ZURICH UNIVERSITY OF APPLIED SCIENCES
DEPARTMENT LIFE SCIENCES AND FACILITY MANAGEMENT
INSTITUTE FOR COMPUTATIONAL LIFE SCIENCE

Assignment 1 - DSHEALTH

Detection of Pneumonia with the Kaggle Dataset: Pediatric Pneumonia Chest X-ray

by
Meggie Krymowski, Alexandra Schibli
BSc Applied Digital Life Sciences 2022

Corrector:
Norman Juchler

Contents

1 Introduction

Pneumonia, a severe respiratory infection caused by bacteria, viruses, or fungi, inflames lung air sacs, affecting approximately 450 million people annually and causing 4 million deaths, with 740,180 children under 5 dying in 2019, per the World Health Organization (WHO) (WHO, 2022). Symptoms like fever, cough with phlegm, extreme fatigue and breathing difficulty often start mildly but can worsen rapidly, especially in vulnerable groups such as the elderly, infants and those with weakened immune systems, though anyone can be affected. In Switzerland, pneumonia affects 42,000 people yearly, with a hospital mortality rate of 7.3% xxxxxxx(Source: Swiss Federal Office of Public Health). The infection obstructs lung structures with pus or fluid, impairing oxygen intake and is typically treated with antibiotics and symptomatic relief, though recovery may take days. Traditional diagnosis via chest X-rays and radiologist review is time-consuming and error-prone, particularly in resource-limited settings with scarce skilled professionals. Recent advancements in medical imaging and artificial intelligence (AI), particularly Convolutional Neural Networks (CNNs), have enabled automated pneumonia detection with high accuracy, often outperforming radiologists in controlled settings (Kermany et al., 2018). However, real-world applicability across diverse populations and imaging conditions remains underexplored and standardized image preprocessing techniques to enhance model performance are lacking. Early detection is critical to prevent complications like sepsis or acute respiratory distress syndrome (ARDS). This project aims to find the best-fitting CNN model for classifying pneumonia and normal chest X-rays from a given dataset, using data augmentation for diversity and cross-validation for reliability, optimizing accuracy, sensitivity and specificity.

2 Materials

The main dataset used in this project was the Pediatric Chest X-ray dataset by Kermany et al., available on Kaggle Kaggle, 2020. It contains a total of 5,856 chest X-ray images of pediatric patients, each labeled as either “normal” or “pneumonia.” The dataset is unbalanced: around 1,583 images show normal lungs, while about 4,273 images show pneumonia. This imbalance reflects real-world conditions, where X-rays are more often taken when pneumonia is suspected.

To evaluate model generalization, a second external dataset was used for validation. This dataset, also sourced from Kaggle, contains 180 X-ray images (90 normal and 90 with pneumonia) and was used exclusively for final testing after training.

3 Methodology

3.1 Data Exploration

The project began with an exploration of the structure and content of the Pediatric Chest X-ray dataset. The dataset contains clear and well-labeled images. A pronounced class imbalance was observed: approximately 1,500 images were labeled as Normal, while over 4,300 were labeled as Pneumonia, resulting in a ratio of roughly 1:2.9. Such imbalances are common in clinical datasets, as X-rays are more frequently taken when disease is suspected. A visual inspection of the images revealed noticeable differences between healthy and infected lungs, particularly in terms of texture and brightness. These distinctions indicated that deep learning models could potentially learn to differentiate between the two classes based on such features.

