Lung Diseases Detection from X-Rays using Neural Networks

Emmanouil Lykos
University of Piraeus
NCSR Demokritos
manolislykos97@hotmail.gr

June 26, 2022

Abstract

In Deep Learning, given some task, we can either try to create our own models that excel on the desired task either use pretrained layers and then we can build upon them. However, the former approaches need more experimentation while the latter will give better results but they are more storage and computationally intensive. In this project I experimented with a custom network and different ResNet version on an X-Ray dataset that can be used in order to identify Viral Pneumonia, COVID-19, Lung Opacity or if its completely healthy. I evaluated these models using different time and predictive performance metrics. After the experiments, we saw that we developed a pretty lightweight model that is not much worse that the pretrained ones and also perform pretty decently to the given task. Finally, we cite our conclusions and provide directions for further work.

Keywords: Machine Learning, Classification, Deep Learning, Convolutional Neural Networks, Transfer Learning

1 Introduction

Nowadays, applications of Artificial Intelligence in the field of medicine have attracted much attention from researchers. Those applications vary from just simple algorithms that rely on biometric data that are extracted from simple sensors that the users wear like smartwatches or by just filling forms, to more complicated tasks like assisting in drug discovery or even the identification of mental illnesses using audio or visual samples from the candidate patients. However, one group of applications that attracted researchers is the identification of physical illnesses that can be extracted from photos, like the melanoma or pneumonia because if we developed models that can excel in those tasks we will make doctors' life easier, especially if those algorithm can reason about their decision. The most common techniques used in those kind of tasks fall in the field of Deep Learning and Computer Vision.

Deep Learning is a group of Machine Learning techniques that utilize artificial neural networks in order to excel in a particular task. Also, there was the common belief that these techniques are designed to simulate the functionality of a human brain because the human brain is also a network that consists of neurons. However, there are many more reasons that today Deep Learning has gained massive popularity and now are preferred more than traditional Machine Learning techniques. Initially, back in the 60s, when Deep Learning was introduced, most people did not had the required computational resources in order to run those models, but today-thanks to Moore's law- anyone can run, in the worst case, a small model in their computers. Furthermore, Deep Learning techniques, in most cases, do not require much preprocessing or feature extraction and also can handle different architectures of data like image or sequential data, but still they require much hyperparameter tuning. Finally, even if the training of Neural Networks can be computationally intensive, they usually yield the best performance than any other approaches.

More specifically, a Deep Learning architecture that can be used for Computer Vision are the Convolutional Neural Networks. The Convolutional layers are using the convolution image filtering technique which applies some filters(or kernels) in the image in order to extract some features from it, like the edges or to just modify the image like apply blurring. An example of how convolutions work is shown in Figure 1. But Convolutional Layers, do not use predefined kernels, instead they start with random ones and then find the best kernels that should be applied in order to reduce the training loss of the neural network. The main power of those networks is that they can extract by themselves specific features according to the different patterns of the image that can excel at the given task. This was achieved because the kernels that can be applied in an image are 2D or 3D ones, thus they utilize the structure of the image instead of other techniques that cannot, thus they ignore it.

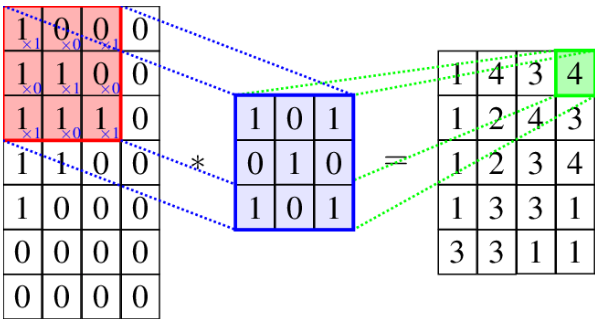


Figure 1: Example of application of image kernel in an image

Last but not least, as we will show on this report, the convolutional layers can be kept along with their pretrained weights and be applied to another problem, by just adding those layers and keep them still as the other model's layers train, just add them and then let them tuned according to the problem or use a combination where we freeze some layers and some other not. The last two ways of use might seem unnecessary and pointless but it is shown that using pretrained weights gives to the model a head start, thus, it can converge faster than training from completely random weights.

