CLASSIFICATION OF CHEST X-RAYS INFECTED WITH PNEUMONIA USING TRANSFER LEARNING

Saurabh Purkar (G01275131)

December 12, 2021

Abstract

Pneumonia is a form of acute respiratory infection that is most caused by viruses or bacteria. It can cause mild to life-threatening illness in people of all ages, however it is the single largest infectious cause of death in children worldwide. The most used method for diagnosis of pneumonia is Chest X-ray, however the examination of chest X-rays is susceptible to different representations and is very challenging. Thus, there is need for a solution to aid radiologists diagnose pneumonia quickly this reducing mortality rates among children significantly in countries that lack equipment and skilled radiologists. In this project I worked on a transfer learning method that distinguishes between normal chest X-rays and chest X-rays with pneumonia and further into different types of pneumonia. Also, transfer learning-based solution achieves higher training and validation accuracy and converges faster particularly on smaller dataset. I used different pretrained models on ImageNet dataset such as VGG16, VGG19, DenseNet201, Inception Net and achieved an accuracy of 82.89% on unseen test data.

1. Introduction

Pneumonia is a respiratory infection that causes inflammation in one or both the lungs and may be caused by a virus, bacteria, fungi or other germs. According to global burden of disease studies, 2.56 million people died from pneumonia in 2017. Almost a third of all victims were children younger than 5 years, it is the leading cause of death for children under 5. People at risk for pneumonia also include adults over the age 65 and people with preexisting health problems. The diagnosis of pneumonia can be done by several radiology-based methods such as chest X-ray, CT scan, MRI. The most frequently used amongst them is chest X-ray imaging as its effective and economical. When interpreting the X-ray, the radiologist will look for white spots (called infiltrates) that identify an infection. However, the analysis of x-rays is a very challenging task is susceptible to different interpretations as the very attributes which determines the existence of pneumonia often get mixed with other diseases. Thus, there is a need for an intelligent solution which can aid the radiologists decide regarding the existence of pneumonia. The deep learning algorithms has the potential to replace the traditional manual methods of analysis of X-rays and aid the doctors and radiologists to judge accurately weather the X-ray has pneumonia or not and further if it does have pneumonia to classify it into two classes i.e., viral and bacterial.

2. Problem Statement

The objective of the project are as follows:

1. Classify between normal chest X-rays and chest X-rays infected with Pneumonia.
2. Classify between viral Pneumonia and bacterial pneumonia.
3. Use transfer learning and compare the results with basic CNN to check for performance improvement.

2.1 Notations

CNN : Convolutional Neural Network.
VGG : Visual Geometry Group

DenseNet : Densely Connected Convolutional Networks
TP : True Positive
TN : True Negative
FP : False Positive
FN : False Negative

3. Literature Review

Machine learning techniques have been used extensively in health informatics and diagnosis of diseases from medical images. Lot of work has been done for pneumonia detection using chest X-rays. For e.g., Rohit Kundu et al. [1] came up with an approach where they used an ensemble of deep learning models to detect pneumonia. They classified the images into normal and pneumonia images by using ensemble of three convolutional neural network models: GoogleNet, ResNet-18 and DenseNet-121. A weighted average ensemble technique was adopted, wherein the weights assigned to the base learners were determined using a novel approach. Stephen et al. [2] proposed a convolutional neural network model trained from scratch to classify and detect the presence of pneumonia. Also, they used data augmentation to compensate for the lack of data. To evaluate the effect to dataset size on the performance of CNN, they trained the proposed CNN's using both the original as well as augmented data. Yusuf et al. [3] developed a transfer learning-based approach to detect and diagnose pneumonia induced by Covid-19. They used three CNN architectures VGG16, DenseNet-121, ResNet-50 to carry out experiments. Jaiswal A.K et al. [4] developed an identification model based on Mask-RCNN, a deep neural network which incorporates both global and local features for pixel-wise segmentation. They used image augmentation, alongside with dropout and regularization. Juan E., et al. [5] Used Xception network pretrained weights on ImageNet dataset as initialization and classified between normal and pneumonia chest X-rays.

4.Methods and Techniques

4.1 Data Preprocessing

Each image is in JPEG format with resolution of 2090 X 1858. Used 'Img_to_array ' function from keras library to convert PIL image instance into a numpy array so that the images can be used with deep learning models to make processing faster. The image was converted to an array of size 224 X 224 and has three channels for red, green, blue (RGB), hence the image size is (224,224,3) with dtype float 32. Also scaled these images with pixel values between (0,255) to values between (0,1) because models work well with small input values.

