CSCI 1430 Final Project Report: Classifying Chest X-Ray Images

X-Ray Vision: Hossam Zaki, Mohamad Abouelafia, Isaac Nathoo, Muhammad Haider Asif
Brown University
8th May 2020

Abstract

The number of chest x-rays performed each year has been increasing and each requires an expert radiologist to manually analyze and provide a diagnosis. In order to assist these physicians and tackle the problem of doctor error, we designed a CNN to classify common pulmonary diseases as well as COVID-19. In this project, we also compare our model to DenseNet121 and MobileNet for the task of chest x-ray image classification. We find that although our model has slighly higher binary accuracy, both MobileNet and DenseNet121 have greater AUC-ROC values. Our project integrated elements from a kaggle notebook that we found here. Code and dataset are available at this Githuhb Repo here.

1. Introduction and Motivation

Chest X-Rays are one of the most commonly performed radiological examinations for screening and diagnosis of many lung diseases. According to the WHO, each year, about 4 million people die from lower respiratory infections and pneumonia [5]. Within healthcare systems, radiologists manually classify several hundreds of x-rays each day without any automated assistance. Furthermore, the retrospective error rate among radiologic examinations is approximately 30%, with real-time errors in daily radiology practice averaging 3–5%. A study from the Mayo Clinic found that when seeking a second opinion, the original diagnosis was only confirmed 12% of the time. Conversely, as many as 88% of patients are given a new or refined diagnosis [9]. Diagnostic errors are estimated to account for 40,000–80,000 deaths annually in U.S. hospitals alone. [4]

In addition, during the COVID-19 pandemic, hospitals in the US and around the world have been overwhelmed by the number of patients seeking admission for respiratory illnesses. COVID-19 predominantly targets the lungs and is a highly infectious disease with an estimated mortality rate of 1.3% [2]. The early and automatic diagnosis of COVID-19 may be beneficial for timely recommendation of the patient to quarantine, rapid intubation of serious cases in hospitals, and monitoring of the spread of the disease.

2. Problem Statement

Our goal was to build a convolutional neural network model to assist radiologists with correctly classifying different common pulmonary conditions, such as pneumonia, atelectasis, hernia, edema. This would help to reduce the immense work burden on doctors, especially because the number of x-rays performed is constantly increasing, and allow for patients to get more accurate diagnosis. Furthermore, we wanted to use COVID-19 chest x-rays to see if our model could accurately classify COVID-19 x-rays. This would allow for greater efficiency in diagnosing COVID-19 cases, which would enable hospitals to provide better patient care and help more people.

3. Related Work

Deep learning has been applied to many fields within medicine. Some noteworthy achievements in this field have been the use of CNNs to achieve dermatologist-level classification of skin cancer [3] and radiologist-level pneumonia detection on chest x-rays [7]. As more complex neural network models are being developed, the accuracy of these deep learning classification systems have been increasing. Furthermore, the medical research community has come together to release large medical information datasets for purposes such as deep learning.

In 2017, the NIH released a chest x-ray image dataset with 112,120 frontal-view x-ray images of 30,805 unique patients with fourteen text-mined disease image labels [10]. Recently, a Kaggle dataset containing about 190 COVID-19 x-ray images was released. In this project, we combined these two image datasets and built our own CNN model to classify these images. We also fed our data through the existing DenseNet121 and MobileNet models, and then compared the results with the model we implemented.

4. Methods

Before we could use deep learning to classify our image data, we had to perform extensive preprocessing in order to rescale the images, convert them to gray-scale, generate the image data, and create dictionaries for the labels. The images in the NIH chest x-ray dataset were 1024x1024 pixels; however, the COVID-19 images were variable in their sizes. We resized all of our images to a standard 256x256 pixels, which also helped to reduce the computational time due to the size of our dataset. Furthermore, we chose to use gray-scale rather than color images for our model because was previously shown that models pre-trained on grayscale ImageNet outperformed color ImageNet models for disease classification based on chest x-ray images [11]. Reducing our images to only 1 channel as compared to 3 channels also improved our training speed. Lastly, for image data augmentation within our preprocessing pipeline, we chose to randomly flip our images horizontally as well as add a slight height shift, width shift, rotation, sheer and zoom.