3.2 Preprocessing and Augmentation

To prepare the images for training and testing, two transformation pipelines were implemented using PyTorch's `transforms.Compose` function. The training pipeline included several data augmentation steps to improve model generalization. For testing and validation, only basic preprocessing was applied to ensure consistency across the data.
The training transformations included:

- `Grayscale(num_output_channels=1)`: Converts the image to single-channel grayscale, reducing input dimensionality from three channels (RGB) to one, as X-rays are inherently grayscale.
- `Resize((128, 128))`: Resizes the image to 128×128 pixels to standardize input size, reduce memory usage and speed up training.
- `RandomHorizontalFlip(p=0.5)`: Flips the image horizontally with a 50% probability, mimicking the natural left/right symmetry in X-rays.
- `RandomVerticalFlip(p=0.1)`: Flips the image vertically with a 10% probability, simulating rare orientation variations.
- `RandomRotation(15)`: Rotates the image by a random angle between -15 and +15 degrees, accounting for slight differences in patient positioning.
- `RandomAffine(degrees=0, translate=(0.05, 0.05), scale=(0.95, 1.05))`: Applies small translations (5% shift) and scaling (5% zoom in/out), simulating minor positional variations in X-ray imaging.
- `ToTensor()`: Converts the image to a PyTorch tensor with shape (C, H, W), where C=1, H=128 and W=128.
- `Normalize(mean=[0.5], std=[0.5])`: Normalizes pixel values to the range [-1, 1], ensuring consistent input distributions for training stability.

These transformations were applied automatically while loading the data, using the `ImageFolder` class. This method is efficient because it does not store extra images but instead changes them each time during training. It also helps the model learn from more diverse examples, especially from the minority class (normal).

3.3 Handling Class Imbalance

Due to the significantly higher number of pneumonia cases compared to normal cases in the dataset, it was necessary to address the class imbalance. Without correction, the model would likely favour predicting the majority class (pneumonia) while under-representing the minority class. To mitigate this issue, class weights were calculated based on the number of samples in each class. The formula used was:

$$\text{class\_weight}[i] = \frac{N}{C \cdot n_i}$$

Here, N denotes the total number of training samples, C represents the number of classes (2 in this case) and n_i refers to the number of samples in each class.
Based on the class weights, a `pos_weight` value was calculated for use with PyTorch's `BCEWithLogitsLoss` function. This parameter increases the influence of the minority class (Normal) during training. The formula used was:

$$\text{pos\_weight} = \frac{\text{class\_weight}_{\text{Normal}}}{\text{class\_weight}_{\text{Pneumonia}}}$$

This means that errors in detecting pneumonia (false negatives) are punished more strongly during training. This is especially important in medical use cases, where missing a pneumonia case can be dangerous for the patient.

4 Model Building

To ensure a fair evaluation of the models, two different data splitting strategies were applied. The first approach divided the data set into 60% for training, 20% for validation during training and 20% for testing. In the second setup, the same internal split was used, but an additional external dataset containing 180 images (90 normal and 90 pneumonia) was included to assess the models' generalization on unseen data. This external dataset originated from a different Kaggle source and was not involved in the training process.

4.1 Model Architectures

Two deep learning models were evaluated for the task of classifying chest X-rays as either normal or pneumonia: a custom-built convolutional network referred to as FourLayerCNN and a modified ResNet18 model utilizing transfer learning.

4.1.1 FourLayerCNN

The FourLayerCNN was designed especially for grayscale X-ray images of size 128×128. It contains four convolutional blocks. Each block includes a convolutional layer, followed by batch normalization, a ReLU activation and a max-pooling operation. These blocks help the model extract important visual features while keeping the model small and fast. After the convolutional part, the output is flattened and passed through a fully connected layer with 128 units. A dropout layer is used here to reduce overfitting by randomly turning off some neurons during training. Finally, a single output neuron produces a prediction score that indicates whether the input image shows signs of pneumonia. Table 1 provides a detailed overview of the architecture. It lists the layers in order, the size of their outputs and the number of trainable parameters. As shown, the model has approximately 2.49 million trainable parameters. This makes it relatively lightweight and well-suited for small to medium-sized medical datasets like ours.

