CNN-Based Classification of Chest X-ray Images for COVID-Related Pneumonia Detection

Hung Liu
hkliu@cougarnet.uh.edu
University of Houston
Houston, Texas, USA

Jake Abeyta
jaabeyta@cougarnet.uh.edu
University of Houston
Houston, Texas, USA

Abstract

This paper presents a convolutional neural network (CNN) approach to classify chest X-ray images into pneumonia and normal categories, with the specific goal of detecting COVID-19-related pneumonia. Using the Coronahack Chest X-ray Dataset, we developed and trained multiple CNN models, systematically varying key architectural parameters, such as dense layer size and dropout rate, to evaluate their impacts on model performance and generalization. Our baseline CNN achieved a validation accuracy of approximately 96%. Through targeted experimentation, we found that reducing the dense layer size maintained high validation accuracy with fewer parameters, while increasing the dropout rate improved model generalization by mitigating overfitting. This study demonstrates the effectiveness of CNN-based models in medical image classification and highlights the importance of model tuning in achieving optimal performance and reliability.

CCS Concepts

• **Computing methodologies → Machine learning approaches**; **Computer vision problems**; • **Applied computing → Health informatics**.

Keywords

Convolutional Neural Networks; Pneumonia Detection; Chest X-ray Images; Deep Learning; COVID-19; Image Classification; Medical Imaging

1 Introduction

Chest X-rays have become a critical diagnostic tool for detecting respiratory illnesses, particularly during outbreaks such as COVID-19. Early and accurate diagnosis is essential for effective patient management and reducing disease spread. Traditionally, diagnosis relies heavily on expert radiologists who manually interpret X-ray images, which can be both time-consuming and subjective. Automated detection methods leveraging machine learning, especially Convolutional Neural Networks (CNNs), offer a powerful alternative to streamline diagnosis and improve reliability.

In this project, we employ deep learning techniques to classify chest X-ray images into two categories: pneumonia and normal. Using the publicly available CoronaHack Chest X-Ray Dataset from Kaggle, we trained and evaluated a CNN-based model capable of accurately distinguishing between these classes. To enhance the robustness of our approach, we conducted multiple experiments, adjusting key parameters such as the size of dense layers and dropout rates to analyze their effects on performance and model stability.

Our results demonstrate that CNN models can achieve high accuracy, generalize well to unseen data, and maintain performance even with variations in architecture. These findings not only underscore the potential of deep learning in medical image analysis but also highlight the importance of systematic experimentation in developing reliable diagnostic tools. The remainder of this paper details the dataset preparation, model architecture, experimental methods, results, and a comprehensive discussion of findings and implications.

2 Dataset

The dataset [2] used in this project is the *CoronaHack Chest X-Ray Dataset*, publicly available on Kaggle. It consists of a collection of labeled chest X-ray images sourced from various medical repositories. The original dataset includes over 5,800 images categorized into “Normal” and “Pneumonia” classes.

For this project, we focused on binary classification: distinguishing between “Normal” and “Pneumonia” cases. All other labels and images with missing or unmatched metadata were excluded. After preprocessing, a total of 5,910 images were retained—1,576 labeled as “Normal” and 4,334 labeled as “Pneumonia.”

Each image was converted to grayscale and resized to a fixed input shape of 128×128 pixels. Pixel values were normalized to the range [0, 1] to aid model convergence. The dataset was split into training and validation sets using an 80/20 ratio with stratification to preserve class distribution. This resulted in 4,728 training images and 1,182 validation images.

To support model experimentation and ablation studies, the dataset was kept consistent across all runs. No additional data augmentation techniques (e.g., flipping, rotation) were applied, in order to isolate the effects of architectural changes on model performance.

3 Methodology

In this project, we utilized a Convolutional Neural Network (CNN) architecture, implemented using TensorFlow and Keras, to perform binary classification on chest X-ray images. The core objective was to distinguish accurately between images labeled as “Normal” and those labeled as “Pneumonia.”

