Detecting Pneumonia in X-Ray Images using Deep Convolutional Neural Networks

Rajdeep Singh Lather
George Mason University
rlather@gmu.edu

Cody Kidwell
George Mason University
ckidwel@gmu.edu

Abstract - This paper will demonstrate how a model based on a deep convolutional neural network can be used to detect pneumonia in X-Ray images of lungs. We will show the results from various architectures, including in-house, Alexnet, and ResNet. For our best model, we will ensure to check for the best parameters to most efficiently train our model and demonstrate the use of transfer learning for the ResNet architecture. Finally, we will compare our results against a Kaggle competition.

I. Introduction

Pneumonia is an inflammatory condition of the lung affecting primarily the small air sacs known as alveoli. Symptoms typically include some combination of productive or dry cough, chest pain, fever, and difficulty breathing.

With the ongoing COVID-19 pandemic, we have seen an increase in the number of pneumonia cases and at a time when healthcare workers all over the world are extremely taxed. With this in mind, we are building a model that can look at X-Ray images of patients and identify pneumonia. To do this, we are training a Deep Convolutional Neural Network to learn to differentiate between X-Rays of healthy and pneumonia stricken lungs.

II. Related Work

Deep Convolutional Neural networks (CNNs) have made significant strides in the problem of domain classification in recent years[1]. As such, we have seen adapting these techniques to other fields that use images. One such field is that of medical imaging and diagnosis, where Deep CNNs have been used in applications ranging from looking at pictures of skin to detect cancers[2], to analyzing results of ultrasound imaging[3].

Due to the ongoing COVID-19 pandemic, there has been a sharp increase in interest for using deep learning to analyze X-Ray images, as we can see in a pre-published paper like “Classification of COVID-19 in chest X-ray images using DeTraC deep convolutional neural network”[4]. In fact, we have even seen papers on “Detecting Pneumonia using Convolutions and Dynamic Capsule Routing for Chest X-ray Images”[5], published this year.

The above papers use a variety of techniques in their approaches, what we wanted to test was how does a neural network trained on a traditional image classification dataset fare when it’s knowledge is transferred to a domain-specific dataset like medical imaging. Finally, we want to present a model that can accurately predict the presence of pneumonia in the lungs.

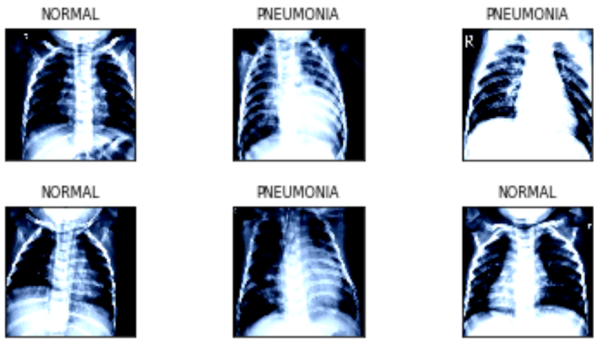


Figure. 1. Sample Data from the Kaggle dataset showing X-Rays of normal and pneumonia stricken lungs.

III. Dataset

We are using a Dataset of 5,863 labeled chest anterior-posterior X-Rays of infants between the ages of one to five. The images were originally sourced from a Mendeley Dataset[6], with the final version (used in this paper) available on Kaggle[7]. These images were obtained from retrospective cohorts of pediatric patients at Guangzhou Women and Children's Medical Center, Guangzhou.

The dataset was first screened for quality control by removing all low quality or unreadable images. Next, the labeling for the images was done by two expert physicians and a third expert checked the evaluation set to account for any grading errors.

The dataset contained 5,216 train images and 626 test images. Out of these 5,216 training images, 3875 X-Rays were of pneumonia stricken lungs and the remaining 1341 were of healthy lungs.

The images were of different sizes thus, requiring various preprocessing. Included in these preprocessing steps is resizing the images to be 227 pixels long in their shortest edge. Then we take a center crop and make the longest length 227 as well. Once we have flipped and resized the images, we then normalize the image to *mean*=[0.485, 0.456, 0.406], *std*= [0.229, 0.224, 0.225]. Then to generalize better on the training data we use random horizontal flips and randomized crops of the images from the training set as well.

