Surgical Finetuning Improves Pneumonia Prediction with Distribution Shift

Angela Zhao
Computer Science Department
Stanford University
Stanford, United States
angezhao@stanford.edu

I. Abstract

Pneumonia kills more children than any other infection disease. While it can be diagnosed through x-rays, this is still difficult for trained physicians. Additionally, adult xrays with pneumonia is more publicly available compared to children's pneumonia x-rays, although children are more at risk from pneumonia. This project aims to train a classifier on adult x-rays to predict pneumonia in children x-rays. After pretraining the model using transfer learning on ResNet18, the project will explore the use of surgical finetuning in helping models perform better due to the distribution shift.

II. Introduction

Artificial intelligence is capable of solving difficult medical issues, such as the impact of childhood pneumonia. Since more children die from pneumonia than any other infections disease [1], it is crucial to accurately identify and treat pneumonia.

The first issue is that is not much publicly available child x-rays of pneumonia, but there are large datasets containing adult x-rays of pneumonia. Additionally, CNNs trained on data from one hospital or a group of hospitals generalize poorly to data from different hospital or hospital groups [2]. I hope to combat these issues on two fronts. The first is training on adult x-rays to generalize to children x-rays so that children can be more quickly identified for pneumonia. The second is attempting to improve the generalization between x-rays taken from different hospital groups.

I aim to build a classifier to determine whether an x-ray image contains signs of pneumonia or not. The goal is to build a classifier that classifies images with greater accuracy than a doctor and can handle classifying images taken on different machines from different hospitals, which is a common problem when classifying x-ray images. The classifier will be first trained on a set of x-ray images from the NIH and generalized to children's x-rays.

This project is shared across CS 230 and CS 229. For CS 229, the goal is to create a classifier that performs well through finetuning only on the children's x-rays. Here, the goal is to explore distribution shift by first training on adult x-rays and then testing on children's x-rays. CS 229 and CS 230 will use the same pretrained model, resnet18, and the same children's x-ray datasets. CS 229 and CS 230 both share the use of confusion matrices and saliency maps.

III. Related Work

Prior work on identifying pneumonia in children and adult x-rays have mostly used convolutional neural networks (CNNs). Kermany et al 2018 demonstrated the success of transfer learning combined with deep convolutional neural networks to classify pediatric x-rays as viral pneumonia, bacterial pneumonia, or normal [3]. Kermany et al was able to achieve an accuracy of 92.8%, a sensitivity of 93.2%, and a specificity of 90.1% [3]. Additional work also proves the usefulness of pretrained models. Ibrahim et al 2021 used a pretrained model, AlexNet, to classify different kinds of pneumonia – COVID-19 pneumonia, non-COVID-19 pneumonia, bacterial pneumonia, and viral pneumonia [4]. AlexNet uses the ReLU activation function in place of the sigmoid activation function, and is composed of five convolutional blocks, followed by two fully connected layers and one output layer. The model achieved 94.43% testing accuracy, 98.19% sensitivity, and 95.78% specificity when predicting between normal and non-COVID-19 viral pneumonia [4]. Additionally, prior work has shown that a combination of data augmentation and dropout boosts performance [5].

Additionally, prior work on distribution shift has presented a variety of approaches. One common approach is to first train on the source data and then generalize to the target data [6]. However, since the goal of transfer learning is to preserve useful aspects of the model while learning new features, additional methods have also been introduced to prevent overfitting. Some common methods include using different learning rates per layer [7].

Surgical finetuning is a method that selectively freezes different blocks in the model for better performance based on different distribution shifts, instead of the classic method of either finetuning the entire model or freezing all but the last few layers of the model [8], [9]. In the case of a distribution shift of subgroups where the training data was on one subgroup and the testing data is on another subgroup, such as training on cucumbers and artichokes to identify squash and cauliflower, surgical finetuning argues that freezing the middle layers of the model produces better performance [8].