3.1 CNN Architecture

Our baseline CNN model consisted of two convolutional blocks, each including a convolutional layer followed by a max-pooling operation. Specifically, the model architecture was:

- **Convolutional Layer 1:** 32 filters, 3×3 kernel, ReLU activation.
- **MaxPooling Layer 1:** 2×2 pooling.
- **Convolutional Layer 2:** 64 filters, 3×3 kernel, ReLU activation.
- **MaxPooling Layer 2:** 2×2 pooling.

- **Flattening Layer:** converts 2D feature maps into a 1D vector.
- **Dense (Fully Connected) Layer:** 64 neurons with ReLU activation.
- **Dropout Layer:** dropout rate of 0.3 for regularization.
- **Output Layer:** single neuron with sigmoid activation for binary classification.

This architecture resulted in approximately 3.7 million trainable parameters.

3.2 Training Configuration

The model was trained using the Adam optimizer with a binary cross-entropy loss function, a learning rate of 0.001, and a batch size of 32. Each model was trained over 10 epochs, and training progress was evaluated using both accuracy and loss metrics on training and validation datasets.

3.3 Experimental Approach

To assess the effects of architecture variations on model performance and generalization, we conducted additional experiments by systematically altering two hyperparameters:

(1) **Dense Layer Size:** Reduced from 64 to 32 units to evaluate the impact on performance and overfitting.
(2) **Dropout Rate:** Increased from 0.3 to 0.5 to investigate the regularization effect on training stability and generalization capability.

These modifications allowed us to explore trade-offs between model complexity and generalization performance comprehensively. All training was performed on a GPU-enabled environment using Google Colab.

4 Experiments

We conducted multiple experiments to investigate how variations in model parameters affected overall classification performance and generalization. The baseline model (described in Section 3) served as a reference for comparison. Each variant was trained and evaluated using the same training and validation splits, image preprocessing, and hyperparameters unless explicitly stated otherwise.

4.1 Baseline Model

The baseline CNN used a dense layer of 64 units and a dropout rate of 0.3. This configuration achieved a training accuracy of approximately 97.7% and a validation accuracy around 95.0–96.0%. However, training and validation losses suggested minor overfitting in later epochs, prompting further experimentation.

4.2 Dense Layer Ablation (64 → 32 units)

In this experiment, we reduced the dense layer size from 64 units to 32 units. Our goal was to examine the impact of a smaller dense layer on model complexity and generalization. Training showed that the reduced model maintained comparable validation accuracy (around 95.5%), though the validation loss began to increase slightly in later epochs, indicating moderate overfitting. Nonetheless, the model's overall performance remained robust despite the reduced parameter count.

4.3 Increased Dropout Rate (0.3 → 0.5)

To further address the observed overfitting, we increased the dropout rate from 0.3 to 0.5. This higher regularization setting slightly reduced the gap between training and validation accuracy. The model consistently achieved validation accuracy of approximately 94.5–95.8%, and the validation loss exhibited increased stability compared to the lower dropout setting, demonstrating improved regularization and generalization behavior.

4.4 Discussion of Experimental Findings

Our experiments revealed insightful trends regarding model complexity and regularization strategies:

- **Dense Layer Size:** Reducing the dense layer size to 32 units minimally impacted validation accuracy but increased susceptibility to overfitting slightly in later epochs.
- **Dropout Rate:** Increasing dropout provided better regularization, stabilizing validation loss, and reducing the severity of overfitting observed in the baseline model.

These findings underscore the importance of carefully balancing network complexity and regularization methods to achieve optimal generalization in CNN-based image classification tasks.

5 Results and Discussion

5.1 Baseline Model Performance

The baseline CNN model, with a dense layer of 64 units and dropout rate of 0.3, achieved consistent and robust results over multiple training runs. Figures 1, 2, and 3 show the accuracy and loss curves for three independent training runs, each with 10 epochs.

