Machine Learning Final Project Report: Enhancing Disease Detection in X-Ray Images Using Deep CNN Architectures

Jay Gadhia
Hamzah Al-Bedaiwi

Erik Johnson School of Engineering and Computer Science
The University of Texas at Dallas
JKG210001, HMA210004
{Jay.Gadhia, Hamzah.Al-Bedaiwi}@utdallas.edu

Abstract (Jay: 50%, Hamzah: 50%)

This research primarily aims to utilize deep convolutional neural networks (CNN) to enhance infectious disease detection in chest x-ray imaging. There will be a focus on Pneumonia with application of other diseases, such as Covid-19 diagnosis from two public datasets, with a clear indication of which is which. The overall approach will consist of utilizing three CNN architectures - ResNet, DenseNext, and a custom-designed CNN implemented with transfer learning to leverage pre-trained weights. Each model will be trained, tested, and validated through dataset splits to achieve model effectiveness. The key performance metrics will include accuracy, precision, recall, support, and ROC-AUC. We are proposing creating a custom CNN model including a framework surrounded by transfer learning, taking any prior knowledge as input into the following task.

1. Introduction(Jay:60%,Hamzah:40%)

The use of machine learning(ML) techniques has shaken the healthcare and medical industry by showing promising improvements in diagnosis efficiency. The primary sector we are targeting is the improvements in x-ray imaging and medical image analysis/diagnosis. Among the various types of medical imaging, chest x-rays are very common for diagnosing respiratory diseases. The very common practice is highly effective due to its low-cost nature, speed, and inability to be any sort of invasive to patients. However, it is subject to human error, as it requires visual perception to classify patients between infection types and severity.

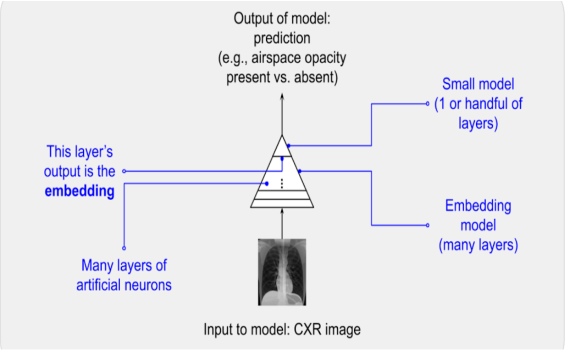


Figure 1: The most common neural network anatomy[9] portraying how a CXR image is processed to answer the clinical questions as the output.

In times of high clinical demand and resource limitation, human error is prone to increase due to medical environmental changes and scarce access to information. Challenges such as these require the introduction of automated ML systems to better assist clinics in providing reliable, fast, and reproducible image assessments.

1.1 Motivation

Respiratory diseases, including Pneumonia and COVID-19, have been considered global public health concerns, raising high alert and social distancing. Illnesses like these require rapid and accurate diagnostics, especially when there is limited information on the infection at hand. Pneumonia, primarily caused by bacterial, viral, and fungal infections, continues to be a global risk with a 12.3 per

100,000[1] death ratio and 41,108 total deaths within 2025[1] alone. Additionally, COVID-19 was a major global pandemic that caused widespread panic and future uncertainty. The disease placed unprecedented strain on the healthcare industry, highlighting the requirements for scalable diagnostic support. A recent test in 2024[2] revealed the chances of a false positive on COVID-19 tests. Between a select group of 11,000 individuals, 1.7% exhibited false positives, meaning 187 were incorrectly diagnosed with the gold standard rapid antigen test. This is where the concept of scalable ML systems is needed to reduce human error. Both Pneumonia and COVID-19 exhibit visible patterns within CXR imaging; however, the visibility can be subtle or ambiguous. The solution of supervised machine learning integrated with convolutional neural networks(CNNs) offers a robust approach in analyzing complex visual patterns from labeled data. Fine-tuned ML models can achieve levels to fully assist in differentiating normal from pathological cases of respiratory diseases, potentially saving radiologists a stressful workload.

1.2 Problem Statement

Machine learning has progressed exponentially with continuous advancements in technology and research. Despite these advancements in areas such as Computer vision and learning techniques, fully implementing- ing machine learning algorithmic systems into the medical field is still miles away. The imbalanced and limited data available and variability in imaging qualities will pose a major obstacle. We hope to achieve a high-accuracy model utilizing CNNs and transfer learning to correctly distinguish between healthy, Pneumonia, and COVID-19 categories. We aim to evaluate whether or not modern-day pre-existing CNN architectures can generalize accurately across thousands of images within the split datasets.