Lee et. al 2022 suggested two different applicable methods. The first was to freeze all model layers except the last layer for data with spurious correlations. The second was to freeze a model layer in the middle-to-end for problems handling a subgroup shift.

I will be exploring both methods in this project, in conjunction with data augmentation and dropout.

IV. Dataset and Features

The pretraining dataset is from the RSNA 2018 pneumonia challenge. The data is found here [10]. This is a collection of 30,000 adult x-rays, either with pneumonia or normal, that was sourced from the public National Institutes of Health (NIH) CXR8 dataset. The images were labeled by 18 physicians from 16 different institutions. These images were randomly selected from the NIH repository of common thorax illnesses, and span a range of age and gender. Since the data was for a pneumonia challenge, only the training set was labeled, and therefore only the training set was used. 10% of the training set was randomly augmented by flipping the images horizontally and flipping the images vertically. The training set contains 29,352 images, 10,510 x-rays of pneumonia and 18,842 normal images.

The training, validation and testing set were obtained from the Guangzhou Women and Children's Medical Center [3]. This dataset contains 5848 images of viral pneumonia, bacterial pneumonia, and healthy lungs in children 1-5 yrs old. These images were obtained from Guangzhou Women and Children's Medical Center and were classified by two physicians [11]. Together, they consist of 1575 x-rays of normal lungs and 4273 x-rays of pneumonia. The validation set consisted of 2924 images, 2143 of pneumonia and 781 of normal. The test set contained 2924 images, 2130 of pneumonia and 794 of normal. The training set to finetune the models were 100 randomly selected images from the validation set.

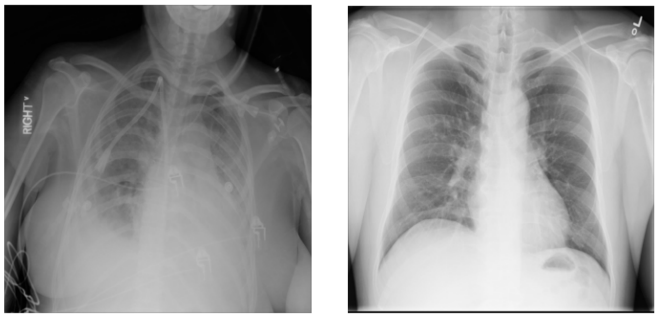

Fig. 1. RSNA: pneumonia chest x-ray (left) and normal chest x-ray (right)

V. Methods

The learning algorithms consisted of using data augmentation in conjunction with surgical finetuning. The pretrained model used is ResNet 18 using the latest weights after ResNet18 was trained to identify 1000 categories using ImageNet. This model consists of 18 individual layers and 4 blocks, which are referred to as block 1, block 2, block 3, and block 4. I chose to unfreeze block 2, block 3 and block 4 individually since surgical finetuning suggested that unfreezing

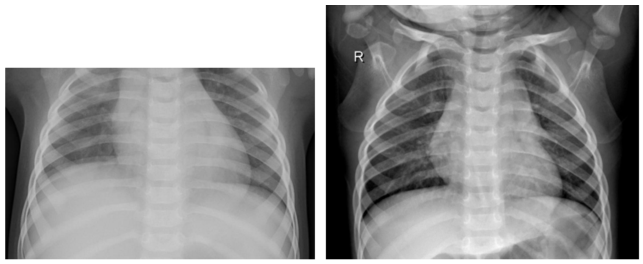

Fig. 2. Guangzhou: pneumonia chest x-ray (left) and normal chest x-ray (right)

the middle or later blocks in the model work the best for subgroup shift.

