COMS 4030A ACML Project – Report

Lucian Irsigler 2621933, Prashan Rajaratnam 2436566, Banzile Nhlebela 2571291, Pramit Kanji 2551233

May 30, 2025

1 Introduction

Convolutional Neural Networks (CNNs) have seen many developments in recent years since the inclusion of GPUs in machine learning which demonstrated that CNN models can become very accurate and balance overfitting when training to recognise images in large datasets. In our project we will demonstrate the application of CNNs in healthcare spesifically on Pneumonia classification using chest x-rays. Pneumonia is a contagious respiratory condition characterised by inflammation of the lung parenchyma. This inflammation leads to the accumulation of fluid and immune cells within the alveoli. Chest X-rays, are a key tool in the diagnosis of pneumonia as these images visualise pneumonia typically presenting as areas of increased opacity, known as consolidations, indicating regions where air in the alveoli has been replaced by fluid or inflammatory exudate. We chose a CNN because of the strong assumption that is made about the nature of medical images such as the locality of pixel dependencies which CNNs take advantage of by detecting features regardless of their position in the image, this is a property known as translation invariance. This is particularly beneficial in medical imaging, where pathological features may appear in different locations across patients.

2 The Dataset

For our project on pneumonia detection, we employed the publicly available dataset **Labeled Optical Coherence Tomography (OCT) and Chest X-Ray Images for Classification** Kermany et al. [2018]. The chest X-ray component of the dataset comprises 5,856 anterior-posterior (AP) radiographic images of pediatric patients aged between one to five years, collected from Guangzhou Women and Children's Medical Center in China. Each image was labeled by physicians into one of three categories: NORMAL, PNEUMONIA_BACTERIA, or PNEUMONIA_VIRUS. Specifically, the dataset includes 1,583 images labeled as NORMAL, 2,780 as PNEUMONIA_BACTERIA, and 1,493 as PNEUMONIA_VIRUS, but these are just labelled as PNEUMONIA collectively. In terms of ratios, 27% of the data is NORMAL, and 73% is PNEUMONIA.

The dataset is initially organized into predefined training and testing subsets with a small amount of data set aside for validation.We took the same dataset, but rather merged all the separate folders into one, so that we could split the data as we saw fit. There is a total of 5,856 images in the dataset.The dimensions of the images in the dataset are inconsistent. Additionally, for the PNEUMONIA data, there are multiple scans for the same person, as indicated by different filenames sharing the same 'personX' label in their names, where X represents a number. The dataset thus has a data leakage problem.

2.1 Data Cleaning

For the analysis of chest x-ray images, all chest radiographs were initially screened for quality control by removing all low quality or unreadable scans. The diagnoses for the images were then graded by two expert physicians before being cleared for training the CNN system. This data cleaning was done by the dataset creators.

2.2 Dataset Partitioning and Leakage Prevention

Patient-wise splitting was applied to prevent data leakage.For PNEUMONIA, each X-ray filename includes a patient ID. A list containing all the unique patient IDs was created. When assigning data to the train,validation, or test dataset split, the program would ensure that if a dataset has some patient files for PNEUMONIA, then all files for that patient would be in that dataset only. Additionally, a function was created to check the relative data loaders for train,validation, and test for any data leakage.

3 The Architecture

The CNN architecture implemented for classifying chest X-ray images into NORMAL and PNEUMONIA categories is thoughtfully designed to balance performance and computational efficiency inspired by VassiliPh [2023]. The network comprises three convolutional blocks, each followed by batch normalisation, ReLU activation, max pooling, and dropout layers, culminating in fully connected layers for classification.

3.1 Convolutional Layers and Feature Extraction

Convolutional layers are fundamental in extracting spatial hierarchies of features from input images. By applying learnable filters, these layers capture local patterns such as edges, textures, and shapes, which are crucial for identifying pneumonia indicators like lung opacities in chest X-rays. The use of multiple convolutional layers allows the network to learn increasingly abstract representations, enhancing its ability to distinguish between normal and pathological cases.

