Effects of Hyperparameters and Data Augmentation on Deep Learning Models for Pneumonia Classification from Chest X-ray Images

Saanvi Chawla, Morteza Sarmadi

Abstract

More than 2 million people die annually from pneumonia around the world. Pneumonia is the second most misdiagnosed condition leading to hospital readmission, which displays the difficulty of accurate and timely diagnosis. Various studies have applied deep learning (DL), including hybrids and ensembles, to medical analysis, yielding promising results. However, these models utilized have led to high computational costs as well as a consumption of resources. The datasets used to train the models have also shown class imbalance and a lack of diversity in the data. In this paper, we analyzed the effects of hyperparameters on DL models to optimize the performance of these models trained on a balanced Kaggle pneumonia chest X-ray dataset. We evaluated five architectures (MobileNetV2, VGG16, InceptionV3, EfficientNetB0, ResNet50) across four hyperparameters– learning rate, batch size, target epochs, and data augmentation– conducting 450 total simulations. After accumulating the data, feature importance was rank-ordered using XGBoost (XGB) and Random Forest (RF), with decision trees extracted from RF. Key findings of this study involved learning rate being the most influential hyperparameter and augmentation resulting in higher accuracy and F1-score than non-augmented data. Furthermore, RF and XGB hyperparameter importance ranking varied within each model but remained the same whether evaluated by accuracy or F1-score, with XGB consistently achieving a higher predictive fit. This study provides a practical framework for future work related to optimizing DL models for classifying pneumonia or other diseases based on medical images.

Introduction

Pneumonia is a serious disease that affects millions of people internationally. In 2024, more than 40,000 people died in the United States alone [1]. In general, pneumonia is the cause of 14% of all deaths for children under 5 [2]. Pneumonia takes the lives of around 2000 kids under 5 each day, with the majority of these deaths being preventable [3]. With limited access to vaccines, clinics, and doctors, it is more challenging both to diagnose it in the first place and receive treatment for pneumonia. Hospitalized patients with community-acquired pneumonia spent around $37,948 in total during the acute phase, while patients in the control group spent $2075 [4]. Although diagnosis is an important part of the treatment, especially if it is done in a timely manner, diagnosing pneumonia is difficult to do in the first place because its symptoms are

shared by other diseases too, and there is no conclusive test for verifying the presence of pneumonia [5, 6, 7]. Recent studies have demonstrated that DL has been verified to play a crucial role in improving the quality of medical diagnosis, as seen in multiple papers [2, 8, 9, 10, 11]. DL has improved pneumonia prediction and enhanced medical decision-making, helping both patients and individuals in the field [2]. Additionally, DL frameworks have been able to yield accuracies exceeding 99% and even reaching 100%, displaying how this can be as effective, if not more, than human diagnosis [12].

The application of DL in diagnosis of pneumonia has been investigated by many researchers. Xu and Zhang used the pneumonia MNIST dataset that indicated a class imbalance, constructed a ResNet18-SA classification model, and also implemented data augmentation [13]. Their model, ResNet18-SA, showed superior results, reaching 95.83%, 95.87%, 95.21%, and 95.52%, in accuracy, precision, recall, and F1 score, respectively, compared to conventional models such as VGG16, ResNet50, and MobileNetV2. Their original dataset started off with only 5856 original, non-augmented images [13]. Furthermore, Han et al. introduced a framework that employed contrastive learning and found that their models outperformed baseline models. Specifically, they found that ResNet-18Radi and ResNet-18AttRadi (their models) achieved superior performance, with accuracies of 85.1% and 88.6%, respectively, compared to the baseline models ResNet-18 and ResNet-18Att, which achieved accuracies of 76.3% and 81.5%, respectively [11]. A study by Slimi et al. in 2025 implemented a method that relied on GRU and Spiking Neural Networks (SNN), but when GRU was removed, the model's (hybrid SNN-GRU) performance worsened. With GRU, the model not only achieved a high accuracy of 99.35% but also had a faster learning rate [14]. According to Effah et al., ensemble ML models RF and XGB were the best models to distinguish pneumonia from other lower respiratory tract infections, with RF and XGB achieving an accuracy of 92% and 90.8%, respectively [10]. Additionally, as seen in the study by Bhattarai et al., CNNs, specifically the DenseNet121, performed best for classification only, achieving a recall of 92% [15].

The studies presented above display the potential of using DL for various models and even ensembles, but there are a variety of models and hyperparameters implemented across the publications. Since there is no methodical study to analyze how different hyperparameters affect each model, it remains uncertain particularly using chest X-ray images which hyperparameters can improve the quality of diagnosis, how they compare to each other regarding their impact on the models' performance, and importance on the model. Various approaches have been used to explore this, but there is a lack of understanding [9, 11, 14]. This demonstrates the need to gain insight on the impact of different hyperparameters for developing DL models focused on chest X-ray datasets for pneumonia.