Since the classes are deeply unbalanced, I used weights to rebalance the cross-entropy loss. The most common class had a weight of 1, while the less common class was a value such that when the proportion of the less common class was multiplied with the value the less common class had the same proportion of data as the common class. For example, since the validation data is 26.7% normal x-rays and 73.2% pneumonia x-rays, the weight for the normal class loss is 73.2%/26.7% = 2.7439. The weight of the pneumonia class is 1. Similarly, for the training data, the weight for normal loss is 1 and the weight for pneumonia images is 2.1635. The testing set did not have any weights applied to the loss.

I explored these models: finetuned, finetuned with dropout, surgical finetuning with only unfreezing block 2, surgical finetuning with only unfreezing block 3, and surgical finetuning with only unfreezing block 4.

A. Finetuned with Dropout

I used the pretrained model in conjunction with a dropout layer right before the fully connected layer. The dropout layer uses a dropout probability of 0.5. The model was then trained on the same 100 images from the validation dataset to finetune the model.

B. Surgical Finetuning

For surgical finetuning, I unfroze only blocks 2, 3, and 4 individually so that there were three different models.

Block 2 refers to the second block out of the four blocks that consist of the ResNet 18 model. This block is called conv3_x in the figure above. Similarly, Block 3 refers to the third block (conv4_x) out of the four blocks and Block 4 refers to the fourth block (conv5_x) of ResNet18.

These blocks consist of two basic blocks. The first basic block contained a convolutional layer, followed by a batch norm layer, a relu activation function, another convolutional layer, and then a batch norm layer. The second basic block was a copy of the first basic block.

The models were then trained on the 100 validation images for 3 epochs. Only the gradients of the block that was unfrozen was updated. The gradients of all of the other layers were not and were considered frozen.

TABLE I
EVALUATION METRICS OF ALL MODELS ON PNEUMONIA CLASSIFICATION OF X-RAY IMAGES ON TEST SET.

	Baseline (Finetuned Model)	Finetuned with Dropout	Unfroze Block 2	Unfroze Block 3	Unfroze Block 4
Accuracy	0.790	0.72845	0.784	0.791	0.7941
Precision	0.814	0.72845	0.828	0.817	0.815
Recall	0.923	1.00	0.889	0.919	0.928
F1 score	0.865	0.843	0.857	0.865	0.8679

TABLE II
CLASS BREAKDOWN.

	Baseline (Finetuned Model)	Finetuned with Dropout	Unfroze Block 2	Unfroze Block 3	Unfroze Block 4
Normal	43.30%	0%	50.50%	44.70%	43.50%
Pneumonia	92.30%	100%	88.90%	91.90%	92.80%

Layer Name	Output Size	ResNet-18
conv1	$112 \times 112 \times 64$	7×7, 64, stride 2
conv2_x	$56 \times 56 \times 64$	3×3 max pool, stride 2
		$\begin{bmatrix} 3 \times 3, 64 \\ 3 \times 3, 64 \end{bmatrix} \times 2$
conv3_x	$28 \times 28 \times 128$	$\begin{bmatrix} 3 \times 3, 128 \\ 3 \times 3, 128 \end{bmatrix} \times 2$
conv4_x	$14 \times 14 \times 256$	$\begin{bmatrix} 3 \times 3, 256 \\ 3 \times 3, 256 \end{bmatrix} \times 2$
conv5_x	$7 \times 7 \times 512$	$\begin{bmatrix} 3 \times 3, 512 \\ 3 \times 3, 512 \end{bmatrix} \times 2$
average pool	$1 \times 1 \times 512$	7×7 average pool
fully connected	1000	512×1000 fully connections
softmax	1000	

Fig. 3. Resnet18 Architecture

VI. EXPERIMENTS, RESULTS, AND DISCUSSION

A. Metrics

I used accuracy, precision, recall, and the F1 score as the primary metrics. The F1 score will be the main metric. Accuracy is the percentage of correctly classified images divided by the number of images classified.