3.2 Batch Normalisation for Training Stability

Incorporating batch normalisation after each convolutional layer standardises the inputs to subsequent layers, mitigating issues related to internal covariate shift. This normalisation accelerates training, allows for higher learning rates, and reduces sensitivity to initialisation, leading to improved convergence and generalisation. We used a batch size of 64 for most experiments, which provided stable convergence and reliable generalisation. To evaluate its impact, we also tested a batch size of 32, which resulted in a lower test accuracy of 86.21%. While precision remained high, the recall dropped to 0.82, indicating that the model missed more pneumonia cases. The F1-score also declined, reflecting a trade-off in predictive balance. These results suggest that while smaller batch sizes can introduce more gradient noise (potentially improving generalisation), they may also lead to instability and poorer recall in imbalanced medical datasets.

3.3 ReLU Activation

The Rectified Linear Unit (ReLU) activation function introduces non-linearity into the network, enabling it to learn complex mappings between inputs and outputs. ReLU is computationally efficient and helps alleviate the vanishing gradient problem, facilitating the training of deeper networks.

3.4 Max Pooling

Max pooling layers reduce the spatial dimensions of feature maps, retaining the most important features while decreasing computational load. This downsampling also provides a form of translation invariance, ensuring that the network's predictions are robust to minor shifts in the input images.

3.5 Dropout

To prevent overfitting, dropout layers randomly deactivate a fraction of neurons during training. This regularisation technique forces the network to learn redundant representations, enhancing its ability to generalise to unseen data.

3.6 Hyperparameter Tuning

We experimented with various configurations to improve model performance. In particular, we tested initial learning rates of 0.01 and 0.001. While 0.001 provided the best balance between convergence speed and stability, the higher rate of 0.01 led to erratic training and suboptimal results. Specifically, it caused the model to prioritise recall over precision, yielding a lower precision of 0.78 for normal cases despite high recall for pneumonia. The overall accuracy also dropped to 91.00%, confirming that 0.01 was too aggressive. These learning rates were further adapted during training using the Adam (Adaptive Moment Estimation) optimiser, which helps dynamically adjust step sizes.

We also explored increasing the depth of the network, but found diminishing returns and increased overfitting, reinforcing the importance of a balanced architecture and appropriate regularisation.

3.7 Adaptive Average Pooling for Fixed-Size Output

Adaptive average pooling ensures that the output of the convolutional blocks has a consistent spatial dimension, regardless of the input size. This consistency is vital for feeding the feature maps into the fully connected layers, which require fixed-size inputs.

3.8 Pytorch

PyTorch was selected as the deep learning framework for this project due to its dynamic computation graph and intuitive interface, which are particularly beneficial for research and development in medical imaging. PyTorch's dynamic graph construction allows for flexible model building, facilitating experimentation with different architectures and hyperparameters.

Moreover, PyTorch seamlessly integrates with Python's ecosystem and supports GPU acceleration, enabling efficient processing of large datasets like chest X-rays. Its extensive community and comprehensive documentation further provide valuable resources for implementing and troubleshooting complex models.

4 Details of the Learning

To train the CNN model for pneumonia classification, we configured a training pipeline that ensures fairness, robustness, and proper evaluation.

4.1 Training Configuration

The model was trained using the following hyperparameters:

- **Epochs:** 10
- **Batch size:** 64
- **Image size:** 56×56 (resized from original)
- **Learning rate:** 0.001
- **Train-validation-test split:** 60%-20%-20%
- **Weighted sampling:** Enabled to address class imbalance

4.2 Custom Dataset Class and Transformations

We implemented a custom PyTorch Dataset class that loads and labels images based on their path. Images are converted to RGB and then transformed using a series of augmentations (e.g., normalisation, resizing) to improve model generalisation and consistency.

4.3 Loss Function and Optimisation

The model was trained using the Cross-Entropy Loss function, optimized with the Adam optimizer. This combination is effective for binary classification tasks and handles learning dynamics well across different feature distributions.

The loss function was designed to address class imbalance by incorporating class weights. This ensures that misclassification of the NORMAL class contribute more significantly to the loss, allowing the model to better recognize and correct these errors.

4.4 Regularisation

To mitigate overfitting:

- **Dropout layers** were inserted after each convolutional block.
- **Batch normalisation** was applied to stabilise and accelerate training.

5 Training Graphs

The graph below illustrates the training and validation loss over the course of training.