Table 1: FourLayerCNN Architecture

Layer	**Output Shape**	**Number of Parameters**
Input	(None, 128, 128, 1)	0
Conv2D (32 filters, 3×3)	(None, 128, 128, 32)	320
MaxPooling2D (2×2)	(None, 64, 64, 32)	0
Conv2D (64 filters, 3×3)	(None, 64, 64, 64)	18,496
MaxPooling2D (2×2)	(None, 32, 32, 64)	0
Conv2D (128 filters, 3×3)	(None, 32, 32, 128)	73,856
MaxPooling2D (2×2)	(None, 16, 16, 128)	0
Conv2D (256 filters, 3×3)	(None, 16, 16, 256)	295,168
MaxPooling2D (2×2)	(None, 8, 8, 256)	0
Dropout (rate = 0.25)	(None, 8, 8, 256)	0
Flatten	(None, 16384)	0
Dense (128 units)	(None, 128)	2,097,280
Dense (1 unit)	(None, 1)	129
Total parameters	–	**2,485,249**
Trainable parameters	–	**2,485,249**
Non-trainable parameters	–	**0**

4.1.2 ResNet18

The second model is based on ResNet18, a well-known deep neural network originally trained on natural color images (ImageNet). The first convolutional layer was modified to accept grayscale images instead of RGB. All original layers were frozen to preserve the model's pre-trained feature extraction capabilities. Only the final fully connected layer was replaced and trained on the X-ray dataset. This approach enables the use of a powerful pretrained model with minimal training effort.

4.2 Model Training and Evaluation

For training, the `BCEWithLogitsLoss` loss function was used along with the calculated `pos_weight` to address the class imbalance between normal and pneumonia cases. This ensured that the minority class (normal) had a stronger influence on the loss, helping the model avoid bias towards the majority class.

Both the FourLayerCNN and the modified ResNet18 were trained using the Adam optimizer with a learning rate of 0.001 and a batch size of 32. The models were trained for a maximum of 20 epochs. To avoid overfitting, early stopping with a patience of 4 epochs was applied: if the validation loss did not improve for 4 consecutive epochs, the training process was stopped early.

Where available, mixed precision training via `torch.cuda.amp` was used, reducing memory usage and improving training speed on compatible GPUs.

During training, loss and accuracy were tracked on both the training and validation sets. After training, the final models were evaluated on two datasets: the internal test set (from the original 60/20/20 split) and an external validation set containing 180 chest X-ray images. Evaluation metrics included accuracy, precision, recall and F1-score. Predictions were based on sigmoid outputs, using a threshold of 0.5 to distinguish between pneumonia and normal cases.

To better understand model behavior, confusion matrices and ROC curves were generated for both models and both datasets. These visual tools enabled the analysis of error types (false positives and false negatives) and provided insight into how well the models could distinguish between the two classes across different thresholds.

5 Results

To evaluate classification performance, two deep learning models were compared: the custom-built FourLayerCNN and a modified ResNet18 using transfer learning. Both models were adapted for grayscale chest X-ray images and trained on the same dataset. Two evaluation setups were used: an internal split (60% training, 20% validation, 20% test) and an additional test on an external dataset with 180 labeled images to assess generalizability.

5.1 Training and Validation Performance

As shown in Figure 1, the FourLayerCNN achieved stable and fast convergence. It reached high accuracy after only a few epochs and showed consistent validation performance. The final training accuracy was 95.64% and the best validation accuracy reached 95.47%. The validation loss steadily decreased and there was only a small gap between training and validation metrics, indicating low overfitting.

The ResNet18 model showed slower convergence and more fluctuations in both loss and accuracy. As seen in Figure 1, it achieved a final validation accuracy of 86.85%. The learning curve

was less stable and the gap between training and validation performance was larger, which suggests that ResNet18 had more difficulty generalizing to unseen data.

