CS7643 Project: Enhancing Pneumonia Detection with Ensemble Deep Learning Models

Ayman E., Alishan P., Sudharshan R.

07/30/2024

1 Introduction

1.1 Background

Pneumonia is a serious infection caused by bacteria, fungi and viruses. A severe and potentially fatal consequence of pneumonia is pleural effusion, a condition in which excess mucus builds up between the lungs and chest wall. In many underdeveloped nations, pneumonia is highly prevalent due to factors such as lack of clean water, poor sanitation facilities, overcrowded and unhygienic living conditions, insufficient vaccination coverage and a shortage of medical professionals. Chest X-Rays are currently the most common method of diagnosing pneumonia, and early detection can significantly increase the likelihood of curability and survival.

Past research has tackled this problem using convolutional neural networks (CNNs), with traditional standalone CNNs such as DenseNet-201 and ResNet-152 performing poorly [1]. For the first time in 2021, an ensemble technique was used to classify pneumonia in X-Ray images where transfer learning models were used as base learners, which this project aims to explore further. Developing a reliable technology which can accurately detect pneumonia in chest X-Ray images would be of crucial importance to assist the limited number of healthcare professionals in these regions where medical infrastructure is inadequate. It is for these reasons that we are motivated to address the critical healthcare challenges faced by these underdeveloped nations. We aim to use advanced deep learning models for pneumonia detection to improve patient healthcare and provide assistance to medical professionals by enhancing diagnostic capabilities [9].

1.2 Objectives

Our goal is to assist healthcare professionals in diagnosing pneumonia with greater accuracy and efficiency. We set out to enhance the accuracy of pneumonia detection using chest X-ray images by leveraging deep learning models. For this project, our primary objective was to reproduce results from [1], which included developing and fine tuning base models (ResNet18, DenseNet121, GoogLeNet) and an ensemble of said models used by [1]. This effort aims to provide a robust tool that can be particularly beneficial in regions with limited access to experienced radiologists.

1.3 Current Practice and Limitations

Currently, pneumonia detection relies heavily on manual examination of chest X-rays by radiologists. This method is not only time-consuming but also susceptible to human error and variability in diagnostic accuracy. Automated systems exist but often fall short in terms of precision and robustness, particularly when applied across diverse patient demographics and different datasets. These limitations highlight the need for more reliable and generalized automated diagnostic tools.

1.4 Impact of Success

If successful, our project could significantly enhance diagnostic accuracy and efficiency, thus improving patient outcomes. An effective automated pneumonia detection system would be a valuable asset for healthcare providers, enabling quicker and more accurate diagnoses. This is especially crucial in underdeveloped regions where access to skilled radiologists is scarce, and timely diagnosis can significantly affect patient survival rates and overall healthcare quality.

1.5 Data Description

Our research utilized the RSNA Pneumonia Detection Challenge dataset from Kaggle. The RSNA dataset comprises over 30,000 anonymized chest X-ray images, labeled to indicate the presence or absence of pneumonia. Key aspects of the dataset include:

- **Data Collection:** Images were gathered from various hospitals and diagnostic centers.
- **Labeling Process:** Each image was annotated by certified radiologists to ensure labeling accuracy.
- **Data Splits:** The dataset was divided into training and test sets to facilitate model development and evaluation.

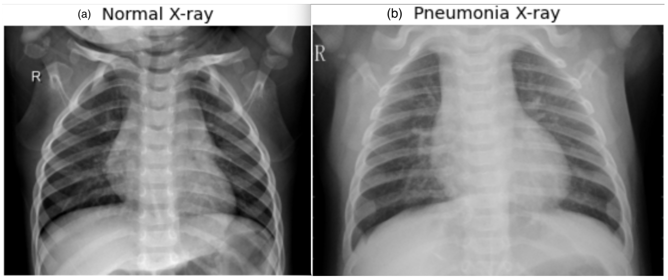

Figure 1: Examples of chest X-ray images from the RSNA dataset: (a) Normal lung, (b) Pneumonic lung