In this project, we will train a classifier that can identify if a patient has COVID, Lung Opacity, Pneumonia or is completely healthy from X-Ray images using different Convolutional Neural Networks. Initially, we will define a Handcrafted Neural Network that we saw that was the best due to the hyperparameter tuning that we did. Afterwards, we will present the pretrained ResNet model and how we applied it on our problem. Finally, we will compare the results in terms of accuracy and recall and discuss our results. Hence, the rest of the report will have the following structure. In Section 2, we will discuss what we could have done instead of deep learning and why its better to use a deep learning approach. In Section 3, we will show our handcrafted Convolutional Neural Network and also how pretrained ResNet model can be applied to our problem along with what it is. In Section 4, we will present and evaluate our experimental results. In Section 5, we will summarize our report and also provide directives for further work.

2 Non-Deep Learning Approaches

Because this problem is in essence a classification task, we could have used Traditional Machine Learning algorithms like Support Vector Machines or Random Forests along with Feature Extraction techniques from images. However, these approaches have some drawbacks. Initially, in order to get features that would work well for our problem, we should have done too much preprocessing on our image like applying various filters, object detection etc. Also, after the preprocessing we would need to find representative features that can capture as they can the basic information of the image, but we will still lose information about the structure of the image. Furthermore, those approaches, seem to have on image classification less accuracy than the Convolutional Neural Networks. On the other hand, by using Convolutional Neural Networks we do not need to waste time on Feature Engineering and Preprocessing and also we do not lose any information about the structure of the image. Furthermore, these networks do pretty good job at Feature Extraction because they extract features by taking into consideration local 2D or 3D parts of the image and also the trained layers might be transferred and can be used to solve another problem where in Machine Learning this is rarely the case. In summary, we can see that even Neural Networks are computationally intensive, they make our life easier in terms of running them because we need to do only minimal preprocessing and also yield better predictive performance.

3 Methodology

3.1 Preprocessing

Initially, the only necessary preprocessing step that we needed to do was to normalize the images' pixels, resize it to 128x128 and if needed, convert them to RGB form(3 channels) from greyscale(1 channel) that they originally are. This is done simple by repeating other 2 times the image. However, we tried to do the following transformations but they yield worse performance. Initially, because the dataset has the object recognition masks that denote where the lungs are in the picture along with the original images, we thought that it would be better to mask those images before passing them to the neural network, however this approach yielded worse by 6% accuracy score than passing the images as they are. Also, we tried image augmentation because it is said that stabilizes the training but we didn't got better accuracy or more stable training.

3.2 Handcrafted Convolutional Neural Network

Our first approach is to try to find a Convolutional Neural Network(CNN) that does not only have competitive predictive performance in comparison with the ResNet one, but also has significantly less parameters than the ResNet models. After trial and error we found that the best model is a one with 5 convolutional layers with output channels equal to $(32, 64, 128, 256, 512)$ and kernel sizes equal to $(5, 5, 3, 3, 3)$, followed by 2D maximum pooling layers of kernel size equal to 2 and Batch Normalization ones. Then, we added a linear layer that takes 512 as input and 4 neurons as output, one for each class. The activation function is at all layers the ReLu one, except for the last one where it has softmax because we want to give a probability to each class and all of them should be add to 1. Also, the optimizer used is the Adam one.

3.3 ResNet Approach

ResNet was a pretrained model using the ImageNet Dataset and achieved the highest performing accuracy on the corresponding challenge because it had an error rate of 3.57. The ImageNet challenge is a classification problem of 150000 photographs that should be classified to 1000 classes where another 50000 images will be used as validation set. Also, this task is a multi-label classification one because one image can take multiple labels. In general, ResNet works with residual connections, i.e. some output is not only directed to the next layer but bypasses it and gets right away to some layers ahead. This patent is used to fix the degradation problem where a deeper network has greater loss than the shallower one. An illustration of how residual connections work, is shown in Figure 2.