As part of one of the experiments, I used Image Augmentation where I used 'ImageDataGenerator' from keras library to dynamically augment the images while passing the data in bataches during training. The dataset had two subfolders for normal and pneumonia images. By using the images names, I segregated the pneumonia subfolder for training and validation data into two folders viral and bacterial such that we have three subfolders normal, viral and bacterial. The 'flow_from_directory' function takes the path to a directory and generates batches of augmented data.

The dataset is imbalanced as the number of bacterial pneumonia images is quite high than the number of viral pneumonia images. Class imbalance represents an important problem for intelligent classification algorithms. The goal is to identify bacterial/viral pneumonia, but we don't have many of those viral pneumonia samples and so we would want to have the classifier heavily weight the few examples that are available. This will cause the model to "pay more attention" to examples from an under-represented class. To handle this, I calculated weights for each class using the 'class-weight' method from 'sklearn' library. While implementing CNN models using Keras we can pass the weights for each class using the class-weight parameter in model fit method.

4.2 Transfer Learning

Transfer learning is a machine learning method where a model developed for a task is reused as the starting point for a model on a second task. In contrast to traditional machine learning algorithms where knowledge gained is not retained, the transfer learning enables us to utilize the knowledge gained from the previously learned tasks and apply them to newer tasks.

For e.g., as seen in figure above, there is CNN model which is pretrained on ImageNet dataset which can have 1000 categories. The knowledge learned from this step can be applied on target dataset to extract meaningful features. For e.g., we have DenseNet201 model pretrained on ImageNet dataset and then we use this model on target dataset which in our case is chest X-ray dataset and on that we add a Pooling layer, dropout layer and finally, a dense layer with Sigmoid activation for making predictions. Depending on both the size of the new dataset and the similarity of the new dataset to the original dataset, the approach for using transfer learning will be different.

Transfer learning is much more effective when source and target tasks possess similar low-level features. It also works well on small datasets and that is the reason I choose transfer learning for this project. Due to use of use of transfer learning the initial skill on the source model is higher than it otherwise would be. Also, the rate of improvement of skill during training of the source model is steeper than it otherwise would be. Deep learning models follow layered architecture, with different layers learning different features, the initial layers capture generic features and therefore can be used with any other classification model while the deep layers are focused more on the specific features of the data. We can use pretrained models as feature extractors by freezing the convolution blocks or we can fine tune certain layers according to the target domain.

4.3 Methods Used

In this project I have used multiple well-known pretrained deep learning CNNs such as VGG16, VGG19, DenseNet201, Inception Net and then used them on chest X-ray dataset.

4.3.1 VGG16

VGG16 is a CNN architecture which was used to win ILSVR(Imagenet) competition in 2014. instead of having a large number of hyper-parameters they focused on having convolution layers of 3x3 filter with a stride 1 and always used same padding and maxpool layer of 2x2 filter of stride 2. It follows this arrangement of convolution and max pool layers consistently throughout the whole architecture. In the end it has 2 FC(fully connected layers) followed by a softmax for output. The 16 in VGG16 refers to it has 16 layers that have weights. This network is a large network, and it has about 138 million (approx) parameters.

4.3.2 VGG19

The concept of the VGG19 model (also VGGNet-19) is the same as the VGG16 except that it supports 19 layers. The "16" and "19" stand for the number of weight layers in the model (convolutional layers). This means that VGG19 has three more convolutional layers than VGG16. The model consists of 19 layers (16 convolution layers, 3 Fully connected layer, 5 MaxPool layers and 1 SoftMax layer). Fixed size of (224 * 224) RGB image was given as input to this network i.e., matrix of shape (224,224,3). It uses kernels of (3 *3) size with a stride size of pixel 1, this enabled them to cover the whole notion of the image.

4.3.3 DenseNet201

DenseNet, which is a short form of the Dense Convolutional Network, needs less numbers of parameters than a conventional CNN, as it does not learn redundant feature maps. The layers in DenseNet are very narrow, i.e., 12 filters, which add a lesser set of new feature-maps. Each layer in DenseNet has direct access to the original input image and gradients from the loss function. Therefore, the computational cost significantly reduced, which makes DenseNet a better choice for image classification. For this project I have used DenseNet-201version of DenseNet which is network 201 layers deep.