A false positive is an image that had been classified as pneumonia but was really a normal x-ray. Icare about minimizing the number of false negatives, since the goal is to detect as many patients with pneumonia as possible. Ialso care about minimizing the number of false negatives, since people who do not have pneumonia should not be treated for it.

$$\text{precision} = \frac{\text{true positive}}{\text{true positive} + \text{false positive}} \tag{1}$$

$$\text{recall} = \frac{\text{true positive}}{\text{true positive} + \text{false negative}} \tag{2}$$

$$\text{F1 score} = 2 * \frac{\text{precision + recall}}{\text{true positive} + \text{false negative}} \tag{3}$$

B. Hyperparameters

I used a minibatch of 4, a learning rate of 0.001, the Adam optimizer, cross-entropy loss, and a training epoch of 3 across all models, including the pretrained model.

I chose a learning rate of 0.001 out of 0.01, 0.001, and 0.0001 after testing the models on the validation set using the baseline finetuned model trained on 3 epochs. A learning rate of 0.01 produced 71% of accuracy, compared to the learning rate of 0.001 that produced a 75% accuracy and the learning rate of 0.0001 that produced 0.77% accuracy. I chose a learning rate of 0.001 because the learning rate of 0.0001, while it had higher accuracy, trained the model too slowly. In contrast, the results produced by 0.01 learning rate had an accuracy that would not improve significantly despite increasing the pretrained model's epochs to 10 from 3 while keeping the finetuned model's training epochs at 3.

Next, I experimented with using the Adam optimizer compared to stochastic gradient descent. Despite increasing the training epochs from 10 to 3 for both pretraining and finetuned baseline models, I noticed that the training loss would remain constant when using stochastic gradient descent combined with momentum. On the other hand, using Adam showed a noticeable difference in decreasing training loss when pretraining the model. The loss went from 3663.334 and an accuracy of 75% to a loss of 3178.997 and an accuracy of 79%. However, the training loss stayed relatively constant for the finetuned model. Since Adam made a noticeable difference in the pretrained model, I continued using the Adam optimizer across all models.

I experimented with the number of epochs needed for pretraining and for training the models. I first pretrained the model on 10 epochs and then trained the finetuned baseline model for 3 and 5 epochs. At 3 epochs, the baseline model performed well with a training accuracy of 83% and a validation accuracy of 72%. However, after training the baseline model for 5 epochs, the baseline model showed clear signs of overfitting with a 95% training accuracy and a 59% validation accuracy. Since it was much easier to overfit the data when pretraining the model for 10 epochs, I experimented with pretraining the model for 3 epochs. This resulted in in a baseline that did not