2 Approach

2.1 Methodology

We implemented, trained and fine-tuned (hyperparameter tuning) three convolutional neural network (CNN) models: GoogLeNet, ResNet-18, and DenseNet-121. These specific models were used since they were part of the ensemble in [1]. In short, ResNet-18 uses residual learning to combat the vanishing gradient problem; DenseNet-121 consists of Dense blocks (receiving input from all preceding layers in a feedforward manner), which helps improve gradient (information) flow; GoogLeNet implements the inception module, which consists of multiple convolutional layers in parallel and concatenating their inputs and helps capture features at various scales. Cross entropy loss function was chosen to help with this binary categorization problem.

The paper [1] did not fine tune each of the base models, so this part is new in our approach. Each model was trained separately on the RSNA dataset and also on an augmented version of the dataset. The latter approach (tuning with augmented RSNA dataset) is new and was not performed in [1]. The predictions from these models were then combined using an ensemble method where weights were assigned via a novel hyperbolic tangent function (used in [1]). This ensemble approach aimed to leverage the strengths of each model to enhance overall detection accuracy. We also chose not to reproduce the 5 fold cross validation technique used in [1] on the ensemble because of time and resource constraints, and would recommend this for future work.

2.2 Anticipated and Encountered Problems

We anticipated challenges such as over-fitting and computational resource constraints. We anticipated over-fitting because the RSNA dataset is only 30,000 images large, whereas each of the base models have been pre-trained on millions of images (which shows their architecture capacity to learn). To counter this, we applied data augmentation techniques, including random rotations, shifts, and flips, to artificially expand the dataset, which were not employed in [1]. Additionally, there was an imbalance between pneumonia-positive and pneumonia-negative cases, because of which we anticipated model bias toward learning about more positive samples - we did not perform any resampling or augmentation for this (Figure 2). Because of the large size of the RSNA dataset, we anticipated large tuning times - each epoch took around 8-11 minutes to train on GPU in Google Colab, which severely limited further fine-tuning/transfer learning strategies we could attempt. We increased batch size to mitigate this, but this was minimally effective.

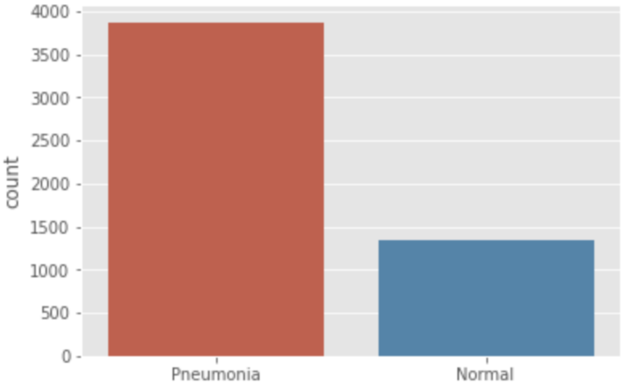

Figure 2: Data proportion in RSNA dataset [10].

2.3 Experimental Setup

We used the Adam optimizer with various learning rates and batch sizes. We did not vary momentum and other parameters because of time and computational resource constraints, and would like to recommend this for future work. Data augmentation was applied during training to prevent over-fitting and to enable better generalization. The ensemble model combined the predictions of the three networks, with the final output being a weighted average of the individual predictions. Each of the base models were downloaded as pre-trained and were trained for 10 epochs on the RSNA data. None of the layers were frozen because freezing most of the layers resulted in far less accuracy (about -0.1 lesser) than not freezing any layers. We suggest experimenting more with optimizing freezing layers for future work, since this might impact base model performance.

2.4 Novel Hyperbolic Tangent Approach

In our ensemble model, weights assigned to each base learner were determined using a novel strategy involving the hyperbolic tangent function, adopted from [1]. The weights were computed based on four evaluation metrics: precision, recall, F1-score, and AUC-ROC. This method ensures that the ensemble leverages the strengths of each model effectively, leading to improved overall performance.