4.3.4 InceptionNet

The Inception V3 model allows for increasing the depth and width of the deep learning network but maintaining the computational cost constant at the same time. This model was trained on the original ImageNet dataset with over 1 million training images. It works as a multi-level feature generator by computing 1 × 1, 3 × 3 and 5 × 5 convolutions. This allows the model to use all kinds of kernels on the image and to get results from all of those. All such outputs are stacked along the channel dimension and used as input to the next layer.

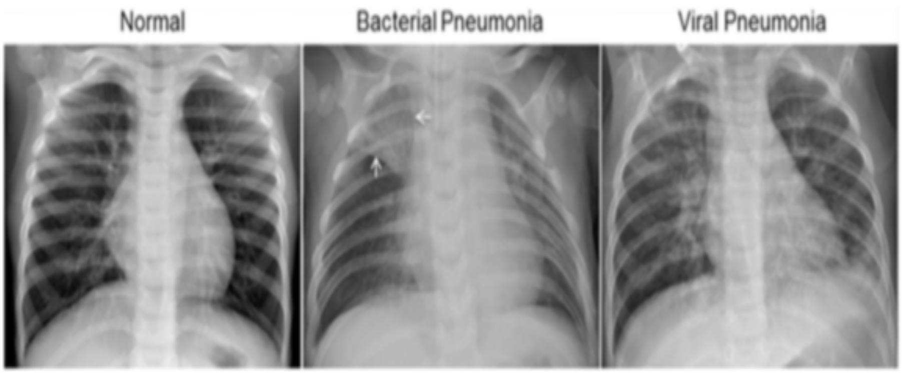
5. Discussion and Results:

5.1. Datasets

For this project, I choose the dataset which is available on Kaggle. The link for the dataset is https://www.kaggle.com/paultimothymooney/chest-xray-pneumonia.

The dataset is organized into 3 folders train, test and val. Each folder contains a subfolder for normal chest X-rays and chest X-rays infected with pneumonia. There is total 5,863 X-ray images (JPEG) each image is of resolution 2090 X 1858 the resulting image vector after converting to array will be of shape 1 X 3.8 million. The subfolders having pneumonia images has both the viral and bacterial pneumonia images.

Train data: Normal - 1341 images, Pneumonia - 3875 images
Test data: Normal - 234 images, Pneumonia - 390 images
Validation data: Normal – 8 images, Pneumonia - 8 images
In addition to that for pretrained models I will be using ImageNet dataset.



5.2. Evaluation Metrics

All the models were tested on the test dataset after the completion of the training phase. I validated the performance using the accuracy, recall, precision, F1 score. I used classification report from 'sklearn' metrics to get these values. In the below equations, while classifying normal and pneumonia patients, TP, TN, FP, and FN were used to denote the number of pneumonia images identified as pneumonia, the number of normal images identified as normal, the number normal images incorrectly identified as pneumonia images, and the number of pneumonia images incorrectly identified as normal, respectively. On the other hand, while classifying viral and bacterial pneumonia, TP, TN, FP, and FN were used to denote the number of viral pneumonia images identified as viral pneumonia, the number of bacterial pneumonia images identified as bacterial pneumonia, the number bacterial pneumonia images incorrectly identified as viral pneumonia images, and the number of viral pneumonia images incorrectly identified as bacterial pneumonia, respectively. Below given are the equations used:

$$\text{Accuracy} = (TP + TN) / (TP + FN) + (FP + TN)$$

$$\text{Precision} = TP / (TP + FP)$$

$$\text{Recall} = TP / (TP + FN)$$

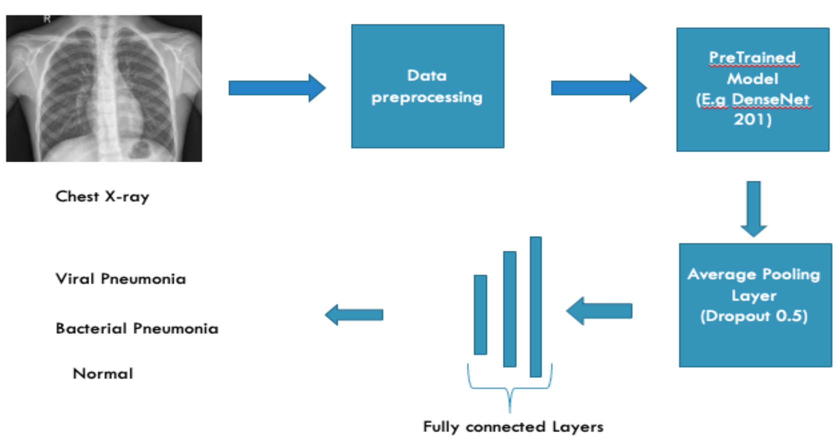
$$F1 = 2 \times (\text{Precision} \times \text{Recall}) / (\text{Precision} + \text{Recall})$$

For calculating the loss while the model goes through all the epochs I have used 'sparse_ categorical_crossentropy' while compiling the model. This loss function is used when there are two to more label classes and the labels are expected to be in the integer format.