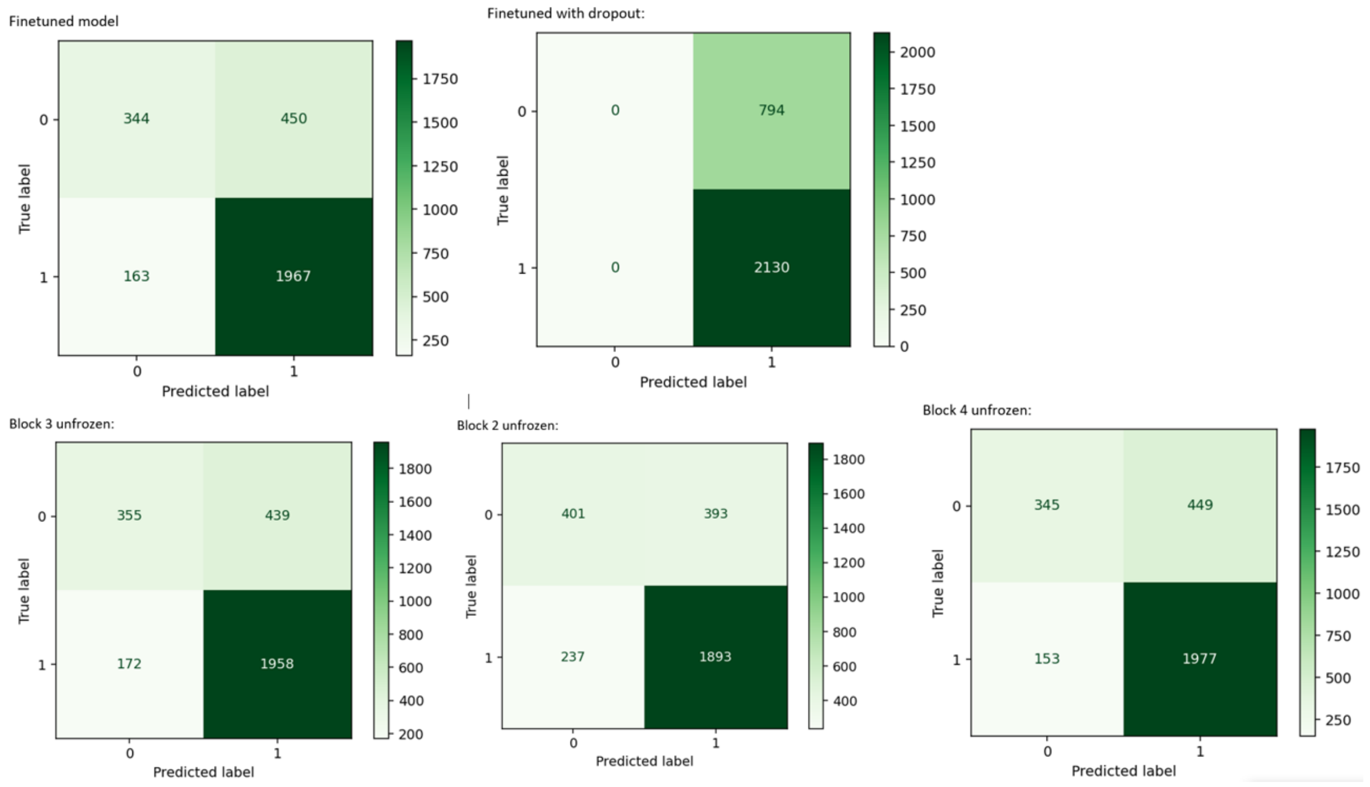

Fig. 4. Confusion matrix for all 5 models.