3 Experiments and Results

3.1 Model Structure and Parameters

The models we used have several learned parameters within their convolutional layers. The input consisted of preprocessed chest X-ray images resized to 224x224 pixels, and the output was a binary classification indicating the presence or absence of pneumonia. The loss function employed was binary cross-entropy, suitable for our binary classification task. We tried implementing an ensemble model leveraging the performance of all 3 models. 2 ensemble models were implemented and tested, based on base models trained on the original dataset, and augmented dataset respectively.

3.2 Generalization and Overfitting

To combat overfitting, we used data augmentation techniques such as random rotations, flips, and shifts. Regularization methods like dropout and batch normalization were also implemented. Our models showed good generalization capabilities on the test set, indicating the effectiveness of our approach.

3.3 Hyperparameters and Optimization

Hyperparameters such as learning rate, batch size, and the number of epochs were tuned through experimentation. We used the Adam optimizer for its balance of convergence speed and stability. The learning rate was set to 10^{-4}, and the batch size was 32. These hyperparameters were chosen based on their performance during preliminary experiments on the original dataset.

3.4 Framework and Starting Points

We utilized the PyTorch framework due to its flexibility and extensive support for model building and training. Our starting points included pre-trained models from the torchvision library, which were fine-tuned on our dataset to improve performance.

3.5 Measuring Success

Success was measured using several evaluation metrics: accuracy, precision, recall, F1-score, and the Area Under the Receiver Operating Characteristic Curve (AUC-ROC). These metrics provided a comprehensive assessment of model performance, capturing both the ability to correctly identify pneumonia cases (sensitivity) and the ability to correctly identify non-pneumonia cases (specificity).

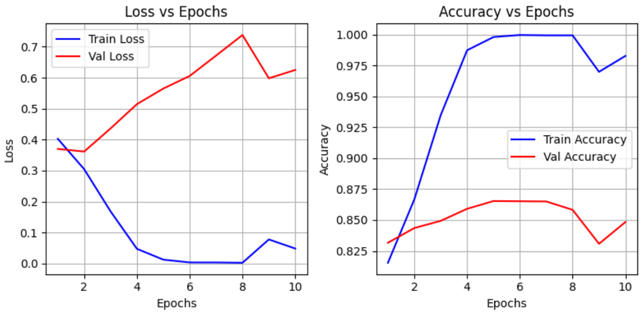

Figure 3: Learning curve for GoogleNet

3.6 Results

Each base model was downloaded pretrained and a 2-node fully connected layer was added at the end of each architecture. No model layers were frozen. The transfer learning strategy was not optimized/experimented with due to lack of time and [1] did not mention their strategy. All three base models had this structure. Each model was individually fine-tuned on the RSNA dataset to achieve the best possible performance within the time and resource constraints. We varied learning rate and batch size for all three base models.

GoogleNet on Original Dataset: Based on the performance of the model based on various learning rates, we found that the model performed best when the learning rate was 0.0001 and batch size was 128. We varied learning rate from 0.001 to 0.0001 and the batch size from 64 to 256. This model had a **validation accuracy of 83.75%**. Interestingly, validation accuracy dropped when learning rate was increased or decreased (to about 0.77-0.79). This might be because the model was learning too fast or too slow and was not traversing the loss function at the right pace. A higher batch size decreased accuracy to about 0.78 and a lower batch size increased training time - so we picked 128 to strike a good balance. However, the **training accuracy obtained was 98%**, which means that over-fitting was occurring. This was expected because of the large capacity of each of the base models. In theory, we could have added in regularization such as dropout or fiddled with the regularization variable in the adam optimizer to optimize this, but were not able to because of the large training time for this dataset. With the model being trained on the original dataset, almost in all variants, the training accuracy was much higher, when compared to the validation accuracy, almost always >90%.

ResNet18 on Original Dataset: For this model, the learning rates, batch sizes and weight decays were varied and tuned. Based on the performance of the model based on various parameters, it was found that the model performed best when the learning rate was 0.0001 and batch size was 64. This model had a **validation accuracy of 85.3%**. However, the **training**

accuracy obtained was 99%. This made the possibility of over fitting evident. With the model being trained on the original dataset, almost in all variants, the training accuracy was much higher, when compared to the validation accuracy, almost always >90%.