5.3. Experimental results

To start with I implemented basic CNN model with three convolutional layers coupled with max pooling for auto extraction of feature from our images and down sampling the output convolution feature maps. I did this so that I can compare the improvement the transfer learning offers when compared with the basic CNN model which trains itself from scratch unlike transfer learning which uses pretrained models. For basic CNN model I got an accuracy of 66.92 % by using image augmentation.

The setup used for classifying the chest X-rays using pretrained model is as shown in the figure below. For e.g., we have pretrained dataset such as DenseNet201 on ImageNet dataset. And to the output of pretrained model I have added an 'Average Pooling Layer' with pool size of (4,4). After that I have added a dropout layer and then 3 fully connected layers (Dense layers) with 'relu' activation function with last layer classifying the images into viral, bacterial and normal. I experimented with adding dense layers and dropout layers in between them, also tried varying the size of dense layers. I tried adding just one dense layer as well. but the setup which I mentioned above with three fully connected layers worked better.



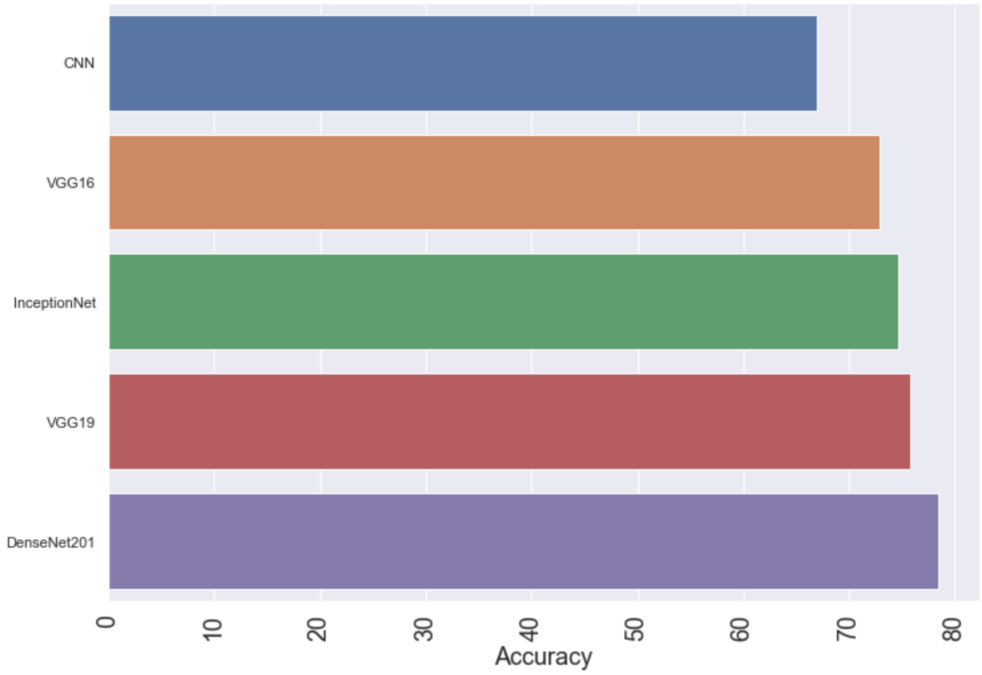
The pretrained model can be used in two ways:

1. Using a pre-trained model as feature extractor
2. Fine tuning the pretrained model as per the target domain.

5.3.1 Pretrained model as feature extractor

I used pre-trained model first as feature extractor and then eventually I would fine tune the model which performed best as feature extractor.