In this paper, we investigated the performance across a list of models and hyperparameter combinations. To this end, we investigated five highly prevalent models utilized for classification

of chest X-ray images for pneumonia patients: MobileNetV2, VGG16, InceptionV3, EfficientNetB0, and ResNet50. The hyperparameters investigated were learning rate, batch size, augmentation, and number of epochs. Accordingly, 90 runs per model, accounting for 450 total runs, were conducted to achieve a full factorial design matrix accounting for all combinations of hyperparameters. We investigated accuracy and F-1 score to determine the best performing model based on all combinations of hyperparameters, and which hyperparameters improved the accuracy and F1-score more significantly across all models as a criteria to identify both the most effective models and influential hyperparameters. This study can serve as a preliminary resource providing insight into the top-tier models for pneumonia detection based on chest X-ray images as well as understanding the impact of crucial hyperparameters on model performance.

Materials and Methods

We used a Kaggle dataset of 8,530 chest X-ray images with two classes: NORMAL and PNEUMONIA [16]. The dataset is balanced as there are 15 files (0.35%) for each class in the testing folder, 3400 files (79.72%) for each in the training folder, and 850 files (19.93%) for each in the validation folder. The equal distribution is ideal since the models have the same amount of samples to learn from for each class. We implemented image augmentation, including random flips, rotation, zooming, and contrast, on the training set only. An example of a few samples before and after augmentation is shown in Figure 1.

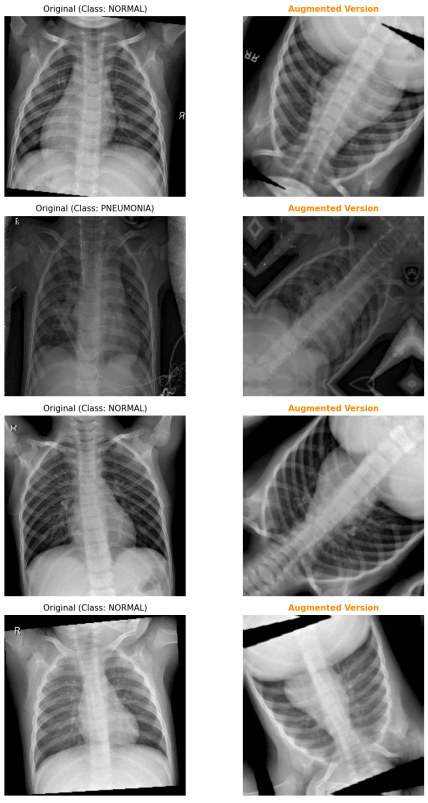


Fig 1. Examples of original images versus the augmented versions

We implemented a full factorial design of experiments (DOE) approach as shown in Table 2: we explored all the possible combinations of parameters. Figure 2 visualizes the data processing pipeline. We ran 5 different models on the images: MobileNetV2, VGG-16, Inception-V3, EfficientNet, and ResNet-50, selecting these models due to their prevalence across various studies [2, 5, 6, 14, 15]. To see the effects of hyperparameter optimization, we chose to tweak the learning rate, number of epochs, batch size, and augmentation (Table 1).

Table 1. Overview of models and hyperparameters used in this study

Inputs	Input 1	Input 2	Input 3	Input 4	Input 5	Count
Models	MobileNetV2	VGG16	Inception V3	EfficientNet	ResNet50	5
Learning rate	0.01	0.001	0.0001			3
Epoch	10	15	20	30	50	5
Batch size	16	32	64			3
Augmented?	Yes	No				2

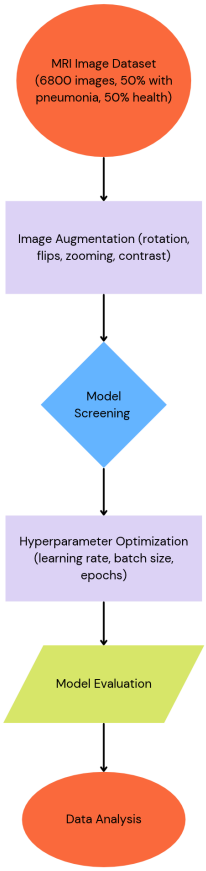


Fig 2. Flow chart summarizing the data modeling pipeline

All DL models and hyperparameter evaluations were implemented in Python (version 3.13.15). Model training used TensorFlow (version 2.19.0) and Keras (version 3.13.2) for convolutional neural network training, with NumPy (version 2.4.6) and pandas (version 2.3.3). Feature importance analysis used scikit-learn (version 1.6.1) and pandas (version 2.3.3), with XGBoost 3.2.0 and NumPy 2.0.2 in one analysis notebook and XGBoost 3.4.1 and NumPy 2.1.3 in the other. The Python code was executed on the Kaggle platform with a GPU T4 x2 accelerator, with the total elapsed training time for each model ranging from 12 to 18 hours. The complete source code and notebooks are publicly accessible via Kaggle (https://www.kaggle.com/code/chawlas/notebook4b1b348f25, https://www.kaggle.com/code/chawlas/analyzing-results, https://www.kaggle.com/code/chawlas/analysis-of-most-impactful-hyperparameter).