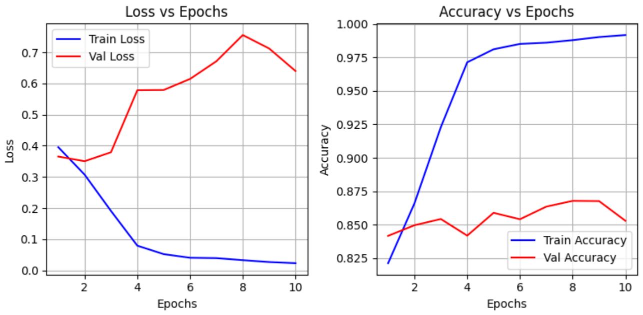

Figure 4: Learning Curve for ResNet18

DenseNet121 on Original Dataset: Based on the performance of the model based on various learning rates, it was found that the model performed best when the learning rate was 0.001 and batch size was 96. This model had a **validation accuracy of 84%**. However, the **training accuracy obtained was 90%**, which is also a big indicator of over-fitting. With the model being trained on the original dataset, almost in all variants, the training accuracy was much higher, when compared to the validation accuracy, almost always >=90%.

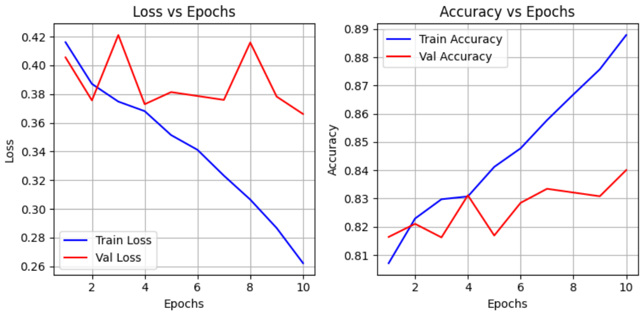

Figure 5: learning Curve for DenseNet121

Need for Data Augmentation:
From the above best performing models, each trained from the original dataset, we can note one similarity: we observe that the train accuracy is very high in comparison to the validation accuracy. This would mean that our models may not generalize well on unseen chest xray scans. The solution that we tried and tested for this particular problem, is augmenting the data and training our models on that data, especially because it was not done in [1].

GoogleNet on Augmented Dataset:
For Data Augmentation, we used random horizontal flips and rotations of up to 20 degrees. This methodology should ideally bring more variance to the train data, and should stop the models from over-fitting. We stayed away from very harsh rotations because our validation data did not have this issue - xray scans are generally provided upright so we did not see the need to rotate too much. The **Validation accuracy was now 83.6%**, however the **Training accuracy was 89%**. This shows that overfitting has decreased (which was one of our hopes) but we also note a slight decrease in validation accuracy; the latter was unexpected, though not entirely. Since x-ray scans are always upright in our case, we expected our validation accuracy to increase only slightly; however, we actually observe slightly lower validation accuracy, which suggests that data augmentation might not be providing much value in terms of model's ability to generalize. We see this pattern in both the next models when trained on the augmented dataset: DenseNet and ResNet, as below.

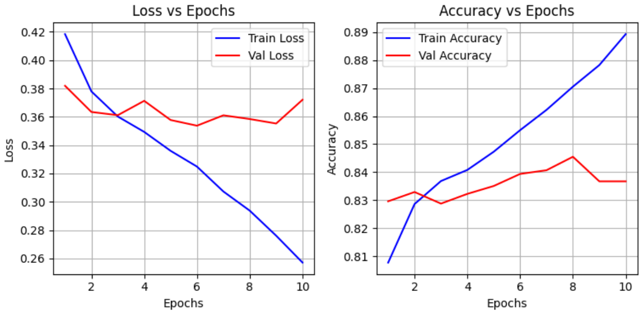

Figure 6: Learning curve for GoogleNet with augmented dataset

DenseNet121 on Augmented Dataset:
After augmenting the train dataset with random horizontal flips and rotations, we found that the **Validation accuracy slightly dropped to 82%** with the corresponding **training accuracy dropping to 83%**.