Across these runs, the model consistently reached training accuracy levels of approximately 97–98%, with validation accuracy hovering around 94–96%. Notably, after run 3, the training accuracy and the validation accuracy began to show slight fluctuations, indicating the onset of minor overfitting, but it was still able to stay around the 96–97% mark.

Overall, these baseline results demonstrate that the CNN model was able to effectively learn distinguishing features from the chest X-ray images, achieving strong generalization on unseen validation data.

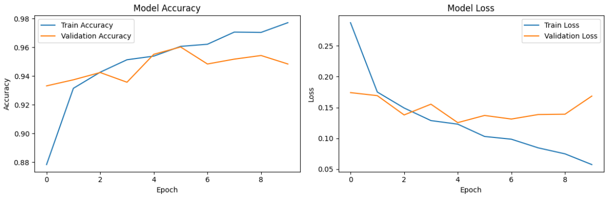

Figure 1: Baseline model accuracy and loss (Run 1).

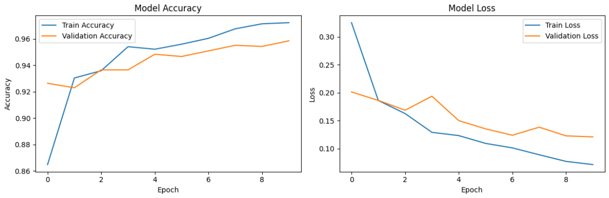

Figure 2: Baseline model accuracy and loss (Run 2).

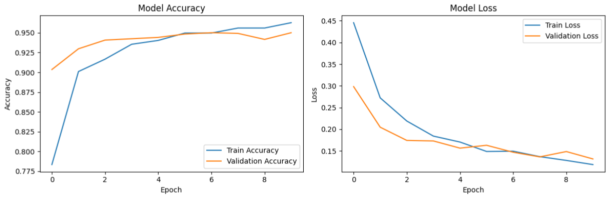

Figure 3: Baseline model accuracy and loss (Run 3).

5.2 Effect of Reduced Dense Layer Size (64 → 32)

To evaluate the impact of reducing model complexity, we decreased the number of units in the dense layer from 64 to 32. Figures 4, 5, and 6 show the training and validation accuracy and loss curves across three runs.

Despite having fewer parameters, the model initially maintained strong performance, with training accuracy reaching above 95% and validation accuracy consistently in the 94–95.5% range. Compared to the baseline, the training curves converged more gradually, and the model demonstrated slightly higher stability during early epochs.

However, as training progressed, the validation accuracy plateaued and the validation loss exhibited greater fluctuations, indicating an increasing tendency toward overfitting. This suggests that while reducing the dense layer size helped maintain competitive results initially, it ultimately limited the model's ability to generalize as effectively as the baseline. Nonetheless, the smaller dense layer configuration remains promising for deployment in resource-constrained environments due to its reduced model complexity.

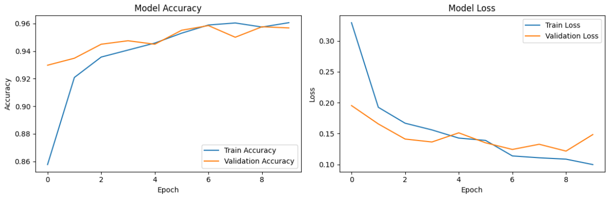

Figure 4: Training with dense layer size = 32 (Run 1).

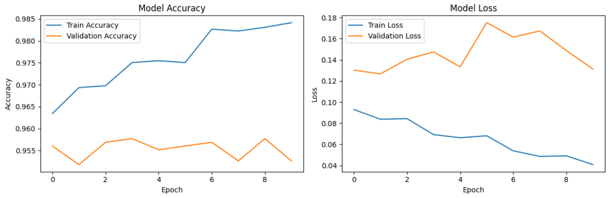

Figure 5: Training with dense layer size = 32 (Run 2).