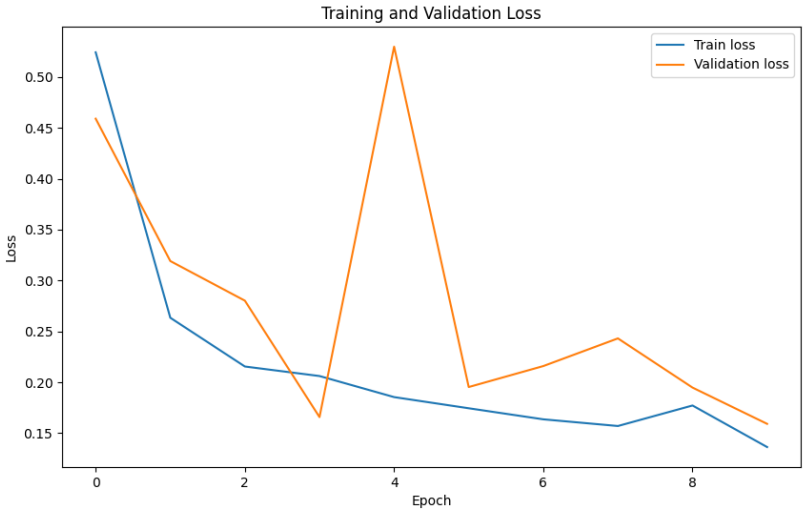

Figure 1: Training and Validation Loss over Epochs

As observed in Figure 1, the training loss consistently decreases across epochs, indicating that the model is learning effectively from the training data. The validation loss follows a similar downward trend, demonstrating good generalisation and no signs of severe overfitting. However, the validation loss exhibits a sharp spike at epoch 4 before quickly recovering.

This temporary increase could be attributed to a number of factors:

- **Weight update instability:** A large gradient update might have briefly pushed the model into a suboptimal region of the parameter space.
- **Batch variability:** The spike may result from statistical variation in a particular validation batch that was not representative.

Importantly, the model quickly recovered and continued to improve, suggesting robustness in the optimisation process. This also reflects the benefits of techniques like batch normalisation and dropout in stabilising learning.

6 Results

The performance of the model was assessed using accuracy, precision, recall, and F1-score. Confusion matrices were generated for both validation and test sets to understand class-wise prediction performance. The model achieved a strong overall performance on the test set, with an accuracy of 93.90%, precision of 0.9798, recall of 0.9349, and F1-score of 0.9568. These metrics suggest the model is both precise in its predictions and sensitive to detecting true cases, which is especially important in medical diagnosis.

Class	Precision	Recall	F1-score	Support
NORMAL	0.85	0.95	0.90	318
PNEUMONIA	0.98	0.93	0.96	829
Accuracy		**93.90%**		

Table 1: Classification Report on the Test Set

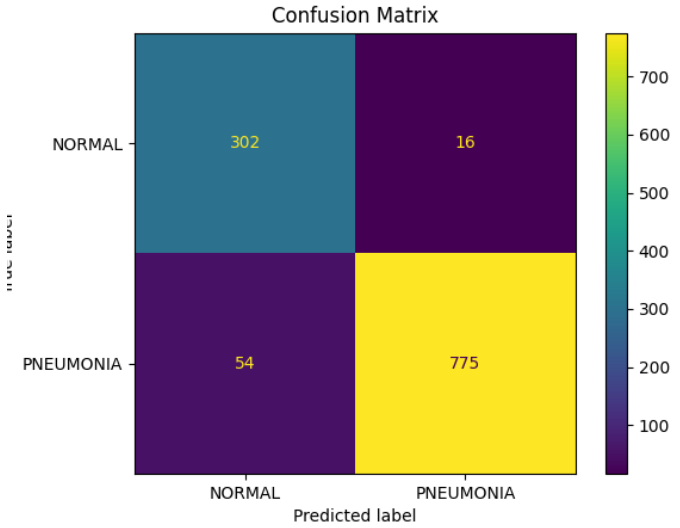

Figure 2: Confusion Matrix on the Test Set

The confusion matrix in Figure 2 confirms this performance. Out of 318 normal cases, only 16 were misclassified as pneumonia, while 302 were correctly identified. In contrast, the model identified 775 out of 829 pneumonia cases correctly, with 54 false negatives.