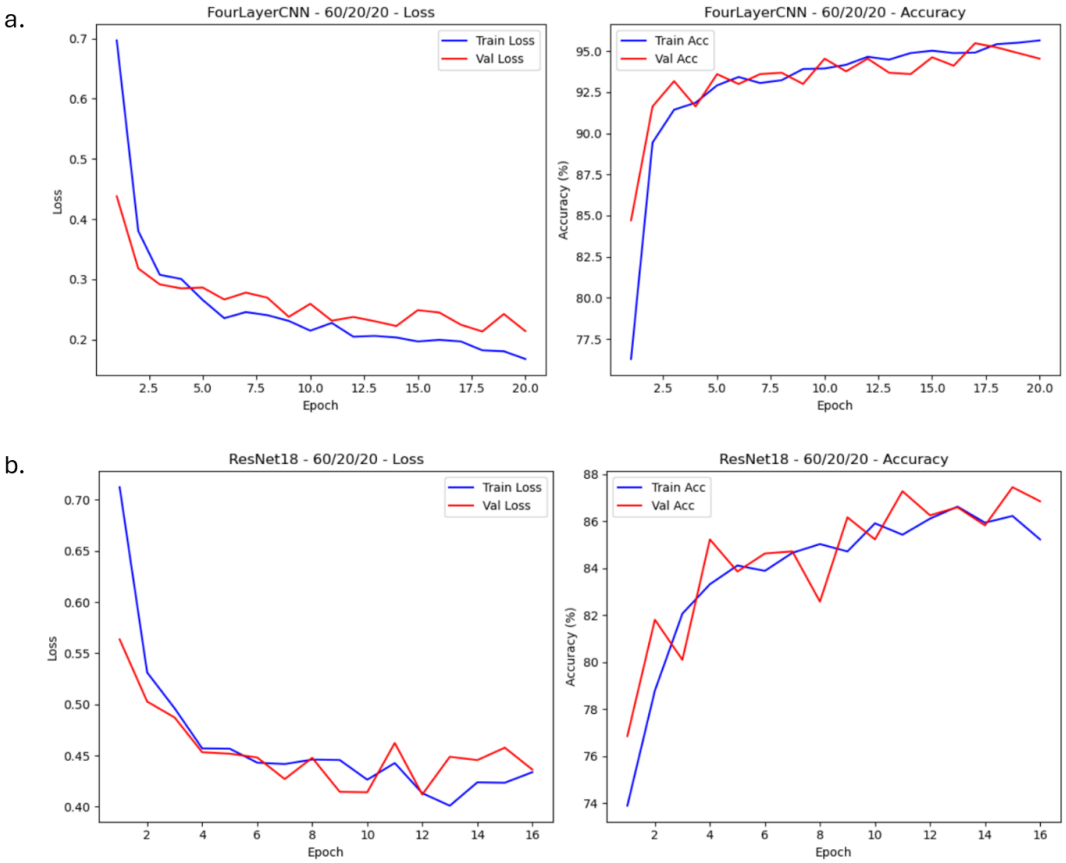

Figure 1: Training and validation loss and accuracy curves for FourLayerCNN (a.) and ResNet18 (b.) on the 60/20/20 split.

5.2 Classification Metrics

Table 2 shows the final evaluation metrics for both models on the internal and external datasets. The FourLayerCNN outperformed ResNet18 in every metric. On the internal test set, it reached 94.45% accuracy and a high recall of 98.57%, resulting in an F1-score of 96.24%. On the external dataset, it still achieved 91.11% accuracy and an F1-score of 91.49%, indicating strong generalization.

In contrast, ResNet18 achieved 87.97% accuracy and 92.03% F1-score on the internal test set, but dropped to 79.44% accuracy and 77.02% F1-score on the external set. This drop highlights weaker robustness compared to FourLayerCNN.

Table 2: Evaluation metrics for FourLayerCNN and ResNet18 across internal and external splits.