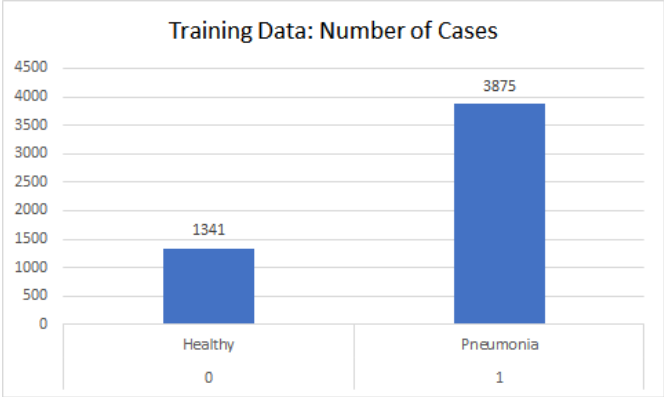


Figure. 2. The distribution of healthy to diseased lungs in the training set of the Kaggle dataset

IV. Approaches

In order to pick out the best approach for this problem, we tested out a few different ones during the validation phase. We then used the validation accuracy of these models to pick out the best model for this approach.

All the approaches that we tried utilize Deep Convolutional Neural Networks (CNN)[1]. These are a type of Deep Artificial Neural Networks that use convolution in place of general matrix multiplication in at least one of their layers. These convolution layers are used for feature extraction and the output from these layers is then fed into fully connected layers. This allows the feature extraction layers to be trained with the fully connected ones.

The first architecture we tested was a custom architecture that consisted of three convolutional layers with batch normalization between them. Followed by a max-pooling layer and then four fully connected layers. For the non-linearity, we used the Rectified Linear Unit (ReLU) .

The second approach we tried was AlexNet[8]. AlexNet is a CNN designed by Alex Krizhevsky it has five convolution layers and three fully connected

ones. It also uses ReLU as the activation function and uses dropout for regularization.

Finally, we tried two approaches with the ResNet[9] architecture. First, we tested ResNet with randomized weights. Next, we tested it with pre-trained weights to test the effect of transfer learning[11] on the model performance.

ResNet is a Residual Neural Network, these networks attempt to solve the gradient vanishing problem in deep neural networks. ResNet solves this by using residual connections also known as skip connections over some layers. ResNet50, the model that we used in this paper has double layer skips with ReLU and batch normalization in between. The mathematical formulation of ResNet is expressed in the following equations[1] -

$$\mathbf{F}_{m+1}^{k'} = g_c(\mathbf{F}_{l\to m}^{k}, \mathbf{k}_{l\to m}) + \mathbf{F}_{l}^{k} \qquad m \geq l$$

$$\mathbf{F}_{m+1}^{k} = g_a(\mathbf{F}_{m+1}^{k'})$$

$$g_c(\mathbf{F}_{l\to m}^{k}, \mathbf{k}_{l\to m}) = \mathbf{F}_{m+1}^{k'} - \mathbf{F}_{l}^{k}$$

Where, F_l^K is an input of the lth layer, $k_{l\to m}$ is the k^{th} processing unit (kernel), $l \to m$ is the residual connection and $g_c(F^k_{l\to m}, k_{l\to m})$ is a transformed signal.

The results of validation on different approaches can be seen in Table 1. As we can see, ResNet50 performed the best out of all the approaches. Thus, that is the one we picked for our paper. However, we saw an insignificant difference between ResNet and Pretrained ResNet. So, we decided to test and compare these to methods going forward.

Table 1. Validation accuracies of different Deep Learning techniques

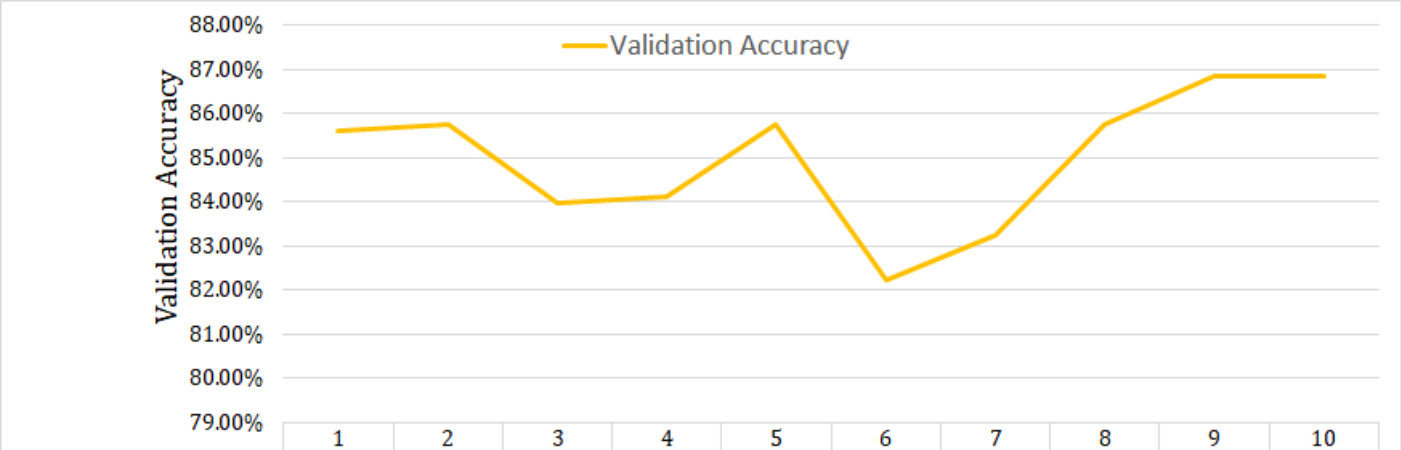
Architecture	Validation Accuracy
In House	65.15%
Alex Net	68.78%
ResNet50	**86.86%**
Pretrained ResNet50	**86.86%**