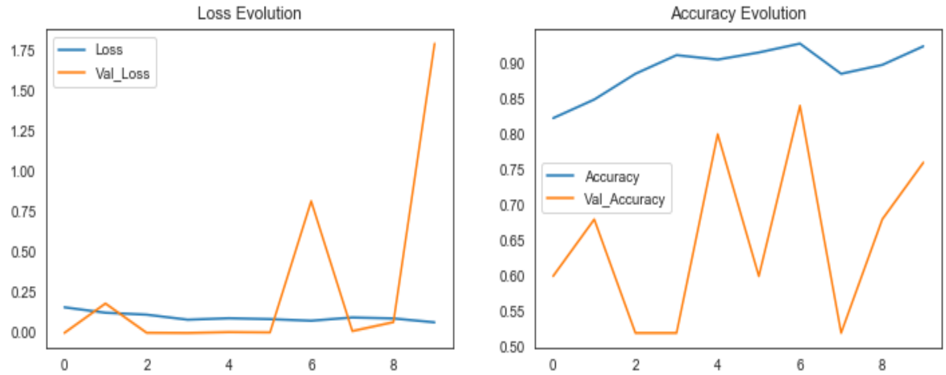
In Keras, freezing (by setting layer.trainable = False) prevents the weights in each layer from being updated during training. I trained this model for 10 epochs. I tried this with four CNN models VGG16, VGG19, InceptionNet and DenseNet201. The accuracy for each of these models is as given in figure below.



As we can see in the figure the VGG16 model has accuracy of 72.89%, the Inception Net has accuracy of 74.64%, the VGG 19 has accuracy of 75.80% and DenseNet201 is producing highest accuracy of 78.41 for test data. These results show improvement due to transfer learning when compared with basic CNN model trained from scratch which has accuracy of 66.92%.

5.3.2 Fine Tuning the pretrained model

I tried to further improve this accuracy by fine tuning the weights of the top layers of the pre-trained model alongside the training of the classifier. The training process will force the weights to be tuned from generic feature maps to features associated specifically with the chest X-ray dataset. Fine tuning should only be attempted after we have trained the top-level classifier with the pre-trained model set to non-trainable. In my first attempt of fine tuning, I attempted to train all layers jointly and my model got overfit on test data with accuracy of 45%. The reason for this was the magnitude of updates while running over epochs is very high and thus the pretrained model forgets what it has learned. In most convolutional networks, the higher up a layer is, the more specialized it is. The goal of fine-tuning is to adapt these specialized features to work with the new dataset, rather than overwrite the generic learning. Then I trained the top-level classifier with the pretrained model set to non-trainable. After that I unfreeze all the layers of base model by using ‘baseModel.trainable =True’ and compiled the model again. As I am training a much large model and want to readapt the pretrained weights it’s important to use a lower learning rate. By using this approach, I got a gradual improvement in accuracy, and I got accuracy of 82.89% when trained over 10 epochs. The graph for this process is given below. Thus, the best performing model was DenseNet201 when fine tuning all of its layers which gave an accuracy of 82.89%.



6. Conclusion

This project presents a CNN based transfer learning methods for detection and classification of pneumonia. Four popular CNN models were used as pretrained models and were tested for classifying normal and pneumonia images. Also, I did extensive experiments with freezing and fine tuning the layers of pretrained models. I observed that DenseNet201 when used as feature extractor by freezing all its layers gave best accuracy. I further fine-tuned the model according to the chest X-ray dataset and got an accuracy of 82.89%. These results conclude that deep learning model can be used to solve the problems related to diagnosis of diseases.

6.1. Direction for future work

To further improve the accuracy of the project, we can experiment on freezing some of the layers of pretrained models and fine tuning some of the layers and compare the results. Also, we can use different preprocessing and data augmentation techniques that involve brightness and contrast changes. Use of hyperparameters optimization techniques such as ‘GridSearch’, ‘Bayesian Optimization’, etc. to find the right set of hyper parameters.

7. References

[1] Rohit K., Ritacheta Das., Zong Woo Geem., Gi-Tae Han., Ram Sarkar Pneumonia detection in chest X-ray images using an ensemble of deep learning models.
[2] Sharma H., Jain J., Bansal P. & Gupta S. Feature extraction and classification of chest x-ray images using cnn to detect pneumonia.
[3] Yusuf B., Marcellin A., Stive T., Jaures E., Franklin T. Transfer Learning for the Detection and Diagnosis of Pneumonia induced by Covid-19 from Chest X-ray Images.
[4] Jaiswal, A.K.; Tiwari, P.; Kumar, S.; Gupta, D.; Khanna, A.; Rodrigues, J.J.P.C. Identifying pneumonia in chest X-rays.
[5] Juan E.,Corneli Y.,Yenny V.,Oscar C., A transfer learning method for Pneumonia Classification and Visualization.
[6] https://towardsdatascience.com/a-comprehensive-hands-on-guide-to-transfer-learning-with-real-world-applications-in-deep-learning-212bf3b2f27a