The architecture of the CNN model that we built is shown in Figure 1. We chose this design based on a combination of three previous CNN architectures that had been shown to be relatively successful in classifying x-ray images [8] [6] [1]. Our model used the Adam optimizer and binary cross entropy loss, and we tracked the performance using the binary accuracy and mean absolute error metrics. In regards to hyperparameters, our model used a batch size of 32 with a learning rate of 0.001 and ran for 7 epochs. Once learning stagnates, the learning rate is reduced by a factor of 3 in an attempt to improve model performance. Regularization was also implemented with a 50% dropout for each epoch to prevent overfitting. The final layer of our model was an multilayer perceptron with 16 units and a sigmoid activation function to assign probabilities to each of the 16 disease classes for a given image.

In addition to this model, we also included modified versions of the DenseNet121 and MobileNet, whose architectures are shown in Figure 2. These models contained few parameters (all trainable) than our model; however, their designs are much more complex. Our model was simpler in order to achieve computational efficiency.

To run these different CNN models, we split our data up into 75% for training and validation and 25% for testing. Based on our initial results, we found a strong imbalance within our dataset and reduced accuracy due to multiclass images. To resolve this, we excluded images with the label "No Findings," which accounted for more than half of our dataset. We also removed x-ray images which were under a duplicate Patient ID and image label to reduce any skew that may have been present. Additionally, because some images were under several labels, we excluded these from our dataset; however, in the future, we hope to adapt our model for multiclass predictions.

Layer (type)	Output Shape	Param #
block1_conv1 (Conv2D)	(None, 256, 256, 32)	320
block1_conv2 (Conv2D)	(None, 256, 256, 32)	9248
block1_pool (MaxPooling2D)	(None, 64, 64, 32)	0
block2_conv1 (Conv2D)	(None, 64, 64, 64)	18496
block2_conv2 (Conv2D)	(None, 64, 64, 64)	36928
block2_pool (MaxPooling2D)	(None, 32, 32, 64)	0
block3_conv1 (Conv2D)	(None, 32, 32, 128)	204928
block3_conv2 (Conv2D)	(None, 32, 32, 128)	409728
block3_conv3 (Conv2D)	(None, 32, 32, 128)	409728
block3_pool (MaxPooling2D)	(None, 16, 16, 128)	0
block4_conv1 (Conv2D)	(None, 16, 16, 256)	819456
block4_conv2 (Conv2D)	(None, 16, 16, 256)	1638656
block4_conv3 (Conv2D)	(None, 16, 16, 256)	1638656
dropout (Dropout)	(None, 16, 16, 256)	0
flatten (Flatten)	(None, 65536)	0
dense (Dense)	(None, 512)	33554944
dropout_1 (Dropout)	(None, 512)	0
dense_1 (Dense)	(None, 16)	8208

Total params: 38,749,296

Figure 1. Our CNN Architecture

Layer (type)	Output Shape	Param #
mobilenet_1.00_256 (Model)	(None, 1000)	4253288
dropout (Dropout)	(None, 1000)	0
flatten (Flatten)	(None, 1000)	0
dense (Dense)	(None, 512)	512512
dropout_1 (Dropout)	(None, 512)	0
dense_1 (Dense)	(None, 15)	7695

Total params: 4,773,495

Layer (type)	Output Shape	Param #
densenet121 (Model)	(None, 1000)	8056232
dropout (Dropout)	(None, 1000)	0
flatten (Flatten)	(None, 1000)	0
dense (Dense)	(None, 512)	512512
dropout_1 (Dropout)	(None, 512)	0
dense_1 (Dense)	(None, 15)	7695

Total params: 8,576,439

Figure 2. Existing model architectures used for comparison - Top: DenseNet. Bottom: MobileNet.

5. Results

We ran three different models on our pre-processed data set of X-ray images and calculated the val_loss, val_binary_accuracy and val_mean_abs_error. The reason we specifically chose DenseNet121 was because it has previously been shown to be good at medical image classification

tasks, so using it would allow us to compare how our model did compared to models already being used for such purposes. MobileNet, is a very efficient model, and hence was used for both its computational complexity and time efficiency. This made it a good choice for us to compare and improve or own model.

Model	val_binary_acc	val_mean_abs_err
Our Own Model	0.9579	0.0730
Without "No Finding"	0.9333	0.0696
MobileNet	0.9333	0.1136
DenseNet121	0.9333	0.1121

Table 1. Results for Our Own Model, MobileNet and DenseNet121, all with different architectures (Including "No Finding" in the data set.)