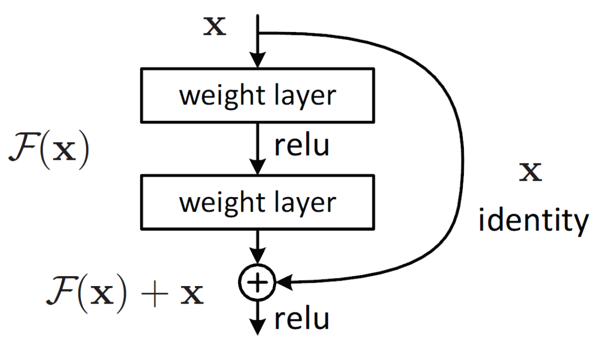


Figure 2: Example of a residual connection

In this project we dealt with two versions of ResNet, ResNet18 and ResNet34, mostly due to computational resource restrictions. From those models we will use only the feature extraction layers, i.e. the layers before the fully connected one and then we will add up a new fully connected that outputs 4 classes instead of 1000, because we have those in our problem. Those ResNet models contain 4 blocks that each one has some number of Convolutional layers along with some residuals. The architecture of ResNet18 is shown in Figure 3 and we use also the Adam optimizer and the Output layer has a Softmax activation function. The architecture of ResNet34 is virtually the same but with different residuals and more layers in each block.

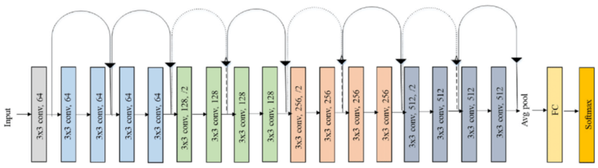


Figure 3: ResNet18

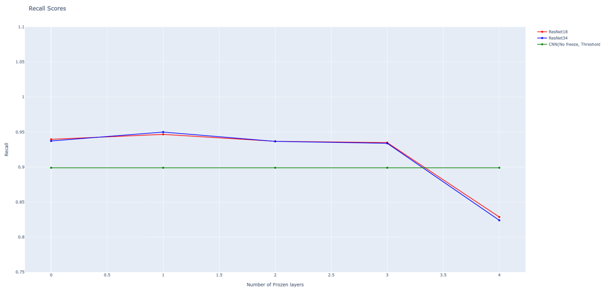
4 Experimental Evaluation

In order to evaluate our handcrafted Convolutional Neural Network and ResNet approach, we used the COVID-19 Radiography Database, which contains 20000 Lung X-Ray images with resolution 256x256 and each lung shown can be healthy, or have COVID-19, Viral Pneumonia or Lung Opacity. However, this dataset is imbalanced because 50% of the instances are showing healthy lungs which was mitigated by using weighted sampler which means that in each batch, its data will be balanced. In terms of train-test split I just took 10% of images from each class in order to do testing. Last but not least, this dataset contains image maps for each image that denotes where the lungs are but because masking yielded worse performance, we did not used them.

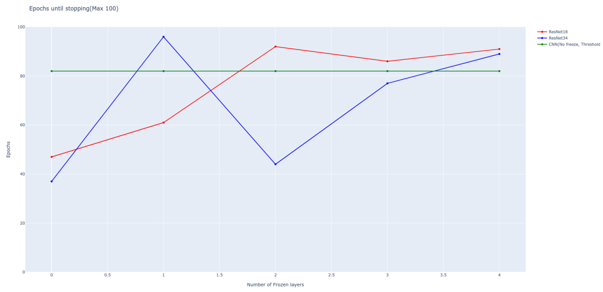
In terms of hyperparameter tuning I found that our models work best with patience equal to 20, batch size equal to 128 and learning rate equal to 0.0005. Also, given the current patience, 100 epochs seem more than enough for training. We found the best combination of the aforementioned hyperparameters in the background in order to evaluate the best model in terms of resource allocation, time and predictive efficiency with respect to how many ResNet blocks will froze during training and also how much better or worse they are in comparison with our handcrafted network. Our evaluating metrics in order to provide a decision are the following:

1. Testing Micro-Recall is the average recall of each class. Also, we saw in our experiments that it happens to be equal to accuracy score and micro-precision score. We believe that its better for our application to try to maximize recall because in lung diseases we want to apply the correct treatment as early as possible because, if not, those diseases can have various side-effects in the organism.
2. Training epochs to converge, using our early stopping criterion.
3. Training time per epoch.
4. Trainable parameters, which denotes the complexity of each model and also by watching the parameters when we not freeze the model its displayed if it needs more memory than the others because the greater the model is, the more memory it needs.