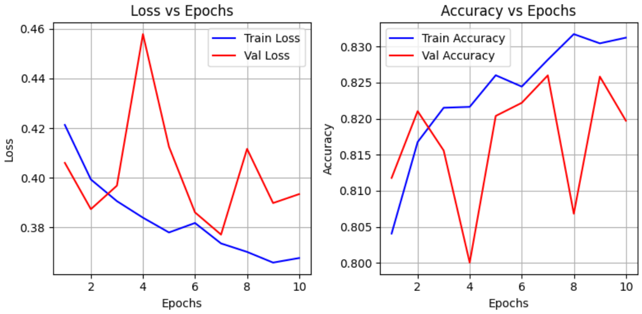

Figure 7: Learning curve for DenseNet121 with augmented dataset

ResNet18 on Augmented Dataset:
After augmenting the train dataset with random horizontal flips and rotations, the **Validation accuracy**

remained at 84.5% with the corresponding **training accuracy decreased to 87.8%**. The order of ascending performance was found to be: ResNet, GoogLeNet, DenseNet. It seems like perhaps information propagation (residual connections in ResNet and inception module in GoogLeNet) is helpful to generalize for xray images for pneumonia detection.

Overall, the loss curves tend to fluctuate more with training with augmented data, especially with DenseNet in Figure 7, which was interesting. This make it seem like the training is more unstable, about which we are not sure why this might be the case.

On another note, something we could have tried, if given more time/computational resources, would have been to append the augmented dataset to the original train dataset (instead of randomly augmenting samples of the original dataset in place). This might output at least the same accuracy as the regular dataset, if not better, instead of lowering the validation accuracy.

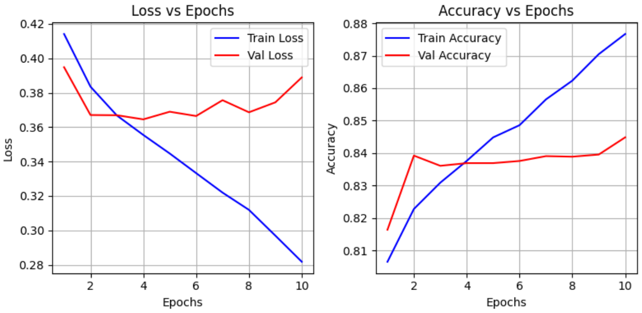

Figure 8: Learning curve for ResNet18 with augmented dataset

Ensemble Model: Each of the above models, captures different patterns in the data and has its own unique perspective. We wanted to test if combining the strengths of all 3 models, if we could improve the overall performance and robustness of its predictive capabilities.

We tested by using an ensemble model, where, we incorporated using hyperbolic tangent function for determining the weights given to each of the models, based on their original performances. Two ensemble models were tried and tested, in order to understand if and how they improved the performance of the model.

Ensemble Model with Original dataset: The first ensemble model that was tried, was combining and incorporating the first 3 neural networks that were trained on the original dataset. Our ensemble model achieved the following results on the test set:

- **Accuracy:** 84.03%
- **Precision:** 78.66%
- **Recall:** 66.14%
- **F1-Score:** 71.44%
- **AUC-ROC:** 0.78

Ensemble Model with Augmented dataset: The next ensemble model we tried was combining and incorporating the 3 neural networks trained with the augmented dataset, in order to circumvent over-fitting and achieve better overall performance. Our ensemble model achieved the following results on the test set:

- **Accuracy:** 83.4%
- **Precision:** 76.22%
- **Recall:** 64.04%
- **F1-Score:** 70.6%
- **AUC-ROC:** 0.77

On comparing the 2 models, the first model is seen to have better overall performance; however, even though it tends to overfit. We expected this since all three individual models' performances dropped when trained on the augmented dataset. Given more resources, we could have 1) trained on various augmentation iterations and 2) implemented 5 fold cross validation as implemented by [1] to see if our performance improves.