Model	Accuracy	Precision	Recall	F1-score	Split
FourLayerCNN	94.45%	94.00%	98.58%	96.24%	60/20/20
FourLayerCNN	91.11%	87.76%	95.56%	91.49%	external
ResNet18	87.97%	87.82%	96.68%	92.04%	60/20/20
ResNet18	79.44%	87.32%	68.89%	77.02%	external

5.3 ROC Curves and AUC Scores

The ROC curves in Figure 2 illustrate model performance in terms of classification thresholds. FourLayerCNN achieved excellent class separation with an Area Under Curve (AUC) of 0.98 for both internal and external data. ResNet18 performed slightly worse, achieving an AUC of 0.95 internally and 0.88 externally. This confirms the superior discriminative ability of the custom CNN across both seen and unseen datasets.

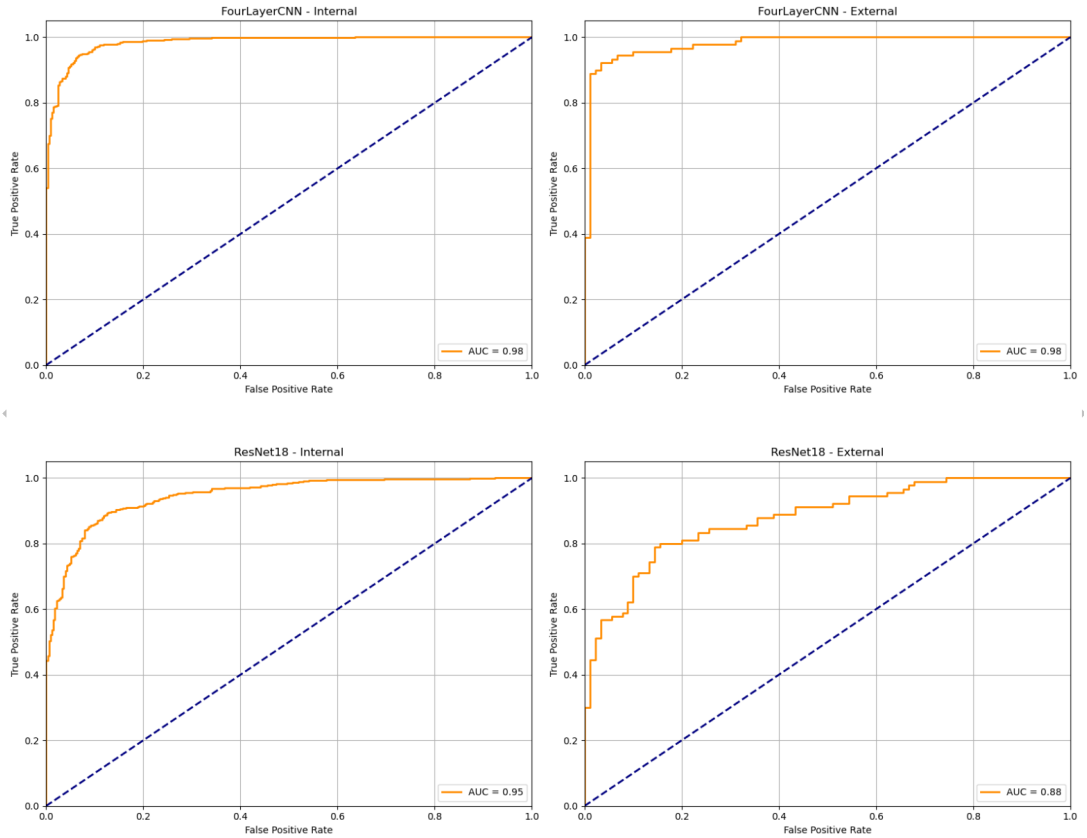

Figure 2: ROC curves showing classification performance for both models. **Top:** FourLayerCNN, AUC = 0.98 (left: internal, right: external). **Bottom:** ResNet18, AUC = 0.95 (internal) and 0.88 (external).

5.4 Confusion Matrices

Figure 3 shows the confusion matrices for both models on the internal 60/20/20 split. FourLayerCNN made fewer false positive and false negative predictions, achieving more balanced classification results. ResNet18 showed a higher tendency to misclassify normal cases as pneumonia, resulting in lower precision and accuracy.