Results

We took the average accuracy and F1-score for each model (see Table 2). We obtained accuracies of 0.9211, 0.9241, 0.9285, 0.9311, and 0.9381 for VGG16, MobileNetV2, InceptionV3, EfficientNetB0, and ResNet50, respectively. F1-scores of 0.9206, 0.9230, 0.9278, 0.9307, and 0.9378 were yielded for each model, respectively. VGG16 had the lowest STD for accuracy and F1-score of 0.0306 and 0.0310. On the other hand, the highest STD was 0.0455 and 0.0475 for both MobileNetV2's accuracy and F1-score.

Table 2. Accuracy and F-1 score for each model across all runs

	Accuracy					F-1 Score				
	VGG16	**Mobile Net V2**	**Incept ion V3**	**Efficient NetB0**	**ResNet 50**	**VGG16**	**Mobile Net V2**	**Incept ion V3**	**Efficient NetB0**	**ResNet 50**
All Runs	0.9211	0.9241	0.9285	0.9311	0.9381	0.9206	0.9230	0.9278	0.9307	0.9378
STD	0.0306	0.0455	0.0454	0.0346	0.0355	0.0310	0.0475	0.0465	0.0348	0.0358

Increasing the learning rate from 0.0001 to 0.001 resulted in the highest accuracy and F1-score improvement across all models, with the accuracy and F1-score for EfficientNetB0 increasing from 0.9000 to 0.9467 (+5.19%) and 0.8994 to 0.9465 (+5.24%), respectively (see Table 3). ResNet50 also improved in both metrics, as the accuracy increased from 0.9144 to 0.9556 (+4.51%) and the F1-Score increased from 0.9139 to 0.9554 (+4.54%). The lowest STD was at a learning rate of 0.001, with EfficientNetB0 only having an STD of 0.0188 for accuracy. The highest STD was at a learning rate of 0.0001, since MobileNetV2 had the highest STDs for accuracy and F1-Score of 0.0547 and 0.0579.

Table 3. Accuracy and F-1 score across the models while modifying the learning rate

Learning Rate	Accuracy					F-1 Score				
	VGG16	Mobile Net V2	Inception V3	Efficient Net B0	ResNet 50	VGG16	Mobile Net V2	Inception V3	Efficient Net B0	ResNet 50
0.0001	0.8986 (±0.0302)	0.8944 (±0.0547)	0.9156 (±0.0545)	0.9000 (±0.0339)	0.9144 (±0.0286)	0.8980 (±0.0304)	0.8921 (±0.0579)	0.9146 (±0.0558)	0.8994 (±0.0340)	0.9139 (±0.0289)
0.001	0.9333 (±0.0196)	0.9355 (±0.0262)	0.9400 (±0.0308)	0.9467 (±0.0188)	0.9556 (±0.0308)	0.9330 (±0.0197)	0.9351 (±0.0267)	0.9396 (±0.0314)	0.9465 (±0.0189)	0.9554 (±0.0309)
0.01	0.9233 (±0.0329)	0.9422 (±0.0360)	0.9300 (±0.0458)	0.9467 (±0.0257)	0.9444 (±0.0343)	0.9228 (±0.0336)	0.9418 (±0.0368)	0.9292 (±0.0470)	0.9464 (±0.0260)	0.9441 (±0.0345)

Epochs demonstrated the least consistent pattern, although the general trend was that as the number of epochs increased, the performance metrics increased (seen in Table 4). This can be seen in EfficientNetB0, as the accuracy improved from 0.9204 at 10 epochs to 0.9426 (+2.41%) at 50 epochs. The F1-score also jumped from 0.9200 to 0.9423 (+2.42%). The lowest STD was at 15 epochs, with VGG16 at 0.0259 for accuracy and 0.0263 for the F1-score. The highest STD was at 20 epochs, with MobileNetV2 at 0.0563 and 0.0591 for accuracy and F1-score.

Table 4. Accuracy and F-1 score across the models while modifying number of epochs