3.7 Impact of Data Augmentation

While data augmentation is crucial for improving the robustness of models, it also introduces additional noise and variability. The results indicate that while the models trained on augmented data performed slightly worse in terms of accuracy, they are likely more generalizable to diverse datasets not seen during training.

4 Conclusion

In conclusion, our project achieved close accuracy to [1] without using 5-fold cross validation technique. The combination of GoogLeNet, ResNet-18, and DenseNet-121, with weights assigned through a hyperbolic tangent approach, demonstrated an inferior performance compared to individual models, especially the ResNet18 without data augmentation, which alone provided 85% validation accuracy. One thing to note would be that we split the training data 80/20 train:validation ratio, which might not have been the specs [1] used (unmentioned in paper). Our methodology of data augmentation decreased overfitting but also decreased performance slightly, which was not expected. Nevertheless, the results of this project have significant potential for improving diagnostic capabilities in healthcare, especially in regions with limited access to experienced radiologists.

Author	Contributions
Ayman Elsaedi	Provided a boiler plate implementation for the project. Implemented and trained the DenseNet-121 model. Conducted data preprocessing and augmentation. Implemented and evaluated the ensemble on the augmented dataset. Analyzed the reasoning of results. Drafted the introduction, approach, experiments, and results sections. Created visualizations and figures. Primary owner of DenseNet-121 and training with augmentation.
Alishan Premani	Provided visualization implementation for the project. Implemented and trained the GoogLeNet model. Conducted data preprocessing and augmentation. Implemented and evaluated the ensemble on the augmented dataset. Analyzed the reasoning of results. Contributed to the introduction, approach, experiments, and results sections. Created visualizations and figures. Primary owner of GoogLeNet and training with augmentation.
Sudharshan Ramesh	Implemented and trained the ResNet-18 model. Conducted data preprocessing and augmentation. Implemented and evaluated the ensemble on the augmented dataset. Analyzed the reasoning of results. Contributed to the introduction, approach, experiments, and results sections. Created visualizations and figures. Primary owner of ResNet-18 and hyperbolic tangent ensemble implementation.

Table 1: Author Contributions

References

[1] Kundu, R., Das, R., Geem, Z. W., Han, G. T., & Sarkar, R. (2021). Pneumonia detection in chest X-ray images using an ensemble of deep learning models. *PLoS One*, Sep 7. https://www.ncbi.nlm.nih.gov/pmc/articles/PMC8423280/

[2] Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., & Rabinovich, A. (2015). Going deeper with convolutions. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 1-9. (GoogLeNet)

[3] Huang, G., Liu, Z., Van Der Maaten, L., & Weinberger, K. Q. (2017). Densely connected convolutional networks. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 4700-4708. (DenseNet-121)

[4] He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 770-778. (ResNet-18)

[5] Kermany, D., Zhang, K., & Goldbaum, M. (2018). Labeled Optical Coherence Tomography (OCT) and Chest X-Ray Images for Classification. *Mendeley.*

[6] Wang, X., Peng, Y., Lu, L., Lu, Z., Bagheri, M., & Summers, R. (2017). Chestx-ray8: Hospital-scale chest x-ray database and benchmarks on weakly-supervised classification and localization of common thorax diseases. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2097-2106.

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

[8] Ibrahim, A., Ozsoz, M., Serte, S., Al-Turjman, F., & Yakoi, P. (2021). Pneumonia classification using deep learning from chest X-ray images during COVID-19. *Cognitive Computation*, 1-13. https://doi.org/10.1007/s12559-020-09787-5

[9] Premani, A., Ramesh, S., Elsaidi, A,. Deep Learning Project Proposal, 2024.

[10] Chanchal. (2024). Pneumonia Detections Using Deep Learning. *Kaggle.* https://www.kaggle.com/code/chanchal24/pneumonia-detections-using-deep-learning