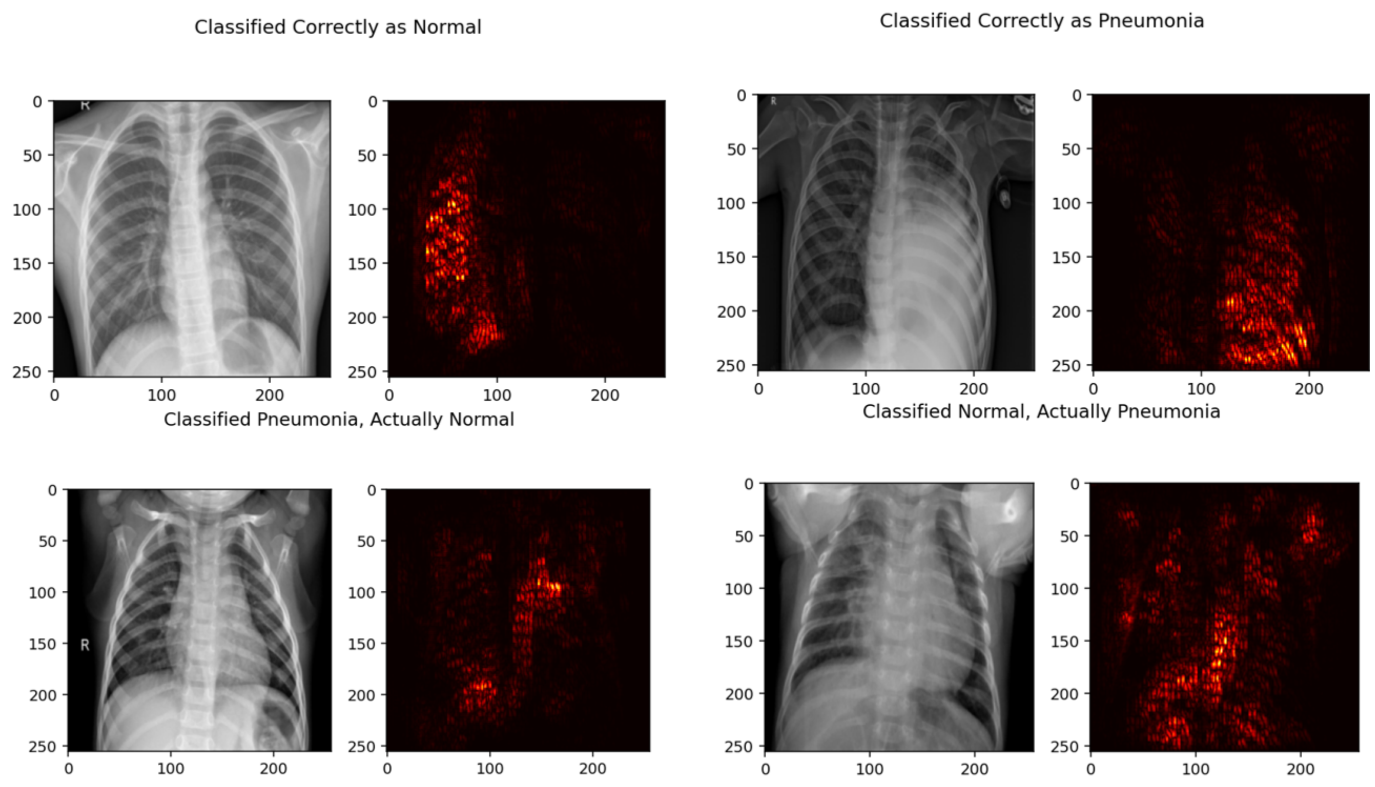
Classified Correctly as Normal
Classified Correctly as Pneumonia
Classified Pneumonia, Actually Normal
Classified Normal, Actually Pneumonia

Fig. 5. Saliency map using model with unfrozen Block 4 for correctly and incorrectly classified images.

easily overfit, so I chose to use 3 epochs for pretraining and 3 epochs for training the model.

I used a minibatch of 4 due to memory constraints of Google Colab.

C. Results of Metrics

We can see the results of the models in Table 1 and the confusion matrices in Figure 4. The best model was unfreezing layer 4 while keeping the rest of the model frozen. This produced an F1 score of 0.868. The second-best model was the baseline, with an F1 score of 0.865. Finally, the worst model was a finetuned model with a dropout layer with an F1 score of 0.842. This model had a perfect recall, which indicates that the model only learned to predict pneumonia for every single image. Since about 72% of the validation and test sets are pneumonia x-rays, finetuned with dropout produced an accuracy of 72.8%.

As hypothesized, unfreezing the middle or later blocks of the model produced similar or better results than finetuning the model. Isee that although unfreezing blocks 2, 3, and 4 produced similar results, unfreezing block 4 produced the best result, with unfreezing block 3 close behind. Ican see this most clearly in the differences in the recall scores. A higher recall score indicates a better identification of pneumonia. This may be because the subgroup distribution shift relied on sharing mostly similar fundamental features, and as a result the blocks unfrozen towards the output layer of the model were most necessary in distribution shift.

Since the hyperparameters were chosen to best prevent overfitting, the finetuned with dropout model may have performed the worse because it added regularization when it was not needed. It was most interesting that the finetuned model with dropout learned only to guess a single class every time. I believe this is due to the dropout probability being too high at p=0.5 as well as the position of the dropout layer being right before the output layer.

D. Class Prediction Breakdowns

Let's take a closer look at the class prediction breakdowns for each model in Table II.

