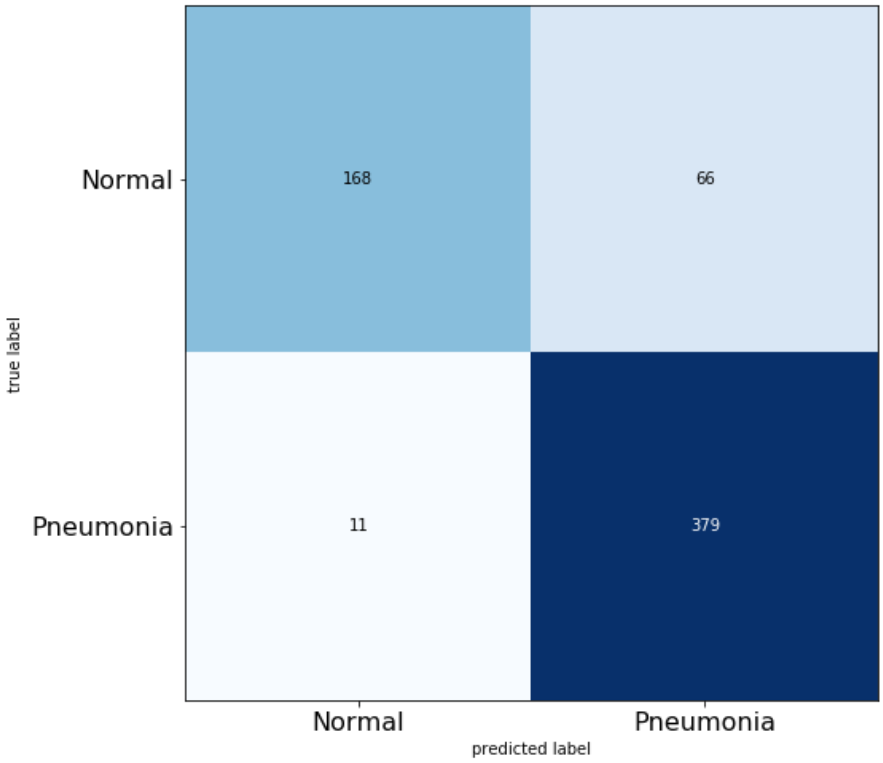
I trained the model for 20 epochs with a batch size of 16. Please note that usually a higher batch size gives better results but at the expense of higher computational burden. My model stops at 13 epochs because after that point, the model will be overfitting.

I visualize the loss and accuracy.



The model is able to reach 95% validation accuracy in just 13 epochs. Also, I plot the confusion matrix and get some of the other results also like precision, recall, F1 score and accuracy.

Confusion Matrix for the model from scratch



Accuracy of the model is 0.88

Recall of the model is 0.97

Precision of the model is 0.85

f1_score of the model is 0.91

- The **accuracy** means the fraction predicted correctly to total data.
- The **recall** means “What percent of the positive cases did you catch?”
- The **precision** means “What percent of your predictions were correct?”
- The **f1-score** is the harmonic mean between precision & recall. It means “What percent of positive predictions were correct? “

The model is able to achieve an accuracy of 88% which is quite good considering the size of data that is used.

6. Transfer Learning

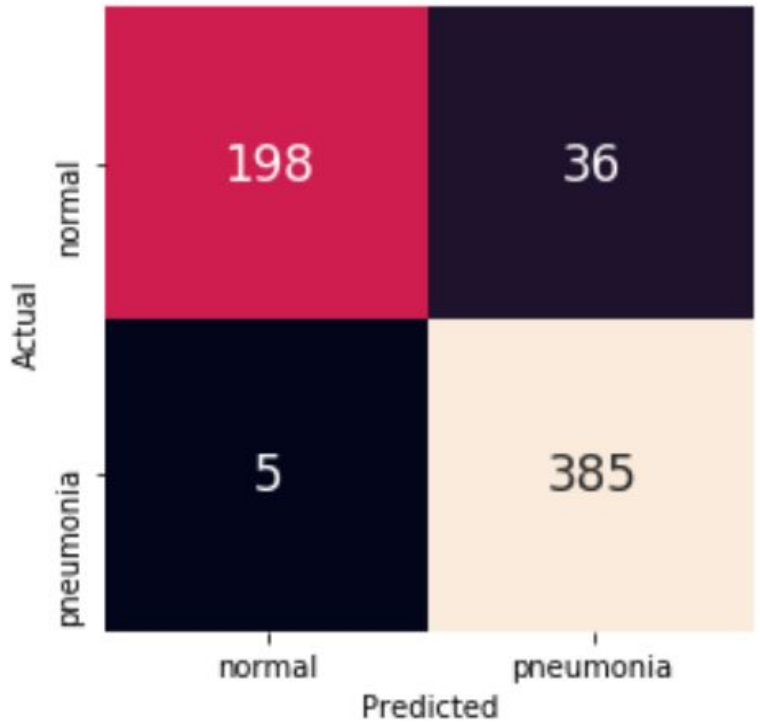
Transfer learning is a research problem in machine learning that focuses on storing knowledge gained while solving one problem and applying it to a different but

related problem. I applied transfer learning with different models. The models are VGG16, VGG19, InceptionV3, ResNet50. Also, I used fine-tuning on two convolutional layers which are block4_conv1 and block5_conv1 in VGG16 and VGG19 models. When we compare them, I got the best result from the ResNet50 model. I also add some layers but my best result from these transfer learning are following:

Comparison of Deep Learning Models

	Accuracy	Recall	Precision	f1-score
Model from scratch	0.88	0.97	0.85	0.91
VGG16	0.75	0.82	0.78	0.80
VGG16 fine-tune	0.73	0.87	0.74	0.80
ResNet50	**0.93**	**0.98**	**0.91**	**0.94**
InceptionV3	0.82	0.98	0.78	0.87
VGG19 fine-tune	0.75	0.83	0.78	0.80

Confusion Matrix for the ResNet50 Model



7. Limitations

When we check the training dataset, the number of the pictures could be larger. Normal pictures in training data sets are not enough for good prediction. I could try only a few models to improve the accuracy of the model because of computational difficulties.

8. Conclusion and Next Steps

I used a CNN model from scratch and transfer learning models which are VGG16, VGG19, InceptionV3, ResNet50. The best model was the model ResNet50 which has good accuracy which is 93% to detect pneumonia disease. If we have enough data for the reason for pneumonia such as bacteria, virus, and fungi, we can categorize the pneumonia reasons. Chest X-ray images can be used to detect pneumonia and other diseases like lung cancer, emphysema, and tuberculosis. I used the images for pneumonia detection. In the future, we can use the images with different models to detect other diseases. Especially lung cancer has been a big issue recently.