Epochs	Accuracy					F-1 Score				
	VGG16	Mobile Net V2	Inception V3	Efficient Net B0	ResNet 50	VGG16	Mobile Net V2	Inception V3	Efficient Net B0	ResNet 50
10	0.9185 (±0.0400)	0.9278 (±0.0502)	0.9296 (±0.0456)	0.9204 (±0.0364)	0.9222 (±0.0343)	0.9181 (±0.0402)	0.9266 (±0.0530)	0.9292 (±0.0461)	0.9200 (±0.0366)	0.9217 (±0.0346)
15	0.9130 (±0.0259)	0.9278 (±0.0383)	0.9222 (±0.0560)	0.9259 (±0.0389)	0.9352 (±0.0370)	0.9123 (±0.0263)	0.9270 (±0.0392)	0.9212 (±0.0576)	0.9255 (±0.0391)	0.9349 (±0.0373)
20	0.9222 (±0.0323)	0.9167 (±0.0563)	0.9333 (±0.0428)	0.9315 (±0.0387)	0.9389 (±0.0366)	0.9218 (±0.0326)	0.9151 (±0.0591)	0.9327 (±0.0438)	0.9312 (±0.0389)	0.9386 (±0.0368)
30	0.9148 (±0.0261)	0.9222 (±0.0379)	0.9167 (±0.0475)	0.9352 (±0.0291)	0.9500 (±0.0286)	0.9142 (±0.0270)	0.9214 (±0.0390)	0.9155 (±0.0489)	0.9348 (±0.0295)	0.9498 (±0.0288)
50	0.9361 (±0.0264)	0.9259 (±0.0465)	0.9407 (±0.0335)	0.9426 (±0.0276)	0.9444 (±0.0379)	0.9358 (±0.0266)	0.9249 (±0.0484)	0.9403 (±0.0340)	0.9423 (±0.0278)	0.9441 (±0.0382)

Batch size had a relatively low effect on the accuracy and F1-score for each model, with 16 being the optimal batch size; as the batch sizes increased, the performance metrics generally decreased (as seen in Table 5). For example, ResNet50's accuracy declined as the batch sizes increased, dropping from 0.9489 to 0.9355 to 0.9300 (-1.99% overall). The F1 score also displayed a similar pattern, decreasing from 0.9487 to 0.9351 to 0.9296 (-2.01% overall). With a batch size of 16, VGG16 had the lowest STD for accuracy and F1 score, at 0.0228 and

0.0231. MobileNetV2 showed the highest STD with a batch size of 64, with 0.0559 for accuracy and 0.0590 for F1-score.

Table 5. Accuracy and F-1 score across the models while modifying the batch size

Batch Size	Accuracy					F-1 Score				
	VGG16	Mobile Net V2	Inception V3	Efficient Net B0	ResNet 50	VGG16	Mobile Net V2	Inception V3	Efficient Net B0	ResNet 50
16	0.9298 (±0.0228)	0.9255 (±0.0347)	0.9356 (±0.0338)	0.9311 (±0.0327)	0.9489 (±0.0325)	0.9294 (±0.0231)	0.9248 (±0.0357)	0.9351 (±0.0344)	0.9307 (±0.0330)	0.9487 (±0.0326)
32	0.9143 (±0.0389)	0.9289 (±0.0444)	0.9244 (±0.0454)	0.9356 (±0.0327)	0.9355 (±0.0371)	0.9136 (±0.0396)	0.9281 (±0.0457)	0.9236 (±0.0466)	0.9352 (±0.0329)	0.9351 (±0.0374)
64	0.9155 (±0.0279)	0.9178 (±0.0559)	0.9256 (±0.0552)	0.9267 (±0.0386)	0.9300 (±0.0354)	0.9151 (±0.0281)	0.9162 (±0.0590)	0.9247 (±0.0564)	0.9263 (±0.0388)	0.9296 (±0.0356)

When data augmentation was implemented, it typically yielded higher performance metrics than non-augmented data across all models. As seen in Table 6, Inception V3 showed the most significant improvement: its accuracy increased from 0.8978 to 0.9593 (+6.85%). EfficientNetB0, on the other hand, demonstrated a slightly higher accuracy and F1-score for non-augmented data: 0.9341 against 0.9281 (-0.64%) and 0.9339 against 0.9276 (-0.67%). The lowest STD can be seen in InceptionV3 when it utilized augmentation, having an STD of 0.0173 for accuracy and 0.0174 for the F1-score. MobileNetV2 displayed the highest STD on non-augmented data for accuracy and F1-score of 0.0552 and 0.0578.

Table 6. Accuracy and F-1 score across the models while modifying augmentation

Augmented?	Accuracy					F-1 Score				
	VGG16	Mobile Net V2	Inception V3	Efficient Net B0	ResNet 50	VGG16	Mobile Net V2	Inception V3	Efficient Net B0	ResNet 50
True	0.9222 (±0.0282)	0.9422 (±0.0218)	0.9593 (±0.0173)	0.9281 (±0.0275)	0.9481 (±0.0412)	0.9215 (±0.0289)	0.9419 (±0.0221)	0.9591 (±0.0174)	0.9276 (±0.0279)	0.9477 (±0.0416)
False	0.9175 (±0.0339)	0.9059 (±0.0552)	0.8978 (±0.0440)	0.9341 (±0.0405)	0.9281 (±0.0256)	0.9171 (±0.0341)	0.9041 (±0.0578)	0.8964 (±0.0453)	0.9339 (±0.0406)	0.9279 (±0.0257)

From this preliminary analysis, the most impactful hyperparameter was the learning rate when it increased from 0.0001 to 0.001. The least impactful hyperparameter was batch size, showing similar variations across all the batch sizes of 16, 32, and 64. Data augmentation and greater learning rates achieved the lowest STDs of just below 0.02. Lower epochs, non-augmented data, and a greater batch size produced the highest STDs of above 0.05.