We see that across all the models, all models performed better in identifying pneumonia compared to identifying normal images. This may be what is contributing to the low accuracy and f1 scores. As suspected, the finetuned model with dropout predicted only pneumonia for all of the images.

E. Overfitting

I do not believe any of the models experienced overfitting. If there was overfitting, there should have been a large difference in the performance between the training set and the validation set, and no such difference in performance was observed. The poor performance of the finetuned model with dropout suggests that overfitting was not an issue.

F. Saliency Maps

Next, let us examine the saliency images from the best performing model, unfreezing block 4, to see what the model learned. These maps are shown in Figure 4.

The presence of pneumonia is determined by small white spots in the lungs, which should show up on the x-ray. We expect the model to focus on the lungs and in particular white spots in the lungs that indicate pneumonia.

First, we see that in the case of a correctly classified normal x-ray, the model focuses on only one side of the chest, with a particular focus on the lung. The model's focus is represented by the red dots of color – the brighter the dots, the higher the focus. Next, in the case of a correctly classified pneumonia x-ray, the model also focuses on one side of the chest – this time, on the lower side of the chest.

In the false positive case, where the x-ray was normal but it was classified as pneumonia, we see that the model focused on both sides of the chest, one up high and one down below. Oddly, it also focuses on the spine and the ribcage, which may indicate the cause of the incorrect classification – the model had been focusing on the wrong spot. Finally, let's look at the false negative, when the x-ray was indicative of pneumonia but was classified as a normal x-ray. The model is again focusing on the bones of the body instead of the x-ray.

Overall, it seems that in the correctly classified images, the model focuses on the area where the lungs are. In contrast, when the model predicts x-rays incorrectly, the model focuses on the bones of the body. Since pneumonia presents as cloudy white spots in the lungs, one possible explanation is that the bones may have had some spots that mimicked pneumonia on the x-ray.

VII. Conclusion and Future Work

The best model that was produced was using surgical finetuning to freeze the last block in a pretrained ResNet18 model. This may be because of the distribution shift from training on adults to children. Since pneumonia presents similarly on x-rays regardless of age group, freezing earlier layers of the pretrained model ensures that only the parts of the model necessary for learning subgroup shifts is modified to produce better results. However, the finetuned model was a close second, followwed by the model with the unfrozen block 3, unfrozen block 2, and then the finetuned model with dropout.

A few areas of future work are needed to increase the F1 score and combat the confusion the model has when classifying images. First, the images should be preprocessed using an autoencoder to denoise the image. Additionally, the images should also be preprocessed to remove or reduce the brightness of the bones in the x-rays. These bones confused the model during incorrect predictions. What may also help is image segmentation to focus only on the lungs of the image.

Additionally, the model was highly sensitive to the 100 images used for finetuning the model. Future work should include testing out the best distributions and types of images for the best finetuning.

VIII. Contributions

I worked alone on this project. Some work was shared with CS 229, and I worked with a partner for CS 229.

References

[1] F. SHANN, "Etiology of severe pneumonia in children in developing countries," *The Pediatric infectious disease journal*, vol. 5, no. 2, pp. 247–252, 1986.

[2] J. R. Zech, M. A. Badgeley, M. Liu, A. B. Costa, J. J. Titano, and E. K. Oermann, "Variable generalization performance of a deep learning model to detect pneumonia in chest radiographs: a cross-sectional study," *PLoS medicine*, vol. 15, no. 11, p. e1002683, 2018.

[3] D. S. Kermany, M. Goldbaum, W. Cai, C. C. Valentim, H. Liang, S. L. Baxter, A. McKeown, G. Yang, X. Wu, F. Yan *et al.*, "Identifying medical diagnoses and treatable diseases by image-based deep learning," *Cell*, vol. 172, no. 5, pp. 1122–1131, 2018.