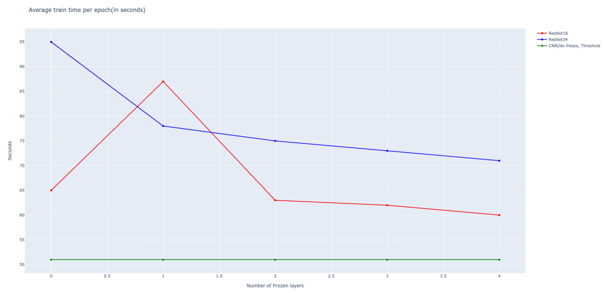
Note that, when we say that we will freeze X blocks, we mean the first X blocks and not any complex combinations. This is done because is more clear semantically because in ”earlier” blocks there are contained higher lever features than the subsequent ones. The results of each experiments can be shown in Figure 4. From the results, we can see that the best performance is yielded by the ResNet models when we freeze the first block wit performance of 94.5%, and if we consider the time efficiency and the memory needed the best between them is the ResNet18 version. This is because ResNet34 does not give any massive improvement in recall and we can see that it has the double amount of hyperparameters. In comparison to our handcrafted Convolutional Neural Network we can see that ResNet18 performs better by approximately 5% and has approximately 10-times more the hyperparameters, thus I believe that we developed a pretty good lightweight model in order to solve our



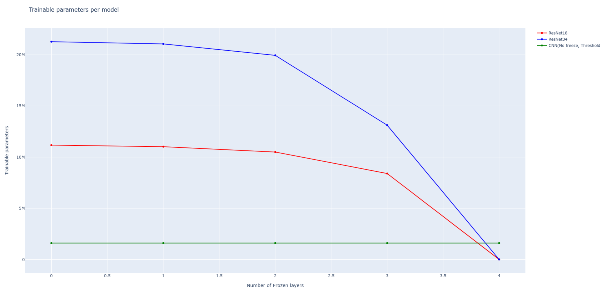
(a) Test Micro-Recall per different model with respect to how many layers we froze on ResNet ones. This happens to be equal to Micro-Precision.



(b) Epochs that each model needed to converge.



(c) Average Training time per epoch.



(d) Trainable parameters of each model. In the 0 value of x-axis you can see the total parameters of each model.

Figure 4: Evaluation metrics for proposed models in terms of how many ResNet blocks were froze during training.(Green Line=Handcrafted CNN, works as threshold, Red Line=ResNet18, Blue Line=ResNet34)

problem because 89.5% is still a high recall score. Finally, note that when we freeze all blocks ResNet performs worse than CNN which is expected because ResNet was trained on completely different task. However in those applications we probably need higher scores, especially if they will be run standalone without the consult of a doctor. Consequently, we can see that with minimal tuning we can use a pretrained neural network that excels in a given task but it come with the drawback that usually these models are more computationally intensive than the handcrafted ones because we need to still train its layers because the pretrained weights only work as a head start.

5 Conclusion & Further Work

In this work, we presented two Neural Network approaches that we can use for Lung Disease prediction from x-ray images. The former one is a Handcrafted Convolutional Neural Network which was found by ourselves with intention to be more lightweight than ResNet and have similar, or worse than a small margin, predictive performance. The latter were two versions of ResNet model using pretrained weights that were extracted by training it on ImageNet Challenge's Dataset. After evaluating them with respect to how many blocks we were freezing on ResNet models we found out that our custom model performs only 5% worse than ResNet model but it is also 10 times more lightweight and efficient on training because we still need to let some blocks to be trainable. However, if we only care about the predictive performance we will choose the ResNet18 model by freezing only the first block because it is the more lightweight ResNet version and ResNet34 did not showed any improvement in predictions. Last but not least, in terms of further work we can use the pretrained models or autoencoders in order to extract features from the images and then train a Machine Learning model using those representations.