As the next step to do a more quantitative ranking of the hyperparameters, we utilized RF and XGB to rank the importance of the hyperparameters in terms of accuracy. By incorporating the model as another variable, starting from most important, the RF ranking was model type, target epochs, learning rate, batch size, and augmentation, with an R^2 score of 0.8989 (see Table 7). For XGB the output ranking was learning rate, augmentation, model, target epochs, and batch size, with an R^2 of 0.8932.

Table 7. Ranking of hyperparameter importance by accuracy, including model as a variable

Feature	RF Ranking	RF Coefficient	XGB Ranking	XGB Coefficient
Model	1	0.2631	3	0.1989
Learning Rate	3	0.2221	1	0.3149
Target Epochs	2	0.2376	4	0.1268
Batch Size	4	0.1675	5	0.1145
Augmentation	5	0.1096	2	0.2448

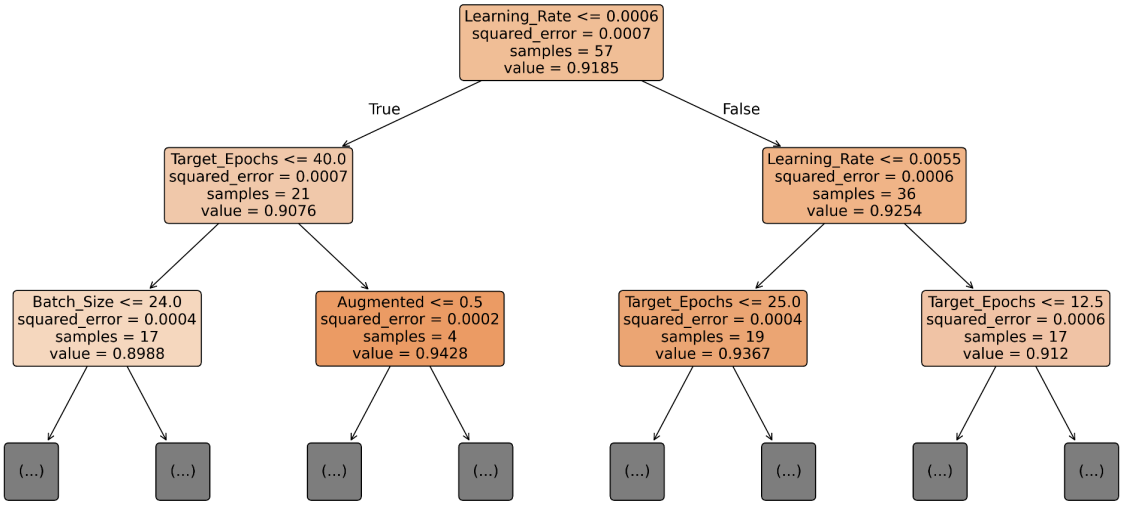
As detailed in Table 8, feature importance rankings varied across architectures. Across all models, XGB consistently outperformed RF in predictive fit, yielding much higher R^2 scores (0.97 to 0.99 versus 0.85 to 0.93). InceptionV3 stood out as an exception where both algorithms generated identical feature rankings, whereas other models exhibited distinct shifts between algorithms. Overall, learning rate and target epochs contributed to the highest importance ranks across the models. Notably, the RF hyperparameter importance rankings produced remained completely consistent across models when we conducted a ranking of hyperparameter importance by F1-score (rather than accuracy).

Table 8. Ranking of hyperparameter importance by accuracy within each model

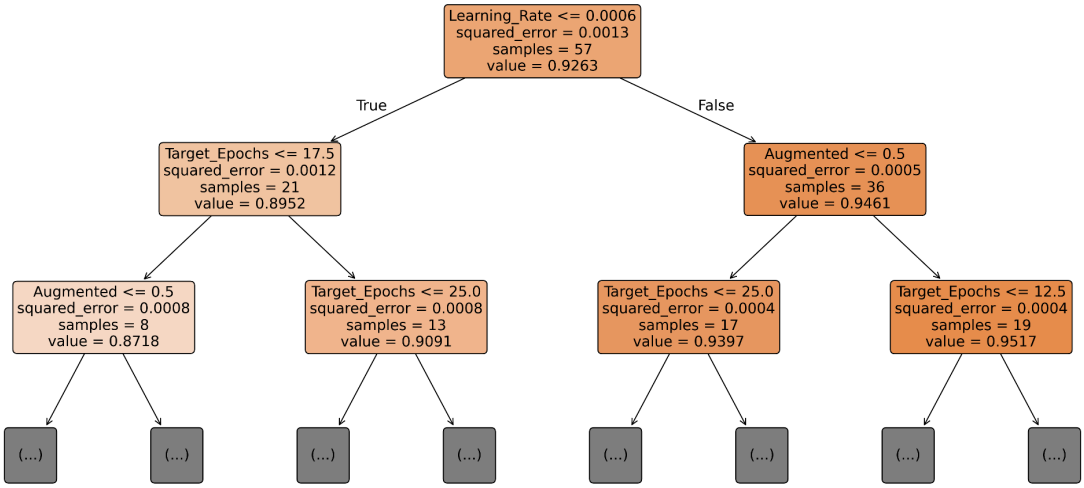
Model	Feature	RF Ranking	RF Coefficient	XGB Ranking	XGB Coefficient
MobileNetV2	Augmentation	4	0.1367	3	0.1088
	Learning Rate	1	0.4692	1	0.6636
	Target Epochs	2	0.2423	2	0.1270
	Batch Size	3	0.1519	4	0.1006
VGG16	Augmentation	1	0.4744	1	0.9280
	Learning Rate	3	0.1734	2	0.0284