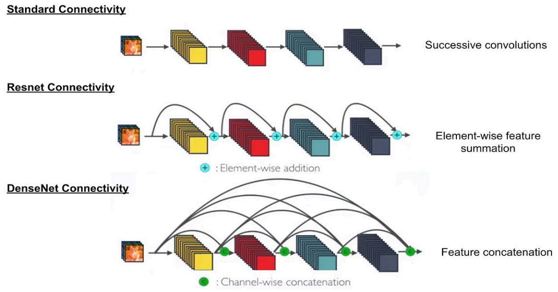


Figure 2: This modeling image[4] portrays the standard CNN architecture alongside the two CNNs we plan to test, ResNet and DenseNet.

1.3 Main Contributions

This work will explore the use of deep convolutional neural networks in supervised learning by classifying CXR images related to two common respiratory infections. The main contributions will include model comparison between two modern CNN architectures, as shown in Figure 2, and a custom-built architecture revolving around transfer learning. We will be pre-processing two public datasets involving thousands of CXR images to train the models and feed them through all three with performance metrics. The dataset will be primarily split between three different categories, training, validation, and testing sets, to examine general capability.

1.4 Background Concepts

Understanding the upcoming project and results requires a basic knowledge of several vital machine learning concepts:

- Supervised Learning: The overall problem is framed such that it is a labeled classification task, where CXR images will be used to train the CNN models to calculate disease class predictions.
- CNN: A specialized neural network designed for image/visual data. The NN applies convolutional filters to detect prominent feature patterns, formally known as kernels.
- Transfer Learning: Deep learning where a trained model is adapted to a target task, viable when labeled data is limited.
- Evaluation Metrics: Accuracy, precision, recall, F1-score, and confusion matrices- performance evaluated across CNNs.

2. Related Work(J: 50%, H: 50%)

In recent years, with the introduction of such pandemic-level diseases, medical ML research has been rapidly growing, specifically utilizing CNN architectures- res within supervised learning of CXR images. The public availability of open-source datasets has allowed for tremendous progression within this domain, leading to systems with strong diagnostic capabilities and potential to fully assist clinics.

One early study[5], also considered one of the most cited, was back in 2018 by Kermany et al., who developed a CNN architecture on a large CXR dataset to detect bacterial and viral pneumonia. The early work demonstrated the feasibility of deep learning models in outperforming computer-aided tools used for diagnostics. The model had achieved over 90% accuracy with the use of early stages of transfer learning. In 2017, Rajpurkar et al. and a group of fellow researchers introduced a CNN architecture known as CheXNet, a 121-layer modified DenseNet trained on the ChestX-ray14 dataset to identify 14 thoracic pathologies, including pneumonia. However, the model had raised major concerns of label inaccuracy and limited scope to solely frontal CXR images.

The rapid spread of the COVID-19 pandemic caused a wildfire of researchers to regroup and repurpose CNN architectures for CXR detection. As public datasets began to be released, Apostolopoulos and Mpesiana(2020) tested several pre-trained models, including MobileNetV2, VGG19, and Inception, on composite datasets of respiratory illness. With the use of transfer learning, they had reported very promising results with accuracy scores exceeding almost 96%. During this period, however, COVID-19 images were a notable limitation as they may have caused significant dataset imbalance. Similarly, Ozturk et al. in 2020, proposed a custom CNN model trained particularly on COVID-positive and normal XR images. The model had achieved a 98% accuracy for binary classification. The downside to this accuracy rating was a very small dataset and the inability to distinguish COVID-19 from pneumonia, showing weak multi-class classification settings.

Transfer Learning has become a standard approach within medical imaging research. This is due to scarce information on newer diseases as they must require quick adaptability in terms of CNN architecture. The limited CXR images need the support of pre-trained medical models to enable faster convergence and improved classification. A major benefit of using transfer learning lies in the capabilities of using low-level image features and adapting them towards a target domain[11].