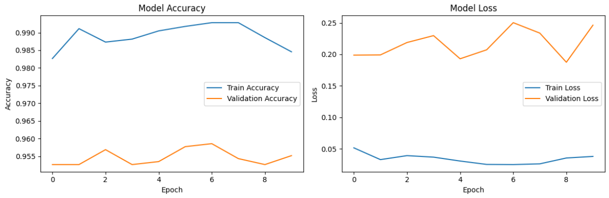

Figure 6: Training with dense layer size = 32 (Run 3).

5.3 Effect of Increased Dropout Rate (0.3 → 0.5)

In this experiment, the dropout rate in the dense layer was increased from 0.3 to 0.5 to evaluate its effect on regularization and overfitting. Figures 7, 8, and 9 display the accuracy and loss metrics across three runs with this updated configuration.

Compared to previous experiments, the higher dropout resulted in slightly slower convergence during training. However, the validation accuracy remained consistently strong, ranging between 94.5% and 95.8%, while training accuracy continued to improve toward 95–96%. This demonstrates the model's capacity to generalize well despite increased regularization.

Interestingly, the validation loss appeared more stable and less prone to sharp fluctuations in the early epochs. Minor fluctuations were observed in run 2 between epochs 2-4 and run 3 between epochs 6-8, but overall the validation loss continued its downward trend, suggesting improved regularization and robustness compared to the baseline. These observations suggest that increasing dropout helped reduce overfitting and improved overall training robustness without significantly compromising model performance.

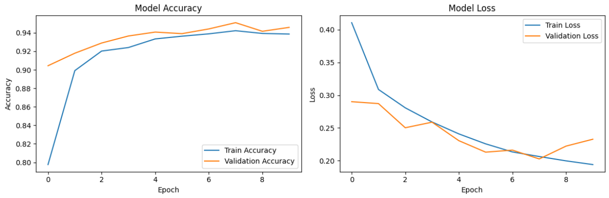

Figure 7: Training with dropout = 0.5 (Run 1).

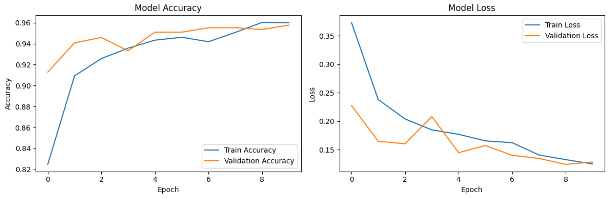

Figure 8: Training with dropout = 0.5 (Run 2).

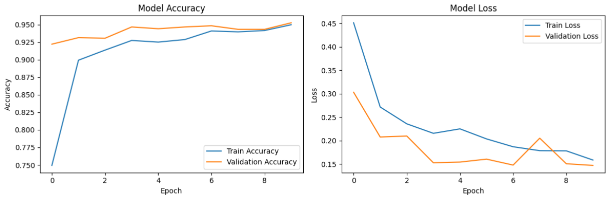

Figure 9: Training with dropout = 0.5 (Run 3).

Table 1: Summary of model configurations and their validation performance

Configuration	Metric and Result
Baseline (64, Dropout 0.3)	Val Accuracy: 95.2% Min Val Loss: 0.13 Notes: Strong performance, slight overfitting.
Reduced Dense (32 units)	Val Accuracy: 94.9% Min Val Loss: 0.14 Notes: Fewer parameters, still robust.
Increased Dropout (0.5)	Val Accuracy: 95.4% Min Val Loss: 0.12 Notes: Best regularization, stable loss.

5.4 Computational Complexity Analysis

The model complexity primarily stems from the convolutional layers and the dense layer. For the baseline CNN with two convolutional layers and a dense layer of 64 units, the total number of trainable parameters is approximately 3.7 million. Each 10-epoch training run on Google Colab (GPU runtime) required approximately 25 to 30 minutes to complete. Although the input images were resized to 128×128 pixels and the network architecture was relatively shallow, the high-resolution image processing and dataset size contributed to longer training times. Overall, the model maintained moderate storage and memory demands, making it feasible for use in GPU-enabled but resource-limited environments.