	Target Epochs	2	0.2139	3	0.0224
	Batch Size	4	0.1384	4	0.0212
InceptionV3	Augmentation	1	0.3140	1	0.5390
	Learning Rate	2	0.3081	2	0.3318
	Target Epochs	3	0.2213	3	0.0753
	Batch Size	4	0.1567	4	0.0539
EfficientNetB0	Augmentation	4	0.1503	2	0.2225
	Learning Rate	1	0.3230	1	0.4664
	Target Epochs	2	0.3021	4	0.1333
	Batch Size	3	0.2245	3	0.1778
ResNet50	Augmentation	4	0.1567	4	0.1900
	Learning Rate	3	0.2230	2	0.2642
	Target Epochs	1	0.3830	1	0.3502
	Batch Size	2	0.2372	3	0.1957

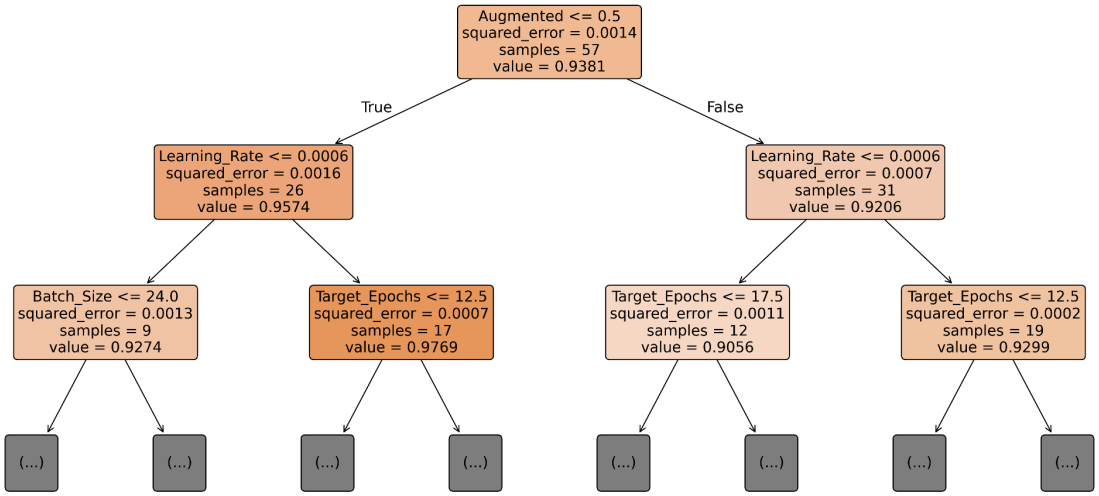
To develop interpretable guidelines for model design purposes, we extracted Decision Trees for each model to summarize the information from the RF accuracy rankings, including a max depth of 2 to avoid complex images (see Figure 3).