Comparing our model to current and previous research, most CNNs focus on one major respiratory disease. We are developing a model to be universal with CXR datasets with minor tweaks regarding disease binary classification. Currently, our code focuses on a Pneumonia dataset; however as mentioned, the application of COVID-19 can be applied with minor naming tweaks and dataset download. We are not aiming to outperform prior benchmarks. The work is to provide practical insights into model behavior and establish a foundation for future improvements on resulting metrics.

3. Proposed Approach(J:50%, H: 50%)

Before designing our custom architecture, we implemented a fine-tuned ResNet-50 and DenseNet-121 convolutional neural network (CNN) to classify various chest X-ray images into 2 categories: **NORMAL** and **PNEUMONIA**. Both models were trained on two different datasets located on Kaggle, incorporating strong data augmentation and precise dataset stratification.

3.1 ResNet-50

Our first fine-tuned model, ResNet-50, is a 50-layer deep network influenced by He et al. This CNN solves the common problem of vanishing gradients in deep networks through residual connections, allowing for essentially shortcut paths that allow gradients to propagate more during backpropagation. Using ResNet-50 provided many advantages when fine-tuning, such as effective training due to skip connections, and the CNN being transfer learning ready, as it generally performs well across image classification.

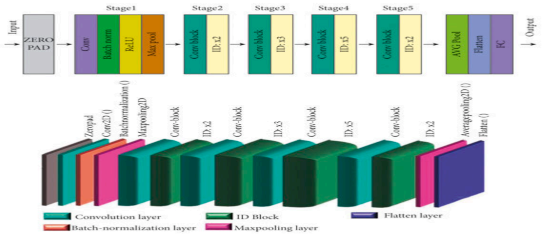


Figure 3: CNN architecture of ResNet-50

Although we considered different image classifiers, such as VGG16 and LeNet, because of their high parameters and shallowness for complex images, respectively, they were not used in the final implementation.

Using two different datasets, they were both compiled using PyTorch's **transforms**. Our dataset images were augmented through horizontal flipping, random rotation, color jitter, and random resized cropping. We split our combined dataset into 70% training, 20% validation, and 10% testing to maintain class balance. We trained our model using Stochastic Gradient Descent (SGD) with weight decay and momentum.

For our loss function, we used the cross-entropy loss function with class weights that were inversely proportional to the frequencies, ensuring the model does not become biased toward the majority class.

$$\mathcal{L}(x, y) = -\sum_{i=1}^{C} w_i \cdot y_i \cdot \log(p_i)$$

3.2 DenseNet-121

Our second fine-tuned model, DenseNet-121, is a CNN architecture containing 121 layers. We utilized the pre-trained CNN available in the PyTorch **torchvision.models** library. DenseNet-121 is known to mitigate vanishing gradient problems and improve parameter efficiency. The CNN archives this through its unique feature of reuse through dense layers. The final fully connected layer of DenseNet-121 was modified to output logistics according to binary classes.

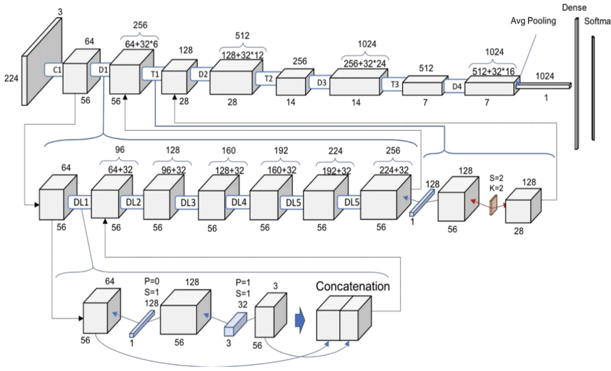


Figure 4: CNN architecture of DesNet-150

Some of the advantages of using DenseNet-121 include the efficient gradient flow, which allows dense connections to ensure a stronger back propagation. DenseNet-121 also contains fewer parameters compared to most other CNNs, like VGG. Lastly, being pre-trained on ImageNet, this CNN generalizes well to medical image tasks, allowing for a strong transfer learning performance.

As for the disadvantages of using DenseNet-121, the computational cost is high, as this CNN requires more memory compared to various other lightweight architectures. DenseNet-121 also contains connections that increase over time, leading to slower training times. Although different architectures, such as VGG16, may be ideal, DenseNet-121 was chosen due to its accuracy and generalization in medical image tasks.