V. Training

When training our model we needed to decide the loss function and optimizer. We also included a scheduler to decrease our learning rate the further the training process went. Cross-Entropy Loss was the loss function that we resulted in. It gave us our best results, while also being efficient. While training, we decided to choose Adam[10], an adoptive learn rate optimization algorithm.

Once we had our type of loss function and optimizer chosen, tuning our hyperparameters was important. To find the best learning rate for our use case, we tested various values, .01, .001, .0005. All of which resulted in similar results, although the learning rate of .001 had minor improvement over the others. In addition, we used StepLR for our scheduler which required additional tuning. We tested a step size of 8, 10, and 16 epochs, a step size of 10 doing the best.

Finally, as we trained our Resnet50 model with these hyperparameters, we needed to determine which platform to train. We had an option between



	1	2	3	4	5	6	7	8	9	10
Learning rate	0.01	0.001	0.0005	0.001	0.001	0.001	0.001	0.001	0.001	0.001
StepLR for the Scheduler	10	10	10	8	10	16	10	10	10	10
Number of Epochs	30	30	30	30	30	30	20	30	40	60
Validation Accuracy	85.61%	85.76%	83.96%	84.11%	85.76%	82.24%	83.23%	85.76%	86.86%	86.86%

Figure. 3. Hyperparameter Optimization for our ResNet50 model

Google Colab and our local machine with a 1080ti GPU. We decided to choose the local machine since it has outperformed in similar tasks. Testing various epoch ranges, 20, 30, 40, 60, the local machine completed the best result for the performance, 40 epochs in around 3.5 hours.

Table 2. Validation Accuracies for different values of Learning Rate

Learning rate	Validation Accuracy
0.01	85.6134 %
0.001	**85.7612 %**
0.0005	83.9612 %

Table 3. Validation Accuracies for different values of StepLR for the Scheduler

StepLR for the Scheduler	Validation Accuracy
8	84.1141 %
10	**85.7612 %**
16	82.2435 %

Table 4. Validation Accuracies for different values of number of epochs

Number of Epochs	Validation Accuracy
20	83.2342 %
30	85.7612 %
40	**86.8590 %**
60	86.8590 %

VI. Metrics and Performance Evaluation

Measuring the results of our model was important. The first metric we tested for was accuracy, taking the number of correct results divided by the total number of guesses. This was important, but Recall and Precision were also added for more information on our results. Recall, one of the most important would measure the false negatives. This would be important because you do not want to incorrectly diagnose a positive case of pneumonia. We also calculated Precision, measuring the false positives. Listed are results from both the Resnet50 and pre-trained Resnet50.

Table 5. Test results of ResNet trained on pneumonia X-Ray images

Accuracy	Recall	Precision
86.8589%	71.5867%	82.9059%

Table 6. Test results of using a pre-trained ResNet model and then training it on pneumonia X-Ray images

Accuracy	Recall	Precision
86.8637%	71.7712%	82.9765%

Finally, we compared our results to a Kaggle competition[12] with 75 participants which used the same dataset. Listed in Table 7 are the ranks and accuracies of the different models in that competition. We have also inserted our model within the table at the rank we would have gotten with our accuracy.