The results have been tabulated in Table 1 above. As it shows, our model had the highest binary accuracy on the validation data set, and had the lowest mean absolute error on the validation data set. However, this was the case only when we included "No Finding" in the data set, which raises questions regarding how to design the optimal data set. Moreover, it also confirmed our preset assumption that Dense Net 121 had the potential to do better than Mobile Net because it also had a smaller mean absolute error on the validation data set when compared to Mobile Net. Another metric that we used to compare all the models were the ROC curves, Figure 3, 4 and 5 show the three ROC curves for our own model, DenseNet121, and MobileNet.

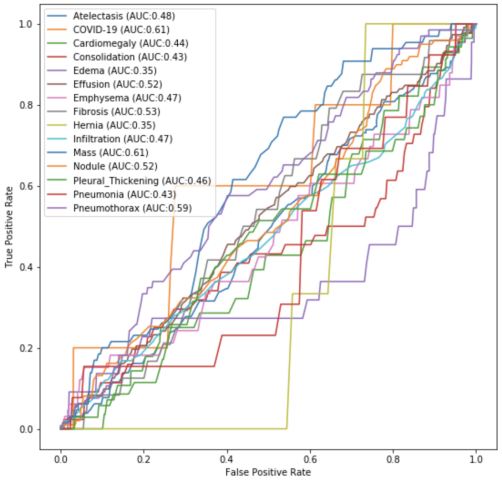

Figure 3. ROC curve for our own model.

The ROC curves, plots of False positive rates to True positive rates for each model's predictions, for all the three different kinds of models show us how well the models do on our NIH and COVID-19 data sets beyond just accuracy

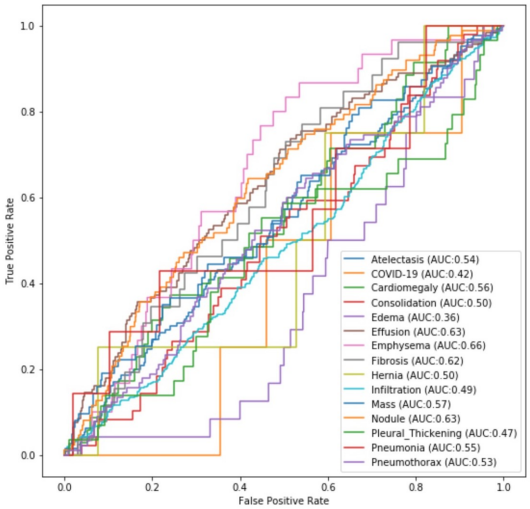

Figure 4. ROC curve for Dense Net.

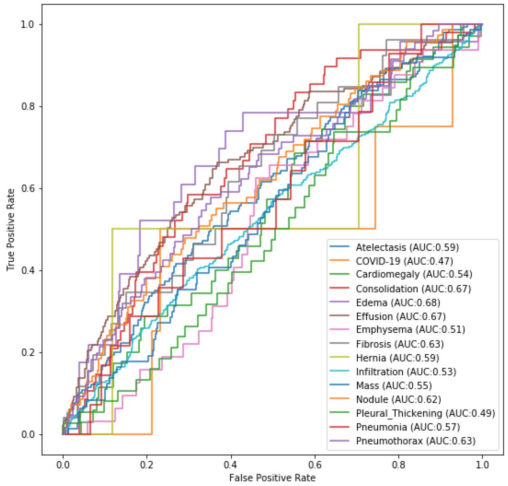

Figure 5. ROC curve for Mobile Net.

numbers. As the false positive rate (the rate of incorrectly diagnosing a patient with a given disease), the true positive rate should climb and at a high gradient. This would tell us that the model is performing well as it is classifying a large number of cases correctly compared to the incorrect classifications.The AUC (area under the ROC curve) represents the degree of separability in classes, are shown next to the class labels in the graphs and higher the AUC value, the better the model distinguished between a patient who has the disease and a patient who doesn't. We can see in these figures that the AUC values for MobileNet were the highest and the AUC values of DenseNet121 and our model are lower yet closer to each other. In general we do have higher than standard, 0.5 AUC value which shows us that our model has the capability to determine which patient has the disease or not accurately and effectively, and especially

it performs at a net positive rate on most, and way more than that on some of the diseases.