Similar to ResNet-50, to improve generalization, our training pipeline included methods of horizontal flipping, random rotation, color jitter, and random resized cropping for data augmentation.

A weighted loss function was required, assigning higher loss penalties to underrepresented classes. This was due to the imbalance between the **NORMAL** and **PNEUMONIA** samples. Our variable, while training, **WeightedRandomSampler,** was used in each batch to balance class distribution, improving stability.

The loss function we used was a weighted cross-entropy represented as:

$$\mathcal{L} = -w_0 \cdot y_0 \cdot \log(p_0) - w_1 \cdot y_1 \cdot \log(p_1)$$

With w_0 and w_1 representing class weights computed inversely proportional to the class frequencies in the training set.

3.3 Custom Architecture

Our custom architecture, PneumoniaNet, is an effective CNN tailored specifically for medical imaging tasks, such as pneumonia detection. The custom CNN contains four convolutional blocks, with channel sizes increasing over time, going from 32, 64, 128, and 56. Each block includes:

- A convolutional layer with padding
- Batch normalization (with the intent of training stability)
- Max pooling for spatial downsampling
- ReLu non-literarity

- Spatial dropout (with intent to regularize training and prevent overfitting)

Upon completion of the feature extraction, spatial dimensions are reduced through a global average pooling layer, following a fully connected classifier containing dropout layers. Lastly, a linear layer outputs all class logits.

Some of the advantages of our custom-built architecture opposed to pre-built CNNs are that our's is designed with medical imaging in mind, and the smaller kernel sizes and dropout make it realistic for small datasets. As for the disadvantages, our custom model requires more epochs to converge and to be compared with pre-trained models. Performance is also a limiting factor, as without access to large-scale data, our model may not grow as expected.

We also used transfer learning with ResNet-18, which is a model that has been pre-trained on ImageNet. While some of the earlier convolutional layers were partially frozen, mostly to retain basic image features, the final fully connected layers were replaced with a smaller custom classifier for binary classification.

Some of the advantages of transfer learning include faster convergence, and higher accuracy from the pretrained weights, and a stronger baseline performance. Some of the disadvantages, however, include high memory usage/ complexity, and being less transparent when compared to custom networks, as they are less adaptable to domain-specific constraints.

Our loss function can be defined as:

$$\mathcal{L}(x, y) = -\sum_i y_i \log(\text{softmax}(x_i))$$

This function is ideal for multi-classification tasks. Because of our binary classification, it simplifies the log-loss between any predicted probabilities and true labels. Class weights were also calculated from the training data and integrated into the loss function to penalize miscalculations. This was also to address any class imbalance between **NORMAL** and **PNEUMONIA** cases.

4. Experiments(J: 40%, H: 60%)

For all 3 CNNs, ResNet-50, DenseNet-121, and PneumoniaNet, we constructed a custom dataset by merging two different datasets, totalling nearly 10,000 images of x-rays. Each dataset contained two classes: **NORMAL** and **PNEUMONIA**.

After combining the datasets and cleaning out/ skipping any corrupt files, the resulting dataset, while training all three models, was split into:

- 70% training set
- 10% validation set
- 20% testing set

The distribution in the training set showed class imbalance; therefore, it was mitigated using a weighted random sampler and class-weighted cross-entropy loss function.

4.1 ResNet-50

We applied various types of transformations:

- Train set
  - Random resized crop to 244x244
  - Horizontal flip
  - Rotation of +- 15 degrees
  - Brightness contrast
  - Normalization
- Val & Test sets
  - Resize to 256x256
  - Center crop to 244x244
  - Normalization

We fine-tuned a ResNet-50 model that was initialized with pretrained ImageNet weights. After the training, the final layer was replaced with output for binary classes {**NORMAL**, **PNEUMONIA**}