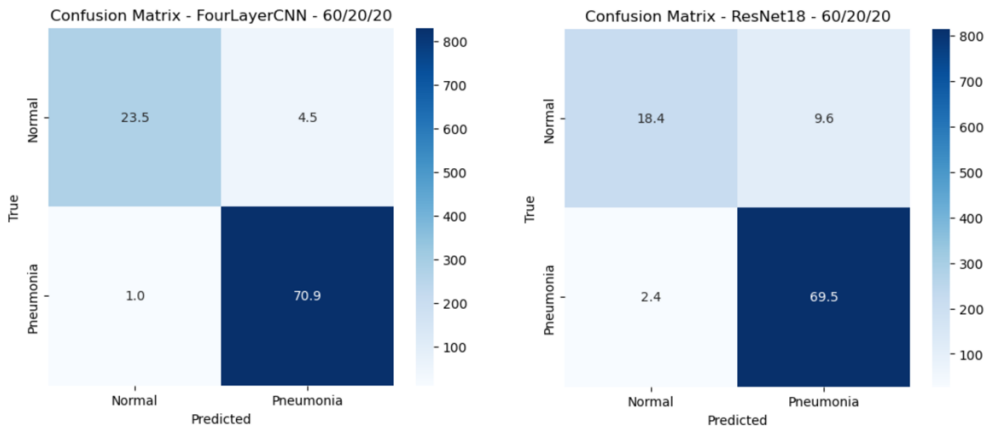

Figure 3: Confusion matrices for internal evaluation (60/20/20 split). **Left:** FourLayerCNN. **Right:** ResNet18. FourLayerCNN demonstrates fewer misclassifications and more balanced predictions.

5.5 Overall Comparison

Figure 4 summarizes model performance in terms of accuracy and F1-score. FourLayerCNN consistently outperformed ResNet18 across all evaluation splits, particularly in external testing, which is crucial for real-world deployment. While both models performed well on the internal set, FourLayerCNN proved to be more robust and reliable on unseen data.

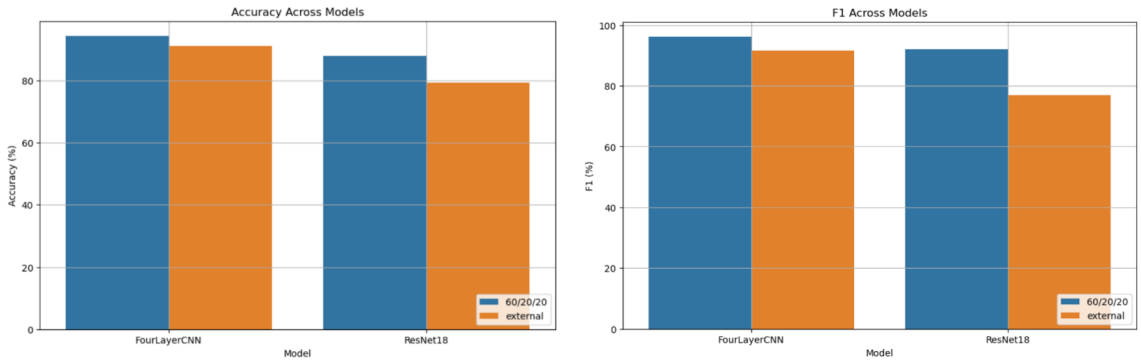

Figure 4: Overall performance comparison. **Left:** Accuracy of both models on internal and external datasets. **Right:** F1-scores show FourLayerCNN performs better, especially in generalization.

6 Discussion

This project demonstrated that the custom-built FourLayerCNN consistently outperformed the pre-trained ResNet18, especially when tested on external validation data. This is a key advantage for real-world clinical applications, where models must generalize well to unseen data.