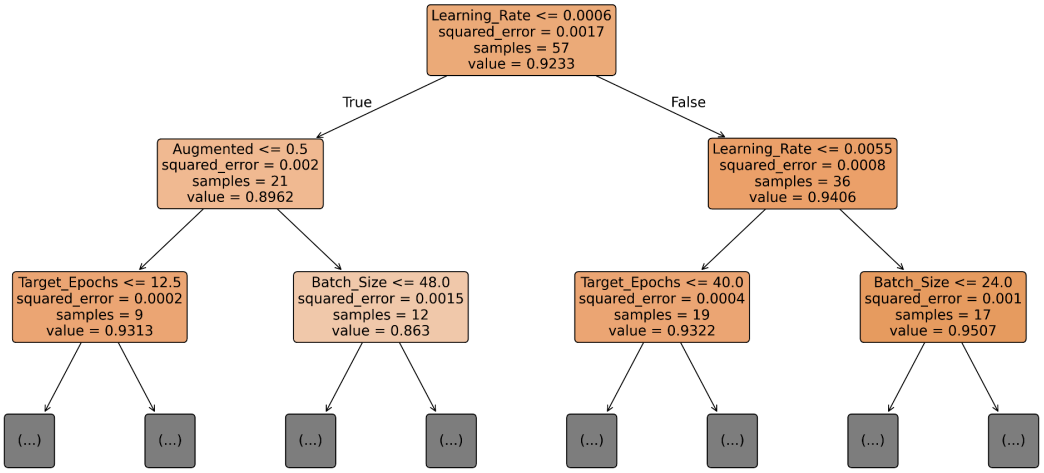
a. ResNet50



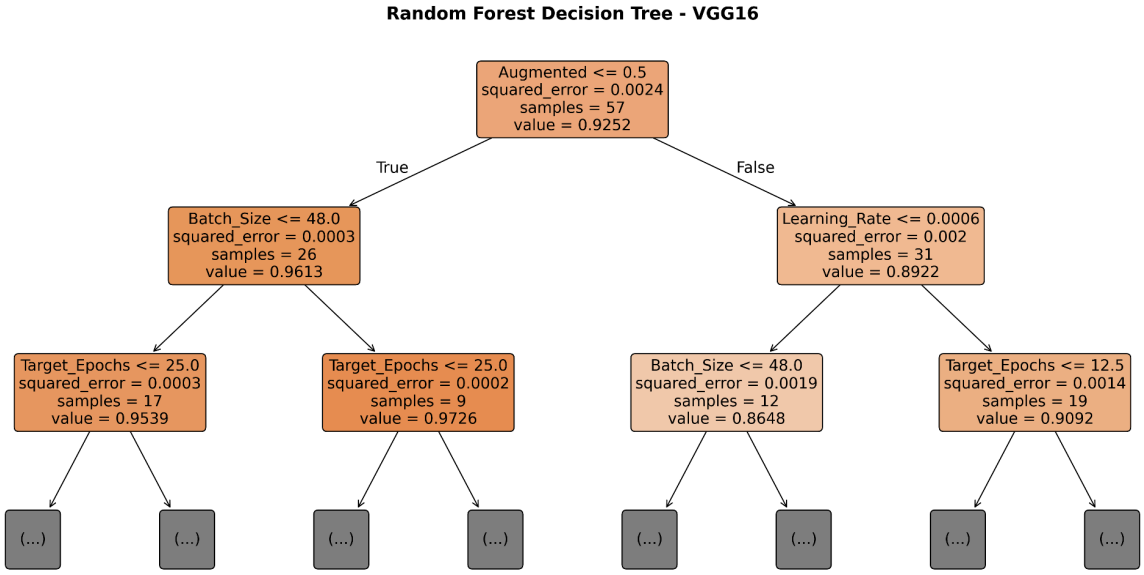
b. MobileNetV2



c. EfficientNetB0



d. InceptionV3



e. VGG16

Fig 3. RandomForest decision trees for each model

Discussion

In this study, we implemented a Kaggle dataset of chest X-ray images for pneumonia

detection. We utilized five pre-trained models that are often used for image classification: MobileNetV2, VGG16, InceptionV3, EfficientNetB0, and ResNet50. To evaluate how hyperparameter optimization affected the accuracy and F1-score of the models' classification, we chose to investigate the effects of learning rate, batch size, number of epochs, and augmentation, being widespread hyperparameters. After tabling the data, we produced a ranking of the hyperparameter importance across and within the models to see if this aligned with our observations from the data.

Some limitations in the previous studies include that the models implemented require a significant amount of time to train due to the complexities of the model and involve random computations for superior accuracy. In the publication by Alshanketi et al., it was noted that the learners sometimes produced some miscategorized samples [8]. Furthermore, some restrictions of the paper by Xu and Zhang include its reliance on pre-trained models, which can possibly affect the accuracy; gaps in addressing unbalanced data; and the use of clean, organized datasets, which can contrast with medical images in the real world that are often unrefined and messy. One setback in Xu and Zhang's study is that its dependence on large models leads to more resources being depleted. Their models include a few limitations: they have not been adapted in healthcare scenarios, some have poor data diversity, overlook synthetic data diversity, have a slow inference speed, or rely excessively on the quality of original images [13]. Other publications have also come to a similar conclusion of learning rate being the most impactful hyperparameter. Wang et al. determined that learning rate, among other hyperparameters, is one of the most influential in DL CNN models on image datasets [17]. Bengio also claims that the learning rate is the most significant hyperparameter, since an unsuitable learning rate can lead to divergence or underfitting [18].

In this paper, we only used imaging data, rather than training the models on both imaging and clinical data, which can lead to higher accuracy because of the greater variety the models are exposed to. The dataset is balanced, which means we don't know how the models would have performed on skewed data. We also implemented five pre-trained models, which all have certain limitations, instead of creating a new model with parts of already existing models. As previously mentioned, we selected a limited amount of hyperparameters to experiment with out of a large number of other parameters we didn't consider. Being the most popular and relevant hyperparameters, we looked at these specific ones for fine-tuning, but there are additional hyperparameters we didn't look into. Finally, we also implemented some models, such as VGG-16, that had a significantly longer training time than the other models.

Future work for this study would include running the same models and altering the same hyperparameters on a less balanced dataset to see if the same results would be yielded. We may also process other clinical data, such as clinical audio or handwritten doctor notes, besides images, to introduce more variety of data to the models and therefore produce stronger results.