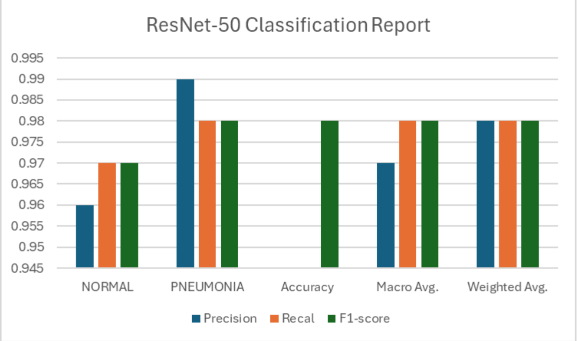


Figure 5: Classification report for ResNet-50

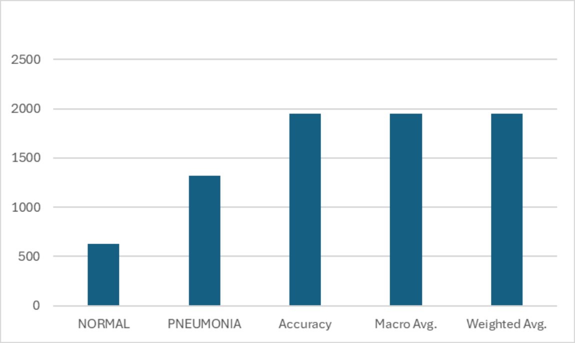


Figure 6: Support report for ResNet-50 & DenseNet-121

Hyperparameters included:

- Learning rate: 0.001
- Weight decay: 1e-4
- Precision: Mixed-precision training
- Epochs: 25
- Batch size: 32
- Loss function: Weighted CrossEntropyLoss depending on training class frequencies
- Scheduler: StepLR (step size = 7 epochs, gamma =0.1)
- Optimizer: SGD with momentum = 0.9

It is important to note the steady convergence in both training and validation accuracy, with a validation accuracy peak around epoch N. Overfitting was minimal, as training loss consistently declined, while validation loss fell slightly towards the end.

4.2 DenseNet-121

To address the issue of class imbalance, we used class-weighted loss functions and WeightedRandomSampler for the training data loader.

Data Preprocessing:

- Images were center-cropped or resized to 2224x224
- Augmentations for training included horizontal flipping, color jitter, slight rotation, and a resized crop
- Normalization

We used DenseNet-121, a pre-trained CNN, and replaced its classifier head with a binary class output layer {**NORMAL** vs. **PNEUMONIA**}

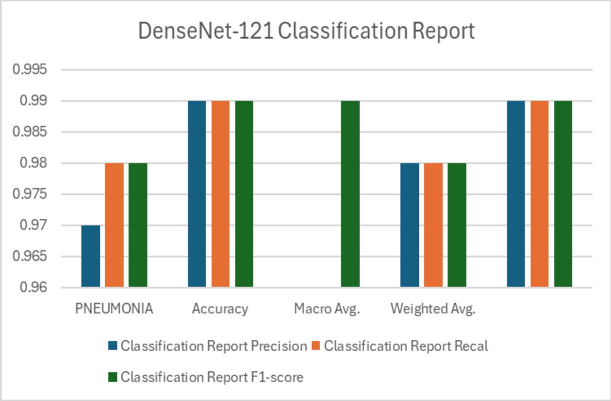


Figure 7: Classification report for DenseNet-121

Hyperparameters included:

- Optimizer: SGD with momentum
- Momentum: 0.9
- Learning rate: 0.001
- Weight decay: 1e-4
- Mixed precision
- Epochs: 25
- Batch size: 32
- Loss function: CrossEntropyLoss with class weights
- Scheduler: StepLR (step size = 7 epochs, gamma =0.1)

After training for 25 epochs, the model achieved the best validation accuracy. The validation and training loss curves show steady convergence without major signs of overfitting, proving that the data augmentation techniques were effective.

4.3 PneumoniaNet

Using 2 datasets that were merged, and later split into training, validation, and testing, the class distribution was preserved to maintain balance between our classifiers{**Normal**, **Pneumonia**}.

To further enhance generalization, training images were augmented with random resized crops, horizontal flips, rotations, and color jittering. The validation and test images were center-cropped after resizing.

Hyperparameters included:

- Batch size: 16
- Learning rate: 0.001
- Optimizer: Adam
- DropoutL Up to 0.5 in classifier
- Loss function: Cross-entropy loss

- Class balancing: WeightedRandomSampler to address imbalance in the training set
- Training Epochs: 30

Plots were generated for training and validation loss over time, visualizing convergence and signs of overfitting or underfitting.