The simpler architecture of the FourLayerCNN likely contributed to its robustness. With fewer parameters and the use of Dropout (but no Batch Normalization), the model was less prone to overfitting. Real-time data augmentation and a weighted loss function (`pos_weight` in `BCEWithLogitsLoss`) also helped the model learn effectively despite the class imbalance.

In contrast, ResNet18 showed more unstable training behavior and lower performance on external data. Although the input layer was modified to accept grayscale images, the model was originally trained on natural RGB images (ImageNet). This domain mismatch likely limited its ability to extract useful features from X-rays. As a result, ResNet18 tended to overpredict pneumonia and showed a higher false-positive rate.

These findings are also reflected in the confusion matrices and evaluation metrics. The FourLayerCNN performed more consistently and made fewer classification mistakes, especially for the Normal class. It achieved higher accuracy and F1-scores across all datasets and splits.

Table 2 in the Results section summarizes the evaluation metrics. It shows that FourLayerCNN achieved better scores across almost all key indicators, particularly on the external dataset, where generalization matters most.

It should also be noted that the codebase was improved after the project presentation. These changes enhanced structure, visualizations and evaluation, which improved model interpretation.

Another open point was the unclear origin of the X-ray data. While metadata like patient age was not available, it is likely that most images in both datasets show pediatric cases, which is consistent with the disease's high incidence in young children.

Despite strong results, some cases remained difficult to classify, likely due to visual similarity or labeling noise. The class imbalance, although addressed, may also still impact predictions.

Despite these promising results, there are still challenges. Some X-ray images are difficult to classify due to visual similarities, unclear boundaries, or possible labeling errors. In addition, the dataset remains unbalanced, with more pneumonia cases than normal ones. Although this was addressed during training, the imbalance may still affect the models' behavior in borderline cases.

To improve future models, several strategies could be useful:

- Collecting more and better-balanced data to support more accurate learning.
- Fine-tuning transfer learning models such as ResNet18 instead of freezing all layers, or testing newer architectures like EfficientNet or Vision Transformers (ViTs).

- Using model explainability techniques such as Grad-CAM to visualize which areas of the image were important for the decision.
- Testing the models on additional external datasets from other hospitals or imaging devices to validate their robustness.

7 Conclusion

This project demonstrated that deep learning methods can be effectively applied to detect pneumonia in pediatric chest X-rays, even when working with relatively simple architectures. The custom-built FourLayerCNN consistently outperformed a modified ResNet18, especially in generalization to external datasets. Key success factors included targeted preprocessing, data augmentation and handling class imbalance through weighted loss functions

Even though the results were good, there are still some problems. For example, the data set had little additional information about the patients and some images may have been labeled incorrectly. In addition, the number of normal and pneumonia images was not balanced.

In the future, it may be beneficial to explore newer model architectures, obtain higher-quality and more balanced datasets and apply interpretability tools that highlight which regions of the image contribute to the model's decisions.

Finally, the results highlight the importance of domain-specific model design and comprehensive evaluation across multiple data sources, as well as the potential of tailored convolutional networks for medical imaging tasks.

References

Kaggle. (2020). 1. VGG19_pediatric_pneumonia_classification. Retrieved March 13, 2025, from https://kaggle.com/code/mohamedelsadek44/1-vgg19-pediatric-pneumonia-classification

Kermany, D. S., Goldbaum, M., Cai, W., Valentim, C. C. S., Liang, H., Baxter, S. L., McKeown, A., Yang, G., Wu, X., Yan, F., Dong, J., Prasadha, M. K., Pei, J., Ting, M. Y. L., Zhu, J., Li, C., Hewett, S., Dong, J., Ziyar, I., ... Zhang, K. (2018). Identifying Medical Diagnoses and Treatable Diseases by Image-Based Deep Learning. *Cell*, *172*(5), 1122–1131.e9. https://doi.org/10.1016/j.cell.2018.02.010

WHO. (2022). Pneumonia in children. Retrieved March 13, 2025, from https://www.who.int/news-room/fact-sheets/detail/pneumonia

List of Figures

List of Tables