Table 7. Comparing model performance against a Kaggle Competition[12]

Rank	Team Name	Accuracy
1	610182	0.9423
2	610139	0.9311
3	610117	0.9311
4	610018	0.9295
5	610107	0.9295
6	610841	0.9151
7	610838	0.9135
8	610110	0.9119
9	610157	0.9054
10	610709	0.9038
11	610134	0.899
...	...	...
30	610151	0.8734
31	510757	0.8734
32	610146	0.8718
33	610138	0.8718
34	610126	0.8702
35	610187	0.8702
36*	**Rajdeep and Cody**	0.8687
36	610103	0.8654
37	610767	0.8654
38	610105	0.8638
39	610152	0.8638
40	610133	0.8622
41	610027	0.8622
...	...	...
70	710117	0.8029
71	612208	0.8013
72	610855	0.7997
73	610164	0.7869
74	610185	0.7837
75	611097	0.7772

VII. Conclusion

Recent advances in using deep learning for image classification has opened up new avenues in clinical applications. As these methods continue to improve in performance, we will soon see them approach human-level capabilities, perhaps even surpass them. However, there is room for improvement in such applications, especially if peoples' lives are concerned.

Our model's accuracy of 86.86% is on par with the other competitors training this data on Kaggle. Out of 75 participants, we would have been ranked 36, coming in the middle of the pack. We can see that ResNet was relatively successful in classifying pneumonia data.

We also demonstrated that using an existing model, designed for and trained on traditional image classification tasks can be modified to be used for the classification of X-Ray images. Since the pre-trained ResNet's final accuracy was functionally the same as training it from scratch.

VIII. Future Work

We believe that an application working on diagnosis must have as few false negatives as possible. Hence, the first aspect we would like to improve upon is the recall.

To this end, we would also like to try our hand at modifying the ResNet architecture to improve performance. Finally, we would like to train this network on a larger and more diverse dataset to be able to generalize better.

IX. References

[1] Khan, Asifullah & Sohail, Anabia & Zahoora, Umme & Saeed, Aqsa. (2019). "A Survey of the Recent Architectures of Deep Convolutional Neural Networks". Artificial Intelligence Review. 10.1007/s10462-020-09825-6.

[2] Songtao Guo, Zhouwang Yang. (2018). "Multi-Channel-ResNet: An integration framework towards skin lesion analysis". Informatics in Medicine Unlocked, Volume 12. ISSN 2352-9148.

[3] R. J. G. van Sloun, R. Cohen and Y. C. Eldar. (2020). "Deep Learning in Ultrasound Imaging", in Proceedings of the IEEE, vol. 108, no. 1, pp. 11-29. doi: 10.1109/JPROC.2019.2932116.

[4] Abbas, Asmaa and Abdelsamea, Mohammed and Gaber, Mohamed. (2020). "Classification of COVID-19 in chest X-ray images using DeTraC deep convolutional neural network" Cold Spring Harbor Laboratory Press.

[5] Mittal A, Kumar D, Mittal M, Saba T, Abunadi I, Rehman A, Roy S. (2020). "Detecting Pneumonia using Convolutions and Dynamic Capsule Routing for Chest X-ray Images". Sensors (Basel);20(4):1068. doi: 10.3390/s20041068. PMID: 32075339; PMCID: PMC7070644.

[6] Kermany, Daniel; Zhang, Kang; Goldbaum, Michael. (2018). "Labeled Optical Coherence Tomography (OCT) and Chest X-Ray Images for Classification", Mendeley Data, v2. http://dx.doi.org/10.17632/rscbjbr9sj.2

[7] Paul Mooney. (2018). "Chest X-Ray Images (Pneumonia)", Kaggle. https://www.kaggle.com/paultimothymooney/chest-xray-pneumonia

[8] Krizhevsky, Alex & Sutskever, Ilya & Hinton, Geoffrey. (2012). ImageNet Classification with Deep Convolutional Neural Networks. Neural Information Processing Systems. 25. 10.1145/3065386.

[9] He, Kaiming & Zhang, Xiangyu & Ren, Shaoqing & Sun, Jian. (2015). Deep Residual Learning for Image Recognition.

[10] Kingma, Diederik & Ba, Jimmy. (2014). Adam: A Method for Stochastic Optimization. International Conference on Learning Representations.

[11] Yosinski, Jason & Clune, Jeff & Bengio, Y. & Lipson, Hod. (2014). How transferable are features in deep neural networks?. Advances in Neural Information Processing Systems (NIPS).

[12] Pneumonia Classification. NCTU-ML-2020-Lab4. https://www.kaggle.com/c/nctu-ml-2020-lab4/leaderboard