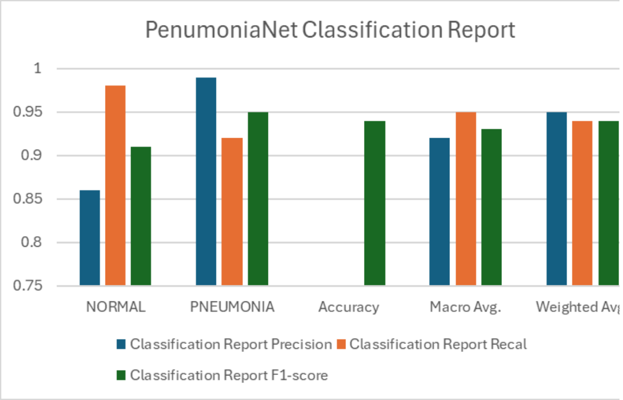


Figure 8: PneumoniaNet Classification Report

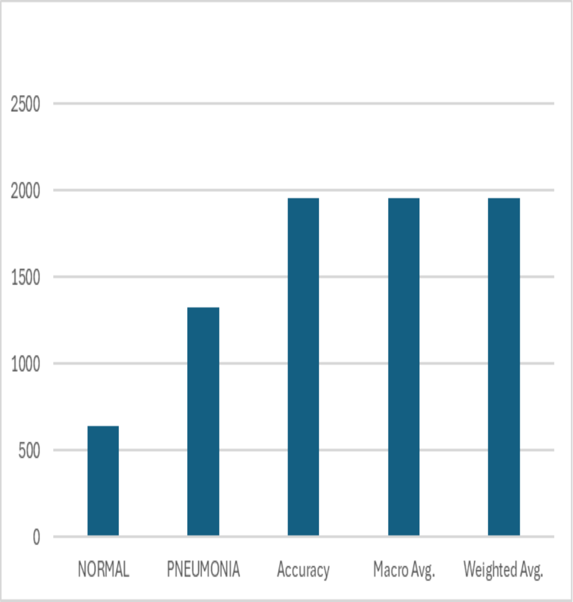


Figure 9: Support report for PneumoniaNet

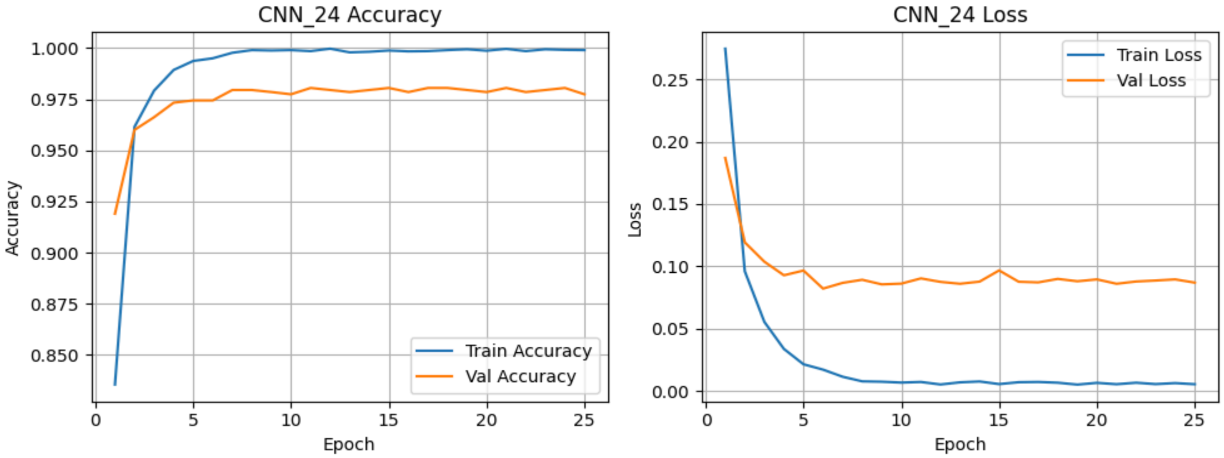
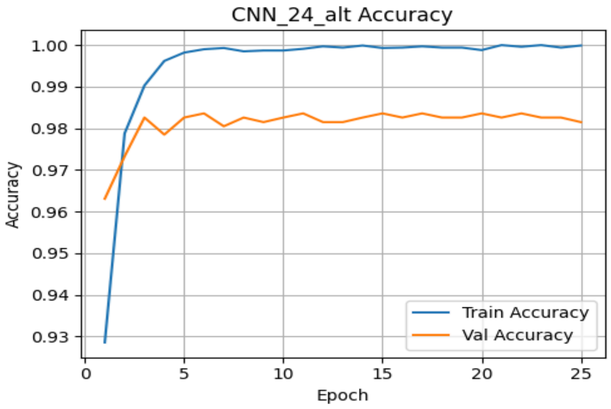


Figure 5: The ResNet convolutional neural network architecture. The two graphs above illustrate the accuracy and loss functions between the validation and testing datasets.