7 Discussion

The results indicate that the model performed strongly on the pneumonia classification task, achieving an accuracy of 93.90% on the test set, along with high precision and F1-scores for both classes. The model exhibits particularly high

precision for pneumonia cases (0.98), which is critical in reducing false positives and avoiding unnecessary treatment. It also demonstrates strong recall for normal cases (0.95), meaning it is highly effective at correctly identifying healthy patients. However, recall for pneumonia cases is slightly lower (0.93), indicating a minor tendency to miss some pneumonia instances. Overall, the model shows a well-balanced and robust discriminatory ability, especially in prioritising pneumonia detection, while still maintaining reliable performance across both classes. Nevertheless, when interpreting these outcomes, it is essential to consider the influence of class imbalance in the dataset, which may have contributed to the slight disparity in recall rates.

7.1 Impact of Class Imbalance

Figure 3, shows the class distributions across the training, validation, and test sets. In all three subsets, pneumonia images significantly outnumber normal images. For instance, in the training set, there are roughly 2.7 times more pneumonia images than normal images. This imbalance may bias the model towards the majority class, potentially leading to lower sensitivity for the underrepresented normal class.

While the model shows high recall for the normal class (0.95), the confusion matrix reveals that it still misclassifies 54 pneumonia cases as normal. In a clinical context, such false negatives could be critical, as missed pneumonia diagnoses can delay treatment and increase patient risk.

7.2 Performance and Generalisation

Despite the imbalance, the use of weighted sampling and regularisation techniques like dropout and batch normalisation appears to have mitigated overfitting and helped the model generalise well. The validation loss trend is largely stable (see Figure 1), with a temporary spike, likely due to batch variability or local optimisation challenges, which the model quickly recovers from.

7.3 Limitations and Future Work

One limitation is the relatively small number of normal cases available for training, which could impact robustness across different populations. Moreover, the model might benefit from techniques such as:

- Synthetic oversampling (e.g., SMOTE) or data augmentation targeted at the minority class.
- Incorporation of transfer learning using pretrained CNNs (e.g., ResNet, DenseNet).
- Calibrated thresholds or cost-sensitive loss functions to better manage false negatives.

In future work, evaluating the model on an external dataset or with additional patient demographics would also help assess its broader clinical applicability.

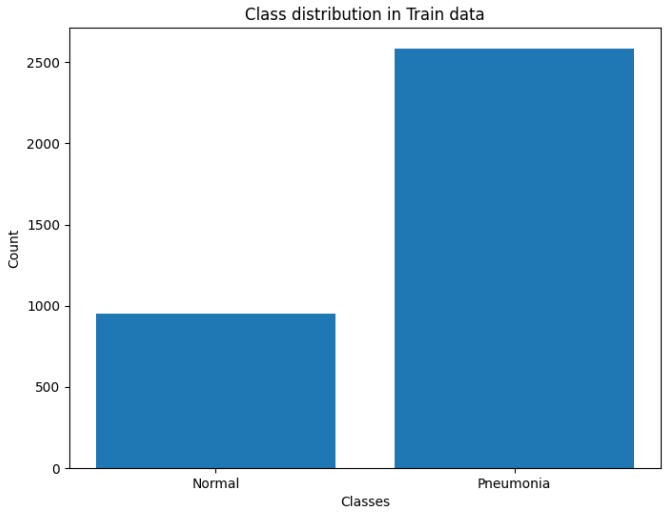

Figure 3: Class distribution in Training Data

References

Kermany, D., Zhang, K., & Goldbaum, M. (2018). *Labeled Optical Coherence Tomography (OCT) and Chest X-Ray Images for Classification*. Mendeley Data, v2. https://data.mendeley.com/datasets/rscbjbr9sj/2

VassiliPh , (2023) , *Chest XRay Images | PyTorch Step-by-step | Acc 94%*. Kaggle Code, https://www.kaggle.com/code/vassiliph/chest-xray-images-pytorch-step-by-step-acc-94#Pneumonia-Chest-X-Ray-Images-Dataset,-PyTorch-Step-by-Step,-94%25-Accuracy