5.1. Discussion

Some of the decisions we made while pre-processing our data raise some interesting questions. Since X-rays of infected patients vary so little, the decision that we made to resize the images from 1024 x 1024 to 256 x 256, just to accommodate the COVID-19 data set, was that enough of a reason to lose out on the possibly higher accuracy and AUC that we could have achieved if all the images were not resized and COVID-19 was not taken into account, since when we resize the image we lose detail from the entirety of the image. Also, we cannot forget some issues with our data set. With the set of images we got from the NIH, the label "No Finding" was found to be more prevalent than any other label. This would skew the results, and thus make our model more biased to a specific label. Therefore, we decided to test how the removal of "No Finding" would change our results. This was definitely not the most optimal method of training our model, but given the time and computational constraints, it was the best we could do. Therefore, in the future, we would like to work on improving our data set, to make sure that every label is accounted for and and is balanced.

6. Conclusion

Analyzing all the results together, combined with the fact that our model to classify X-ray images of 15 different disease types including COVID-19, (hence is able to work on the latest challenges faced by the medical community), and is computationally faster and more accurate than DenseNet and even MobileNet, even when running with very small number of epochs due to time constraints and on only CPU's because we had issues with integrating Google Cloud Platform (GCP) GPU's with tensor flow, we feel confident that it can be a very high end choice for radiologists and medical institutions, who would need efficient yet accurate results in a time frame that would allow them to operate and pass out results effectively while not compromising on their quality standards. However, we must continue refining our model and our data set so that we can improve the results we get. Perhaps building a more balanced data set, or filtering out any images that are hard to interpret would be useful. As long as there are medical issues around the world, we should be using the power of computer science to help ease them.

References

[1] A. Abbas, M. M. Abdelsamea, and M. M. Gaber. Classification of covid-19 in chest x-ray images using detrac deep convolutional neural network. *arXiv preprint arXiv:2003.13815*, 2020. 2

[2] A. Basu. Estimating the infection fatality rate among symptomatic covid-19 cases in the united states. *Health Affairs*, 0(0):10.1377/hlthaff.2020.00455, 2020. 1

[3] A. Esteva, B. Kuprel, R. A. Novoa, J. Ko, S. M. Swetter, H. M. Blau, and S. Thrun. Dermatologist-level classification of skin cancer with deep neural networks. *Nature*, 542(7639):115–118, 2017. 1

[4] C. S. Lee, P. G. Nagy, S. J. Weaver, and D. E. Newman-Toker. Cognitive and system factors contributing to diagnostic errors in radiology. *American Journal of Roentgenology*, 201(3):611–617, 2013. 1

[5] F. of International Respiratory Societies. The global impact of respiratory disease – second edition. *European Respiratory Society*, 2017. 1

[6] F. Pasa, V. Golkov, F. Pfeiffer, D. Cremers, and D. Pfeiffer. Efficient deep network architectures for fast chest x-ray tuberculosis screening and visualization. *Scientific reports*, 9(1):1–9, 2019. 2

[7] P. Rajpurkar, J. Irvin, K. Zhu, B. Yang, H. Mehta, T. Duan, D. Ding, A. Bagul, C. Langlotz, K. Shpanskaya, et al. Chexnet: Radiologist-level pneumonia detection on chest x-rays with deep learning. *arXiv preprint arXiv:1711.05225*, 2017. 1

[8] O. Stephen, M. Sain, U. J. Maduh, and D.-U. Jeong. An efficient deep learning approach to pneumonia classification in healthcare. *Journal of healthcare engineering*, 2019, 2019. 2

[9] M. Van Such, R. Lohr, T. Beckman, and J. M. Naessens. Extent of diagnostic agreement among medical referrals. *Journal of evaluation in clinical practice*, 23(4):870–874, 2017. 1

[10] X. Wang, Y. Peng, L. Lu, Z. Lu, M. Bagheri, and R. M. Summers. Chestx-ray8: Hospital-scale chest x-ray database and benchmarks on weakly-supervised classification and localization of common thorax diseases. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2097–2106, 2017. 1

[11] Y. Xie and D. Richmond. Pre-training on grayscale imagenet improves medical image classification. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 0–0, 2018. 2

Appendix

Team contributions

Hossam Zaki Worked on research, building the model/preprocessing, and training and testing the model

Isaac Nathoo Worked on research, building the model/experimented with preprocessing, and writing the report

Muhammad Haider Asif Worked on research, building the model/experimented with preprocessing, and writing the report

Mohammad Abouelafia Worked on research, building the model/preprocessing, and training and testing the model