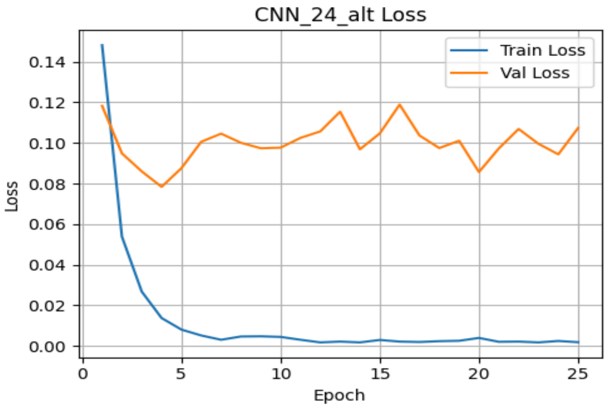


Figure 6: The DenseNet convolutional neural network architecture. The two graphs above illustrate the accuracy and loss functions between the validation and testing datasets.

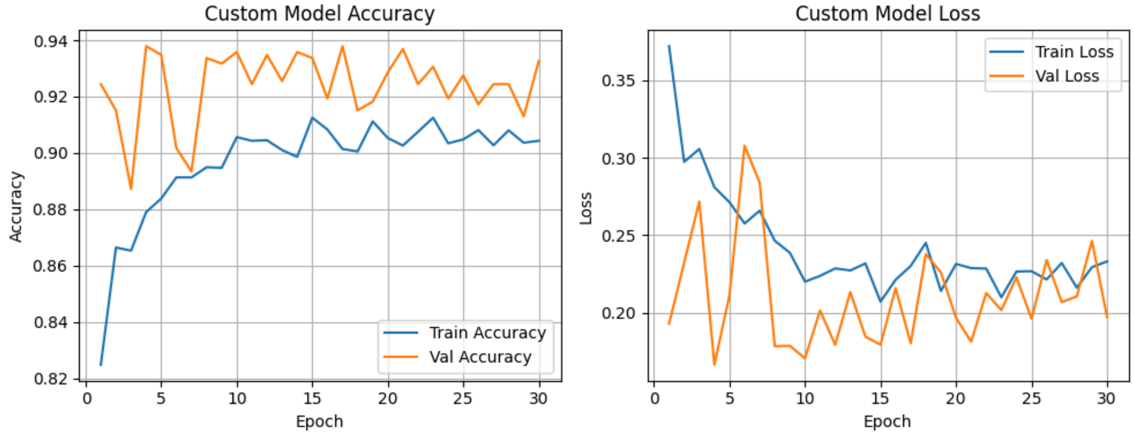


Figure 7: The PneumoniaNet convolutional neural network architecture. The two graphs above illustrate the accuracy and loss functions between the validation and testing datasets.

CNN / Metrics	Accuracy	Precision	Recall	Specificity	F1-Score
ResNet	97.8%	98.6%	98.2%	97.1%	98.4%
DenseNet	98.6%	99.1%	98.8%	98.1%	98.9%
CustomNet	94.0%	98.9%	92.1%	97.9%	95.4%

Figure 8: The summary table above shows all performance metrics of each convolutional neural network architecture. The same two datasets were split into validation, testing, and training, and processed through each NN model. DenseNet indicates the highest performance metrics in every aspect measured through each epoch and calculated from the confusion matrices. However, one key note is that the PneumoniaNet was able to outperform ResNet in terms of precision and specificity, and fell just short of recall.