To prevent the excessive run time from some of the models implemented, we may look into developing ensembles.

References

[1] Centers for Disease Control and Prevention. “FastStats: Pneumonia.” *National Center for Health Statistics*, cdc.gov/nchs/fastats/pneumonia.htm.

[2] Kheirdoust, Azam, et al. “Evaluation of Machine Learning Methods Developed for Prediction and Diagnosis of Pneumonia: A Systematic Review.” *Health Science Reports*, vol. 8, no. 12, 30 Nov. 2025, article e71446, doi.org/10.1002/hsr2.71446.

[3] UNICEF. “Pneumonia.” *UNICEF Data: Monitoring the Situation of Children and Women*, data.unicef.org/topic/child-health/pneumonia/.

[4] Reed, Shawn D., et al. “Attributable Cost of Adult Hospitalized Pneumonia Beyond the Acute Episode.” *Pharmacoeconomics*, vol. 39, no. 3, 2021, pp. 325–337. *PubMed Central*, pmc.ncbi.nlm.nih.gov/articles/PMC8160038/.

[5] Izadnegahdar, R., et al. “Childhood Pneumonia in Developing Countries.” *The Lancet Respiratory Medicine*, vol. 1, 2013, pp. 574–584, thelancet.com/journals/lanres/article/PIIS2213-2600(13)70075-4/abstract.

[6] Popovsky, Erica Y., and Todd A. Florin. “Community-Acquired Pneumonia in Childhood.” *Encyclopedia of Respiratory Medicine*, edited by Sam M. Janes, 2nd ed., Academic Press, 2022, pp. 119–131, doi.org/10.1016/B978-0-08-102723-3.00013-5.

[7] Ullah, Nadir, et al. “Diagnostic Approach to Pneumonia in Immunocompromised Hosts.” *Journal of Clinical Medicine*, vol. 14, no. 2, 2025, article 389, doi.org/10.3390/jcm14020389.

[8] Alshanketi, Faisal, et al. “Pneumonia Detection from Chest X-Ray Images Using Deep Learning and Transfer Learning for Imbalanced Datasets.” *Journal of Imaging Informatics in Medicine*, vol. 38, no. 4, 2025, pp. 2021–2040, doi.org/10.1007/s10278-024-01334-0.

[9] Kundu, Rohit, et al. “Pneumonia Detection in Chest X-Ray Images Using an Ensemble of Deep Learning Models.” *PLOS ONE*, vol. 16, no. 9, 7 Sept. 2021, article e0256630, doi.org/10.1371/journal.pone.0256630.

[10] Effah, Clement Yaw, et al. “Machine Learning-Assisted Prediction of Pneumonia Based on Non-Invasive Measures.” *Frontiers in Public Health*, vol. 10, 28 July 2022, article 938801, doi.org/10.3389/fpubh.2022.938801.

[11] Han, Yan, et al. “Pneumonia Detection on Chest X-Ray Using Radiomic Features and Contrastive Learning.” *Proceedings of the IEEE International Symposium on Biomedical Imaging*, 2021, pp. 247–251, doi.org/10.1109/isbi48211.2021.9433853.

[12] Shilpa, N., et al. “Revolutionizing Pneumonia Diagnosis: AI-Driven Deep Learning Framework for Automated Detection from Chest X-Rays.” *IEEE Access*, 15 Nov. 2024, doi.org/10.1109/ACCESS.2024.3498944.

[13] Xu, Zhijing, and Haoyang Zhang. “Pneumonia Detection from Enhanced Chest X-Ray Images Based on Double SGAN Model.” *Scientific Reports*, vol. 16, no. 1, 19 Feb. 2026, article 9922, doi.org/10.1038/s41598-026-39785-w.

[14] Slimi, Houmem, et al. “Trustworthy Pneumonia Detection in Chest X-Ray Imaging Through Attention-Guided Deep Learning.” *Scientific Reports*, vol. 15, 2025, article 40029, doi.org/10.1038/s41598-025-23664-x.

[15] Bhattarai, Pankaj, et al. “Detection of Pneumonia Using Machine Learning.” *Proceedings of the 5th International Conference on Information Management & Machine Intelligence (ICIMMI ’23)*, article 159, ACM, 13 May 2024, pp. 1–11, doi.org/10.1145/3647444.3652464.

[16] Murtaza, Yusuf. “Chest X-Ray Pneumonia Balanced Dataset.” *Kaggle*, kaggle.com/datasets/yusufmurtaza01/chest-xray-pneumonia-balanced-dataset .

[17] Wang, Ruinan, et al. “Efficient Hyperparameter Importance Assessment for CNNs.” *arXiv*, 11 Oct. 2024, doi.org/10.48550/arXiv.2410.08920.

[18] Bengio, Yoshua. “Practical Recommendations for Gradient-Based Training of Deep Architectures.” *arXiv*, 2012, arxiv.org/pdf/1206.5533.