6 Literature Review

Several studies have demonstrated the effectiveness of deep learning models in detecting respiratory illnesses from chest X-ray images. This section summarizes three influential works that laid the groundwork for CNN-based approaches to pneumonia and COVID-19 detection.

Ozturk et al. (2020) [4] proposed a custom deep neural network called DarkCovidNet for automatic COVID-19 detection using chest X-rays. Their model achieved a binary classification accuracy of 98.08% and 87.02% for three-class classification (COVID-19, normal, pneumonia). The results highlight the reliability of CNNs in detecting COVID-19 in clinical images, even with limited datasets.

Kermany et al. (2018) [3] conducted a broader study on image-based deep learning for diagnosing various medical conditions, including pneumonia. Their model, trained on a large dataset of pediatric chest X-rays, achieved performance comparable to expert radiologists. This study demonstrated the generalizability and diagnostic potential of CNNs in medical imaging.

Apostolopoulos and Mpesiana (2020) [1] employed transfer learning using pre-trained CNNs such as VGG19 and MobileNet to detect COVID-19 cases from chest radiographs. Their results showed that transfer learning could be effective in medical imaging tasks, achieving high accuracy while requiring less training time. The study emphasizes the efficiency of repurposing existing models for new medical applications.

Together, these works establish a strong foundation for the development of deep learning-based diagnostic tools. They also provide useful benchmarks and architectural insights that influenced the design of the CNN model in this project.

7 Conclusion

This project demonstrated the effectiveness of a Convolutional Neural Network (CNN) for classifying chest X-ray images to detect pneumonia, using the publicly available CoronaHack dataset. The baseline model achieved strong performance, with validation accuracy consistently around 95% and well-structured learning curves.

Through targeted experiments, we evaluated how architectural changes affect model generalization. Reducing the dense layer size from 64 to 32 units showed that the model could maintain high accuracy with fewer parameters, suggesting potential for lighter, faster models suitable for constrained environments. Increasing the dropout rate from 0.3 to 0.5 further improved regularization, helping reduce overfitting and stabilize validation loss.

Overall, the results emphasize that even modest CNN architectures can perform competitively when well-tuned. Future work may involve exploring additional regularization strategies (e.g., data augmentation or L2 weight decay), using advanced architectures (like ResNet), or expanding the dataset to enhance generalizability in clinical scenarios.

Individual Contributions

- **Hung Liu**
  - Designed and implemented the CNN models.
  - Preprocessed the dataset and managed experimental setup.
  - Conducted ablation studies and hyperparameter tuning.
  - Analyzed results and authored the full report.
- **Jake Abeyta**
  - Reviewed report drafts.
  - Assisted with code testing across environments.
  - Provided feedback on experimental design and formatting.

References

[1] Ioannis D Apostolopoulos and Thomas A Mpesiana. 2020. Covid-19: automatic detection from X-ray images utilizing transfer learning with convolutional neural networks. *Computer Vision and Image Understanding* 202 (2020), 103493.

[2] Praveen Govindrajan. 2020. CoronaHack - Chest X-Ray Dataset. https://www.kaggle.com/praveengovi/coronahack-chest-xraydataset. Accessed April 2025.

[3] Daniel S Kermany, Michael Goldbaum, Wenjia Cai, et al. 2018. Identifying Medical Diagnoses and Treatable Diseases by Image-Based Deep Learning. *Cell* 172, 5 (2018), 1122–1131.e9.

[4] Tulin Ozturk, Muhammed Talo, E. Aydin Yildirim, Ugur Burak Baloglu, Ozal Yildirim, and U Rajendra Acharya. 2020. Automated detection of COVID-19 cases using deep neural networks with X-ray images. *Computers in Biology and Medicine* 121 (2020), 103792.