4.4 Confusion Matrix

As the research focuses on supervised classification models, a confusion matrix is drawn out to calculate various metrics to determine the efficiency of your model. The matrix itself compares 4 different values together: the predicted positives and negatives, and the actual positives and negatives. The equation below is used to find such metrics:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

$$Precision = \frac{TP}{TP + FP}$$

$$Recall = \frac{TP}{TP + FN}$$

$$F1 = \frac{Precision * Recall}{Precision + Recall} * 2$$

$$Specificity = \frac{TN}{TN + FP}$$

I have calculated 3 different confusion matrices, one for each CNN architecture utilized throughout our work. The following three figures will portray the resulting CMs, as well as provide an example CM to help establish basic knowledge of each component:

Example CM	PN	PP
AN	True -	False +

Figure 10: Example confusion matrix

ResNet CM	PN	PP
AN	613	18
AP	23	1295

Figure 11: ResNet confusion matrix

DenseNet CM	PN	PP
AN	619	12
AP	16	1303

Figure 12:DenseNet confusion matrix

Custom CM	PN	PP
AN	621	13
AP	105	1220

Figure 13: CustomNet confusion matrix

5. Conclusion

Although we were unable to fully outperform ResNet and DenseNet, the overall research yielded some success as we were able to outperform ResNet in two of five performance metrics through supervised transfer learning. A major learning curve was tackled, not only learning transfer learning, but applying it to real-world solutions, potentially benefiting the medical and clinical fields. The highlight of this research is that my partner and I were able to yield a 94% accuracy with datasets containing over 6000+ CXR images, which, of course, can be fine-tuned. We were able to gain valuable experience in utilizing transfer learning, making it much easier to train datasets with limited resources. As we plan to dive deeper into deep convolutional neural networks, we can take this valuable experience and apply it going forward, hopefully creating ground-breaking discoveries.

6. References

[1] CDC, “FastStats - Pneumonia,” Centers for Disease Control and Prevention, 2019. https://www.cdc.gov/nchs/fastats/pneumonia.htm

[2] K. Kahn, “How Common Are False Positive Rapid COVID Tests?” Medpagetoday.com, Feb. 21, 2024. https://www.medpagetoday.com/infectiousdisease/covid19/108837

[3] Neville, “Why Isn’t DenseNet Adopted as Extensive as ResNet? - Neville - Medium,” Medium, Jul. 15, 2023. https://cvinvolution.medium.com/why-isnt-densenet-adopted-as-extensive-as-resnet-1bee84101160 (accessed May 16, 2025).

[4] D. S. Kermany et al., “Identifying Medical Diagnoses and Treatable Diseases by Image-Based Deep Learning,” Cell, vol. 172, no. 5, pp. 1122-1131.e9, Feb. 2018, doi: https://doi.org/10.1016/j.cell.2018.02.010.

[5] T. Ozturk, M. Talo, E. A. Yildirim, U. B. Baloglu, O. Yildirim, and U. Rajendra Acharya, “Automated detection of COVID-19 cases using deep neural networks with X-ray images,” Computers in Biology and Medicine, vol. 121, Apr. 2020, doi: https://doi.org/10.1016/j.compbiomed.2020.103792.

[6] S. Das, “Implementing DenseNet-121 in PyTorch: A Step-by-Step Guide,” deepkapha notes, Mar. 19, 2023. https://medium.com/deepkapha-notes/implementing-densenet-121-in-pytorch-a-step-by-step-guide-c0c2625c2a60

[7] A. J. Kalita et al., “Artificial Intelligence in Diagnostic Medical Image Processing for Advanced Healthcare Applications,” Biological and medical physics series, pp. 1–61, Jan. 2024, doi: https://doi.org/10.1007/978-981-97-5345-1_1.

[8] A. B. Sellergren et al., “Simplified Transfer Learning for Chest Radiography Models Using Less Data,” Radiology, Jul. 2022, doi: https://doi.org/10.1148/radiol.212482.

[9] Google, “Simplified Transfer Learning for Chest Radiography Model Development,” Research.google, 2022. https://research.google/blog/simplified-transfer-learni

ng-for-chest-radiography-model-development/ (accessed 2025).

[10] A. Jain, “Deep Learning Architecture 7: DenseNet - Abhishek Jain - Medium,” Medium, Dec. 17, 2024. https://medium.com/@abhishekjainindore24/deep-learning-architecture-7-densenet-feee44d57f89

[11] “What is Transfer Learning? - Transfer Learning in Machine Learning Explained - AWS,” Amazon Web Services, Inc. https://aws.amazon.com/what-is/transfer-learning/