[4] A. U. Ibrahim, M. Ozsoz, S. Serte, F. Al-Turjman, and P. S. Yakoi, "Pneumonia classification using deep learning from chest x-ray images during covid-19," *Cognitive Computation*, pp. 1–13, 2021.

[5] H. Sharma, J. S. Jain, P. Bansal, and S. Gupta, "Feature extraction and classification of chest x-ray images using cnn to detect pneumonia," in *2020 10th International Conference on Cloud Computing, Data Science & Engineering (Confluence)*. IEEE, 2020, pp. 227–231.

[6] M. Arjovsky, L. Bottou, I. Gulrajani, and D. Lopez-Paz, "Invariant risk minimization," *arXiv preprint arXiv:1907.02893*, 2019.

[7] Y. Ro and J. Y. Choi, "Autolr: Layer-wise pruning and auto-tuning of learning rates in fine-tuning of deep networks," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, no. 3, 2021, pp. 2486–2494.

[8] Y. Lee, A. S. Chen, F. Tajwar, A. Kumar, H. Yao, P. Liang, and C. Finn, "Surgical fine-tuning improves adaptation to distribution shifts," *arXiv preprint arXiv:2210.11466*, 2022.

[9] P. Kirichenko, P. Izmailov, and A. G. Wilson, "Last layer re-training is sufficient for robustness to spurious correlations," *arXiv preprint arXiv:2204.02937*, 2022.

[10] G. Shih, C. C. Wu, S. S. Halabi, M. D. Kohli, L. M. Prevedello, T. S. Cook, A. Sharma, J. K. Amorosa, V. Arteaga, M. Galperin-Aizenberg *et al.*, "Augmenting the national institutes of health chest radiograph dataset with expert annotations of possible pneumonia," *Radiology. Artificial intelligence*, vol. 1, no. 1, 2019.

[11] D. Kermany, K. Zhang, M. Goldbaum *et al.*, "Labeled optical coherence tomography (oct) and chest x-ray images for classification," *Mendeley data*, vol. 2, no. 2, 2018.

[12] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, "Pytorch: An imperative style, high-performance deep learning library," in *Advances in Neural Information Processing Systems 32*. Curran Associates, Inc., 2019, pp. 8024–8035. [Online]. Available: http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf

[13] Wes McKinney, "Data Structures for Statistical Computing in Python," in *Proceedings of the 9th Python in Science Conference*, Stéfan van der Walt and Jarrod Millman, Eds., 2010, pp. 56 – 61.

[14] C. R. Harris, K. J. Millman, S. J. van der Walt, R. Gommers, P. Virtanen, D. Cournapeau, E. Wieser, J. Taylor, S. Berg, N. J. Smith, R. Kern, M. Picus, S. Hoyer, M. H. van Kerkwijk, M. Brett, A. Haldane, J. F. del Río, M. Wiebe, P. Peterson, P. Gérard-Marchant, K. Sheppard, T. Reddy, W. Weckesser, H. Abbasi, C. Gohlke, and T. E. Oliphant, "Array programming with NumPy," *Nature*, vol. 585, no. 7825, pp. 357–362, Sep. 2020. [Online]. Available: https://doi.org/10.1038/s41586-020-2649-2

[15] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg *et al.*, "Scikit-learn: Machine learning in python," *Journal of machine learning research*, vol. 12, no. Oct, pp. 2825–2830, 2011.

[16] G. Bradski, "The OpenCV Library," *Dr. Dobb's Journal of Software Tools*, 2000.

[17] P. Umesh, "Image processing in python," *CSI Communications*, vol. 23, 2012.

[18] J. D. Hunter, "Matplotlib: A 2d graphics environment," *Computing in science & engineering*, vol. 9, no. 3, pp. 90–95, 2007.

[19] G. Van Rossum, *The Python Library Reference, release 3.8.2*. Python Software Foundation, 2020.

[20] Python package index - pypi. [Online]. Available: https://pypi.org/